

**RANCANG BANGUN KAMERA EMBEDDED BERBASIS
MIKROKONTROLER ATMEGA 32 DAN OMNIVISION OV6620
PADA SIMULATOR MENEMBAK**

SKRIPSI



**Disusun oleh :
Muhammad Rusdi Azhari
NIM. 0712907**

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2014**

LEMBAR PERSETUJUAN

**RANCANG BANGUN KAMERA EMBEDDED BERBASIS
MIKROKONTROLER ATMEGA 32 DAN OMNIVISION OV6620
PADA SIMULATOR MENEMBAK**

SKRIPSI

*Disusun dan diajukan untuk melengkapi dan memenuhi persyaratan
guna mencapai gelar Sarjana Teknik*

**Disusun oleh :
Muhammad Rusdi Azhari
NIM. 0712907**

**Mengetahui,
Ketua Program Studi Teknik Elektro S-1**

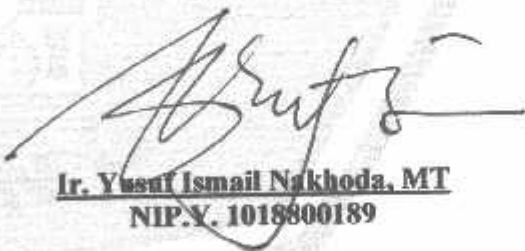

**M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358**

Diperiksa dan Disetujui,

Dosen Pembimbing I


**Dr. Eng. Aryanto Soetedjo, ST, MT
NIP.P. 1030800417**

Dosen Pembimbing II


**Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 1018800189**

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2014**

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : Muhammad Rusdi Azhari

NIM : 0712907

Program Studi : Teknik Elektro S-1

Konsentrasi : Teknik Elektronika

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, 17 November 2014

Yang membuat Pernyataan,



Muh. Rusdi Azhari
NIM : 0712907

ABSTRAK

RANCANG BANGUN KAMERA EMBEDDED BERBASIS MIKROKONTROLER ATMEGA 32 DAN OMNIVISION OV6620 PADA SIMULATOR MENEMBAK

Muhammad Rusdi Azhari, NIM 0712907

Dosen Pembimbing : Dr. Eng. Aryunto Soetedjo, ST, MT dan
Ir. Yusuf Ismail Nakhoda, MT

Kamera embedded merupakan salah satu sistem pengolahan citra yang dapat berdiri sendiri, karena sudah di benamkan di dalamnya prosessor yang dapat memproses citra dan dapat mengambil keputusan atau perintah yang akan diproses selanjutnya.

Dalam penelitian ini telah di realisasikan suatu alat kamera embedded yang dapat di aplikasikan pada simulator menembak. Mikrokontroler mendeteksi warna titik sinar laser dari output citra sensor kamera. Jika terdeteksi mikrokontroler mengirim koordinat posisi titik sinar laser ke komputer. Untuk mendapatkan nilai skor hasil tembak.

Kamera Embedded ini bekerja dengan baik pada resolusi 176 x 144 piksel. Atau pada opsi ke 2, format resolusi sensor kamera OV6620. Pada simulator menembak, nilai posisi titik koordinat diambil dari nilai *counter0* pada ATmega 32 untuk baris dan nilai *counter1* pada ATmega 32 untuk kolom. Setelah perbandingan data masing-masing warna primer pada sensor kamera dengan data 3 warna primer yang di setting untuk sinar laser sama. Dimana data 3 warna primer untuk deteksi sinar laser pada pengujian ini adalah :Merah = 199 s/d 240, Biru = 208 s/d 240, Hijau = 208 s/d 240. Mikrokontroler ATmega 8 berfungsi mengaktifkan bersama antara sensor kamera dan mikrokontroler ATmega 32, menggunakan rangkaian transistor 2N3904 sebagai saklar dimana kaki kolektor terhubung ke pin reset ATmega 32 dengan tingkat kesalahan (*error*) saat saturasi Vce (Tegangan Kolektor-Emitor) 0,01% dan Ic (Arus Kolektor) 0,01%.

Kata Kunci : Kamera Embedded, Simulator Menembak, Laser, Mikrokontroler



KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala limpahan rahmat-Nya sehingga penelitian berjudul Rancang Bangun Kamera Embedded Berbasis Mikrokontroler ATmega 32 dan Omnivision OV6620 Pada Simulator Menembak dapat terselesaikan.

Penelitian ini dibuat untuk memenuhi salah satu syarat dalam memperoleh gelar sarjana teknik. Ucapan terima kasih yang sebesar-besarnya kami sampaikan pada :

1. Bapak Ir. Soeparno Jiwo, MT selaku Rektor ITN Malang.
2. Bapak Ir. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri ITN Malang.
3. Bapak M. Ibrahim Ashari, ST, MT selaku Ketua Program Studi Teknik Elektro S-1 ITN Malang.
4. Bapak Dr. Eng. Aryunto Soetedjo, ST, MT selaku Dosen Pembimbing I.
5. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Dosen Pembimbing II.
6. Kedua orang tua yang telah memberi motivasi dalam penyelesaian skripsi ini.
7. Rekan-rekan Asisten Laboratorium Robotika dan Elektronika Digital ITN Malang.
8. Semua Pihak yang telah membantu dalam penulisan dan penyusunan penelitian ini.

Penulis menyadari bahwa penelitian ini masih jauh dari sempurna, untuk itu kritik dan saran dari pembaca sangat penulis harapkan untuk perbaikan penelitian ini.

Malang, Februari 2014

Penulis

DAFTAR ISI

Lembar Persetujuan	i
Surat Pernyataan Orisinalitas.....	ii
Abstrak.....	iii
Kata Pengantar.....	iv
Daftar Isi	v
Daftar Gambar	vii
Daftar Tabel	ix
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi Penelitian.....	2
1.7 Sistematika Penulisan	2
BAB II LANDASAN TEORI	
2.1 Pengolahan Citra digital.....	4
2.1.1 Model Citra Digital.....	5
2.1.2 Pencuplikan dan Kuantisasi.....	6
2.1.3 Sistem Koordinat Pada Sistem Citra Digital	7
2.2 Modul Sensor Kamera OV6620.....	8
2.2.1 Akurasi Nilai Data Digital OV6620	11
2.2.2 Format Data Keluaran OV6620.....	11
2.3 Mikrokontroler ATmega 32	12
2.3.1 Interupsi Mikrokontroler ATmega 32.....	13
2.3.2 <i>Timer / Counter 0</i> Mikrokontroler ATmega 32	15
2.3.3 <i>Timer / Counter 1</i> Mikrokontroler ATmega 32	19
2.3.4 USART Mikrokontroler ATmega 32	26
2.4 Mikrokontroler ATmega 8	32
2.5 Transistor	33
2.5.1 Transistor Sambungan Bipolar	33
BAB III PERENCANAAN DAN PEMBUATAN ALAT	
3.1 Rancangan Sistem Keseluruhan.....	34

3.2 Diagram Sistem Kamera Embedded.....	34
3.3 Perancangan Perangkat Keras	35
3.3.1 Perancangan Rangkaian Mikrokontroler ATmega 8	35
3.3.2 Perancangan Antarmuka Kamera ke Mikrokontroler.....	36
3.4 Perancangan Perangkat Lunak.....	37
3.4.1 Set Instruksi.....	37
3.4.2 Interupsi Serial.....	38
3.4.3 Program Utama	39
3.4.4 Format Keluaran Data Kamera Embedded	41
BAB IV PENGUJIAN SISTEM	
4.1 Pengujian Komunikasi Serial.....	42
4.2 Pengujian Rangkaian Transistor sebagai Saklar.....	43
4.3 Pengujian Pengambilan Gambar Dari Kamera Embedded	45
4.3 Pengujian Sistem.....	47
4.4 Hasil Pengujian Sistem	48
BAB V PENUTUP	
5.1 Kesimpulan	52
5.2 Saran.....	52
DAFTAR PUSTAKA	53
LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1 : Diagram Blok Sistem Pengolahan Citra Digital	4
Gambar 2.2 : Sistem Koordinat Piksel	7
Gambar 2.3 : Sistem Koordinat Spasial	8
Gambar 2.4 : Modul Sensor Kamera OV6620	8
Gambar 2.5 : Layout bagian atas (<i>top side</i>) OV6620	9
Gambar 2.6 : <i>timing pulse</i> OV6620	10
Gambar 2.7 : Pin-Out ATmega32	13
Gambar 2.8 : Register SREG	13
Gambar 2.9 : Register MCUCR	14
Gambar 2.10: Blok Diagram <i>Timer/Counter0</i>	15
Gambar 2.11: Register TCCR0	16
Gambar 2.12: Register TCNT0	18
Gambar 2.13: Register OCR0	18
Gambar 2.14: Register TIMSK untuk <i>Timer/Counter0</i>	18
Gambar 2.15: Register TIFR untuk <i>Timer/Counter0</i>	19
Gambar 2.16: Blok Diagram <i>Timer/Counter1</i>	20
Gambar 2.17: Register TCCR1A	21
Gambar 2.18: Register TCCR1B	23
Gambar 2.19: Register OCR1A dan OCR1B	24
Gambar 2.20: Register ICR1	24
Gambar 2.21: Register TIMSK untuk <i>Timer/Counter1</i>	25
Gambar 2.22: Register TIFR untuk <i>Timer/Counter1</i>	25
Gambar 2.23: Blok Diagram USART	26
Gambar 2.24: Register UDR	27
Gambar 2.25: Register UCSRA	28
Gambar 2.26: Register UCSRB	29
Gambar 2.27: Register UCSRC	30
Gambar 2.28: Register UBRR	31
Gambar 2.29: Pin-Out ATmega8	32
Gambar 2.30: Transistor Sambungan Bipolar	33
Gambar 3.1 : Diagram Keseluruhan Simulator Menembak menggunakan <i>box target</i>	34
Gambar 3.2 : Blok Diagram Sistem	34
Gambar 3.3 : Perancangan Rangkaian ATmega 8	35
Gambar 3.4 : Perancangan Antarmuka Kamera ke Mikrokontroler	37
Gambar 3.5 : Flowchart bagian interupsi serial	38
Gambar 3.6 : Flowchart Program Utama	39
Gambar 3.7 : Flowchart Subrutin Kirim Data Gambar	40
Gambar 3.8 : Flowchart Subrutin <i>Setting Track</i>	40
Gambar 3.9 : Flowchart Subrutin <i>Tracking Color</i>	41
Gambar 4.1 : Rangkaian Komunikasi Serial	42
Gambar 4.2 : Hasil Pengujian Komunikasi Serial	42
Gambar 4.3 : Pengujian Rangkaian Transistor 2N3904	43
Gambar 4.4 : Blok diagram pengujian	45
Gambar 4.5 : Foto Pengujian	45
Gambar 4.6 : Objek Menggunakan Kamera Digital	46

Gambar 4.7 : Hasil pengujian menampilkan gambar dari kamera embedded	46
Gambar 4.8 : Diagram blok pengujian sistem.....	47
Gambar 4.9 : Pengujian deteksi warna sinar laser	47
Gambar 4.10: Melakukan setting <i>Map Tracking Color</i>	48
Gambar 4.11: Program Antarmuka Simulator Menembak Memberikan Perintah Mendeteksi Sinar Laser ke Kamera Embedded	49
Gambar 4.12: Pengujian menembak sinar laser	49
Gambar 4.13: Hasil pengujian pada <i>software</i> hasil tembak	49
Gambar 4.14: Implementasi Kamera Embedded pada <i>Box Simulator Menembak</i>	51

DAFTAR TABEL

Tabel 2.1	: Deskripsi Pin Kamera OV6620	9
Tabel 2.2	: <i>Prescaler timer/counter0</i>	16
Tabel 2.3	: Mode Operasi <i>Wave Generation Mode</i> pada <i>Timer/Counter0</i>	17
Tabel 2.4	: Mode <i>Normal</i> dan <i>CTC</i> pada <i>Timer/Counter0</i>	17
Tabel 2.5	: Mode <i>Fast PWM</i> pada <i>Timer/Counter0</i>	17
Tabel 2.6	: Mode <i>Phase correct PWM</i> pada <i>Timer/Counter0</i>	18
Tabel 2.7	: Pin-pin <i>Timer/Counter1</i> pada mikrokontroler ATmega 32	20
Tabel 2.8	: Mode <i>Normal</i> dan <i>CTC</i> pada <i>Timer/Counter1</i>	21
Tabel 2.9	: Mode <i>Fast PWM</i> pada <i>Timer/Counter1</i>	21
Tabel 2.10	: Mode <i>Phase Correct</i> dan <i>Phase & Frekuensi Correct PWM</i> pada <i>Timer/Counter1</i>	21
Tabel 2.11	: Mode Operasi <i>Waveform Generation Mode</i> pada <i>Timer/Counter1</i>	22
Tabel 2.12	: <i>Prescaler timer/counter1</i>	24
Tabel 2.13	: <i>Baudrate</i>	27
Tabel 2.14	: <i>Parity</i>	30
Tabel 2.15	: Opsi Lebar Data USART	31
Tabel 2.16	: Polaritas <i>clock</i>	31
Tabel 4.1	: Hasil Pengukuran Rangkaian Transistor 2N3904 Sebagai Saklar	43
Tabel 4.2	: Pengukuran dan Perhitungan Rangkaian Transistor 2N3904 Sebagai Saklar	44
Tabel 4.3	: Kesalahan (Tingkat Error) Rangkaian Transistor 2N3904 Sebagai Saklar	45

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penerapan teknologi kamera pada saat ini mengalami perkembangan yang cukup pesat. Dan juga dari sistem pengolahan citra yang dapat memproses mulai dari sederhana hingga yang rumit atau mulai dari pra-proses, pemrosesan sampai dengan pengambilan keputusan dan dapat dikerjakan secara *real-time* ataupun tidak.

Terkadang sistem pengolahan citra yang ada sekarang masih berbasis komputer. Dikarenakan rumitnya algoritma program dan membutuhkan memori yang cukup untuk menyimpan hasil data dari kamera. Aplikasi ini tentu sangat mahal disamping itu sistem ini tidak dapat dibuat berdiri sendiri. Padahal dengan aplikasi yang berdiri sendiri sangat bermanfaat terutama aplikasi yang bersifat *portable*.

Kamera Embedded (*Embedded Camera*) merupakan salah satu sistem pengolahan citra yang dapat berdiri sendiri, karena sudah ditenamkan didalamnya prosessor yang dapat memproses citra dan dapat mengambil keputusan atau perintah yang akan diproses selanjutnya. Perkembangan prosesor / *controller* saat ini membuka peluang bagi pengembangan lebih lanjut dalam pengolahan citra yang berdiri sendiri.

Pada penelitian ini penulis membangun sebuah sistem kamera embedded yang dapat diaplikasikan pada simulator menembak. Mikrokontroler mendeteksi warna titik sinar laser dari output citra kamera. Jika terdeteksi mikrokontroler mengirim koordinat posisi titik sinar laser ke komputer. Untuk mendapatkan nilai skor hasil tembak.

1.2. Rumusan Masalah

Dalam rancang bangun kamera cmbdedd pada hasil bidikan simulator tembak, maka permasalahannya adalah sebagai berikut :

1. Bagaimana cara merancang dan membuat perangkat keras sistem kamera embedded.
2. Bagaimana membuat perangkat lunak untuk proses citra dari kamera. Sehingga bisa di aplikasikan pada simulator tembak.

1.3. Tujuan

Tujuan dari pembuatan alat ini adalah membuat kamera embedded dan perangkat lunak yang kompatibel. Sehingga bisa dipakai pada aplikasi *portable* yang memerlukan proses pencitraan. Dan pada aplikasi ini dapat memproses hasil dari simulator tembak.

1.4. Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan yang diharapkan, maka pembatasan dibatasi oleh hal-hal sebagai berikut:

1. Pembahasan menitikberatkan pada sistem kamera embedded dengan sensor kamera OV6620 dan Mikrokontroler ATmega 32.
2. Tidak membahas secara mendalam tentang konstruksi senjata, senjata yang digunakan hanya miniatur / mainan. Dan menembak menggunakan sinar laser.
3. Rancang bangun kamera embedded ini di implementasikan pada kotak (*box*) simulator tembak yang sudah memiliki spesifikasi target lingkaran dan dimensi kotak.

1.5 Metodologi Penulisan

Metode yang digunakan dalam penulisan tugas akhir ini adalah

1. Studi Pustaka
Memperoleh data dengan cara membaca dan mempelajari buku literature yang berhubungan dengan penyusunan skripsi ini.
2. Studi Lapangan
Memperoleh data dengan cara praktek secara langsung untuk menunjang pembuatan alat.
3. Pengolahan Data
Mengolah data dengan jalan membuat analisa dan menarik kesimpulan dari hasil pengujian yang ada.

1.6 Sistematika Penulisan

Sistematika pembahasan dari skripsi ini terdiri dari pokok pembahasan yang saling berkaitan antara satu dengan lainnya, yaitu :

BAB I : PENDAHULUAN

Pada bab ini dibahas tentang latar belakang permasalahan, rumusan masalah, batasan masalah, sistematika pembahasan dari Hardware yang direncanakan.

BAB II : LANDASAN TEORI

Pada bab ini dibahas teori tentang proses pencitraan digital, pengenalan sensor kamera CMOS OV6620, cara kerjanya serta tentang mikrokontroler beserta fitur didalamnya.

BAB III : PERENCANAAN DAN PEMBUATAN ALAT

Pada bab ini dibahas tentang perencanaan dan pembuatan alat kamera embedded.

BAB IV : PENGUJIAN SISTEM

Pada bab ini akan dibahas tentang proses serta hasil dari pengujian sistem, yaitu koneksi kamera ke mikrokontroler, koneksi mikrokontroler ke komputer secara serial. Dan pengujian secara keseluruhan.

BAB V : PENUTUP

Pada bab ini akan disampaikan kesimpulan dan saran dari perencanaan dan pembuatan sistem ini.

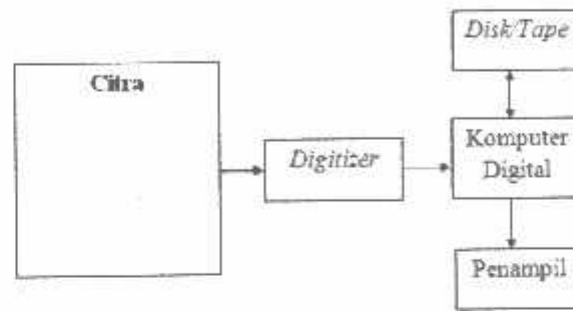
BAB II

LANDASAN TEORI

Pada bab ini berisi teori-teori yang akan digunakan dalam penyusunan skripsi ini.

2.1 Pengolahan Citra Digital^[3]

Berikut ini adalah blok sistem Pengolahan Citra Digital :



Gambar 2.1.
Diagram Blok Sistem Pengolahan Citra Digital^[3]

Digitizer mengubah citra masukan menjadi sinyal listrik dan kemudian mencuplik sinyal listrik tersebut menggunakan *A/Dconverter (Analog to Digital Converter)*. *Digitizer* ini dapat berupa *scanner* atau kamera video yang mengubah citra kontinu kedalam suatu representasi numerik, sehingga citra ini dapat diproses oleh komputer digital.

Proses pengolahan data dapat dilakukan oleh komputer, baik berupa mikrokomputer sederhana (*microprocessor based computer*) atau komputer besar (*mainframe computer*), tergantung jumlah data dan jenis pengolahan.

Proses penampilan data merupakan salah satu segi yang penting karena bagaimanapun juga citra digital hasil olahan harus dapat dinilai oleh mata manusia melalui suatu penampil (*display*). Penampil yang digunakan biasanya berupa suatu *graphic monitor* atau suatu *graphic printer/plotter*.

2.1.1 Model Citra Digital^[3]

Sesungguhnya citra merupakan suatu fungsi intensitas dalam bidang dua dimensi. Karena intensitas yang dimaksud berasal dari sumber cahaya, dan cahaya adalah suatu bentuk energi, maka keadaan dimana fungsi intensitas terletak diantara : $0 < f(x,y) < \infty$.

Pada dasarnya, citra yang dilihat terdiri atas berkas-berkas cahaya yang dipantulkan oleh benda-benda disekitarnya. Jadi secara alamiah, fungsi intensitas cahaya merupakan fungsi sumber cahaya yang menerangi obyek, serta jumlah cahaya yang dipantulkan oleh obyek, atau di tulis :

$$f(x,y) = i(x,y) \cdot r(x,y) \quad (2.1)$$

yaitu : $0 < i(x,y) < \infty$ (iluminasi sumber cahaya)

$0 < r(x,y) < 1$ (koefisien pantul cahaya)

Beberapa contoh harga iluminasi $i(x,y)$:

- Pada hari cerah (tak berawan), matahari menghasilkan iluminasi sebesar ± 9000 foot-candles.
- Pada hari mendung (berawan), matahari hanya menghasilkan iluminasi sebesar ± 1000 foot-candles.
- Pada malam bulan purnama, sinar bulan menghasilkan iluminasi sebesar $\pm 0,01$ foot-candles.
- Pada ruang kantor normal, iluminasi sebesar ± 100 foot-candles.

Beberapa harga koefisien pantul $r(x,y)$:

- Benda hitam : $\pm 0,01$
- Dinding putih : $\pm 0,80$
- *Stainless steel* : $\pm 0,65$
- Logam berlapis perak : $\pm 0,90$
- Salju : $\pm 0,93$

Fungsi intensitas f pada suatu titik (x,y) disebut derajat keabuan atau *grey level* (I), dengan I terletak diantara :

$$L_{min} \leq I \leq L_{max} \quad (2.2)$$

Dengan demikian :

$$L_{min} = i_{min} \cdot i'_{min} \quad (2.3)$$

$$L_{max} = i_{max} \cdot i'_{max} \quad (\text{dalam foot-candles}) \quad (2.4)$$

Selang (L_{min}, L_{max}) sering disebut sebagai skala keabuan. Pada representasi suatu citra hitam putih secara numerik, biasanya selang digeser menjadi $(0, L)$, dengan 0 menyatakan hitam dan L menyatakan putih. Semua bilangan yang terletak di antara 0 dan L merupakan derajat keabuan.

2.1.2 Pencuplikan dan Kuantisasi^[3]

Suatu citra dapat dipresentasikan secara numerik, maka citra harus didigitalisasi, baik terhadap ruang (koordinat (x, y)) maupun terhadap skala keabuannya $f(x, y)$. Proses digitalisasi koordinat (x, y) dikenal sebagai "pencuplikan citra" (*image sampling*), sedangkan proses digitalisasi skala keabuan $f(x, y)$ disebut sebagai "kuantisasi derajat keabuan" (*gray level quantization*).

Sebuah citra kontinu $f(x, y)$ akan didekati oleh cuplikan-cuplikan yang seragam jaraknya dalam bentuk matriks $N \times N$. Nilai elemen-elemen matriks menyatakan derajat keabuan citra, sedangkan posisi elemen tersebut (dalam baris dan kolom) menyatakan koordinat titik-titik (x, y) dari citra.

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1, N-1) \end{bmatrix} \quad (2.5)$$

Bentuk matriks ini dikenal sebagai suatu citra digital.

Dengan alasan untuk memudahkan implementasi, dalam praktek sebagian besar diambil jumlah pencuplikan pada baris (N) dan kolom (N) sebagai bilangan pangkat dua dengan jarak cuplikan yang seragam. Jadi diambil :

$$N = 2^n ; n = \text{bilangan bulat positif}$$

Dengan alasan serupa, maka lazimnya skala keabuan $(0, L)$ dibagi kedalam G selang dengan panjang selang yang sama:

$$G = 2^m ; m = \text{bilangan bulat positif}$$

Bila hal ini diterapkan, maka penyimpanan sebuah citra digital membutuhkan sejumlah b bit data :

$$B = N \times N \times m$$

Sebagai contoh, untuk menyimpan citra berukuran $512 \times 512 \text{ pixel}$ dengan 256 derajat keabuan diperlukan jumlah *bit* sebanyak: $512 \times 512 \times 256 = 2.048.000 \text{ bit}$.

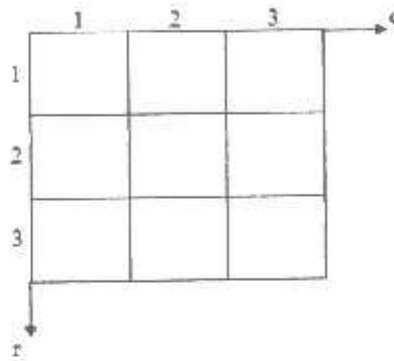
Makin tinggi nilai N dan m , maka citra kontinu $f(x, y)$ akan makin didekati oleh citra digital yang dihasilkan.

2.1.3 Sistem Koordinat Pada Citra Digital^[3]

Dalam sistem koordinat pada citra digital, bisa dikategorikan sebagai berikut :

a. Sistem koordinat piksel

Secara umum, metode yang umum untuk mengekspresikan lokasi dari sebuah citra adalah menggunakan koordinat piksel. Dalam sistem koordinat, suatu citra diperlukan sebagai garis-garis seperti dibawah ini :



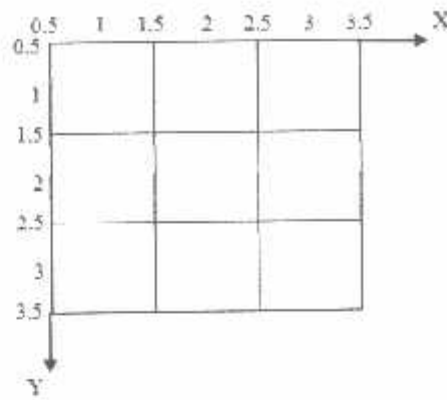
Gambar 2.2.
Sistem Koordinat Piksel^[3]

Untuk sistem koordinat piksel, komponen pertama yaitu r (baris) bertambah kebawah, sementara komponen kedua c (kolom) bertambah kekanan. Sistem koordinat piksel bertipe integer dan bernilai antara 1 sampai panjang baris atau kolom.

b. Sistem koordinat spasial

Pada sistem koordinat piksel, maka sebuah piksel diperlakukan sebagai unit yang diskrit, secara ini diidentifikasi dengan sepasang koordinat tunggal seperti $(5,2)$. Dari sudut pandang ini sebuah lokasi seperti $(5.3,2.2)$ tidak mempunyai arti.

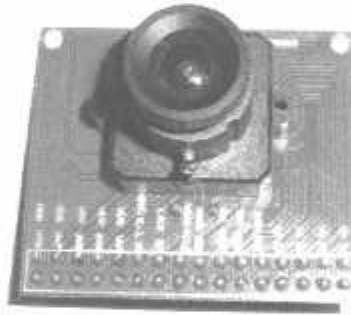
Tetapi sekarang berguna untuk memikirkan sebuah piksel seperti kotak mempunyai area. Dari sudut pandang ini, sebuah lokasi seperti $(5.2,2.2)$ mempunyai arti dan berbeda dari $(5,2)$. Pada sistem koordinat spasial, lokasi dalam sebuah citra diposisikan pada suatu daerah yang dijabarkan pada x dan y



Gambar 2.3.
Sistem Koordinat Spasial^[3]

2.2 Modul Sensor Kamera OV6620^[7]

Merupakan sebuah modul sensor kamera sebagai masukan dari pencitraan. Dengan keluaran langsung berupa data digital, dikemas dalam bentuk *chip*, dan memiliki pengaturan lensa fokus dapat diatur secara manual sehingga bisa diatur jarak penangkapan citra.



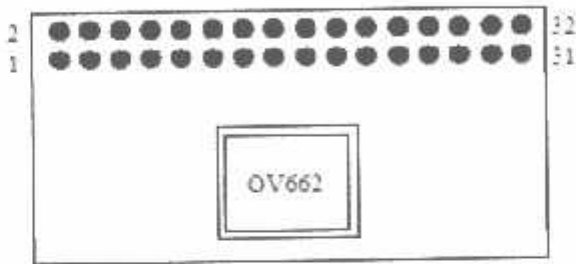
Gambar 2.4.
Modul Sensor Kamera OV6620^[8]

Spesifikasi dari Kamera CMOS OV6620 sebagai berikut^[7] :

- 101376 piksel, diameter lensa 1/4".
- Resolusi gambar (*Array Size*) 2 Mode : 1 (356 x 292), 2 (176 x 144).
- 8/16 bit data gambar, dengan format data keluaran YCrCb 4:2:2 / GRB 4:2:2 / RGB.
- Pembacaan citra secara (*Scanning Progressive*) atau baris per baris.
- *Electronic Exposure* 500:1.
- *Gamma Correction* 0.45/0.55/1.0

- *S/N Ratio* 48 dB.
- Interface kontrol set register kamera menggunakan I2C. Mempunyai 80 set register. Termasuk didalamnya register set *brightness, contrast, gamma saturation, sharpness* dengan protokol SCCB (*Single Camera Control Bus*).
- *Frequency clock input* 17.7344 MHz
- Tegangan operasi +5Vdc, dengan daya sedang aktif 80mW, 30μW keadaan stand-by.

Deskripsi tiap-tiap pin pada modul kamera OV6620



Gambar 2.5
Layout bagian atas (*top side*) OV6620^[8]

Tabel 2.1
Deskripsi Pin Kamera OV6620

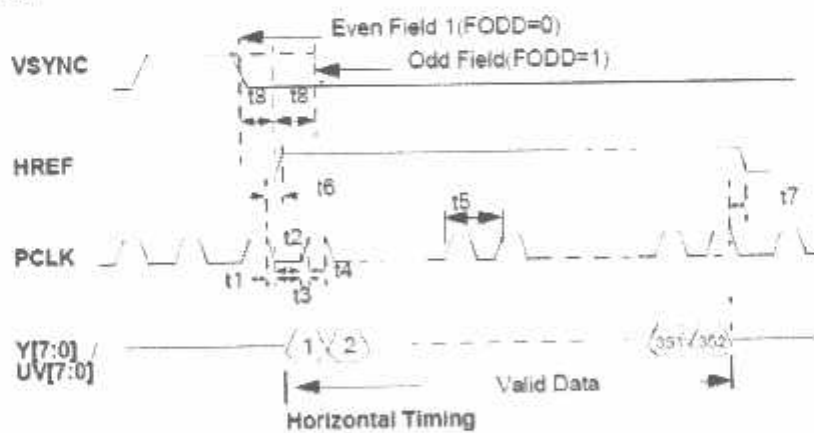
<i>Pin</i>	<i>Name</i>	<i>Description</i>
1~8	Y0~Y7	Digital Output Y Bus
9	PWDN	Power down mode
10	RST	Reset
11	SDA	I2C Serial data
12	FODD	Odd Field flag
13	SCL	I2C Serial clock input
14	HREF	Horizontal window reference output
15	AGND	Analog Ground
16	VSYNC	Vertical Sync output
17	AGND	Analog Ground
18	PCLK	Pixel clock output

19	EXCLK	External Clock input (Remove Crystal)
20	VCC	Power Supply 5VDC
21	AGND	Analog Ground
22	VCC	Power Supply 5VDC
23~30	UV0~UV7	Digital output UV bus.
31	GND	Common ground
32	VTO	Video Analog Output (75Ω monochrome)

Dalam proses pengambilan citra, ada 3 pin output sinyal kontrol dari kamera yaitu HREF, VSYNC, PCLK. Pin-pin ini berfungsi sebagai berikut :

- VSYNC (*Vertical Sync Output*) mendeteksi sebagai awalan citra jika berlogika satu (1).
- HREF (*Horizontal Ref Output*) berlogika satu (1) sebagai tanda awalan baris kemudian berlogika nol (0) sebagai akhiran baris.
- PCLK (*Pixel Clock*) berfungsi sebagai clock setiap output data piksel, jika berlogika satu (1) maka untuk pin keluaran Y/UV bus menjadi status output dan jika berlogika nol (0) berstatus default.

Jika HREF =1, dan PCLK =1 maka data dari kamera adalah valid ini bisa dilihat pada *timing pulse* berikut ini :



Gambar 2.6
timing pulse OV6620^[7]

2.2.1 Akurasi nilai data digital OV6620^[7]

Ketentuan nilai akurasi dari proses pencitraan digital pada kamera OV6620 dapat dijelaskan sebagai berikut :

1. Level puncak warna hitam adalah 16.
2. Level puncak warna putih adalah 240.
3. Nilai minimum 16 dan maksimum 240 pada setiap komponen warna primer.

2.2.2 Format Data Keluaran OV6620^[7]

Format citra digital dari sensor kamera sebagai berikut :

Baris 1: $B_{(1,1)}, H_{(1,2)}, B_{(1,3)}, H_{(1,4)} \dots B_{(1,355)}, H_{(1,356)}$

Baris 2: $H_{(2,1)}, M_{(2,2)}, H_{(2,3)}, M_{(2,4)} \dots H_{(2,355)}, M_{(2,356)}$

Baris 3: $B_{(3,1)}, H_{(3,2)}, B_{(3,3)}, H_{(3,4)} \dots B_{(3,355)}, H_{(3,356)}$

Baris 4: $H_{(4,1)}, M_{(4,2)}, H_{(4,3)}, M_{(4,4)} \dots H_{(4,355)}, M_{(4,356)}$

.

.

.

Baris 292: $H_{(292,1)}, M_{(292,2)}, H_{(292,3)}, M_{(292,4)} \dots H_{(292,355)}, M_{(292,356)}$

Keterangan : Warna_(Baris,Kolom)

(B : Biru, H : Hijau, M : Merah)

Pada sensor kamera OV6620 terdapat 2 keluaran (*Output*) yaitu Channel Y (pin1~pin8) dan Channel UV (pin23~30). Dengan format 16 bit, 8 bit atau 4 bit.

- Format 16 Bit (Total HREF 292)

- Mode *Default* (Setting awal)

HREF ke-1 UV : $B_{(1,1)}, H_{(1,2)}, B_{(1,3)}, H_{(1,4)} \dots$

HREF ke-2 Y : $H_{(2,1)}, M_{(2,2)}, H_{(2,3)}, M_{(2,4)} \dots$

UV : $B_{(1,1)}, H_{(1,2)}, B_{(1,3)}, H_{(1,4)} \dots$

HREF ke-3 UV : $B_{(3,1)}, H_{(3,2)}, B_{(3,3)}, H_{(3,4)} \dots$

Y : $H_{(2,1)}, M_{(2,2)}, H_{(2,3)}, M_{(2,4)} \dots$

HREF ke-4 Y : $H_{(4,1)}, M_{(4,2)}, H_{(4,3)}, M_{(4,4)} \dots$

UV : $B_{(3,1)}, H_{(3,2)}, B_{(3,3)}, H_{(3,4)} \dots$

dan seterusnya.

- Mode *Y/G*

HREF ke-1 UV dan Y (*Unstable*) abaikan

HREF ke-2 Y : $H_{(2,1)}, H_{(1,2)}, H_{(2,3)}, H_{(2,4)} \dots$

UV : $B_{(1,1)}, M_{(2,2)}, B_{(1,3)}, M_{(2,4)} \dots$

HREF ke-3 Y : $H_{(2,1)}, H_{(3,2)}, H_{(2,3)}, H_{(3,4)} \dots$
 UV : $B_{(3,1)}, M_{(2,2)}, B_{(3,3)}, M_{(2,4)} \dots$
 HREF ke-4 Y : $H_{(4,1)}, H_{(3,2)}, H_{(4,3)}, H_{(3,4)} \dots$
 UV : $B_{(3,1)}, M_{(4,2)}, B_{(3,3)}, M_{(4,4)} \dots$

dan seterusnya.

- Mode *One Line* (Satu Channel)

HREF ke-1 Y : $B_{(1,1)}, H_{(1,2)}, B_{(1,3)}, H_{(1,4)} \dots$
 HREF ke-2 Y : $H_{(2,1)}, M_{(2,2)}, H_{(2,3)}, M_{(2,4)} \dots$
 HREF ke-3 Y : $B_{(3,1)}, H_{(3,2)}, B_{(3,3)}, H_{(3,4)} \dots$
 HREF ke-4 Y : $H_{(4,1)}, M_{(4,2)}, H_{(4,3)}, M_{(4,4)} \dots$

dan seterusnya. Kondisi output channel UV bisa diabaikan.

- Format 8 Bit (Total HREF 292)

HREF ke-1 : Abaikan
 HREF ke-2 Y : $B_{(1,1)}, H_{(2,1)}, M_{(2,2)}, H_{(1,2)} \dots$
 HREF ke-3 Y : $B_{(3,1)}, H_{(2,1)}, M_{(2,2)}, H_{(3,2)} \dots$

dan seterusnya. Kondisi output channel UV bisa diabaikan.

- Format 4 Bit (Total HREF 292)

Format sama dengan 8 bit, namun di bagi menjadi 2 *nibble* tinggi dan rendah.

Dikirim pada port Y (7...4). Kecepatan clock piksel dua kali lebih lambat dari format 8 bit.

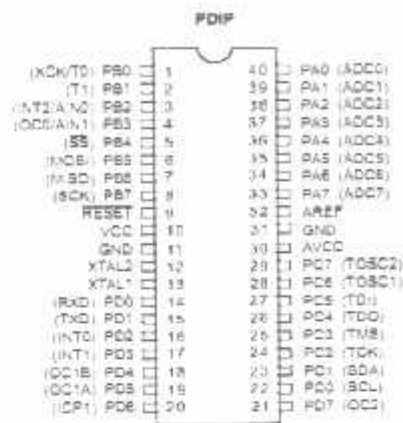
2.3 Mikrokontroler ATmega 32^[6]

Mikrokontroler ATmega32 memiliki fitur seperti (ADC internal, EEPROM internal, Timer/Counter, Watchdog Timer, PWM, Port I/O, komunikasi serial, Komparator, I2C, dll).

Berikut ini merupakan beberapa spesifikasi ATmega32 :

1. Memiliki kapasitas flash memori 32Kbyte, EEPROM 512 Byte, dan SRAM 1Kbyte.
2. Saluran Port I/O sebanyak 32 buah, yaitu Port A, Port B, Port C, dan Port D.
3. CPU yang terdiri atas 32 buah register.
4. Interupsi internal dan eksternal.
5. Port USART sebagai komunikasi serial.
6. Konsumsi daya rendah (DC 5V).
7. Fitur peripheral, yang terdiri dari :

- a. Tiga buah Timer/Counter dengan perbandingan
- b. 4 channel PWM.
- c. 8 channel, 10-bit ADC



Gambar 2.7
Pin-Out ATMega32^[6]

2.3.1 Interupsi Mikrokontroler ATMega 32^[2]

Dalam eksekusi aliran program mikrokontroler dengan bahasa c dimulai dari fungsi *main* terus ke bawah, adapun lompat/panggil fungsi lain diluar fungsi *main* bergantung pada instuksi/pernyataan dalam fungsi *main*. Setiap *pheripheral* mikrokontroler dilengkapi *flag/bit* status kejadian tertentu. *Flag* tersebut dapat digunakan untuk membangkitkan sistem interupsi atau digunakan untuk sistem *pooling*. Sistem interupsi adalah menghentikan aliran program akibat terjadinya *triger* tertentu dan memaksa eksekusi rutin/fungsi layanan interupsi, setelah selesai maka aliran program akan kembali ke pernyataan program sebelum terjadinya interupsi.

Register SREG (*status register*) adalah (*register*) khusus yang memantau status CPU sekaligus tempat *bit-I* berada. Untuk menjaga isi SREG dalam rangka rutin penanganan interupsi maka harus disimpan secara *manual* melalui *software* yang dibuat oleh pengguna.

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial/Value	0	0	0	0	0	0	0	0	

Gambar 2.8
Register SREG^[2]

- Bit-7 - I : *Global Interrupt Enable*
Bit ini (jika set) berstatus mengaktifkan semua interupsi setelah interupsi *peripheral* yang bersangkutan diaktifkan.
- Bit-6 - T : *Bit Copy Storage*
Instruksi dalam asm, *copy bit* BLD (*bit Load*) dan BST (*Bit Store*) menggunakan *bit-T* sebagai sumber dan tujuan operasi *copy bit*.
- Bit-5 - H : *Half Carry Flag*
Bit ini akan set jika dalam operasi aritmatika terjadi *half carry*.
- Bit-4 - S : *Sign Bit*, S = N (+) V
Bit ini hasil *exclusive or* antara *flag negative* N dan *flag overflow* V.
- Bit-3 - V : *Two's Complement Overflow Flag*
Flag ini akan set jika terjadi *overflow* dalam operasi aritmatika.
- Bit-2 - N : *Negative Flag*
Flag ini akan set jika dalam operasi aritmatika atau logika menghasilkan negatif.
- Bit-1 - Z : *Zero Flag*
Flag ini akan set jika dalam operasi aritmatika atau logika menghasilkan nol.
- Bit-0 - C : *Carry Flag*
Flag ini akan set jika dalam operasi aritmatika atau logika menghasilkan sisa (*Carry*).

Interupsi eksternal adalah sebuah *peripheral* dalam *chip* yang bertugas mendeteksi *trigger* dari luar yang akan membangkitkan interupsi yang bersangkutan. Interupsi eksternal *chip* disediakan tiga buah pada *pin* INT0, INT1 atau INT2 dimana *trigger* yang digunakan dapat kita pilih definisikan sendiri.

Bit	7	6	5	4	3	2	1	0	
	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.9
Register MCUCR^[2]

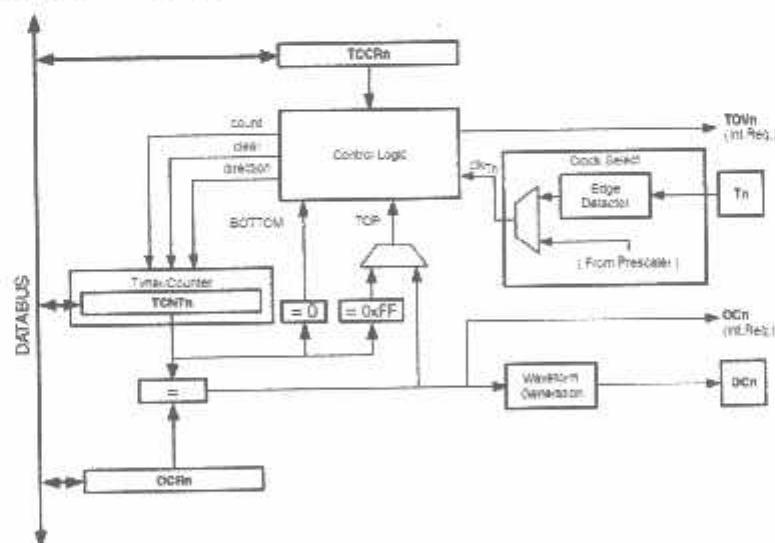
Dalam *register* MCUCR hanya 4-bit yang berhubungan dengan interupsi eksternal, dimana keempat *bit* tersebut digunakan untuk mendefinisikan *input trigger* yang digunakan sebagai sinyal interupsi eksternal.

- Bit-3,2 - ISC11,ISC10 : *Interrupt Sense Control 1 (Bit 1 & 0)* Kontrol trigger interupsi eksternal 1 (INT1)
 - ISC11 = 0, ISC10 = 0 : Level Rendah (*Low Level*)
 - ISC11 = 0, ISC10 = 1 : *Toggle*
 - ISC11 = 1, ISC10 = 0 : Perpindahan 1 ke 0 (*Falling edge*)
 - ISC11 = 0, ISC10 = 0 : Perpindahan 0 ke 1 (*Rising Edge*)
- Bit-1,0 - ISC01,ISC00 : *Interrupt Sense Control 0 (Bit 1 & 0)* Kontrol trigger interupsi eksternal 0 (INT0)
 - ISC01 = 0, ISC00 = 0 : Level Rendah (*Low Level*)
 - ISC01 = 0, ISC00 = 1 : *Toggle*
 - ISC01 = 1, ISC00 = 0 : Perpindahan 1 ke 0 (*Falling edge*)
 - ISC01 = 0, ISC00 = 0 : Perpindahan 0 ke 1 (*Rising Edge*)

2.3.2 Timer / Counter 0 Mikrokontroler ATmega 32^[2]

Timer/counter 0 adalah sebuah timer/counter yang dapat mencacah sumber pulsa/clock baik dari dalam chip (timer) ataupun dari luar chip (counter) dengan kapasitas 8-bit atau 256 cacahan.

Dapat digunakan untuk : Timer/counter biasa, *Clear Timer on Compare Match*, generator frekuensi, counter pulsa eksternal. Mempunyai hingga 10-bit (1024) *Clock Prescaler* (pemilih clock yang masuk ke timer/counter).



Gambar 2.10
Blok Diagram timer/counter0^[6]

Pulsa yang masuk ke dalam TCNT0 diseleksi oleh aturan *clock select register* TCCR0. Pada *clock select* ada dua sumber *clock* dari luar dan dari dalam *chip*. Untuk *clock* dari dalam melewati *prescaler* yang bertindak sebagai *timer*, dan untuk *clock* dari luar melewati *edge select detector* untuk menentukan kondisi *clock* yang akan masuk *falling* atau *rising* dalam hal ini bertindak sebagai counter. Jika kondisi TCNT0 sudah maksimal kemudian kembali ke nol maka akan ada sinyal *overflow* berupa *flag/bit*-TOV0 yang mengindikasikan permintaan interupsi.

Untuk interupsi *compare match* (TCNT0 = OCR0), nilai TCNT0 akan selalu dibandingkan oleh cpu dengan isi register OCR0. Jika terjadi *compare match* maka akan menyebabkan sinyal permintaan interupsi ditandai dengan set-nya *flag* OCF0, sinyal interupsi dapat kita salurkan ke luar melalui pin PB3(OC0) dengan kelakuan tertentu yang didefinisikan oleh bit COM01 dan COM00.

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.11
Register TCCR0^[2]

Bit CS00 s/d 02 bertugas untuk memilih (*prescaler*) atau mendefinisikan pulsa / *clock* yang akan masuk ke dalam *timer/counter0*.

Tabel 2.2
Prescaler timer/counter0

CS02	CS01	CS00	Keterangan
0	0	0	Stop/Berhenti
0	0	1	Clk (sama dengan <i>clock</i> CPU)
0	1	0	Clk/8 (1 clk <i>timer/counter0</i> = 8 clk CPU)
0	1	1	Clk/64 (1 clk <i>timer/counter0</i> = 64 clk CPU)
1	0	0	Clk/256 (1 clk <i>timer/counter0</i> = 256 clk CPU)
1	0	1	Clk/1024 (1 clk <i>timer/counter0</i> = 1024 clk CPU)
1	1	0	Clk eksternal dari pin T0 pada kondisi <i>falling edge</i>
1	1	1	Clk eksternal dari pin T0 pada kondisi <i>rising edge</i>

Clk/8 (1 clk *timer/counter0* = 8 clk CPU) artinya tiap 8 *clock* CPU yang masuk kedalam *timer/counter0* dihitung satu oleh register pencacah TCNT0. *Falling edge* adalah perubahan pulsa / *clock* dari 1 ke 0. *Rising edge* adalah perubahan pulsa / *clock* dari 0 ke 1.

Bit 7 (FOC0) *Force Output Compare*, bit ini hanya dapat digunakan untuk mode pembandingan. Jika bit FOC0 di set maka akan memaksa terjadinya *compare match* ($TCNT0 == OCR0$).

Bit 6 dan 3 (WGM00 dan WGM01) *Waveform Generation Mode*, kedua bit ini di gunakan untuk memilih mode yang digunakan seperti pada tabel berikut.

Tabel 2.3
Mode Operasi *Wave Generation Mode* pada *Timer/Counter0*

WGM01	WGM00	Mode Operasi	TOP	Update OCR0	Flag TOV0 on
0	0	Normal	0xFF	Imidiet	MAX
0	1	Phase correct PWM	0xFF	TOP	Bottom
1	0	CTC	OCR0	Imidiet	MAX
1	1	Fast PWM	0xFF	Bottom	MAX

Bit 5 dan 4 (COM01 dan COM00) *Compare Match Output Mode*, kedua bit ini berfungsi mendefinisikan pin OC0 sebagai output timer 0 (atau sebagai saluran *output PWM*).

Tabel 2.4
Mode Normal dan CTC pada *Timer/Counter0*

COM01	COM00	Output pin OC0
0	0	Tidak dihubungkan
0	1	Toggle saat <i>Compare Match</i>
1	0	Clear saat <i>Compare Match</i>
1	1	Set saat <i>Compare Match</i>

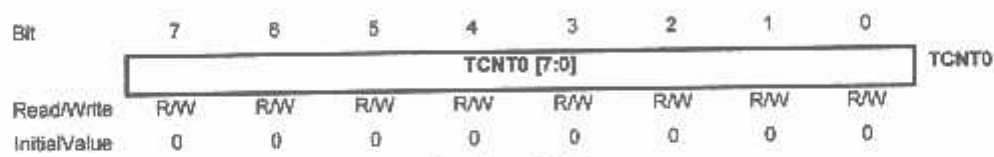
Tabel 2.5
Mode Fast PWM pada *Timer/Counter0*

COM01	COM00	Output pin OC0
0	0	Tidak dihubungkan
0	1	-
1	0	Clear saat <i>Compare Match</i> , set saat bottom (<i>noninverting</i>)
1	1	Set saat <i>Compare Match</i> , clear saat bottom (<i>inverting</i>)

Tabel 2.6
Mode *Phase correct PWM* pada *Timer/Counter0*

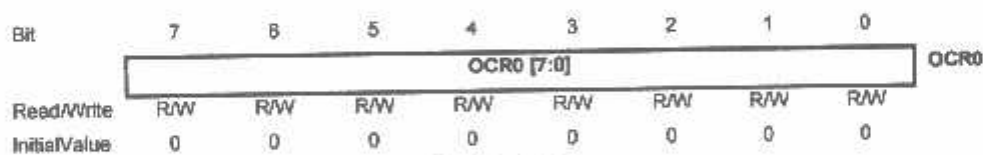
COM01	COM00	Output pin OC0
0	0	Tidak dihubungkan
0	1	-
1	0	Clear saat <i>Compare Match</i> ketika <i>up-counting</i> , set saat <i>compare match</i> ketika <i>down-counting</i> (<i>noninverting</i>)
1	1	Set saat <i>Compare Match</i> ketika <i>up-counting</i> , clear saat <i>compare match</i> ketika <i>down-counting</i> (<i>inverting</i>)

Register *TCNT0* (*Timer/Counter 0*), register ini bertugas menghitung pulsa yang masuk kedalam *timer/counter*. Kapasitas register ini 8-bit atay 255 hitungan, setelah mencapai hitungan maksimal maka akan kembali ke nol (*overflow*/limpahan).

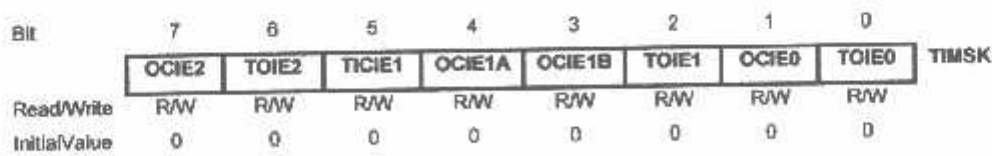


Gambar 2.12
Register *TCNT0*^[2]

Register *OCR0* (*Output Compare Register 0*), register ini bertugas sebagai register pembanding yang bisa ditentukan besarnya sesuai dengan kebutuhan. Dalam prakteknya pada saat *TCNT0* mencacah maka otomatis oleh CPU akan dibandingkan dengan isi *OCR0* secara kontinyu dan jika isi *TCNT0* sama dengan isi *OCR0* maka akan terjadi *compare match* yang dapat dimanfaat untuk mode *CTC* dan *PWM*.



Gambar 2.13
Register *OCR0*^[2]



Gambar 2.14
Register *TIMSK* untuk *Timer/Counter0*^[2]

Bit 0 TOIE0 (*Timer/Counter 0 Overflow Interrupt Enable*), sebagai bit pengaktif interupsi *timer/counter 0* (TOIE0 =1 *enable*, TOIE0 =0 *disable*).

