LAPORAN TUGAS AKHIR

LENGAN ROBOT PEMISAH BOLA BERWARNA BERBASIS MIKROKONTROLER AT89C51

Disusun oleh:

EXZAN TRIYONO

0152065

KONSENTRASI TEKNIK ENERGI LISTRIK

JURUSAN TEKNIK ELEKTRO DIII

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

MARET 2007

LAPORAN AKHIR TUGAS AKHIR

PERBANDINGAN MIKROKONTROLER TIPE ATMEGA 16 DENGAN ATMEGA 8535 PADA PENERAPAN ROBOT PENJEJAK BOLA BERWARNA

Oleh :

ENDAH ARUMONO

3100210

Dosen Pembimbing :

KONSERTIANSI TEKNIK EMERCI PGRIDK

MUHAJIR TEKNIK ELEKTRO DIII

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

2006 TAHUN

LEMBAR PERSETUJUAN

LENGAN ROBOT PEMISAH BOLA BERWARNA BERBASIS MIKRO KONTROLER AT89C51

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat Untuk Memperoleh Gelar Pada Teknik Elektro D III Konsentrasi Energi Listrik

Disusun oleh:

EXZAN TRIYONO

01.52.065

Mengetahui	Di periksa dan Disetujui oleh
Ketua jurusan Elektro DIII	Dosen Pembimbing
Ir. H.CHOIRUL SALEH, MT	**Ir TEGUH HERBASUKI, MT**
NIP.Y. 1010088190	NIP . P . 1038900209

KONSENTRASI TEKNIK ENERGI LISTRIK

JURUSAN TEKNIK ELEKTRO DIII

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLIGI NASIONAL MALANG

2007

KATA PENGANTAR

Dengan memanjatkan puja dan puji syukur kehadirat Tuhan Yang Maha Esa atas segala rahmat-Nya, sehingga proses penyusunan Laporan Tugas Akhir yang berjudul Lengan Robot Pemisah Bola Berwarna Berbasis Mikrokontroler AT89C51 dapat terselesaikan dengan baik.

Dengan terselesaikannya penyusunan Tugas Akhir ini, penyusun mengucapkan banyak terimakasih kepada:

1. Prof . Dr. Ir. Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang.
2. Ir. Choirul Saleh, MT selaku Ketua Jurusan Elektro DIII Institut Teknologi Nasional Malang.
3. Ir. Teguh Herbasuki, MT selaku Dosen Pembimbing yang telah membimbing penyusun dalam menyelesaikan Tugas Akhir ini.
4. Para rekan-rekan Mahasiswa dan semua pihak yang telah membantu dalam menyelesaikan Tugas Akhir disini.

Penyusun menyadari bahwa dalam penyusunan laporan Tugas Akhir disini masih banyak kekurangan dan kelemahan, oleh karena itu kritik dan saran yang membangun sangat kami harapkan, guna perbaikan di masa yang akan datang.

Malang, April 2007

Penulis

ABSTRAK

Exzan, Triyono 2007. **Lengan Robot Pemisah Bola Berwarna Berbasis Mikrokontroler AT89C51**. Tugas Akhir, Konsentrasi Teknik Energi Listrik, Jurusan Teknik Elektro D III, Fakultas Teknologi Industri, Institut Teknologo Nasional Malang. Pembimbing Ir. Teguh Herbasuki, MT.

Kata kunci: Mikrokontroler AT89C51, ADC, Sensor.

Kemajuan teknologi saat ini telah merambah kesegala bidang, sebagian orang menginginkan segala sesuatu dapat dikerjakan dengan mudah, otomatis, cepat dan akurat. Mikrokontroler merupakan salahsatu teknologi yang dapat mewujudkannya. Oleh karena itu telah dibuat prototype lengan robot Pemisah bola berwarna berbasis mikrokontroler AT89C51. alat ini terdiri dari beberapa bagian, yatu bagian sensor bagian ADC, bagian Driver motor DC, bagian drver motor Stepper, bagian catu daya dan mikrokontroler AT89C51 yang deprogram dengan menggunakan bahasa assembler.

Alat ini bekerja mengambil bola berwarna dan kemudian meletakkan ketempat yang sesuai dengan warna benda yang diambil, berdasarkan data atau instruksi program yang dimasukkan dalam mikrokontroler. Dengan memanfaatkan mikrokontroler maka secara bergantian putaran motor akan menggerakkan lengan untuk turun kemudian mencengkram bola setelah itu lengan akan kembali naik dan berputar menuju tempat dengan warna yang sama seperti seperti warna bola yang diambil jika lwngan telah berada didepan tempat yang dituju akan kembali turun membuka cengkraman kemudian naik dan kembali keposisi semula.

Dari hasil pengujian alat ini maka dapat disimpulkan bahwa alat dibuat belum sempurna dikarenakan sensor yang digunakan adalah LDR, sehingga kurang mampu membedakan kecerahan warna yang sama atau hamper sama, hal ini disebabkan LDR hanya membedakan tingkat intensitas pantulan cahaya dari bola berwarna. Untuk pengembangan alat yang lebih baik lagi hendaknya menggunakan sensor warna.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

INSTITUT TEKNOLOGI NASIONAL

ITN

MALANG

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dewasa ini perkembangan teknologi dunia robot memang sangat pesat, penggunaan robot untuk membantu pekerjaan manusia sudah suatu hal yang umum saat ini, dan tidak sedikit pula para ilmuan-ilnuan muda yang menciptakan robot hanya sekedar untuk mewujutkan kreatifitas semata. Saat ini banyak ilmuan-ilmuan muda yang berlomba untuk menciptakan robot yang lebih inovatif,canggih dan yang bermanfaat bagi manusia, dan tidak sedikit pula robot yang di gunakan dalam pekerjaan industri-industri.

Dalam industri atau suatu pabrik, sebenarnya akan lebih efisien dan lebih menekan biaya produksi menjadi lebih murah apabila dalam pekerjaan-pekerjaan yang membutuhkan ketelitian dan proses yang cepat,dapat di lakukan oleh robot yang telah di seting dan di program untuk menjalankan pekerjaan tersebu,sehingga kan lebih menguntungkan dalam proses produksi. Karena kecanggihan teknologi saat ini maka pekerjaan manusia kan lebih di mudahkan karena adanya teknologi robot tersebut.

Salah satu permasalahan yang mendasar adalah bagaimana cara menciptakan robot yang sesuai dengan keinginan suatu proses produksi

atau suatu pekerjaan yang lebih cepat dan lebih efisien. Serta bagaimana merancang robot dan program dari robot tersebut.

Disini untuk mengatasi tersebut maka, banyak para ilmuan yang berlomba-lomba untuk menciptakan robot yang sesuai dengn keinginan, dan banyaak juga sekarang penemuan-penemuan baru tentang robot yang bias di pakai dalam industri-industri.

1.2. Permasalahan

Permasalahan yang timbul dalam penyusunan tugas akhir ini, antara lain:

1. Bagaimana merancang rangkaian mekanik robot tersebut.
2. Bagaimana merancang program yang akan di jalankan oleh robot tersebut.
3. Bagaimana cara pengoprasian robot apakah sesuai dengan progaram.

1.3. Tujuan

Tujuan dari penyusunan tugas akhir ini adalah merancang dan membuat Lengan Robot Pemisah Bola Berwarna Berbasis Mikrokontroler TA89C51.

1.4. Batasan Masalah

Untuk membatasi ruang lingkup pembahasan dengan pertimbangan waktu, dana dan kemampuan intelektual, di beri batasan-batasan sebagai berikut:

1. Pokok masalah utama adalah perwujudan rangkaian mekanik dari robot tersebut.
2. Mikrokontroler yang di gunakan adalan IC AT89C51
3. Robot tersebut hanya mampu memisahkan 3 warna yaitu putih, kuning dan hitam.

1.5. Manfaat.

Dengan di buatnya lengan robot pemisah bola berwarna berbasis mikrokontroler AT89C51 di harapkan dapat di pakai untuk modul salah satu praktikum matakuliah di ITN Malang.

1.6. Metodologi Penelitian

Pembuatan tugas akhir ini menyangkut beberapa proses yang saling berurutan dan berkaitan : penelitian, studiy literatut, percobaan dan pembuatan alat. Untuk mewujudkan proses di atas maka metode-metode yang penulis tempuh guna mendapatkan data tersebut adalah sebagai berikut:

1. Metode Penelitian

   Metode ini di lakukan dengan meneliti kehidupan sehari-hari dalam suatu proses produksi yang masih menggunakan proses manual, sehingga timbulah pemikiran untuk membuat alat tersebut.

2. Metode Literatur

   Metode ini di lakukan penulis dengan cara membaca buku-buku pendukung yang berkaitan dengan pembuatan alat.

3. Metode Interview

   Penulis melakukan Tanya jawab langsung denagn dosen yang mengacu pada mata kuliah tertentu yang erat kaitannya dengan kegiatan membuat alat.

4. Metode Eksperimen

Penulis melakukan serangkaian eksperimen terhadap rangkaian-rangkaian dasar serta uji coba software untuk mendapatkan system yang benar-benar setabil dan berfungsi dengan baik

1.7. Sistematika Tugas Akhir

Untuk mempermudah dalam penyusunan laporan tugas akhir ini maka penulis membuat laporan materi secara sistematis. Adapun sistematika penulisan laporan adalah sebagai berikut:

BAB I PENDAHULUAN

Berisi tentang Latarbelakang, Permasalahan, Batasan masalah, Tujuan, Manfaat, Sumberdata dan Sistematika Penulisan Tugas Akhir.

BAB II LANDASAN TEORI

Berisi tentang komponen-komponen pendukung dalam Tugas Akhir ini, antaralain Mikrokontroler AT89C51 dan lain-lain.

BAB III PERANCANGAN LENGAN ROBOT PEMISAH BOLA BERWARNA BERBASIS MIKROKONTROLER AT89C51

Berisi Blok Diagram, cara kerja Mikrokontroler AT89C51 dan Software program.

BAB IV PENGUJIAN ALAT

Berisi tentang hasil pengujian keluaran mikrokontroler AT89C51 dalam mengontrol lengan robot ini.

BAB V PENUTUP

Berisi tentang kesimpulan dan saran dari tugas akhir, bab ini juga berisi tentang daftar pustaka dari buku-buku yang di jadikan referensi dan lampiran-lampiran yang menunjang terwujudnya alat.

BAB II

LANDASAN TEORI

2.1. Pengertian Tentang Robot

Kata "robot" berasal bahasa Chech (Ceko) yang berarti pekerja. Saat ini, secara sadar atau tidak, robot memang telah hadir di dalam kehidupan manusia dalam bentuk yang bermacam-macam. Terdapat bentuk desain robot yang sederhana untuk mengerjakan kegiatan yang mudah atau berulang-ulang. Ada pula robot yang dirancang untuk "berperilaku" sangat kompleks dan sampai batas tertentu dapat mengontrol dirinya sendiri. Di kalangan umum pengertian robot selalu dikaitkan dengan "makhluk hidup" berbentuk orang maupun binatang yang terbuat dari logam dan bertenaga listrik. Sementara itu dalam arti luas robot berarti alat yang dalam batas-batas tertentu dapat bekerja sendiri (otomatis) sesuai dengan perintah yang sudah diberikan oleh perancangnya. Dengan pengertian ini sangat erat hubungan antara robot dan otomatisasi sehingga dapat dipahami bahwa hampir setiap aktivitas kehidupan modern makin tergantung pada robot dan otomatisasi.

2.2. Pengenalan Mikrokontroler

Mikrokontroler merupakan kombinasi CPU *(Central Processing Unit)* dengan memori dan I/O (Input / Output) dalam level chip, yang menghasilkan *Single Chip Mikrokomputer* (SCM). Pada mikrokontroler

digunakan EPROM *(Erasable Programable Read Only Memory)* sebagai media simpan programnya. Karena pada mikrokontroler sudah ada ROM *(Read Only Memory)* dan peralatan I/O pendukung, sehingga tidak perlu menambahkannya lagi.

Sebuah mikrokontroler tidak dapat bekerja bila tidak diberikan program kepadanya. Karena program tersebut yang memberi perintah dan memberitahu mikrokontroler apa yang harus dilakukan. Program untuk mengendalikan kerja dari mikrokontroler disimpan dalam memori program yang merupakan kumpulan dari instruksi kerja mikrokontroler. Mikrokontroler memiliki konponen-komponen dasar sebagai berikut :

1). CPU *(Central Processing Unit)*

CPU terdiri dari dua bagian, yaitu unit pengendali (*Control Unit*/CU) serta unit aritmatika dan logika (*Arithmatic and Logic Unit*/ ALU). CU berfungsi untuk mengambil *(fetch),* mengkode *(decode)* dan melaksanakan urutan instruksi *(execute)* sebuah program yang tersimpan dalam memori, juga mengatur urutan operasi seluruh sistem melalui bus data, bus alamat, dan bus kendali.

a). *Address bus* (Bus Alamat)

Bus alamat (16 bit) digunakan untuk memilih peralatan (memori dan peralatan Input/ output) yang terpasang dalam suatu sistem komputer. Apabila suatu alat dihubungkan dengan mikrokontroler maka harus ditetapkan dahulu alamat *(address)* dari alat tersebut.

Hal ini bertujuan untuk menghindari terjadinya dua alat yang bekerja secara bersamaan, yang bisa menyebabkan kesalahan atau kerusakan.

b). Data bus (Bus *Data*)

Bus data (8 bit), digunakan untuk membaca instruksi dari program memori (EPROM), membaca dan menulis dari memori data (RAM) atau peralatan input/output lainnya.

c). *Control Bus* (Bus Pengendali)

Selain bus alamat, bus data, mikrokontroler juga memiliki bus pengendali (*Control Bus*). yang berfungsi unutk menyerempakkan operasi mikrokontroler dengan peralatan lainnya (memori dan input/output).

2). *Memory*

Pada suatu sistem mikrokontroler maupun sistem komputer memori digunakan untuk menyimpan program atau data.

a). RAM *(Random Access Memory)*

Merupakan memori yang dapat dibaca atau ditulisi, dan digunakan untuk menyimpan data sementara. Data pada RAM akan terhapus bila catu daya dihilangkan.

b). ROM *(Read Only Memory*

ROM merupakan memori yang hanya dapat dibaca. Data dalam ROM tidak akan terhapus meski catu daya diputus. Ada beberapa tipe ROM diantaranya EPROM (*Erasable Programable Read Only Memory)*, EEPROM *(Electricaly Erasable Programable Read Only Memory*).

3). Input / Output

Untuk melakukan hubungan dengan piranti di luar sistem, dibutuhkan peralatan Input/Output (I/O). Peralatan I/O dapat mengirim data keluar sistem dan menerima data dari luar sistem ke dalam mikrokontroler. Ada 2 macam perantara I/O yaitu serial dan paralel. UART (*Universal Asynchronous Receiver Transmitter*) mengubah masukan serial menjadi keluaran paralel dan mengubah masukan paralel menjadi keluaran serial. Sedangkan PIO *(Programable Input/Output)* merupakan alat yang dapat diprogram untuk perantara masukan dan keluaran dalam format paralel.

2.3. Mikrokontroler MCS51

Salah satu jenis mikrokontroler yang sangat populer adalah keluarga MCS51. Sampai saat ini sudah ada lebih dari 100 macam mikrokontroler turunan 8051, sehingga terbentuk sebuah ' keluarga besar mikrokontroler' yang kemudian biasa disebut sebagai keluarga MCS51.

ATMEL menambahkan anggota keluarga MCS51 dengan memproduksi 2 macam mikrokontroler MCS51, yaitu 40 pin (AT89C51, AT89C52, AT89C53, AT89C55 dan AT89C8252) dan 20 pin (AT80C1051, AT89C2051, AT89C4051). ATMEL mempermudah penggunanya dengan memproduksi mikrokontroler dengan teknologi ISP *(In System Programming)* yang dapat diprogram secara serial. (AT89S51, AT89S52, AT89S53, AT89S8252).

2.3.1. Register MCS51

Pada mikrokontroler keluarga MCS-51 semua terdapat register baku yang harus dimiliki oleh setiap mikroprosesor atau mikrokontroler, register tersebut antara lain adalah sebagai berikut :

1). Akumulator (Acc / A)

Acc atau akumulator merupakan register untuk penyimpanan data sementara. Akumulator digunakan untuk pengolahan data, yaitu manipulasi data, operasi aritmatika dan logika.

2). Register B (B)

Biasanya digunakan bersama register A untuk operasi aritmatika.

3). Program Status Word (PSW)

Program Status Word sering juga disebut *Flag Register* yang berisikan informasi status program yang dihasilkan setelah suatu instruksi dijalankan.

4). Stack pointer (SP)

Register ini berfungsi menyimpan alamat stack. Stack adalah memori yang digunakan untuk menyimpan alamat/data pada saat terjadi interupsi, mengeksekusi perintah call, reti, push, dan pop.

5). Program Counter Register (PC)

Register 16 bit, berfungsi untuk menyimpan alamat program yang sedang dijalankan.

6). Register 0-7 (R0-R7)

Register serba guna 8 bit yang dapat dipilih mulai bank 0, 1, 2 sampai bank 3. Khusus R0 dan R1 dapat digunakan untuk pengalamatan tak langsung (indirect).

7). Data Pointer Register (DPTR)

Register pointer sepanjang 16 bit berguna untuk menyimpan alamat yang datanya akan dibaca oleh mikrokontroler. Register ini dapat dipecah menjadi Data Pointer High Byte (DPH) dan Data Pointer Low Byte (DPL) yang masing-masing merupakan register dengan kapasitas 8 bit.

8). Interrupt Enable Register (IE)

Untuk mengatur sumber interupsi yang akan diaktifkan dengan cara membuat bit yang bersesuaian menjadi 1 (set). Untuk mengaktifkan interupsi, bit Enable All (EA) harus diset lebih dulu.

Sumber interupsi tersebut adalah External Interupt 0 (EX0), External Interupt 1 (EX1), Timer 0 (ET0), Timer 1(ET1), dan Serial Interupt (ES).

Sumber interupsi tersebut adalah sebagai berikut :

Table 2.1 Sumber Interupsi

Sumber Interupsi	Vector Interupsi
External Interupt 0	03 H
Timer/Counter 0	0B H
External Interupt 1	13 H
Timer/Counter 1	1B H
Serial Port	23 H

9). Interupt Priority Register (IP)

Interupsi dengan prioritas tinggi tidak dapat diinterupsi yang lebih rendah. Bit-bit pada register IP adalah Priority Serial (PS), Priority Timer 1 (PT 1), Priority External Intrupt 1 (PX 1), PT 0 dan PX 0. Nilai 1 (set) menunjukan prioritas lebih tinggi.

10). Timer Mode Register (TMOD)

Mengatur mode pemakaian timer / counter sesuai bit-bit berikut :

MSB LSB

Gate	C/T	M1	M0	Gate	C/T	M1	M0

Gate digunakan untuk mengatur kerja timer / counter secara hardware atau software. Jika gate diset, timer bekerja pada saat TRx = 1 dan kaki interrupt x berlogika 1, sehingga dapat digunakan untuk mengukur lebar pulsa. Jika *Gate* direset, timer bekerja saat TRx diset. M0 dan M1 digunakan untuk mengatur mode kerja timer.

Tabel 2.2 Mode Kerja Timer

M1	M0	Mode	Operasi
0	0	0	Timer/Counter 13 bit
0	1	1	Timer/Counter 16 bit
1	0	2	Timer auto reload 8 bit
1	1	3	TL0 = timer/counter 8 bit dikontrol TR0 TH0 = timer counter 8 bit dikontrol TR1

11). Timer Control Register (TCON)

Digunakan untuk menjalankan (setb TRx) dan menghentikan (clr TRx) timer x, juga berfungsi untuk mengatur sisi turun (setb ITx) atau level rendah (clr ITx) dari trigger yang mengaktifkan external interrupt x. TFx adalah timer x overflow flag, IEx adalah External interrupt x edge flag.

12). *Timer High Byte Register* (THx) dan *Timer Low Byte Register* (TLx)

Mengatur waktu tunda timer :

Timer 16 bit. $T = (2^{16} - (256 * THx + TLx)) * 12/fosc$

Timer 8 bit. $T = (2^{8} - TLx) * 12/fosc$

13). Serial Control Register (SCON)

Mengatur mode kerja serial port :

MSB LSB

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Bit SMO dan SM1 mengatur mode, SM2 mengatur komunikasi multi processor. Jika SM2 = 1, pada mode 2 dan 3 RI tidak diaktifkan jika data ke 9 (RB8) berlogika 0, pada mode 1 RI tidak diaktifkan jika stop bit tidak diterima. Pada mode 0, SM2 harus berlogika 0. REN (Receive Enable), jika diset, mengaktifkan penerimaan (RX). Bit TB8 adalah data ke 9 yang akan dikirim pada mode 2 dan 3. RB8 adalah data ke 9 yang diterima pada mode 2 dan 3. TI *(Transmit Interupt Flag)* diset oleh hardware yang menandakan pengiriman telah selesai, direset secara software. RI *(Receive Interupt Flag)* diset oleh hardware yang menandakan data telah diterima secara utuh.

14). Serial Data Buffer (SBUFF)

Adalah register untuk menerima dan mengirimkan data serial yang secara fisik terpisah, sehingga dapat melakukan transmisi data serial secara full duplex.

15). Power Control Register (PCON)

Digunakan untuk mengatur mode penghematan daya dan pengaturan baudrate.

Jika SMOD = 1, dan timer 1 digunakan sebagai baudrate generator, maka baudrate menjadi 2 kali lipat pada mode 1, 2 dan 3. GF1 dan GF2 adalah General Purpose Flag bit. PD (Power Down bit) jika diset, mikrokontroler masuk ke mode Power Down. IDL (Idle Mode Bit) jika diset, Mikrokontroler masuk ke mode idle.

Register-register diatas kecuali R0-R7, disimpan dalam *Special Function Register* (SFR) pada RAM internal dengan alamat 80H sampai FFH. Berikut adalah denah alamat register SFR beserta nilai defaultnya :

Table 2.3 Special Function Register MCS51

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0O8H									0CFH
0O0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000								0AFH
0A0H	P2 11111111		AUXR1 XXXXXXX0				WDTRST XXXXXXXX		0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0		8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XXX0000	87H

2.3.2. Format Bahasa Program MCS51

Program bahasa assembly berisikan instruksi mesin, pengarah assembler, kontrol assembler, dan komentar. Baris-baris program yang mengandung instruksi mesin atau pengarah assembler harus mengikuti aturan program assembler ASM51. Format umum bahasa program ASM51 adalah sebagai berikut :

[label:] mnemonic [operan] [operan] [...] [;komentar]

1). Label

Sebuah label mewakili alamat dari instruksi (atau data) yang mengikat. Label ini digunakan sebagai operan pada instruksi-instruksi percabangan (misal : SJMP TERUS). Simbol dan label adalah dua hal yang berbeda. Simbol tidak menggunakan titik dua, sedangkan label menggunakan. Contoh :

PAR EQU 500	;PAR adalah suatu simbol dari nilai 500
START: MOV A,#0FFH	;START adalah label yang menunjuk pada lokasi instruksi MOV.

2). Mnemonic

Mnemonic instruksi atau pengarah assembler dimasukkan dalam *"Mnemonic Field"* yang mengikuti *"Label Mnemonic"*. Mnemonic instruksi misalnya ADD, MOV, INC, dan lain-lain. Sedangkan pengarah assembler misalnya ORG, EQU, DB, dan lain-lain.

3). Operan

Operan ditulis setelah mnemonic, bisa berupa alamat atau data yang digunakan instruksi yang bersangkutan. Bisa juga merupakan label yang mewakili alamat suatu data atau berupa simbol yang mewakili suatu data konstanta. Ada beberapa instruksi MCS51 yang tidak memerlukan operan diantaranya adalah RET dan lain-lain.

4). Komentar.

Komentar harus diawali dengan titik koma (;). Suatu baris atau bagian dari suatu baris akan dianggap suatu komentar jika diawali dengan titik koma.

2.3.3. Addressing Modes MCS51

Mode pengalamatan pada mikrokontroler MCS51 dapat dikelompokkan sebagai berikut :

1). Immediate Constant.

MOV A,#DATA

Nilai constant (#DATA), mengikuti opcode pada program memori.

Contoh :

```
MOV A, #128          ;register A diisi data 128 desimal (80H).
MOV DPTR,#1234H      ;register DPTR (16 bit) diisi data 1234H.
```

2). Direct Addressing.

```
MOV A,Daddr          ;A diisi data dari RAM pada alamat Daddr
```

Operand yang ditunjukkan oleh 8 bit alamat (Daddr) pada instruksi, sehingga hanya data pada RAM internal dan SFR yang dapat dialamati secara direct.

MOV 20H,#255 ; RAM alamat 20H diisi data 255 desimal (FFH).

MOV A,20H ; Register A diisi data dari RAM alamat 20H, sehingga A = 255.

3). Indirect Addressing.

MOV A,@Ri

MOVX A,@Ri

MOVX A,@DPTR

Alamat operand yang ditunjukkan oleh register Ri. RAM internal dan RAM eksternal dapat dialamati dengan cara ini.

Register 8 bit yang dapat digunakan adalah R0 dan R1, sehingga yang 16 bit hanya dapat digunakan register DPTR.

Contoh :

MOV 50HM,#100 ;alamat memori 100H diisi data 100 desimal.

MOV R0,#50H ;register R0 data 50H

MOV A,@R0 ;register A diisi data dari alamat yang ditunjuk R0 (50H), sehingga A=100.

4). Register Instruction.

MOV A,Rn

Operan diambil dari register R0 – R7. Pada mode ini, pemakaian memori program sangat efisien, karena instruksi tidak perlu menyebutkan alamat operand.

Contoh :

MOV R7,#10H ;register R7 diisi data 10H

MOV A,R7 ;register A diisi data dari R7, sehingga A = 10H.

5). Indexed Addressing.

MOVC A,@A+DPTR

MOVC A,@A+PC

Hanya program memori yang dapat diakses dengan cara ini dan hanya bisa dibaca. Biasanya digunakan untuk membaca "look up table" atau instruksi "case jump".

Register DPTR atau PC dapat digunakan untuk base address dan accumulator digunakan untuk data keberapa yang akan dibaca, dengan cara menjumlahkan base address dengan isi akumulator.

Contoh:

MOV DPTR,#TBL ;register TBL diisi TBL

Mov A,#5 ;register A diisi 5

MOVC A, @A+DPTR ;register A diisi data dari program memori dengan alamat (DPTR+A), sehingga A = 15.

TBL: DB 10,11,12,13,14,15,16,17,18,19

2.4. Mikrokontroler AT89C51

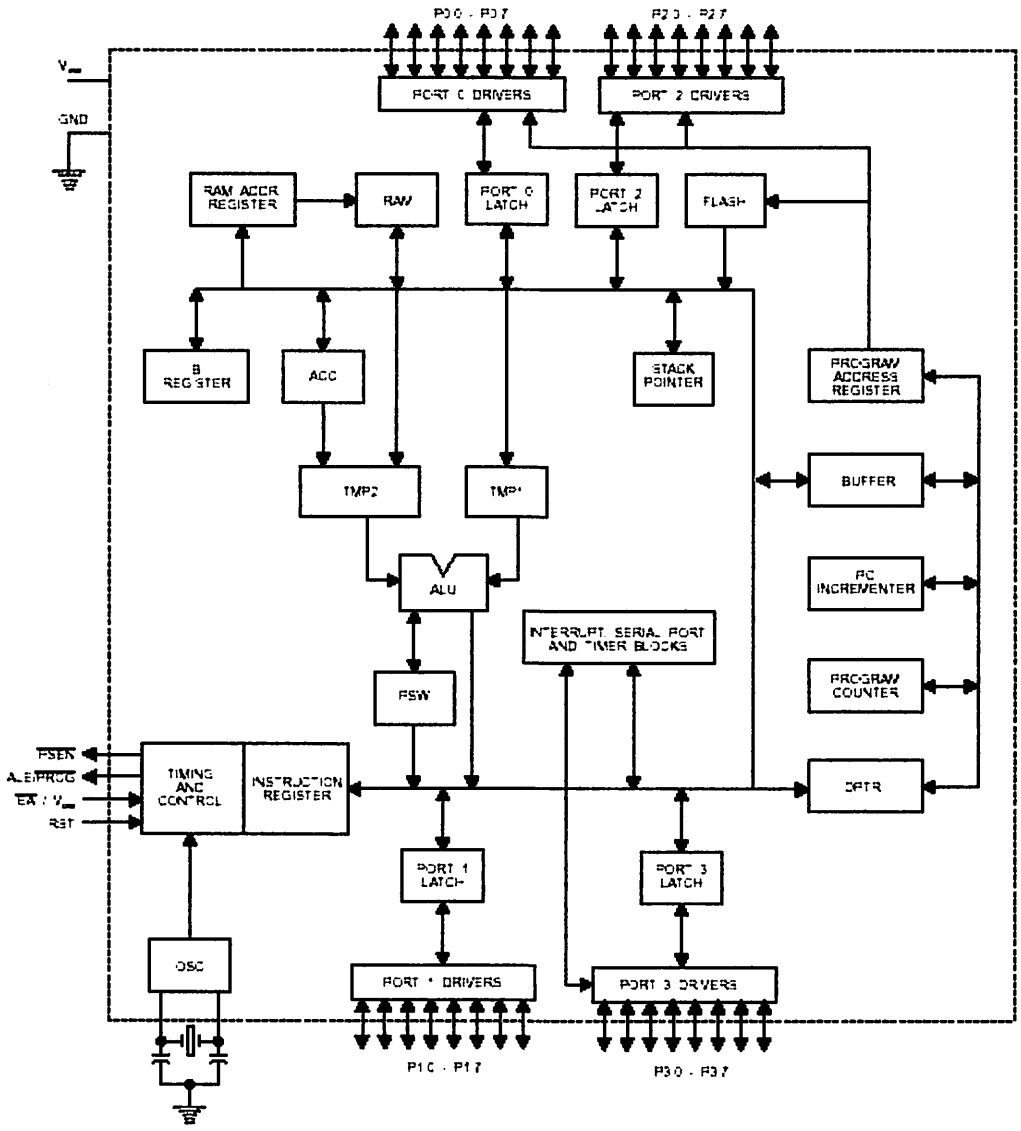
Mikrokontroler AT89C51 merupakan salah satu anggota keluarga dari mikrokontroler 8051, sehingga memiliki persamaan tipe yaitu pemroses data pusat (central Processing Unit), memori baca (Random Access Memory), dan Pewaktu (Timer / Counter), Pintu Paralel (Paralel Port) dan Pintu serial (Serial Port).

Keistimewaan dari Mikrokontroler AT89C51 yaitu :

Sebuah Central Processing Unit 8 bit :

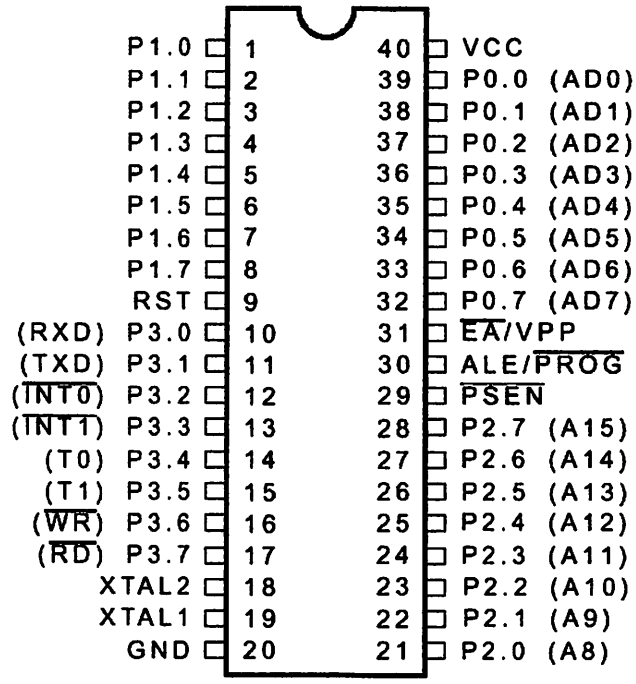
1. Oscilator dalam dan rangkaian pewaktu
2. Ram dan Rom 4Kb
3. Empat buah pintu masukan / keluaran, masing – masing terdiri dari delapan jalur masukan / keluaran.
4. Dua buah pewaktu 16 bit.
5. Lima buah jalur penyelaan luar
6. Enam buah pintu serial dengan UART (Universal Asyncronous Receiver Transmitter) hubungan penuh (Full duplex).
7. Kemampuan melaksanakan operasi perkalian, pembagian dan operasi Boolean.
8. Kecepatan waktu pelaksanaan perintah tiap periode adalah 1 mikro detik pada frekwensi kerja 1 Mhz.

Berikut adalah gambar blok diagram dari mikrokontroler AT89C51:



Gambar 2.1 Blok Diagaram Mikrokontroler AT89C51

Susunan pin Mikrokontroler AT89C51 dapat dilihat pada gambar berikut :



Gambar 2.2. Susunan Pin AT89C51

1). Pin 1 sampai 8

Pin ini merupakan pintu Paralel 8 bit dua buah yang dapat digunakan untuk berbagai keperluan.

2). Pin 9

Pin ini merupakan masukan reset (Aktive tinggi). Pulsa transisi dari daerah rendah ke tinggi akan mereset mikrokontroler AT89C51. Pin ini dihubungkan dengan rangkaian power on reset.

3). Pin 10 sampai 17

Pin ini adalah pintu Paralel 8 bit dua arah (I / O) yang memiliki fungsi pengganti. Bila fungsi pengganti tidak dipakai, Pin dapat digunakan sebagai pintu Paralel 8 bit serba guna.

4). Pin 18

Pin ini merupakan Pin masukan ke rangkaian osilator dalam. Sebuah osilator kristal atau sumber osilator luar yang dapat digunakan.

5). Pin 19

Pin ini merupakan Pin keluaran dari rangkaian osilator dalam. Pin ini dipakai bila menggunakan osilator kristal.

6). Pin 20

Dihubungkan ke Ground.

7). Pin 21 sampai 28

Pin ini merupakan pintu Paralel 8 bit dua arah (bi-directional). Bit ini mengirimkan byte alamat bila dilakukan pengaksesan memori luar.

8). Pin 29

Pin ini merupakan sinyal pengontrol yang memperbolehkan program memori luar masuk ke dalam bus selama proses pemberian / pengambilan perintah (Fetching).

9). Pin 30

Pin ini merupakan Pin yang digunakan untuk menahan alamat memori luar selama pelaksanaan perintah.