Bit 1 OCIE0 (*Timer/Counter 0 Output Compare Match Interrupt Enable*), sebagai bit pengaktif interupsi *compare match timer/counter 0* (OCIE0 =1 *enable*, OCIE0 =0 *disable*).

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.15
Register TIFR untuk *Timer/Counter0*^[2]

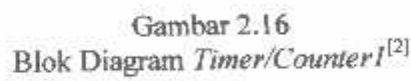
Bit 1 OCF0 (*Output Compare Flag 0*), flag OCF0 akan set sebagai indikator terjadinya *compare match*, dan akan *clear* sendiri bersamaan eksekusi vektor interupsi *timer compare match*.

Bit 0 TOV0 (*Timer/Counter Overflow Flag*), akan set secara otomatis ketika terjadi limpahan / *overflow* pada register TCNT0 dan akan *clear* bersamaan dengan eksekusi vektor interupsi.

2.3.3 *Timer / Counter 1* Mikrokontroler ATmega 32^[2]

Timer/Counter 1 adalah sebuah *timer/counter* yang mempunyai kapasitas cacahan 16-bit (65535) baik pulsa/clock internal maupun eksternal yang dilengkapi *prescaler* sumber pulsa/clock hingga 10-bit (1024).

Dapat digunakan untuk : *Timer/Counter* biasa, *Clear Timer on Compare Match* (*auto Reload*), *Counter* pulsa eksternal, *Capture Unit* (unit pelengkap isi TCNT1 akibat *trigger pin* ICP1/PB0), generator frekuensi biasa dan generator frekuensi PWM.



Tabel 2.7
Pin-pin Timer/Counter1 pada mikrokontroler ATmega 32

20

Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	W	W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.17
Register TCCR1A^[2]

Bit 7 dan 6 (COM1A1 dan COM1A0) *Compare Output Mode for channel A*, juga bit 5 dan 4 (COM1B1 dan COM1B0) *Compare Output Mode for channel B*. Bit-bit ini bertugas mengendalikan sifat / kelakuan pin OC1A dan OC1B yang berhubungan dengan mode operasi yang digunakan.

Tabel 2.8
Mode Normal dan CTC pada *Timer/Counter 1*

COM1A1 / COM1B1	COM1A0 / COM1B0	Keterangan
0	0	Tidak dihubungkan pin OC1A / OC1B
0	1	<i>Toggle pin OC1A / OC1B pada saat Compare Match</i>
1	0	<i>Clear pin OC1A / OC1B pada saat Compare Match</i>
1	1	<i>Set pin OC1A / OC1B pada saat Compare Match</i>

Tabel 2.9
Mode Fast PWM pada *Timer/Counter 1*

COM1A1 / COM1B1	COM1A0 / COM1B0	Keterangan
0	0	Tidak dihubungkan pin OC1A / OC1B
0	1	WGM13:0=15; <i>Toggle pin OC1A pada saat Compare Match</i> , pin OC1B tidak dihubungkan untuk setting WGM13:0 yang lain pin OC1A/OC1B tidak dihubungkan.
1	0	<i>Clear pin OC1A / OC1B pada saat Compare Match</i> , <i>set pin OC1A/OC1B pada saat BOTTOM (non-inverting)</i>
1	1	<i>Set pin OC1A / OC1B pada saat Compare Match</i> , <i>clear pin OC1A/OC1B pada saat BOTTOM (inverting)</i>

Tabel 2.10
Mode *Phase Correct* dan *Phase & Frekuensi Correct PWM* pada *Timer/Counter 1*

COM1A1 / COM1B1	COM1A0 / COM1B0	Keterangan
0	0	Tidak dihubungkan pin OC1A / OC1B
0	1	WGM13:0=9 atau 14; <i>Toggle pin OC1A pada saat Compare Match</i> , pin OC1B tidak dihubungkan untuk setting WGM13:0 yang lain pin OC1A/OC1B tidak dihubungkan.

1	0	Clear pin OC1A / OC1B pada saat Compare Match ketika counting-up, set pin OC1A/OC1B pada saat Compare Match ketika counting-down (non-inverting)
1	1	Set pin OC1A / OC1B pada saat Compare Match ketika counting-up, clear pin OC1A/OC1B pada saat Compare Match ketika counting-down (inverting)

Bit 3 (FOC1A) Force Output Compare for Channel A dan Bit 2 (FOC1B) Force Output Compare for Channel B, hanya dapat digunakan ketika menggunakan mode operasi non-PWM. Jika bit-bit ini di-set maka akan memaksa terjadinya compare match.

Bit 1 dan 0 (WGM11 dan WGM10) Waveform Generation Mode, kedua bit ini bersamaan dengan bit WGM13:2 dalam register TCCR1B berguna untuk memilih mode operasi yang akan kita gunakan.

Tabel 2.11
Mode Operasi Waveform Generation Mode pada Timer/Counter1

WGM13	WGM12	WGM11	WGM10	Mode Operasi	TOP	Update OCR1x	Set Flag TOV1
0	0	0	0	Normal	0xFFFF	Immediat	MAX
0	0	0	1	PWM Phase Correct 8-bit	0x00FF	TOP	BOTTOM
0	0	1	0	PWM Phase Correct 9-bit	0x01FF	TOP	BOTTOM
0	0	1	1	PWM Phase Correct 10-bit	0x03FF	TOP	BOTTOM
0	1	0	0	CTC	OCR1A	Immediat	MAX
0	1	0	1	Fast PWM 8-bit	0x00FF	BOTTOM	TOP
0	1	1	0	Fast PWM 9-bit	0x01FF	BOTTOM	TOP
0	1	1	1	Fast PWM 10-bit	0x03FF	BOTTOM	TOP

1	0	0	0	PWM Phase & Frequency Correct	ICR1	BOTTOM	BOTTOM
1	0	0	1	PWM Phase & Frequency Correct	OCR1A	BOTTOM	BOTTOM
1	0	1	0	PWM Phase Correct	ICR1	TOP	BOTTOM
1	0	1	1	PWM Phase Correct	OCR1A	TOP	BOTTOM
1	1	0	0	CTC	ICR1	Immediate	MAX
1	1	0	1	Not used	-	-	-
1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	W	W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.18
Register TCCR1B^[2]

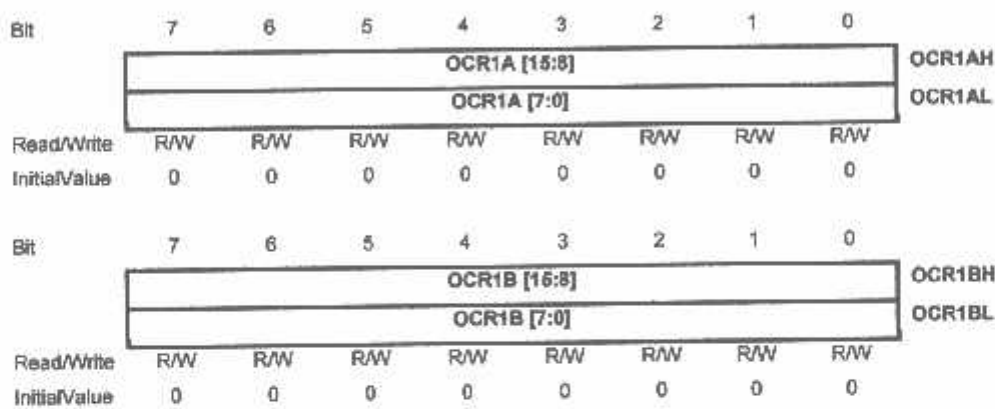
Bit 7 (ICNC1) *Input Capture Noise Canceler*, penge-set-an bit ini akan mengaktifkan *Input Capture Canceler* pada saat menggunakan mode normal yang *capture event*. Dimana *noise canceler* akan memfilter *trigger* yang masuk ke pin ICP1 akan disaring selama 4 siklus *clock*, jika selama 4 siklus *clock* tersebut berubah maka akan diabaikan.

Bit 6 (ICES1) *Input Capture Edge Select*, bit ini mendefinisikan *trigger* yang masuk ke pin ICP1(PB0) yang digunakan untuk menangkap kejadian (*capture event*). Jika ICES1=0 maka *falling edge* (perpindahan dari 1 ke 0) digunakan sebagai *trigger* dan jika ICES=1 maka *rising edge* (perpindahan dari 0 ke 1) digunakan sebagai *trigger*.

Bit 0 s/d 2 (CS10,CS11 dan CS12) *Clock Select*, bit-bit ini bertugas untuk memilih / mendefinisikan / *prescaler* pulsa / *clock* yang masuk ke register TCNT1.

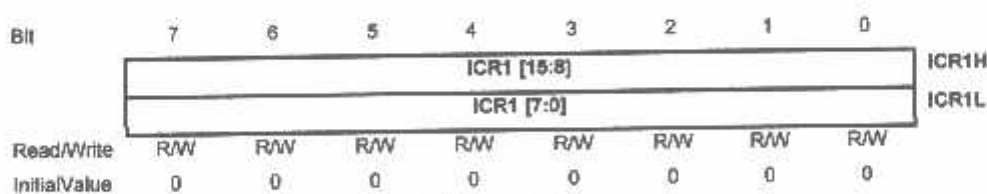
Tabel 2.12
Prescaler timer/counter1

CS02	CS01	CS00	Keterangan
0	0	0	Stop/Berhenti
0	0	1	Clk (sama dengan <i>clock</i> CPU)
0	1	0	Clk/8 (1 clk <i>timer/counter0</i> = 8 clk CPU)
0	1	1	Clk/64 (1 clk <i>timer/counter0</i> = 64 clk CPU)
1	0	0	Clk/256 (1 clk <i>timer/counter0</i> = 256 clk CPU)
1	0	1	Clk/1024 (1 clk <i>timer/counter0</i> = 1024 clk CPU)
1	1	0	Clk eksternal dari pin T1 pada kondisi <i>falling edge</i>
1	1	1	Clk eksternal dari pin T1 pada kondisi <i>rising edge</i>



Gambar 2.19
 Register OCR1A dan OCR1B^[2]

Register ini bertugas sebagai register pembanding yang bisa kita tentukan besarnya sesuai dengan kebutuhan. Dalam prakteknya pada saat TCNT1 (TCNT1H:TCNT1L) mencacah maka otomatis oleh CPU akan dibandingkan dengan isi OCR1 (OCR1H:OCR1L) secara kontinyu dan jika isi TCNT1 sama dengan isi OCR1 maka akan terjadi *compare match* yang dapat dimanfaatkan untuk mode CTC dan PWM.



Gambar 2.20
 Register ICR1^[2]

Register ICR1 (ICR1H:ICR1L) akan selalu diperbarui dengan isi *register* pencacah TCNT1 (pada saat tersebut) sewaktu terjadi *trigger (capture event)* pada pin ICP1. Register ICR1 juga mempunyai fungsi lain untuk mendefinisikan TOP *value* pada mode tertentu (lihat table 2.11 mode operasi).

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.21
Register TIMSK untuk *Timer/Counter1*^[2]

Bit 5 (TICIE1) *Input Capture Interrupt Enable*, bit ini berguna untuk meng-aktifkan interupsi *input capture* (penangkap kejadian pada pin ICP1/PB0) ketika bit ini di-set.

Bit 4 (OCIE1A) *Output Compare A Match Interrupt Enable*, bit ini berguna untuk meng-aktifkan interupsi *Output Compare A Match* ketika bit ini di-set.

Bit 3 (OCIE1B) *Output Compare B Match Interrupt Enable*, bit ini berguna untuk meng-aktifkan interupsi *Output Compare B Match* ketika bit ini di-set.

Bit 2 (TOIE1) *Timer/Counter1, Overflow Interrupt Enable*, bit ini berguna untuk mengaktifkan interupsi *overflow* TCNT1 ketika bit ini di-set.

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.22
Register TIFR untuk *Timer/Counter1*^[2]

Bit 5 (ICF1) *Input Capture Flag*, bit ini akan set secara otomatis ketika menangkap *trigger* pada pin ICP. Bit ini akan clear juga secara otomatis ketika mengeksekusi vector interupsi *input capture*. Untuk meng-clear secara manual bit ini maka bit ini harus di-set.

Bit 4 (OCF1A) *Output Compare A Match Flag*, bit ini akan set secara otomatis ketika terjadi *compare match A*. Bit ini akan clear juga secara otomatis ketika mengeksekusi vektor interupsi *output compare A*. Untuk meng-clear secara manual bit ini maka bit ini harus di-set.

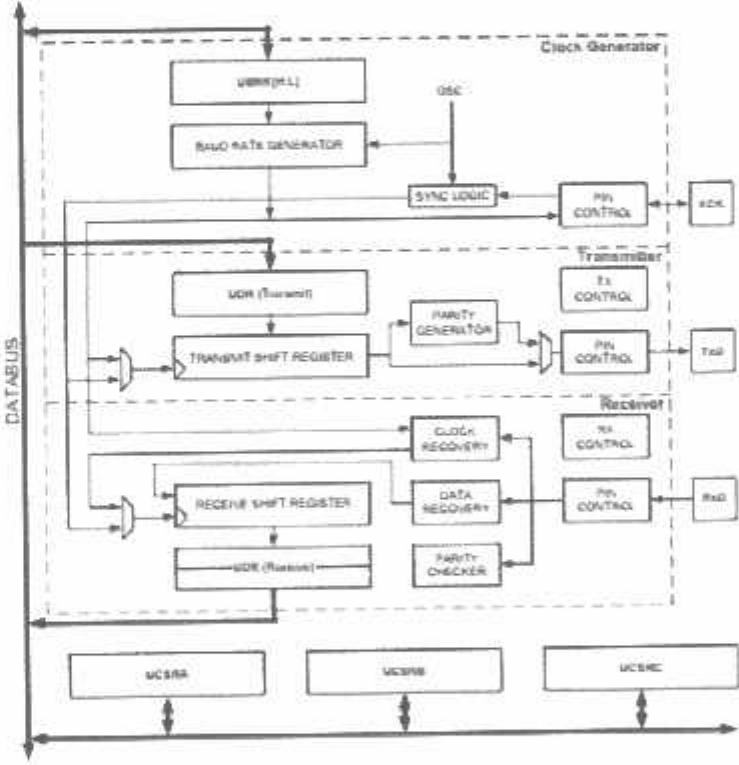
Bit 3 (OCF1B) *Output Compare B Match Flag*, bit ini akan set secara otomatis ketika terjadi *compare match B*. Bit ini akan *clear* juga secara otomatis ketika mengeksekusi vektor interupsi *output compare B*. Untuk meng-*clear* secara manual bit ini maka bit ini harus di-set.

Bit 2 (TOV1) *Timer Overflow Flag*, bit ini akan set secara otomatis ketika terjadi *overflow* pada register pencacah TCNT1. Bit ini akan *clear* juga secara otomatis ketika mengeksekusi vektor interupsi *overflow timer/counter1*. Untuk meng-*clear* secara manual bit ini maka bit ini harus di-set.

2.3.4 USART Mikrokontroler ATmega 32^[2]

USART dapat difungsikan sebagai transmisi data sinkron, dan asinkron. Sinkron berarti *clock* yang digunakan antara transmitter dan *receiver* satu sumber *clock*. Sedangkan asinkron berarti transmitter dan *receiver* mempunyai sumber *clock* sendiri-sendiri.

USART terbagi dalam tiga blok yaitu *clock generator*, *transmitter* dan *receiver*.



Gambar 2.23
Blok Diagram USART^[2]

Clock generator, berhubungan dengan kecepatan transfer data (*baud rate*), register yang bertugas menentukan *baud rate* adalah register pasangan UBRR.

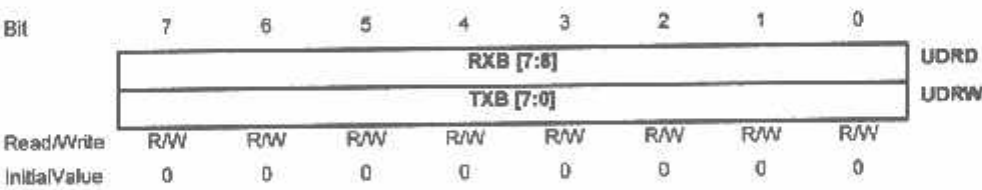
Tabel 2.13
Baudrate

Operating Mode	Equation for Calculating Baud Rate	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X=0)	$BAUD = \frac{f_{osc}}{16(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X=1)	$BAUD = \frac{f_{osc}}{8(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{osc}}{2(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{2BAUD} - 1$

Dimana, Fosc adalah frekuensi osilator yang digunakan. BAUD adalah transfer bit per detik.

USART *transmitter*, berhubungan dengan pengiriman data pada pin Tx. Perangkat yang sering digunakan seperti register UDR sebagai tempat penampung data yang akan ditransmisikan, *flag* TXC sebagai akibat dari data yang ditransmisikan telah sukses (*complete*), dan *flag* UDRE sebagai indikator jika UDR kosong dan siap untuk diisi data yang akan ditransmisikan lagi.

USART *receiver*, berhubungan dengan penerimaan data pada pin Rx. Perangkat yang sering digunakan seperti register UDR sebagai tempat penampung data yang telah diterima, dan *flag* RXC sebagai indikator bahwa data telah sukses (*complete*) diterima.



Gambar 2.24
Register UDR^[2]

Register UDR digunakan untuk *buffer* data yang akan dikirim dan *buffer* data yang diterima. UDR untuk *buffer transmitter* hanya dapat ditulis ketika bit/*flag* UDRE dalam register UCSRA set.

Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.25
Register UCSRA^[2]

Bit 7 (RXC) *USART Receive Complete*, flag ini akan set ketika data yang masuk (lengkap/complete) ke dalam UDR belum dibaca dan akan *clear* ketika sudah di baca atau kosong. Flag ini dapat digunakan untuk membangkitkan interupsi RX complete (jika diaktifkan) ketika set dan akan *clear* secara otomatis bersamaan dengan vektor interupsi yang bersangkutan. Flag ini juga dapat digunakan untuk *pooling* data yang masuk dan flag ini dapat di *clear* dengan memberi logika satu.

Bit 6 (TXC) *USART Transmit Complete*, flag ini akan set ketika data yang akan dikirim/transmit telah keluar. Flag ini dapat digunakan untuk membangkitkan interupsi TX complete (jika diaktifkan) ketika set dan akan *clear* secara otomatis bersamaan dengan vektor interupsi yang bersangkutan. Flag ini juga dapat digunakan untuk *pooling* data yang masuk dan flag ini dapat di *clear* dengan memberi logika satu.

Bit 5 (UDRE) *USART Data Register Empty*, flag ini sebagai indikator isi UDR. Jika flag set maka UDR dalam keadaan kosong dan siap menerima ditulis/isi dengan data berikutnya, jika flag clear berarti sebaliknya. Flag ini dapat digunakan untuk membangkitkan interupsi data register empty (jika diaktifkan) ketika set dan akan *clear* secara otomatis bersamaan dengan eksekusi vektor interupsi yang bersangkutan.

Bit 4 (FE) *Frame Error*, flag ini sebagai indikator ketika data yang diterima error, misalnya ketika stop bit pertama data dibaca berlogika nol maka flag akan set. Flag akan clear ketika stop bit data yang diterima berlogika satu.

Bit 3 (DOR) *Data OverRun*, flag ini berfungsi untuk mendeteksi data yang tumpang tindih (*overrun*). Flag akan set ketika terjadi *overrun*.

Bit 2 (PE) *Parity Error*, flag ini berfungsi untuk mendeteksi *parity error*. Flag akan set ketika terjadi *bit parity* yang error apabila *parity bit* digunakan.

Bit 1 (U2X) *Double the USART transmission speed*, bit ini hanya berguna untuk mode operasi *asynchronous*. Bit ini berguna untuk melipatkan *baud rate* yang digunakan.

Bit 0 (MPCM) *Multi-processor Communication Mode*, bit ini berfungsi untuk mengaktifkan mode *multi-processor communication*, dimana ketika data yang diterima oleh USART tidak mengandung informasi alamat akan diabaikan.

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.26
Register UCSRB^[2]

Bit 7 (RXCIE) *RX Complete Interrupt Enable*, bit ini berfungsi untuk mengaktifkan interupsi *RX complete* (1=*enable*, 0=*disable*). Ketika satu data diterima dalam register UDR sudah lengkap (tanpa *error*) maka otomatis akan meng set *flag* RXC dalam register UCSRA dan kemudian membangkitkan interupsi *RX complete*.

Bit 6 (TXCIE) *TX Complete Interrupt Enable*, bit ini berfungsi untuk mengaktifkan interupsi *TX complete* (1=*enable*, 0=*disable*). Ketika satu data dikirim oleh register UDR sudah lengkap (tanpa *error*) maka otomatis akan meng set *flag* TXC dalam register UCSRA dan kemudian membangkitkan interupsi *TX complete*.

Bit 5 (UDRIE) *USART Data Register Empty Interrupt Enable*, bit ini berfungsi untuk mengaktifkan interupsi data *register empty* (UDR kosong, 1=*enable*, 0=*disable*). Ketika data yang kita tulis ke UDR telah dikirimkan kemudian UDR akan kosong, hal ini yang dimanfaatkan untuk membangkitkan interupsi data *register empty*.

Bit 4 (RXEN) *Receiver Enable*, bit ini berfungsi untuk mengaktifkan pin RX saluran USART (1=*enable*, 0=*disable*). Dimana ketika diaktifkan maka pin tersebut tidak bisa digunakan untuk fungsi pin I/O karena sudah digunakan sebagai saluran penerima USART.

Bit 3 (TXEN) *Transmitter Enable*, bit ini berfungsi untuk mengaktifkan pin TX saluran USART (1=*enable*, 0=*disable*). Dimana ketika diaktifkan maka pin tersebut tidak bisa digunakan untuk fungsi pin I/O karena sudah digunakan sebagai saluran pemancar USART.

Bit 2 (UCSZ2) *Character Size*, bit ini bersama dengan bit UCSZ1,UCSZ0 dalam register UCSRC digunakan untuk memilih format lebar data bit (*character size*) yang digunakan.

Bit 1 (RXB8) *Receive Data Bit 8*, bit ini digunakan sebagai bit ke-8 ketika menggunakan format data 9-bit, dan bit ini harus dibaca dahulu sebelum membaca UDR.

Bit 0 (TXB8) *Transmit Data Bit 8*, bit ini digunakan sebagai bit ke-8 ketika menggunakan format data 9-bit, dan bit ini harus ditulis dahulu sebelum menulis ke UDR.

Bit	7	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.27
Register UCSRC^[2]

Register UCSRC dan register UBRRH berbagi lokasi yang sama dalam register I/O.

Bit 7 (URSEL) *Register Select*, berfungsi untuk memilih antara register UCSRC dengan UBRRH, dimana untuk menulis atau membaca register UCSRC maka bit ini harus di set (secara *default*/keadaan awal sudah set). Sedangkan untuk mengakses register UBRRH maka bit ini harus di *clear*.

Bit 6 (UMSEL) *USART Mode Select*, bit ini berfungsi untuk memilih mode USART (*Asynchronous* / UMSEL=0, atau *Synchronous* / UMSEL=1).

Bit 5 dan 4 (UPM1 dan UPM0) *Parity Mode*, bit ini berfungsi untuk memilih *mode parity* yang akan digunakan. Secara otomatis transmitter USART akan membuat *parity* yang dipilih. Dan kita tidak perlu tahu seperti apa yang penting bagi kita adalah data dapat dikirim dan diterima dengan baik (hal ini untuk vadilitas data, jika bit *parity* tidak diterima oleh *Receiver* UASRT maka data tidak dianggap karena *error/corrupt* dan *flag* PE dalam (UCSRA akan set).

Tabel 2.14
Parity

UPM1	UPM0	<i>Mode Parity</i>
0	0	<i>Disable</i>
0	1	Tidak digunakan
1	0	<i>Enable, Even Parity</i>
1	1	<i>Enable, Odd Parity</i>

Bit 3 (USBS) *Stop Bit Select*, bit ini berfungsi untuk memilih jumlah *stop* bit (1-*stop* bit: USBS=0, 2-*stop* bit USBS=1). Banyaknya *stop* bit yang digunakan hanya untuk vadilitas data dan ini urusan USART kita tidak perlu tahu seperti apa yang penting data tidak hilang dan diterima dengan baik.

Bit 2 dan 1 (UCSZ1 dan UCSZ0) *Character Size*, bit ini berfungsi untuk memilih lebar data bit yang digunakan dikombinasikan dengan bit UCSZ2 dalam register UCSRB.

Tabel 2.15
Opsi Lebar Data USART

UCSZ2	UCSZ1	UCSZ0	Lebar Data
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Tidak digunakan
1	0	1	Tidak digunakan
1	1	0	Tidak digunakan
1	1	1	9-bit

Bit 0 (UCPOL) *Clock Polarity*, bit ini berguna hanya untuk mode *synchronous*. Bit ini berhubungan dengan perubahan *output* data dan sampel *input*, dan *clock synchronous* (XCK).

Tabel 2.16
Polaritas *clock*

UCPOL	Perubahan data transmisi (pin output TxD)	Sampel data (pin input RxD)
0	<i>Rising XCK Edge</i>	<i>Falling XCK Edge</i>
1	<i>Falling XCK Edge</i>	<i>Rising XCK Edge</i>

Bit	15	14	13	12	11	10	9	8	
	URSEL	-	-	-	UBRR [11:8]				UBRRH
	UBRR [7:0]								UBRRL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Gambar 2.28
Register UBRR^[2]

Bit 15 (URSEL) *Register Select*, bit ini berguna untuk memilih register yang akan diakses, UBRRH atau UCSRC. Bit ini dibaca nol ketika membaca UBRRH. Bit URSEL harus di *clear* ketika menulis ke UBRRH.

Bit 14:12 *Reserved Bits*, bit-bit ini tidak digunakan dan harus di *clear* ketika menulis ke UBRRH.

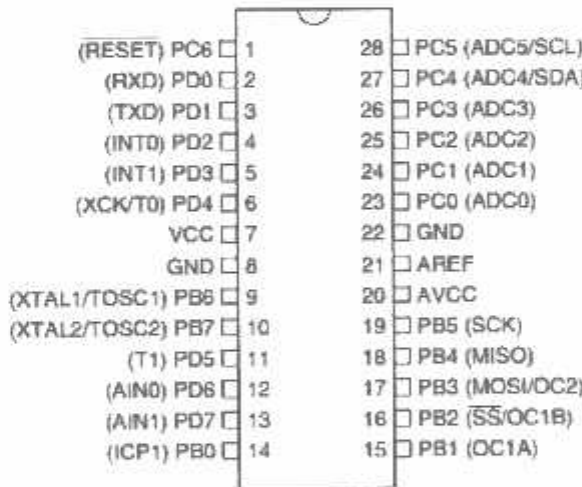
Bit 11:10 *USART Baud Rate Register*, ke 12-bit ini digunakan untuk menentukan *baudrate* USART.

2.4 Mikrokontroler ATmega 8^[10]

Mikrokontroler ATmega8 memiliki fitur seperti (ADC internal, EEPROM internal, Timer/Counter, Watchdog Timer, PWM, Port I/O, komunikasi serial, Komparator, I2C,dll).

Berikut ini merupakan beberapa spesifikasi ATmega8 :

- 1. Memiliki kapasitas flash memori 8Kbyte, EEPROM 512 Byte, dan SRAM 1Kbyte.
- 2. Saluran Port I/O sebanyak 23 buah, yaitu Port B, Port C, dan Port D.
- 3. CPU yang terdiri atas 32 buah register.
- 4. Interupsi internal dan eksternal.
- 5. Port USART sebagai komunikasi serial.
- 6. Konsumsi daya rendah (DC 5V).
- 7. Fitur peripheral, yang terdiri dari :
 - a. Tiga buah Timer/Counter dengan perbandingan
 - b. 3 channel PWM.
 - c. 6 channel, 10-bit ADC



Gambar 2.29
Pin-Out ATmega8^[10]

2.5 Transistor^[4]

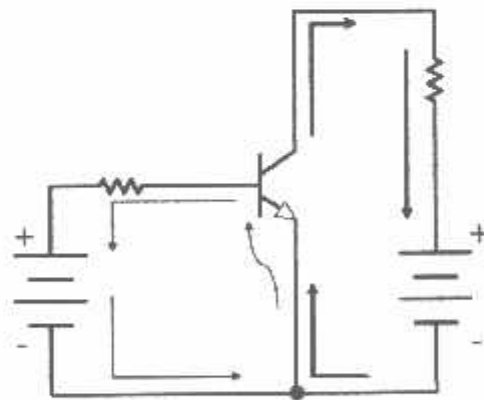
Transistor adalah komponen semikonduktor yang mempunyai tiga kaki atau lebih sehingga daya dapat diperkuat. Fungsi transistor sebagai penguat atau *amplifier* dari sinyal listrik, tahanan variabel atau sebagai saklar.

2.5.1 Transistor Sambungan Bipolar^[4]

Banyak sistem elektronis yang sangat tergantung pada kemampuan transistor untuk bertindak sebagai saklar. Transistor yang digunakan sebagai saklar mempunyai keuntungan yaitu tidak mempunyai bagian yang berputar, yang dapat beroperasi ON dan OFF pada kecepatan tinggi, memerlukan tegangan dan arus penggerak yang sangat rendah untuk memicu aksi penghubungan.

Gambar 2.10 menggambarkan aksi penghubungan transistor sambungan bipolar (NPN). Emitor yang diberi banyak bahan campuran bertindak sebagai sumber utama dari arus elektron. Basis dengan sedikit bahan campuran bertindak untuk mengontrol aliran arus. Pada transistor NPN, kolektor diberi dengan bahan yang cukup dan menerima sebagian besar elektron dari emitor. Arus pada ujung basis disebut arus basis dan arus pada ujung kolektor disebut arus kolektor. Jumlah arus basis (I_B) menentukan jumlah arus kolektor (I_C). Dengan tidak adanya arus basis tidak ada arus kolektor (Normally OFF). Sedikit kenaikan pada (I_B) mengakibatkan kenaikan yang besar pada (I_C), jadi arus basis bertindak mengontrol jumlah arus kolektor.

Gain arus adalah perbandingan arus kolektor terhadap arus basis. Transistor sambungan bipolar mempunyai dua variabel: NPN dan PNP. Aksi dari masing-masing adalah sama tetapi polaritasnya terbalik.



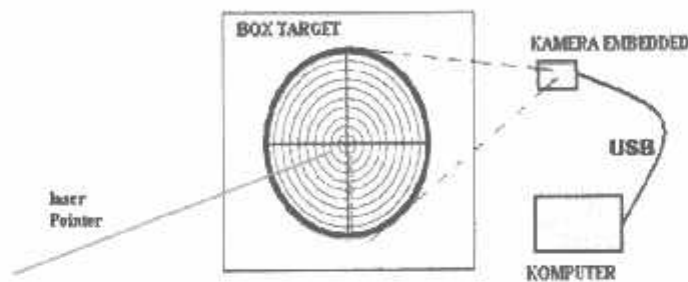
Gambar 2.30
Transistor sambungan bipolar (BJT)^[4]

BAB III
PERENCANAAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas perancangan dari keseluruhan sistem yang akan di buat dalam tugas akhir ini.

3.1 Rancangan Sistem Keseluruhan

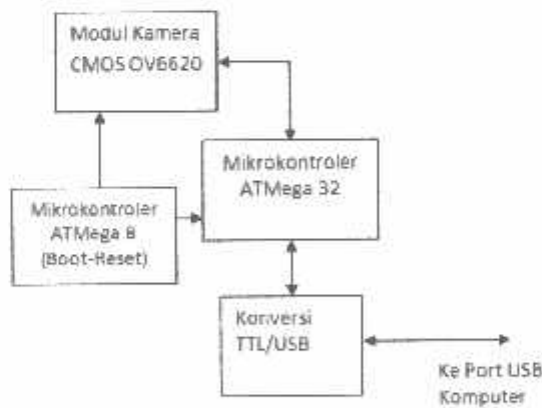
Sistem ini menggunakan kamera embedded yang di implementasikan pada *box target*^[9] simulator menembak :



Gambar 3.1
Diagram Keseluruhan Simulator Menembak menggunakan *box target*^[9]

Pada simulator menembak ini menggunakan sensor kamera sebagai pendeteksi laser yang telah terpasang pada senjata. Fungsi komputer pada sistem ini sebagai penampil posisi hasil tembakan dan menghitung nilai skor.

3.2 Diagram Sistem Kamera Embedded



Gambar 3.2
Blok Diagram Sistem

Fungsi dari tiap blok diagram dijelaskan sebagai berikut :

1. Modul Kamera CMOS OV6620

Modul ini berfungsi merubah setiap hasil pencitraan menjadi data digital.

2. Mikrokontroler ATmega 32

Berfungsi inialisasi awal kamera, dan menjalankan instruksi (*Command*) dari komputer/host.

3. Mikrokontroler ATmega 8 (boot-reset) Berfungsi sebagai boot reset mikrokontroler ATmega 32 dengan Kamera.

4. Konversi TTL/USB

Berfungsi sebagai mengubah komunikasi serial TTL ke USB agar data yang di kirim kompatibel dengan komputer / laptop.

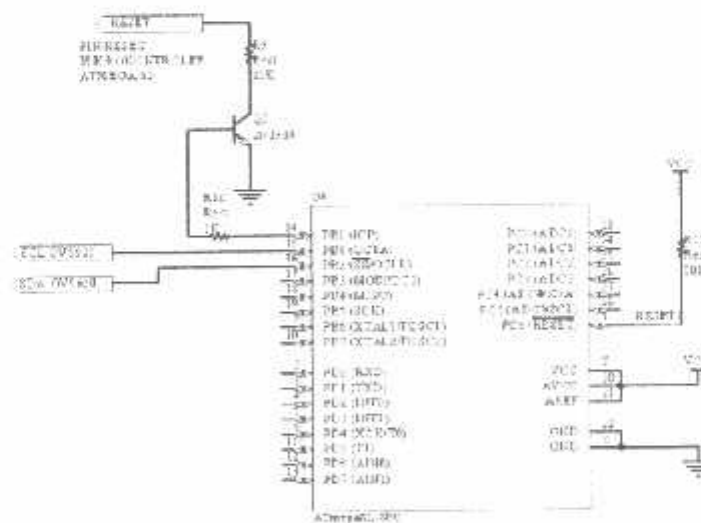
3.3 Perancangan Perangkat Keras

3.3.1 Perancangan Rangkaian Mikrokontroler ATmega 8

Pada modul kamera CMOS OV6620, sudah terdapat clock internal untuk sistem kerjanya, yaitu sebesar 17,7344 Mhz. Untuk menyamakan clock dengan mikrokontroler ATmega 32 maka diperlukan rangkaian ini Dimaksudkan agar kamera dan mikrokontroler ATmega 32 bekerja pada clock yang sama. Sehingga data ouput piksel dari kamera tidak hilang (*corrupt*).

Keluaran dari mikrokontroler ATmega 8 dihubungkan ke pin RESET mikrokontroler ATmega 32 melalui transistor 2N3904 sebagai saklar.

Rangkaian boot-reset ATmega 8 dapat dilihat sebagai berikut :



Gambar 3.3
Perancangan Rangkaian ATmega 8

Untuk rancangan rangkaian transistor, dapat dihitung sebagai berikut :

Diketahui β_{dc} 2N3904 = 100, $V_{be} = 0,75$ V, tegangan pd pin reset mikrokontroler ATmega 32 ialah 5Vdc, $R_c = 10K$, maka arus kolektor didapat :

$$I_c = \frac{V}{R_c} = \frac{5}{10} = 0,5mA$$

Untuk mengetahui arus basis :

$$\beta_{dc} = \frac{I_c}{I_b} \longrightarrow I_b = \frac{I_c}{\beta_{dc}}$$

$$I_b = \frac{I_c}{\beta_{dc}} = \frac{0,5}{100} = 0,005mA$$

Diketahui tegangan output Port ATmega 8 = 5Vdc, maka besaran untuk resistor basis

$$R_b = \frac{5 - 0,75}{0,005} = 850\Omega$$

Maka Resistor basis sebesar 850Ω atau yang mendekati yaitu 1K.

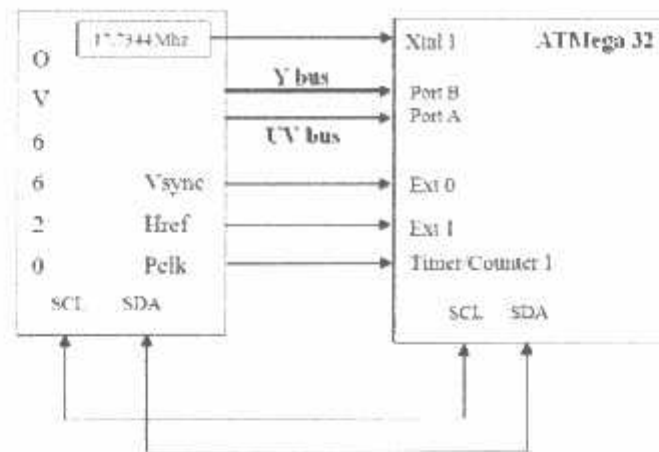
3.3.2 Perancangan Antarmuka Kamera ke Mikrokontroler

Pada sistem kerja kamera CMOS OV6620, output data piksel dikirim secara paralel, ada 2 port yaitu Y dan UV. Port Y untuk data piksel warna hijau (*green*) dan Port UV untuk data piksel merah dan biru (*Red and Blue*). Pada mikrokontroler ATmega 32, Port Y dihubungkan ke Port A dan Port UV dihubungkan ke Port B.

Untuk sistem kontrol, V_{sync} dihubungkan ke Interupsi Eksternal 0 untuk mengetahui awalan citra. H_{ref} dihubungkan ke Interupsi Eksternal 1 untuk mendeteksi setiap awalan baris dan dihubungkan ke Counter 0 untuk menghitung kolom citra. P_{clk} dihubungkan ke Counter 1 untuk menghitung data piksel yang diterima.

Untuk set register, menggunakan antarmuka I2C dihubungkan ke PortC.0 dan PortC.1 mikrokontroler.

Perancangan antarmuka kamera CMOS OV6620 ke Mikrokontroler ATmega 32 dapat digambarkan sebagai berikut :



Gambar 3.4
Perancangan Antarmuka Kamera ke Mikrokontroler

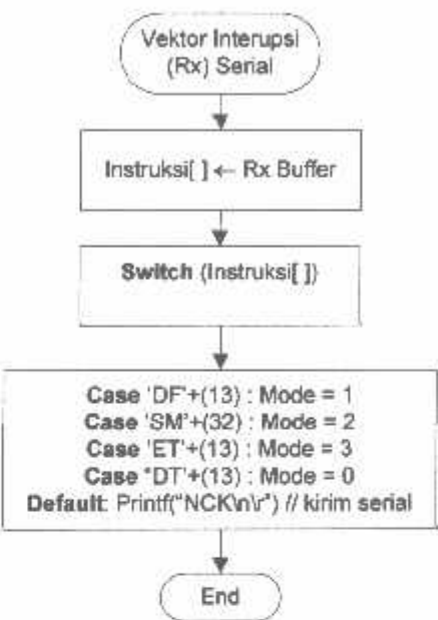
3.4 Perancangan Perangkat Lunak

3.4.1 Set Instruksi

Ada 4 instruksi yang dirancang pada perangkat kamera embedded ini yakni :

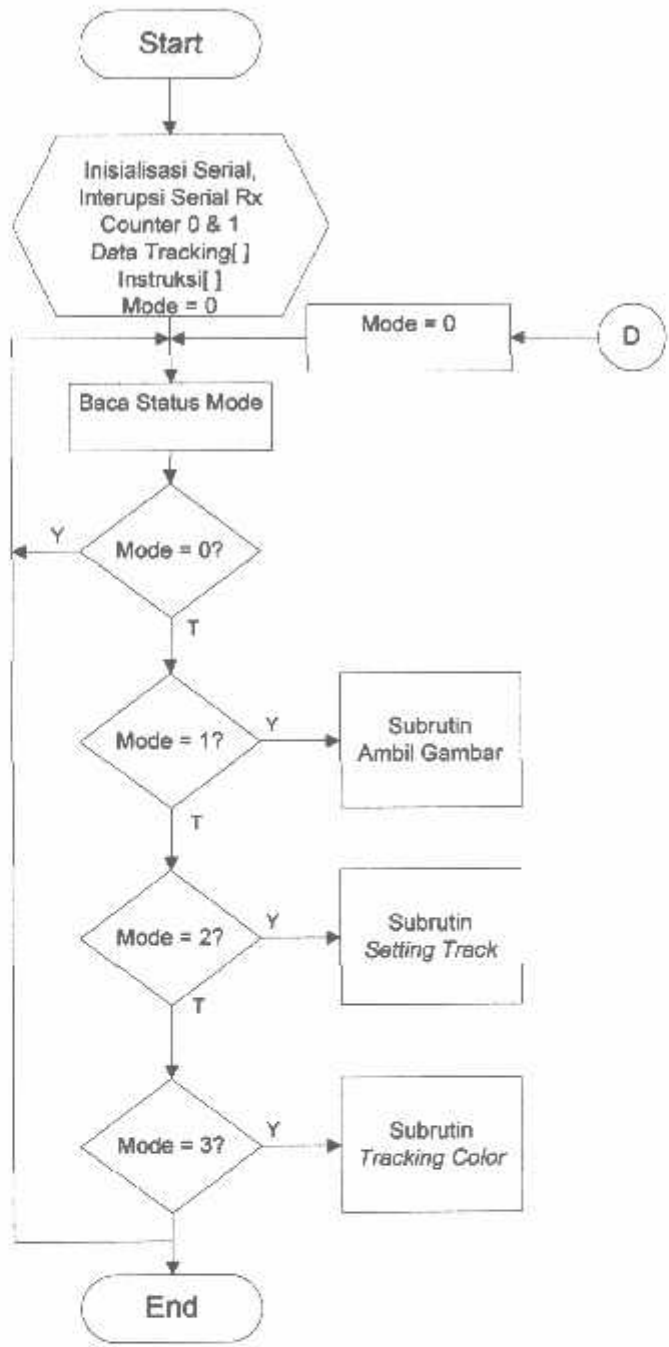
1. **'DF'+(cr)**
(*Dumping Frame*) Kirim Data Gambar.
2. **'SM'+(sp)+Nilai Track [Merah]min+(sp)+ Nilai Track [Merah]maks+(sp)+ Nilai Track[Hijau]min+(sp)+ Nilai Track[Hijau]maks +(sp)+Nilai Track[Biru]min +(sp) + Nilai Track[Biru]maks(cr)**
(*Set map color*) Set Warna yang ingin di track, diikutkan nilai track.
Contoh untuk track warna merah ;
SM 120 240 0 0 0 0
3. **'ET'+(cr)**
(*Enable Tracking*) Tracking warna aktif.
4. **'DT'+(cr)**
(*Disable Tracking*) Tracking warna stop

3.4.2 Interupsi Serial



Gambar 3.5
Flowchart bagian interupsi serial

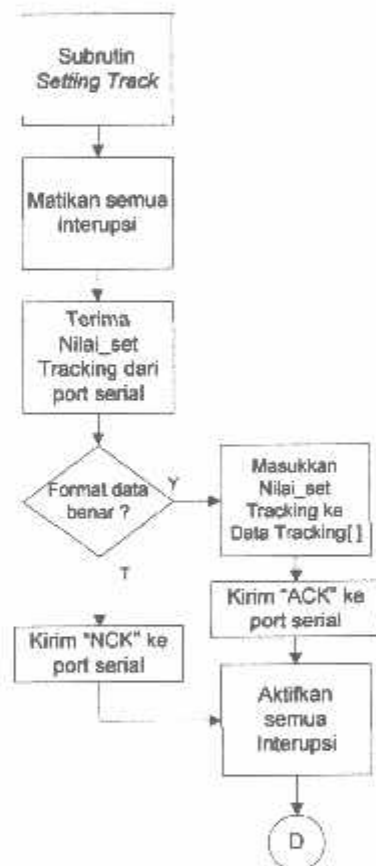
3.4.3 Program Utama



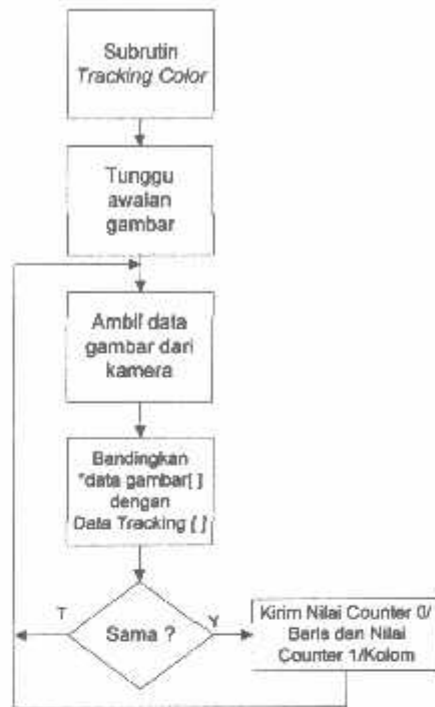
Gambar 3.6
Flowchart Program Utama



Gambar 3.7
Flowchart Subrutin Kirim Data Gambar



Gambar 3.8
Flowchart Subrutin Setting Track



Gambar 3.9
Flowchart Subrutin *Tracking Color*

3.4.4 Format Keluaran Data Kamera Embedded

Format data keluaran dari kamera embedded sebagai berikut :

- Format data gambar

0x0F	Baris	Data piksel 1 ...176
------	-------	----------------------

0x0F = Indikasi setiap awalan baris

Baris = Nilai baris yang dikirim, dalam bentuk heksa desimal

Data Piksel = Data piksel yang dikirim (H,B,H,M,...)

Baris dari 0x01 s/d 0x90.

- Format tracking warna

Jika warna terdeteksi maka format data outputnya :

0x0A	r	c	0xFF
------	---	---	------

0x0A = Indikasi awalan tracking

r = Nilai posisi baris, dalam bentuk heksadesimal

c = Nilai posisi kolom, dalam bentuk heksadesimal

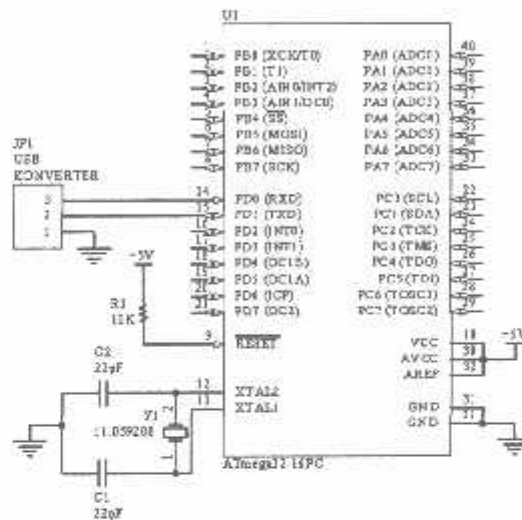
0xFF = Akhiran tracking

BAB IV

PENGUJIAN SISTEM

4.1 Pengujian Komunikasi Serial

Pengujian komunikasi serial ini bertujuan untuk menguji kesesuaian karakter yang dikirimkan oleh program mikrokontroler dengan hasil pada komputer dengan menggunakan *software* hyperterminal.



Gambar 4.1
Rangkaian Komunikasi Serial

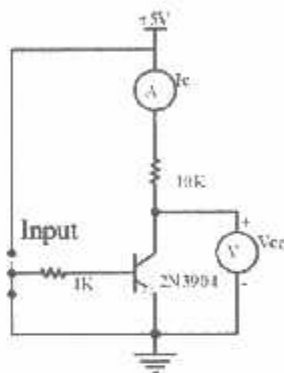


Gambar 4.2
Hasil pengujian komunikasi serial

Dari hasil pengujian pada *software* hyperterminal, didapatkan hasil yang sesuai dengan program mikrokontroler mengirim karakter “KOMUNIKASI SERIAL BERHASIL”.

4.2 Pengujian Rangkaian Transistor Sebagai Saklar

Tujuan pengujian ini ialah menguji apakah transistor mampu men-*trigger* pin reset pada mikrokontroler ATmega 32. Dimana besar tegangan pacu (*trigger*) untuk pin reset sebesar 5V.



Gambar 4.3
Pengujian Rangkaian Transistor 2N3904

Data hasil pengujian rangkaian dapat dilihat pada tabel dibawah ini :

Tabel 4.1
Hasil Pengukuran Rangkaian Transistor 2N3904 Sebagai Saklar

No	Input (V)	Vce (V)	Ic (mA)
1	0	5	0
2	5	0,01	0,49

Analisa hasil pengujian adalah sebagai berikut :

- Jika Input dihubungkan dengan ground atau input = 0, maka :
Arus kolektor tidak mengalir ke emitor sehingga $I_c = 0$,
Hasil perhitungan tegangan kolektor-emitor (V_{ce}) :
$$V_{ce} = V_{cc} - (I_c.R_c)$$
$$= 5 - (0.10000)$$
$$= 5 - 0 = 5 \text{ V}$$
- Jika Input dihubungkan dengan +5 Vdc, maka :
Arus kolektor dapat dihitung sebagai berikut

$$I_c = \frac{V_{cc}}{R_c} = \frac{5}{10} = 0,5mA$$

Dan tegangan kolektor-emitor (V_{ce}) dapat diperoleh :

$$\begin{aligned} V_{ce} &= V_{cc} - (I_c.R_c) \\ &= 5 - (0,5.10) \\ &= 5 - 5 = 0 \text{ V} \end{aligned}$$

Dari analisa diatas dapat tabelkan pengukuran dan perhitungan transistor 2N3904.

Tabel 4.2
Pengukuran dan Perhitungan Rangkaian Transistor 2N3904 Sebagai Saklar

No.	Input (V)	Pengukuran		Perhitungan	
		Vce (V)	Ic (mA)	Vce (V)	Ic (mA)
1.	0	5	0	5	0
2.	5	0,01	0,49	0	0,5

Sehingga presentase kesalahan dapat dihitung dengan persamaan berikut ini :

$$\%error = \frac{HasilPengukuran - HasilPerhitungan}{HasilPerhitungan} \times 100\%$$

Pada Saat input = 0

- V_{ce}

$$\%error = \frac{5 - 5}{5} \times 100\% = 0\%$$

- I_c

$$\%error = \frac{0 - 0}{0} \times 100\% = 0\%$$

Pada Saat input = 5 Volt

- V_{ce}

$$\%error = \frac{0,01 - 0}{0} \times 100\% = 0,01\%$$

- I_c

$$\%error = \frac{0,49 - 0,5}{0,5} \times 100\% = 0,01\%$$

Dari hasil diatas dapat disimpulkan bahwa tidak ada error atau kesalahan jika input transistor (0 Volt), ini disebabkan rangkaian transistor belum bekerja. Sedangkan

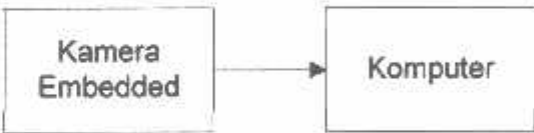
jika input transistor dihubungkan ke 5 Volt terdapat kesalahan atau error yaitu sebesar 0,01% untuk Vce dan 0,01% untuk Ic.