10). Pin 31

Pin ini bila diberi logika tinggi (high), pengontrol mikrokontroler akan melaksanakan perintah dari ROM atau EPROM ketika isi program counter kurang dari 4096. Bila diberi logic rendah (low) pengontrol mikro akan melaksanakan seluruh perintah dari memori program luar.

11). Pin 32 sampai 39

Pin ini merupakan pintu Paralel 8 bit dua arah (bi-directional). Bit ini mengirimkan byte alamat bila dilakukan pengaksesan memori luar.

12). Pin 40

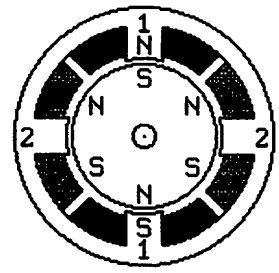
Pin ini merupakan Pin yang dihubungkan dengan tegangan catu atau VCC.

2.5. Komponen Pendukung

1. Motor Steper

Motor steper banyak digunakan untuk aplikasi yang biasanya cukup menggunakan torsi yang kecil, seperti piringan disket atau CD. Dalam hal kecepatan motor steper cukup cepat dibanding motor DC. Motor steper meupakan motor yang tidak memiliki komutator. Pada umumnya motor steper hanya mempunyai kumparan pada statornya dan pada bagian rotor merupakan magnet permanen. Dan dengan karakteristik yang dimiliki maka motor steper dapat diatur posisinya

berputar kearah yang diinginkan searah maupun berlawanan dengan arah jarum jam.



Gambar 2.3 Motor Steper

2. Resistor

Resistor atau sering disebut sebagai tahanan fungsinya untuk menahan atau memberikan hambatan terhadap arus listrik, dan resistor memberikan hambatan agar komponen yang diberi tegangan tidak dialiri dengan arus yang besar. Resistor mempunyai dua kaki, bodinya bergelang-gelang cat dan mempunyai warna yang berbeda. Warna pada gelang bodi inilah yang merupakan sebagai kode ukuran.

Kode warna tersebut antara lain adalah standar manufaktur yang dikeluarkan oleh EIA (*Electronic Industries Association*) seperti gambar dibawah ini :

Gambar 2.4 Gelang Resistor

Cara membaca ukuran melalui warna yang ada pada resistor adalah sebagai berikut:

Tabel 2.4 Kode Warna Resistor

Warna	Garis 1	Garis 2	Garis 3	Garis 4
Hitam	-	-	-	2%
Coklat	1	1	0	1%
Merah	2	2	00	2%
Oranye	3	3	000	3%
Kuning	4	4	0000	4%
Hijau	5	5	00000	5%
Biru	6	6	000000	6%
Ungu	7	7	0000000	7%
Abu-abu	8	8	1/100	8%
Putih	9	9	1/10	9%
Emas	-	-	1/10	5%
Perak	-	-	1/100	10%
Tak berwarna	-	-	-	20%

Keterangan : Pada gelang-gelang keempat adalah niali polaritas.

Contoh :

Jika resistan berwarna :	**Merah**	**Kuning**	**Coklat**	**Perak**
	2	**4**	**0**	**10%**

Jadi nilai resistan tersebut adalah : 240 ohm dengan polaritas 10%.

24×10^1 U x (100-10)% = 24×10^1 U x 90% = 21.6 U (Ohm).

3. Kapasitor

Kapasitor adalah komponen elektronik yang digunakan untuk menyimpan muatan listrik secara positif. Kapasitor terdiri dari dua bahan konduktor yang dipisahkan oleh bahan perekat (bahan elektrolit), kedua konduktor ini diberi muatan yang besarnya sama, tetapi tandanya berlawanan (satu bermuatan positif dan yang kedua bermuatan negatif). Kapasitor terdiri dari beberapa jenis, diantaranya adalah kapasitor elektrolit (elco), kapasitor keramik, kapasitor mylar, kapasitor mika, kapasitor polyester, dan kapasitor udara (Varco).

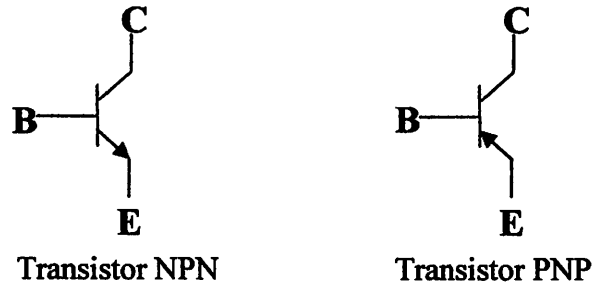
4. Kristal

Kristal adalah komponen yang difungsikan sebagai rangkaian osilator untuk mendapatkan frekuensi tertentu. Kristal mempunyai nilai yang bermacam-macam dan nilai itu menunjukkan dari frekuensi yang bisa dihasilkan oleh kristal tersebut.

Gambar 2.5 Simbol Kristal

5. Transistor

Transistor ada dua jenis yaitu PNP dan NPN, transistor mempunyai 3 bagian pokok yaitu emitor, basis, dan kolektor. Kolektor mempunyai jenis bahan yang sama negatif / positif. Sedangkan basis yang terletak diantara kolektor dan emitor mempunyai jenis yang berlawanan yaitu jika kolektor jenis negatif maka basis jenisnya positif dan sebaliknya.



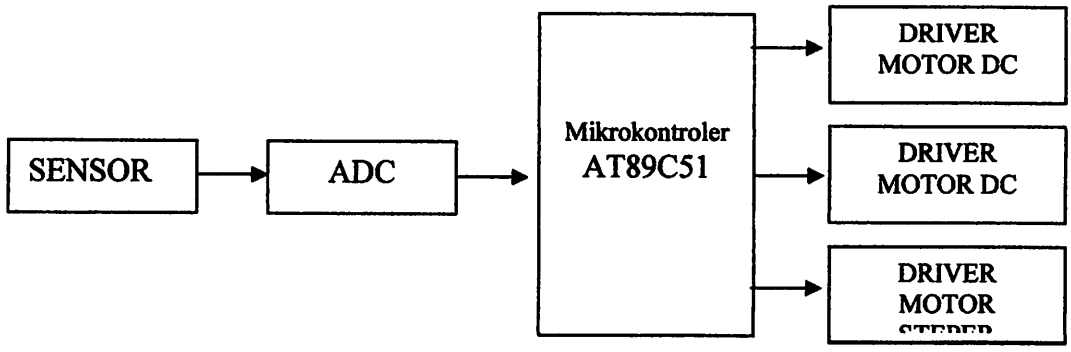
Gambar 2.6 Simbol Transistor

BAB III

PERANCANGAN LENGAN ROBOT PEMISAH BOLA BERWARNA BERBASIS MIKROKONTROLER AT89C51

3.1. Blok Diagram Perencanaan

Dalam perencanaan keseluruhan rangkaian terdiri dari beberapa blok yang saling berhubungan satu dengan yang lainnya. Untuk lebih jelasnya dapat dilihat pada Gambar 3.1.



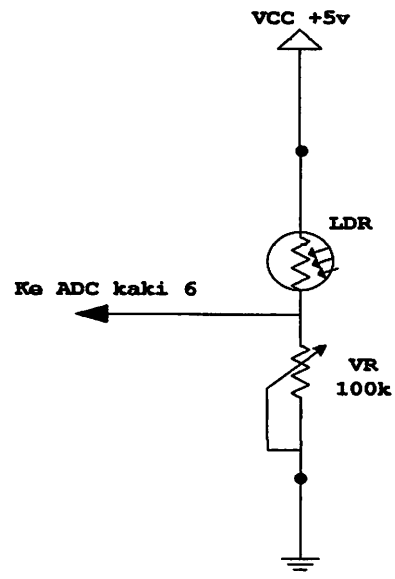
Gambar 3.1 Blok Diagram Perencanaan

Cara kerja secara umum blok diagram pada gambar 3.1 adalah dimulai ketika rangkaian sensor mendeteksi adanya perubahan intensitas cahaya yang disebabkan oleh pantulan permukaan benda yang telah diberi warna, perubahan intensitas cahaya yang diterima juga akan mempengaruhi resistansi sensor, hal ini akan menjadikan perubahan sinyal analog pada output rangkaian sensor. Perubahan atau selisih sinyal analog

pada output sensor akan diterima oleh rangkaian ADC yang kemudian akan diubah menjadi digital (bilangan biner), keluaran ADC yang berupa bilangan biner inilah yang akan dijadikan sebagai data masukan, sehingga mikrokontroler akan menggerakkan motor steper agar dapat berputar ke kanan dan ke kiri, dan motor DC agar dapat bergerak naik turun, kemudian menempatkan objek sesuai dengan tempatnya.

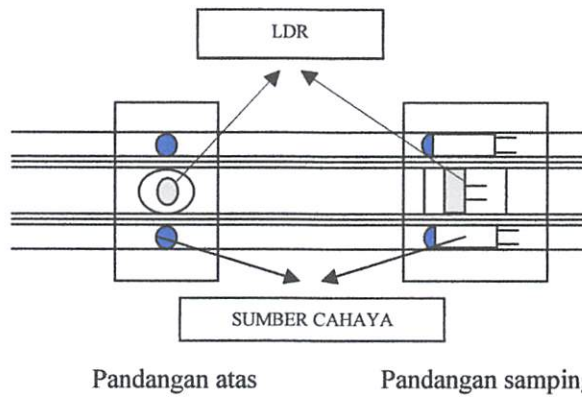
3.2. Rangkaian Sensor

Pada perancangan rangkaian sensor disini, komponen utamanya menggunakan LDR yang difungsikan untuk mengidentifikasi (sebagai mata), agar dapat mendeteksi perubahan intensitas cahaya yang dipantulkan, sehingga LDR dapat membedakan warna benda yang berada didepannya.



Gambar 3.2 Rangkaian Sensor

Agar dapat menerima cahaya yang dipantulkan oleh benda berwarna maka posisi LDR harus sejajar dengan sumber cahaya, namun LDR tidak boleh terkena cahaya secara langsung, jika terkena cahaya secara langsung maka intensitas cahaya hasil pantulan benda berwarna tidak akan mempengaruhi resistansi LDR, oleh karenanya LDR diberi pelindung agar tidak secara langsung menerima cahaya dari sumber cahaya.

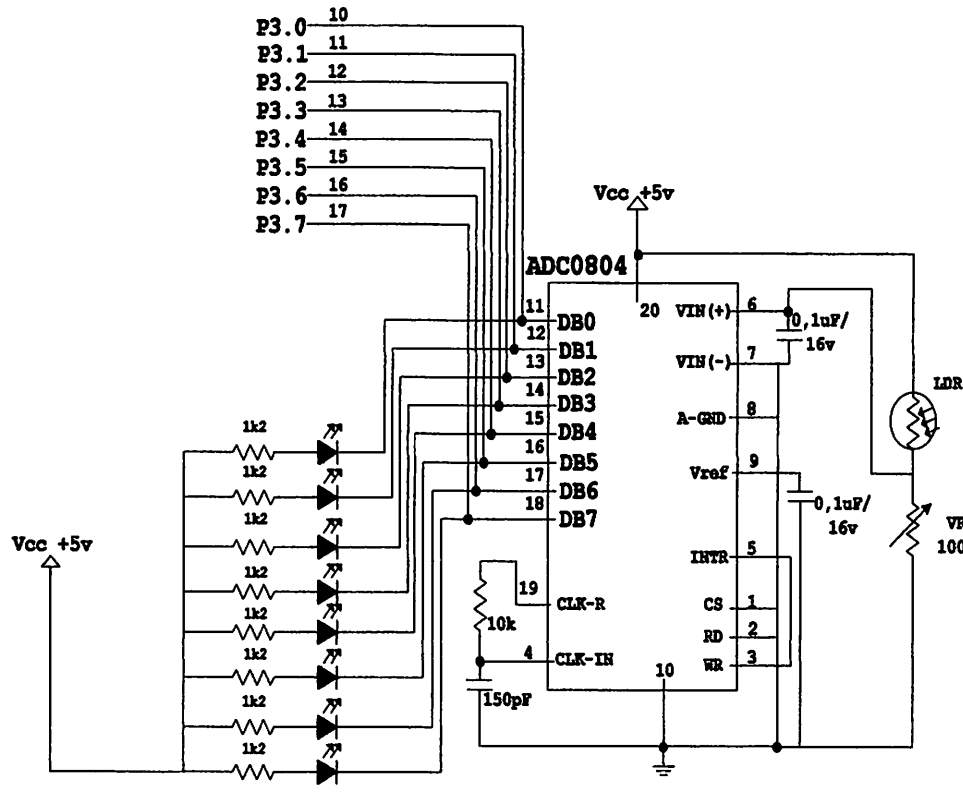


Gambar 3.3 Konstruksi Pemasangan LDR

Dari gambar 3.3 dapat diperoleh intensitas cahaya pantulan dari tiap-tiap benda berwarna yang ada di depan sensor. Intensitas cahaya yang dipantulkan benda berwarna yang ada di depan akan menyebabkan resistansi dari LDR berubah, dan perubahan resistansi ini juga akan mempengaruhi sinyal analog output rangkaian sensor, perubahan sinyal analog tersebut akan dibaca dan kemudian akan diubah menjadi data bilangan biner oleh ADC.

3.3. Rangkaian ADC (Analog Digital Converter)

Perancangan dan pembuatan rangkaian ADC dapat digambarkan seperti di bawah ini :



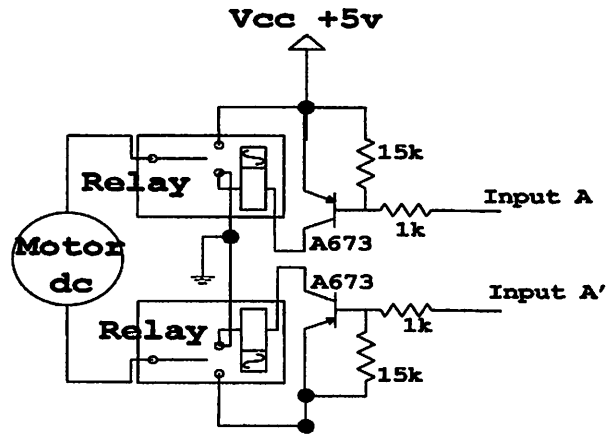
Gambar 3.4 Skema Rangkaian ADC

LDR yang digunakan sebagai sensor cahaya akan mengalami perubahan resistansi ketika benda berada di depan sensor, perubahan resistansi tersebut juga akan menyebabkan kondisi sinyal analog juga berubah, perubahan sinyal analog akan dibaca dan diubah menjadi data digital 8 bit dengan kondisi keluaran ADC hanya berupa kondisi HIGH (1) dan LOW (0), data keluaran output ADC terdapat pada pin DB0 – DB7,

ketika ada benda di depan sensor yang menyebabkan perubahan tegangan pada keluaran rangkaian sensor yang secara bertahap akan diubah ke dalam data biner pada indikator dengan kondisi 0/1 pada DB0 – DB7 data digital tersebut akan dijadikan data input pada mikrokontroler AT89C51 yang selanjutnya digunakan sebagai data pembanding pada bahasa pemrograman.

3.4. Rangkaian Driver Motor DC

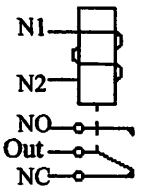
Rangkaian Driver motor DC yang berfungsi untuk mengubah arah putaran motor DC. Dalam pembuatan rangkaian driver ini digunakan dua buah transistor yang dirangkai sebagai saklar. Prinsip kerja dari rangkaian ini adalah jika salah satu basis transistor mendapat bias negatif maka transistor akan aktif, dan ini menyebabkan arus dapat mengalir dari kolektor menuju emitor, akibatnya adalah tegangan kolektor akan sama atau mendekati tegangan emitor, dengan demikian salah satu relay akan bekerja dan motor akan bergerak.



Gambar 3.5. Rangkaian Driver Motor DC

Transistor yang terpasang difungsikan sebagai saklar otomatis, disebut otomatis karena sebagai pengendali dari saklar tersebut digunakan tegangan triger yang terpasang pada kaki basis yang mana tegangan triger tersebut berasal dari keluaran mikrokontroler. Untuk seri transistor digunakan transistor A673.

Selain transistor juga digunakan komponen relay DC, kaki NO artinya bahwa dalam keadaan biasa tanpa adanya tegangan selalu dalam keadaan terbuka (Normal Open), tetapi ketika dalam keadaan ada tegangan maka akan berubah menjadi tertutup (Normal Close). Sedangkan kaki NC artinya bahwa dalam keadaan tidak ada tegangan akan selalu dalam keadaan tertutup (Normal Close), namun ketika ada tegangan akan berubah menjadi terbuka (Normal Open). Untuk kaki N1 dan N2 adalah kaki kumparan yang berfungsi sebagai pengaktif relay.



Gambar 3.6 Susunan kaki Relay

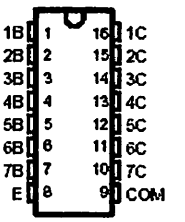
Dalam penyambungan, untuk semua kaki NO disambung menjadi satu dengan tegangan sumber (VCC), sedangkan semua kaki NC disambung menjadi satu dengan ground.

Penggunaan input ada dua macam yaitu input pertama (A) digunakan untuk mengaktifkan motor putar kanan, sedangkan input kedua (A') digunakan untuk motor putar kiri.

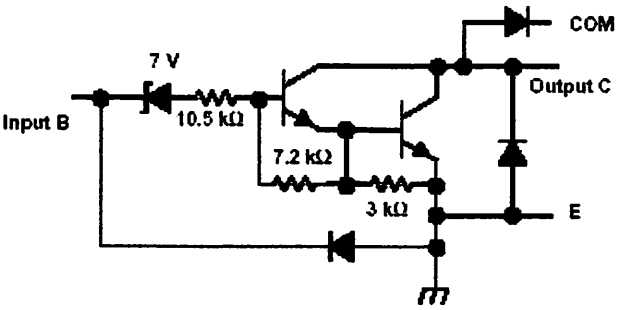
3.5. Driver Motor Steper

Output dari mikrokontroler hanya berupa kondisi high (1) dengan tegangan sebesar 5V DC dan kondisi low (0) dengan tegangan 0V DC, karena keterbatasan ini maka agar motor steper dengan tegangan kerja sebesar 12V DC dapat berputar dan menggerakkan mekanik diperlukan suatu driver, driver motor steper berfungsi mengubah data digital yang dikeluarkan mikrokontroler menjadi tegangan yang lebih besar agar motor steper dapat berputar, driver motor steper yang biasanya menggunakan rangkaian transistor konfigurasi pasangan Darlington untuk mendapatkan penguatan arus/beta (β) lebih besar, jika beta makin tinggi maka akan

didapat penguatan pada (I) arus juga naik. Hubungan kaki kolektor Tr1 dan Tr2 dihubungkan dan kaki emitor Tr1 menjadi input pada basis Tr2, hal ini menjadikan β naik karena $\beta_{total} = \beta_1 \times \beta_2$, dari uraian singkat maka dalam pembuatan driver motor steper menggunakan IC ULN2003 yang mempunyai 7 pasang rangkaian Darlington dalam 1 paket IC.

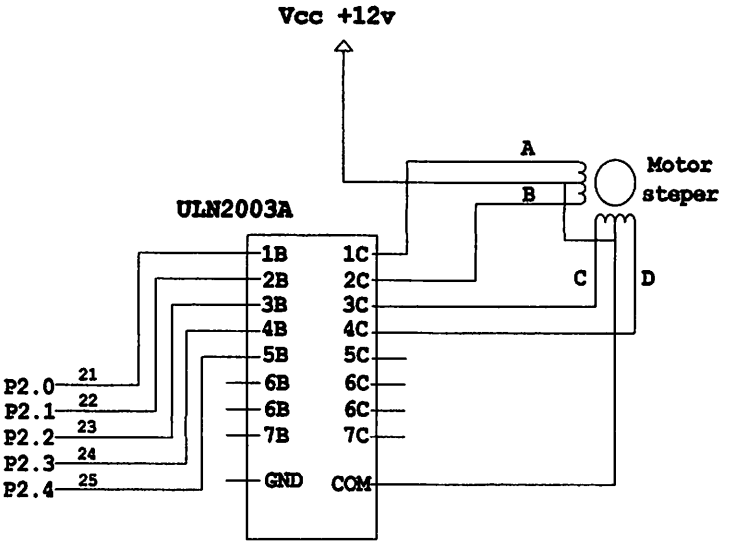


Gambar 3.7 IC ULN2003



Gambar 3.8 Rangkaian Darlington Pada IC ULN2003

Pada rangkaian driver motor steper di bawah ini terlihat keluaran dari mikrokontroler melalui port2.0 – port2.3 dijadikan input IC ULN2003 dengan tujuan agar keluaran IC mendapatkan tegangan yang lebih besar juga mendapatkan penguatan arus.



Gambar 3.9 Skema Rangkaian Motor Steper

3.6. Mikrokontroler

Pada bagian ini berfungsi sebagai pengolah data yang dihasilkan oleh sensor dimana sensor membaca keadaan bendungan dan menghasilkan data empat bit yang berbeda – beda. Rangkaian pengolah data ini tidak lain adalah sistem minimum AT89C51. Sistem minimum AT89C51 merupakan pusat dan sebagai pengatur dari kerja pada keseluruhan sistem. Rangkaian ini dapat dibuat dengan tepat kecil dan komponen yang sedikit dengan memanfaatkan fasilitas yang telah dimiliki oleh pengontrol AT89C51.

Pada dasarnya Mikrokontroler Atmel keluarga MCS51 mempunyai dua kelompok instruksi untuk mengeluarkan data ke port paralel :

1). Kelompok instruksi pertama bekerja pada port seutuhnya, artinya 8 jalur dari port bersangkutan, misalnya MOV P3,#0FFH membuat ke-delapan jalur 3 semuanya dalam logika '1' (atau isinya 1111 1111 dalam biner).

2). Kelompok instruksi kedua hanya berpengaruh pada salah satu jalur atau bit dari port, misalnya SETB P3.4 artinya men-set bit-4 dari port 3, atau instruksi CLR P3.3 digunakan untuk me-nolkan bit-3 dari Port 3.

Selain itu port paralel bisa pula dipakai untuk menerima masukan sinyal digital dari luar mikrokontroler :

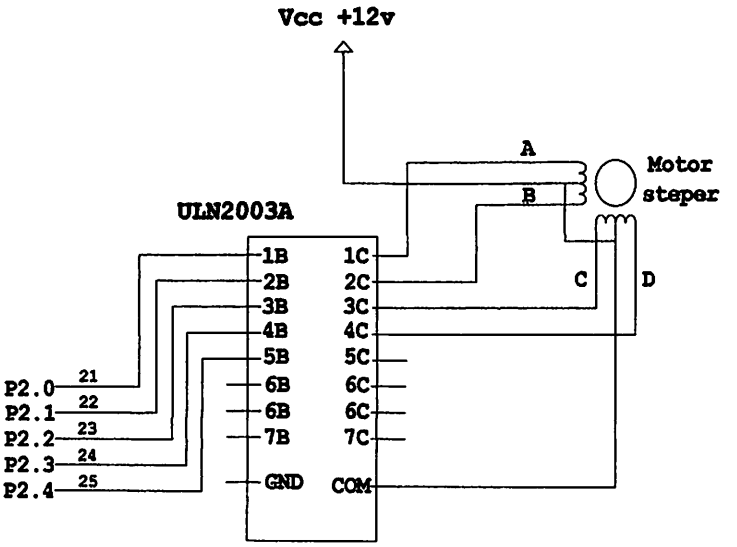
1). Instruksi MOV A,P3 digunakan untuk membaca data (digital) pada seluruh bit port 1 kemudian menyimpannya ke dalam akumulator.

2). Pembacaan data bisa juga dilakukan hanya pada satu bit port saja, misalnya instruksi JNB P3.7, $ digunakan untuk memantau bit P3.7, jika P3.7 = 0 mikrokontroler akan kembali melaksanakan instruksi tersebut, mikrokontroler akan meneruskan kembali instruksi berikutnya jika P3.7 = 1.

Tabel 3.1 Data Keluaran ADC

Warna Bola	Data 8 Bit Keluaran ADC0804
Kuning	0001 0001
Putih	0000 1110
Hitam	0001 1010
Kosong	0000 0000

Setelah data 8 bit hasil keluaran dari ADC 0804 diperoleh dan tidak mudah berubah-ubah. Dan agar lebih memperoleh pembuatan program lebih daluluu membuat flowchart program. seperti pada gambar 3.15 di bawah ini :



Gambar 3.9 Skema Rangkaian Motor Steper

3.6. Mikrokontroler

Pada bagian ini berfungsi sebagai pengolah data yang dihasilkan oleh sensor dimana sensor membaca keadaan bendungan dan menghasilkan data empat bit yang berbeda – beda. Rangkaian pengolah data ini tidak lain adalah sistem minimum AT89C51. Sistem minimum AT89C51 merupakan pusat dan sebagai pengatur dari kerja pada keseluruhan sistem. Rangkaian ini dapat dibuat dengan tepat kecil dan komponen yang sedikit dengan memanfaatkan fasilitas yang telah dimiliki oleh pengontrol AT89C51.

Pada dasarnya Mikrokontroler Atmel keluarga MCS51 mempunyai dua kelompok instruksi untuk mengeluarkan data ke port paralel :

1). Kelompok instruksi pertama bekerja pada port seutuhnya, artinya 8 jalur dari port bersangkutan, misalnya MOV P3,#0FFH membuat ke-delapan jalur 3 semuanya dalam logika '1' (atau isinya 1111 1111 dalam biner).

2). Kelompok instruksi kedua hanya berpengaruh pada salah satu jalur atau bit dari port, misalnya SETB P3.4 artinya men-set bit-4 dari port 3, atau instruksi CLR P3.3 digunakan untuk me-nolkan bit-3 dari Port 3.

Selain itu port paralel bisa pula dipakai untuk menerima masukan sinyal digital dari luar mikrokontroler :

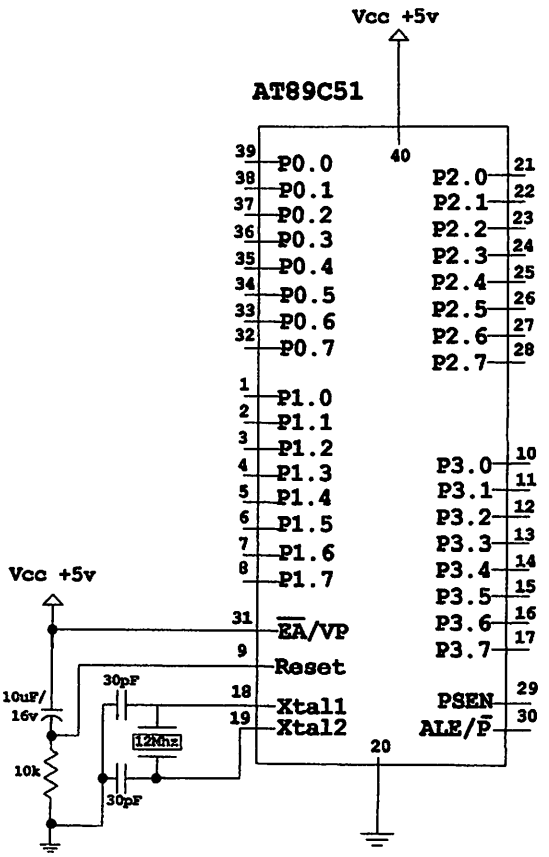
1). Instruksi MOV A,P3 digunakan untuk membaca data (digital) pada seluruh bit port 1 kemudian menyimpannya ke dalam akumulator.

2). Pembacaan data bisa juga dilakukan hanya pada satu bit port saja, misalnya instruksi JNB P3.7, $ digunakan untuk memantau bit P3.7, jika P3.7 = 0 mikrokontroler akan kembali melaksanakan instruksi tersebut, mikrokontroler akan meneruskan kembali instruksi berikutnya jika P3.7 = 1.

Tabel 3.1 Data keluaran ADC

Warna Bola	Data 8 Bit Keluaran ADC0804
Kuning	0001 0001
Putih	0000 1110
Hitam	0001 1010
Kosong	0000 0001

Setelah data 8 bit hasil keluaran dari ADC 0804 diperoleh dan tidak mudah berubah-ubah. Dan agar lebih memperermudah pembuatan program lebih dahulu membuat flowchart program, seperti pada gambar 3.15 di bawah ini :

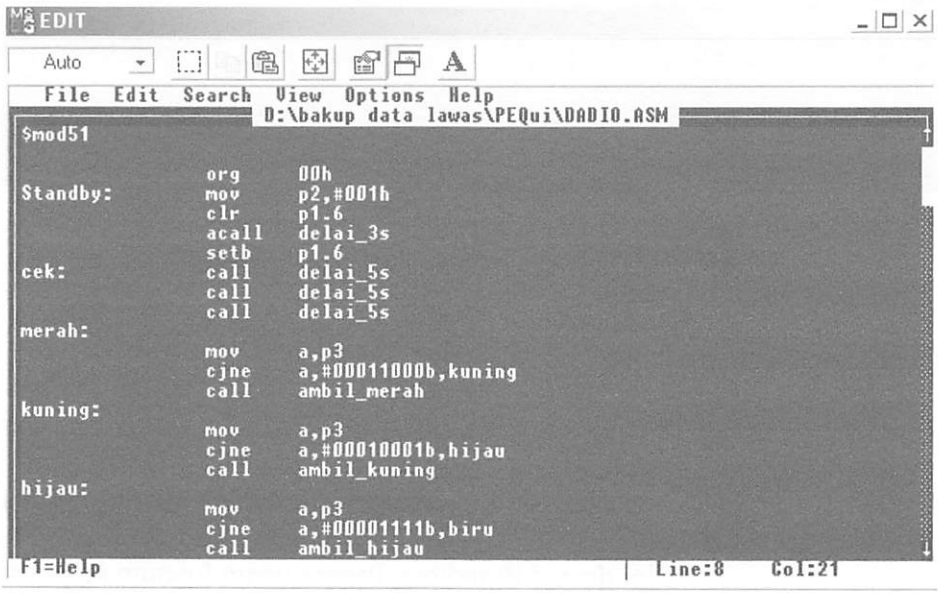


Gambar 3.10 Rangkaian Sistem Minimum AT89C51

Dalam membuat sistem minimum AT89C51 hanya dibutuhkan sebuah kristal 12 Mhz dan kapasitor 30 pf sebagai sistem clocknya, dan sebuah resistor 10k dan kapasitor 10 Mikro dipasang seri sebagai rangkaian reset, dan yang terakhir adalah IC mikrokontroler AT89C51 itu sendiri.

3.7. Pembuatan dan Pengisian Program

Cara kerja robot lengan pemisah benda berwarna ini tidak sepenuhnya ditangani oleh perangkat lunak, karena mikrokontroler yang telah diberikan program didalamnya juga memerlukan inputan dari luar untuk dapat bekerja. Pembuatan program dilakukan dengan menggunakan program EDIT, file yang sudah dibuat lalu disimpan dengan ekstensi ASM. Tampilan menu software EDIT adalah seperti di bawah ini :

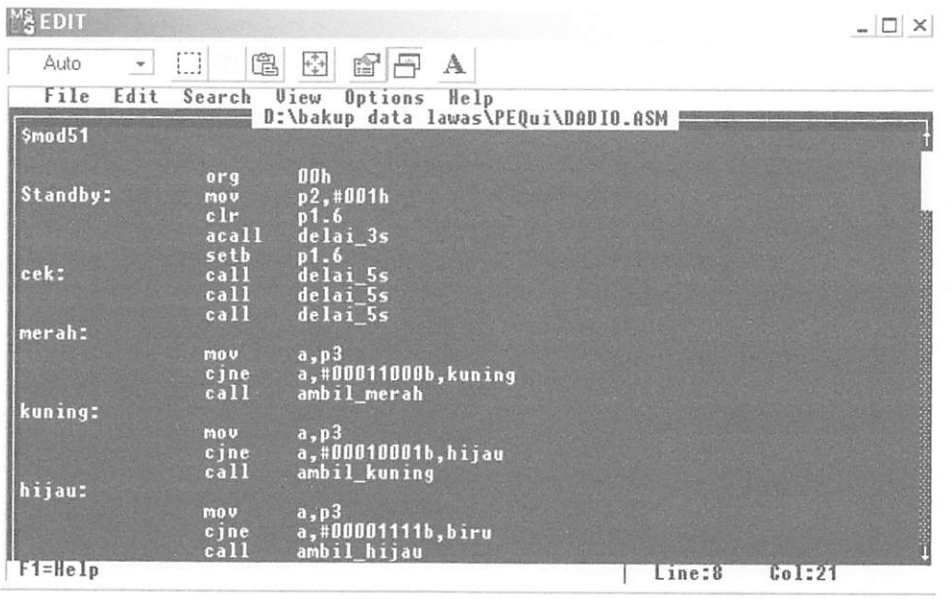


Gambar 3.11 Tampilan Program Edit

Untuk mengubah program yang telah dibuat dan disimpan dengan ekstensi asm menjadi kode heksadesimal digunakan program cross assembler dari Metalink Corporation, yaitu ASM51.EXE. Dari hasil kompilasi didapatkan dua file yaitu file listing dan heksa. File listing ditandai dengan extensi LST dan heksa ditandai dengan ekstensi HEX, sedangkan program yang dimasukkan ke dalam mikrokontroler AT89C51

3.7. Pembuatan dan Pengisian Program

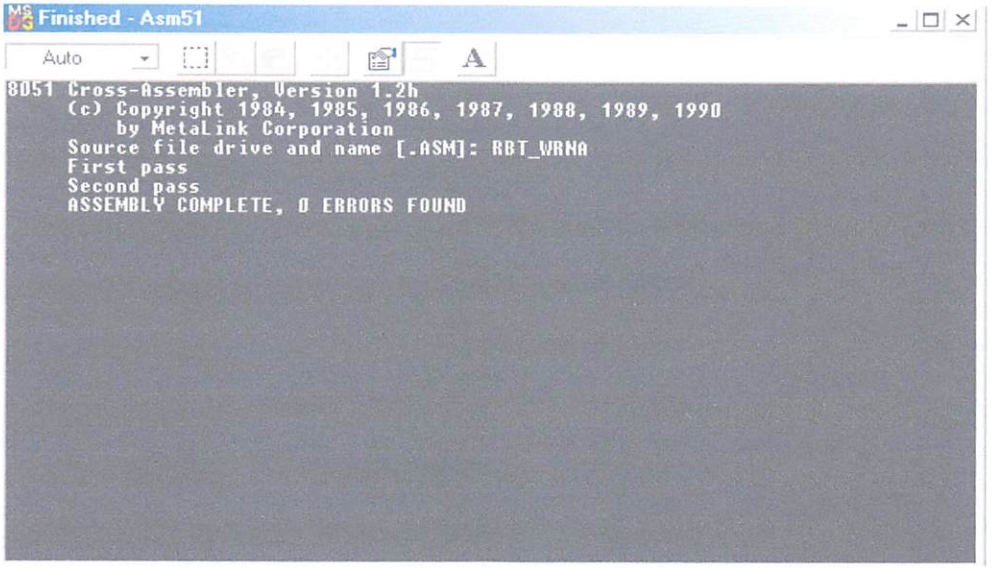
Cara kerja robot lengan pemisah benda berwarna ini tidak sepenuhnya ditangani oleh perangkat lunak, karena mikrokontroler yang telah diberikan program didalamnya juga memerlukan inputan dari luar untuk dapat bekerja. Pembuatan program dilakukan dengan menggunakan program EDIT, file yang sudah dibuat lalu disimpan dengan ekstensi ASM. Tampilan menu software EDIT adalah seperti di bawah ini :