Tabel 4.3
Kesalahan (Tingkat Error) Rangkaian Transistor 2N3904 Sebagai Saklar

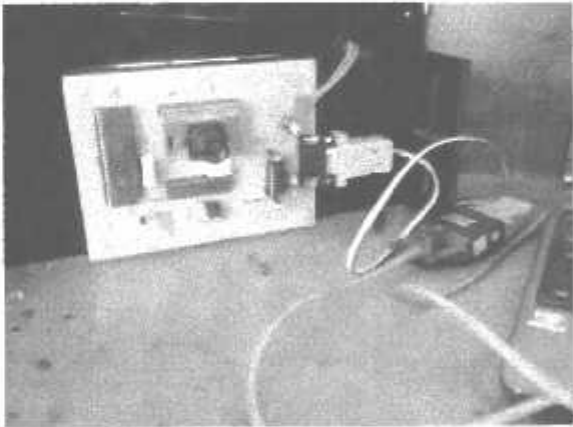
No.	Input (V)	Pengukuran		Perhitungan		Error	
		Vce (V)	Ic (mA)	Vce (V)	Ic (mA)	Vce (%)	Ic (%)
1.	0	5	0	5	0	0	0
2.	5	0,01	0,49	0	0,5	0,01	0,01

4.3 Pengujian Pengambilan Gambar Dari Kamera Embedded

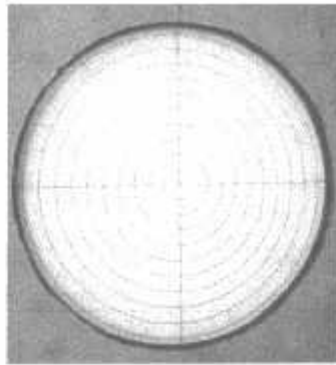
Pengujian ini merupakan pengujian pengambilan gambar dari Kamera Embedded dan ditampilkan dengan komputer.



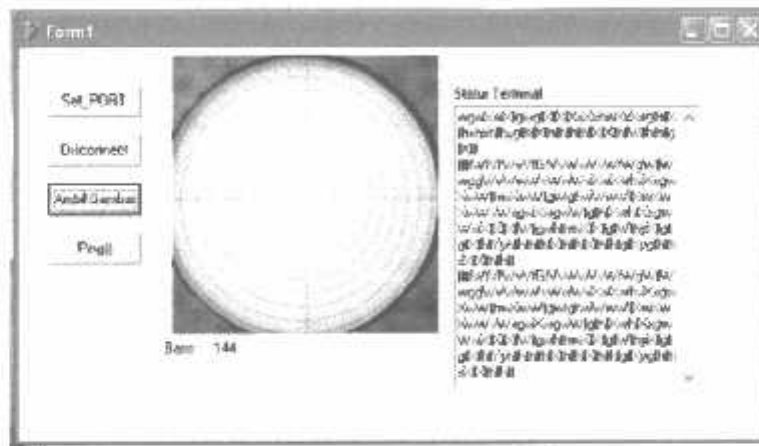
Gambar 4.4
Blok diagram pengujian



Gambar 4.5
Foto Pengujian



Gambar 4.6
Objek menggunakan kamera digital



Gambar 4.7
Hasil pengujian menampilkan gambar dari kamera embedded

Sebelum dilakukan pengujian, kamera embedded yang sudah terprogram dengan baik dihubungkan ke komputer melalui komunikasi serial. Gambar (4.4) merupakan blok diagram pengujian. Persiapan pengujian dilakukan seperti pada Gambar (4.5).

Sebelum menguji gambar dari kamera embedded, terlebih dahulu menggunakan kamera digital sebagai perbandingan dengan kamera embedded. Gambar (4.6) pengambilan objek menggunakan kamera digital.

Selanjutnya pada komputer telah dipersiapkan sebuah program untuk menampilkan gambar dari kamera embedded. Gambar (4.7) merupakan hasil pengujian menampilkan gambar dari kamera embedded dengan menggunakan program tersebut. Untuk mengetahui adanya data dari kamera embedded, pada status terminal di program tersebut data gambar di ubah menjadi karakter. Ini hanya sebagai indikasi bahwa ada data masuk ke komputer dari kamera embedded.

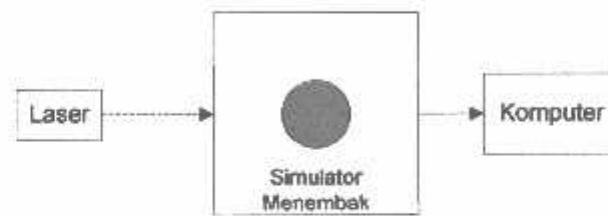
Dari hasil pengujian pengambilan gambar. Objek yang ditampilkan sama. Dan data gambar dari sensor kamera dapat diterima oleh mikrokontroler dengan baik.

4.4 Pengujian Sistem

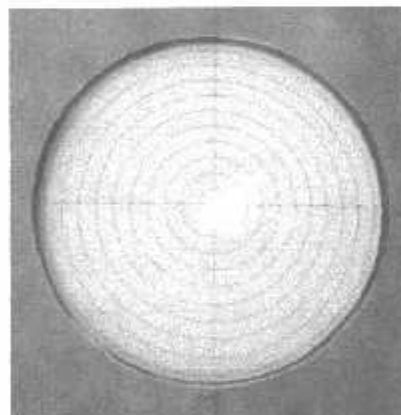
Pengujian sistem ini dilakukan untuk mengetahui hasil dari sistem yang dibuat. Hasil yang diamati adalah kesesuaian hasil tembakan pada sistem kamera embedded dengan dengan *software* di komputer.

Langkah pengujian yang dilakukan sebagai berikut :

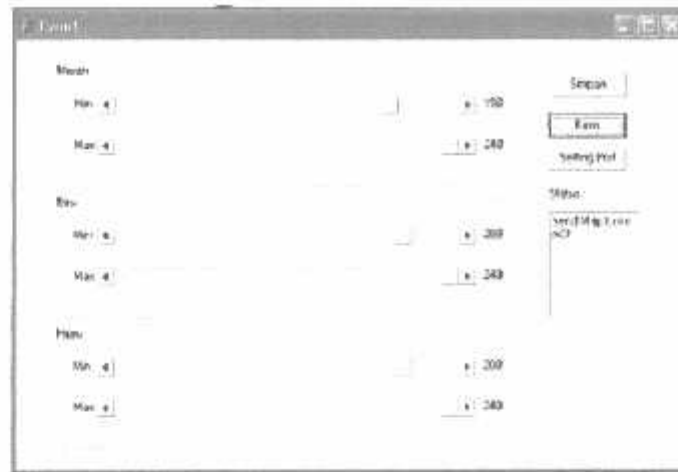
- Menghubungkan antara alat dengan komputer melalui konektor USB.
- Melakukan *setting* warna yang ingin di *tracking*. Pada sistem ini ialah warna titik sinar laser.
- Melakukan penembakan dengan sinar laser.



Gambar 4.8
Diagram blok pengujian sistem



Gambar 4.9
Pengujian deteksi warna sinar laser

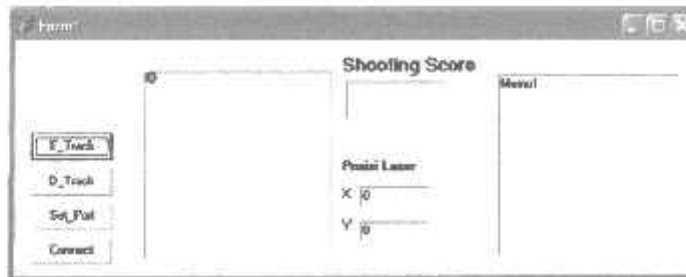


Gambar 4.10
Melakukan setting *Map Tracking Color*

Pada Gambar (4.9) melakukan pengujian deteksi warna titik sinar laser. Seperti pada gambar tersebut, titik tengah dari sinar laser ini cenderung berwarna putih. Pada Gambar (4.10) dibuat sebuah program *user interface setting map tracking color* (setting deteksi warna) berfungsi memasukkan data *range* dari masing-masing tiga warna primer yang akan dideteksi oleh kamera embedded ini pada saat melakukan deteksi warna. Pada pengujian ini besaran *range* untuk tiga warna primer masing-masing saat deteksi sinar laser ialah : Merah = 199 s/d 240, Biru = 208 s/d 240, Hijau = 208 s/d 240.

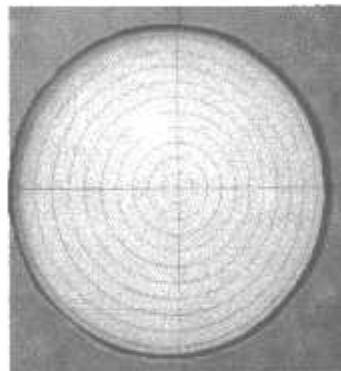
4.5 Hasil Pengujian Sistem

Setelah memasukkan besaran *range* data tiga warna primer masing-masing untuk mendeteksi sinar laser pada kamera embedded ini. Maka dilakukan pengujian keseluruhan. Dengan menggunakan program seperti pada gambar (4.11), pertama memberikan perintah ke sistem kamera embedded untuk mendeteksi warna yang di peroleh dari sensor kamera dengan besaran *range* data yang sudah di setting.



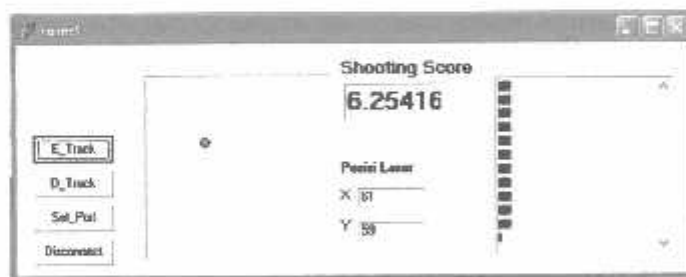
Gambar 4.11
Program Antarmuka Simulator Menembak
Memberikan Perintah Mendeteksi Sinar Laser ke Kamera Embedded

Selanjutnya melakukan pengujian menembak sinar laser seperti pada gambar (4.12).



Gambar 4.12
Pengujian menembak sinar laser

Jika besaran nilai data dari sensor kamera sama dengan besaran *range* data yang telah disetting tadi, maka akan muncul hasil nilai skor pada program seperti pada gambar (4.13). Sistem kamera embedded, dalam hal ini yaitu mikrokontroler ATmega 32 akan mengirim data posisi laser ke komputer.



Gambar 4.13
Hasil pengujian pada *software* simulator menembak

Dari hasil tembakan terlihat hasil posisi sinar laser dari kamera embedded yang ditampilkan oleh *software*.

Untuk mendapatkan nilai *score* seperti pada Gambar (4.13) maka menggunakan persamaan matematika pada referensi pustaka [5] dan [9], Yaitu :

$$Score = 10 - \left(\left(10 \sqrt{(Cx_2 - Cx_1)^2 + (Cy_2 - Cy_1)^2} \right) / r \right) \quad (4.1)$$

Dimana : Cx_1 : Titik tengah posisi koordinat x target lingkaran.

Cy_1 : Titik tengah posisi koordinat y target lingkaran.

Cx_2 : Posisi titik koordinat x sinar laser.

Cy_2 : Posisi titik koordinat y sinar laser.

r : Radius,

Radius dapat diperoleh dengan persamaan sebagai berikut :

$$r = (Panjang sumbu X) \frac{10}{22} \quad (4.2)$$

Penggunaan persamaan (4.1) dan (4.2) pada sistem kamera embedded ini, untuk mendapatkan nilai *score* bisa dijelaskan sebagai berikut :

Diketahui resolusi kamera embedded ini adalah 176 x 144 piksel, x = 176 dan y = 144. Maka :

- Titik tengah koordinat x :

$$Cx_1 = \frac{176}{2} = 88$$

- Titik tengah koordinat y :

$$Cy_1 = \frac{144}{2} = 72$$

- Radius:

$$r = 176 \times \frac{10}{22} = \frac{1760}{22} = 80$$

Pada saat melakukan pengujian dengan sinar laser, kamera embedded mengirim posisi laser ke *software* komputer seperti pada gambar (4.13) yaitu posisi x = 61 dan posisi y = 59 atau ($Cx_2 = 61$ dan $Cy_2 = 59$). Maka memasukkan nilai-nilai tersebut ke persamaan (4.1) :

$$Score = 10 - \left(\left(10 \sqrt{(Cx_2 - Cx_1)^2 + (Cy_2 - Cy_1)^2} \right) / r \right)$$

$$Score = 10 - \left(\left(10\sqrt{(61-88)^2 + (59-72)^2} \right) / 80 \right)$$

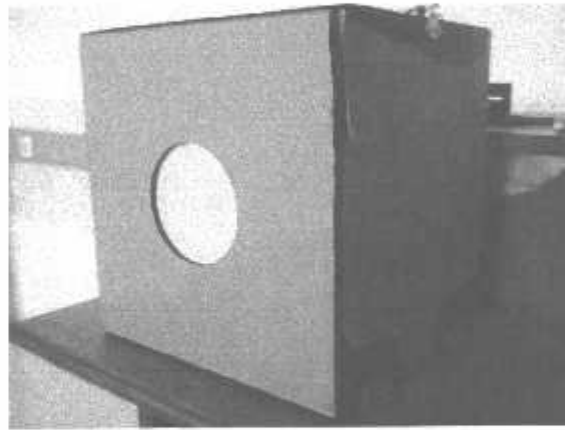
$$Score = 10 - \left(\left(10\sqrt{729 + 169} \right) / 80 \right)$$

$$Score = 10 - (299,66 / 80)$$

$$Score = 10 - (3,745)$$

$$Score = 6,255$$

Dari hasil perhitungan nilai *score* diatas dengan tampilan nilai *score* pada *software* komputer di Gambar (4.13) sama, dengan menggunakan persamaan matematika mengacu pada referensi pustaka [5] dan [9]. Maka sistem kamera embedded yang telah dibuat ini bisa di implementasikan pada *box target* simulator menembak.



Gambar 4.14
Implementasi Kamera Embedded pada *Box Simulator Menembak*^[9]

BAB V

PENUTUP

5.1 Kesimpulan

Setelah dilakukan proses perancangan dan pembuatan serta pengujian alat, maka dapat diambil kesimpulan :

1. Kamera Embedded bekerja dengan baik pada resolusi 176 x 144 piksel. Atau pada opsi ke 2, format resolusi sensor kamera OV6620.
2. Pada simulator menembak, nilai posisi titik koordinat diambil dari nilai counter0 pada ATmega 32 untuk baris dan nilai counter1 pada ATmega 32 untuk kolom. Setelah perbandingan data masing-masing warna primer pada sensor kamera dengan data 3 warna primer yang di setting untuk sinar laser sama. Dimana data 3 warna primer untuk deteksi sinar laser pada pengujian ini adalah :Merah = 199 s/d 240, Biru = 208 s/d 240, Hijau = 208 s/d 240.
3. Mikrokontroler ATmega8 berfungsi mengaktifkan bersama antara sensor kamera dan mikrokontroler ATmega 32. Dalam frekuensi *clock* yang sama. Untuk mengaktifkan mikrokontroler ATmega 32 digunakan rangkaian transistor 2N3904 sebagai saklar dimana kaki kolektor terhubung ke pin reset ATmega 32 dengan tingkat kesalahan (*error*) saat saturasi Vce (Tegangan Kolektor-Emitor) 0,01% dan Ic (Arus Kolektor) 0,01%.

5.2 Saran

1. Penggunaan kamera embedded ini di usahakan jangan menghadap langsung ke sumber cahaya misalkan lampu / matahari, karena akan mengganggu sistem *tracking* warna.
2. Untuk rancang bangun aplikasi *portable* lain dengan kamera embedded ini, maka perlu diperhatikan sistem komunikasi data kamera embedded ini. Agar bisa bekerja dengan baik.

DAFTAR PUSTAKA

- [1] Pitowarno, Endra., 2006. *Mikroprosesor dan Interfacing*. Yogyakarta: Penerbit Andi
- [2] Winoto, Ardi., 2008. *Mikrokontroler AVR ATmega 8/32/16/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR*. Bandung: Penerbit INFORMATIKA
- [3] Wijaya, Marvin Ch., Prijono, Agus., 2007. *Pengolahan Citra Digital Menggunakan MATLAB*. Bandung : Penerbit INFORMATIKA
- [4] Petruzella, Frank D. *Elektronik Industri Edisi Bahasa Indonesia*. Yogyakarta: Penerbit Andi
- [5] Soetedjo, Aryuanto., Nurcahyo, Eko., Februari 2011. "Developing of Low Cost Vision-Based Shooting Range Simulator". Departmen of Electrical Engineering, National Institute of Technology, Malang, Indonesia: IJCSNS International of Journal Computer Science and Network Security 11,2:109-113.
- [6] Atmel, 2003. *Datasheet Microcontroller ATMEGA32*, URL: <http://www.atmel.com/Images/doc2503.pdf>
- [7] Omnivison Technologies, 2003. *Datasheet OV6620/6120*, URL: <http://www.cs.cmu.edu/~cmucam/Downloads/ov6620DSL.F.PDF>
- [8] COMedia, C3088 1/4" Color Camera Module With Digital Output, URL: <http://www.datasheetarchive.com/C3088-datasheet.html>
- [9] Soetedjo, Aryuanto., Mahmudi, Ali., Ashari, M. Ibrahim., Nakhoda, Yusuf Ismail., Desember 2013. *Aplikasi Sistem Embedded Pada Simulator Menembak Senjata Ringan*. Laporan Penelitian Strategi Nasional (STRANAS)
- [10] Atmel, 2013. *Datasheet Microcontroller ATMEGA8A*, URL: http://www.atmel.com/Images/Atmel-8159-8-bit-AVR-microcontroller-ATmega8A_datasheet.pdf

LAMPIRAN



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT.BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No.2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SKRIPSI FAKULTAS TEKNOLOGI INDUSTRI

NAMA : Muhammad Rusdi Azhari
NIM : 07.12.907
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Elektronika
MASA BIMBINGAN : SEMESTER GANJIL 2013/2014
JUDUL : RANCANG BANGUN KAMERA EMBEDDED BERBASIS
MIKROKONTROLER ATMEGA 32 DAN OMNIVISION OV6620
PADA SIMULATOR MENEMBAK

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Rabu
Tanggal : 19 Februari 2014
Dengan Nilai : 92,00 (A)

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji

M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358

Sekretaris Majelis Penguji

Dr. Eng. Aryanto Soetedjo, ST, MT
NIP.P. 1030800417

ANGGOTA PENGUJI

Dosen Penguji I

Bambang Prio Hartono, ST, MT
NIP.Y. 1028400082

Dosen Penguji II

M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi Teknik Elektronika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Muhammad Rusdi Azhari
NTM : 07.12.907
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Elektronika
MASA BIMBINGAN : SEMESTER GANJIL 2013/2014
JUDUL : RANCANG BANGUN KAMERA EMBEDDED BERBASIS
MIKROKONTROLER ATMEGA 32 DAN OMNIVISION OV6620
PADA SIMULATOR MENEMBAK

Tanggal	Uraian	Paraf
Penguji I 19 Februari 2014	1. Abstrak tambahkan hasil penelitian (Kesimpulan). 2. Untuk Bab IV semua data dalam bentuk tabel beri penjelasan.	
Penguji II 19 Februari 2014		

Disetujui,

Dosen Penguji I

Bambang Prio Hartono, ST, MT
NIP.Y. 1028400082

Dosen Penguji II

M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358

Mengetahui,

Dosen Pembimbing I

Dr. Eng. Aryanto Soetedjo, ST, MT
NIP.P. 1030800417

Dosen Pembimbing II

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 1018800189



FORMULIR BIMBINGAN SKRIPSI

NIM : 07.12.907
Nama : Muhammad Rusdi Azhari
Masa Bimbingan : Semester Ganjil Tahun Akademik 2013-2014
Judul Skripsi : **Rancang Bangun Kamera Embedded Berbasis Mikrokontroler ATmega 32 dan Omnicision OV6620 Pada Simulator Menembak**

No.	Tanggal	Uraian	Paraf Pembimbing
1.	30-09-2013	BAB I Pendahuluan : Latar belakang, Tujuan, Batasan Masalah	
2.	17-10-2013	BAB II : Penjelasan tentang OV 6620, spesifikasi, cara kerja dan antarmuka. Spesifikasi ATmega 32 .	
3.	12-11-2013	ACC BAB I dan BAB II BAB III : Blok Diagram, Perhitungan Transistor, <i>Flowchart</i>	
4.	07-01-2014	ACC BAB III BAB IV : Pengujian komunikasi serial ATmega 32, pengambilan gambar dari kamera embedded, pengujian dengan sinar laser.	
5.	20-01-2014	ACC BAB IV Pengujian Alat, Perhitungan Skor BAB V : Kesimpulan dan Saran	
6.	23-01-2014	ACC BAB V Di lanjutkan ke seminar hasil dan sidang skripsi	

Malang, Januari 2014
Dosen Pembimbing,

(Dr. Eng. Arvianto Soetedjo, ST, MT)
NIP.P. 1030800417

Form.S-4b



FORMULIR BIMBINGAN SKRIPSI

NIM : 07.12.907
Nama : Muhammad Rusdi Azhari
Masa Bimbingan : Semester Ganjil Tahun Akademik 2013-2014
Judul Skripsi : Rancang Bangun Kamera Embedded Berbasis Mikrokontroler
ATMega 32 dan Omnivision OV6620 Pada Simulator Menembak

No.	Tanggal	Uraian	Paraf Pembimbing
1.	02-10-2013	REVISI BAB I : Pendahuluan	
2.	18-10-2013	REVISI BAB II : Landasan Teori	
3.	15-11-2013	ACC BAB I dan BAB II REVISI BAB III : Perencanaan dan Pembuatan Alat	
4.	10-01-2014	ACC BAB III REVISI BAB IV : Pengujian Sistem	
5.	20-01-2014	ACC BAB IV REVISI BAB V : Penutup	
6.	23-01-2014	ACC BAB V Di lanjutkan ke seminar hasil dan sidang skripsi	

Malang, Januari 2014
Dosen Pembimbing,

(Ir. Yusuf Ismail Nakhoda, MT)
NIP.Y. 1018800189

Form.S-4b



```

/*****
This program was produced by the
CodeWizardAVR V2.05.3 Standard
Automatic Program Generator
© Copyright 1998-2011 Pavel Haiduc, HF InfoTech s.r.l.
http://www.hpinfotech.com
Project :
Version :
Date    : 3/1/2013
Author  : Rusdi
Company : Rusdi_camera
Comments:
Chip type      : ATmega8
Program type   : Application
AVR Core Clock frequency: 1.000000 MHz
Memory model   : Small
External RAM size : 0
Data Stack size : 256
*****/

```

```

#include <mega8.h>
#include <delay.h>

```

```

// I2C Bus functions
#include <i2c.h>

```

```

#define reset_ctrl PORTB.0
#define cam_addr 0xC0
#define reg_1 0x13
#define dat_1 0x05
#define reg_2 0x3F
#define dat_2 0x42

```

```

// Declare your global variables here

```

```

void tri_state()
{
    i2c_start();
    i2c_write(cam_addr);
    i2c_write(reg_1);
    i2c_write(dat_1);
    i2c_stop();
}

```

```

void ext_clock()
{
    i2c_start();
    i2c_write(cam_addr);
    i2c_write(reg_2);
    i2c_write(dat_2);
    i2c_stop();
}

```

```

void main(void)
{
    PORTB=0x01;
    DDRB=0x01;
    // I2C Bus initialization
    // I2C Port: PORTB
    // I2C SDA bit: 2
    // I2C SCL bit: 1

```

```
// Bit Rate: 100 kHz
// Note: I2C settings are specified in the
// Project|Configure|C Compiler|Libraries|I2C menu.
i2c_init();

delay_ms(1000);
tri_state();
delay_ms(1000);
ext_clock();
delay_ms(1000);
while (1)
{
    // Place your code here
    reset_ctrl=0;
}
}
```

```

//Program Kamera Embedded : Rusdi
//*****
#include <mega32.h>
#include <delay.h>
//#include <math.h>
#define UV PINB
#define Y PINA
#define baris TCNT0
#define slave_camera 0xC0
//#define lampu PORTD.6
#define href PIND.3
#define vsync PIND.2
#define pclk PINB.1
#define awal 0x6F
#define hidup 1
#define mati 0
#define true 1
#define false 0

#define merah 0
#define hijau 0x10
#define biru 0x20

// I2C Bus functions
#include <i2c.h>
bit status,current_state,current_statel,less=0;

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here

}

// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
// Place your code here
// if (status==true)
// {
//   less=~less;
// }

}

#define RXB0 1
#define TXB0 0
#define UFE 2
#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<OVR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

```

```

// USART Receiver buffer
#define RX_BUFFER_SIZE 8
char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE<256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
unsigned int rx_wr_index,rx_rd_index,rx_counter;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
{
rx_buffer[rx_wr_index]=data;
if (++rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
if (++rx_counter == RX_BUFFER_SIZE)
{
rx_counter=0;
rx_buffer_overflow=1;
};
};
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter==0);
data=rx_buffer[rx_rd_index];
if (++rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
asm("cli")
--rx_counter;
asm("sei")
return data;
}
#pragma used-
#endif

// Standard Input/Output functions
#include <stdio.h>

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Reinitialize Timer 0 value
TCNT0=0x6E;
current_state=false;
status=false;
// Place your code here

```

```

}

// Timer 1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    // Reinitialize Timer 1 value
    TCNT1H=0xFF;
    TCNT1L=0x4E;

    current_state1=false;
    current_state=true;
    // Place your code here

}

void cam_init();
void debug_serial();
void get_model();
void get_mode3();
void get_mode2();
void capture_image();
void debug_track();
void set_col();
void get_stop();
void print_ack();
void print_nck();

unsigned char txcomm[15];
void camera_set_register(unsigned char addr_reg,unsigned char set_reg)
{
    i2c_start();
    i2c_write(slave_camera);
    i2c_write(addr_reg);
    i2c_write(set_reg);
    i2c_stop();
}

unsigned char trmerah,trhijau,trbiru,track;
unsigned char color_map[48];
// Declare your global variables here
void main(void)
{
    // Declare your local variables here
    char detec,cont;

    // Input/Output Ports initialization
    // Port A initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
    Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T
    State0=T
    PORTA=0x00;
    DDRA=Cx00;

    // Input/Output Ports initialization
    // Port B initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
    Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T
    State0=T
    PORTB=Cx00;

```

```

DDRB=0x00;

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=Out Func5=In Func4=In Func3=In Func2=In Func1=In
// Func0=In
// State7=T State6=0 State5=T State4=T State3=T State2=T State1=T
// State0=T
PORTD=0x00;
DDRD=0x40;

// Timer/Counter 0 initialization
// Clock source: T0 pin Rising Edge
TCCR0=0x07;
TCNT0=0x6E;

// Timer/Counter 1 initialization
// Clock source: T1 pin Rising Edge
// Mode: Normal top=FFFFh
// OClA output: Discon.
// OClB output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x07;
TCNT1H=0xFF;
TCNT1L=0x4E;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Rising Edge
// INT1: On
// INT1 Mode: Rising Edge
GICR|=0xC0;
MCUCR=0x0F;
GIFR=0xC0;

```

```

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x05;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 115200
UCSRA=0x02;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
UBPRL=0xE6;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFI CR=0x00;

// I2C Bus initialization
i2c_init();
cam_init();
detec=0x00;
// Global enable interrupts
#asm("sei")

while (1)
{
    // Place your code here
    if (detec!=0x0d)
    {
        detec=getchar();
        cont++;
        txcomm[cont] = detec;
    }
    else
    {
        detec=0x00;
        cont=0;
        debug_serial();
    }
}

);

void cam_init()
{
    camera_set_register(0x14,0x20); //set frame 176 x 144
    delay_ms(500);
    camera_set_register(0x39,0x40); //pclk,vsync,href --> output
    delay_ms(500);
    camera_set_register(0x12,0x28); //set output RGB no AWB
    delay_ms(500);
    camera_set_register(0x28,0x05); //set out port: Y->G UV->BR
    delay_ms(500);
    camera_set_register(0x28,0x81); //one-line port Y
    delay_ms(500);
    camera_set_register(0x13,0x01); //auto adjust mode

```

```

    delay_ms(500);
}

void debug_serial()
{
    char poling;
    poling=1;
    if (txcomm[poling]=='D')
        (get_model());
    else if
        (txcomm[poling]=='S')
        (get_mode2());
    else if (txcomm[poling]=='E')
        (get_mode3());
    else
        (print_nck());
}

void get_model()
{
    char set;
    set=1;
    set++;
    if (txcomm[set]=='F')
    {set++;
    if (txcomm[set]==0x0D)
    {
        print_ack();
        capture_image();
    }

    else
    if (txcomm[set]=='T')
    {set++;
    if (txcomm[set]==0x0D)
    {
        get_stop();
    }
    }

    print_nck();
}

void get_mode3()
{
    char set;
    set=1;
    set++;
    if (txcomm[set]=='T')
    {set++;
    if (txcomm[set]==0x0D)
    {
        print_ack();
        debug_track();
    }

    else
    print_nck();
}

void get_mode2()

```

```

{
    char set;
    set=1;
    set++;
    if (txcomm[set]=='M')
    {set++;
        if (txcomm[set]==0x20)
            set_col();}
    else
        print_nck();
}

void capture_image()
{
    unsigned char y,x;
    camera_set_register(0x11,0x01);
    while(vsync!=true){};
    while(vsync!=false){};
    while(baris!=awal){};
    status=true;
    current_state=true;
    putchar(0x0F);

    for (y=0;y<144;y++)
    {
        while (href==true);
        putchar (baris);
        for (x=0;x<176;x++)
        {
            while(pclk==true);
            putchar(Y);
            while(pclk==false)
            while(pclk==true);
            putchar(UV<<4);
            while(pclk==false);
        }
    }
}

void debug_track()
{
    unsigned char a,b,c;
    while(track==true)
    {
        while(vsync!=true){};
        while(vsync!=false){};
        while(baris!=awal){};
        status=true;
        current_state=true;
        //
        trhi_jau = Y;
        a = color_map[trhi_jau+hijau];
        #asn("nop")
        trmerah = UV;
        b = color_map[trmerah+merah];
        #asn("nop")
        trbiru = UV;
        c = color_map[trbiru+biru];
        //
        if (a || b || c)
        {

```

```

        putchar(0x0A);
        putchar(TCNT0);
        putchar(TCNT1L);
        putchar(0xFF);
    }
}

void set_col()
{
    unsigned char a,redmin,redmax,greenmin,greenmax,bluemin,bluemax;
    redmin=getchar();
    while(getchar()!=0x20);
    redmax=getchar();
    while(getchar()!=0x20);
    greenmin=getchar();
    while(getchar()!=0x20);
    greenmax=getchar();
    while(getchar()!=0x20);
    bluemin=getchar();
    while(getchar()!=0x20);
    blucmax=getchar();
    print_ack();
    for (a=redmin;a<=redmax;a++)
    {
        color_map[a+merah]=1;
    }
    for (a=greenmin;a<=greenmax;a++)
    {
        color_map[a+hi]au]=1;
    }
    for (a=bluemin;a<=bluemax;a++)
    {
        color_map[a+biru]=1;
    }
}

void get_stop()
{
    track=false;
}

void print_ack()
{
    printf("ACK");
    putchar(10);
}

void print_nck()
{
    printf("NCK");
    putchar(10);
}

```

OV6620 SINGLE-CHIP CMOS CIF COLOR DIGITAL CAMERA OV6120 SINGLE-CHIP CMOS CIF B&W DIGITAL CAMERA

Features

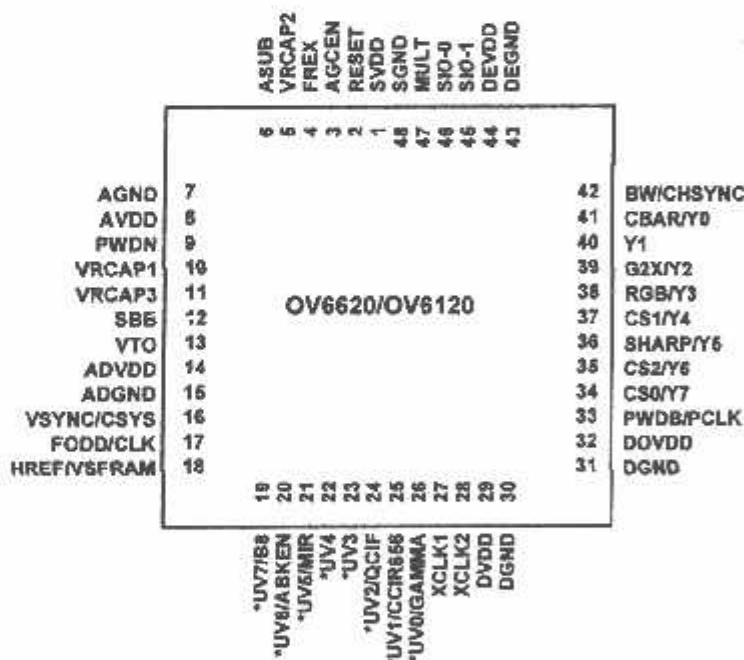
- 101,376 pixels, 1/4" lens, CIF/QCIF format
- Progressive scan read out
- Data format - YCrCb 4:2:2, GRB 4:2:2, RGB Raw Data
- 8/16 bit video data: CCIR601, CCIR656, ZV port
- Wide dynamic range, anti-blooming, zero smearing
- Electronic exposure / Gain / white balance control
- Image enhancement - brightness, contrast, gamma, saturation, sharpness, window, etc.
- Internal/external synchronization
- Frame exposure/line exposure option
- 5-Volt operation, low power dissipation
 - < 80 mW active power
 - < 10 μ A in power-save mode
- Gamma correction (0.45/0.65/1.00)
- SCCB programmable (400 kb/s):
 - color saturation, brightness, contrast, white balance, exposure time, gain

General Description

The OV6620 (color) and OV6120 (black and white) CMOS image sensors are single-chip video/imaging camera devices designed to provide a high level of functionality in a single, small-footprint package. Both devices incorporate a 352 x 288 image array capable of operating up to 60 frames per second image capture. Proprietary sensor technology utilizes advanced algorithms to cancel Fixed Pattern Noise (FPN), eliminate smearing, and drastically reduce blooming. All needed

camera functions including exposure control, gamma, gain, white balance, color matrix, windowing, and more, are programmable through an SCCB (Serial Camera Control Bus) interface. Both devices can be programmed to provide image output in either 4-, 8- or 16-bit digital formats.

Applications include: Video Conferencing, Video Phone, Video Mail, Still Image, and PC Multimedia.



* Note: Outputs UV0-UV7 are not available on the OV6120. The inputs associated with these respective pins are still functional.

OV6620/OV6120 PIN ASSIGNMENTS

Array Elements (CIF) (QCIF)	356 x 282 (176 x 144)
Pixel Size	9.0 x 8.2 μ m
Image Area	3.1 x 2.5 mm
Max Frames/Sec	Up to 60 FPS
Electronic Exposure	Up to 500 : 1 (for selected FPS)
Scan Mode	progressive
Gamma Correction	0.45/0.55/1.0
Min. Illumination (3000K)	OV6620 - < 3 lux @ f1.2 OV6120 - < 0.5 lux @ f1.2
S/N Ratio (Digital Camera Out)	> 48 dB (AGC = Off, Gamma = 1)
FPN	< 0.03% V _{p-p}
Dark Current	< 0.2 nA/cm ²
Dynamic Range	> 72 dB
Power Supply	5VDC, $\pm$ 5% (Anal.) 5VDC or 3.3VDC (DIO)
Power Requirements	< 80mW Active < 30 μ W Standby
Package	48 pin LCC

Table 1. Pin Description

Pin No.	Name	Pin Type	Function/Description
01	SVDD	V _{in}	Array power (+5VDC)
02	RESET	Function (Default = 0)	Chip Reset, active high
03	AGCEN	Function (Default = 0)	Automatic Gain Control (AGC) selection "0" - Disable AGC "1" - Enable AGC NOTE: This function is disabled when OV6620/OV6120 sensor is configured in SCCB mode.
04	FREX	Function (Default = 0)	Frame Exposure Control "0" - Disable Frame Exposure Control "1" - Enable Frame Exposure Control
05	VRCAP2	V _{ref} (2.5V)	Array reference. Connect to ground through 0.1 uF capacitor.
06	ASUB	V _{in}	Analog substrate voltage
07	AGND	V _{in}	Analog ground
08	AVDD	V _{in}	Analog power supply (+5VDC)
09	PWDN	Function (Default = 0)	Power down mode selection "0" - normal mode "1" - power down mode
10	VrCAP1	N/C	Internal voltage reference. Connect to ground through 0.1 uF capacitor.
11	VrCAP3		Internal voltage reference. Connect to ground through 1 uF capacitor.
12	SBB	Function (Default = 0)	SCCB (Serial Camera Control Bus) enable selection "0" - Enable SCCB "1" - Enable autocontrol mode
13	VTO	O	Luminance Composite Signal Output
14	ADVDD	V _{in}	Analog power supply (+5VDC)
15	ADGND	V _{in}	Analog signal ground
16	VSYS/CSYS	I/O	Vertical sync output. At power up, read as CSYS.
17	FODD/CLK	I/O	Field ID FODD output or main clock output
18	HREF/VSFRAM	I/O	HREF output. At power up, read as VSFRAM
19	*UV7/B8	I/O	Bit 7 of U video component output. At power up, sampled as B8. * Note: Output UV7 is not available on the OV6120 sensor
20	*UV6/ABKEN	I/O	Bit 6 of U video component output. At power up, sampled as ABKEN. * Note: Output UV6 is not available on the OV6120 sensor
21	*UV5/MIR	I/O	Bit 5 of U video component output. At power up, sampled as MIR. * Note: Output UV5 is not available on the OV6120 sensor
22	*UV4	I/O	Bit 4 of U video component output. * Note: This output (UV4) is not available on the OV6120 sensor

OV6620/OV6120

SINGLE IC CMOS COLOR AND B/W DIGITAL CAMERAS

Table 1. Pin Description

Pin No.	Name	Pin Type	Function/Description
23	*UV3	I/O	Bit 3 of U video component output. * Note: This output (UV3) is not available on the OV6120 sensor
24	*UV2/QCIF	I/O	Bit 2 of U video component output. At power up, sampled as QCIF. * Note: Output UV2 is not available on the OV6120 sensor
25	*UV1/CC656	I/O	Bit 1 of U video component output. At power up, sampled as CC656. * Note: Output UV1 is not available on the OV6120 sensor
26	*UV0/GAMMA	I/O	Bit 0 of U video component output. At power up, sampled as GAMMA. * Note: Output UV0 is not available on the OV6120 sensor
27	XCLK1	I	Crystal clock input
28	XCLK2	O	Crystal clock output
29	DVDD	V _{in}	Digital power supply (+5VDC)
30	DGND	V _{in}	Digital ground
31	DOGND	V _{in}	Digital interface output buffer ground
32	DOVDD	V _{in}	Digital interface output buffer power supply (+5VDC)
33	PCLK/PWDB	I/O	PCLK output. At power up sampled as PWDB.
34	Y7/CS0	I/O	Bit 7 of Y video component output. At power up, sampled as CS0.
35	Y6/CS2	I/O	Bit 6 of Y video component output. At power up, sampled as CS2.
36	Y5/SHARP	I/O	Bit 5 of Y video component. At power up, sampled as SHARP.
37	Y4/CS1	I/O	Bit 4 of Y video component. At power up, sampled as CS1
38	Y3/RGB	I/O	Bit 3 of Y video component output. At power up, sampled as RGB.
39	Y2/G2X	I/O	Bit 2 of Y video component output. At power up, sampled as G2X.
40	Y1	I/O	Bit 1 of Y video component output.
41	Y0/CBAR	I/O	Bit 0 of Y video component output. At power up, sampled as CBAR.
42	CHSYNC/BW	I/O	CHSYNC output. At power up, sampled as BW.
43	DEGND	V _{in}	Decoder ground.
44	DEVDD	V _{in}	Decoder power supply (+5VDC)
45	SIO-1	I	SCCB serial interface clock input
46	SIO-0	I/O	SCCB serial interface data input and output.
47	MULT	Function (Default = 0)	SCCB slave selection "0" - Select single slave ID "1" - Enable multiple (8) slaves
48	SGND	V _{in}	Array ground

OV6620/OV6120

SINGLE IC CMOS COLOR AND B/W DIGITAL CAMERAS

Functional Description

Note: References to color features do not apply to the OV6120 B/W Digital Image Sensor.

1 Overview

Referring to Figure 1, OV6620/OV6120 CMOS Image Sensor Block Diagram below, the OV6620 sensor includes a 356 x 292 resolution image array, an analog signal processor, dual 8-bit Analog-to-Digital converters, analog video multiplexer, digital data formatter and video port, SCCB interface and registers, digital controls including timing block, exposure, and black and white balance.

The OV6620/OV6120 sensor is a 1/4-inch CMOS imaging device. The sensor contains approximately 101,376 pixels. Its design is based on a field integration read-out system with line-by-line transfer and an electronic shutter with a synchronous pixel read out scheme. The color filter of the sensor consists of a primary color RG/GB array arranged in line-alternating fashion.

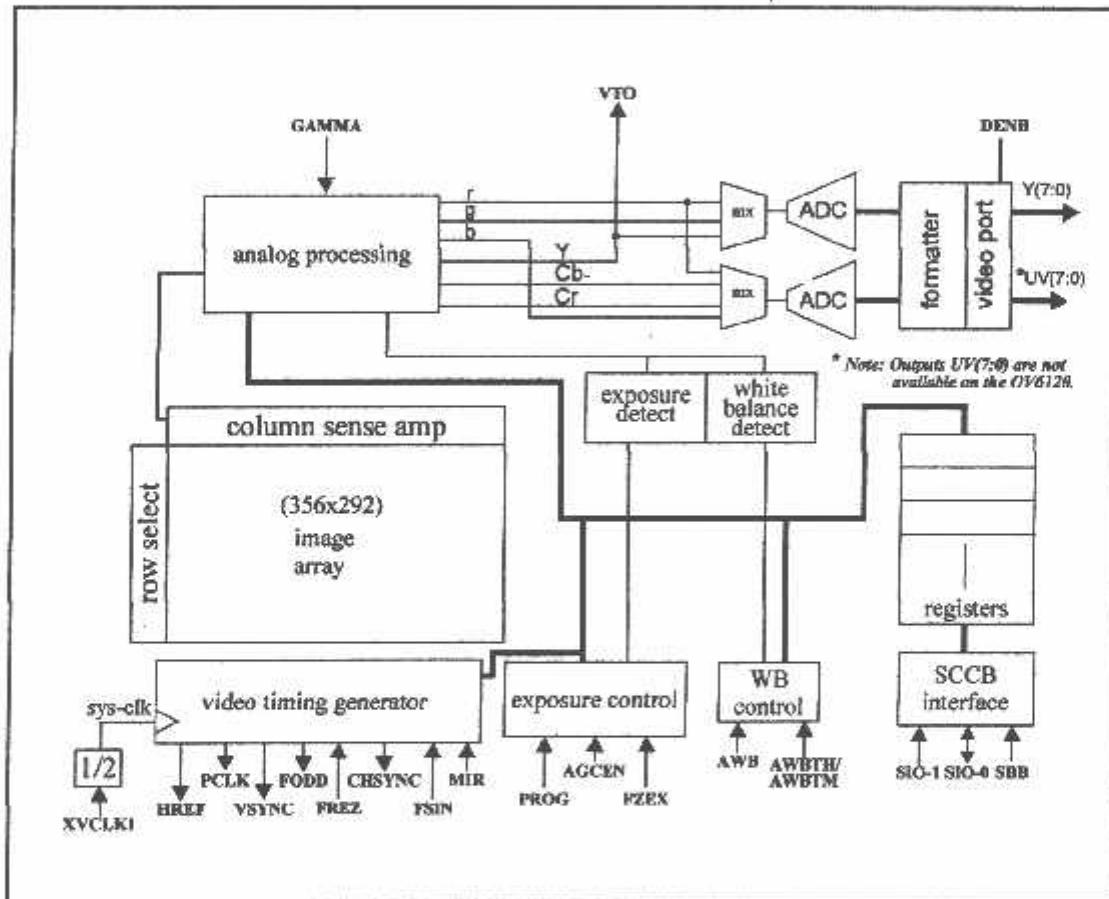


Figure 1. OV6620/OV6120 CMOS Image Sensor Block Diagram

1.2 Analog Processor Circuits

1.2.1 Overview

The image is captured by the 356 x 292 pixel image array and routed to the analog processing section where the majority of signal processing occurs. This block contains the circuitry which performs color separation, matrixing, Automatic Gain Control (AGC), gamma correction, color correction, color balance, black level calibration, "knee" smoothing, aperture correction, and controls for picture luminance, chrominance, and anti-alias filtering. The analog video signals are based on the following formula:

$$\begin{aligned} Y &= 0.59G + 0.31R + 0.11B \\ U &= R - Y \\ V &= B - Y \end{aligned}$$

where R, G, B are the equivalent color components in each pixel.

A YCrCb format is also supported, based on the formula below:

$$\begin{aligned} Y &= 0.59G + 0.31R + 0.11B \\ Cr &= 0.713 \times (R - Y) \\ Cb &= 0.564 \times (B - Y) \end{aligned}$$

The YCrCb/RGB Raw Data signal from the analog processing section is fed to two on-chip 8-bit Analog-to-Digital (A-to-D) converters: one for the Y/RG channel and one shared by the CrCb/BG channels. The A-to-D converted data stream is further conditioned in the digital formatter. The processed signal is delivered to the digital video port through the video multiplexer which routes the user-selected 16-, 8-, or 4-bit video data to the correct output pins.

The on-chip 8-bit A-to-D converters operate at up to 9 MHz, fully synchronous to the pixel rate. Actual conversion rate is set as a function of the frame rate. A-to-D black-level calibration circuitry ensures the following:

- the black level of Y/RGB is normalized to a value of 16
- the peak white level is limited to 240
- CrCb black level is 128
- Peak/Bottom is 240/16
- RGB raw data output range is 16/240

(Note: Values 0 and 255 are reserved for sync flag)

1.2.2 Image Processing

The algorithm used for the electronic exposure control is based on the brightness of the full image. The exposure is optimized for a "normal" scene which assumes the subject is well lit relative to the background. In situations where the image is not well lit, the Automatic Exposure Control (AEC) White/Black ratio may be adjusted to suit the needs of the application.

Additional on-chip functions include Automatic Gain Control (AGC) which provides a gain boost of up to 24dB. White balance control enables setting of proper color temperature and can be programmed for automatic or manual operation. Separate saturation, brightness, contrast, and sharpness adjustments allow for further fine tuning of the picture quality and characteristics. The OV6620 image sensor also provides control over the White Balance ratio for increasing/decreasing the image field Red/Blue component ratio. The sensor provides a default setting which may be sufficient for many applications.

1.2.3 Windowing

The windowing feature of the OV6620/OV6120 image sensors allows user-definable window sizing as required by the application. Window size setting (in pixels) ranges from 2 x 2 to 356 x 292, and can be positioned anywhere inside the 356 x 292 boundary. Note that modifying window size and/or position does not change frame or data rate. The OV6620/OV6120 imager alters the assertion of the HREF signal to be consistent with the programmed horizontal and vertical region. The default output window is 352 x 288.

1.2.4 Zoom Video Port (ZV)

The OV6620/OV6120 image sensor includes a Zoom Video (ZV) function that supports standard ZV Port interface timing. Signals available include VSYNC, CHSYNC, PCLK and 16-bit data bus: Y[7:0] and UV[7:0]. The rising edge of PCLK clocks data into the ZV port. See Figure 2, Zoom Video Port Timing below.

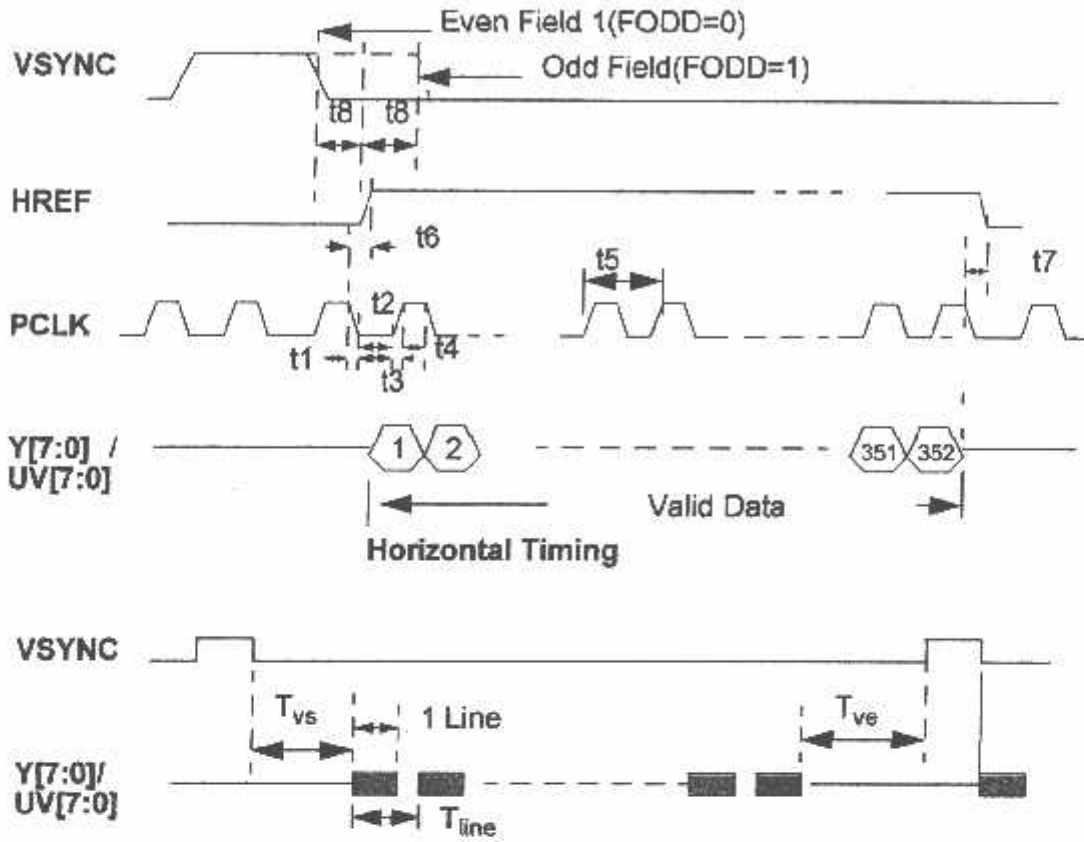


Figure 2. Zoom Video Port Timing

Notes:

- Zoom Video Port format output signal includes:
 - VSYNC: Vertical sync pulse.
 - HREF: Horizontal valid data output window.
 - PCLK: Pixel clock used to clock valid data and CHSYNC into Zoom V Port. Default frequency is 8.86MHz when use 17.73MHz as system clock. Rising edge of PCLK is used to clock the 16 Bit data.
 - Y[7:0]: 8 Bit luminance data bus.
 - UV[7:0]: 8 Bit chrominance data bus.
- All timing parameters are provided in Table 13. Zoom Video Port AC Parameters

OV6620/OV6120

SINGLE IC CMOS COLOR AND BW DIGITAL CAMERAS

1.2.5 QCIF Format

A QCIF mode is available for applications where high resolution image capture is not required. When programmed in this mode, the pixel rate is reduced by one-half. Default resolution is 176 x 144 pixels and can be user-programmed for other resolutions. Refer to Table 7. QCIF Digital Output Format (YUV, beginning of line) and Table 8. QCIF Digital Output Format (RGB Raw Data, Beginning of Line) for further information.

1.2.6 Video Output

The video output port of the OV6620/OV6120 image sensors provides a number of output format/standard options to suit many different application requirements. Table 2. Digital Output Formats, below, indicates the output formats available. These formats are user-programmable through the SCCB interface (See Section 3.1 SCCB Bus Protocol Format).

The OV6620/OV6120 imager supports both CCIR601 and CCIR656 output formats in the following configurations (See Table 3. 4:2:2 16-bit Format for further details):

- 16-bit, 4:2:2 format

(This mode complies with the 60/50 Hz CCIR601 timing standard. See Table 3. 4:2:2 16-bit Format below)

- 8-bit data mode

(In this mode, video information is output in Cb Y Cr Y order using the Y port only and running at twice the pixel rate during which the UV port is inactive. See Table 4. 4:2:2 8-bit Format below)

- 4-bit nibble mode

(In the nibble mode, video output data appears at bits Y4-Y7. The clock rate for the output runs at twice the normal output speed when in B/W mode, and 4 times the normal output speed in when in color mode.)