Gambar 3.11 Tampilan Program Edit

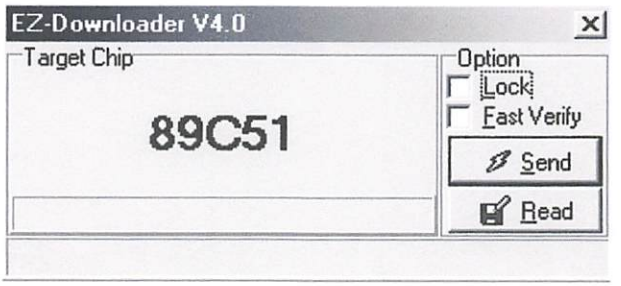
Untuk mengubah program yang telah dibuat dan disimpan dengan ekstensi asm menjadi kode heksadesimal digunakan program cross assembler dari Metalink Corporation, yaitu ASM51.EXE. Dari hasil kompilasi didapatkan dua file yaitu file listing dan heksa. File listing ditandai dengan extensi LST dan heksa ditandai dengan ekstensi HEX, sedangkan program yang dimasukkan ke dalam mikrokontroler AT89C51

adalah file dengan ekstensi HEX. Tampilan software ASM51.EXE adalah seperti di bawah ini :



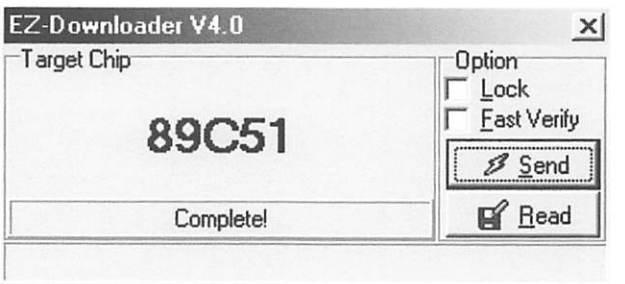
Gambar 3.12 Tampilan Program ASM51.EXE

Software yang digunakan untuk memasukkan program ke dalam mikrokontroler adalah Ezdl4 (Eazy Downloder 4) Cara menggunakannya adalah dengan membuka aplikasi program Ezdl4 (Eazy Downloder 4) sehingga muncul menu seperti gambar di bawah ini :



Gambar 3.13 Tampilan Program Ezdl4

Langkah selanjutnya adalah membuka file heksa dengan menekan Send, lalu memasukkan nama program (file.hex) yang telah dibuat, menunggu beberapa saat untuk verifikasi, setelah pengisian program ke dalam mikrokontroler telah komplit, kemudian melepaskan IC mikrokontroler dan memasang IC pada rangkaian yang telah dibuat. Dan di bawah ini adalah tampilan Ezdl4 (Eazy Downloder 4) saat pengisian program telah selesai (Completed).



Gambar 3.14 Tampilan Program Ezdl4 Jika Pengisian Program Telah Selesai

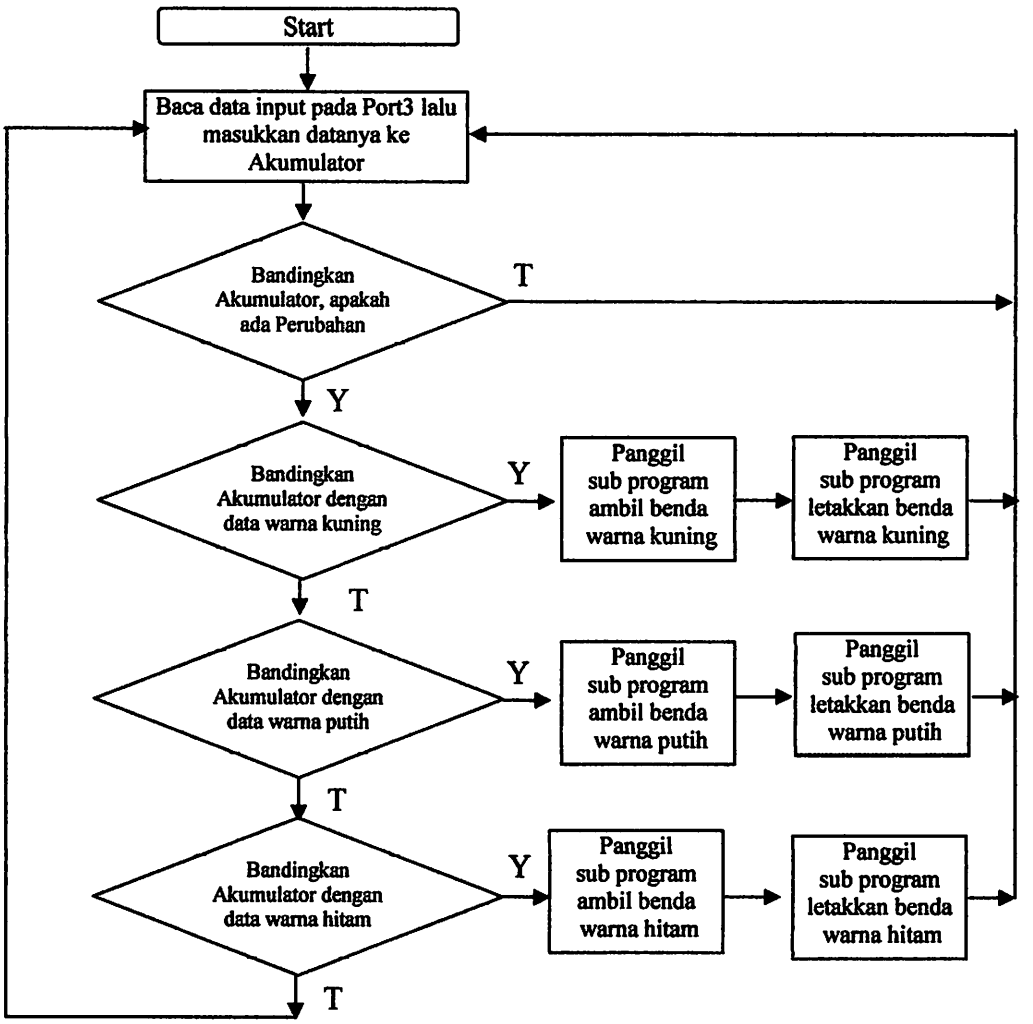
3.8. Program pemisah warna

Sebelum pembuatan program agar mikrokontroler AT89C51 bisa membaca dan membedakan warna juga dapat menggerakkan mekanik untuk mengambil dan menempatkan benda yang telah diberi warna sesuai pada tempatnya masing-masing, maka terlebih dahulu harus melakukan pendataan terhadap data-data 8 bit yang dihasilkan oleh ADC 0804, di bawah ini merupakan tabel hasil pengamatan terhadap data 8 bit yang merupakan hasil keluaran ADC :

Tabel 3.1 Data keluaran ADC

Warna Bola	Data 8 Bit Keluaran ADC0804
Kuning	0001 0001
Putih	0000 1110
Hitam	0001 1010
Kosong	0000 0001

Setelah data 8 bit hasil keluaran dari ADC 0804 diperoleh dan tidak mudah berubah-ubah. Dan agar lebih mempermudah pembuatan program lebih dahulu membuat flowchart program, seperti pada gambar 3.15 di bawah ini :



Gambar 3. 15 Flowchart Program

Program dimulai dari start, kemudian membaca data yang ada di port 3 kemudian data tersebut dimasukkan ke akumulator yang nantinya akan digunakan sebagai data pembanding, kemudian jika ada perubahan pada data yang masuk ke akumulator akan dibandingkan dengan data yang ada pada tabel 3.1, dan jika ternyata data pada akumulator ada yang sama dengan salah satu data 8 bit pada tabel 3.1 maka mikrokontroler akan menggerakkan motor steper dan motor DC untuk mengambil dan meletakkan benda sesuai dengan data 8 bit yang diterima. Dan bila tidak ada data 8 bit yang sama dengan tabel 3.1 maka mikrokontroler akan standby.

Untuk mengetahui lebih jelas tentang jalannya program, di bawah ini akan diterangkan beberapa bagian program dari program prototype robot lengan pemisah warna berbasis mikrokontroler AT89C51.

1). Program cek warna :

```
;-----------------------------------------------------------------------------------------
;Program cek warna
;Untuk cek apakah ada perubahan warna tidak dan apakah ada data yang
;sama antara data pembanding  dengan data yang masuk
;-----------------------------------------------------------------------------------------
standby:   call      delai              ; panggil subprogram dengan alamat delai

cek:                                    ; cek ada tidaknya benda
      call   delai5                     ; panggil sub program dengan alamat delai5
      cjne   a,#00011000b,kuning        ; bandingkan data pada A dengan 00011000b,
                                          jika sama panggil sub program ambil_kuning,
                                          jika tidak  lompat  ke alamat kuning
      sjmp   standby

kuning:
      mov    a,p3
      cjne   a,#00010001b,putih
      call   ambil_kuning
```

```
        sjmp  standby

putih:
        mov   a,p3
        cjne  a,#00001110b,hitam
        call  ambil_putih
        sjmp  standby
hitam:
        mov   a,p3
        cjne  a,#00011010b,cek
        call  ambil_hitam
        sjmp  standby
```

Program akan membandingkan data pada port 3 yang kemudian data pada port 3 tersebut dipindahkan ke akumulator terlebih dahulu, apakah data pada akumulator terjadi perubahan atau tidak, jika tidak program akan kembali menuju alamat stanby, jika terjadi perubahan maka program akan menuju ke alamat merah, demikian seterusnya hingga mikrokontroler membaca terjadinya perubahan pada akumulator, kemudian pada alamat merah perubahan data yang terjadi pada akumulator akan kembali dibandingkan dengan data warna merah pada tabel 3.1 jika data pada akumulator sama dengan data warna merah pada tabel 3.1 mikrokontroler akan memanggil program dengan alamat ambil_merah, jika tidak terjadi kesamaan program akan loncat ke alamat kuning untuk kembali membandingkan data pada akumulator dengan data warna kuning pada tabel 3.1, demikian seterusnya hingga data pada akumulator ada yang sama dengan data warna pada tabel 3.1.

Jika terjadi kesamaan data antara data yang ada pada akumulator dengan data pada tabel 3.1 mikrokontroler akan melanjutkan ke program

untuk mengambil, berputar dan meletakkan benda. Untuk lebih jelasnya di bawah ini adalah potongan program untuk mengambil, berputar dan meletakkan benda.

SUB PROGRAM PROSES PENGAMBILAN, BERPUTAR &

```
ambil_kuning:
        call    ambil
        call    putar_kanan
        call    putar_kanan
        call    lepas
        call    putar_kiri
        call    putar_kiri
        ret
ambil_putih:
        call    ambil
        call    putar_kiri
        call    putar_kiri
        call    lepas
        call    putar_kanan
        call    putar_kanan
        ret
ambil_hitam:
        call    ambil
        call    putar_kiri
        call    putar_kiri
        call    putar_kiri
        call    lepas
        call    putar_kanan
        call    putar_kanan
        call    putar_kanan
        ret
```

SUB PROGRAM MENGGERAKKAN LENGAN &

```
; lengan turun untuk mengambil benda
ambil:
        clr     p1.0            ; keluarkan logika 0 pada port 1.0 ( lengan bergerak turun)
        call    Delai           ; panggil sub program dengan alamat delai
        setb    p1.0            ; keluarkan logika 1 pada port 1.0  (lengan berhenti
bergerak)
        clr     p1.2            ; keluarkan logika 0 pada p0rt 1.2 ( jari bergerak
                                  mencengkaram)
        call    Delai3
        setb    p1.2            ; keluarkan logika 1 pada port 1.2 (( jari berhenti
                                  mencengkaram)

        call    delai3
        clr     p1.1            ; keluarkan logika 0 pada port 1.1 ( lengan bergerak naik)
```

```
        call    Delai
        call    Delai3
        setb    p1.1            ; keluarkan logika 1 pada port 1,1  (lengan berhenti
bergerak)
        call    Delai3
        ret
; lengan turun untuk meletakkan benda
lepas:
        clr     p1.0            ; keluarkan logika 0 pada port 1.0 ( lengan bergerak turun)
        call    Delai           ; panggil sub program dengan alamat delai
        setb    p1.0            ; keluarkan logika 1 pada port 1.0  (lengan berhenti
bergerak)
        clr     p1.2            ; keluarkan logika 0 pada port 1.2 ( jari bergerak
                                  membuka cengkaraman)
        call    Delai3
        setb    p1.3            ; keluarkan logika 1 pada port 1.3 (( jari berhenti
                                  membuka cengkaraman)

        call    delai3
        clr     p1.1            ; keluarkan logika 0 pada port 1.1 ( lengan bergerak naik)

        call    Delai
        call    Delai3
        setb    p1.1            ; keluarkan logika 1 pada port 1,1  (lengan berhenti
bergerak)
        call    Delai3
        ret
```

SUB PROGRAM PUTARAN MOTOR STEPER

```
; motor steper berputar kekanan
putar_kanan:    mov     r5,#20          ; isi r5 dengan 20
kanan:
        mov     p2,#001h        ; keluarkan data 01h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        mov     p2,#002         ; keluarkan data 02h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        mov     p2,#004h        ; keluarkan data 04h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        mov     p2,#008h        ; keluarkan data 08h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        djnz    r5,kanan        ; kurangi isi r5 dengan 1 jika belum 0 loncat ke alamat
                                  kanan, jika telah sama dengan 0 lanjutkan program
        Ret
; motor steper berputar kekiri
putar_kiri:     mov     r5,#20          ; isi r5 dengan 20
kiri:
        mov     p2,#008h        ; keluarkan data 08h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        mov     p2,#004h        ; keluarkan data 04h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        mov     p2,#002h        ; keluarkan data 02h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        mov     p2,#001h        ; keluarkan data 01h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        djnz    r5,kiri         ; kurangi isi r5 dengan 1 jika belum 0 loncat ke alamat kiri
```

```
                                        Jika telah sama dengan 0 lanjutkan program
        ret
```

SUB PROGRAM DELAI (tunda waktu)

```
Delai:    Mov   r1,#10        ; isi r1 (register1)dengan 10
Loop_1:   Mov   r2,#150       ; isi r2 (register1)dengan 150
Loop_2:   Mov   r3,#250       ; isi r3 (register1)dengan 250
Loop_3:   Djnz  r3,loop_3     ; kurangi isi r3 dengan 1 jika belum 0 loncat ke alamat
loop_3
                              Jika telah sama dengan 0 lanjutkan program
          Djnz  r2,loop_2
          Djnz  r1,loop_1
          ret
Delay1:   Mov   r0,#70
Delay2:   Mov   r2,#70
          Djnz  r2,$
          Djnz  r0,Delay2
          Ret
Delai3:   Mov   r1,#30
Loop_a:   Mov   r2,#150
Loop_b:   Mov   r3,#150
Loop_c:   Djnz  r3,loop_c
          Djnz  r2,loop_b
          Djnz  r1,loop_a
          ret
Delai5:   Mov   r1,#10
Loop_d:   Mov   r2,#250
Loop_e:   Mov   r3,#250
Loop_f:   Djnz  r3,loop_f
          Djnz  r2,loop_e
          Djnz  r1,loop_d
          Ret
```

Dari potongan program di atas pada alamat ambil, dapat diketahui bahwa untuk menggerakkan lengan agar dapat naik dan turun menggunakan instruksi clr p1.0 yaitu mengeluarkan logika low (0) pada port 1.0 sebagai input A driver motor DC pertama untuk menggerakkan lengan turun dan sedangkan port 1.1 instruksi clr p1.1 yaitu mengeluarkan logika low (0) pada port 1.1 sebagai input A' pada driver motor DC pertama untuk menggerakkan lengan naik, penggunaan instruksi clr p1.0 dan p1.1 dikarenakan jenis transistor yang digunakan pada rangkaian driver motor DC untuk dapat mengaktifkan transistor dibutuhkan logika

low (0). Instruksi clr p1.2 dan clr p1.3 juga digunakan sebagai input pada driver motor DC kedua agar dapat mencengkeram serta membuka cengkeraman untuk meletakkan benda. Sedangkan instruksi **setb** atau mengubah inputan driver motor DC yang tadinya low menjadi high. Hal ini digunakan untuk menghentikan motor jika dirasa posisi lengan atau pencengkeram telah sesuai pada posisi yang diinginkan.

Pada sub program untuk memutar motor steper menggunakan port2.0 sampai port2.3 sebagai inputan pada IC ULN2003, arah putaran motor steper tergantung dari data yang dikeluarkan mikrokontroler melalui port2.0 sampai port2.3, agar lebih jelasnya dapat dilihat pada tabel di bawah ini :

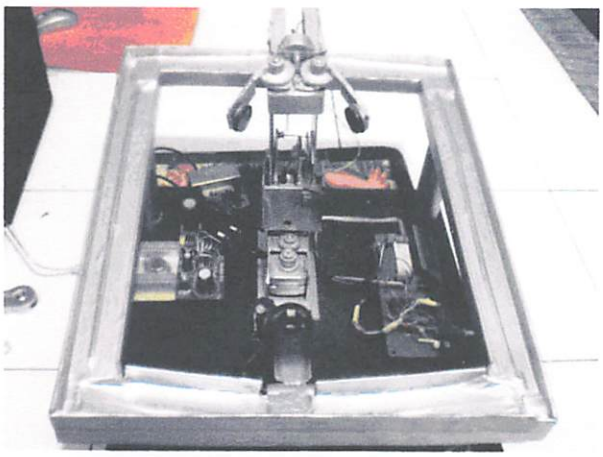
Tabel 3.2 Data Keluaran Pada Port2.0 Sampai Port2.3

Arah putaran motor steper	Data Keluaran pada port2.0 sampai port2.3			
	(P2.0)	(P2.1)	(P2.2)	(P2.3)
Kanan Data 1	1	0	0	0
Kanan Data 2	0	1	0	0
Kanan Data 3	0	0	1	0
Kanan Data 4	0	0	0	1
Kiri Data 1	0	0	0	1
Kiri Data 2	0	0	1	0
Kiri Data 3	0	1	0	0
Kiri Data 4	1	0	0	0

3.9. Gambar Hasil Perakitan Alat.

Setelah merancang dan merakit lengan robot pemisah bola tersebut maka lengan robot tsrsebut siap untuk diuji dan siap untuk melakukan kerja yang telah di tentukan dalam program, gambar alat yang telah kami rancang dan kami rakit adalah di bawah ini:

Gambar 3.16. Gambar Alat Tampak Atas.



Gambar 3.17. Gambar Alat Tampak Depan.

BAB IV

PENGUJIAN ALAT

Pengujian keseluruhan sistem dilakukan dengan menggabungkan tiap-tiap blok rangkaian yang ada. Pengujian ini dilakukan untuk mengetahui kerja seluruh system yang telah di rangkai menjadi satu. Adapun tujuan dari proses pengujian ini adalah:

1. Untuk mengetahui apakah alat yang telah dibuat sudah sesuai dengan yang direncanakan atau masih ada kesalahan yang menyebabkan alat tidak bekerja sesuai dengan yang diharapkan.
2. Untuk mengetahui hasil kerja alat yang merupakan realisasi rancangan rangkaian tersebut, sejauah mana tingkat error dari alat yang telah dibuat.

Pengujian alat meliputi pengujian perangkat keras dan pengujian perangkat lunak. Pengujian dilakukan pada beberapa bagian untuk mempermudah dalam menganalisa hasil perancangan dan pengujian yang dilakukan.

a. pengujian perangkat keras.

   1. Pengujian sumber tegangan pada system.
   2. Pengujian sensor warna.
   3. Pengujian ADC.
   4. Pengujian rangkaian mekanik lengan robot yang telah dirangkai.

b. pengujian perangkat lunak.

Pengujian perangkat lunak disini adalah pengujian terhadap set program yang telah di buat.

4.1. Pengujian Sumber Tegangan

Hasil penelitian dari nilai-nilai sumber tegangan pada perancangan maupun dari hasil pengukuran pada alat adalah :

Sumber tegangan DC	Perancangan Vcc	Pengukuran ke				
		1 (V)	2 (V)	3 (V)	4 (V)	Σ (V)
5 Volt	5 Volt	4.6	4.9	5.0	4.7	4.8
12 Volt	12 Volt	11.8	12	11.6	11.6	11.75

Tabel 4.1 Pengujian Power Supply :

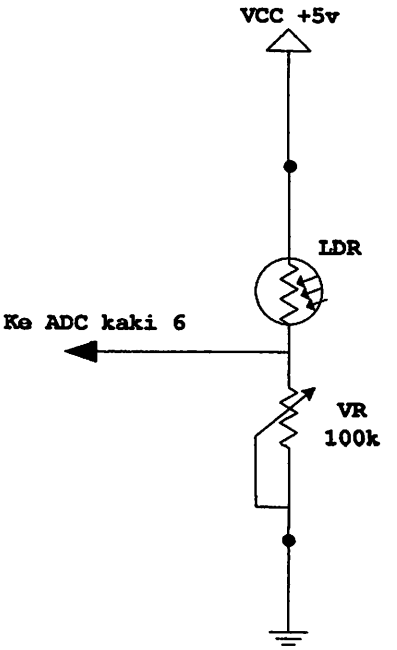
Saat pengukuran tegangan 5V pada perancangan dilakukan beberapa kali dengan menggunakan multitester digital muncul beberapa kejadian. Pada pengukuran kesatu nilai VCC adalah 4,8V hal ini dikarenakan banyaknya beban pada rangkaian, sehingga sumber tegangan

berat memikulnya, sehingga tegangan VCC turun 5V menjadi 4,8V. Pada pengukuran kedua tegangan terbaca 4.9V, pada pengukuran ketiga 5V dan terakhir muncul angka 4,7V sehingga dapat ditarik kesimpulan rata-rata tegangan stabilnya adalah 4,8V.

Saat pengukuran sumber tegangan 12Vdilakukan pengukuran beberapa kali dan muncul kenyataan pengukuran satu sampai dengan empat adalah 11.8V, 12V, 11.6V dan 11.6V sehingga dapat ditarik kesimpulan tegangan stabilnya adalah 11.75V

4.2. Pengujian Rangkaian Sensor Warna.

Pada pengujian rangkaian sensor warna ini, untuk rangkaian yang akan diuji adalah rangkaian yang akan membaca warna-warna yang dikehendaki, dan rangkaian yang dirancang adalah menggunakan komponen LDR, komponen ini adalah pomponen yang peka terhadap cahaya, dan pada sensor waran ini kita menitik beratkan pada intensitas cahaya yang dipantulkan pada tiap-tiap warna yang dihasilkan ,kemudian intensitas tiap-tiap warna tersebut akan di terima oleh LDR,kemudian intensitas cahaya yang di terima olaeh LDR akan dikonversi dalam bentuk tegangan, dan tegangan yang di keluarkan oleh LDR kamudian akan diolah olah ADC.



gambar 4.1. Sensor warna.

Setelah diadakan pengukuran beberapa kali maka diperoleh tabel seperti berikut:

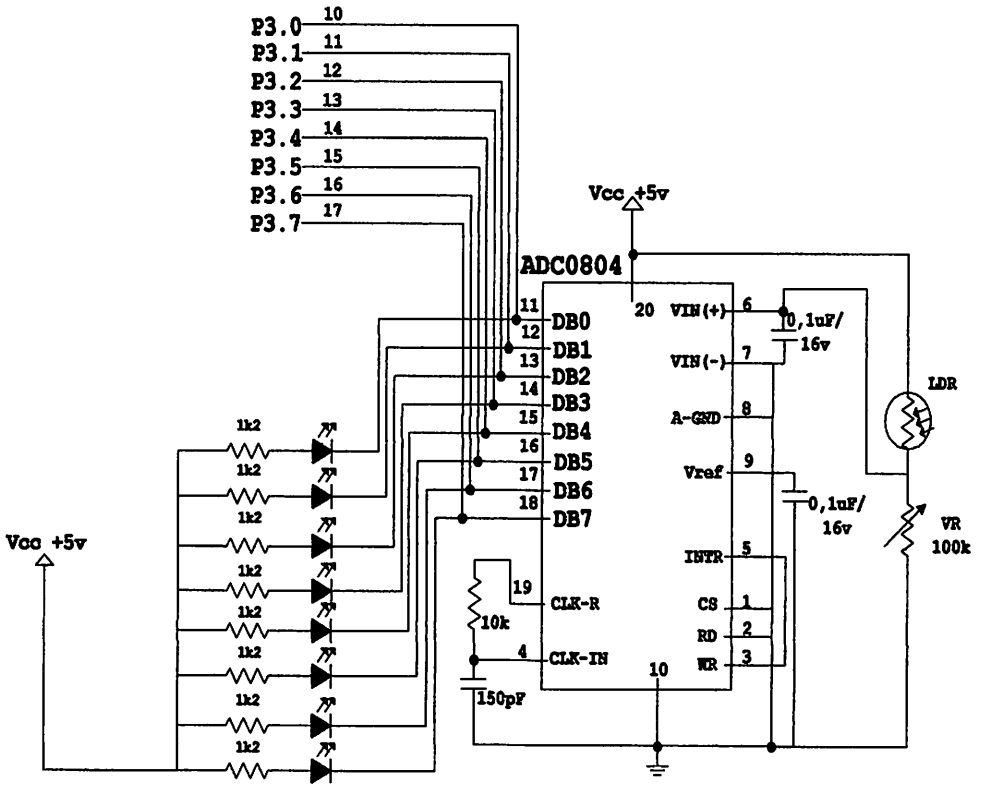
Warna bola	Tegangan input (5V)	Output tegangan rangkaian sensor (V)
Kuning	5	4,25
Putih	5	3,41
Hitam	5	1,50

Tabel 4.2. hasil pengukuran

Dari tabel di atas dapat disimpulkan bahwa pada dasarnya sensor warna bekerja dengan baik dan untuk selisih hasil pengukuran yang dilakukan pada warna yang sama karena disebabkan oleh sumber cahaya dari luar, sehingga mengakibatkan perubahan nilai pada output tegangan yang dihasilkan oleh rangkaian sensor warna.

4.3. Pengujian ADC.

Seperti yang kita ketahui bahwa mikrokontroler AT89C51 memerlukan data biner maka ADC harus bekerja dengan baik, untuk mengetahui bahwa ADC bekerja dengan baik agar nantinya data yang di jadikan biner tersebut bisa di proses oleh mikrokontroler AT89C51 sesuai tegangan dari sensor.



Gambar 4.2. pengujian ADC 0804

Dari hasil pengujian diperoleh data seperti tabel dibawah ini:

No	input tegangan	Hasil konversi ADC D1	D2	D3	D4	D5	D6	D7	D8
1	4,25	0	0	0	1	0	0	0	1
2	3,41	0	0	0	0	1	1	1	0
3	1,50	0	0	0	1	1	0	1	0

Dari pengujian output ADC maka ADC dapat digunakan karena dari tabel di atas nilai output dari tiap warna yang dibaca oleh sensor warna setelah diolah oleh ADC dan keluaran ADC adalah biner pada pot-pot ADC.

4.4. Pengujian rangkaian mekanik lengan robot

Pada pengujian rangkaian lengan robot yang telah di rangkai dengan perangkat keras yang lain maka lengan robot dapat dioprasikan untuk melakukan pekerjaan yang telah ditujukan kepada lengan robot tersebut, langkah yang dilakukan adalah melakukan percobaan terhadap lengan robot tersbut, untuk melihat lengan apakah dapat bekerja dengan baik maka lengan robot di oprasikan untuk mengambil bola yang telah disediakan, dan dilaksanakan sebanyak 20 kali.

Untuk tingkat kesalahan-kesalahan yang akan di amati adah kesalahan bola tidak pada tempatnya, bola terlepas, bola tidak diletakkan dan bola tidak terangkat.

Setelah melakukan beberapakali percobaan maka diperoleh tabel seperti di bawah ini:

Pengambilan bola	Bola tidak pada tempatnya	Bola terlepas	Bola tidak diletakkan	Bola tidak terangkat
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	1
8	0	1	1	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	1	0	0
15	0	0	0	0
16	0	0	1	0
17	0	0	0	1
18	0	0	0	1
19	0	0	0	0
20	0	0	0	0

Tabel 4.3. tabel hasil percobaan.

Kesalahan bola tidak pada tempatna : 0

Kesalahan bola terlepas : 2 kali

Kesalahan bola tidak diletakkan : 2 kali

Kesalahan bola tidak terangkat : 3 kali

Seperti yang kita lihat pada tabel, setelah lakukan sebanyak 20 kali lengan robot melakukan pengambilan bola, tingkat kesalahan yang dilakukan oleh lengan robot tersebut hampir tidak terjadi, dan kesalahan-kesalahan yangdilakukan oleh lengan robot tersebut adalah suatu hal yang wajar, mungkin jika dilakukan penyempurnaan pada peralatan-peralatan yang mendukung maka tingkat kesalahan tersebut bisa dikurangi.

4.2. Pengujian Pengujian Perangkat lunak.

Pengujian perangkat lunak di sini adalah pengujian terhadap program yang di masukkan kedalam mikrokontroler AT89C51. Program yang telah di buat adalah seperti di bawah ini:

1). Program cek warna :

```
;-------------------------------------------------------------------------------
----------
;Program cek warna
;Untuk cek apakah ada perubahan warna tidak dan ; apakah ada data
yang sama antara
data pembanding ; dengan data yang masuk
```

```
;-----------------------------------------------------------------------------------
----------
standby:    call        delai           ; panggil subprogram dengan alamat delai

cek:                                    ; cek ada tidaknya benda
        call    delai5                  ; panggil sub program dengan alamat delai5
        mov     a,p3                    ; pindahkan data pada port 3 ke Akumulator (A)
        cjne    a,#00000011b,merah      ; bandingkan data pada A dengan 00000011b,
                                         jika sama/ tidak ada perubahan lompat ke
                                         alamat cek, jika tidak lompat ke alamat merah
        sjmp    cek                     ; lompat ke alamat cek

kuning:
        mov     a,p3
        cjne    a,#00010001b,hijau
        call    ambil_kuning
        sjmp    standby
putih:
        mov     a,p3
        cjne    a,#00001110b,hitam
        call    ambil_putih
        sjmp    standby
hitam:
        mov     a,p3
        cjne    a,#00011010b,cek
        call    ambil_hitam
        sjmp    standby
```

```
ambil_kuning:
        call    ambil
        call    putar_kanan
        call    putar_kanan
        call    lepas
        call    putar_kiri
        call    putar_kiri
        ret
ambil_putih:
        call    ambil
        call    putar_kiri
        call    putar_kiri
        call    lepas
        call    putar_kanan
        call    putar_kanan
        ret
ambil_hitam:
        call    ambil
        call    putar_kiri
        call    putar_kiri
        call    putar_kiri
```

```
        call    lepas
        call    putar_kanan
        call    putar_kanan
        call    putar_kanan
        ret
```

SUB PROGRAM MENGGERAKKAN LENGAN & PENCENGKERAM

; lengan turun untuk mengambil benda

```
ambil:
        clr     p1.0            ; keluarkan logika 0 pada port 1.0 ( lengan bergerak turun)
        call    Delai           ; panggil sub program dengan alamat delai
        setb    p1.0            ; keluarkan logika 1 pada port 1.0  (lengan berhenti
bergerak)
        clr     p1.2            ; keluarkan logika 0 pada p0rt 1.2 ( jari bergerak
                                  mencengkaram)
        call    Delai3
        setb    p1.2            ; keluarkan logika 1 pada port 1.2 (( jari berhenti
                                  mencengkaram)

        call    delai3
        clr     p1.1            ; keluarkan logika 0 pada port 1.1 ( lengan bergerak naik)

        call    Delai
        call    Delai3
```

```
        setb    p1.1            ; keluarkan logika 1 pada port 1,1  (lengan berhenti
bergerak)
        call    Delai3
        ret
; lengan turun untuk meletakkan benda
lepas:
        clr     p1.0            ; keluarkan logika 0 pada port 1.0 ( lengan bergerak turun)
        call    Delai           ; panggil sub program dengan alamat delai
        setb    p1.0            ; keluarkan logika 1 pada port 1.0  (lengan berhenti
bergerak)
        clr     p1.2            ; keluarkan logika 0 pada port 1.2 ( jari bergerak
                                  membuka cengkaraman)
        call    Delai3
        setb    p1.3            ; keluarkan logika 1 pada port 1.3 (( jari berhenti
                                  membuka cengkaraman)

        call    delai3
        clr     p1.1            ; keluarkan logika 0 pada port 1.1 ( lengan bergerak naik)

        call    Delai
        call    Delai3
        setb    p1.1            ; keluarkan logika 1 pada port 1,1  (lengan berhenti
bergerak)
        call    Delai3
        ret
```

```
; motor steper berputar kekanan
putar_kanan:    mov     r5,#20          ; isi r5 dengan 20
kanan:
        mov     p2,#001h        ; keluarkan data 01h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        mov     p2,#002         ; keluarkan data 02h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        mov     p2,#004h        ; keluarkan data 04h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        mov     p2,#008h        ; keluarkan data 08h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        djnz    r5,kanan        ; kurangi isi r5 dengan 1 jika belum 0 loncat ke alamat
kanan
                                Jika telah sama dengan 0 lanjutkan program
        Ret
; motor steper berputar kekiri
putar_kiri:     mov     r5,#20          ; isi r5 dengan 20
kiri:
        mov     p2,#008h        ; keluarkan data 08h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        mov     p2,#004h        ; keluarkan data 04h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        mov     p2,#002h        ; keluarkan data 02h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        mov     p2,#001h        ; keluarkan data 01h pada port 2
        call    delay1          ; panggil sub program dengan alamat delay1
        djnz    r5,kiri         ; kurangi isi r5 dengan 1 jika belum 0 loncat ke alamat kiri
```

```
                                        Jika telah sama dengan 0 lanjutkan program
            ret
SUB PROGRAM DELAI (tunda waktu)

Delai:      Mov   r1,#10          ; isi r1 (register1)dengan 10
Loop_1:     Mov   r2,#150         ; isi r2 (register1)dengan 150
Loop_2:     Mov   r3,#250         ; isi r3 (register1)dengan 250
Loop_3:     Djnz  r3,loop_3       ; kurangi isi r3 dengan 1 jika belum 0 loncat ke alamat
loop_3
                                        Jika telah sama dengan 0 lanjutkan program
            Djnz  r2,loop_2
            Djnz  r1,loop_1
            ret
Delay1:     Mov   r0,#70
Delay2:     Mov   r2,#70
            Djnz  r2,$
            Djnz  r0,Delay2
            Ret
Delai3:     Mov   r1,#30
Loop_a:     Mov   r2,#150
Loop_b:     Mov   r3,#150
Loop_c:     Djnz  r3,loop_c
            Djnz  r2,loop_b
            Djnz  r1,loop_a
            ret
Delai5:     Mov   r1,#10
Loop_d:     Mov   r2,#250
Loop_e:     Mov   r3,#250
```

```
Loop_f:   Djnz  r3,loop_f
          Djnz  r2,loop_e
          Djnz  r1,loop_d
          Ret
```

Program di atas adalah program perintan terhadap perangkat keras lengan robot tersebut, dan jika pada sensor yang di pasang telah mendeteksi warna yang telah di programkan pada mikrokontroler AT89C51 maka mikrokontroler tersebut akan mengeluarkan perintah yang sesuai dengan set penempatan bola yang sesuai dengan program yang kita buat.