- 704 x 288 format

(When programmed for this mode, the OV6620/OV6120 pixel clock is doubled and the video output sequence is Y0Y0Y1Y1 ... and U0U0V0V0 ... See Figure 3, Pixel Data Bus (YUV Output), below.)

The OV6620/OV6120 imaging devices provide VSYNC, HREF, PCLK, FODD, CHSYNC as standard output video timing signals.

The OV6620/OV6120 image sensor can also be programmed to provide video output in RGB Raw Data 16-bit/8-bit/4-bit format. The output sequence is matched to the OV6620 Color Filter Pattern (See Section Figure 4. Pixel Data Bus (RGB Output), below):

- Y channel output sequence is G R G R

- UV channel output sequence is B G B G

For 8-bit RGB Raw Data video output appears on the Y channel (with an output sequence of B G R G) and the UV channel is disabled.

In RGB Raw Data CCIR656 modes, the OV6620/OV6120 imager asserts SAV (Start of Active Video) and EAV (End of Active Video) to indicate the beginning and the ending of HREF window. As a result, SAV and EAV change with the active pixel window. The 8-bit RGB raw data is also accessible without SAV and EAV information.

The OV6620/OV6120 imagers offer flexibility in YUV output format. The devices may be programmed as standard YUV 4:2:2. These devices may be configured to "swap" the U V sequence. When swapped, the UV channel output format for 16-bit configurations becomes:

- V U V U ...etc.

and for 8-bit configurations becomes:

- V Y U Y ...etc.

A third format is available for the 8-bit configurations and OV6620/OV6120 enables the Y/UV sequence swap:

- Y U Y V ...etc.

The OV6620 color single-IC camera can be configured for use as a black and white imaging device. In this mode, vertical resolution is greater than in color. Video data output is provided at the Y port (pins 34:41) and the UV port is tri-stated. The data (Y/RGB) output rate is equivalent to operating in 16-bit mode.

The Y/UV or RGB output byte MSB and LSB can be reverse-ordered on the OV6620/OV6120 device. The Y7 - Y0 default sequence sets Y7 as MSB and Y0 as LSB. Programming a reverse order configuration sets Y7 as LSB and Y0 is MSB, with bits Y2-Y6 reversed-ordered appropriately.

Table 2. Digital Output Formats

Resolution	Pixel Clock	352 x 288	704 x 288	176 x 144
YUV 4:2:2	16-bit	Y	Y	Y
	8-bit	Y	Y	Y
	CCIR656	Y	Y	Y
	Nibble	Y	Y	Y
RGB	16-bit	Y	Y	Y
	8-bit	Y	Y	Y
	CCIR656 ¹	Y	Y	Y
	Nibble	Y	Y	Y
Y/UV swap ²	16-bit			
	8-bit	Y	Y	Y
U/V swap	YUV ³	Y	Y	Y
	RGB ⁴	Y	Y	Y
YG	16-bit	Y	Y	Y
	8-bit			
One Line	16-bit	Y		
	8-bit			
MSB/LSB swap ⁵		Y	Y	Y

Notes:

(*Y* indicates mode/combination is supported by OV6620/OV6120.)

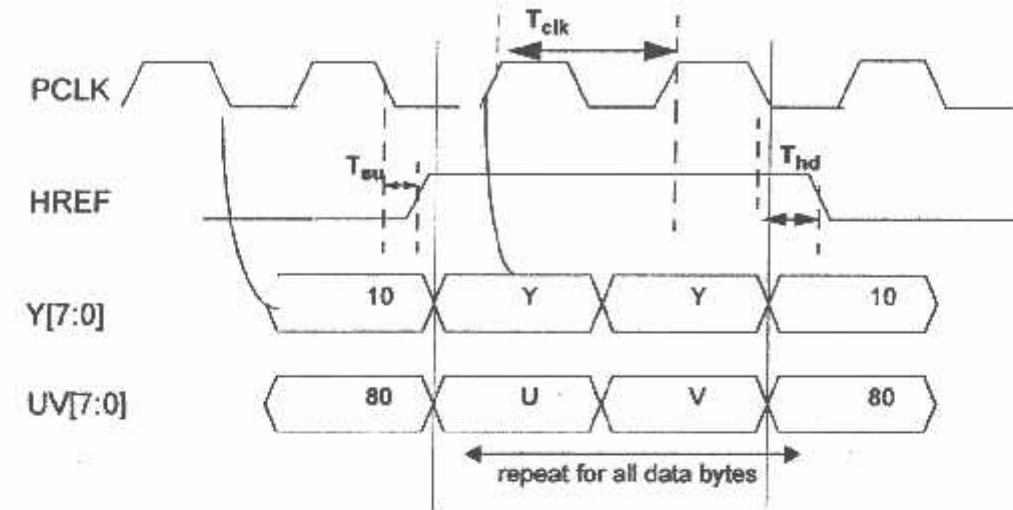
1. When in RGB CCIR656 format, output is 8 bits. SAV and EAV are inserted at the beginning and ending of HREF, which synchronize the acquisition of Vsync and Hsync. In this format, an 8-bit data bus configuration (without VSYNC and CHSYNC) may be used.
2. Y/UV swap is valid only in 8-bit mode. Y channel output sequence is Y U Y V...
3. In YUV format, U/V swap means UV channel output sequence swap. V U V U... for 16 bit; V Y U Y ... for 8-bit.
4. In RGB format, U/V swap means neighbor row B R output sequence swap. Refer to RGB raw data output format for further details

Table 3. 4:2:2 16-bit Format

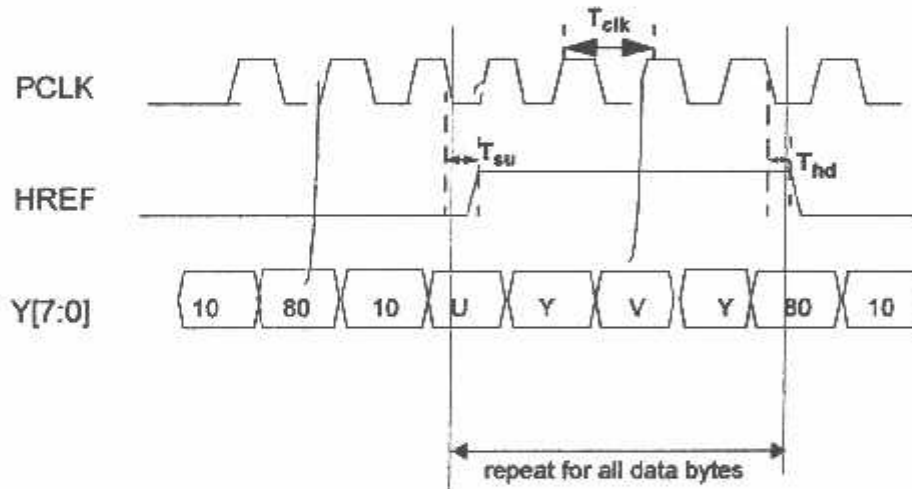
Data Bus	Pixel Byte Sequence					
Y7	Y7	Y7	Y7	Y7	Y7	Y7
Y6	Y6	Y6	Y6	Y6	Y6	Y6
Y5	Y5	Y5	Y5	Y5	Y5	Y5
Y4	Y4	Y4	Y4	Y4	Y4	Y4
Y3	Y3	Y3	Y3	Y3	Y3	Y3
Y2	Y2	Y2	Y2	Y2	Y2	Y2
Y1	Y1	Y1	Y1	Y1	Y1	Y1
Y0	Y0	Y0	Y0	Y0	Y0	Y0
UV7	U7	V7	U7	V7	U7	V7
UV6	U6	V6	U6	V6	U6	V6
UV5	U5	V5	U5	V5	U5	V5
UV4	U4	V4	U4	V4	U4	V4
UV3	U3	V3	U3	V3	U3	V3
UV2	U2	V2	U2	V2	U2	V2
UV1	U1	V1	U1	V1	U1	V1
UV0	U0	V0	U0	V0	U0	V0
Y FRAME	0	1	2	3	4	5
UV FRAME	0		2		4	

Table 4. 4:2:2 8-bit Format

Data Bus	Pixel Byte Sequence							
Y7	U7	Y7	V7	Y7	U7	Y7	V7	Y7
Y6	U6	Y6	V6	Y6	U6	Y6	V6	Y6
Y5	U5	Y5	V5	Y5	U5	Y5	V5	Y5
Y4	U4	Y4	V4	Y4	U4	Y4	V4	Y4
Y3	U3	Y3	V3	Y3	U3	Y3	V3	Y3
Y2	U2	Y2	V2	Y2	U2	Y2	V2	Y2
Y1	U1	Y1	V1	Y1	U1	Y1	V1	Y1
Y0	U0	Y0	V0	Y0	U0	Y0	V0	Y0
Y FRAME	0		1		2		3	
UV FRAME	0 1				2 3			



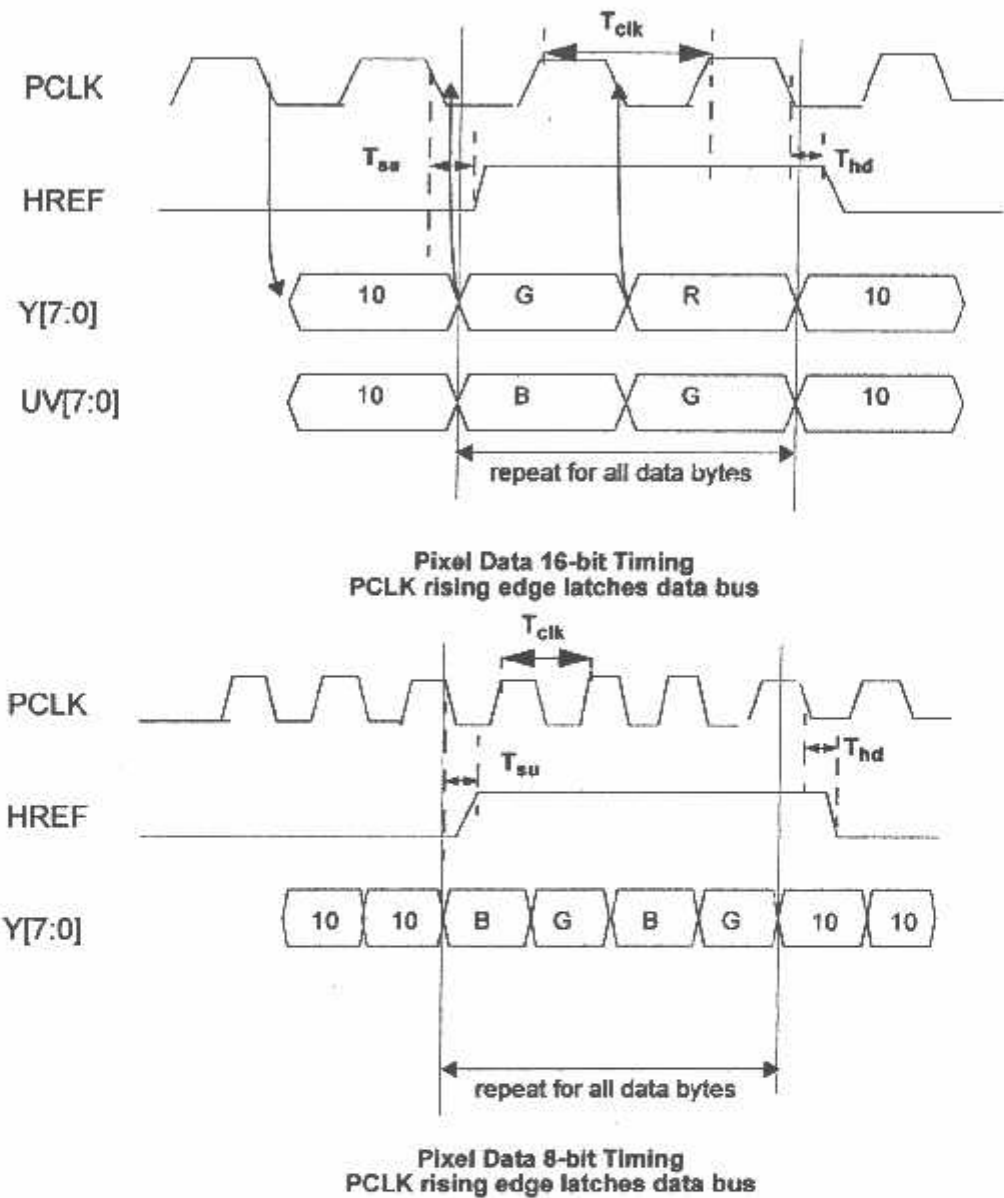
Pixel Data 16-bit Timing
(PCLK rising edge latches data bus)



Pixel Data 8-bit Timing
(PCLK rising edge latches data bus)

Note: T_{clk} is pixel clock period. When the OV6620 system clock is 17.73 MHz, $T_{clk} = 112$ ns for 16-bit output; $T_{clk} = 56$ ns for 8-bit output. T_{su} is HREF set-up time, maximum is 15 ns; T_{hd} is HREF hold time, maximum is 15 ns.

Figure 3. Pixel Data Bus (YUV Output)



Note: T_{clk} is pixel clock period. When the OV6620 system clock is 17.73MHz, T_{clk} = 112ns for 16-bit output T_{clk} = 56ns for 8-bit output T_{su} is HREF set-up time, maximum is 15 ns; T_{hd} is HREF hold time, maximum is 15 ns.

Figure 4. Pixel Data Bus (RGB Output)

OV6620/OV6120

SINGLE IC CMOS COLOR AND B/W DIGITAL CAMERAS

MSB/LSB swap means: Default Y/UV channel output port relationship is:

Table 5. Default Output Sequence

	MSB							LSB
Output Port	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
Internal Output data	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

If the device is programmed for data swap, the sequence is changed to:

Table 6. Swapped MSB/LSB Output Sequence

	MSB							LSB
Output Port	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
Internal Output data	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7

Table 7. QCIF Digital Output Format (YUV, beginning of line)

Pixel #	1	2	3	4	5	6	7	8
Y	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
UV	U0,V0	U1,V1	U2,V2	U3,V3	U4,V4	U5,V5	U6,V6	U7,V7

Y channel output Y2 Y3 Y6 Y7 Y10 Y11 ...

– UV channel output U2 V3 U6 V7 U10 V11 ...

– Every line output data number is half(176 pixels) and only one half line data (every other line, total 144 line) in one frame will be output.

Table 8. QCIF Digital Output Format (RGB Raw Data, Beginning of Line)

Pixel #	1	2	3	4	5	6	7	8
Line 1	B0	G1	B2	G3	B4	G5	B6	G7
Line 2	G0	R1	G2	R3	G4	R5	G6	R7

1. Default RGB two line output mode:

– Y channel output G0 R1 G4 R5 G8 R9 ...

– UV channel output B0 G1 B4 G5 B8 G9 ...

– Every line output half data(176 pixels) and all line(144 line) data in one frame will be output.

3. QCIF Resolution Digital Output Format

– Y channel output Y2 Y3 Y6 Y7 Y10 Y11 ...

– UV channel output U2 V3 U6 V7 U10 V11 ...

– Every line output data number is half (176 pixels) and only one half line data (every other line, total 144 line) in one frame will be output.

YG two line output mode:

– Y channel output G0 R1 G4 R5 G8 R9 ...

– UV channel output B0 G1 B4 G5 B8 G9 ...

– Every line outputs half data (176 pixels) and all line (144 line) data in one frame will be output.

OV6620/OV8120

SINGLE IC CMOS COLOR AND BW DIGITAL CAMERAS

Table 9. RGB Raw Data Format

R/C	1	2	3	4	.	353	354	355	356
1	B ₁₁	G ₁₂	B ₁₃	G ₁₄		B	G	B	G
2	G ₂₁	R ₂₂	G ₂₃	R ₂₄		G	R	G	R
3	B ₃₁	G ₃₂	B ₃₃	G ₃₄		B	G	B	G
4	G ₄₁	R ₄₂	G ₄₃	R ₄₄		G	R	G	R
5	B ₅₁	G ₅₂	B ₅₃	G ₅₄		B	G	B	G
.									
289	B	G	B	G		B	G	B	G
290	G	R	G	R		G	R	G	R
291	B	G	B	G		B	G	B	G
292	G	R	G	R		G	R	G	R

Notes:

A. Y port output data sequence: G R G R G R ... or G G G G ...; UV port output data sequence: B G B G B G ... or B R B R ...;
Array Color Filter Pattern is Bayer-Pattern

B. Output Modes

16-bit Format (HREF total 292)

Default mode:

- 1st HREF Y channel output unstable data, UV output B₁₁ G₁₂ B₁₃ G₁₄ ...
- 2nd HREF Y channel output G₂₁ R₂₂ G₂₃ R₂₄ ..., UV output B₁₁ G₁₂ B₁₃ G₁₄ ...
- 3rd HREF Y channel output G₂₁ R₂₂ G₂₃ R₂₄ ..., UV output B₃₁ G₃₂ B₃₃ G₃₄ ...
- Every line of data is output twice.

YG mode:

- 1st HREF Y and UV output unstable data.
- 2nd HREF Y channel output G₂₁ G₁₂ G₂₃ G₁₄ ..., UV output B₁₁ R₂₂ B₁₃ R₂₄ ...
- 3rd HREF Y is G₂₁ G₃₂ G₂₃ G₃₄ ..., UV channel is B₃₁ R₂₂ B₃₃ R₂₄ ...
- Every line data output twice.

One line mode:

- 1st HREF Y channel output B₁₁ G₁₂ B₁₃ G₁₄ ...
- 2nd HREF Y channel output G₂₁ R₂₂ G₂₃ R₂₄ ...
- UV channel tri-state.

8-bit Format (HREF total 292)

- 1st HREF Y channel output unstable data.
- 2nd HREF Y channel output B₁₁ G₂₁ R₂₂ G₁₂ ...
- 3rd HREF Y channel output B₃₁ G₂₁ R₂₂ G₃₂ ..., etc.
- PCLK timing is double and PCLK rising edge latch data bus. UV channel tri-state. Every line data output twice.

4-bit Nibble Mode Output Format

- Uses higher 4 bits of Y port (Y[7:4]) as output port.
- Supports YCrCb/RGB data, CCIR601/CCIR656 timing, Color/B&W.
- Output sequence: High order 4 bits followed by lower order 4 bits
Y0h Y0l Y1h Y1l ...
U0h U0l V0h V0l ...

For B&W or one-line RGB raw data, the output data clock speed is doubled. For color YUV, output clock is four times that of the 16-bit output data. In color mode, sensor must be set to 8-bit mode, and the nibble timing, clock divided by 2.

- Output sequence: U0h U0l Y0h Y0l V0h V0l Y1h Y1l ...

2.7 Slave Mode Operation

The OV6620/OV6120 sensors can be programmable to operate in slave mode configuration (COMI[6] = 1, default is master mode). HSYNC and VSYNC output signals are provided.

When used as a slave device, the external master must provide the OV6620/OV6120 imager with the following:

1. System clock CLK to XCLK1 pin;
2. Horizontal sync, Hsync, to CHSYNC pin, positive assertion;
3. Vertical frame sync, Vsync, to VSYNC pin, positive assertion

When in slave mode, the OV6620/OV6120 tri-states HSYNC (pin 42) and VSYNC (pin 16) output pins, which may then be used as input pins. To synchronize multiple devices, the OV6620/OV6120 image sensors use external system clock, CLK, to synchronize external horizontal sync, HSYNC, which is then used to synchronize external vertical frame sync, Vsync. See Figure 5, Slave Mode External Sync Timing for timing considerations.

2.8 Frame Exposure Mode

The OV6620/OV6120 sensors support frame exposure mode when programmed for Progressive Scan. FREX (pin 4) is asserted by an external master device to set exposure time. When FREX = 1, the OV6620/OV6120 pixel array will be quickly precharged. Based on the external master's assertion of FREX, the OV6620/OV6120 devices capture the image. When the master de-asserts FREX (FREX = 0), the video output data stream is delivered to the OV6620/OV6120 output port in a line-by-line manner.

It should be noted that FREX must remain active long enough to ensure the complete image array has been recharged.

When data is being output from the OV6620/OV6120 image sensor, care must be taken so as not to expose the image array to light. This may affect the integrity of the image data captured. A mechanical shutter synchronized with the frame exposure rate can be used to minimize this situation. Frame exposure mode timing is shown in Section Figure 6. Frame Exposure Timing below.

2.9 Reset

The OV6620/OV6120 image sensors include a RESET pin (pin 2) which forces a complete hardware reset when

pulled high (Vcc). When a hardware reset occurs, the OV6620/OV6120 sensor clears all registers or sets them to their default values. Reset may also be initiated through the SCCB interface.

1.2.10 Power Down Mode

Two methods are available for placing the OV6620/OV6120 devices into power-down mode: hardware power down and SCCB/software power down.

To initiate hardware power down the PWDN pin (pin 9) must be tied to high (+5VDC). When this occurs, the OV6620/OV6120 internal device clock is halted and all internal registers (except SCCB registers) are reset. In this mode, current draw is less than 10uA.

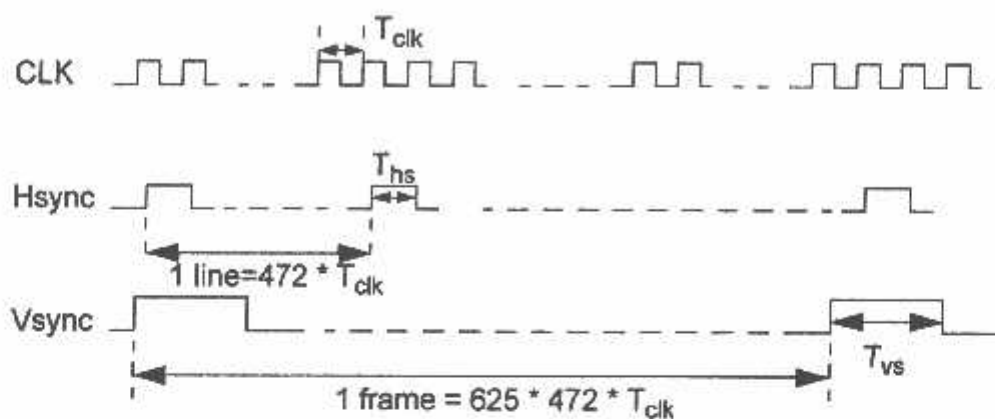
Executing a software power down through the SCCB interface suspends internal circuit activity, but does halt the device clock. In this mode, current requirements drop to less than 1mA.

1.2.11 Configuring the OV6620/OV6120 Image Sensors

Two methods are provided for configuring the OV6620/OV6120 ICs for specific application requirements.

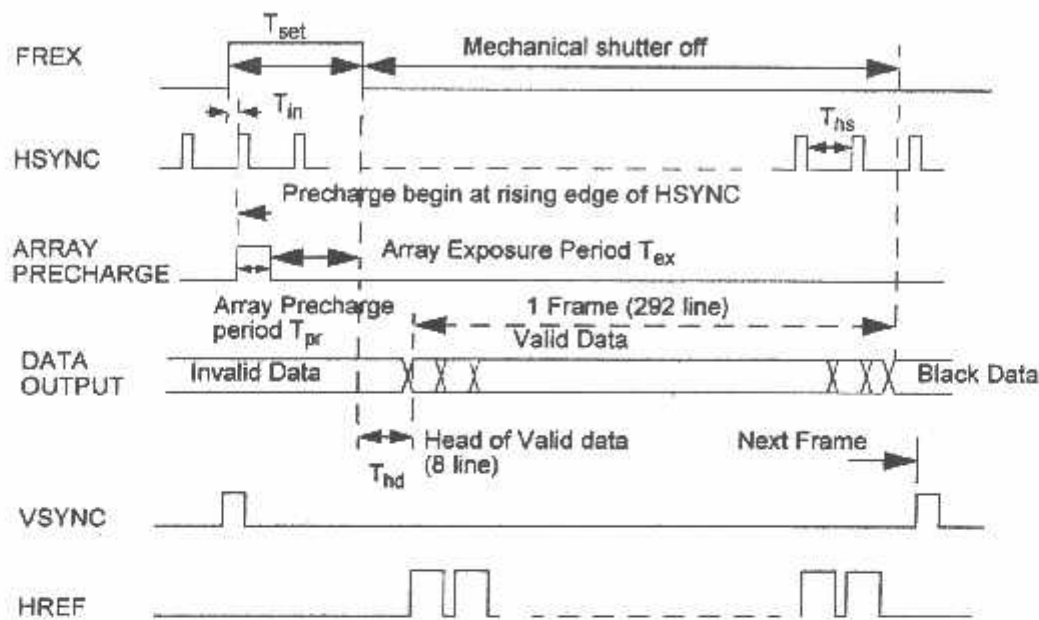
At power up, the OV6620/OV6120 sensors read the status of certain pins to determine what, if any, power up default settings are requested. Once the reading of the external pins is completed, the device configures its internal registers according to the specified pins. Not all device functions are available for configuration through external pin.

A more flexible and comprehensive method for configuring the OV6620/OV6120 ICs is to use its on-chip SCCB register programming capability. The SCCB interface provides access to all of the device's programmable internal registers. See Section 3.1 SCCB Bus for further details about using the SCCB interface on the OV6620/OV6120 camera device.

**Notes:**

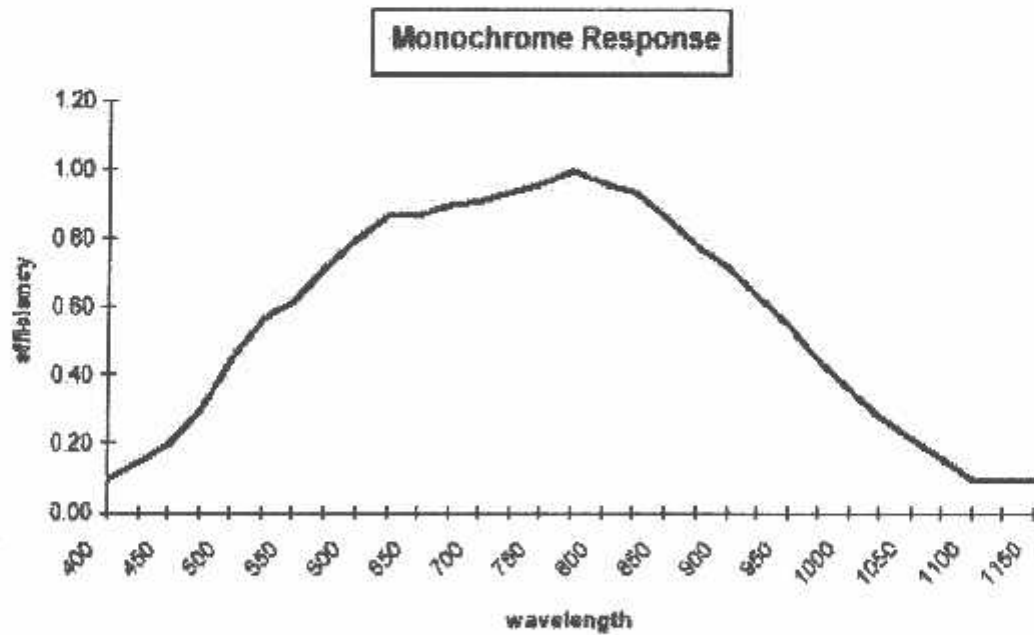
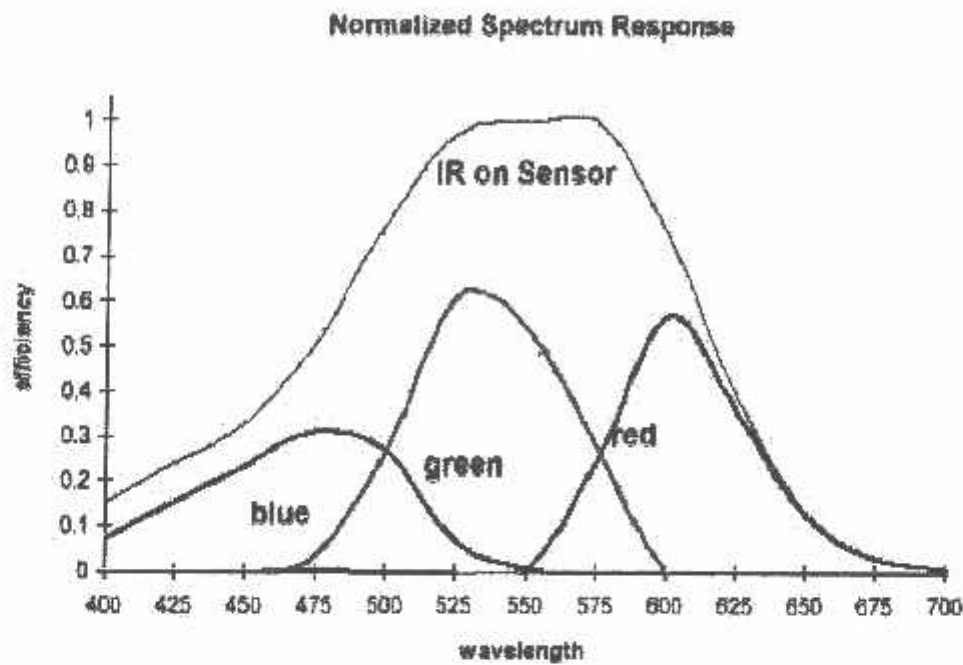
1. $T_{hs} > 6 * T_{clk}$ (2) $T_{hs} < T_{vs} < 472 * T_{clk}$
2. Hsync period is $472 * CLK$
3. Vsync period is $625 * 472 * CLK$
4. OV6620 will be stable after 1 frame. (2nd Vsync).

Figure 5. Slave Mode External Sync Timing



Note: $T_{pr} = 292 \times 4 \times T_{clk}$, T_{clk} is internal pixel period. For default 17.73 MHz, $T_{clk} = 112 \mu s$. If CLK[5:0] set to divided number, T_{clk} will increase accordingly.
 T_{ex} is array exposure time which is decided by external master device.
 T_{ln} is uncertain time due to using HSYNC rising edge synchronize FREX, $T_{ln} < T_{ht}$.
After FREX=0, there are 8 line data output before valid data output. $T_{hd} = 4 \times T_{ht}$. Valid data is output when HRE
 $T_{set} = T_{ln} + T_{pr} + T_{ex}$, $T_{set} > T_{pr} + T_{ln}$. Because T_{ln} is uncertain, so exposure time setting resolution is T_{ht} (one line).

Figure 6. Frame Exposure Timing



Electrical Characteristics

Table 10. DC Characteristics ($0^{\circ}\text{C} \leq T_A \leq 85^{\circ}\text{C}$, Voltages referenced to GND)

Symbol	Descriptions	Max	Typ	Min	Units
Supply					
V_{DD1}	Supply voltage- internal analog (DEVDD,ADVDD,AVDD,SVDD,AOVDD,DVDD)	5.25	5.0	4.75	V
V_{DD2}	Supply voltage - Internal digital & output digital. (DOVDD)	5.5 3.6	5.0 3.3	4.5 3.0	V V
I_{DD1}	Supply Current (@ 50Hz frame rate & 5 volt digital I/O, 25pf + 1TTL load on 16 bit data bus)	40	-	-	mA
I_{DD2}	Standby supply current	10	5	-	uA
Digital Inputs					
V_{IL}	input voltage LOW	0.8	-	-	V
V_{IH}	input voltage HIGH	-	-	2.0	V
C_{in}	input capacitor	10	-	-	pF
Digital Outputs - standard load 25pf, 1.2kΩ to 3.0volts					
V_{OH}	output voltage HIGH	-	-	2.4	V
V_{OL}	output voltage LOW	0.6	-	-	V
SCCB Inputs -					
V_{IL}	SIO-0 and SIO-1 ($V_{DD2}=5V$)	1.5	0	-0.5	V
V_{IH}	SIO-0 and SIO-1 ($V_{DD2}=5V$)	$V_{dd} + .5$	5	3.0	V
V_{IL}	SIO-0 and SIO-1 ($V_{DD2}=3V$)	1	0	-0.5	V
V_{IH}	SIO-0 and SIO-1 ($V_{DD2}=3V$)	3.5	3	2.5	V

Table 11. AC Characteristics (T_A=25°C; V_{dd}=5V)

Symbol	Descriptions	Max	Typ	Min	Units
RGB/YCrCb output					
I _{so}	maximum sourcing current		15		mA
V _y	DC level at zero signal		1.2		V
	Y peak-peak 100% amplitude (without sync)		1		V
	sync amplitude		0.4		V
ADC parameters					
B	analog bandwidth				MHz
Φ _{diff}					
DLE	DC differential linearity error		0.5		LSB
ILE	DC integral linearity error		1		LSB

Table 12. Timing Characteristics

Symbol	Descriptions	Max	Typ	Min	Units
Oscillator & Clock In					
f _{osc}	frequency (XCLK1,XCLK2)	30	17.734	10	MHz
t _r , t _f	clock input rise/fall time	5			ns
	clock input duty cycle	55	50	45	%
SCCB timing(400kbit/s)					
t _{BUF}	Bus free time between STOP & START	-	-	1.3	ms
t _{HD:SAT}	SIO-1 change after START status	-	-	0.6	μs
t _{LOW}	SIO-1 low period	-	-	1.3	μs
t _{HIGH}	SIO-1 high period	-	-	0.6	μs
t _{HD:DAT}	Data hold time	-	-	0	μs
t _{SU:DAT}	Data set-up time	-	-	0.1	μs
t _{SU:STP}	Set-up time for STOP status	-	-	0.6	μs
Digital timing					
t _{pdk}	PCLK cycle time	-	-	112	ns
	16 bit operation	-	-	56	ns
	8 bit operation	-	-		ns

Table 12. Timing Characteristics

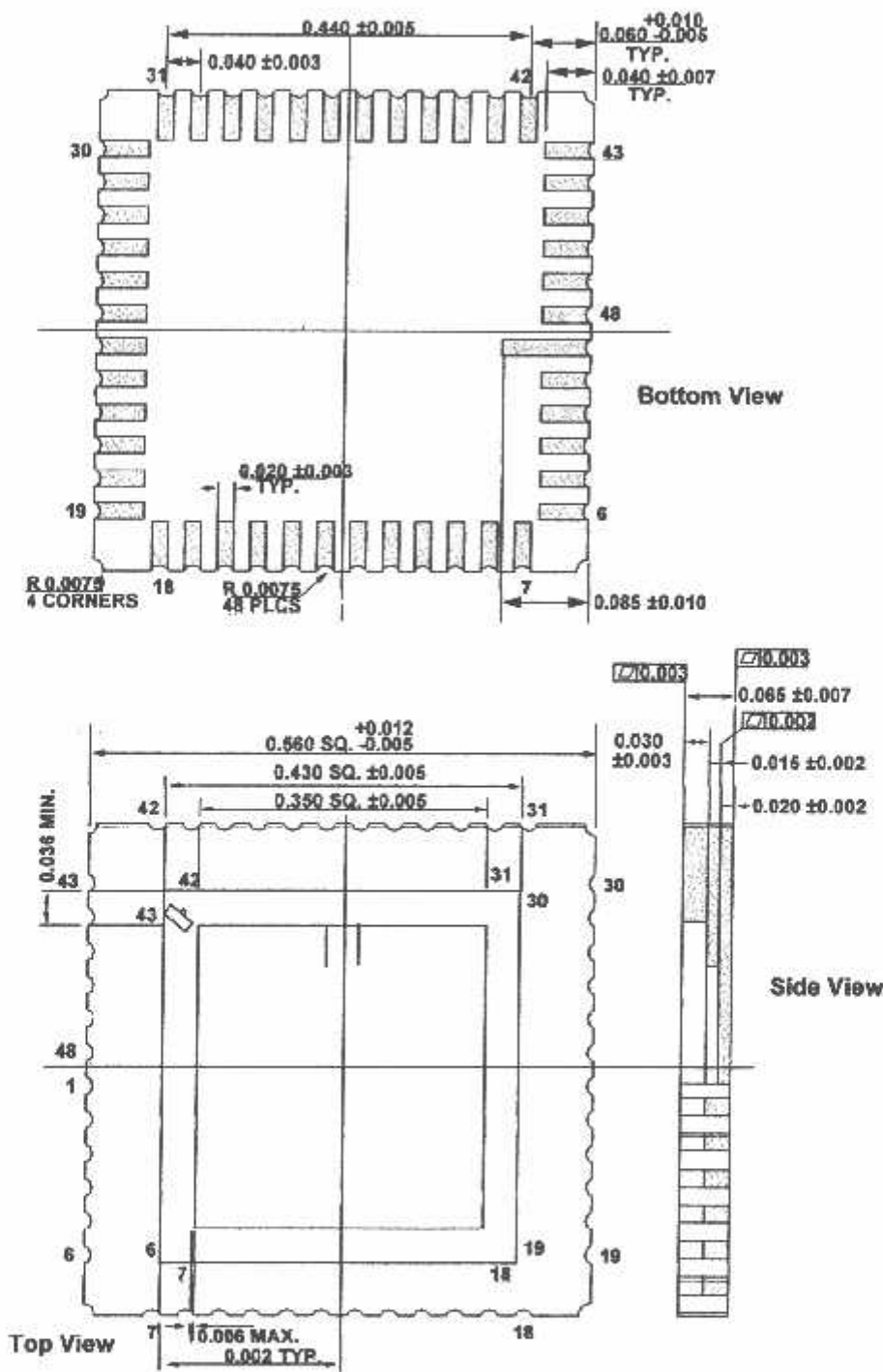
Symbol	Descriptions	Max	Typ	Min	Units
t_r, t_f	PCLK rise/fall time	15	-	-	ns
t_{pdd}	PCLK to data valid	15	-	-	ns
$t_{p:d}$	PCLK to HREF delay	20	10	5	ns

Table 13. Zoom Video Port AC Parameters

Symbol	Parameter	Min.	Max.
t1	PCLK fall timing	4 ns	8 ns
t2	PCLK low time	50 ns	
t3	PCLK rise time	4 ns	8 ns
t4	PCLK high time	50 ns	
t5	PCLK period	106 ns	
t6	Y/UV/HREF setup time	10 ns	
t7	Y/UV/HREF hold time	20 ns	
t8	VSYNC setup/hold time to HREF	1 us	

Notes:

1. In Interlaced Mode, there are Even/Odd field different (t8). When In Progressive Scan Mode, only frame timing same as Even field(t8).
2. After VSYNC falling edge, OV6620 will output black reference level, the line number is T_{vb} , which is the line number between the 1st HREF rising edge after VSYNC falling edge and 1st valid data CHSYNC rising edge. Then valid data, then black reference, line number is T_{vb} , which is the line number between last valid data CHSYNC rising edge and 1st CHSYNC rising edge after VSYNC rising edge. The black reference output line number is dependent on vertical window setting.
3. When in default setting, $T_{vb} = 14 * T_{line}$, which is changed with register VS[7:0]. VS[7:0] step equal to 1 line.
4. When in default setting, $T_{vb} = 4 * T_{line}$ for Odd Field, $T_{vb} = 3 * T_{line}$ for Even Field, which is changed with register VE[7:0]. VE[7:0] step equal to 1 line.



OV6620/OV6120

SINGLE IC CMOS COLOR AND B/W DIGITAL CAMERAS

3. SCCB Bus

Many of the functions and configuration registers in the OV6620/OV6120 image sensors are available through the SCCB interface. The SCCB port is enabled by asserting the SBB line (pin 12) through a 10K ohm

resistor to V_{DD} . When the SCCB capability is enabled (SBB = 1), the OV6620/OV6120 imager operates as a slave device that supports up to 400 kbps serial transfer rate using a 7-bit address/data transfer protocol. Therefore, the OV6620/OV6120 sensor takes the read

subaddress from the previous write cycle. In multi-byte write or multi-byte read cycles, the subaddress is automatically increment after the first data byte so that continuous locations can be accessed in one bus cycle. A multi-byte cycle overwrites its original subaddress; therefore, if a read cycle immediately follows a multi-byte cycle, you must insert a single byte write cycle that provides a new subaddress.

3.1 SCCB Bus Protocol Format

Within each byte, MSB is always transferred first. Read/write control bit is the LSB of the first byte.

The protocol that SIO-0 remain stable during the HIGH period of the SIO-1. Each bit is allowed to change state only when SIO-1 is LOW. The OV6620/OV6120 SCCB supports multi-byte write and multi-byte read. The master must supply the subaddress. in the write cycle, but not in the read cycle.

Therefore, the OV6620/OV6120 sensor takes the read subaddress from the previous write cycle. In multi-byte write or multi-byte read cycles, the subaddress is automatically increment after the first data byte so that continuous locations can be accessed in one bus cycle. A multi-byte cycle overwrites its original subaddress; therefore, if a read cycle immediately follows a multi-byte cycle, you must insert a single byte write cycle that provides a new subaddress.

The OV6620/OV6120 imager can be programmed to one-of-eight slave ID addresses. Function pins CS[2:0] pins 35, 37, 34, respectively).

Table 15. Slave ID Addresses

CS[2:0]	000	001	010	011	100	101	110	111
WRITE ID (hex)	C0	C4	C8	CC	D0	D4	D8	DC
READ ID (hex)	C1	C5	C9	CD	D1	D5	D9	DD

The OV6620/OV6120 sensors support both single chip and multiple chip configurations. By asserting MULT (pin 47) high, the sensor can be programmed for up to 8 slave ID addresses. Asserting MULT low configures the OV6620/OV6120 imagers for single ID slave address with address C0 for writes and address C1 for reads. MULT is internally defaulted to a low condition.

In the write cycle, the second byte in SCCB bus is the subaddress for selecting the individual on-chip registers, and the third byte is the data associated with this register. Writing to unimplemented subaddress is ignored. In the read cycle, the second byte is the data associated with the previous stored subaddress. Reading of unimplemented subaddress returns unknown.

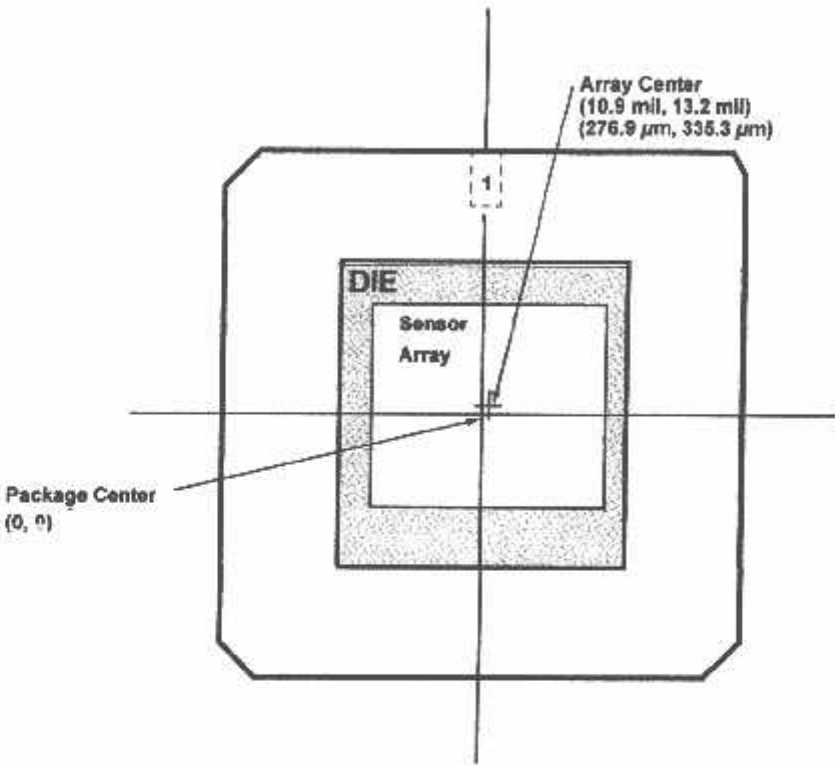


Figure 7. OV6620/OV6120 Package Outline

ble 14. Ordering Information

Part Number	Description	Package
V6620	COLOR Image Sensor, CIF, Digital, SCCB Bus Control	48 pin LCC
V6120	B/W Image Sensor, CIF, Digital, SCCB Bus Control	48 pin LCC

niVision Technologies, Inc. reserves the right to make changes without further notice to any product herein to improve reliability, function, or design. OmniVision Technologies, Inc. does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. No part of this publication may be copied or reproduced, in any form, without the prior written consent of OmniVision Technologies, Inc.

V6620/OV6120

SINGLE IC CMOS COLOR AND B/W DIGITAL CAMERAS

2 Register Set

The table below provides a list and description of available SCCB registers contained in the OV6620/OV6120 image sensor.

Table 16. SCCB Registers

Subaddress (hex)	Register	Default (hex)	Read/Write	Descriptions
0	Gain[6:0]	00	RW	AGC Gain Control GC[7:8] - unimplemented bit, returns 'X' when read. GC[5:0] - Storage for the current AGC Gain setting. This register is updated automatically. If AGC is enabled, the internal control stores the optimal gain value in this register. If AGC is not enabled, a "00" is stored in this register.
1	Blue[7:0]	80	RW	Blue Gain Control BLU[7] - '0' decrease gain, '1' increase gain. BLU[6:0] - blue channel gain balance value. Note: This function is not available on the OV6120 Image Sensor.
2	Red[7:0]	80	RW	Red Gain Control RED[7] - '0' decrease gain, '1' increase gain. RED[6:0] - red channel balance value. Note: This function is not available on the OV6120 Image Sensor.

Subaddress (hex)	Register	Default (hex)	Read/Write	Descriptions
3	Sat	80	RW	Saturation Control SAT[7:0] - saturation adjustment. "FFh"- highest, "00h"-lowest Note: This function is not available on the OV6120 Image Sensor.
4	Rsvd04	XX	-	reserved
5	Cnt	48	RW	Contrast Control CTR[7:0] - contrast adjustment. "FFh"-highest, "00h"-lowest
6	Bri	80	RW	Brightness Control BRI[7:0] - brightness adjustment. "FFh"-highest, "00h"-lowest
7	Sharpness	06	RW	Sharpness Control SHP[7:4] - Threshold of sharpness. Range: 0~80mV, Step: 5 mV SHP[3:0] - Sharpness control. Range: 0 ~ 8x, Step: 0.5x
8	Rsvd08	XX	-	reserved
9	Rsvd09	XX	-	reserved
A	Rsvd0A	XX	-	reserved
B	Rsvd0B	XX	-	reserved
C	AWB - Blue	20	R/W	White Balance Background: Blue Channel ABLU[7:6] - rsvd ABLU[5] - Sign bit. "0" - decrease background blue component "1" - increase background blue component ABLU[4:0] - White balance blue ratio adjustment. Note: This function is not available on the OV6120 Image Sensor.

OV6620/OV6120

SINGLE IC CMOS COLOR AND B/W DIGITAL CAMERAS

Subad- dress (hex)	Register	Default (hex)	Read/ Write	Descriptions
0D	AWB - Red	26	R/W	White Balance Background: Red Channel ARED[7:6] - rsvd ARED[5] - Sign bit "0" - decrease background red component "1" - increase background red component ABLL[4:0] - White balance red ratio adjustment Note: This function is not available on the OV6120 Image Sensor.
0E	COMR	0D	RW	Common Control R COMR[7] - Analog signal 2x gain control bit. "1" - Additional 2x gain, "0" - normal COMR[6:0] - Reserved

Subad- dress (hex)	Register	Default (hex)	Read/ Write	Descriptions
0F	COMS	05	RW	Common Control S COMS[7:6] - Reserved COMS[5:4] - Black expanding level "00" - 1.2V, "01" - 1.26V, "10" - 1.3V, "11" - 1.4V COMS[3:2] - Set high threshold level "00" - 1.9V, "01" - 2.0V, "10" - 2.1V, "11" - 2.2V COMS[1:0] - Set low threshold level "00" - 1.3V, "01" - 1.45V, "10" - 1.5V, "11" - 1.6V

Subad- dress (hex)	Register	Default (hex)	Read/ Write	Descriptions
10	AEC	9A	R	Automatic Exposure Control AEC[7:0] - Set exposure time Interlaced: $T_{ax} = T_{line} \times AEC[7:0]$ Progressive: $T_{ax} = T_{line} \times AEC[7:0] \times 2$
11	CLKRC	00	R	Clock Rate Control CLKRC[7:5] - Sync output polarity selection "00" - HSYNC=Neg, CHSYNC=Neg, VSYNC=Pos "01" - HSYNC=Neg, CHSYNC=Neg, VSYNC=Neg "10" - HSYNC=Pos, CHSYNC=Neg, VSYNC=Pos "11" - HSYNC=Pos, CHSYNC=Pos, VSYNC=Pos CLKRC[5:0] - Clock prescaler $CLK = (CLK_main / ((CLKRC[5:0] + 1) \times 2)) / 2$
12	COMA	24	RW	Common Control A COMA[7] - SRST, "1" initiates soft reset. Initiate soft reset. All registers are set to default values and chip is reset to known state and resumes normal operation. This bit is automatically cleared after reset. COMA[6] - MIRR, "1" selects mirror image COMA[5] - VSFR, "1" enables AGC COMA[4] - Digital output format, "1" selects 8-bit: YUYVYUYV COMA[3] - Select video data output: "1" - select RGB, "0" - select YCrCb COMA[2] - Auto White Balance "1" - Enable AWB, "0" - Disable AWB COMA[1] - Color Bar Test Pattern: "1" - Enable color bar test pattern COMA[0] - reserved Note: COMA[3] is not programmable on the OV6120 Image Sensor.

/6620/OV6120

SINGLE IC CMOS COLOR AND B/W DIGITAL CAMERAS

Address (hex)	Register	Default (hex)	Read/Write	Descriptions
	COMB	01	RW	Common Control B COMB[7] - reserved COMB[6] - reserved COMB[5] - Select data format, "1" - Select 8-bit format, Y/CrCb and RGB is multiplexed to 8-bit Y bus, UV bus is tri-stated, "0" - Select 16-bit format COMB[4] - "1" - enable digital output in CCIR656 format COMB[3] - CHSYNC output: "1" - Horizontal sync, "0" - composite sync COMB[2] - "1" - Tri-state Y and UV busses, "0" - enable both busses COMB[1] - "1" - Initiate single frame transfer COMB[0] - "1" - Enable auto adjust mode Note: COMB[5] is not programmable on the OV6120 Image Sensor.
	COMC	00	RW	Common Control C COMC[7] - reserved COMC[6] - reserved COMC[5] - QCIF digital output format selection: 1 - 176x144; 0 - 352x288. COMC[4] - Field/Frame vertical sync output in VSYNC port selection: 1 - frame sync, only ODD field vertical sync; 0 - field vertical sync, effect in interlaced mode COMC[3] - HREF polarity selection: 0 - HREF positive effective, 1 - HREF negative. COMC[2] - gamma selection: 1 - RGB Gamma on; 0 - gamma is 1. COMC[1] - reserved COMC[0] - reserved
	COMD	01	RW	Common Control D COMD[7] - reserved bit. COMD[6] - PCLK polarity selection. "0" OV6620 output data at PCLK falling edge and data bus will be stable at PCLK rising edge; "1" rising edge output data and stable at PCLK falling edge. When OV6620 work as CCIR656 format, COMB4=1, this bit is disable and should use PCLK rising edge latch data bus. COMD[5:1] - reserved bit. COMD[0] - U V digital output sequence exchange control. 1 - UV UV ... for 16-bit, U Y V Y ... for 8-bit; 0 - V U V U ... for 16Bit and V Y U Y ... for 8 Bit. Note: COMD[0] is not programmable on the OV6120 Image Sensor.
	FSD	03	RW	Field Slot Division FSD[7:2] - Field Interval selection. Odd Even mode defined by FD[1:0] 000000 - disable digital data output, only output black reference level. 000001 - divide to 2 slots, HREF is active one in every 2 field/frame 000010 - divide to 4 slots, HREF is active one in every 4 field/frame 000100 - divide to 8 slots, HREF is active one in every 8 field/frame 001000 - divide to 16 slots, HREF is active one in every 16 field/frame 010000 - divide to 32 slots, HREF is active one in every 32 field/frame 100000 - divide to 64 slots, HREF is active one in every 64 field/frame FSD[1:0] - field mode selection. Each frame consists of two fields: Odd & Even, these bits defines the assertion of HREF in relation to the two fields. 00 - OFF mode; HREF is not asserted in both fields, one exception is the single frame transfer operation (see the description for the register 13) 01 - ODD mode; HREF is asserted in odd field only. 10 - EVEN mode; HREF is asserted in even field only. 11 - FRAME mode; HREF is asserted in both odd field and even field. FD[7:2] useless.
	HREFST	38	RW	Horizontal HREF Start HS[7:0] - selects the starting point of HREF window, each LSB represents two pixels for CIF resolution mode, one pixels for QCIF resolution mode, this value is set based on an internal column counter, the default value corresponds to 352 horizontal window. Maximum window size is 356. see window description below. HS[7:0] programmable range is [38]- [EB], and should less than HE[7:0]. HS[7:0] should be programmable to value larger than or equal to [38]. Value larger than [EC] is invalid. See window description below.
	HREFEND	EA	RW	Horizontal HREF End HE[7:0] - selects the ending point of HREF window, each LSB represents two pixels for full resolution and one pixels for QCIF resolution, this value is set based on an internal column counter, the default value corresponds to the last available pixel. The HE[7:0] programmable range is [39] - [EC]. HE[7:0] should be larger than HS[7:0] and less than or equal to [EC]. Value larger than [EC] is invalid. See window description below.
	VSTRT	03	RW	Vertical Line Start VS[7:0] - selects the starting row of vertical window, in full resolution mode, each LSB represents 1 scan line in one frame. see window description below. Min. is [03], max. is [93] and should less than VE[7:0].
	VEND	92	RW	Vertical Line End VE[7:0] - selects the ending row of vertical window, in full resolution mode, each LSB represents 1 scan line in one frame. see window description below. Min. is [D4], max. is [94] and should larger than VS[7:0].