Pada pengujian program yang telah kami buat, frekwensi terjadinya kesalahan pada program yang telah kami buat hampir tidak terjadi.

INSTITUT TEKNOLOGI NASIONAL

ITN

MALANG

BAB V

PENUTUP

5.1. Kesimpulan

Dari hasil pembahasan tentang pembuatan rangkaian dan analisa program, maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Cara kerja dari lengan robot pemisah bola berwarna ini adalah ketika bola berwarna diletakkan didepan sensor maka output ADC akan berubah, perubahan ini yang akan dibaca mikrokontroler yang kemudian dibandingkan, jika data yang dibandingkan ada yang sama maka secara langsung mikrokontroler akan memberikan instruksi pada rangkaian driver motor DC dan driver motor steper untuk menggerakkan lengan mengambil benda kemudian berputar menuju tempat dengan warna yang sama dengan warna bola yang diambil, lalu lengan turun meletakkan benda setelah itu lengan kembali naik dan berputar ke tempat semula untuk menunggu benda berwarna selanjutnya.
2. Rangkaian robot lengan pemisah warna ini bekerja sesuai dengan program yang telah disimpan di dalam mikrokontroler AT89C51. Ternyata dengan mikrokontroler AT89C51 dapat digunakan untuk mengontrol LDR sebagai pendeteksi warna dan kemudian mengontrol driver motor DC dan driver motor steper untuk menggerakkan lengan

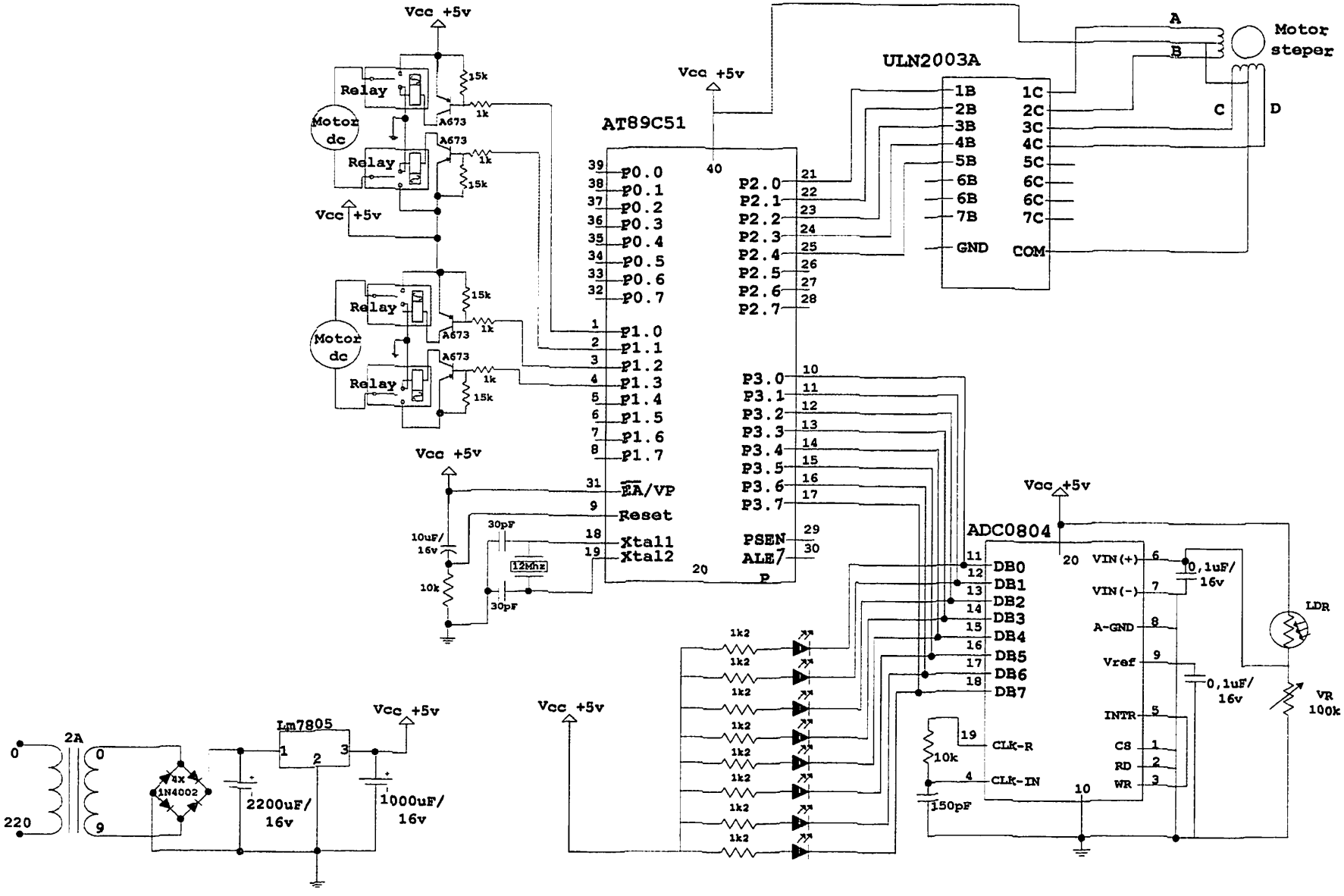
sehingga dapat mengambil benda dan menempatkan benda sesuai dengan tempatnya.

3. Robot lengan pemisah warna tidak dapat membaca warna selain warna yang disediakan, jika menginginkan warna lain harus menset ulang program.

5.2. Saran

Untuk pengembangan robot lengan pemisah warna yang lebih baik, dapat disampaikan saran sebagai berikut :

1. Rangkaian sensor yang digunakan kurang sensitif terhadap perubahan warna, hal ini disebabkan karena LDR hanya membaca intensitas pantulan cahaya dari benda berwarna. Agar lebih sensitif dalam pembacaan warna disarankan menggunakan sensor warna.
2. Agar data yang dikeluarkan ADC tidak bertabrakan antara satu dengan yang lainnya hendaknya benda berwarna diletakkan setelah mekanik telah kembali pada posisi *standby* / semula.


Vcc +5v
Relay
Motor dc
15k
1k
A673
AT89C51
ULN2003A
Motor steper
A
B
C
D
1B
2B
3B
4B
5B
6B
6B
7B
GND
1C
2C
3C
4C
5C
6C
6C
7C
COM
P0.0
P0.1
P0.2
P0.3
P0.4
P0.5
P0.6
P0.7
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
EA/VP
Reset
Xtal1
Xtal2
P2.0
P2.1
P2.2
P2.3
P2.4
P2.5
P2.6
P2.7
P3.0
P3.1
P3.2
P3.3
P3.4
P3.5
P3.6
P3.7
PSEN
ALE/P
10uF/16v
10k
30pF
12Mhz
ADC0804
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
VIN(+)
VIN(-)
A-GND
Vref
INTR
CS
RD
WR
CLK-R
CLK-IN
0,1uF/16v
150pF
LDR
VR 100k
1k2
Lm7805
2A
220
4x 1N4002
2200uF/16v
1000uF/16v

DAFTAR PUSTAKA

Eko Putra Agfianto. 2003 **BELAJAR MIKROKONTROLER AT 89C51/S2/S5**. Teoro dan aplikasi, Gava Media, Jakarta.

Woollard Barry, 1988, **ELEKTRONIKA PRAKTIS**, PT Pradaya Paramita, Jakarta.

Warsito S, 1994, **VODOMINIKUM ELEKTRONIKA**, Gramedia Jakarta.

Samuel C. Lee,1982. **RANGKAIAN DIGITAL DAN LOGIKA**. Profesor OF elektrcal Enginering University Of Oklahoma. (Terjemahan Ir Sutisno, M. Sc) Penerbit Erlangga, JL. H. Baping Raya No: 100 Ciracas Jakarta 13740. E-maile :mahameru @rad.net.id.

Web"www.atmel.com.

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO D III

LEMBAR ASISTENSI BIMBINGAN TUGAS AKHIR

NAMA : Exzan triyono
NIM : 0152065
JUDUL : Lengan Robot Pemisah Bola Berwarna Berbasis MK. AT89C51

NO	TGL	MATERI	PARAF
1	22/12 06.	Konsultasi alat	
2	08/01 07	Konsultasi penqujian	
3	16/01 07	Penqujian alat.	
4	29/01 07	Konsultasi bab I & II	
5	07/02 07	Konsultasi bab III, IV, V	
6	27/02 07	Revisi bab IV & V.	
7.	09/03 07	acc ujian	

Malang 11 januari 2007
Mengetahui
Dosen Pembimbing

Ir Teguh Herbasuki, MT
NIP.

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO D III
PROGRAM STUDI TEKNIK ENERGI LISTRIK
MALANG

LEMBAR BIMBINGAN TUGAS AKHIR

Nama	: EXZAN TRIYONO
NIM	: 0152065
Jurusa	: Teknik Elektro D-III
Program Studi	: Energi Listrik
Hari/Tanggal	: Rabu 21 Maret 2007
Judul Tugas Akhir	: Lengan Robot Pemisah Bola BerwarnaBerbasi Mikrkontroler AT89C51.
Tanggal pengajuan Tugas Akhir	: 21 Desember 2006
Selesai Penulisan Tugas Akhir	: 21 April 2007
Telah Dievaluasi Dengan Nilai	: 85

Mengetahui
Ketua Jurusan Elektro DIII

Ir. H Choirul Saleh, MT
NIP.Y. 1010088190

Diperiksa Dan Disetujui
Dosen Pembimbing

Ir. Teguh Herbasuki, MT
NIP.P. 1038900209

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO D III
PROGRAM STUDI TEKNIK ENERGI LISTRIK
MALANG

LEMBAR PERBAIKAN TUGAS AKHIR

Telah di lakukan perbaikan oleh:

Nama : Exzan triyono

NIM : 0152065

Jurusa : TEKNIK ELEKTRO D-III

Program Studi : ENERGI LISTRIK

Hari/Tanggal : Rabu 21 Maret 2007

No	Materi Perbaikan	Paraf
1.	Dibatasan masalah bahwa robot tersebut hanya mampu memisahkan tiga warna bola yaitu putih, kuning dan hitam.	

Telah Diperiksa/Disetujui:

Penguji I

Ir. Yunior Siahaan

Penguji I

Ir. Djojo Priatmono, MT

Mengetahui:

Dosen Pembimbing

Ir. Teguh Herbasuki, MT

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO D III
PROGRAM STUDI TEKNIK ENERGI LISTRIK
MALANG

BERITA ACARA UJIAN TUGAS AKHIR
FAKULTAS TEKNOLOGI INDUSTRI

Nama	: EXZAN TRIYONO
NIM	: 0152065
Jurusa	: Teknik Elektro D-III
Program Studi	: Energi Listrik
Hari/Tanggal	: Rabu 21 Maret 2007
Judul Tugas Akhir	: Lengan Robot Pemisah Bola Berwarna Berbasis Mikrokontroler AT89C51.

Dipertahankan dihadapan team penguji Tugas Akhir Jenjang Diploma (DIII) pada:

Hari	: Rabu
Tanggal	: 21 Maret 2007
Dengan nilai	: 77,50 (B+)

Panitia Ujian Tugas Akhir

Ketua	Sekretaris
Ir. Moehtar Asroni, MSME	Ir. H Choirul Saleh, MT

Anggota penguji

Pengiji Pertama	Penguiji Kedua
Ir. Yunior Siahaan	Ir. Djojo Priatmono, MT

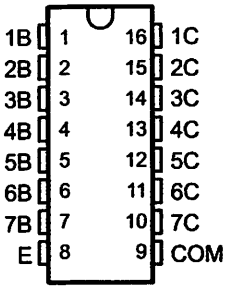
LAMPIRAN

SLRS027 – DECEMBER 1976 – REVISED APRIL 1993

HIGH-VOLTAGE HIGH-CURRENT DARLINGTON TRANSISTOR ARRAYS

- **500-mA Rated Collector Current (Single Output)**
- **High-Voltage Outputs . . . 50 V**
- **Output Clamp Diodes**
- **Inputs Compatible With Various Types of Logic**
- **Relay Driver Applications**
- **Designed to Be Interchangeable With Sprague ULN2001A Series**

D OR N PACKAGE
(TOP VIEW)

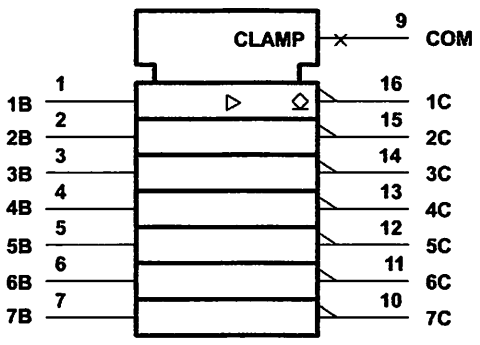


description

The ULN2001A, ULN2002A, ULN2003A, and ULN2004A are monolithic high-voltage, high-current Darlington transistor arrays. Each consists of seven npn Darlington pairs that feature high-voltage outputs with common-cathode clamp diodes for switching inductive loads. The collector-current rating of a single Darlington pair is 500 mA. The Darlington pairs may be paralleled for higher current capability. Applications include relay drivers, hammer drivers, lamp drivers, display drivers (LED and gas discharge), line drivers, and logic buffers. For 100-V (otherwise interchangeable) versions, see the SN75465 through SN75469.

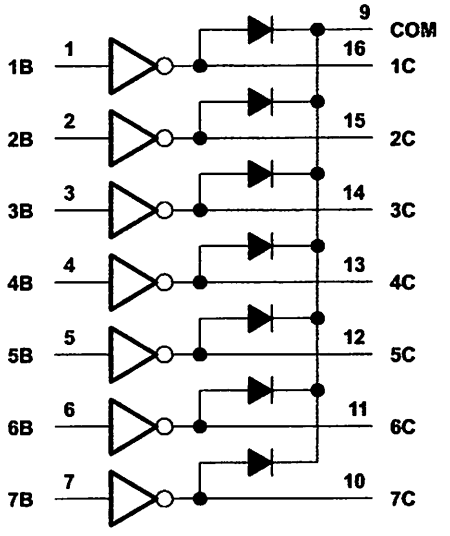
The ULN2001A is a general-purpose array and can be used with TTL and CMOS technologies. The ULN2002A is specifically designed for use with 14- to 25-V PMOS devices. Each input of this device has a zener diode and resistor in series to control the input current to a safe limit. The ULN2003A has a 2.7-kΩ series base resistor for each Darlington pair for operation directly with TTL or 5-V CMOS devices. The ULN2004A has a 10.5-kΩ series base resistor to allow its operation directly from CMOS devices that use supply voltages of 6 to 15 V. The required input current of the ULN2004A is below that of the ULN2003A, and the required voltage is less than that required by the ULN2002A.

logic symbol†

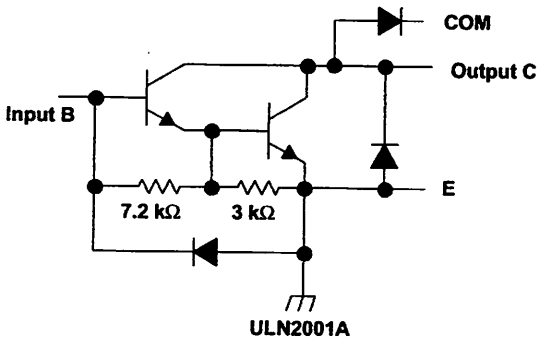


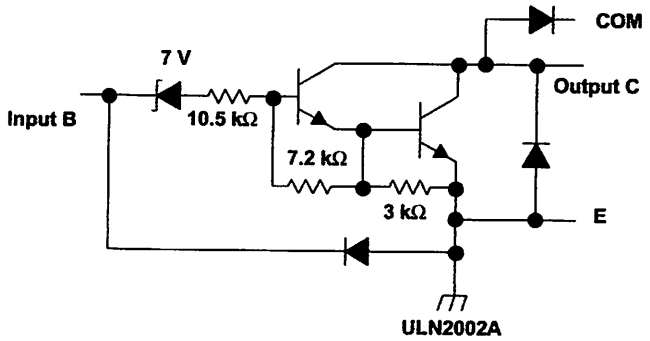
† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

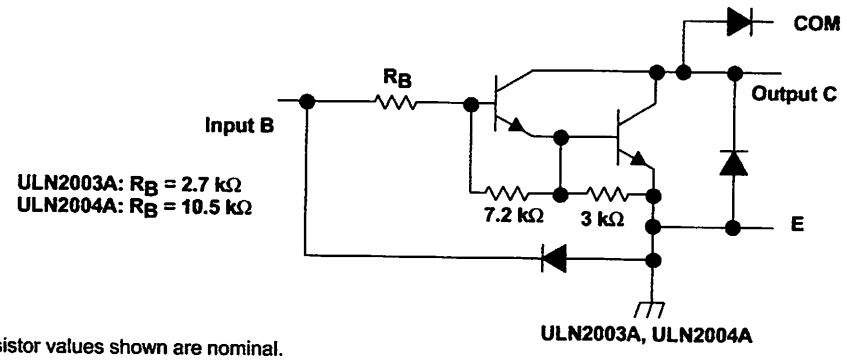
logic diagram



schematics (each Darlington pair)







All resistor values shown are nominal.

absolute maximum ratings at 25°C free-air temperature (unless otherwise noted)

Collector-emitter voltage	50 V
Input voltage, V_I (see Note 1)	30 V
Peak collector current (see Figures 14 and 15)	500 mA
Output clamp current, I_{OK}	500 mA
Total emitter-terminal current	−2.5 A
Continuous total power dissipation	See Dissipation Rating Table
Operating free-air temperature range, T_A	−20°C to 85°C
Storage temperature range, T_{stg}	−65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C

NOTE 1: All voltage values are with respect to the emitter/substrate terminal E, unless otherwise noted.