OV6620/OV6120

SINGLE IC CMOS COLOR AND B/W DIGITAL CAMERAS

Subaddress (hex)	Register	Default (hex)	Read/Write	Descriptions
1B	PSHFT	00	RW	Pixel Shift PS[7:0] - to provide a way to fine tune the output timing of the pixel data relative to that of HREF, it physically shifts the video data output time late in unit of pixel clock as shown in the figure below. This function is different from changing the size of the window as is defined by HS[7:0] & HE[7:0] in register 17&18. Higher than default number shifts the pixel in delay(right) direction, the highest number is "FF", so maximum shift number is: Late: 256 pixels.
1C	MIDH	7F	R	Manufacture ID Byte: High MIDH[7:0] - read only, always returns "7F" as manufacturer's ID no.
1D	MIDL	A2	R	Manufacture ID Byte: Low MIDL[7:0] - read only, always returns "A2" as manufacturer's ID no.
1E	Rsvrd1E	C4	R	reserved
1F	Rsvrd1F	04	R	reserved
20	COME	00	RW	Common Control E COME[7] - HREF pixel number selection. "1" - HREF include 704 PCLK, every data output twice. COME[6] - reserved. COME[5] - "1" First stage aperture correction enable. Correction strength will be decided by register [07]. "0" disable first stage aperture correction. COME[4] - "1" Second stage aperture correction enables. Correction strength and threshold value will be decided by COMF[7:1] ~ COMF[4]. COME[3] - AWB smart mode enable. 1 - Drop out pixel when compare pixel red, blue and green component level to change register [01] and [02], which luminance level is higher than presetting level and lower than presetting level, this two level is set by register [0F]. 0 - calculate all pixels to get AWB result. Valid only when COMB[0]=1 and COMA[2]=1. COME[2] - AWB stop when field/frame image average luminance level is lower than a presetting level enable. 1 - enable stop AWB when image luminance level is low. 0 - AWB is independent with field/frame luminance level. Valid only when COMB[0]=1 and COMA[2]=1. Average compare level is set by GAM[7:5]. COME[1] - AWB fast/slow mode selection. "1" - AWB is always fast mode, that is register [D1] and [D2] is changed every field/frame. "0" AWB is slow mode, [D1] and [D2] change every 16/64 field/frame decided by COMK[1]. When AWB enable, COMA[2]=1, AWB is working as fast mode at first 1024 field/frame, then as slow mode later. COME[0] - Digital output driver capability increase selection: "1" Double digital output driver current; "0" low output driver current status. Note: COME[3] (AWB Smart Mode), COME[2] (AWB Stop), and COME[1] (AWB Fast/Slow) are not programmable on the OV6120 Image Sensor.
21	YOFF	80	RW	Y Channel Offset Adjustment YOFF[7] - Offset adjustment direction 0 - Add Y[6:0]; 1 - Subtract Y[6:0]. YOFF[6:0] - Y channel digital output offset adjustment. Range: +127mV ~ -127mV. If COMG[2]=0, this register will be updated by internal auto A/D BLC circuit, and write a value to this register with SCCB has no effect. If COMG[2]=1, Y channel offset adjustment will use the register stored value which can be changed by SCCB. If COMF[1]=0, this register has no adjustment effect to A/D output data. If output RGB raw data, this register will adjust R/G/B data.
22	UOFF	80	RW	U Channel Offset Adjustment UOFF[7] - Offset adjustment direction: 0 - Add U[6:0]; 1 - Subtract U[6:0]. UOFF[6:0] - U channel digital output offset adjustment. Range: +128mV ~ -128mV. If COMG[2]=0, this register will be updated by internal auto A/D BLC circuit, and write a value to this register with SCCB has no effect. If COMG[2]=1, U channel offset adjustment will use the register stored value which can be changed by SCCB. If COMF[1]=1, this register has no effect to A/D output data. If output RGB raw data, this register will adjust R/G/B data. Note: This function is not available on the OV6120 Image Sensor.
23	REFC	04	RW	Reference Control REFC[7:6] - Select different crystal circuit power level (11 = minimum). REFC[5:4] - reserved REFC[3:0] - Reference Voltage range selection. 2.5V ~ 3.5V and step is 0.0625V.
24	AEW	33	RW	Automatic Exposure Control: Bright Pixel Ratio Adjustment AEW[7:0] - Used as calculate bright pixel ratio. OV6620 AEC algorithm is count whole field/frame bright pixel (its luminance level is higher than a fixed level) and black pixel (its luminance level is lower than a fixed level) number. When bright/black pixel ratio is same as the ratio defined by register [25] and [26], image stable. This register is used to define bright pixel ratio, default is 25%, each LSB represent step: 1.3% Change range is: [D1] ~ [CA]. Increase AEW[7:0] will increase bright pixel ratio. For same light condition, the image brightness will increase if AEW[7:0] increase. Note: AEW[7:0] must combine with register [25] AEB[7:0]. The relation must be as follows: $AEW[7:0] + AEB[7:0] > [CA]$.

/6620/OV6120

SINGLE IC CMOS COLOR AND B/W DIGITAL CAMERAS

Address (hex)	Register	Default (hex)	Read/Write	Descriptions
	AEB	97	RW	Automatic Exposure Control: Black Pixel Ratio Adjustment AEB[7:0] - used as calculate black pixel ratio. OV6620 AEC algorithm is count whole field/frame bright pixel (its luminance level is higher than a fixed level) and black pixel (its luminance level is lower than a fixed level) number. When bright/black pixel ratio is same as the ratio defined by register [25] and [26], image stable. This register is used to define black pixel ratio, default is 75%, each LSB represent step: 1.3%; Change range is: [01] - [CA]; Increase AEB[7:0] will increase black pixel ratio. For same light condition, the image brightness will decrease if AEB[7:0] increase. Note: AEB[7:0] must be combined with register [25] AEW[7:0]. The relation must be as follows: $EW[7:0] + AEB[7:0] > [CA]$.
	COMF	B0	RW	Common Control F COMF[7:6] - Second aperture correction threshold selection. [00] - Difference of neighbor pixel luminance is larger than 8 mV; correction on. [01] - 16 mV. [10] - 32 mV. [11] - 64 mV. COMF[5:4] - Second aperture correction strength selection. [00] and [01] - Strength is 50% of difference of neighbor pixel luminance. [10] - 100%. [11] - 200%. COMF[3] - UV BLC swap. "1" swap; "0" no swap. COMF[2] - Digital data MSB/LSB swap. "1" LSB->Bit7, MSB->Bit0; "0" normal. COMF[1] - "1" A/D Black level calibration enable. "0" Disable A/D BLC. COMF[0] - "1" Output first 4 line black level before valid data output. HREF number will increase 4 relatively. "0" no black level output.
	COMG	A0	RW	Common Control G COMG[7] - reserved. COMG[6] - reserved. COMG[5] - Select CKOUT pin output V flag. 1 - CKOUT output V flag signal. CKOUT=1, means related UV channel output V component (or Red component), CKOUT=0 pointed to U component (or Blue component). 0 - CKOUT output buffered XCLK2 COMG[4] - reserved. COMG[3] - reserved. COMG[2] - "1" A/D offset adjustment manually mode enable: 1 - A/D data will be add/substrate a value defined by register [21] and [22], which content is written by SCCB. 0 - A/D data will be added/substrate a value defined by register [21] and [22], which is updated by internal circuit. COMG[1] - Digital output full range selection. OV6620 output data value range is [10] - [F0], if COMG[1] -1, range change to [01] - [FE] with signal overshoot and undershoot level. COMG[0] - reserved.
	COMH	01	RW	Common Control H COMH[7] - "1" selects One-Line RGB raw data output format, "0" selects normal two-line RGB raw data output, effective only in Progressive Scan mode. COMH[6] - "1" enable Black/White mode. When OV6620 working as BW camera, its vertical resolution will be higher than color mode. At this mode, can't set OV6620 working at 8 bit output mode. OV6620 output data YUV/RGB from Y port. UV port will be tri-state. COMB[5] and COMB[4] will be set to "0". "0" normal color mode. COMH[5] - reserved. COMH[4] - Freeze AEC/AGC value, effective only when COMB0=1. "1" - register [00] and [10] will not be updated and hold latest value. "0" - AEC/AGC normal working status. COMH[3] - AGC disable. 1 - when COMB[0]=1 and COMA[5]=1, internal circuit will not update register [00], register [00] will kept latest updated value before COMH[3]=1. 0 - when COMB0=1 and COMA[5]=1, register [00] will be updated by internal algorithm. COMH[2] - RGB raw data output YG format: 1 - Y channel G, UV channel B R; 0 - Y channel: G R G R..., UV channel B G B G.... COMH[1] - Gain control bit. "1" Double PreAmp gain to 12dB. "0" PreAmp gain is 6dB. COMH[0] - High gain mode. "1" - AGC maximum gain is 24dB. AGC step is 1/8. "0" AGE maximum gain is 18dB, AGC step is 1/16. Only effective when COMB[0]=1, COMA[5]=1 and COMH[3]=0. Note: COMH[2] (RGB Raw Data) is not programmable on the OV6120 image Sensor.

OV6620/OV6120

SINGLE IC CMOS COLOR AND B/W DIGITAL CAMERAS

Subaddress (hex)	Register	Default (hex)	Read/Write	Descriptions
29	COMI	00	RW	Common Control I COMI[7]: - AEC disable. "1" if COMB[0]=1, AEC stop and register [10] value will be held at last AEC value and not be updated by internal circuit. "0" - if COMB[0]=1, register [10] value will be updated by internal circuit. COMI[6]: - Slave mode selection. "1" slave mode, use external Sync and Vsync; "0" master mode. COMI[5]: - reserved COMI[4]: - reserved COMI[3]: - Central 1/4 image area rather whole image used to calculate AEC/AGC. "0" use whole image area to calculate AEC/AGC. COMI[2]: - reserved COMI[1:0]: - Version flag. For Version A, value is [00], these two bits can only be read.
2A	FRARH	84	RW	Frame Rate Adjust High FRARH[7]: - Frame Rate adjustment enable bit. "1" Enable. FRARH[6]: - reserved FRARH[5]: - Highest 1bit of frame rate adjust control byte, see explanation below. FRARH[4]: - reserved FRARH[3]: - Y channel brightness adjustment enable. When COMF[2]=1 active. FRARH[2]: - reserved FRARH[1]: - "1" When in Frame exposure mode, only One frame data output. FRARH[0]: - reserved
2B	FRARL	6E	RW	Frame Rate Adjust Low FRARL[7:0]: - Lowest 8 bit of frame rate adjust control byte. Frame rate adjustment resolution is 0.21%. Control byte is 10 bit. Every LSB equal decrease frame rate 0.21%. Range is 0.21% - 109%. If frame rate adjustment enable, COME7 must set to "0".
2C	Rsvd2C	88	RW	reserved
2D	COMJ	03	RW	Common Control J COMJ[7:5]: - reserved COMJ[4]: - Enable auto black expanding mode. COMJ[3]: - "1" = White Balance update when AGC/AEC stable. "0" = White Balance register update independent with AEC/AGC. COMJ[2]: - Band filter enable. After adjust frame rate to match indoor light frequency, this bit enable a different exposure algorithm to cut light band induced by fluorescent light. COMJ[1]: - reserved COMJ[0]: - A/D U and V BLC separate mode. "1" = U and V offset cancelled by different register. "0" = U V offset cancelled by one common register [2E].
2E	VCOFF	80	RW	V Channel Offset Adjustment VCOFF[7]: Offset adjustment direction: "0" = Add V[6:0]; "1" = Subtract V[6:0]. VCOFF[6:0]: V channel digital output offset adjustment. Range: +128mV ~ -128mV. If COMG[2]=0, this register will be updated by internal auto A/D BLC circuit, and write a value to this register with SCCB has no effect. If COMG[2]=1, V channel offset adjustment will use the register stored value which can be changed by SCCB. If COMF[1]=1, this register has no effect to A/D output data. If output RGB raw data, this register will adjust R/G/B data. Note: This function is not available on the OV6120 Image Sensor.
2F-32	Rsvd2F-Rsvd32	xx	-	Reserved
33	CPP	00	RW	Color Processing Parameter Control CPP[7:6]: - reserved CPP[5]: - Luminance gamma on/off. "1" - luminance gamma on; "0" - luminance gamma is 1. CPP[4:0]: - reserved
34	BIAS	A2	RW	Bias Adjustment BIAS[7:6]: - A/D reference level adjustment. [00] - 110% internal full signal range; [01] - 120%, [10] - 130%, [11] - 140%. BIAS[5:0]: - reserved
35	Rsvd35	80	RW	reserved
36	Rsvd36	48	RW	reserved
37	Rsvd37	41	RW	reserved

/6629/OV6120

SINGLE IC CMOS COLOR AND B/W DIGITAL CAMERAS

Address (hex)	Register	Default (hex)	Read/Write	Descriptions
	COMK	81	RW	Common Control K COMK[7] - HREF edge latched by PCLK falling edge (When COMD[6] = 0). "0" HREF edge is 10 ns after PCLK rising edge. COMK[6] - Output port drive current additional 2x control bit. COMK[5] - reserved. COMK[4] - ZV port Vertical timing selection. "1" VSYNC output ZV port vertical sync signal. "0" = normal TV vertical sync signal. COMK[3] - Quick stable mode when camera mode change. After relative control bit set, the first VS will be the stable image with suitable AEC/AWB setting. "0" - slow mode, after mode change need more field/frame to get stable AEC/AWB setting image. COMK[2] - reserved. COMK[1] - AWB stable time selection when in slow mode. "1" - 4 times less time needed to get stable AWB setting when in slow AWB mode. COMK[0] - reserved. Note: COMK[1] (AWB Stable Time) is not programmable on the OV6120 Image Sensor.
	COML	00	RW	Common Control L COML[7] - reserved COML[6] - PCLK output timing selection. 1 - PCLK valid only when HREF is high; 0 - PCLK is free running. COML[5] - Vertical sync selection, 1 - Same period between 1st HREF and VS falling edge in two field; 0 - Different timing period between 1st HREF and VS falling edge COML[4] - "1" select CHSYNC output from HREF port. "0" normal COML[3] - "1" select HREF output from CHSYNC port. "0" normal COML[2] - Tri-state all control signal output (FODD, CHSYNC, HREF, PCLK) COML[1] - Highest 1 bit of horizontal sync starting position, combined with register [3A] COML[0] - Highest 1 bit of horizontal sync ending position, combined with register [3B]
	HSST	0F	RW	Horizontal Sync Start Position HSST[7:0] - lower 8 bit of horizontal sync starting position, combined with register bit of COML[1], total 9 bit control. range: [00] - [FF]. HSEND[8:0] must less than HSST[8:0]
	HSEND	3C	RW	Horizontal Sync End Position HEND[7:0] - lower 8 bit of horizontal sync ending position, combined with register bit of COML[0], total 9 bit control. range: [00] - [FF]. HSEND[8:0] must be larger than HSST[8:0]
	COMM	21	RW	Common Control M COMM[7:5] - Select minimum AEC number if Banding filter enable. [000] - 1 field, [001] - 1/2; [010] - 1/4; [011] - 1/8; [100] - 1/16; [101]-[111] - 1/32; COMM[4] - AEC/AGC change mode selection COMM[3] - AEC/AGC change mode selection COMM[2] - AEC/AGC change fastest mode COMM[1] - AEC/AGC change fast mode COMM[0] - AEC/AGC change slowest mode
	COMN	08	RW	Common Control N COMN[7] - Enable one frame drop when AEC change to keep data valid when Banding filter mode enable. COMN[6:4] - reserved COMN[3] - Enable 50 Hz PAL video timing, so VTO analog signal can be displayed on TV COMN[2:0] - reserved
	COMO	80	RW	Common Control O COMO[7] - Input main clock divided by 2 or 4 selection. 1 - 2; 0 - 4 COMO[6:5] - reserved COMO[4] - Select 4 bit nibble mode output COMO[3] - reserved COMO[2] - Enable Minimum exposure time is 4 line. Default is 1 line COMO[1] - reserved COMO[0] - reserved
	COMP	02	RW	Common Control P COMP[7] - reserved. COMP[6] - Output main clock output from FODD port COMP[5] - reserved COMP[4] - Software whole chip power down enable, can be waked up by disable this bit COMP[3:2] - reserved COMP[1] - CCR656 output control COMP[0] - Reset internal timing circuit without reset AEC/AGC/AWB value
	Rsvd40 - Rsvd4C	XX	-	reserved

OV6620/OV6120

SINGLE IC CMOS COLOR AND B/W DIGITAL CAMERAS

Sub-address (hex)	Register	Default (hex)	Read/Write	Descriptions
0	YMXA	02	RW	YUV Matrix Control (Main) YMXA[7:5] - reserved YMXA[4:3] - YUV/YC/CB selection: [00] U = u, V = v [01] U = 0.838u, V = 0.838v [10] U = 0.563u, V = 0.714v [11] U = 0.5u, V = 0.677v YMXA[2:0] - Reserved Note: This function is not available on the OV6120 Image Sensor.
3	ARL	A0	RW	AEC/AGC Reference Level ARL[7:5] - Reference Level Voltage Selection (Higher voltage = brighter final stable image) [000] = 1.3v [001] = 1.5v [010] = 1.6v [011] = 1.7v [100] = 1.8v [101] = 1.9v [110] = 2.0v [111] = 2.1v ARL[4:0] - Reserved
4	YMXB	00	RW	YUV Matrix Control (Secondary) YMXB[7:6] - Y channel delay selection: 0 ~ 3 tp YMXB[5:4] - UV delay selection: 0 ~ 8 tp YMXB[3:2] - Select UV average mode. [00] & [10]: UD/V0 (no delay); [01] - 3 point average; [11] - 5 point average mode YMXB[1:0] - Color killer control: [00]:2.4v,[01]:2.6v,[10]:2.8v,[11]:3.0v Note: This function is not available on the OV6120 Image Sensor.
5	Rsvd50 - Rsvd53	XX	-	reserved

Features

High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
Advanced RISC Architecture

- 131 Powerful Instructions – Most Single-clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- Fully Static Operation
- Up to 16 MIPS Throughput at 16 MHz
- On-chip 2-cycle Multiplier

High Endurance Non-volatile Memory segments

- 32Kbytes of In-System Self-programmable Flash program memory
- 1024bytes EEPROM
- 2Kbyte Internal SRAM
- Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
- Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
- Optional Boot Code Section with Independent Lock Bits
- In-System Programming by On-chip Boot Program
- True Read-While-Write Operation

- Programming Lock for Software Security

JTAG (IEEE std. 1149.1 Compliant) Interface

- Boundary-scan Capabilities According to the JTAG Standard
- Extensive On-chip Debug Support
- Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface

Peripheral Features

- Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- Real Time Counter with Separate Oscillator
- Four PWM Channels
- 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels in TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x

- Byte-oriented Two-wire Serial Interface

- Programmable Serial USART

- Master/Slave SPI Serial Interface

- Programmable Watchdog Timer with Separate On-chip Oscillator

- On-chip Analog Comparator

Special Microcontroller Features

- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources
- Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby

Pins and Packages

- 32 Programmable I/O Lines
- 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF

Operating Voltages

- 2.7V - 5.5V for ATmega32L
- 4.5V - 5.5V for ATmega32

Speed Grades

- 0 - 8MHz for ATmega32L
- 0 - 16MHz for ATmega32

Power Consumption at 1 MHz, 3V, 25°C

- Active: 1.1mA
- Idle Mode: 0.35mA
- Power-down Mode: < 1µA



8-bit AVR[®]
Microcontroller
with 32KBytes
In-System
Programmable
Flash

ATmega32
ATmega32L

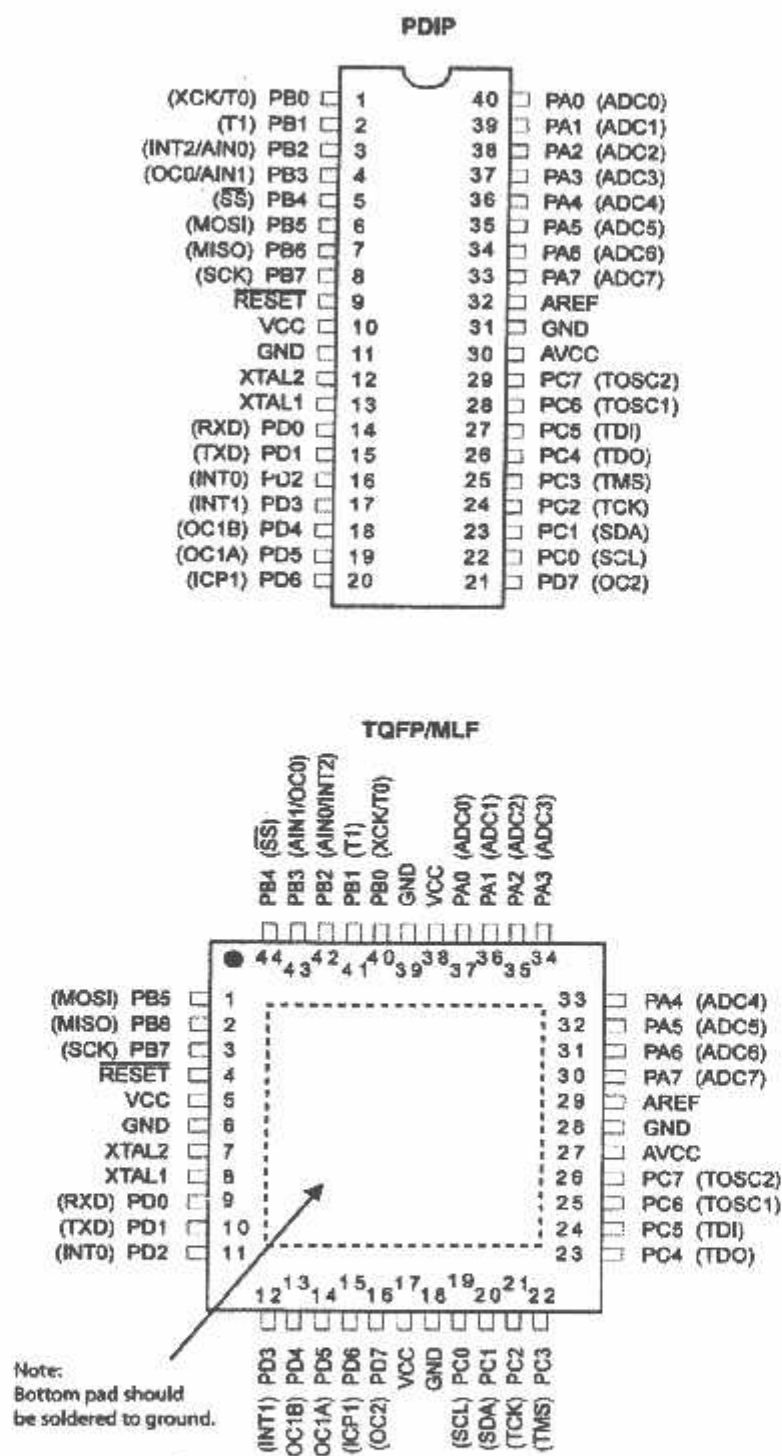
Summary

2503QS-AVR-02/11



Configurations

Figure 1. Pinout ATmega32

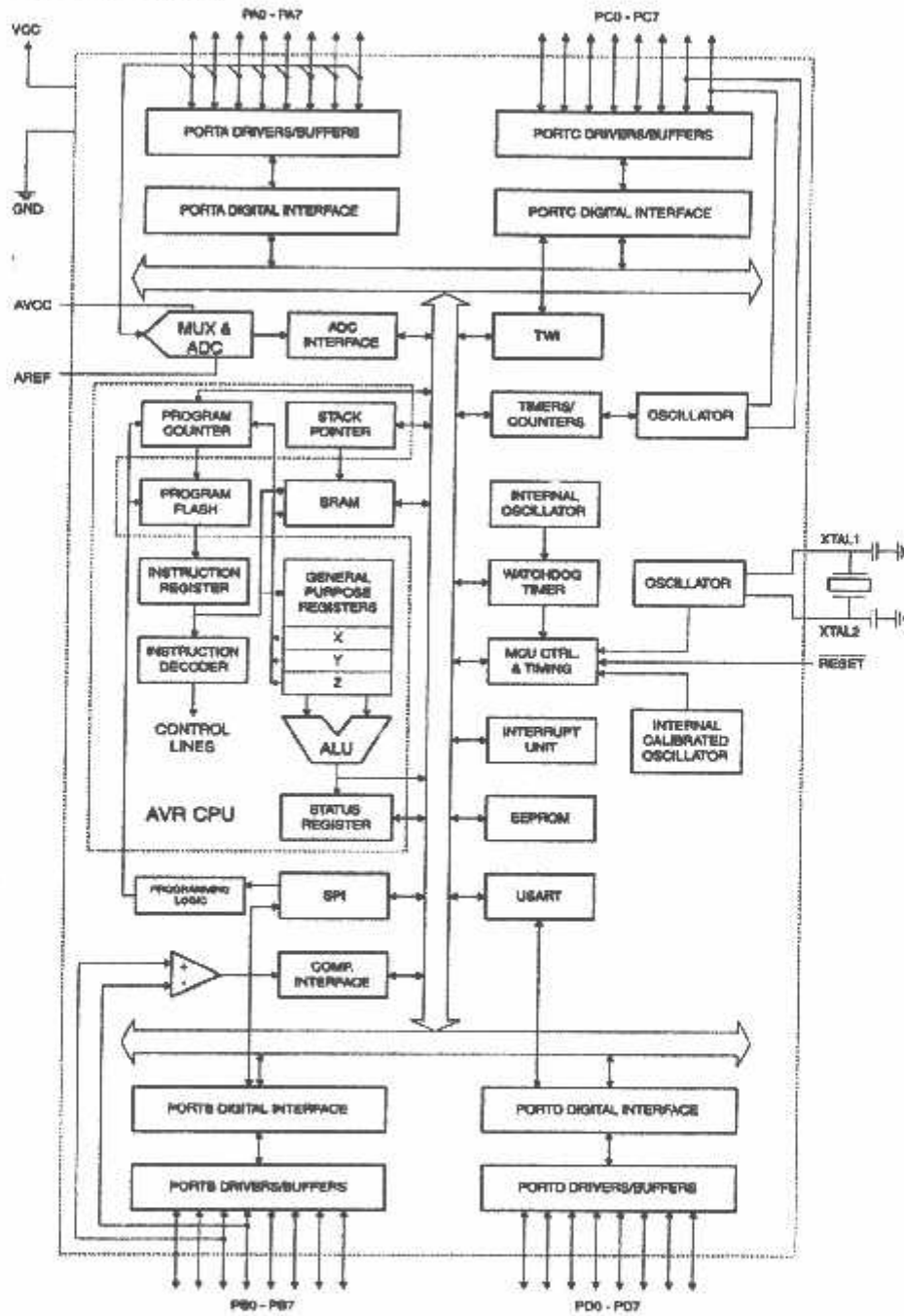


Overview

The Atmel® AVR® ATmega32 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega32 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



The Atmel® AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega32 provides the following features: 32Kbytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 1024bytes EEPROM, 2Kbyte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega32 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The Atmel AVR ATmega32 is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Descriptions

Digital supply voltage.

Ground.

A (PA7..PA0)

Port A serves as the analog inputs to the A/D Converter.

Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.



Port B (PB7..PB0)	Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special features of the ATmega32 as listed on page 57.
Port C (PC7..PC0)	Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs. The TDO pin is tri-stated unless TAP states that shift out data are entered. Port C also serves the functions of the JTAG interface and other special features of the ATmega32 as listed on page 60.
Port D (PD7..PD0)	Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the ATmega32 as listed on page 62.
RESET	Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 37. Shorter pulses are not guaranteed to generate a reset.
AL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
AL2	Output from the inverting Oscillator amplifier.
AVCC	AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.
AREF	AREF is the analog reference pin for the A/D Converter.

Resources

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.

Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C Compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C Compiler documentation for more details.



Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
\$3F (\$3F)	SREG	I	T	H	S	V	N	Z	C	10
\$3E (\$3E)	SPH	—	—	—	—	SP11	SP10	SP9	SP8	12
\$3D (\$3D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	12
\$3C (\$3C)	OCR0	Timer/Counter0 Output Compare Register								82
\$3B (\$3B)	GICR	INT1	INT0	INT2	—	—	—	IVSEL	IVCE	47, 67
\$3A (\$3A)	GIFR	INTF1	INTF0	INTF2	—	—	—	—	—	68
\$39 (\$39)	TIMSK	OCIE2	TOIE2	TCIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	82, 112, 130
\$38 (\$38)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	83, 112, 130
\$37 (\$37)	SPICR	SPMIE	RWWSB	—	RWWSRE	BIBSET	PGWRT	PGERS	SPMEN	248
\$36 (\$36)	TWCR	TWINT	TWSTA	TWSTA	TWSTO	TWAC	TWEN	—	TWSE	177
\$35 (\$35)	MCUCR	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00	32, 66
\$34 (\$34)	MCUCSR	JTD	ISC2	—	JTRF	WDRF	BORF	EXTRF	PORF	40, 67, 228
\$33 (\$33)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	80
\$32 (\$32)	TCNT0	Timer/Counter0 (8 Bits)								82
0177 (\$51) (*)	OSCAL	Oscillator Calibration Register								30
	QCDR	On-Chip Debug Register								274
\$30 (\$30)	SFIO	ADTS2	ADTS1	ADTS0	—	ACME	PUD	PSR2	PSR10	56, 65, 131, 198, 218
\$2F (\$4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	107
\$2E (\$4E)	TCCR1B	ICNC1	ICES1	—	WGM13	WGM12	CS12	CS11	CS10	110
\$2D (\$4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								111
\$2C (\$4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								111
\$2B (\$4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								111
\$2A (\$4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								111
\$29 (\$49)	OCR1EH	Timer/Counter1 – Output Compare Register B High Byte								111
\$28 (\$48)	OCR1EL	Timer/Counter1 – Output Compare Register B Low Byte								111
\$27 (\$47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								111
\$26 (\$46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								111
\$25 (\$45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	125
\$24 (\$44)	TCNT2	Timer/Counter2 (8 Bits)								127
\$23 (\$43)	OCR2	Timer/Counter2 Output Compare Register								127
\$22 (\$42)	ASSR	—	—	—	—	AS2	TCN2UB	OCR2UB	TCR2UB	128
\$21 (\$41)	WDTCR	—	—	—	WDTOE	WDE	WDP2	WDP1	WDP0	42
0x11 (\$40) *	UBRRH	URSEL	—	—	—	—	UBRR[11:8]			164
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	162
\$1F (\$3F)	EEARH	—	—	—	—	—	—	EEAR9	EEAR8	19
\$1E (\$3E)	EEARL	EEPROM Address Register Low Byte								19
\$1D (\$3D)	EEDR	EEPROM Data Register								19
\$1C (\$3C)	EEDCR	—	—	—	—	EERIE	EEMWE	EEWE	EERE	19
\$1B (\$3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	64
\$1A (\$3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	64
\$19 (\$39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	64
\$18 (\$38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	64
\$17 (\$37)	DDRB	DOB7	DOB6	DOB5	DOB4	DOB3	DOB2	DOB1	DOB0	64
\$16 (\$36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	65
\$15 (\$35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	65
\$14 (\$34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	65
\$13 (\$33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	65
\$12 (\$32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	65
\$11 (\$31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	65
\$10 (\$30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	65
\$0F (\$2F)	SPDR	SPI Data Register								138
\$0E (\$2E)	SPSR	SPIF	WCOL	—	—	—	—	—	SPI2X	138
\$0D (\$2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	136
\$0C (\$2C)	UDR	USART I/O Data Register								159
\$0B (\$2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	160
\$0A (\$2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	161
\$09 (\$29)	UBRRL	USART Baud Rate Register Low Byte								164
\$08 (\$28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	198
\$07 (\$27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	214
\$06 (\$26)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	216
\$05 (\$25)	ADCH	ADC Data Register High Byte								217
\$04 (\$24)	ADCL	ADC Data Register Low Byte								217
\$03 (\$23)	TWDR	Two-wire Serial Interface Data Register								179
\$02 (\$22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGE	179

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
\$01 (\$21)	TWGR	TWS7	TWS6	TWS5	TWS4	TWS3	TWS2	TWS1	TWS0	175
\$00 (\$20)	TWBR	Two-wire Serial Interface Bit Rate Register								177

- es:
1. When the OCDEN Fuse is unprogrammed, the OSCCAL Register is always accessed on this address. Refer to the debugger specific documentation for details on how to use the OCSR Register.
 2. Refer to the USART description for details on how to access UBRRH and UCSRC.
 3. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
 4. Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers \$00 to \$1F only.



Instruction Set Summary

mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
D	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z, C, N, V, H	1
C	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z, C, N, V, H	1
ADW	Rd, K	Add Immediate to Word	$RdH:RdL \leftarrow RdH:RdL + K$	Z, C, N, V, H	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z, C, N, V, H	1
SB	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z, C, N, V, H	1
ADC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z, C, N, V, H	1
ADC	Rd, K	Subtract with Carry Constant from Reg	$Rd \leftarrow Rd - K - C$	Z, C, N, V, H	1
ADW	Rd, K	Subtract Immediate from Word	$RdH:RdL \leftarrow RdH:RdL - K$	Z, C, N, V, H	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \& Rr$	Z, N, V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \& K$	Z, N, V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z, N, V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z, N, V	1
EXOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z, N, V	1
COM	Rd	One's Complement	$Rd \leftarrow \sim Rd$	Z, C, N, V	1
NEG	Rd	Two's Complement	$Rd \leftarrow \$00 - Rd$	Z, C, N, V, H	1
RSR	Rd, K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z, N, V	1
RCLR	Rd, K	Clear Bit(s) in Register	$Rd \leftarrow Rd \& (\sim K)$	Z, N, V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z, N, V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z, N, V	1
TEST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \& Rd$	Z, N, V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \& Rd$	Z, N, V	1
SET	Rd	Set Register	$Rd \leftarrow \$FF$	None	1
MUL	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z, C	2
LS	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z, C	2
LSU	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z, C	2
UL	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z, C	2
ULS	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z, C	2
ULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z, C	2
ANCHOR INSTRUCTIONS					
BR	K	Relative Jump	$PC \leftarrow PC + K + 1$	None	2
BRW		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
BR	K	Direct Jump	$PC \leftarrow K$	None	3
CALL	K	Relative Subroutine Call	$PC \leftarrow PC + K + 1$	None	3
CALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
CALL	K	Direct Subroutine Call	$PC \leftarrow K$	None	4
RET		Subroutine Return	$PC \leftarrow Stack$	None	4
IRET		Interrupt Return	$PC \leftarrow Stack$	I	4
CPI	Rd, Rr	Compare, Skip if Equal	$\text{if } (Rd = Rr) \text{ then } PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
CPI	Rd, Rr	Compare	$Rd - Rr$	Z, N, V, C, H	1
CPI	Rd, Rr	Compare with Carry	$Rd - Rr - C$	Z, N, V, C, H	1
CPI	Rd, K	Compare Register with Immediate	$Rd - K$	Z, N, V, C, H	1
SCD	Rr, b	Skip if Bit in Register Cleared	$\text{if } (Rr(b)=0) \text{ then } PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
SCD	Rr, b	Skip if Bit in Register is Set	$\text{if } (Rr(b)=1) \text{ then } PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
SCD	P, b	Skip if Bit in I/O Register Cleared	$\text{if } (P(b)=0) \text{ then } PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
SCD	P, b	Skip if Bit in I/O Register is Set	$\text{if } (P(b)=1) \text{ then } PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
BR	S, k	Branch if Status Flag Set	$\text{if } (SREG(s)=1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	S, k	Branch if Status Flag Cleared	$\text{if } (SREG(s)=0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if Equal	$\text{if } (Z=1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if Not Equal	$\text{if } (Z=0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if Carry Set	$\text{if } (C=1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if Carry Cleared	$\text{if } (C=0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if Same or Higher	$\text{if } (C=0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if Lower	$\text{if } (C=1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if Minus	$\text{if } (N=1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if Plus	$\text{if } (N=0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if Greater or Equal, Signed	$\text{if } (N \oplus V=0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if Less Than Zero, Signed	$\text{if } (N \oplus V=1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if Half Carry Flag Set	$\text{if } (H=1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if Half Carry Flag Cleared	$\text{if } (H=0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if T Flag Set	$\text{if } (T=1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if T Flag Cleared	$\text{if } (T=0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if Overflow Flag is Set	$\text{if } (V=1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BR	k	Branch if Overflow Flag is Cleared	$\text{if } (V=0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2

Instruction	Operands	Description	Operation	Flags	#Cycles
RIE	k	Branch If Interrupt Enabled	If (I = 1) then PC ← PC + k + 1	None	1 / 2
RID	k	Branch If Interrupt Disabled	If (I = 0) then PC ← PC + k + 1	None	1 / 2
ATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rr ← Rd	None	1
MOVW	Rd, Rr	Copy Register Word	Rd ← Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LDD	Rd, X	Load Indirect	Rd ← (X)	None	2
LDD+	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LDD-, X	Rd, -X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LDD, Y	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LDD+, Y	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LDD-, Y	Rd, -Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD, Y+q	Rd, Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LDD, Z	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LDD+, Z	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z + 1	None	2
LDD-, Z	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD, Z+q	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
STS	X, Rr	Store Indirect	(X) ← Rr	None	2
STSD	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
STSD-, X	-X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
STD	Y, Rr	Store Indirect	(Y) ← Rr	None	2
STD+	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
STD-, Y	-Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD, Y+q	Y+q, Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
STD, Z	Z, Rr	Store Indirect	(Z) ← Rr	None	2
STD+, Z	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
STD-, Z	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD, Z+q	Z+q, Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STL	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LDI	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LDD	Rd, Z+	Load Program Memory and Post-Inc.	Rd ← (Z), Z ← Z + 1	None	3
STI	Rd, Z	Store Program Memory	(Z) ← Rd	None	3
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
SHR	Rr	Push Register on Stack	Stack ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← Stack	None	2
AND BIT-TEST INSTRUCTIONS					
SBI	P, b	Set Bit in I/O Register	IO(P,b) ← 1	None	2
CBR	P, b	Clear Bit in I/O Register	IO(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z, C, N, V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z, C, N, V	1
ROL	Rd	Rotate Left Through Carry	Rd(0) ← C, Rd(n+1) ← Rd(n), C ← Rd(7)	Z, C, N, V	1
ROR	Rd	Rotate Right Through Carry	Rd(7) ← C, Rd(n) ← Rd(n+1), C ← Rd(0)	Z, C, N, V	1
RRD	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z, C, N, V	1
SWAP	Rd	Swap Nibbles	Rd(3..0) ← Rd(7..4), Rd(7..4) ← Rd(3..0)	None	1
SF	s	Flag Set	SREG(s) ← 1	SREG(s)	1
CF	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
ST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
LD	Rd, b	Bit Load from T to Register	Rd(b) ← T	None	1
SC		Set Carry	C ← 1	C	1
CC		Clear Carry	C ← 0	C	1
SN		Set Negative Flag	N ← 1	N	1
CN		Clear Negative Flag	N ← 0	N	1
ZS		Set Zero Flag	Z ← 1	Z	1
CZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CPI		Global Interrupt Disable	I ← 0	I	1
SF		Set Signed Test Flag	S ← 1	S	1
CF		Clear Signed Test Flag	S ← 0	S	1
OV		Set Two's Complement Overflow	V ← 1	V	1
CV		Clear Two's Complement Overflow	V ← 0	V	1
ST		Set T in SREG	T ← 1	T	1
CT		Clear T in SREG	T ← 0	T	1
H		Set Half Carry Flag in SREG	H ← 1	H	1

mnemonics	Operands	Description	Operation	Flags	#Clocks
CLH		Clear Half Carry Flag in SREG	$H \leftarrow 0$	H	1
CU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDOR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-Chip Debug Only	None	N/A



Ordering Information

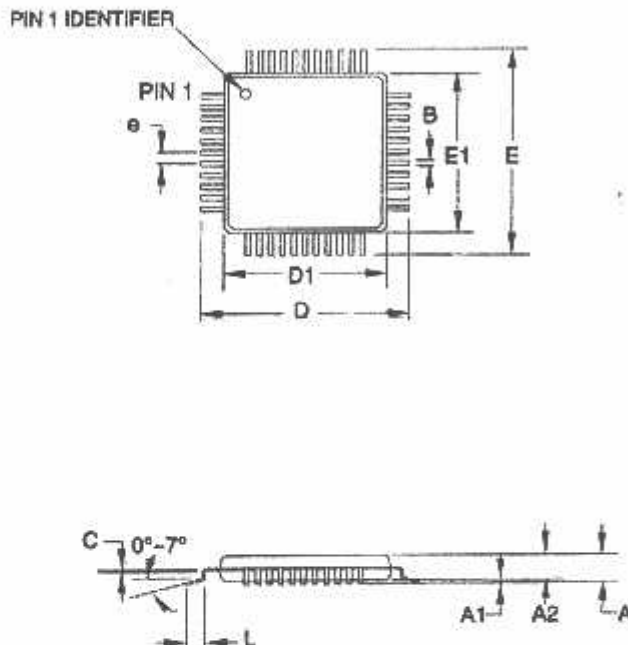
Speed (MHz)	Power Supply	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
8	2.7V - 5.5V	ATmega32L-8AU	44A	Industrial (-40°C to 85°C)
		ATmega32L-8AUR ⁽³⁾	44A	
		ATmega32L-8PU	40P6	
		ATmega32L-8MU	44M1	
		ATmega32L-8MUR ⁽³⁾	44M1	
16	4.5V - 5.5V	ATmega32-16AU	44A	
		ATmega32-16AUR ⁽³⁾	44A	
		ATmega32-16PU	40P6	
		ATmega32-16MU	44M1	
		ATmega32-16MUR ⁽³⁾	44M1	

- Notes:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. Tape & Reel

Package Type	
A	44-lead, 10 × 10 × 1.0mm, Thin Profile Plastic Quad Flat Package (TQFP)
P6	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
M1	44-pad, 7 × 7 × 1.0mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

Packaging Information

A



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	—	—	1.20	
A1	0.05	—	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	—	0.45	
C	0.09	—	0.20	
L	0.45	—	0.75	
e	0.80 TYP			

Notes:

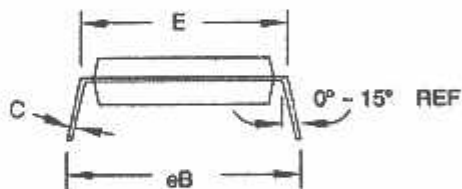
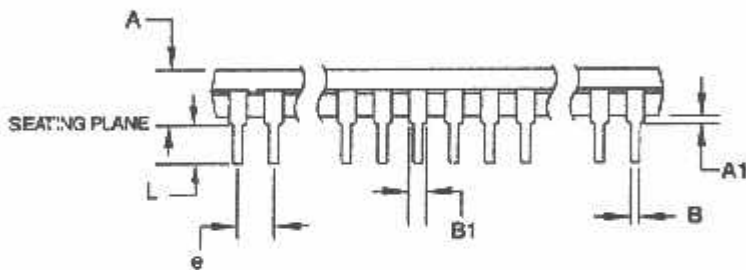
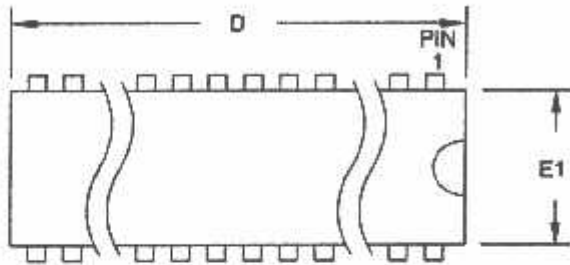
1. This package conforms to JEDEC reference MS-026, Variation ACB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

2010-10-20

TITLE		DRAWING NO.	REV.
 2325 Orchard Parkway San Jose, CA 95131		44A	C
44A, 44-lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)			



P6



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	—	—	4.826	
A1	0.381	—	—	
D	52.070	—	52.578	Note 2
E	15.240	—	15.875	
E1	13.462	—	13.970	Note 2
B	0.356	—	0.559	
B1	1.041	—	1.651	
L	3.048	—	3.556	
C	0.203	—	0.381	
eB	15.494	—	17.526	
e	2.540 TYP			

- Notes:
1. This package conforms to JEDEC reference MS-011, Variation AC.
 2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01



2325 Orchard Parkway
San Jose, CA 95131

TITLE

40P6, 40-lead (0.600"/15.24 mm Wide) Plastic Dual
Inline Package (PDIP)

DRAWING NO.

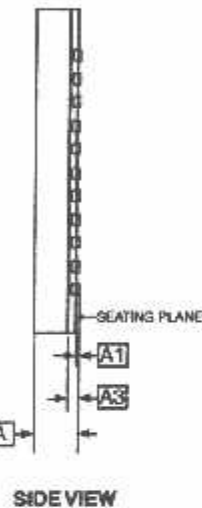
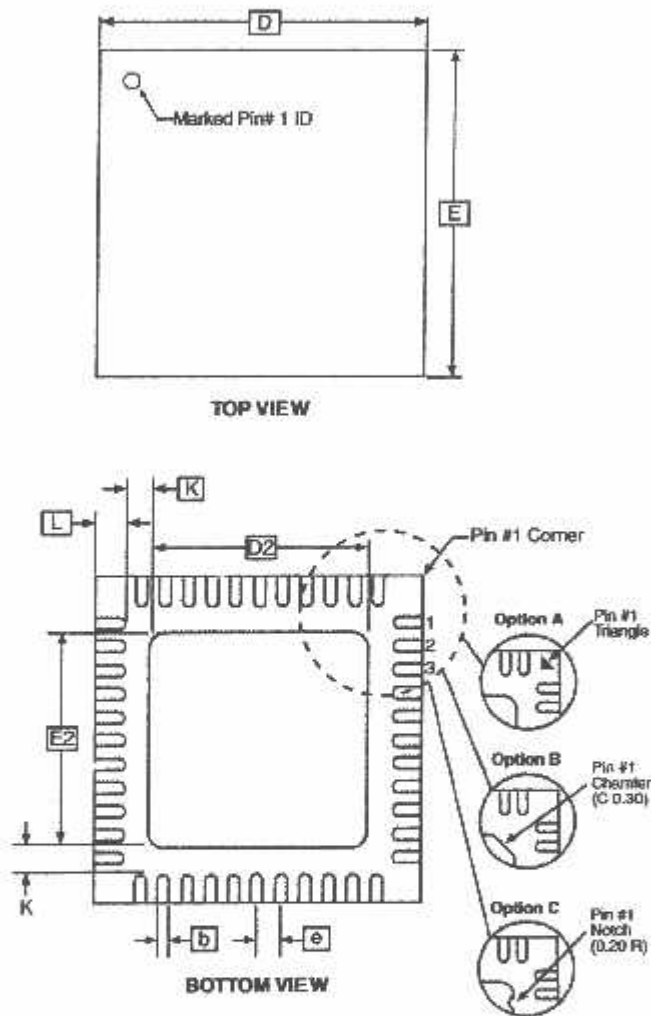
40P6

REV.

B



M1



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.90	1.00	
A1	—	0.02	0.05	
A3	0.20 REF			
b	0.18	0.23	0.30	
D	6.90	7.00	7.10	
D2	5.00	5.20	5.40	
E	6.90	7.00	7.10	
E2	5.00	5.20	5.40	
e	0.50 BSC			
L	0.59	0.64	0.69	
K	0.20	0.26	0.41	

Note: JEDEC Standard MO-220, Fig. 1 (SAW Singulation) VKKD-3.

9/26/08



Package Drawing Contact:
packagedrawings@atmel.com

TITLE
44M1, 44-pad, 7 x 7 x 1.0 mm Body, Lead
Pitch 0.50 mm, 5.20 mm Exposed Pad, Thermally
Enhanced Plastic Very Thin Quad Flat No
Lead Package (VQFN)

GPC
ZWS

DRAWING NO.
44M1

REV.
H



rrata

mega32, rev. A
F

- First Analog Comparator conversion may be delayed
- Interrupts may be lost when writing the timer registers in the asynchronous timer
- IDCODE masks data from TDI input
- Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request.

1. First Analog Comparator conversion may be delayed

If the device is powered by a slow rising V_{CC} , the first Analog Comparator conversion will take longer than expected on some devices.

Problem Fix/Workaround

When the device has been powered or reset, disable then enable the Analog Comparator before the first conversion.

2. Interrupts may be lost when writing the timer registers in the asynchronous timer

The interrupt will be lost if a timer register that is synchronous timer clock is written when the asynchronous Timer/Counter register (TCNTx) is 0x00.

Problem Fix/Workaround

Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register (TCCRx), asynchronous Timer Counter Register (TCNTx), or asynchronous Output Compare Register (OCRx).

3. IDCODE masks data from TDI input

The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

Problem Fix / Workaround

- If ATmega32 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega32 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega32 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega32 must be the first device in the chain.

4. Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request.

Reading EEPROM by using the ST or STS command to set the EERE bit in the EECR register triggers an unexpected EEPROM interrupt request.

Problem Fix / Workaround

Always use OUT or SBI to set EERE in EECR.



**Datasheet
Revision
History**

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

- | | |
|--|--|
| Changes from Rev.
03P-07/09 to
Rev. 2503Q-02/11 | <ol style="list-style-type: none">1. Updated "Packaging Information" on page 333, by replacing the package 44A by a correct one.2. Updated the datasheet according to the Atmel new Brand Style Guide.3. Updated "Ordering Information" on page 332 to include Tape & Reel devices. |
| Changes from Rev.
03O-07/09 to
Rev. 2503P-07/10 | <ol style="list-style-type: none">1. Inserted Note in "Performing Page Erase by SPM" on page 251.2. Note 6 and Note 7 in Table 119 on page 290 have been removed.3. Updated "Performing Page Erase by SPM" on page 251. |
| Changes from Rev.
03N-06/08 to
Rev.
03O-07/09 | <ol style="list-style-type: none">1. Updated "Errata" on page 336 .2. Updated the TOC with new template (version 5.10) |
| Changes from Rev.
03M-05/08 to
Rev.
03N-06/08 | <ol style="list-style-type: none">1. Added the note "Not recommended for new designs" on "Features" on page 1. |
| Changes from Rev.
03L-05/08 to
Rev.
03M-05/08 | <ol style="list-style-type: none">1. Updated "Ordering Information" on page 12:<ul style="list-style-type: none">- Commercial ordering codes removed.- Non Pb-free package option removed.2. Removed note from Feature list in "Analog to Digital Converter" on page 201.3. Removed note from Table 84 on page 215. |
| Changes from Rev.
03K-08/07 to
Rev.
03L-05/08 | <ol style="list-style-type: none">1. Updated "Fast PWM Mode" on page 75 in "8-bit Timer/Counter0 with PWM" on page 69:<ul style="list-style-type: none">- Removed the last section describing how to achieve a frequency with 50% duty cycle waveform output in fast PWM mode. |
| Changes from Rev.
03J-10/06 to
Rev.
03K-08/07 | <ol style="list-style-type: none">1. Renamed "Input Capture Trigger Source" to "Input Capture Pin Source" on page 94.2. Updated "Features" on page 1.3. Added "Data Retention" on page 6.4. Updated "Errata" on page 336. |



5. Updated "Slave Mode" on page 136.

**Changes from Rev.
03I-04/06 to Rev.
03J-10/06**

1. Updated "Fast PWM Mode" on page 99.
2. Updated Table 38 on page 80, Table 40 on page 81, Table 45 on page 108, Table 47 on page 109, Table 50 on page 125 and Table 52 on page 126.
3. Updated typo in table note 6 in "DC Characteristics" on page 287.
4. Updated "Errata" on page 336.

**Changes from Rev.
03H-03/05 to
v. 2503I-04/06**

1. Updated Figure 1 on page 2.
2. Added "Resources" on page 6.
3. Added note to "Timer/Counter Oscillator" on page 31.
4. Updated "Serial Peripheral Interface – SPI" on page 132.
5. Updated note in "Bit Rate Generator Unit" on page 175.
6. Updated Table 86 on page 218.
7. Updated "DC Characteristics" on page 287.

**Changes from Rev.
03G-11/04 to
v. 2503H-03/05**

1. MLF-package alternative changed to "Quad Flat No-Lead/Micro Lead Frame Package QFN/MLF".
2. Updated "Electrical Characteristics" on page 287
3. Updated "Ordering Information" on page 332.

**Changes from Rev.
03F-12/03 to
v. 2503G-11/04**

1. "Channel" renamed "Compare unit" in Timer/Counter sections, ICP renamed ICP1.
2. Updated Table 7 on page 29, Table 15 on page 37, Table 81 on page 206, Table 114 on page 272, Table 115 on page 273, and Table 118 on page 289.
3. Updated Figure 1 on page 2, Figure 46 on page 100.
4. Updated "Version" on page 226.
5. Updated "Calibration Byte" on page 258.
6. Added section "Page Size" on page 258.
7. Updated "ATmega32 Typical Characteristics" on page 296.
8. Updated "Ordering Information" on page 332.

**Changes from Rev.
03E-09/03 to
v. 2503F-12/03**

1. Updated "Calibrated Internal RC Oscillator" on page 29.



**anges from Rev.
03D-02/03 to
v. 2503E-09/03**

1. Updated and changed "On-chip Debug System" to "JTAG Interface and On-chip Debug System" on page 35.
2. Updated Table 15 on page 37.
3. Updated "Test Access Port – TAP" on page 219 regarding the JTAGEN fuse.
4. Updated description for Bit 7 – JTD: JTAG Interface Disable on page 228.
5. Added a note regarding JTAGEN fuse to Table 104 on page 257.
6. Updated Absolute Maximum Ratings⁺, DC Characteristics and ADC Characteristics in "Electrical Characteristics" on page 287.
7. Added a proposal for solving problems regarding the JTAG Instruction IDCODE in "Errata" on page 336.

**anges from Rev.
03C-10/02 to
v. 2503D-02/03**

1. Added EEAR9 in EEARH in "Register Summary" on page 327.
2. Added Chip Erase as a first step in "Programming the Flash" on page 284 and "Programming the EEPROM" on page 285.
3. Removed reference to "Multi-purpose Oscillator" application note and "32 kHz Crystal Oscillator" application note, which do not exist.
4. Added information about PWM symmetry for Timer0 and Timer2.
5. Added note in "Filling the Temporary Buffer (Page Loading)" on page 251 about writing to the EEPROM during an SPM Page Load.
6. Added "Power Consumption" data in "Features" on page 1.
7. Added section "EEPROM Write During Power-down Sleep Mode" on page 22.
8. Added note about Differential Mode with Auto Triggering in "Prescaling and Conversion Timing" on page 204.
9. Updated Table 89 on page 232.
10. Added updated "Packaging Information" on page 333.

**anges from Rev.
03B-10/02 to
v. 2503C-10/02**

1. Updated the "DC Characteristics" on page 287.

**anges from Rev.
03A-03/02 to
v. 2503B-10/02**

1. Changed the endurance on the Flash to 10,000 Write/Erase Cycles.
2. Bit nr.4 – ADHSM – In SFIOR Register removed.
3. Added the section "Default Clock Source" on page 25.
4. When using External Clock there are some limitations regards to change of frequency. This is described in "External Clock" on page 31 and Table 117 on page 289.



5. Added a sub section regarding OCD-system and power consumption in the section "Minimizing Power Consumption" on page 34.
6. Corrected typo (WGM-bit setting) for:
 - "Fast PWM Mode" on page 75 (Timer/Counter0)
 - "Phase Correct PWM Mode" on page 76 (Timer/Counter0)
 - "Fast PWM Mode" on page 120 (Timer/Counter2)
 - "Phase Correct PWM Mode" on page 121 (Timer/Counter2)
7. Corrected Table 67 on page 164 (USART).
8. Updated V_{IL} , I_{IL} , and I_{IH} parameter in "DC Characteristics" on page 287.
9. Updated Description of OSCCAL Calibration Byte.

In the datasheet, it was not explained how to take advantage of the calibration bytes for 2, 4, and 8 MHz Oscillator selections. This is now added in the following sections:
Improved description of "Oscillator Calibration Register – OSCCAL" on page 30 and "Calibration Byte" on page 258.
10. Corrected typo in Table 42.
11. Corrected description in Table 45 and Table 46.
12. Updated Table 118, Table 120, and Table 121.
13. Added "Errata" on page 336.



Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 1-5 & 16, 10/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan
9F, Tonetex Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
JAPAN
Tel: (+81)(3) 3523-3551
Fax: (+81)(3) 3523-7581

© 2011 Atmel Corporation. All rights reserved. / Rev. CORP0XXXX

Atmel®, Atmel logo and combinations thereof, AVR® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

2N3903, 2N3904

2N3903 is a Preferred Device

General Purpose Transistors

NPN Silicon



ON Semiconductor™

<http://onsemi.com>

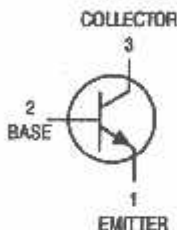
MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Collector-Emitter Voltage	V_{CEO}	40	Vdc
Collector-Base Voltage	V_{CBO}	60	Vdc
Emitter-Base Voltage	V_{EBO}	6.0	Vdc
Collector Current – Continuous	I_C	200	mA dc
Total Device Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	625 5.0	mW mW/ $^\circ\text{C}$
Total Device Dissipation @ $T_C = 25^\circ\text{C}$ Derate above 25°C	P_D	1.5 12	Watts mW/ $^\circ\text{C}$
Operating and Storage Junction Temperature Range	T_J, T_{stg}	-55 to +150	$^\circ\text{C}$

THERMAL CHARACTERISTICS (Note 1.)

Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction to Ambient	$R_{\theta JA}$	200	$^\circ\text{C/W}$
Thermal Resistance, Junction to Case	$R_{\theta JC}$	83.3	$^\circ\text{C/W}$

1. Indicates Data in addition to JEDEC Requirements.

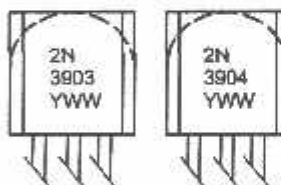


STYLE 1



TO-92
CASE 29
STYLE 1

MARKING DIAGRAMS



Y = Year
WW = Work Week

ORDERING INFORMATION

Device	Package	Shipping
2N3903	TO-92	5000 Units/Box
2N3903RLRM	TO-92	2000/Ammo Pack
2N3904	TO-92	5000 Units/Box
2N3904RLRA	TO-92	2000/Tape & Reel
2N3904RLRE	TO-92	2000/Tape & Reel
2N3904RLRM	TO-92	2000/Ammo Pack
2N3904RLRP	TO-92	2000/Ammo Pack
2N3904RL1	TO-92	2000/Tape & Reel
2N3904ZL1	TO-92	2000/Ammo Pack

Preferred devices are recommended choices for future use and best overall value.

2N3903, 2N3904

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
----------------	--------	-----	-----	------

OFF CHARACTERISTICS

Collector-Emitter Breakdown Voltage (Note 2.) ($I_C = 1.0\text{ mAdc}$, $I_B = 0$)	$V_{(BR)CEO}$	40	—	Vdc
Collector-Base Breakdown Voltage ($I_C = 10\text{ }\mu\text{Adc}$, $I_E = 0$)	$V_{(BR)CBO}$	60	—	Vdc
Emitter-Base Breakdown Voltage ($I_E = 10\text{ }\mu\text{Adc}$, $I_C = 0$)	$V_{(BR)EBO}$	6.0	—	Vdc
Base Cutoff Current ($V_{CE} = 30\text{ Vdc}$, $V_{EB} = 3.0\text{ Vdc}$)	I_{BL}	—	50	nAdc
Collector Cutoff Current ($V_{CE} = 30\text{ Vdc}$, $V_{EB} = 3.0\text{ Vdc}$)	I_{CEX}	—	50	nAdc

ON CHARACTERISTICS

DC Current Gain (Note 2.) ($I_C = 0.1\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$)	2N3903	h_{FE}	20	—	—
	2N3904		40	—	
($I_C = 1.0\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$)	2N3903		35	—	
	2N3904		70	—	
($I_C = 10\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$)	2N3903		50	150	
	2N3904		100	300	
($I_C = 50\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$)	2N3903		30	—	
	2N3904		60	—	
($I_C = 100\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$)	2N3903		15	—	
	2N3904		30	—	
Collector-Emitter Saturation Voltage (Note 2.) ($I_C = 10\text{ mAdc}$, $I_B = 1.0\text{ mAdc}$) ($I_C = 50\text{ mAdc}$, $I_B = 5.0\text{ mAdc}$)		$V_{CE(sat)}$	— —	0.2 0.3	Vdc
Base-Emitter Saturation Voltage (Note 2.) ($I_C = 10\text{ mAdc}$, $I_B = 1.0\text{ mAdc}$) ($I_C = 50\text{ mAdc}$, $I_B = 5.0\text{ mAdc}$)		$V_{BE(sat)}$	0.65 —	0.85 0.95	Vdc

SMALL-SIGNAL CHARACTERISTICS

Current-Gain – Bandwidth Product ($I_C = 10\text{ mAdc}$, $V_{CE} = 20\text{ Vdc}$, $f = 100\text{ MHz}$)	2N3903 2N3904	f_T	250 300	— —	MHz
Output Capacitance ($V_{CB} = 5.0\text{ Vdc}$, $I_E = 0$, $f = 1.0\text{ MHz}$)		C_{obo}	—	4.0	pF
Input Capacitance ($V_{EB} = 0.5\text{ Vdc}$, $I_C = 0$, $f = 1.0\text{ MHz}$)		C_{ibo}	—	8.0	pF
Input Impedance ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	2N3903 2N3904	h_{ie}	1.0 1.0	8.0 10	k Ω
Voltage Feedback Ratio ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	2N3903 2N3904	h_{re}	0.1 0.5	5.0 8.0	$\times 10^{-4}$
Small-Signal Current Gain ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	2N3903 2N3904	h_{fe}	50 100	200 400	—
Output Admittance ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)		h_{oe}	1.0	40	μmhos
Noise Figure ($I_C = 100\text{ }\mu\text{Adc}$, $V_{CE} = 5.0\text{ Vdc}$, $R_S = 1.0\text{ k}\Omega$, $f = 1.0\text{ kHz}$)	2N3903 2N3904	NF	— —	6.0 5.0	dB

SWITCHING CHARACTERISTICS

Delay Time	$(V_{CC} = 3.0\text{ Vdc}$, $V_{BE} = 0.5\text{ Vdc}$, $I_C = 10\text{ mAdc}$, $I_{B1} = 1.0\text{ mAdc}$)	t_d	—	35	ns
Rise Time		t_r	—	35	ns
Storage Time	$(V_{CC} = 3.0\text{ Vdc}$, $I_C = 10\text{ mAdc}$, $I_{B1} = I_{B2} = 1.0\text{ mAdc}$)	t_s	—	175 200	ns
Fall Time		t_f	—	50	ns

2. Pulse Test: Pulse Width $\leq 300\text{ }\mu\text{s}$; Duty Cycle $\leq 2\%$.

2N3903, 2N3904

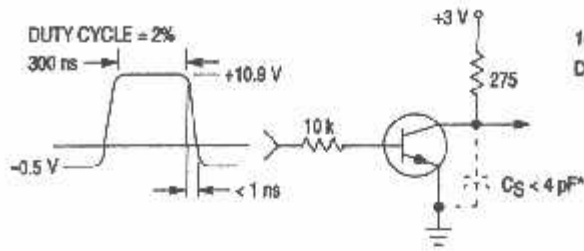


Figure 1. Delay and Rise Time
Equivalent Test Circuit

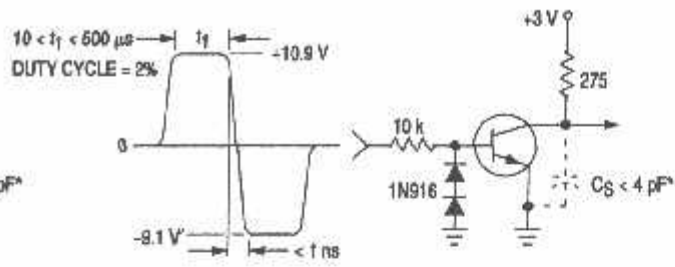


Figure 2. Storage and Fall Time
Equivalent Test Circuit

* Total shunt capacitance of test jig and connectors

TYPICAL TRANSIENT CHARACTERISTICS

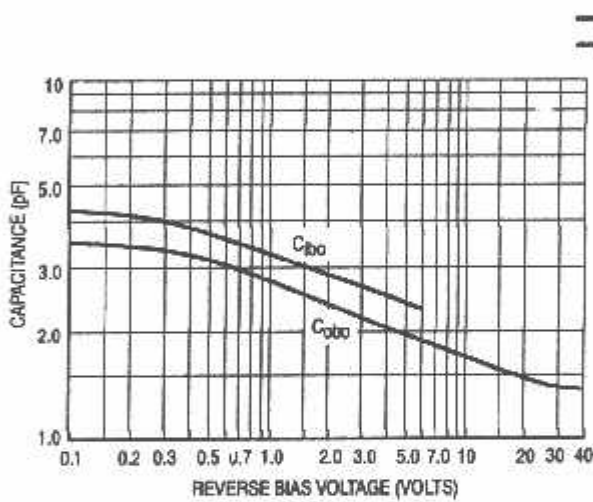


Figure 3. Capacitance

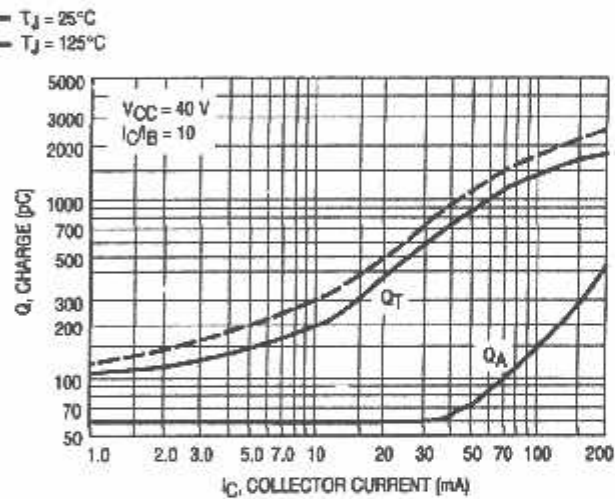


Figure 4. Charge Data

2N3903, 2N3904

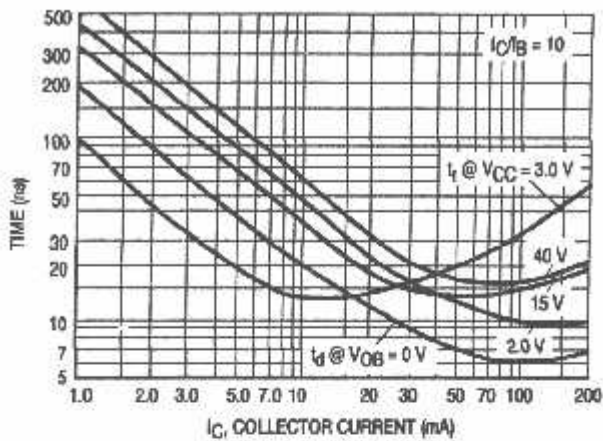


Figure 5. Turn-On Time

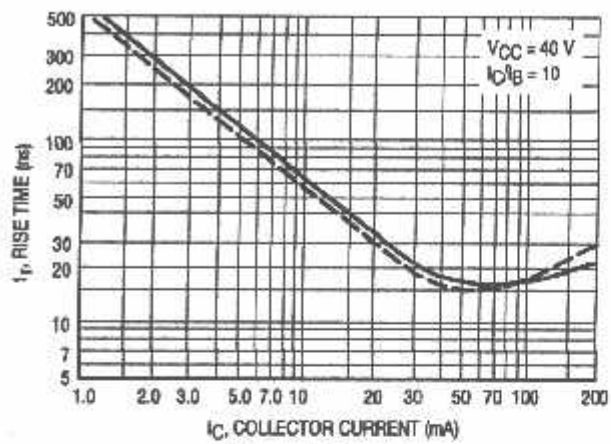


Figure 6. Rise Time

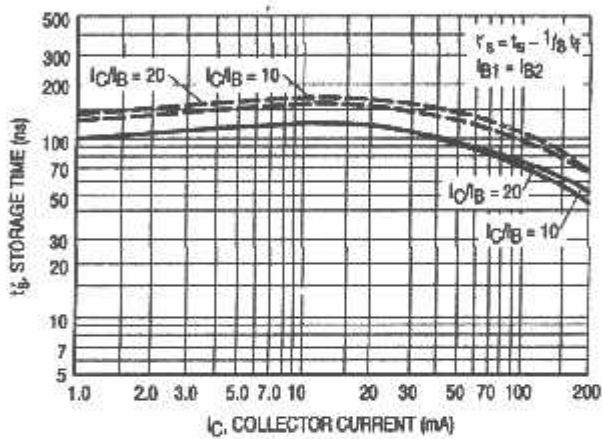


Figure 7. Storage Time

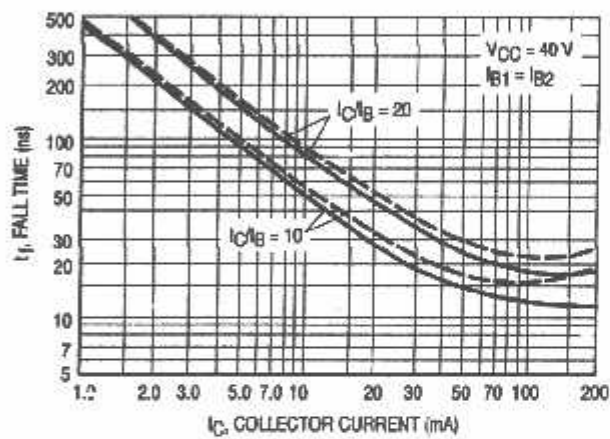


Figure 8. Fall Time

TYPICAL AUDIO SMALL-SIGNAL CHARACTERISTICS NOISE FIGURE VARIATIONS

($V_{CE} = 5.0$ Vdc, $T_A = 25^\circ\text{C}$, Bandwidth = 1.0 Hz)

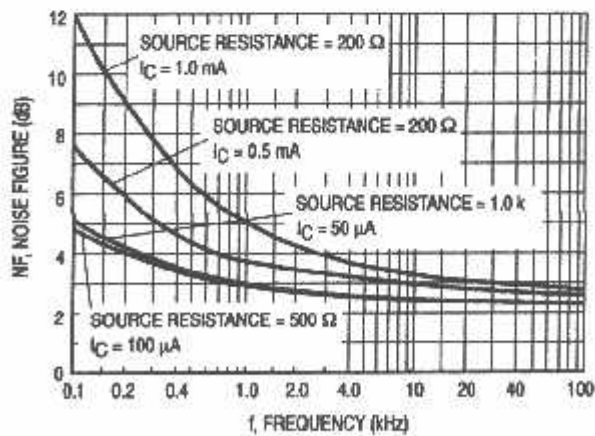


Figure 9.

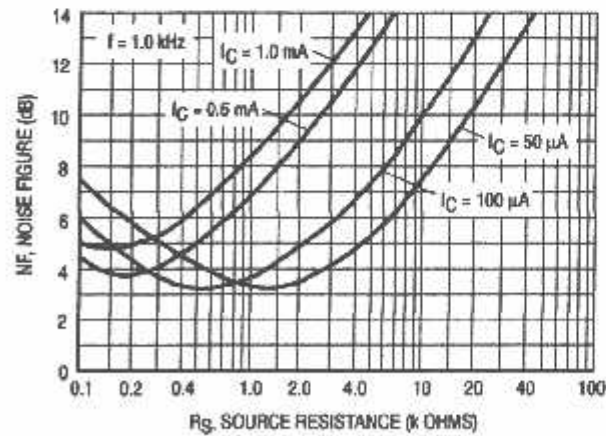


Figure 10.

2N3903, 2N3904

h PARAMETERS

($V_{CE} = 10 \text{ Vdc}$, $f = 1.0 \text{ kHz}$, $T_A = 25^\circ\text{C}$)

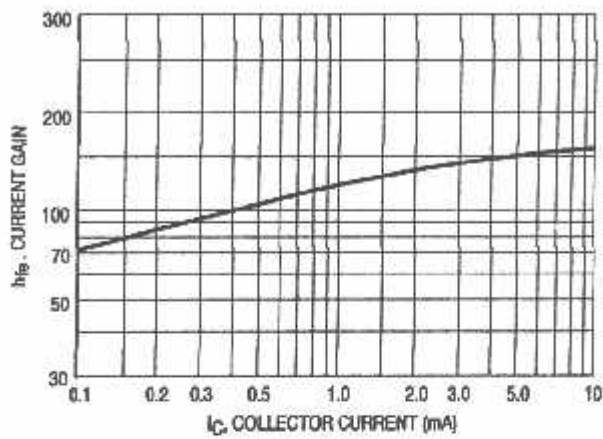


Figure 11. Current Gain

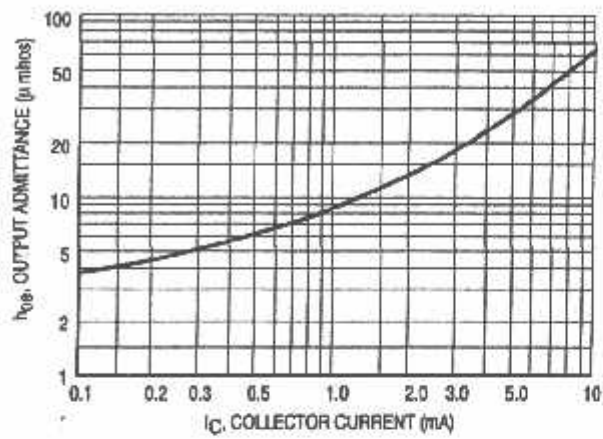


Figure 12. Output Admittance

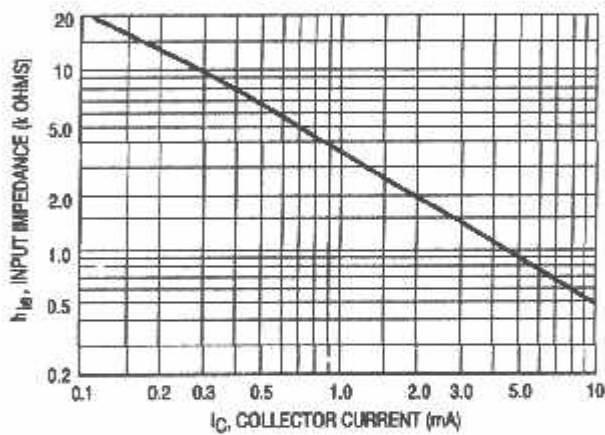


Figure 13. Input Impedance

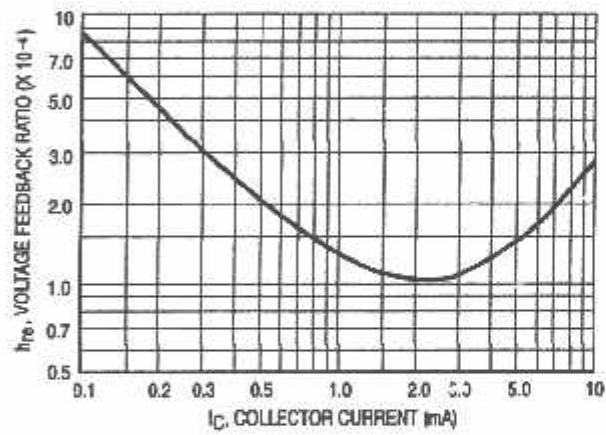


Figure 14. Voltage Feedback Ratio

2N3903, 2N3904

TYPICAL STATIC CHARACTERISTICS

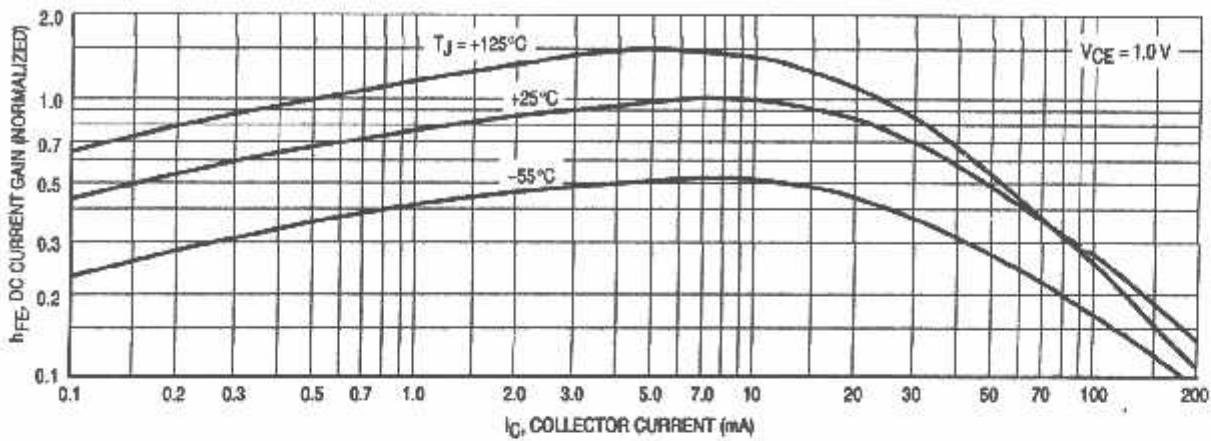


Figure 15. DC Current Gain

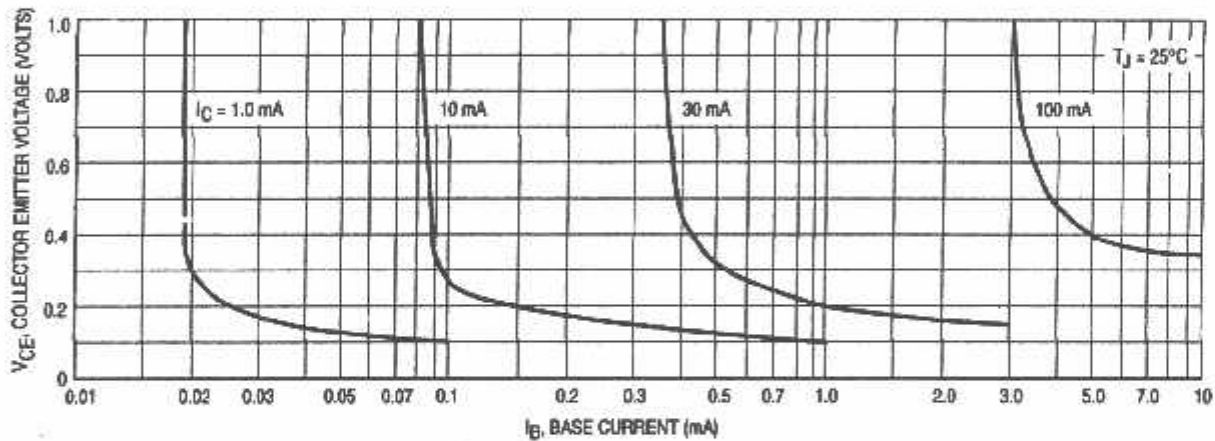


Figure 16. Collector Saturation Region

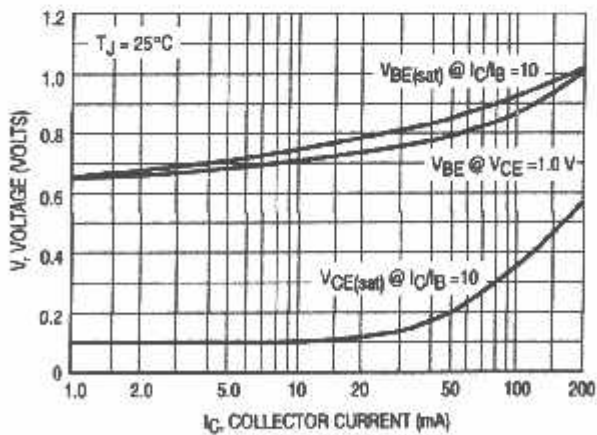


Figure 17. "ON" Voltages

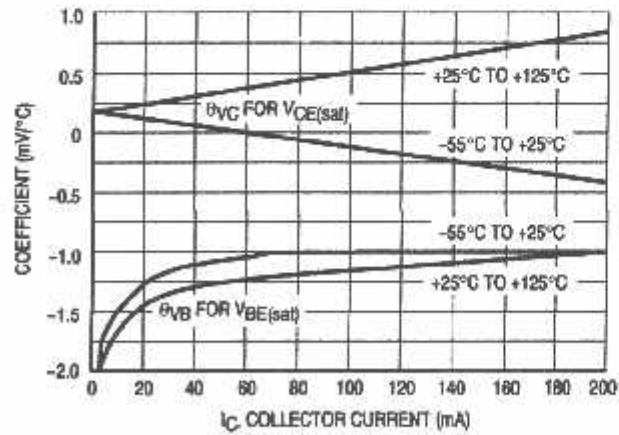
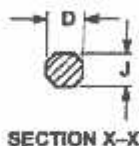
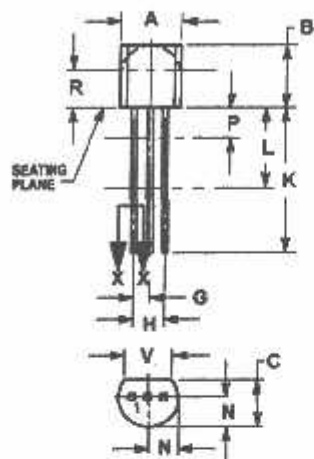


Figure 18. Temperature Coefficients

2N3903, 2N3904

PACKAGE DIMENSIONS

TO-92
TO-226AA
CASE 29-11
ISSUE AL



NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. CONTOUR OF PACKAGE BEYOND DIMENSION R IS UNCONTROLLED.
4. LEAD DIMENSION IS UNCONTROLLED IN P AND BEYOND DIMENSION K MINIMUM.

DIM.	INCHES		MILLIMETERS	
	MIN.	MAX.	MIN.	MAX.
A	0.175	0.205	4.45	5.20
B	0.170	0.210	4.32	5.33
C	0.125	0.165	3.18	4.19
D	0.018	0.021	0.457	0.533
E	0.045	0.055	1.15	1.39
H	0.095	0.105	2.42	2.66
J	0.015	0.020	0.38	0.52
K	0.000	---	12.70	---
L	0.250	---	6.35	---
N	0.000	0.105	2.54	2.66
P	---	0.100	---	2.54
R	0.115	---	2.93	---
V	0.135	---	3.43	---

STYLE 1:

- PIN 1: EMITTER
2: BASE
3: COLLECTOR

STYLE 14:

- PIN 1: EMITTER
2: COLLECTOR
3: BASE

2N3903, 2N3904

ON Semiconductor and  are trademarks of Semiconductor Components Industries, LLC (SCILLC). SCILLC reserves the right to make changes without further notice to any products herein. SCILLC makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does SCILLC assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. "Typical" parameters which may be provided in SCILLC data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. SCILLC does not convey any license under its patent rights nor the rights of others. SCILLC products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SCILLC product could create a situation where personal injury or death may occur. Should Buyer purchase or use SCILLC products for any such unintended or unauthorized application, Buyer shall indemnify and hold SCILLC and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that SCILLC was negligent regarding the design or manufacture of the part. SCILLC is an Equal Opportunity/Affirmative Action Employer.

PUBLICATION ORDERING INFORMATION

Literature Fulfillment:

Literature Distribution Center for ON Semiconductor
P.O. Box 5163, Denver, Colorado 80217 USA
Phone: 303-675-2175 or 800-344-3860 Toll Free USA/Canada
Fax: 303-675-2176 or 800-344-3867 Toll Free USA/Canada
Email: ONlit@nibberrco.com

N. American Technical Support: 800-282-9855 Toll Free USA/Canada

JAPAN: ON Semiconductor, Japan Customer Focus Center
4-32-1 Nishi-Gotanda, Shinagawa-ku, Tokyo, Japan 141-0031
Phone: 81-3-5740-2700
Email: r14525@onsemi.com

ON Semiconductor Website: <http://onsemi.com>

For additional information, please contact your local
Sales Representative.

2N3903/D



8-bit Atmel Microcontroller with 8KB In-System Programmable Flash

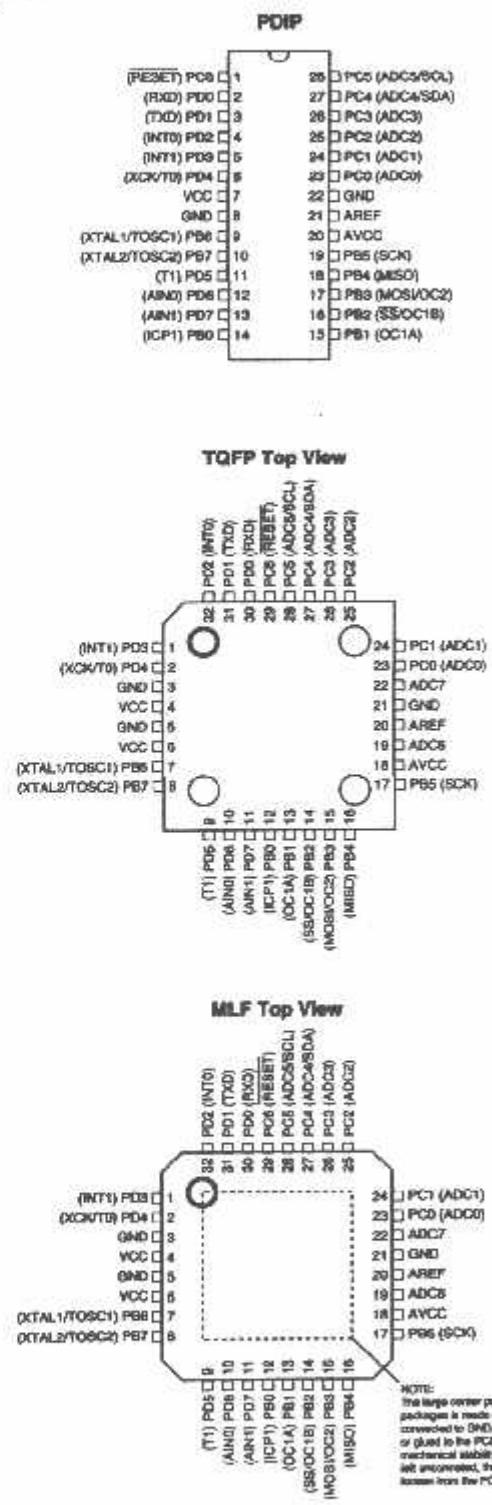
ATmega8A

Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 130 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16MIPS Throughput at 16MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 8KBytes of In-System Self-programmable Flash program memory
 - 512Bytes EEPROM
 - 1KByte Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Atmel QTouch® library support
 - Capacitive touch buttons, sliders and wheels
 - Atmel QTouch and QMatrix acquisition
 - Up to 64 sense channels
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Three PWM Channels
 - 8-channel ADC in TQFP and QFN/MLF package
 - Eight Channels 10-bit Accuracy
 - 6-channel ADC in PDIP package
 - Six Channels 10-bit Accuracy
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF
- Operating Voltages
 - 2.7 - 5.5V
 - 0 - 16MHz
- Power Consumption at 4MHz, 3V, 25°C
 - Active: 3.6mA
 - Idle Mode: 1.0mA
 - Power-down Mode: 0.5µA

Pin Configurations

Figure 1-1. Pinout ATmega8A

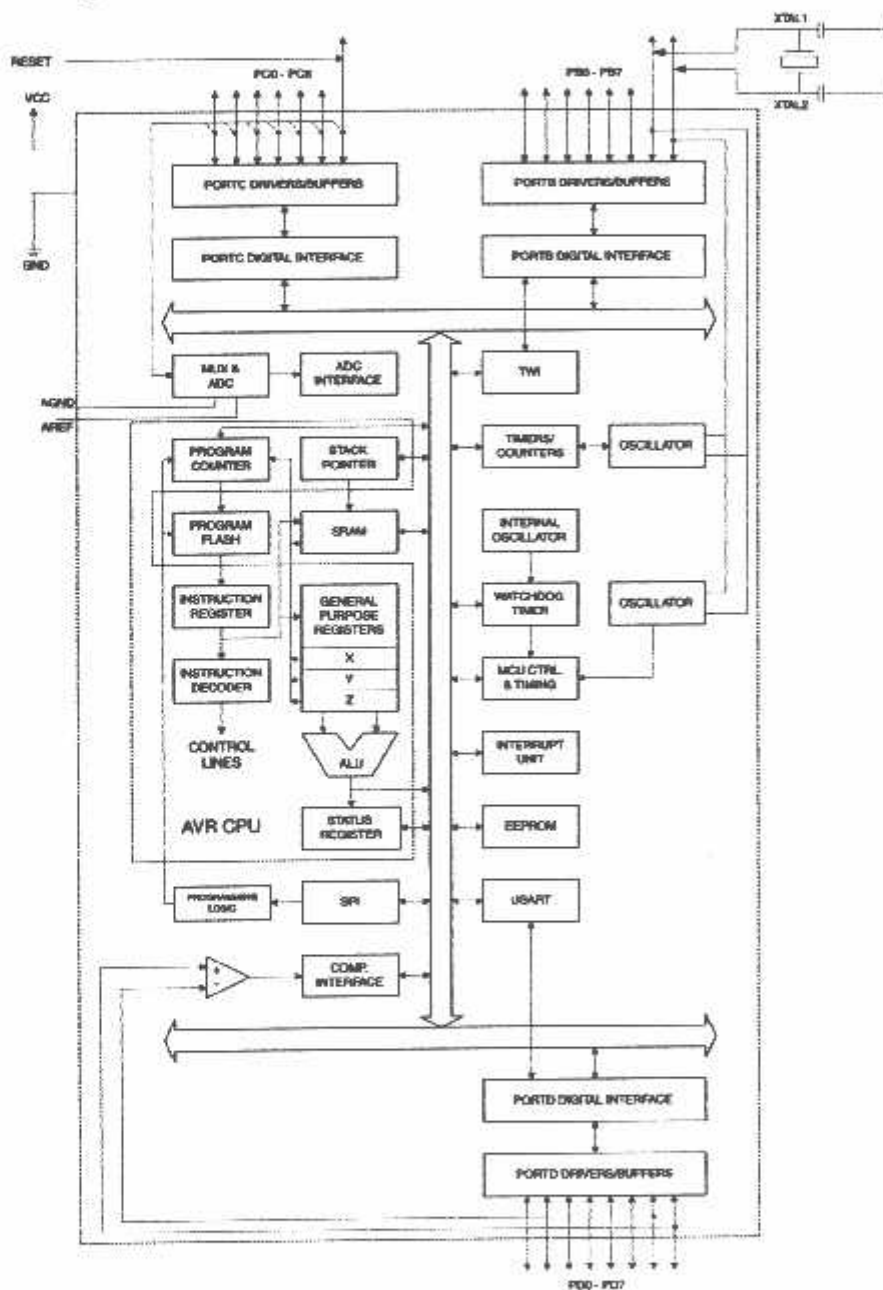


Overview

The Atmel®AVR® ATmega8A is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8A achieves throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2-1. Block Diagram



The Atmel®AVR® AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega8A provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes of EEPROM, 1K byte of SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, a 6-channel ADC (eight channels in TQFP and QFN/MLF packages) with 10-bit accuracy, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next Interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The Flash Program memory can be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash Section will continue to run while the Application Flash Section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega8A is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The Atmel AVR ATmega8A is supported with a full suite of program and system development tools, including C compilers, macro assemblers, program simulators and evaluation kits.

Pin Descriptions

VCC

Digital supply voltage.

GND

Ground.

Port B (PB7:PB0) – XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7:6 is used as TOSC2:1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 56 and "System Clock and Clock Options" on page 24.

Port C (PC5:PC0)

Port C is an 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 26-3 on page 228. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated on page 59.

Port D (PD7:PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega8A as listed on page 61.

RESET

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 26-3 on page 228. Shorter pulses are not guaranteed to generate a reset.

AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, Port C (3:0), and ADC (7:6). It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that Port C (5:4) use digital supply voltage, V_{CC}.

AREF

AREF is the analog reference pin for the A/D Converter.

ADC7:6 (TQFP and QFN/MLF Package Only)

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

Resources

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.

Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

About Code Examples

This datasheet contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

Capacitive touch sensing

The Atmel® QTouch® Library provides a simple to use solution to realize touch sensitive interfaces on most Atmel AVR® microcontrollers. The QTouch Library includes support for the QTouch and QMatrix® acquisition methods.

Touch sensing can be added to any application by linking the appropriate Atmel QTouch Library for the AVR Microcontroller. This is done by using a simple set of APIs to define the touch channels and sensors, and then calling the touch sensing API's to retrieve the channel information and determine the touch sensor states.

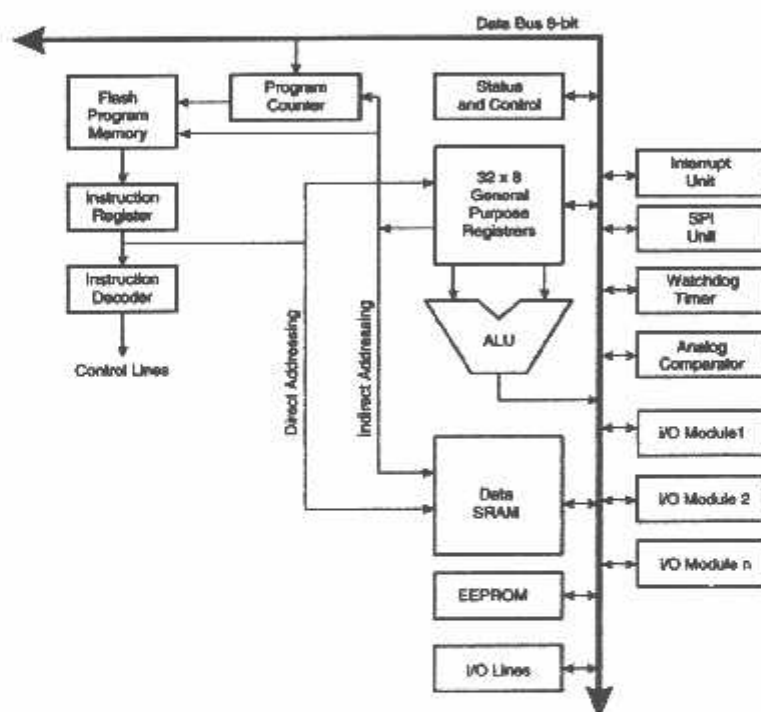
The QTouch Library is FREE and downloadable from the Atmel website at the following location: www.atmel.com/qtouchlibrary. For implementation details and other information, refer to the Atmel QTouch Library User Guide - also available for download from the Atmel website.

AVR CPU Core

Overview

This section discusses the Atmel® AVR® core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Figure 7-1. Block Diagram of the AVR MCU Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the Program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the Program memory. This concept enables instructions to be executed in every clock cycle. The Program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of these address pointers can also be used as an address pointer for look up tables in Flash Program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

The Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every Program memory address contains a 16- or 32-bit instruction.

Program Flash memory space is divided in two sections, the Boot program section and the Application program section. Both sections have dedicated Lock Bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The Stack Pointer SP is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the Status Register. All interrupts have a separate Interrupt Vector in the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, 0x20 - 0x5F.

Arithmetic Logic Unit – ALU

The high-performance Atmel®AVR® ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the "Instruction Set" section for a detailed description.

Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

SREG – The AVR Status Register

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the Instruction Set Reference.

- **Bit 6 – T: Bit Copy Storage**

The Bit Copy instructions BLD (Bit Load) and BST (Bit Store) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry is useful in BCD arithmetic. See the "Instruction Set Description" for detailed information.

- **Bit 4 – S: Sign Bit, $S = N \oplus V$**

The S-bit is always an exclusive or between the Negative Flag N and the Two's Complement Overflow Flag V. See the "Instruction Set Description" for detailed information.

- **Bit 3 – V: Two's Complement Overflow Flag**

The Two's Complement Overflow Flag V supports two's complement arithmetics. See the "Instruction Set Description" for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a Carry in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

General Purpose Register File

The Register File is optimized for the Atmel® AVR® Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input.
- Two 8-bit output operands and one 8-bit result input.
- Two 8-bit output operands and one 16-bit result input.
- One 16-bit output operand and one 16-bit result input.

Figure 7-2 shows the structure of the 32 general purpose working registers in the CPU.

Figure 7-2. AVR CPU General Purpose Working Registers

	7	0	Addr.	
General Purpose Working Registers	R0		0x00	
	R1		0x01	
	R2		0x02	
	...			
	R13		0x0D	
	R14		0x0E	
	R15		0x0F	
	R16		0x10	
	R17		0x11	
	...			
	R26		0x1A	X-register Low Byte
	R27		0x1B	X-register High Byte
	R28		0x1C	Y-register Low Byte
	R29		0x1D	Y-register High Byte
	R30		0x1E	Z-register Low Byte
	R31		0x1F	Z-register High Byte

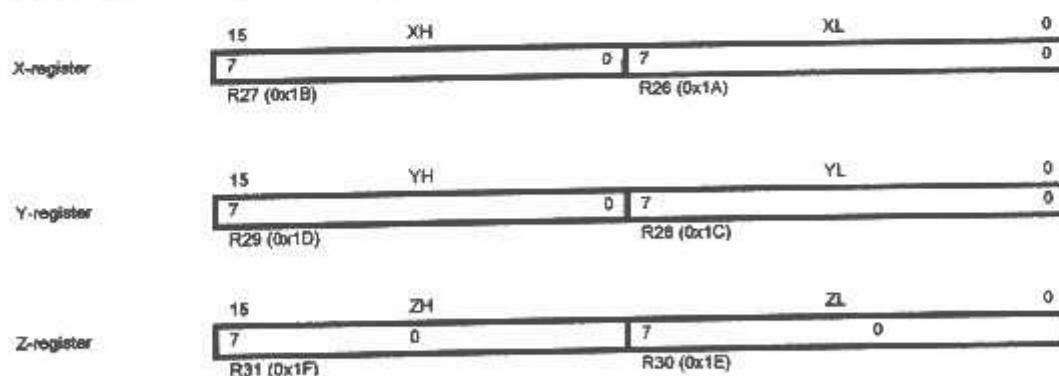
Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 7-2, each register is also assigned a Data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y-, and Z-pointer Registers can be set to index any register in the file.

The X-register, Y-register and Z-register

The registers R26:R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the Data Space. The three indirect address registers X, Y and Z are defined as described in Figure 7-3.

Figure 7-3. The X-, Y- and Z-Registers



In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the Instruction Set Reference for details).

Stack Pointer

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. Note that the Stack is implemented as growing from higher to lower memory

locations. The Stack Pointer Register always points to the top of the Stack. The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. A Stack PUSH command will decrease the Stack Pointer.

The Stack in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. Initial Stack Pointer value equals the last address of the internal SRAM and the Stack Pointer must be set to point above start of the SRAM, see Figure 8-2 on page 16.

See Table 7-1 for Stack Pointer details.

Table 7-1. Stack Pointer instructions

Instruction	Stack pointer	Description
PUSH	Decrement by 1	Data is pushed onto the stack
CALL ICALL RCALL	Decrement by 2	Return address is pushed onto the stack with a subroutine call or interrupt
POP	Increment by 1	Data is popped from the stack
RET RETI	Increment by 2	Return address is popped from the stack with return from subroutine or return from interrupt

The Atmel®AVR® Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

SPH and SPL – Stack Pointer High and Low Register

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Instruction Execution Timing

This section describes the general access timing concepts for instruction execution. The Atmel®AVR®CPU is driven by the CPU clock clk_{CPU} , directly generated from the selected clock source for the chip. No internal clock division is used.

Figure 7-4 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

Figure 7-4. The Parallel Instruction Fetches and Instruction Executions

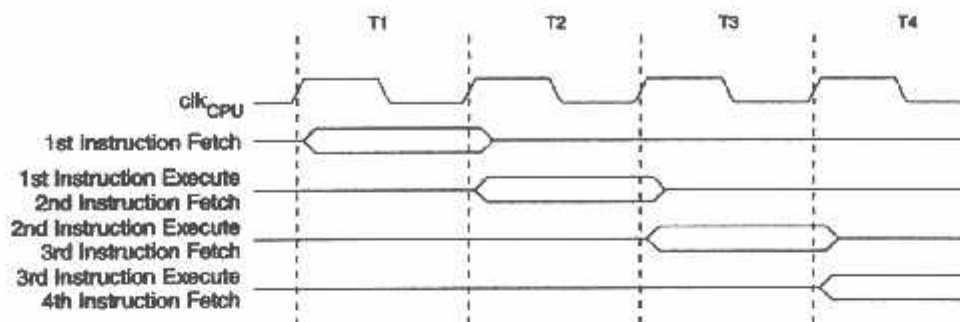
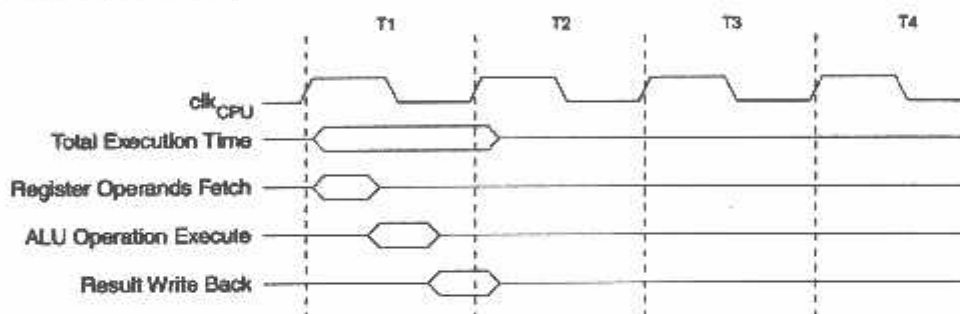


Figure 7-5 shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 7-5. Single Cycle ALU Operation



Reset and Interrupt Handling

The Atmel®AVR® provides several different interrupt sources. These interrupts and the separate Reset Vector each have a separate Program Vector in the Program memory space. All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt. Depending on the Program Counter value, interrupts may be automatically disabled when Boot Lock Bits BLB02 or BLB12 are programmed. This feature improves software security. See the section “Memory Programming” on page 207 for details.

The lowest addresses in the Program memory space are by default defined as the Reset and Interrupt Vectors. The complete list of Vectors is shown in “Interrupts” on page 44. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INTO – the External Interrupt Request 0. The Interrupt Vectors can be moved to the start of the boot Flash section by setting the Interrupt Vector Select (IVSEL) bit in the General Interrupt Control Register (GICR). Refer to “Interrupts” on page 44 for more information. The Reset Vector can also be moved to the start of the boot Flash section by programming the BOOTRST Fuse, see “Boot Loader Support – Read-While-Write Self-Programming” on page 194.

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared and all interrupts are disabled. The user software can write logic one to the I-bit to enable nested interrupts. All enabled interrupts can then interrupt the current interrupt routine. The I-bit is automatically set when a Return from Interrupt instruction – RETI – is executed.

There are basically two types of interrupts. The first type is triggered by an event that sets the Interrupt Flag. For these interrupts, the Program Counter is vectored to the actual Interrupt Vector in order to execute the interrupt

handling routine, and hardware clears the corresponding Interrupt Flag. Interrupt Flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared. If an interrupt condition occurs while the corresponding interrupt enable bit is cleared, the Interrupt Flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software. Similarly, if one or more interrupt conditions occur while the global interrupt enable bit is cleared, the corresponding Interrupt Flag(s) will be set and remembered until the global interrupt enable bit is set, and will then be executed by order of priority.

The second type of interrupts will trigger as long as the interrupt condition is present. These interrupts do not necessarily have Interrupt Flags. If the interrupt condition disappears before the interrupt is enabled, the interrupt will not be triggered.

When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note that the Status Register is not automatically stored when entering an interrupt routine, nor restored when returning from an interrupt routine. This must be handled by software.

When using the CLI instruction to disable interrupts, the interrupts will be immediately disabled. No interrupt will be executed after the CLI instruction, even if it occurs simultaneously with the CLI instruction. The following example shows how this can be used to avoid interrupts during the timed EEPROM write sequence.

Assembly Code Example
<pre> in r16, SREG ; store SREG value cli ; disable interrupts during timed sequence sbi EECR, EEMWE ; start EEPROM write sbi EECR, EWE out SREG, r16 ; restore SREG value (I-bit) </pre>
C Code Example
<pre> char cSREG; cSREG = SREG; /* store SREG value */ /* disable interrupts during timed sequence */ _cli(); EECR = (1<<EEMWE); /* start EEPROM write */ EECR = (1<<EWE); SREG = cSREG; /* restore SREG value (I-bit) */ </pre>

When using the SEI instruction to enable interrupts, the instruction following SEI will be executed before any pending interrupts, as shown in the following example.

Assembly Code Example
<pre> sei ; set global interrupt enable sleep; enter sleep, waiting for interrupt ; note: will enter sleep before any pending ; interrupt(s) </pre>
C Code Example
<pre> _sei(); /* set global interrupt enable */ _sleep(); /* enter sleep, waiting for interrupt */ /* note: will enter sleep before any pending interrupt(s) */ </pre>

1 Interrupt Response Time

The interrupt execution response for all the enabled Atmel®AVR® interrupts is four clock cycles minimum. After four clock cycles, the Program Vector address for the actual interrupt handling routine is executed. During this 4-clock cycle period, the Program Counter is pushed onto the Stack. The Vector is normally a jump to the interrupt routine, and this jump takes three clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served. If an interrupt occurs when the MCU is in sleep mode, the interrupt execution response time is increased by four clock cycles. This increase comes in addition to the start-up time from the selected sleep mode.

A return from an interrupt handling routine takes four clock cycles. During these four clock cycles, the Program Counter (2 bytes) is popped back from the Stack, the Stack Pointer is incremented by 2, and the I-bit in SREG is set.

AVR Memories

Overview

This section describes the different memories in the Atmel®AVR® ATmega8A. The AVR architecture has two main memory spaces, the Data memory and the Program Memory space. In addition, the ATmega8A features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

In-System Reprogrammable Flash Program Memory

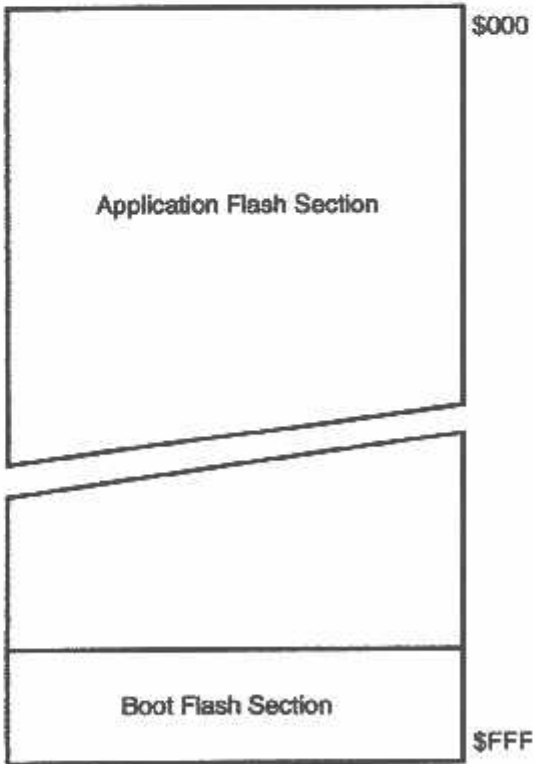
The ATmega8A contains 8K bytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16- or 32-bits wide, the Flash is organized as 4K x 16 bits. For software security, the Flash Program memory space is divided into two sections, Boot Program section and Application Program section.

The Flash memory has an endurance of at least 10,000 write/erase cycles. The ATmega8A Program Counter (PC) is 12 bits wide, thus addressing the 4K Program memory locations. The operation of Boot Program section and associated Boot Lock Bits for software protection are described in detail in "Boot Loader Support – Read-While-Write Self-Programming" on page 194. "Memory Programming" on page 207 contains a detailed description on Flash Programming in SPI- or Parallel Programming mode.

Constant tables can be allocated within the entire Program memory address space (see the LPM – Load Program memory instruction description).

Timing diagrams for instruction fetch and execution are presented in "Instruction Execution Timing" on page 11.

Figure 8-1. Program Memory Map



SRAM Data Memory

Figure 8-2 shows how the Atmel®AVR®ATmega8A SRAM Memory is organized.

The lower 1120 Data memory locations address the Register File, the I/O Memory, and the internal data SRAM. The first 96 locations address the Register File and I/O Memory, and the next 1024 locations address the internal data SRAM.

The five different addressing modes for the Data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement, and Indirect with Post-increment. In the Register File, registers R26 to R31 feature the indirect addressing pointer registers.

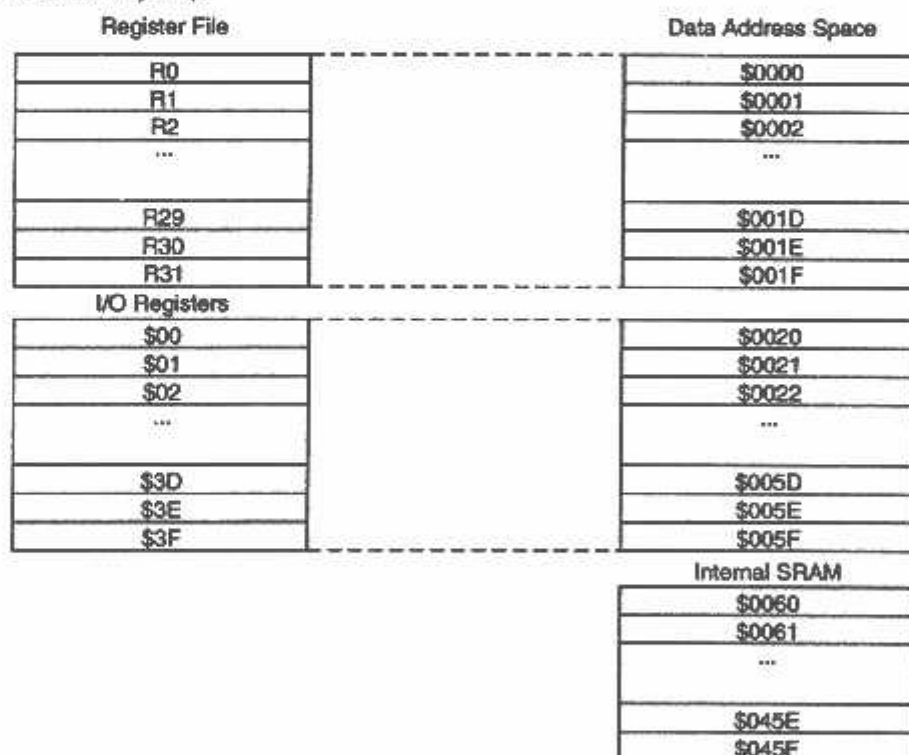
The direct addressing reaches the entire data space.

The Indirect with Displacement mode reaches 63 address locations from the base address given by the Y- or Z-register.

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y and Z are decremented or incremented.

The 32 general purpose working registers, 64 I/O Registers, and the 1024 bytes of Internal data SRAM in the ATmega8A are all accessible through all these addressing modes. The Register File is described in "General Purpose Register File" on page 9.

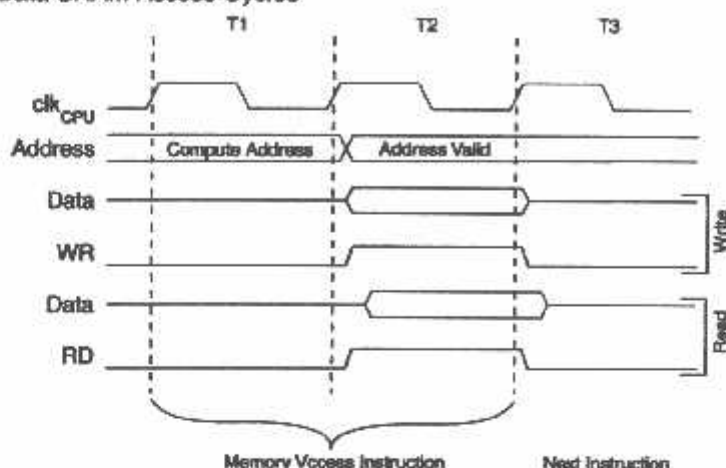
Figure 8-2. Data Memory Map



Data Memory Access Times

This section describes the general access timing concepts for internal memory access. The internal data SRAM access is performed in two clk_{CPU} cycles as described in Figure 8-3.

Figure 8-3. On-chip Data SRAM Access Cycles



EEPROM Data Memory

The Atmel®AVR®ATmega8A contains 512 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described below, specifying the EEPROM Address Registers, the EEPROM Data Register, and the EEPROM Control Register.

"Memory Programming" on page 207 contains a detailed description on EEPROM Programming in SPI- or Parallel Programming mode.

EEPROM Read/Write Access

The EEPROM Access Registers are accessible in the I/O space.

The write access time for the EEPROM is given in Table 8-4 on page 20. A self-timing function, however, lets the user software detect when the next byte can be written. If the user code contains instructions that write the EEPROM, some precautions must be taken. In heavily filtered power supplies, V_{CC} is likely to rise or fall slowly on Power-up/down. This causes the device for some period of time to run at a voltage lower than specified as minimum for the clock frequency used. See "Preventing EEPROM Corruption" on page 22, for details on how to avoid problems in these situations.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to the description of the EEPROM Control Register for details on this.

When the EEPROM is read, the CPU is halted for four clock cycles before the next instruction is executed. When the EEPROM is written, the CPU is halted for two clock cycles before the next instruction is executed.

I/O Memory

The I/O space definition of the ATmega8A is shown in  on page 301.

All ATmega8A I/Os and peripherals are placed in the I/O space. The I/O locations are accessed by the IN and OUT instructions, transferring data between the 32 general purpose working registers and the I/O space. I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI Instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC Instructions. Refer to the instruction set section for more details. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must

be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

The I/O and Peripherals Control Registers are explained in later sections.

Register Description

EEARH and EEARL – The EEPROM Address Register

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	–	EEAR8	EEARH
	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	X	
	X	X	X	X	X	X	X	X	

- **Bits 15:9 – Res: Reserved Bits**

These bits are reserved bits in the ATmega8A and will always read as zero.

- **Bits 8:0 – EEAR8:0: EEPROM Address**

The EEPROM Address Registers – EEARH and EEARL – specify the EEPROM address in the 512 bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 511. The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

EEDR – The EEPROM Data Register

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:0 – EEDR7:0: EEPROM Data**

For the EEPROM write operation, the EEDR Register contains the data to be written to the EEPROM in the address given by the EEAR Register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

EECR – The EEPROM Control Register

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	EERE	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	X	0	

- **Bits 7:4 – Res: Reserved Bits**

These bits are reserved bits in the ATmega8A and will always read as zero.

- **Bit 3 – EERIE: EEPROM Ready Interrupt Enable**

Writing EERIE to one enables the EEPROM Ready Interrupt if the I bit in SREG is set. Writing EERIE to zero disables the interrupt. The EEPROM Ready interrupt generates a constant interrupt when EEWL is cleared.

- **Bit 2 – EEMWE: EEPROM Master Write Enable**

The EEMWE bit determines whether setting EEWL to one causes the EEPROM to be written. When EEMWE is set, setting EEWL within four clock cycles will write data to the EEPROM at the selected address. If EEMWE is zero, setting EEWL will have no effect. When EEMWE has been written to one by software, hardware clears the bit to zero after four clock cycles. See the description of the EEWL bit for an EEPROM write procedure.

- **Bit 1 – EEWL: EEPROM Write Enable**

The EEPROM Write Enable Signal EEWL is the write strobe to the EEPROM. When address and data are correctly set up, the EEWL bit must be written to one to write the value into the EEPROM. The EEMWE bit must be written to one before a logical one is written to EEWL, otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (the order of steps 3 and 4 is not essential):

1. Wait until EEWL becomes zero.
2. Wait until SPMEN in SPMCR becomes zero.
3. Write new EEPROM address to EEAR (optional).
4. Write new EEPROM data to EEDR (optional).
5. Write a logical one to the EEMWE bit while writing a zero to EEWL in EECR.
6. Within four clock cycles after setting EEMWE, write a logical one to EEWL.

The EEPROM can not be programmed during a CPU write to the Flash memory. The software must check that the Flash programming is completed before initiating a new EEPROM write. Step 2 is only relevant if the software contains a boot loader allowing the CPU to program the Flash. If the Flash is never being updated by the CPU, step 2 can be omitted. See "Boot Loader Support – Read-While-Write Self-Programming" on page 194 for details about boot programming.

Caution: An interrupt between step 5 and step 6 will make the write cycle fail, since the EEPROM Master Write Enable will time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEAR or EEDR Register will be modified, causing the interrupted EEPROM access to fail. It is recommended to have the Global Interrupt Flag cleared during all the steps to avoid these problems.

When the write access time has elapsed, the EEWL bit is cleared by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEWL has been set, the CPU is halted for two cycles before the next instruction is executed.

- **Bit 0 – EERE: EEPROM Read Enable**

The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in the EEAR Register, the EERE bit must be written to a logic one to trigger the EEPROM read. The EEPROM read access takes one instruction, and the requested data is available immediately. When the EEPROM is read, the CPU is halted for four cycles before the next instruction is executed.

The user should poll the EEWL bit before starting the read operation. If a write operation is in progress, it is neither possible to read the EEPROM, nor to change the EEAR Register.

The following code examples show one assembly and one C function for writing to the EEPROM. The examples assume that interrupts are controlled (for example by disabling interrupts globally) so that no interrupts will occur during execution of these functions. The examples also assume that no Flash boot loader is present in the software. If such code is present, the EEPROM write function must also wait for any ongoing SPM command to finish.

Assembly Code Example

```
EEPROM_write:
    ; Wait for completion of previous write
    sbic EECR, EWE
    rjmp EEPROM_write
    ; Set up address (r18:r17) in address register
    out EEARH, r18
    out EEARL, r17
    ; Write data (r16) to data register
    out EEDR, r16
    ; Write logical one to EEMWE
    sbi EECR, EEMWE
    ; Start eeprom write by setting EWE
    sbi EECR, EWE
    ret
```

C Code Example

```
void EEPROM_write(unsigned int uiAddress, unsigned char ucData)
{
    /* Wait for completion of previous write */
    while(EECR & (1<<EWE))
    {
        ;
    }
    /* Set up address and data registers */
    EEAR = uiAddress;
    EEDR = ucData;
    /* Write logical one to EEMWE */
    EECR |= (1<<EEMWE);
    /* Start eeprom write by setting EWE */
    EECR |= (1<<EWE);
}
```

The next code examples show assembly and C functions for reading the EEPROM. The examples assume that interrupts are controlled so that no interrupts will occur during execution of these functions.

Assembly Code Example

```
EEPROM_read:
; Wait for completion of previous write
sbic EECR,EWE
rjmp EEPROM_read
; Set up address (r18:r17) in address register
out EEARH, r18
out EEARL, r17
; Start eeprom read by writing EERE
sbi EECR,EERE
; Read data from data register
in r16,EEDR
ret
```

C Code Example

```
unsigned char EEPROM_read(unsigned int uiAddress)
{
    /* Wait for completion of previous write */
    while(EECR & (1<<EWE))
    ;
    /* Set up address register */
    EEAR = uiAddress;
    /* Start eeprom read by writing EERE */
    EECR |= (1<<EERE);
    /* Return data from data register */
    return EEDR;
}
```

EEPROM Write during Power-down Sleep Mode

When entering Power-down sleep mode while an EEPROM write operation is active, the EEPROM write operation will continue, and will complete before the Write Access time has passed. However, when the write operation is completed, the Oscillator continues running, and as a consequence, the device does not enter Power-down entirely. It is therefore recommended to verify that the EEPROM write operation is completed before entering Power-down.

Preventing EEPROM Corruption

During periods of low V_{CC} , the EEPROM data can be corrupted because the supply voltage is too low for the CPU and the EEPROM to operate properly. These issues are the same as for board level systems using EEPROM, and the same design solutions should be applied.

An EEPROM data corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the EEPROM requires a minimum voltage to operate correctly. Second, the CPU itself can execute instructions incorrectly, if the supply voltage is too low.

EEPROM data corruption can easily be avoided by following this design recommendation:

Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD). If the detection level of the internal BOD does not match the

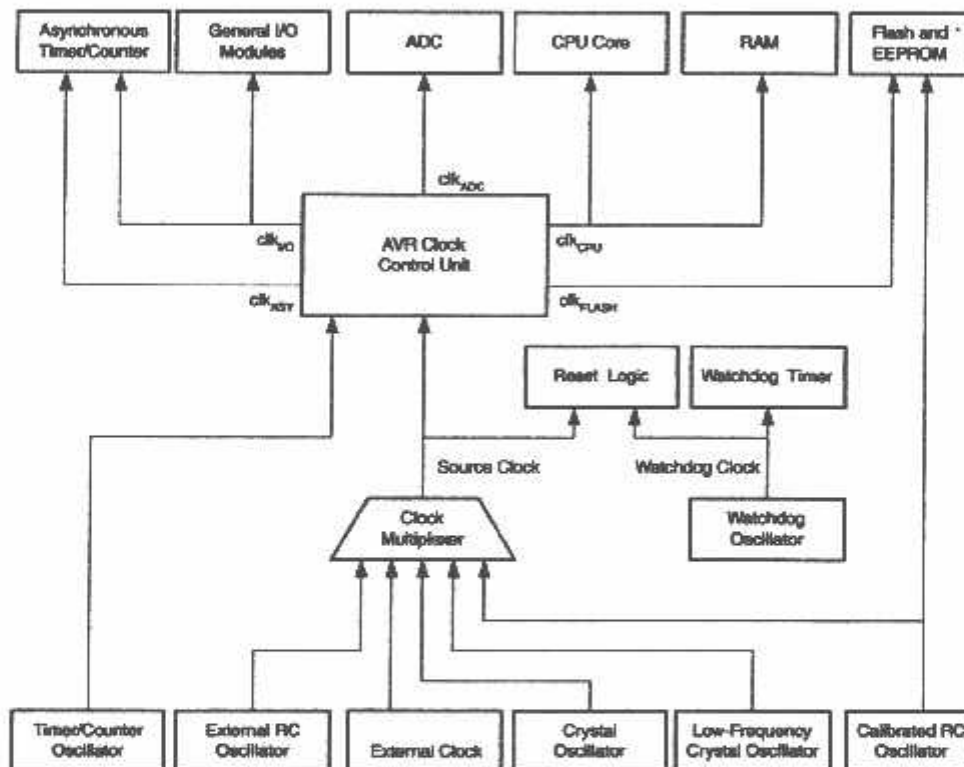
needed detection level, an external low V_{CC} Reset Protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.

System Clock and Clock Options

1 Clock Systems and their Distribution

Figure 9-1 presents the principal clock systems in the AVR and their distribution. All of the clocks need not be active at a given time. In order to reduce power consumption, the clocks to modules not being used can be halted by using different sleep modes, as described in “Power Management and Sleep Modes” on page 32. The clock systems are detailed Figure 9-1.

Figure 9-1. Clock Distribution



.1 CPU Clock – clk_{CPU}

The CPU clock is routed to parts of the system concerned with operation of the AVR core. Examples of such modules are the General Purpose Register File, the Status Register and the Data memory holding the Stack Pointer. Halting the CPU clock inhibits the core from performing general operations and calculations.

.2 I/O Clock – $clk_{I/O}$

The I/O clock is used by the majority of the I/O modules, like Timer/Counters, SPI, and USART. The I/O clock is also used by the External Interrupt module, but note that some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected even if the I/O clock is halted. Also note that address recognition in the TWI module is carried out asynchronously when $clk_{I/O}$ is halted, enabling TWI address reception in all sleep modes.

3 Flash Clock – $\text{clk}_{\text{FLASH}}$

The Flash clock controls operation of the Flash interface. The Flash clock is usually active simultaneously with the CPU clock.

4 Asynchronous Timer Clock – clk_{ASY}

The Asynchronous Timer clock allows the Asynchronous Timer/Counter to be clocked directly from an external 32kHz clock crystal. The dedicated clock domain allows using this Timer/Counter as a real-time counter even when the device is in sleep mode. The Asynchronous Timer/Counter uses the same XTAL pins as the CPU main clock but requires a CPU main clock frequency of more than four times the Oscillator frequency. Thus, asynchronous operation is only available while the chip is clocked on the Internal Oscillator.

5 ADC Clock – clk_{ADC}

The ADC is provided with a dedicated clock domain. This allows halting the CPU and I/O clocks in order to reduce noise generated by digital circuitry. This gives more accurate ADC conversion results.

2 Clock Sources

The device has the following clock source options, selectable by Flash Fuse Bits as shown below. The clock from the selected source is input to the AVR clock generator, and routed to the appropriate modules.

Table 9-1. Device Clocking Options Select⁽¹⁾

Device Clocking Option	CKSEL3:0
External Crystal/Ceramic Resonator	1111 - 1010
External Low-frequency Crystal	1001
External RC Oscillator	1000 - 0101
Calibrated Internal RC Oscillator	0100 - 0001
External Clock	0000

Note: 1. For all fuses "1" means unprogrammed while "0" means programmed.

The various choices for each clocking option is given in the following sections. When the CPU wakes up from Power-down or Power-save, the selected clock source is used to time the start-up, ensuring stable Oscillator operation before instruction execution starts. When the CPU starts from reset, there is an additional delay allowing the power to reach a stable level before commencing normal operation. The Watchdog Oscillator is used for timing this real-time part of the start-up time. The number of WDT Oscillator cycles used for each time-out is shown in Table 9-2. The frequency of the Watchdog Oscillator is voltage dependent as shown in "Typical Characteristics – TA = -40°C to 85°C". The device is shipped with CKSEL = "0001" and SUT = "10" (1MHz Internal RC Oscillator, slowly rising power).

Table 9-2. Number of Watchdog Oscillator Cycles

Typical Time-out ($V_{\text{CC}} = 5.0\text{V}$)	Typical Time-out ($V_{\text{CC}} = 3.0\text{V}$)	Number of Cycles
4.1 ms	4.3 ms	4K (4,096)
65 ms	69 ms	64K (65,536)

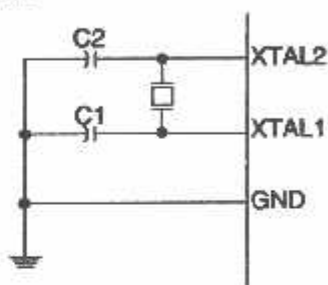
3 Crystal Oscillator

XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier which can be configured for use as an On-chip Oscillator, as shown in Figure 9-2. Either a quartz crystal or a ceramic resonator may be used. The CKOPT Fuse selects between two different Oscillator amplifier modes. When CKOPT is programmed, the Oscillator output will oscillate a full rail-to-rail swing on the output. This mode is suitable when operating in a very noisy environment or when the output from XTAL2 drives a second clock buffer. This mode has a wide frequency range.

When CKOPT is unprogrammed, the Oscillator has a smaller output swing. This reduces power consumption considerably. This mode has a limited frequency range and it cannot be used to drive other clock buffers.

For resonators, the maximum frequency is 8MHz with CKOPT unprogrammed and 16 MHz with CKOPT programmed. C1 and C2 should always be equal for both crystals and resonators. The optimal value of the capacitors depends on the crystal or resonator in use, the amount of stray capacitance, and the electromagnetic noise of the environment. Some initial guidelines for choosing capacitors for use with crystals are given in Table 9-3. For ceramic resonators, the capacitor values given by the manufacturer should be used.

Figure 9-2. Crystal Oscillator Connections



The Oscillator can operate in three different modes, each optimized for a specific frequency range. The operating mode is selected by the fuses CKSEL3:1 as shown in Table 9-3.

Table 9-3. Crystal Oscillator Operating Modes

CKOPT	CKSEL3:1	Frequency Range(MHz)	Recommended Range for Capacitors C1 and C2 for Use with Crystals (pF)
1	101 ⁽¹⁾	0.4 - 0.9	—
1	110	0.9 - 3.0	12 - 22
1	111	3.0 - 8.0	12 - 22
0	101, 110, 111	1.0 ≤	12 - 22

Note: 1. This option should not be used with crystals, only with ceramic resonators.

The CKSEL0 Fuse together with the SUT1:0 Fuses select the start-up times as shown in Table 9-4.

Table 9-4. Start-up Times for the Crystal Oscillator Clock Selection

CKSEL0	SUT1:0	Start-up Time from Power-down and Power-save	Additional Delay from Reset ($V_{CC} = 5.0V$)	Recommended Usage
0	00	258 CK ⁽¹⁾	4.1 ms	Ceramic resonator, fast rising power
0	01	258 CK ⁽¹⁾	65 ms	Ceramic resonator, slowly rising power
0	10	1K CK ⁽²⁾	—	Ceramic resonator, BOD enabled
0	11	1K CK ⁽²⁾	4.1 ms	Ceramic resonator, fast rising power
1	00	1K CK ⁽²⁾	65 ms	Ceramic resonator, slowly rising power
1	01	16K CK	—	Crystal Oscillator, BOD enabled
1	10	16K CK	4.1 ms	Crystal Oscillator, fast rising power
1	11	16K CK	65 ms	Crystal Oscillator, slowly rising power

- Notes:
1. These options should only be used when not operating close to the maximum frequency of the device, and only if frequency stability at start-up is not important for the application. These options are not suitable for crystals.
 2. These options are intended for use with ceramic resonators and will ensure frequency stability at start-up. They can also be used with crystals when not operating close to the maximum frequency of the device, and if frequency stability at start-up is not important for the application.

Low-frequency Crystal Oscillator

To use a 32.768 kHz watch crystal as the clock source for the device, the Low-frequency Crystal Oscillator must be selected by setting the CKSEL Fuses to "1001". The crystal should be connected as shown in Figure 9-2. By programming the CKOPT Fuse, the user can enable internal capacitors on XTAL1 and XTAL2, thereby removing the need for external capacitors. The internal capacitors have a nominal value of 36 pF.

When this Oscillator is selected, start-up times are determined by the SUT Fuses as shown in Table 9-5.

Table 9-5. Start-up Times for the Low-frequency Crystal Oscillator Clock Selection

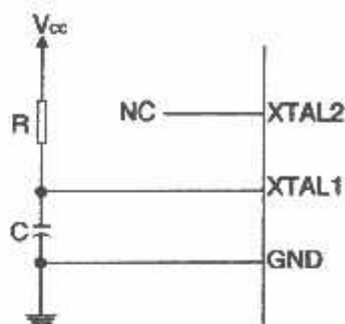
SUT1:0	Start-up Time from Power-down and Power-save	Additional Delay from Reset ($V_{CC} = 5.0V$)	Recommended Usage
00	1K CK ⁽¹⁾	4.1 ms	Fast rising power or BOD enabled
01	1K CK ⁽¹⁾	65 ms	Slowly rising power
10	32K CK	65 ms	Stable frequency at start-up
11	Reserved		

- Note:
1. These options should only be used if frequency stability at start-up is not important for the application.

External RC Oscillator

For timing insensitive applications, the external RC configuration shown in Figure 9-3 can be used. The frequency is roughly estimated by the equation $f = 1/(3RC)$. C should be at least 22 pF. By programming the CKOPT Fuse, the user can enable an internal 36 pF capacitor between XTAL1 and GND, thereby removing the need for an external capacitor.

Figure 9-3. External RC Configuration



The Oscillator can operate in four different modes, each optimized for a specific frequency range. The operating mode is selected by the fuses CKSEL3:0 as shown in Table 9-6.

Table 9-6. External RC Oscillator Operating Modes

CKSEL3:0	Frequency Range (MHz)
0101	0.1 - 0.9
0110	0.9 - 3.0
0111	3.0 - 8.0
1000	8.0 - 12.0

When this Oscillator is selected, start-up times are determined by the SUT Fuses as shown in Table 9-7.

Table 9-7. Start-up Times for the External RC Oscillator Clock Selection

SUT1:0	Start-up Time from Power-down and Power-save	Additional Delay from Reset ($V_{CC} = 5.0V$)	Recommended Usage
00	18 CK	—	BOD enabled
01	18 CK	4.1 ms	Fast rising power
10	18 CK	65 ms	Slowly rising power
11	6 CK ⁽¹⁾	4.1 ms	Fast rising power or BOD enabled

Note: 1. This option should not be used when operating close to the maximum frequency of the device.

Calibrated Internal RC Oscillator

The calibrated internal RC Oscillator provides a fixed 1.0, 2.0, 4.0, or 8.0MHz clock. All frequencies are nominal values at 5V and 25°C. This clock may be selected as the system clock by programming the CKSEL Fuses as shown in Table 9-8. If selected, it will operate with no external components. The CKOPT Fuse should always be unprogrammed when using this clock option. During reset, hardware loads the 1MHz calibration byte into the OSC-CAL Register and thereby automatically calibrates the RC Oscillator. At 5V, 25°C and 1.0MHz Oscillator frequency selected, this calibration gives a frequency within $\pm 3\%$ of the nominal frequency. Using run-time calibration methods as described in application notes available at www.atmel.com/avr it is possible to achieve $\pm 1\%$ accuracy at any given V_{CC} and Temperature. When this Oscillator is used as the chip clock, the Watchdog Oscillator will still be used for the Watchdog Timer and for the Reset Time-out. For more information on the pre-programmed calibration value, see the section "Calibration Byte" on page 209.

Table 9-8. Internal Calibrated RC Oscillator Operating Modes

CKSEL3:0	Nominal Frequency (MHz)
0001 ⁽¹⁾	1.0
0010	2.0
0011	4.0
0100	8.0

Note: 1. The device is shipped with this option selected.

When this Oscillator is selected, start-up times are determined by the SUT Fuses as shown in Table 9-9. PB6 (XTAL1/TOSC1) and PB7 (XTAL2/TOSC2) can be used as either general I/O pins or Timer Oscillator pins:

Table 9-9. Start-up Times for the Internal Calibrated RC Oscillator Clock Selection

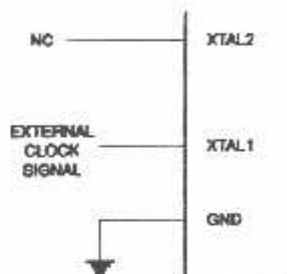
SUT1:0	Start-up Time from Power-down and Power-save	Additional Delay from Reset ($V_{CC} = 5.0V$)	Recommended Usage
00	6 CK	—	BOD enabled
01	6 CK	4.1 ms	Fast rising power
10 ⁽¹⁾	6 CK	65 ms	Slowly rising power
11	Reserved		

Note: 1. The device is shipped with this option selected.

External Clock

To drive the device from an external clock source, XTAL1 should be driven as shown in Figure 9-4. To run the device on an external clock, the CKSEL Fuses must be programmed to "0000". By programming the CKOPT Fuse, the user can enable an internal 36 pF capacitor between XTAL1 and GND, and XTAL2 and GND.

Figure 9-4. External Clock Drive Configuration



When this clock source is selected, start-up times are determined by the SUT Fuses as shown in Table 9-10.

Table 9-10. Start-up Times for the External Clock Selection

SUT1:0	Start-up Time from Power-down and Power-save	Additional Delay from Reset ($V_{CC} = 5.0V$)	Recommended Usage
00	6 CK	—	BOD enabled
01	6 CK	4.1 ms	Fast rising power
10	6 CK	65 ms	Slowly rising power
11	Reserved		

When applying an external clock, it is required to avoid sudden changes in the applied clock frequency to ensure stable operation of the MCU. A variation in frequency of more than 2% from one clock cycle to the next can lead to unpredictable behavior. It is required to ensure that the MCU is kept in Reset during such changes in the clock frequency.

Timer/Counter Oscillator

For AVR microcontrollers with Timer/Counter Oscillator pins (TOSC1 and TOSC2), the crystal is connected directly between the pins. By programming the CKOPT Fuse, the user can enable internal capacitors on XTAL1 and XTAL2, thereby removing the need for external capacitors. The Oscillator is optimized for use with a 32.768 kHz watch crystal. Applying an external clock source to TOSC1 is not recommended.

Note: The Timer/Counter Oscillator uses the same type of crystal oscillator as Low-Frequency Oscillator and the internal capacitors have the same nominal value of 36 pF.

Register Description

OSCCAL – Oscillator Calibration Register

Bit	7	6	5	4	3	2	1	0	
	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	OSCCAL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	Device Specific Calibration Value								

• Bits 7:0 – CAL7:0: Oscillator Calibration Value

Writing the calibration byte to this address will trim the Internal Oscillator to remove process variations from the Oscillator frequency. During Reset, the 1MHz calibration value which is located in the signature row High byte (address 0x00) is automatically loaded into the OSCCAL Register. If the internal RC is used at other frequencies, the calibration values must be loaded manually. This can be done by first reading the signature row by a programmer, and then store the calibration values in the Flash or EEPROM. Then the value can be read by software and loaded into the OSCCAL Register. When OSCCAL is zero, the lowest available frequency is chosen. Writing non-zero values to this register will increase the frequency of the Internal Oscillator. Writing 0xFF to the register gives the highest available frequency. The calibrated Oscillator is used to time EEPROM and Flash access. If EEPROM or Flash is written, do not calibrate to more than 10% above the nominal frequency. Otherwise, the EEPROM or Flash write may fail. Note that the Oscillator is intended for calibration to 1.0, 2.0, 4.0, or 8.0 MHz. Tuning to other values is not guaranteed, as indicated in Table 9-11.

Table 9-11. Internal RC Oscillator Frequency Range

OSCCAL Value	Min Frequency In Percentage of Nominal Frequency (%)	Max Frequency In Percentage of Nominal Frequency (%)
0x00	50	100
0x7F	75	150
0xFF	100	200

1. Power Management and Sleep Modes

1.1 Sleep Modes

Sleep modes enable the application to shut down unused modules in the MCU, thereby saving power. The AVR provides various sleep modes allowing the user to tailor the power consumption to the application's requirements.

Figure 9-1 on page 24 presents the different clock systems in the ATmega8A, and their distribution. The figure is helpful in selecting an appropriate sleep mode. Table 10-1 shows the different clock options and their wake-up sources.

Table 10-1. Active Clock Domains and Wake-up Sources in the Different Sleep Modes

Sleep Mode	Active Clock Domains					Oscillators		Wake-up Sources					
	clk _{CPU}	clk _{FLASH}	clk _{IO}	clk _{ADC}	clk _{ASY}	Main Clock Source Enabled	Timer Osc. Enabled	INT1/INT0	TWI Address Match	Timer2	SPM/EEPROM Ready	ADC	Other I/O
Idle			X	X	X	X	X ⁽²⁾	X	X	X	X	X	X
ADC Noise Reduction				X	X	X	X ⁽²⁾	X ⁽³⁾	X	X	X	X	
Power-down								X ⁽³⁾	X				
Power-save					X ⁽²⁾		X ⁽²⁾	X ⁽³⁾	X	X ⁽²⁾			
Standby ⁽¹⁾						X		X ⁽³⁾	X				

Notes: 1. External Crystal or resonator selected as clock source.

2. If AS2 bit in ASSR is set.

3. Only level interrupt INT1 and INT0.

To enter any of the five sleep modes, the SE bit in MCUCR must be written to logic one and a SLEEP instruction must be executed. The SM2, SM1, and SM0 bits in the MCUCR Register select which sleep mode (Idle, ADC Noise Reduction, Power-down, Power-save, or Standby) will be activated by the SLEEP instruction. See Table 10-1 for a summary.

If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles in addition to the start-up time, it executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the Register File and SRAM are unaltered when the device wakes up from sleep. If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset Vector.

Note that the Extended Standby mode present in many other AVR MCUs has been removed in the ATmega8A, as the TOSC and XTAL inputs share the same physical pins.

1.2 Idle Mode

When the SM2:0 bits are written to 000, the SLEEP instruction makes the MCU enter Idle mode, stopping the CPU but allowing SPI, USART, Analog Comparator, ADC, Two-wire Serial Interface, Timer/Counters, Watchdog, and

the interrupt system to continue operating. This sleep mode basically halts clk_{CPU} and $\text{clk}_{\text{FLASH}}$, while allowing the other clocks to run.

Idle mode enables the MCU to wake up from external triggered interrupts as well as internal ones like the Timer Overflow and USART Transmit Complete interrupts. If wake-up from the Analog Comparator interrupt is not required, the Analog Comparator can be powered down by setting the ACD bit in the Analog Comparator Control and Status Register – ACSR. This will reduce power consumption in Idle mode. If the ADC is enabled, a conversion starts automatically when this mode is entered.

3 ADC Noise Reduction Mode

When the SM2:0 bits are written to 001, the SLEEP instruction makes the MCU enter ADC Noise Reduction mode, stopping the CPU but allowing the ADC, the external interrupts, the Two-wire Serial Interface address watch, Timer/Counter2 and the Watchdog to continue operating (if enabled). This sleep mode basically halts clk_{IO} , clk_{CPU} , and $\text{clk}_{\text{FLASH}}$, while allowing the other clocks to run.

This improves the noise environment for the ADC, enabling higher resolution measurements. If the ADC is enabled, a conversion starts automatically when this mode is entered. Apart from the ADC Conversion Complete interrupt, only an External Reset, a Watchdog Reset, a Brown-out Reset, a Two-wire Serial Interface address match interrupt, a Timer/Counter2 interrupt, an SPM/EEPROM ready interrupt, or an external level interrupt on INT0 or INT1, can wake up the MCU from ADC Noise Reduction mode.

4 Power-down Mode

When the SM2:0 bits are written to 010, the SLEEP instruction makes the MCU enter Power-down mode. In this mode, the External Oscillator is stopped, while the external interrupts, the Two-wire Serial Interface address watch, and the Watchdog continue operating (if enabled). Only an External Reset, a Watchdog Reset, a Brown-out Reset, a Two-wire Serial Interface address match interrupt, or an external level interrupt on INT0 or INT1, can wake up the MCU. This sleep mode basically halts all generated clocks, allowing operation of asynchronous modules only.

Note that if a level triggered interrupt is used for wake-up from Power-down mode, the changed level must be held for some time to wake up the MCU. Refer to “External Interrupts” on page 64 for details.

When waking up from Power-down mode, there is a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is defined by the same CKSEL Fuses that define the Reset Time-out period, as described in “Clock Sources” on page 25.

5 Power-save Mode

When the SM2:0 bits are written to 011, the SLEEP instruction makes the MCU enter Power-save mode. This mode is identical to Power-down, with one exception:

If Timer/Counter2 is clocked asynchronously, i.e. the AS2 bit in ASSR is set, Timer/Counter2 will run during sleep. The device can wake up from either Timer Overflow or Output Compare event from Timer/Counter2 if the corresponding Timer/Counter2 interrupt enable bits are set in TIMSK, and the global interrupt enable bit in SREG is set.

If the asynchronous timer is NOT clocked asynchronously, Power-down mode is recommended instead of Power-save mode because the contents of the registers in the asynchronous timer should be considered undefined after wake-up in Power-save mode if AS2 is 0.

This sleep mode basically halts all clocks except clk_{ASY} , allowing operation only of asynchronous modules, including Timer/Counter 2 if clocked asynchronously.

6 Standby Mode

When the SM2:0 bits are 110 and an external crystal/resonator clock option is selected, the SLEEP instruction makes the MCU enter Standby mode. This mode is identical to Power-down with the exception that the Oscillator is kept running. From Standby mode, the device wakes up in 6 clock cycles.

7 Minimizing Power Consumption

There are several issues to consider when trying to minimize the power consumption in an AVR controlled system. In general, sleep modes should be used as much as possible, and the sleep mode should be selected so that as few as possible of the device's functions are operating. All functions not needed should be disabled. In particular, the following modules may need special consideration when trying to achieve the lowest possible power consumption.

7.1 Analog-to-Digital Converter (ADC)

If enabled, the ADC will be enabled in all sleep modes. To save power, the ADC should be disabled before entering any sleep mode. When the ADC is turned off and on again, the next conversion will be an extended conversion. Refer to "Analog-to-Digital Converter" on page 182 for details on ADC operation.

7.2 Analog Comparator

When entering Idle mode, the Analog Comparator should be disabled if not used. When entering ADC Noise Reduction mode, the Analog Comparator should be disabled. In the other sleep modes, the Analog Comparator is automatically disabled. However, if the Analog Comparator is set up to use the Internal Voltage Reference as input, the Analog Comparator should be disabled in all sleep modes. Otherwise, the Internal Voltage Reference will be enabled, independent of sleep mode. Refer to "Analog Comparator" on page 179 for details on how to configure the Analog Comparator.

7.3 Brown-out Detector

If the Brown-out Detector is not needed in the application, this module should be turned off. If the Brown-out Detector is enabled by the BODEN Fuse, it will be enabled in all sleep modes, and hence, always consume power. In the deeper sleep modes, this will contribute significantly to the total current consumption. Refer to "Brown-out Detection" on page 38 for details on how to configure the Brown-out Detector.

7.4 Internal Voltage Reference

The Internal Voltage Reference will be enabled when needed by the Brown-out Detector, the Analog Comparator or the ADC. If these modules are disabled as described in the sections above, the internal voltage reference will be disabled and it will not be consuming power. When turned on again, the user must allow the reference to start up before the output is used. If the reference is kept on in sleep mode, the output can be used immediately. Refer to "Internal Voltage Reference" on page 39 for details on the start-up time.

7.5 Watchdog Timer

If the Watchdog Timer is not needed in the application, this module should be turned off. If the Watchdog Timer is enabled, it will be enabled in all sleep modes, and hence, always consume power. In the deeper sleep modes, this will contribute significantly to the total current consumption. Refer to "Watchdog Timer" on page 40 for details on how to configure the Watchdog Timer.

7.6 Port Pins

When entering a sleep mode, all port pins should be configured to use minimum power. The most important thing is then to ensure that no pins drive resistive loads. In sleep modes where the both the I/O clock (clk_{IO}) and the ADC clock (clk_{ADC}) are stopped, the input buffers of the device will be disabled. This ensures that no power is consumed by the input logic when not needed. In some cases, the input logic is needed for detecting wake-up conditions, and it will then be enabled. Refer to the section "Digital Input Enable and Sleep Modes" on page 53 for

details on which pins are enabled. If the input buffer is enabled and the input signal is left floating or have an analog signal level close to $V_{CC}/2$, the input buffer will use excessive power.

8 Register Description

8.1 MCUCR – MCU Control Register

The MCU Control Register contains control bits for power management.

Bit	7	6	5	4	3	2	1	0	
	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SE: Sleep Enable**

The SE bit must be written to logic one to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmer's purpose, it is recommended to set the Sleep Enable (SE) bit just before the execution of the SLEEP instruction.

- **Bits 6:4 – SM2:0: Sleep Mode Select Bits 2, 1, and 0**

These bits select between the five available sleep modes as shown in Table 10-2.

Table 10-2. Sleep Mode Select

SM2	SM1	SM0	Sleep Mode
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Standby ⁽¹⁾

Note: 1. Standby mode is only available with external crystals or resonators.

System Control and Reset

1 Resetting the AVR

During Reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the boot section or vice versa. The circuit diagram in Figure 11-1 shows the Reset Logic. Table 26-3 on page 228 defines the electrical parameters of the reset circuitry.

The I/O ports of the AVR are immediately reset to their initial state when a reset source goes active. This does not require any clock source to be running.

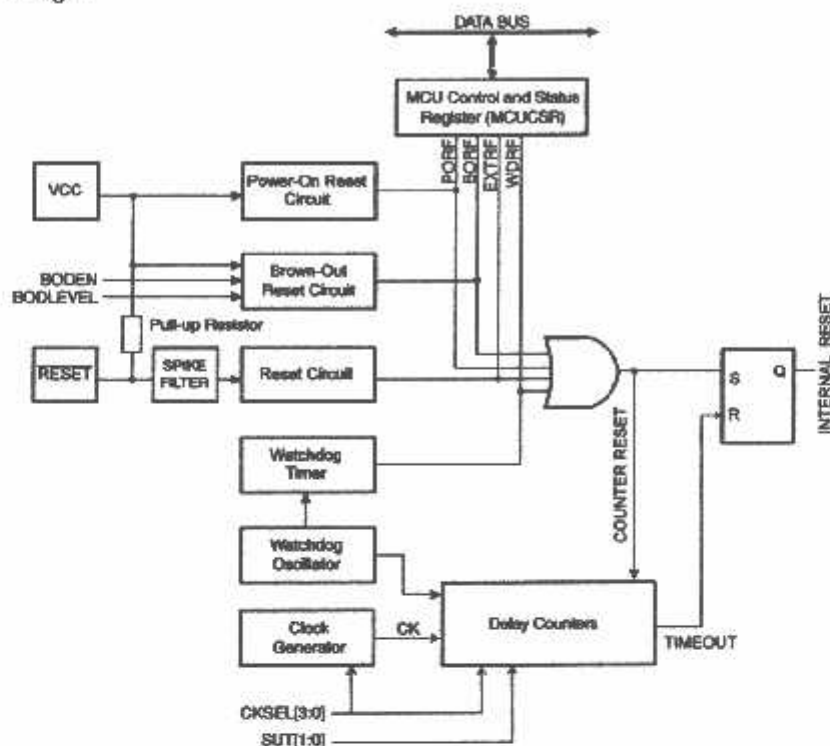
After all reset sources have gone inactive, a delay counter is invoked, stretching the internal reset. This allows the power to reach a stable level before normal operation starts. The time-out period of the delay counter is defined by the user through the CKSEL Fuses. The different selections for the delay period are presented in "Clock Sources" on page 25.

2 Reset Sources

The ATmega8A has four sources of Reset:

- **Power-on Reset.** The MCU is reset when the supply voltage is below the Power-on Reset threshold (V_{POT}).
- **External Reset.** The MCU is reset when a low level is present on the $\overline{RESET}$ pin for longer than the minimum pulse length.
- **Watchdog Reset.** The MCU is reset when the Watchdog Timer period expires and the Watchdog is enabled.
- **Brown-out Reset.** The MCU is reset when the supply voltage V_{CC} is below the Brown-out Reset threshold (V_{BOT}) and the Brown-out Detector is enabled.

Figure 11-1. Reset Logic



2.1 Power-on Reset

A Power-on Reset (POR) pulse is generated by an On-chip detection circuit. The detection level is defined in Table 26-3 on page 228. The POR is activated whenever V_{CC} is below the detection level. The POR circuit can be used to trigger the Start-up Reset, as well as to detect a failure in supply voltage.

A Power-on Reset (POR) circuit ensures that the device is reset from Power-on. Reaching the Power-on Reset threshold voltage invokes the delay counter, which determines how long the device is kept in RESET after V_{CC} rise. The RESET signal is activated again, without any delay, when V_{CC} decreases below the detection level.

Figure 11-2. MCU Start-up, RESET Tied to V_{CC}

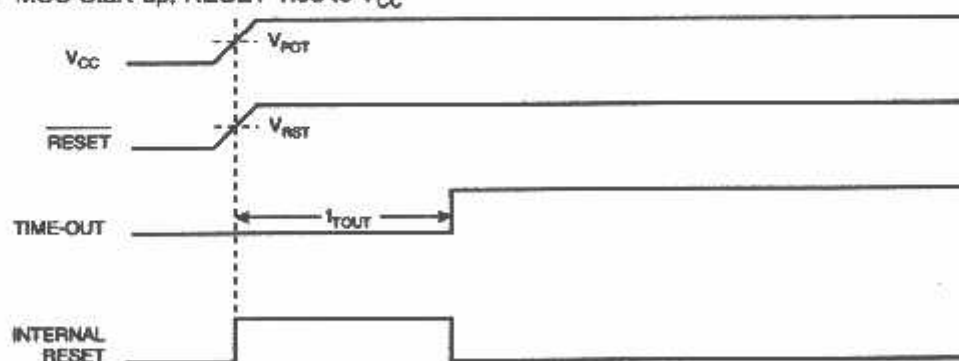
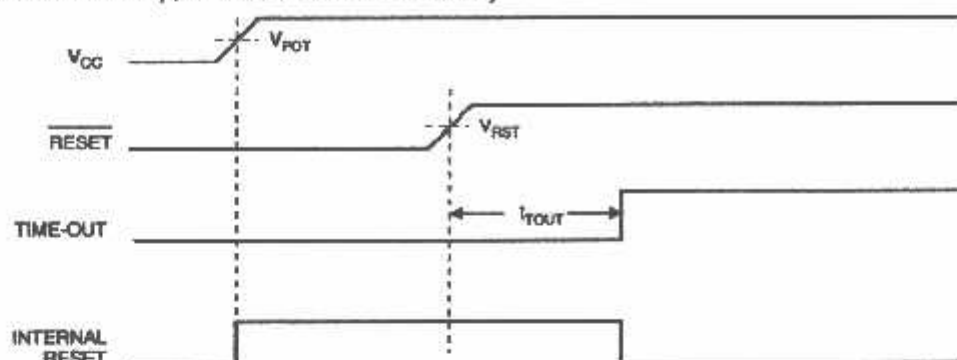


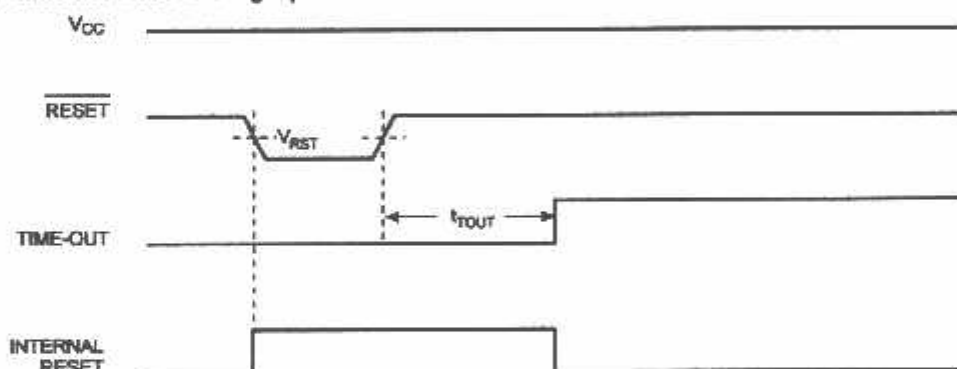
Figure 11-3. MCU Start-up, $\overline{\text{RESET}}$ Extended Externally



2.2 External Reset

An External Reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than the minimum pulse width (see Table 26-3 on page 228) will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage – V_{RST} on its positive edge, the delay counter starts the MCU after the time-out period t_{TOUT} has expired.

Figure 11-4. External Reset During Operation



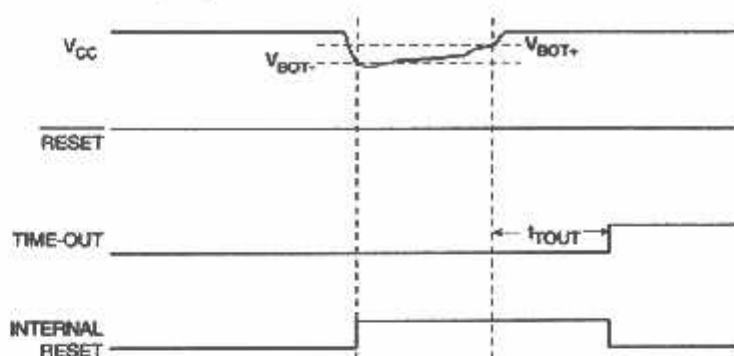
2.3 Brown-out Detection

ATmega8A has an On-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by the fuse BODLEVEL to be 2.7V (BODLEVEL unprogrammed), or 4.0V (BODLEVEL programmed). The trigger level has a hysteresis to ensure spike free Brown-out Detection. The hysteresis on the detection level should be interpreted as $V_{BOT+} = V_{BOT} + V_{HYST}/2$ and $V_{BOT-} = V_{BOT} - V_{HYST}/2$.

The BOD circuit can be enabled/disabled by the fuse BODEN. When the BOD is enabled (BODEN programmed), and V_{CC} decreases to a value below the trigger level (V_{BOT-} in Figure 11-5), the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level (V_{BOT+} in Figure 11-5), the delay counter starts the MCU after the time-out period t_{TOUT} has expired.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than t_{BOD} given in Table 26-3 on page 228.

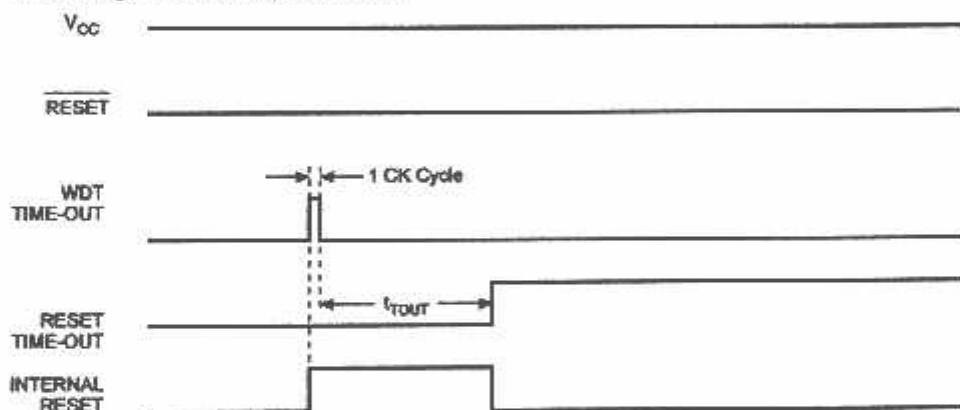
Figure 11-5. Brown-out Reset During Operation



2.4 Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of 1 CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the time-out period t_{TOUT} . Refer to page 40 for details on operation of the Watchdog Timer.

Figure 11-6. Watchdog Reset During Operation



3 Internal Voltage Reference

ATmega8A features an internal bandgap reference. This reference is used for Brown-out Detection, and it can be used as an input to the Analog Comparator or the ADC. The 2.56V reference to the ADC is generated from the internal bandgap reference.

3.1 Voltage Reference Enable Signals and Start-up Time

The voltage reference has a start-up time that may influence the way it should be used. The start-up time is given in Table 26-3 on page 228. To save power, the reference is not always turned on. The reference is on during the following situations:

1. When the BOD is enabled (by programming the BODEN Fuse).
2. When the bandgap reference is connected to the Analog Comparator (by setting the ACBG bit in ACSR).
3. When the ADC is enabled.

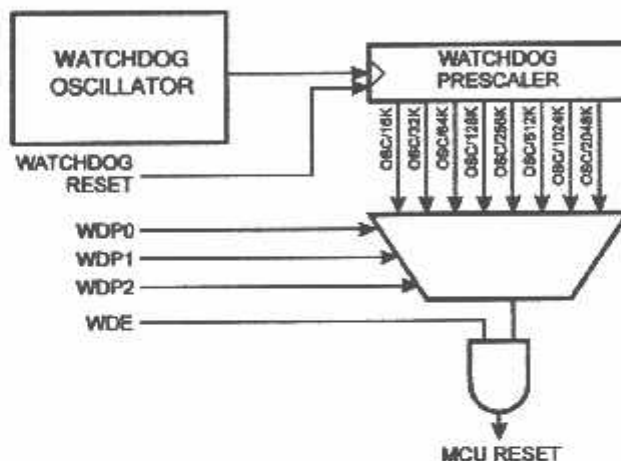
Thus, when the BOD is not enabled, after setting the ACBG bit or enabling the ADC, the user must always allow the reference to start up before the output from the Analog Comparator or ADC is used. To reduce power consumption in Power-down mode, the user can avoid the three conditions above to ensure that the reference is turned off before entering Power-down mode.

4 Watchdog Timer

The Watchdog Timer is clocked from a separate On-chip Oscillator which runs at 1 MHz. This is the typical value at $V_{CC} = 5V$. See characterization data for typical values at other V_{CC} levels. By controlling the Watchdog Timer prescaler, the Watchdog Reset interval can be adjusted as shown in Table 11-7 on page 40. The WDR – Watchdog Reset – instruction resets the Watchdog Timer. The Watchdog Timer is also reset when it is disabled and when a Chip Reset occurs. Eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog Reset, the ATmega8A resets and executes from the Reset Vector. For timing details on the Watchdog Reset, refer to “Watchdog Reset” on page 39.

To prevent unintentional disabling of the Watchdog, a special turn-off sequence must be followed when the Watchdog is disabled. Refer to the description of the Watchdog Timer Control Register for details.

Figure 11-7. Watchdog Timer



5 Timed Sequences for Changing the Configuration of the Watchdog Timer

The sequence for changing the Watchdog Timer configuration differs slightly between the safety levels. Separate procedures are described for each level.

Assembly Code Example

```

WDT_off:
; reset WDT
WDR
; Write logical one to WDCE and WDE
in r16, WDTCR
ori r16, (1<<WDCE) | (1<<WDE)
out WDTCR, r16
; Turn off WDT
ldi r16, (0<<WDE)
out WDTCR, r16
ret

```

C Code Example

```

void WDT_off(void)
{
    /* reset WDT */
    _WDR();
    /* Write logical one to WDCE and WDE */
    WDTCR |= (1<<WDCE) | (1<<WDE);
    /* Turn off WDT */
    WDTCR = 0x00;
}

```

5.1 Safety Level 1 (WDTON Fuse Unprogrammed)

In this mode, the Watchdog Timer is initially disabled, but can be enabled by writing the WDE bit to 1 without any restriction. A timed sequence is needed when changing the Watchdog Time-out period or disabling an enabled Watchdog Timer. To disable an enabled Watchdog Timer and/or changing the Watchdog Time-out, the following procedure must be followed:

1. In the same operation, write a logic one to WDCE and WDE. A logic one must be written to WDE regardless of the previous value of the WDE bit.
2. Within the next four clock cycles, in the same operation, write the WDE and WDP bits as desired, but with the WDCE bit cleared.

5.2 Safety Level 2 (WDTON Fuse Programmed)

In this mode, the Watchdog Timer is always enabled, and the WDE bit will always read as one. A timed sequence is needed when changing the Watchdog Time-out period. To change the Watchdog Time-out, the following procedure must be followed:

1. In the same operation, write a logical one to WDCE and WDE. Even though the WDE always is set, the WDE must be written to one to start the timed sequence.

Within the next four clock cycles, in the same operation, write the WDP bits as desired, but with the WDCE bit cleared. The value written to the WDE bit is irrelevant.

6 Register Description

6.1 MCUCSR – MCU Control and Status Register

The MCU Control and Status Register provides information on which reset source caused an MCU Reset.

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	WDRF	BORF	EXTRF	PORF	MCUCSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	See Bit Description				

- **Bit 7:4 – Res: Reserved Bits**

These bits are reserved bits in the ATmega8A and always read as zero.

- **Bit 3 – WDRF: Watchdog Reset Flag**

This bit is set if a Watchdog Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 2 – BORF: Brown-out Reset Flag**

This bit is set if a Brown-out Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 1 – EXTRF: External Reset Flag**

This bit is set if an External Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 0 – PORF: Power-on Reset Flag**

This bit is set if a Power-on Reset occurs. The bit is reset only by writing a logic zero to the flag.

To make use of the Reset Flags to identify a reset condition, the user should read and then reset the MCUCSR as early as possible in the program. If the register is cleared before another reset occurs, the source of the reset can be found by examining the Reset Flags.

6.2 WDTCR – Watchdog Timer Control Register

Bit	7	6	5	4	3	2	1	0	
	–	–	–	WDCE	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:5 – Res: Reserved Bits**

These bits are reserved bits in the ATmega8A and will always read as zero.

- **Bit 4 – WDCE: Watchdog Change Enable**

This bit must be set when the WDE bit is written to logic zero. Otherwise, the Watchdog will not be disabled. Once written to one, hardware will clear this bit after four clock cycles. Refer to the description of the WDE bit for a Watchdog disable procedure. In Safety Level 1 and 2, this bit must also be set when changing the prescaler bits. See the Code Examples on page 40.

- **Bit 3 – WDE: Watchdog Enable**

When the WDE is written to logic one, the Watchdog Timer is enabled, and if the WDE is written to logic zero, the Watchdog Timer function is disabled. WDE can only be cleared if the WDCE bit has logic level one. To disable an enabled Watchdog Timer, the following procedure must be followed:

1. In the same operation, write a logic one to WDCE and WDE. A logic one must be written to WDE even though it is set to one before the disable operation starts.
2. Within the next four clock cycles, write a logic 0 to WDE. This disables the Watchdog.

• **Bits 2:0 – WDP2, WDP1, WDP0: Watchdog Timer Prescaler 2, 1, and 0**

The WDP2, WDP1, and WDP0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding Timeout Periods are shown in Table 11-1.

Table 11-1. Watchdog Timer Prescale Select

WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at $V_{CC} = 3.0V$	Typical Time-out at $V_{CC} = 5.0V$
0	0	0	16K (16,384)	17.1 ms	16.3 ms
0	0	1	32K (32,768)	34.3 ms	32.5 ms
0	1	0	64K (65,536)	68.5 ms	65 ms
0	1	1	128K (131,072)	0.14 s	0.13 s
1	0	0	256K (262,144)	0.27 s	0.26 s
1	0	1	512K (524,288)	0.55 s	0.52 s
1	1	0	1,024K (1,048,576)	1.1 s	1.0 s
1	1	1	2,048K (2,097,152)	2.2 s	2.1 s

The following code example shows one assembly and one C function for turning off the WDT. The example assumes that interrupts are controlled (for example, by disabling interrupts globally) so that no interrupts will occur during execution of these functions.

Interrupts

This section describes the specifics of the interrupt handling performed by the ATmega8A. For a general explanation of the AVR interrupt handling, refer to "Reset and Interrupt Handling" on page 12.

1 Interrupt Vectors in ATmega8A

Table 12-1. Reset and Interrupt Vectors

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	0x000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset, and Watchdog Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	TIMER2 COMP	Timer/Counter2 Compare Match
5	0x004	TIMER2 OVF	Timer/Counter2 Overflow
6	0x005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	0x006	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	0x007	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	0x008	TIMER1 OVF	Timer/Counter1 Overflow
10	0x009	TIMER0 OVF	Timer/Counter0 Overflow
11	0x00A	SPI, STC	Serial Transfer Complete
12	0x00B	USART, RXC	USART, Rx Complete
13	0x00C	USART, UDRE	USART Data Register Empty
14	0x00D	USART, TXC	USART, Tx Complete
15	0x00E	ADC	ADC Conversion Complete
16	0x00F	EE_RDY	EEPROM Ready
17	0x010	ANA_COMP	Analog Comparator
18	0x011	TWI	Two-wire Serial Interface
19	0x012	SPM_RDY	Store Program Memory Ready

- Notes:
1. When the BOOTRST Fuse is programmed, the device will jump to the Boot Loader address at reset, see "Boot Loader Support – Read-While-Write Self-Programming" on page 194.
 2. When the IVSEL bit in GICR is set, Interrupt Vectors will be moved to the start of the boot Flash section. The address of each Interrupt Vector will then be the address in this table added to the start address of the boot Flash section.

Table 12-2 shows reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the boot section or vice versa.

Table 12-2. Reset and Interrupt Vectors Placement

BOOTSF ⁽¹⁾	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x001
1	1	0x000	Boot Reset Address + 0x001
0	0	Boot Reset Address	0x001
0	1	Boot Reset Address	Boot Reset Address + 0x001

Note: 1. The Boot Reset Address is shown in Table 24-6 on page 204. For the BOOTSF Fuse "1" means unprogrammed while "0" means programmed.

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega8A is:

```

addressLabels Code      Comments
$000      rjmp  RESET      ; Reset Handler
$001      rjmp  EXT_INT0    ; IRQ0 Handler
$002      rjmp  EXT_INT1    ; IRQ1 Handler
$003      rjmp  TIM2_COMP    ; Timer2 Compare Handler
$004      rjmp  TIM2_OVF     ; Timer2 Overflow Handler
$005      rjmp  TIM1_CAPT    ; Timer1 Capture Handler
$006      rjmp  TIM1_COMPA    ; Timer1 CompareA Handler
$007      rjmp  TIM1_COMPB    ; Timer1 CompareB Handler
$008      rjmp  TIM1_OVF     ; Timer1 Overflow Handler
$009      rjmp  TIM0_OVF     ; Timer0 Overflow Handler
$00a      rjmp  SPI_STC      ; SPI Transfer Complete Handler
$00b      rjmp  USART_RXC    ; USART RX Complete Handler
$00c      rjmp  USART_UDRE    ; UDR Empty Handler
$00d      rjmp  USART_TXC    ; USART TX Complete Handler
$00e      rjmp  ADC          ; ADC Conversion Complete Handler
$00f      rjmp  EE_RDY       ; EEPROM Ready Handler
$010      rjmp  ANA_COMP     ; Analog Comparator Handler
$011      rjmp  TWI          ; Two-wire Serial Interface Handler
$012      rjmp  SPM_RDY      ; Store Program Memory Ready Handler
;
$013  RESET: ldi      r16,high(RAMEND); Main program start
$014      out    SPH,r16      ; Set Stack Pointer to top of RAM
$015      ldi    r16,low(RAMEND)
$016      out    SPL,r16
$017      sei                      ; Enable interrupts
$018      <instr> xxx
;
;
;

```

When the BOOTRST Fuse is unprogrammed, the boot section size set to 2K bytes and the IVSEL bit in the GICR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

Address	Labels	Code	Comments
\$000		rjmp RESET	; Reset handler
;			
\$001	RESET:	ldi r16,high(RAMEND);	Main program start
\$002		out SPH,r16	; Set Stack Pointer to top of RAM
\$003		ldi r16,low(RAMEND)	
\$004		out SPL,r16	
\$005		sei	; Enable interrupts
\$006		<instr> xxx	
;			
.org \$c01			
\$c01		rjmp EXT_INT0	; IRQ0 Handler
\$c02		rjmp EXT_INT1	; IRQ1 Handler
..	..	..	;
\$c12		rjmp SPM_RDY	; Store Program Memory Ready Handler

When the BOOTRST Fuse is programmed and the boot section size set to 2K bytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

Address	Labels	Code	Comments
.org \$001			
\$001		rjmp EXT_INT0	; IRQ0 Handler
\$002		rjmp EXT_INT1	; IRQ1 Handler
..	..	..	;
\$012		rjmp SPM_RDY	; Store Program Memory Ready Handler
;			
.org \$c00			
\$c00		rjmp RESET	; Reset handler
;			
\$c01	RESET:	ldi r16,high(RAMEND);	Main program start
\$c02		out SPH,r16	; Set Stack Pointer to top of RAM
\$c03		ldi r16,low(RAMEND)	
\$c04		out SPL,r16	
\$c05		sei	; Enable interrupts
\$c06		<instr> xxx	

When the BOOTRST Fuse is programmed, the boot section size set to 2K bytes, and the IVSEL bit in the GICR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```

AddressLabels      Code      Comments
;
.org $c00
$c00      rjmp    RESET      ; Reset handler
$c01      rjmp    EXT_INT0    ; IRQ0 Handler
$c02      rjmp    EXT_INT1    ; IRQ1 Handler
;.        ;.        ;.
$c12      rjmp    SPM_RDY     ; Store Program Memory Ready Handler
$c13  RESET: ldi      r16,high(RAMEND); Main program start
$c14      out     SPH,r16     ; Set Stack Pointer to top of RAM
$c15      ldi     r16,low(RAMEND)
$c16      out     SPL,r16
$c17      sei                      ; Enable interrupts
$c18      <instr> xxx

```

1.1 Moving Interrupts Between Application and Boot Space

The General Interrupt Control Register controls the placement of the Interrupt Vector table.

2 Register Description

2.1 GICR – General Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	–	–	–	–	IVSEL	IVCE	GICR
Read/Write	R/W	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 1 – IVSEL: Interrupt Vector Select

When the IVSEL bit is cleared (zero), the Interrupt Vectors are placed at the start of the Flash memory. When this bit is set (one), the Interrupt Vectors are moved to the beginning of the Boot Loader section of the Flash. The actual address of the start of the boot Flash section is determined by the BOOTSZ Fuses. Refer to the section "Boot Loader Support – Read-While-Write Self-Programming" on page 194 for details. To avoid unintentional changes of Interrupt Vector tables, a special write procedure must be followed to change the IVSEL bit:

1. Write the Interrupt Vector Change Enable (IVCE) bit to one.
2. Within four cycles, write the desired value to IVSEL while writing a zero to IVCE.

Interrupts will automatically be disabled while this sequence is executed. Interrupts are disabled in the cycle IVCE is set, and they remain disabled until after the instruction following the write to IVSEL. If IVSEL is not written, interrupts remain disabled for four cycles. The I-bit in the Status Register is unaffected by the automatic disabling.

Note: If Interrupt Vectors are placed in the Boot Loader section and Boot Lock bit BLB02 is programmed, interrupts are disabled while executing from the Application section. If Interrupt Vectors are placed in the Application section and Boot Lock bit BLB12 is programmed, interrupts are disabled while executing from the Boot Loader section. Refer to the section "Boot Loader Support – Read-While-Write Self-Programming" on page 194 for details on Boot Lock Bits.

- **Bit 0 – IVCE: Interrupt Vector Change Enable**

The IVCE bit must be written to logic one to enable change of the IVSEL bit. IVCE is cleared by hardware four cycles after it is written or when IVSEL is written. Setting the IVCE bit will disable interrupts, as explained in the IVSEL description above. See Code Example below.

Assembly Code Example

<pre>Move_interrupts: ; Enable change of Interrupt Vectors ldi r16, (1<<IVCE) out GICR, r16 ; Move interrupts to boot Flash section ldi r16, (1<<IVSEL) out GICR, r16 ret</pre>

C Code Example

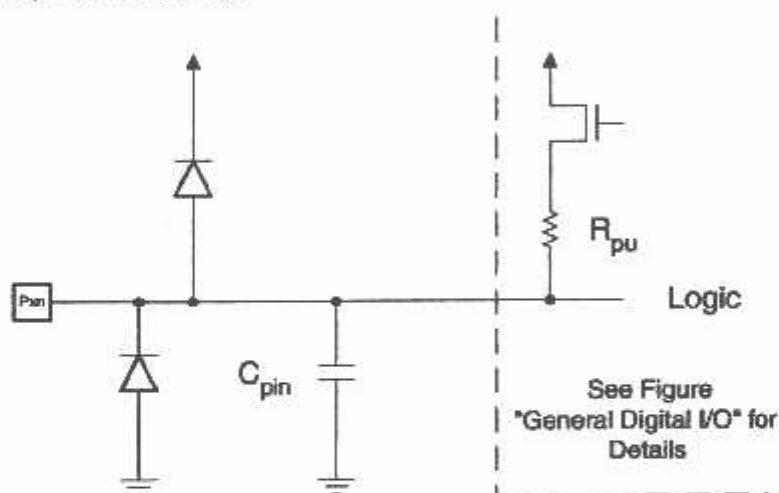
<pre>void Move_interrupts(void) { /* Enable change of Interrupt Vectors */ GICR = (1<<IVCE); /* Move interrupts to boot Flash section */ GICR = (1<<IVSEL); }</pre>

I/O Ports

1 Overview

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies when changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input). Each output buffer has symmetrical drive characteristics with both high sink and source capability. The pin driver is strong enough to drive LED displays directly. All port pins have individually selectable pull-up resistors with a supply-voltage invariant resistance. All I/O pins have protection diodes to both V_{CC} and Ground as indicated in Figure 13-1. Refer to "Electrical Characteristics – $T_A = -40^{\circ}\text{C}$ to 85°C " on page 225 for a complete list of parameters.

Figure 13-1. I/O Pin Equivalent Schematic



All registers and bit references in this section are written in general form. A lower case "x" represents the numbering letter for the port, and a lower case "n" represents the bit number. However, when using the register or bit defines in a program, the precise form must be used (i.e., PORTB3 for bit 3 in Port B, here documented generally as PORTxn). The physical I/O Registers and bit locations are listed in "Register Description" on page 62.

Three I/O memory address locations are allocated for each port, one each for the Data Register – PORTx, Data Direction Register – DDRx, and the Port Input Pins – PINx. The Port Input Pins I/O location is read only, while the Data Register and the Data Direction Register are read/write. In addition, the Pull-up Disable – PUD bit in SFIOR disables the pull-up function for all pins in all ports when set.

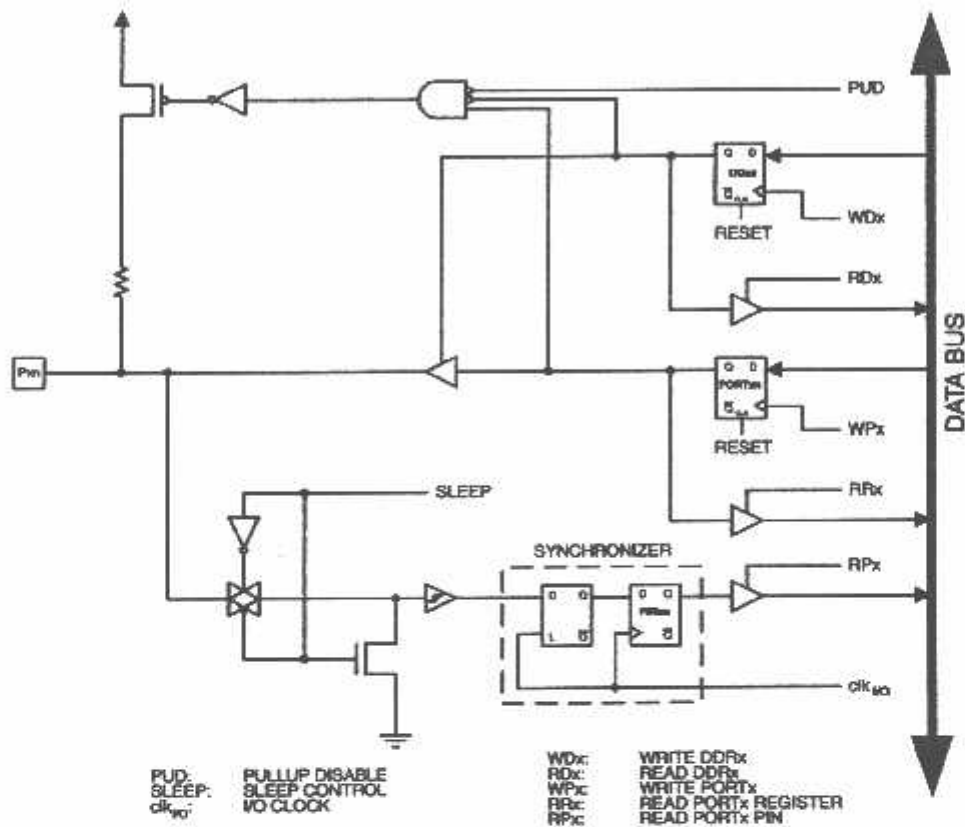
Using the I/O port as General Digital I/O is described in "Ports as General Digital I/O" on page 50. Most port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in "Alternate Port Functions" on page 54. Refer to the individual module sections for a full description of the alternate functions.

Note that enabling the alternate function of some of the port pins does not affect the use of the other pins in the port as general digital I/O.

2 Ports as General Digital I/O

The ports are bi-directional I/O ports with optional internal pull-ups. Figure 13-2 shows a functional description of one I/O port pin, here generically called Pxn.

Figure 13-2. General Digital I/O⁽¹⁾



Note: 1. WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. clk_{IO}, SLEEP, and PUD are common to all ports.

2.1 Configuring the Pin

Each port pin consists of 3 Register bits: DDxn, PORTxn, and PINxn. As shown in "Register Description" on page 62, the DDxn bits are accessed at the DDRx I/O address, the PORTxn bits at the PORTx I/O address, and the PINxn bits at the PINx I/O address.

The DDxn bit in the DDRx Register selects the direction of this pin. If DDxn is written logic one, Pxn is configured as an output pin. If DDxn is written logic zero, Pxn is configured as an input pin.

If PORTx_n is written logic one when the pin is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTx_n has to be written logic zero or the pin has to be configured as an output pin. The port pins are tri-stated when a reset condition becomes active, even if no clocks are running.

If PORTxn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTxn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

When switching between tri-state ($\{DDx_n, PORTx_n\} = 0b00$) and output high ($\{DDx_n, PORTx_n\} = 0b11$), an intermediate state with either pull-up enabled ($\{DDx_n, PORTx_n\} = 0b01$) or output low ($\{DDx_n, PORTx_n\} = 0b10$) must occur. Normally, the pull-up enabled state is fully acceptable, as a high-impedant environment will not notice the

difference between a strong high driver and a pull-up. If this is not the case, the PUD bit in the SFIOR Register can be set to disable all pull-ups in all ports.

Switching between input with pull-up and output low generates the same problem. The user must use either the tri-state ($\{DDxn, PORTxn\} = 0b00$) or the output high state ($\{DDxn, PORTxn\} = 0b11$) as an intermediate step.

Table 13-1 summarizes the control signals for the pin value.

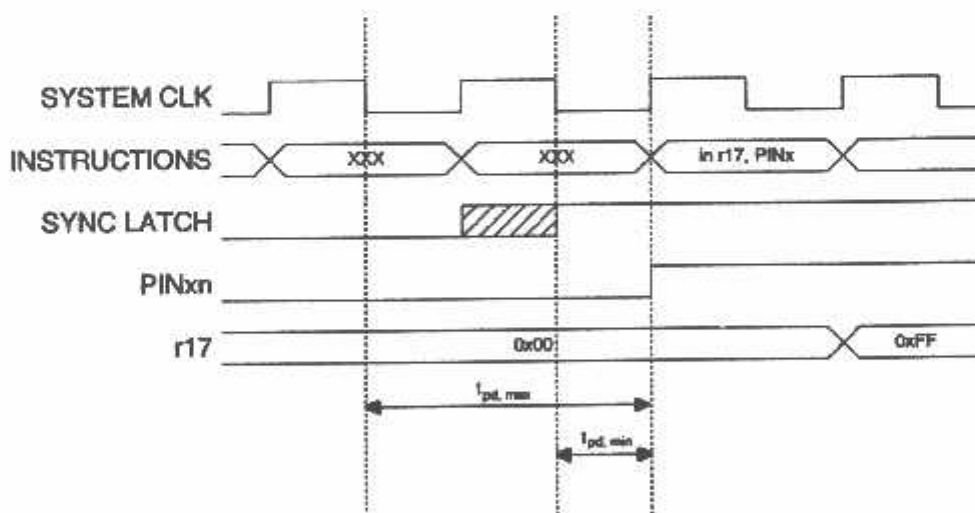
Table 13-1. Port Pin Configurations

DDxn	PORTxn	PUD (In SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if external pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

13.2 Reading the Pin Value

Independent of the setting of Data Direction bit DDxn, the port pin can be read through the PINxn Register Bit. As shown in Figure 13-2, the PINxn Register bit and the preceding latch constitute a synchronizer. This is needed to avoid metastability if the physical pin changes value near the edge of the internal clock, but it also introduces a delay. Figure 13-3 shows a timing diagram of the synchronization when reading an externally applied pin value. The maximum and minimum propagation delays are denoted $t_{pd,max}$ and $t_{pd,min}$, respectively.

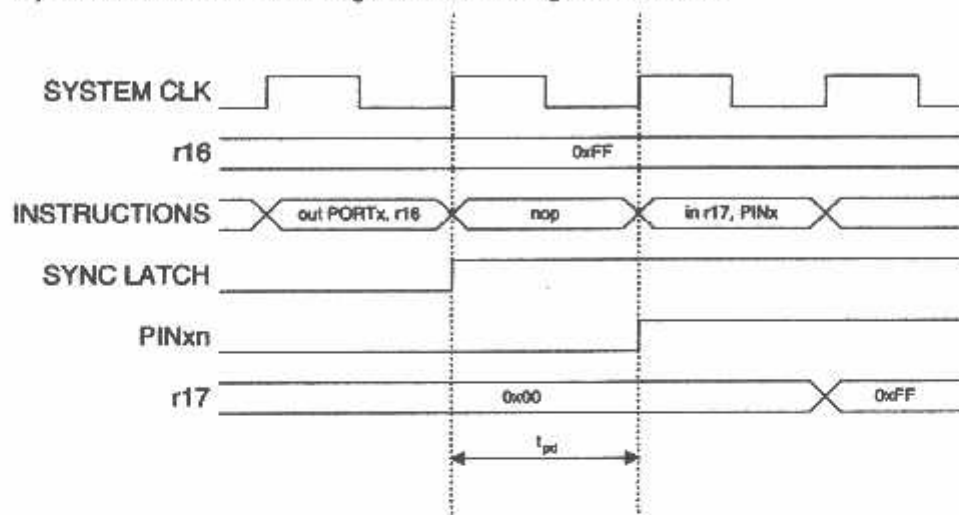
Figure 13-3. Synchronization when Reading an Externally Applied Pin Value



Consider the clock period starting shortly after the first falling edge of the system clock. The latch is closed when the clock is low, and goes transparent when the clock is high, as indicated by the shaded region of the "SYNC LATCH" signal. The signal value is latched when the system clock goes low. It is clocked into the PINxn Register at the succeeding positive clock edge. As indicated by the two arrows $t_{pd,max}$ and $t_{pd,min}$, a single signal transition on the pin will be delayed between $\frac{1}{2}$ and $1\frac{1}{2}$ system clock period depending upon the time of assertion.

When reading back a software assigned pin value, a *nop* instruction must be inserted as indicated in Figure 13-4. The *out* instruction sets the "SYNC LATCH" signal at the positive edge of the clock. In this case, the delay t_{pd} through the synchronizer is 1 system clock period.

Figure 13-4. Synchronization when Reading a Software Assigned Pin Value



The following code example shows how to set port B pins 0 and 1 high, 2 and 3 low, and define the port pins from 4 to 7 as input with pull-ups assigned to port pins 6 and 7. The resulting pin values are read back again, but as previously discussed, a *nop* instruction is included to be able to read back the value recently assigned to some of the pins.

Assembly Code Example⁽¹⁾

```

;
; Define pull-ups and set outputs high
; Define directions for port pins
ldi r16, (1<<PB7) | (1<<PB6) | (1<<PB1) | (1<<PB0)
ldi r17, (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0)
out PORTB, r16
out DDRE, r17
; Insert nop for synchronization
nop
; Read port pins
in r16, PINB
;

```

C Code Example⁽¹⁾

```

unsigned char i;
;
/* Define pull-ups and set outputs high */
/* Define directions for port pins */
PORTB = (1<<PB7) | (1<<PB6) | (1<<PB1) | (1<<PB0);
DDRE = (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
/* Insert nop for synchronization*/
_NOP();
/* Read port pins */
i = PINB;
;

```

Note: 1. For the assembly program, two temporary registers are used to minimize the time from pull-ups are set on pins 0, 1, 6, and 7, until the direction bits are correctly set, defining bit 2 and 3 as low and redefining bits 0 and 1 as strong high drivers.

2.3 Digital Input Enable and Sleep Modes

As shown in Figure 13-2, the digital input signal can be clamped to ground at the input of the Schmitt-trigger. The signal denoted **SLEEP** in the figure, is set by the MCU Sleep Controller in Power-down mode, Power-save mode, and Standby mode to avoid high power consumption if some input signals are left floating, or have an analog signal level close to $V_{CC}/2$.

SLEEP is overridden for port pins enabled as External Interrupt pins. If the External Interrupt Request is not enabled, **SLEEP** is active also for these pins. **SLEEP** is also overridden by various other alternate functions as described in "Alternate Port Functions" on page 54.

If a logic high level ("one") is present on an Asynchronous External Interrupt pin configured as "Interrupt on Rising Edge, Falling Edge, or Any Logic Change on Pin" while the external interrupt is *not* enabled, the corresponding External Interrupt Flag will be set when resuming from the above mentioned sleep modes, as the clamping in these sleep modes produces the requested logic change.

2.4 Unconnected pins

If some pins are unused, it is recommended to ensure that these pins have a defined level. Even though most of the digital inputs are disabled in the deep sleep modes as described above, floating inputs should be avoided to reduce current consumption in all other modes where the digital inputs are enabled (Reset, Active mode and Idle mode).

The simplest method to ensure a defined level of an unused pin, is to enable the internal pull-up. In this case, the pull-up will be disabled during reset. If low power consumption during reset is important, it is recommended to use an external pull-up or pull-down. Connecting unused pins directly to V_{CC} or GND is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.

3 Alternate Port Functions

Most port pins have alternate functions in addition to being general digital I/Os. Figure 13-5 shows how the port pin control signals from the simplified Figure 13-2 can be overridden by alternate functions. The overriding signals may not be present in all port pins, but the figure serves as a generic description applicable to all port pins in the AVR microcontroller family.

Figure 13-5. Alternate Port Functions⁽¹⁾

Table 13-6 summarizes the function of the overriding signals. The pin and port indexes from Figure 13-5 are not shown in the succeeding tables. The overriding signals are generated internally in the modules having the alternate function.

Figure 13-6. Generic Description of Overriding Signals for Alternate Functions

Signal Name	Full Name	Description
PUE	Pull-up Override Enable	If this signal is set, the pull-up enable is controlled by the PUEV signal. If this signal is cleared, the pull-up is enabled when [DDxn, PORTxn, PUD] = 0b010.
PUEV	Pull-up Override Value	If PUE is set, the pull-up is enabled/disabled when PUEV is set/cleared, regardless of the setting of the DDxn, PORTxn, and PUD Register bits.
DOE	Data Direction Override Enable	If this signal is set, the Output Driver Enable is controlled by the DOV signal. If this signal is cleared, the Output driver is enabled by the DDxn Register bit.
DOV	Data Direction Override Value	If DOE is set, the Output Driver is enabled/disabled when DOV is set/cleared, regardless of the setting of the DDxn Register bit.
PVE	Port Value Override Enable	If this signal is set and the Output Driver is enabled, the port value is controlled by the PVEV signal. If PVE is cleared, and the Output Driver is enabled, the port Value is controlled by the PORTxn Register bit.
PVEV	Port Value Override Value	If PVE is set, the port value is set to PVEV, regardless of the setting of the PORTxn Register bit.
DIE	Digital Input Enable Override Enable	If this bit is set, the Digital Input Enable is controlled by the DIEV signal. If this signal is cleared, the Digital Input Enable is determined by MCU-state (Normal mode, sleep modes).
DIEV	Digital Input Enable Override Value	If DIE is set, the Digital Input is enabled/disabled when DIEV is set/cleared, regardless of the MCU state (Normal mode, sleep modes).
DI	Digital Input	This is the Digital Input to alternate functions. In the figure, the signal is connected to the output of the schmitt trigger but before the synchronizer. Unless the Digital Input is used as a clock source, the module with the alternate function will use its own synchronizer.
AIO	Analog Input/output	This is the Analog Input/output to/from alternate functions. The signal is connected directly to the pad, and can be used bi-directionally.

The following subsections shortly describe the alternate functions for each port, and relate the overriding signals to the alternate function. Refer to the alternate function description for further details.

3.1 SFIOR – Special Function IO Register

Bit	7	6	5	4	3	2	1	0	
					AC8E	PUD	PSR2	PSR10	SFIOR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 2 – PUD: Pull-up Disable**

When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0b01). See “Configuring the Pin” on page 50 for more details about this feature.

1.2 Alternate Functions of Port B

The Port B pins with alternate functions are shown in Table 13-2.

Table 13-2. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1)
PB5	SCK (SPI Bus Master clock Input)
PB4	MISO (SPI Bus Master Input/Slave Output)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2 (Timer/Counter2 Output Compare Match Output)
PB2	$\overline{SS}$ (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output)
PB0	ICP1 (Timer/Counter1 Input Capture Pin)

The alternate pin configuration is as follows:

- **XTAL2/TOSC2 – Port B, Bit 7**

XTAL2: Chip clock Oscillator pin 2. Used as clock pin for crystal Oscillator or Low-frequency crystal Oscillator. When used as a clock pin, the pin can not be used as an I/O pin.

TOSC2: Timer Oscillator pin 2. Used only if internal calibrated RC Oscillator is selected as chip clock source, and the asynchronous timer is enabled by the correct setting in ASSR. When the AS2 bit in ASSR is set (one) to enable asynchronous clocking of Timer/Counter2, pin PB7 is disconnected from the port, and becomes the inverting output of the Oscillator amplifier. In this mode, a crystal Oscillator is connected to this pin, and the pin cannot be used as an I/O pin.

If PB7 is used as a clock pin, DDB7, PORTB7 and PINB7 will all read 0.

- **XTAL1/TOSC1 – Port B, Bit 6**

XTAL1: Chip clock Oscillator pin 1. Used for all chip clock sources except internal calibrated RC Oscillator. When used as a clock pin, the pin can not be used as an I/O pin.

TOSC1: Timer Oscillator pin 1. Used only if internal calibrated RC Oscillator is selected as chip clock source, and the asynchronous timer is enabled by the correct setting in ASSR. When the AS2 bit in ASSR is set (one) to enable asynchronous clocking of Timer/Counter2, pin PB6 is disconnected from the port, and becomes the input of the Inverting Oscillator amplifier. In this mode, a crystal Oscillator is connected to this pin, and the pin can not be used as an I/O pin.

If PB6 is used as a clock pin, DDB6, PORTB6 and PINB6 will all read 0.

- **SCK – Port B, Bit 5**

SCK: Master Clock output, Slave Clock input pin for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB5. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB5. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB5 bit.

- **MISO – Port B, Bit 4**

MISO: Master Data input, Slave Data output pin for SPI channel. When the SPI is enabled as a Master, this pin is configured as an input regardless of the setting of DDB4. When the SPI is enabled as a Slave, the data direction of this pin is controlled by DDB4. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB4 bit.

- **MOSI/OC2 – Port B, Bit 3**

MOSI: SPI Master Data output, Slave Data input for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB3. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB3. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB3 bit.

OC2, Output Compare Match Output: The PB3 pin can serve as an external output for the Timer/Counter2 Compare Match. The PB3 pin has to be configured as an output (DDB3 set (one)) to serve this function. The OC2 pin is also the output pin for the PWM mode timer function.

- **$\overline{SS}$ /OC1B – Port B, Bit 2**

$\overline{SS}$: Slave Select input. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB2. As a Slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB2. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB2 bit.

OC1B, Output Compare Match output: The PB2 pin can serve as an external output for the Timer/Counter1 Compare Match B. The PB2 pin has to be configured as an output (DDB2 set (one)) to serve this function. The OC1B pin is also the output pin for the PWM mode timer function.

- **OC1A – Port B, Bit 1**

OC1A, Output Compare Match output: The PB1 pin can serve as an external output for the Timer/Counter1 Compare Match A. The PB1 pin has to be configured as an output (DDB1 set (one)) to serve this function. The OC1A pin is also the output pin for the PWM mode timer function.

- **ICP1 – Port B, Bit 0**

ICP1 – Input Capture Pin: The PB0 pin can act as an Input Capture Pin for Timer/Counter1.

Table 13-3 and Table 13-4 relate the alternate functions of Port B to the overriding signals shown in Figure 13-5 on page 54. SPI MSTR INPUT and SPI SLAVE OUTPUT constitute the MISO signal, while MOSI is divided into SPI MSTR OUTPUT and SPI SLAVE INPUT.

Table 13-3. Overriding Signals for Alternate Functions in PB7:PB4

Signal Name	PB7/XTAL2/ TOSC2 ⁽¹⁾⁽²⁾	PB6/XTAL1/ TOSC1 ⁽¹⁾	PB5/SCK	PB4/MISO
PUOE	EXT • (INTRC + AS2)	INTRC + AS2	SPE • MSTR	SPE • MSTR
PUO	0	0	PORTB5 • PUD	PORTB4 • PUD
DDOE	EXT • (INTRC + AS2)	INTRC + AS2	SPE • MSTR	SPE • MSTR
DDOV	0	0	0	0
PVOE	0	0	SPE • MSTR	SPE • MSTR
PVOV	0	0	SCK OUTPUT	SPI SLAVE OUTPUT
DIEOE	EXT • (INTRC + AS2)	INTRC + AS2	0	0
DIEOV	0	0	0	0
DI	—	—	SCK INPUT	SPI MSTR INPUT
AIO	Oscillator Output	Oscillator/Clock Input	—	—

Notes: 1. INTRC means that the internal RC Oscillator is selected (by the CKSEL Fuse).
2. EXT means that the external RC Oscillator or an external clock is selected (by the CKSEL Fuse).

Table 13-4. Overriding Signals for Alternate Functions in PB3:PB0

Signal Name	PB3/MOSI/OC2	PB2/SS/OC1B	PB1/OC1A	PB0/ICP1
PUOE	SPE • MSTR	SPE • MSTR	0	0
PUO	PORTB3 • PUD	PORTB2 • PUD	0	0
DDOE	SPE • MSTR	SPE • MSTR	0	0
DDOV	0	0	0	0
PVOE	SPE • MSTR + OC2 ENABLE	OC1B ENABLE	OC1A ENABLE	0
PVOV	SPI MSTR OUTPUT + OC2	OC1B	OC1A	0
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	SPI SLAVE INPUT	SPI SS	—	ICP1 INPUT
AIO	—	—	—	—

1.3 Alternate Functions of Port C

The Port C pins with alternate functions are shown in Table 13-5.

Table 13-5. Port C Pins Alternate Functions

Port Pin	Alternate Function
PC6	RESET (Reset pin)
PC5	ADC5 (ADC Input Channel 5) SCL (Two-wire Serial Bus Clock Line)
PC4	ADC4 (ADC Input Channel 4) SDA (Two-wire Serial Bus Data Input/Output Line)
PC3	ADC3 (ADC Input Channel 3)
PC2	ADC2 (ADC Input Channel 2)
PC1	ADC1 (ADC Input Channel 1)
PC0	ADC0 (ADC Input Channel 0)

The alternate pin configuration is as follows:

- **RESET – Port C, Bit 6**

RESET, Reset pin: When the RSTDISBL Fuse is programmed, this pin functions as a normal I/O pin, and the part will have to rely on Power-on Reset and Brown-out Reset as its reset sources. When the RSTDISBL Fuse is unprogrammed, the reset circuitry is connected to the pin, and the pin can not be used as an I/O pin.

If PC6 is used as a reset pin, DDC6, PORTC6 and PINC6 will all read 0.

- **SCL/ADC5 – Port C, Bit 5**

SCL, Two-wire Serial Interface Clock: When the TWEN bit in TWCR is set (one) to enable the Two-wire Serial Interface, pin PC5 is disconnected from the port and becomes the Serial Clock I/O pin for the Two-wire Serial Interface. In this mode, there is a spike filter on the pin to suppress spikes shorter than 50 ns on the input signal, and the pin is driven by an open drain driver with slew-rate limitation.

PC5 can also be used as ADC input Channel 5. Note that ADC input channel 5 uses digital power.

- **SDA/ADC4 – Port C, Bit 4**

SDA, Two-wire Serial Interface Data: When the TWEN bit in TWCR is set (one) to enable the Two-wire Serial Interface, pin PC4 is disconnected from the port and becomes the Serial Data I/O pin for the Two-wire Serial Interface. In this mode, there is a spike filter on the pin to suppress spikes shorter than 50 ns on the input signal, and the pin is driven by an open drain driver with slew-rate limitation.

PC4 can also be used as ADC input Channel 4. Note that ADC input channel 4 uses digital power.

- **ADC3 – Port C, Bit 3**

PC3 can also be used as ADC input Channel 3. Note that ADC input channel 3 uses analog power.

- **ADC2 – Port C, Bit 2**

PC2 can also be used as ADC input Channel 2. Note that ADC input channel 2 uses analog power.

- **ADC1 – Port C, Bit 1**

PC1 can also be used as ADC input Channel 1. Note that ADC input channel 1 uses analog power.

• **ADC0 – Port C, Bit 0**

PC0 can also be used as ADC Input Channel 0. Note that ADC input channel 0 uses analog power.

Table 13-6 and Table 13-7 relate the alternate functions of Port C to the overriding signals shown in Figure 13-5 on page 54.

Table 13-6. Overriding Signals for Alternate Functions in PC6:PC4

Signal Name	PC6/RESET	PC5/SCL/ADC5	PC4/SDA/ADC4
PUE	RSTDISBL	TWEN	TWEN
PUEV	1	PORTC5 • PUD	PORTC4 • PUD
DUE	RSTDISBL	TWEN	TWEN
DUEV	0	SCL_OUT	SDA_OUT
PUE	0	TWEN	TWEN
PUEV	0	0	0
DUE	RSTDISBL	0	0
DUEV	0	0	0
DI	–	–	–
AIO	RESET INPUT	ADC5 INPUT / SCL INPUT	ADC4 INPUT / SDA INPUT

Table 13-7. Overriding Signals for Alternate Functions in PC3:PC0⁽¹⁾

Signal Name	PC3/ADC3	PC2/ADC2	PC1/ADC1	PC0/ADC0
PUE	0	0	0	0
PUEV	0	0	0	0
DUE	0	0	0	0
DUEV	0	0	0	0
PUE	0	0	0	0
PUEV	0	0	0	0
DUE	0	0	0	0
DUEV	0	0	0	0
DI	–	–	–	–
AIO	ADC3 INPUT	ADC2 INPUT	ADC1 INPUT	ADC0 INPUT

Note: 1. When enabled, the Two-wire Serial Interface enables slew-rate controls on the output pins PC4 and PC5. This is not shown in the figure. In addition, spike filters are connected between the AIO outputs shown in the port figure and the digital logic of the TWI module.

3.4 Alternate Functions of Port D

The Port D pins with alternate functions are shown in Table 13-8.

Table 13-8. Port D Pins Alternate Functions

Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input)
PD6	AIN0 (Analog Comparator Positive Input)
PD5	T1 (Timer/Counter 1 External Counter Input)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

The alternate pin configuration is as follows:

- **AIN1 – Port D, Bit 7**

AIN1, Analog Comparator Negative Input. Configure the port pin as input with the internal pull-up switched off to avoid the digital port function from interfering with the function of the Analog Comparator.

- **AIN0 – Port D, Bit 6**

AIN0, Analog Comparator Positive Input. Configure the port pin as input with the internal pull-up switched off to avoid the digital port function from interfering with the function of the Analog Comparator.

- **T1 – Port D, Bit 5**

T1, Timer/Counter1 counter source.

- **XCK/T0 – Port D, Bit 4**

XCK, USART external clock.

T0, Timer/Counter0 counter source.

- **INT1 – Port D, Bit 3**

INT1, External Interrupt source 1: The PD3 pin can serve as an external interrupt source.

- **INT0 – Port D, Bit 2**

INT0, External Interrupt source 0: The PD2 pin can serve as an external interrupt source.

- **TXD – Port D, Bit 1**

TXD, Transmit Data (Data output pin for the USART). When the USART Transmitter is enabled, this pin is configured as an output regardless of the value of `DDD1`.

- **RXD – Port D, Bit 0**

RXD, Receive Data (Data input pin for the USART). When the USART Receiver is enabled this pin is configured as an input regardless of the value of `DDD0`. When the USART forces this pin to be an input, the pull-up can still be controlled by the `PORTD0` bit.

Table 13-9 and Table 13-10 relate the alternate functions of Port D to the overriding signals shown in Figure 13-5 on page 54.

Table 13-9. Overriding Signals for Alternate Functions PD7:PD4

Signal Name	PD7/AIN1	PD6/AIN0	PD5/T1	PD4/XCK/T0
PUOE	0	0	0	0
PUO	0	0	0	0
OOE	0	0	0	0
OO	0	0	0	0
PVOE	0	0	0	UMSEL
PVO	0	0	0	XCK OUTPUT
DIEOE	0	0	0	0
DIEO	0	0	0	0
DI	—	—	T1 INPUT	XCK INPUT / T0 INPUT
AIO	AIN1 INPUT	AIN0 INPUT	—	—

Table 13-10. Overriding Signals for Alternate Functions in PD3:PD0

Signal Name	PD3/INT1	PD2/INT0	PD1/TXD	PD0/RXD
PUOE	0	0	TXEN	RXEN
PUO	0	0	0	PORTD0 • PUD
OOE	0	0	TXEN	RXEN
OO	0	0	1	0
PVOE	0	0	TXEN	0
PVO	0	0	TXD	0
DIEOE	INT1 ENABLE	INT0 ENABLE	0	0
DIEO	1	1	0	0
DI	INT1 INPUT	INT0 INPUT	—	RXD
AIO	—	—	—	—

4 Register Description

4.1 PORTB – The Port B Data Register

Bit	7	6	5	4	3	2	1	0	
	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

4.2 DDRB – The Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

4.3 PINB – The Port B Input Pins Address

Bit	7	6	5	4	3	2	1	0	
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

4.4 PORTC – The Port C Data Register

Bit	7	6	5	4	3	2	1	0	
	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

4.5 DDRC – The Port C Data Direction Register

Bit	7	6	5	4	3	2	1	0	
	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

4.6 PINC – The Port C Input Pins Address

Bit	7	6	5	4	3	2	1	0	
	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

4.7 PORTD – The Port D Data Register

Bit	7	6	5	4	3	2	1	0	
	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

4.8 DDRD – The Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
	DDR7	DDR6	DDR5	DDR4	DDR3	DDR2	DDR1	DDR0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

4.9 PIND – The Port D Input Pins Address

Bit	7	6	5	4	3	2	1	0	
	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	



Enabling Unlimited Possibilities®

Atmel Corporation
210 Technology Drive
Folsom, CA 95110
USA
Tel: (+1) (408) 441-0311
Tel: (+1) (408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-8100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194821

Atmel Japan G.K.
16F Shin-Osaki Kangyo Bldg
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN
Tel: (+81) (3) 6417-0300
Fax: (+81) (3) 6417-0370

©2013 Atmel Corporation. All rights reserved. / Rev.: 8159E-AVR-02/2013

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