DISSIPATION RATING TABLE

PACKAGE	T_A = 25°C POWER RATING	DERATING FACTOR ABOVE T_A = 25°C	T_A = 85°C POWER RATING
D	950 mW	7.6 mW/°C	494 mW
N	1150 mW	9.2 mW/°C	598 mW

electrical characteristics, T_A = 25°C (unless otherwise noted)

PARAMETER		TEST FIGURE	TEST CONDITIONS		ULN2001A MIN	ULN2001A TYP	ULN2001A MAX	ULN2002A MIN	ULN2002A TYP	ULN2002A MAX	UNIT
$V_{I(on)}$	On-state input voltage	6	V_{CE} = 2 V,	I_C = 300 mA						13	V
$V_{CE(sat)}$	Collector-emitter saturation voltage	5	I_I = 250 µA,	I_C = 100 mA		0.9	1.1		0.9	1.1	V
			I_I = 350 µA,	I_C = 200 mA		1	1.3		1	1.3	
			I_I = 500 µA,	I_C = 350 mA		1.2	1.6		1.2	1.6	
V_F	Clamp forward voltage	8	I_F = 350 mA			1.7	2		1.7	2	V
I_{CEX}	Collector cutoff current	1	V_{CE} = 50 V,	I_I = 0			50			50	µA
		2	V_{CE} = 50 V, T_A = 70°C	I_I = 0			100			100	
				V_I = 6 V						500	
$I_{I(off)}$	Off-state input current	3	V_{CE} = 50 V, T_A = 70°C	I_C = 500 µA,	50	65		50	65		µA
I_I	Input current	4	V_I = 17 V						0.82	1.25	mA
I_R	Clamp reverse current	7	V_R = 50 V,	TA = 70°C			100			100	µA
			V_R = 50 V				50			50	
h_{FE}	Static forward current transfer ratio	5	V_{CE} = 2 V,	I_C = 350 mA	1000						
C_i	Input capacitance		V_I = 0,	f = 1 MHz		15	25		15	25	pF

electrical characteristics, T_A = 25°C (unless otherwise noted)

PARAMETER		TEST FIGURE	TEST CONDITIONS		ULN2003A MIN	ULN2003A TYP	ULN2003A MAX	ULN2004A MIN	ULN2004A TYP	ULN2004A MAX	UNIT
$V_{I(on)}$	On-state input voltage	6	V_{CE} = 2 V	I_C = 125 mA						5	V
				I_C = 200 mA			2.4			6	
				I_C = 250 mA			2.7				
				I_C = 275 mA						7	
				I_C = 300 mA			3				
				I_C = 350 mA						8	
$V_{CE(sat)}$	Collector-emitter saturation voltage	5	I_I = 250 µA,	I_C = 100 mA		0.9	1.1		0.9	1.1	V
			I_I = 350 µA,	I_C = 200 mA		1	1.3		1	1.3	
			I_I = 500 µA,	I_C = 350 mA		1.2	1.6		1.2	1.6	
I_{CEX}	Collector cutoff current	1	V_{CE} = 50 V,	I_I = 0			50			50	µA
		2	V_{CE} = 50 V, T_A = 70°C	I_I = 0			100			100	
				V_I = 1 V						500	
V_F	Clamp forward voltage	8	I_F = 350 mA			1.7	2		1.7	2	V
$I_{I(off)}$	Off-state input current	3	V_{CE} = 50 V, T_A = 70°C	I_C = 500 µA,	50	65		50	65		µA
I_I	Input current	4	V_I = 3.85 V			0.93	1.35				mA
			V_I = 5 V						0.35	0.5	
			V_I = 12 V						1	1.45	
I_R	Clamp reverse current	7	V_R = 50 V				50			50	µA
			V_R = 50 V,	T_A = 70°C			100			100	
C_i	Input capacitance		V_I = 0,	f = 1 MHz		15	25		15	25	pF

switching characteristics, T_A = 25°C

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Propagation delay time, low-to-high-level output	See Figure 9		0.25	1	µs
t_{PHL}	Propagation delay time, high-to-low-level output			0.25	1	µs
V_{OH}	High-level output voltage after switching	V_S = 50 V, I_O = 300 mA, See Figure 10	V_S−20			mV

PARAMETER MEASUREMENT INFORMATION

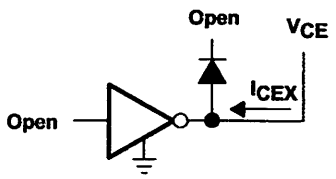


Figure 1. I_{CEX} Test Circuit

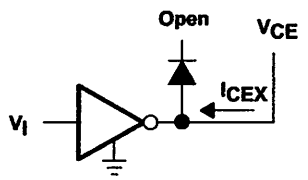


Figure 2. I_{CEX} Test Circuit

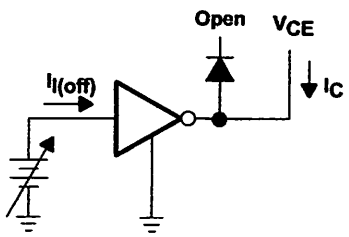


Figure 3. $I_{I(off)}$ Test Circuit

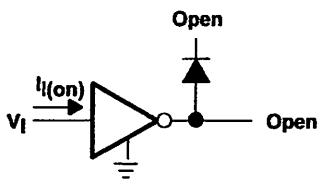
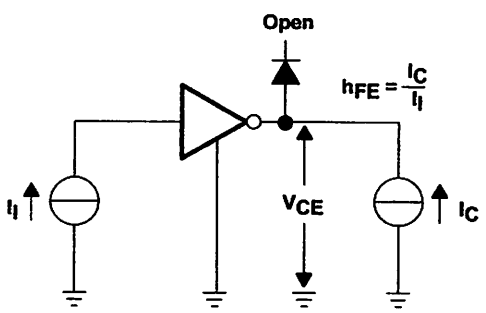


Figure 4. I_I Test Circuit



NOTE: I_I is fixed for measuring $V_{CE(sat)}$, variable for measuring h_{FE}.

Figure 5. h_{FE}, $V_{CE(sat)}$ Test Circuit

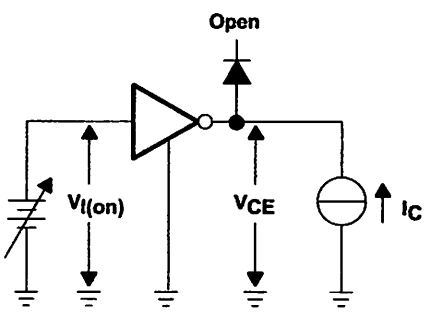


Figure 6. $V_{I(on)}$ Test Circuit

PARAMETER MEASUREMENT INFORMATION

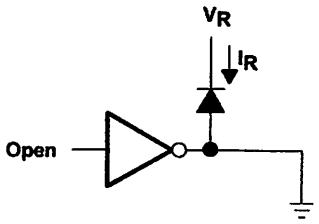


Figure 7. I_R Test Circuit

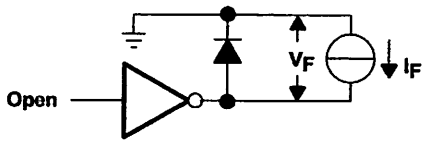


Figure 8. V_F Test Circuit

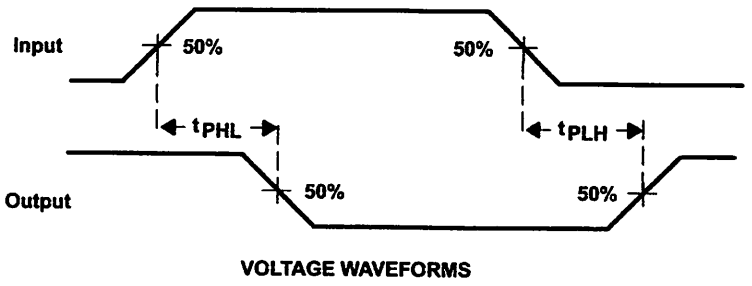
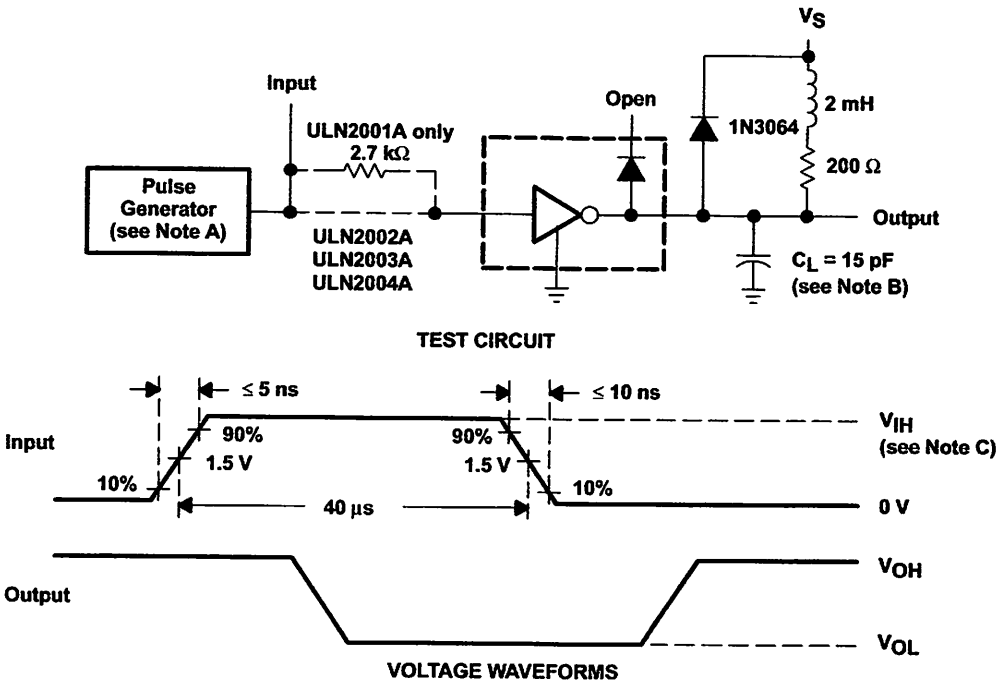


Figure 9. Propagation Delay Time Waveforms



NOTES: A. The pulse generator has the following characteristics: PRR = 12.5 kHz, Z_O = 50 Ω.
B. C_L includes probe and jig capacitance.
C. For testing the ULN2001A and the ULN2003A, V_{IH} = 3 V; for the ULN2002A, V_{IH} = 13 V; for the ULN2004A, V_{IH} = 8 V.

Figure 10. Latch-Up Test Circuit and Voltage Waveforms

TYPICAL CHARACTERISTICS

COLLECTOR-EMITTER SATURATION VOLTAGE vs COLLECTOR CURRENT (ONE DARLINGTON)

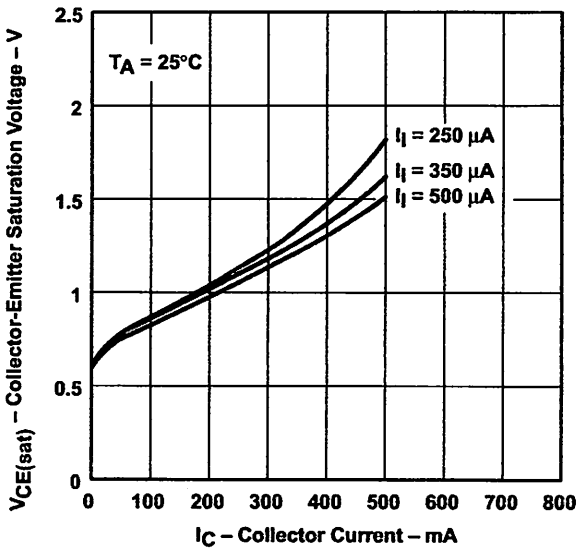


Figure 11

COLLECTOR-EMITTER SATURATION VOLTAGE vs TOTAL COLLECTOR CURRENT (TWO DARLINGTONS PARALLELED)

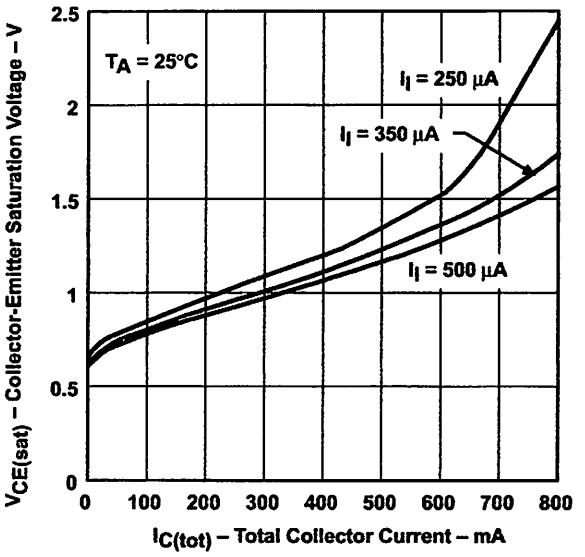


Figure 12

COLLECTOR CURRENT vs INPUT CURRENT

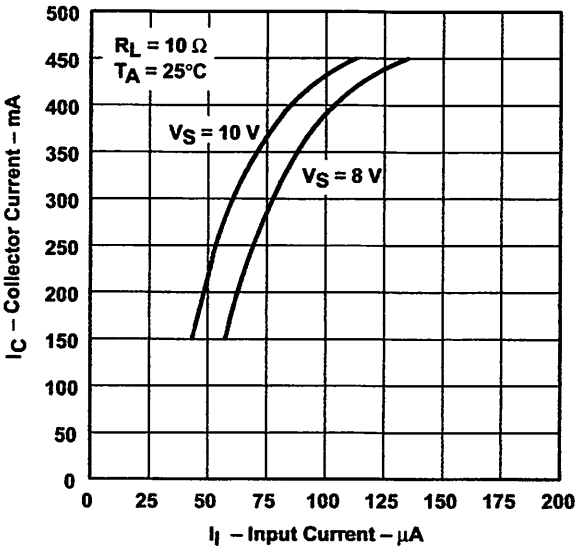


Figure 13

THERMAL INFORMATION

D PACKAGE
MAXIMUM COLLECTOR CURRENT
vs
DUTY CYCLE

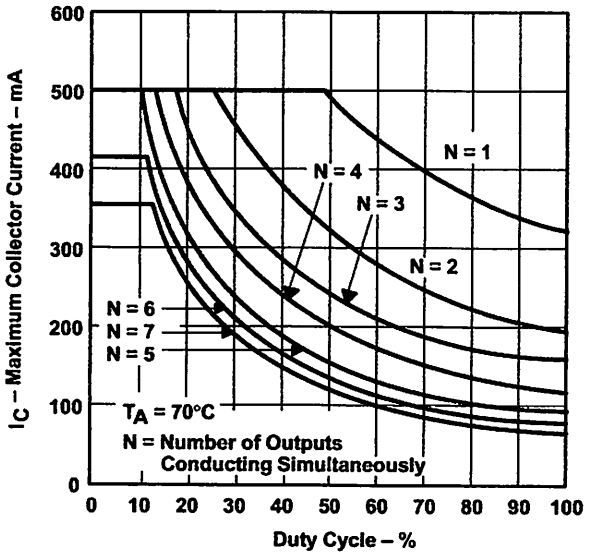


Figure 14

N PACKAGE
MAXIMUM COLLECTOR CURRENT
vs
DUTY CYCLE

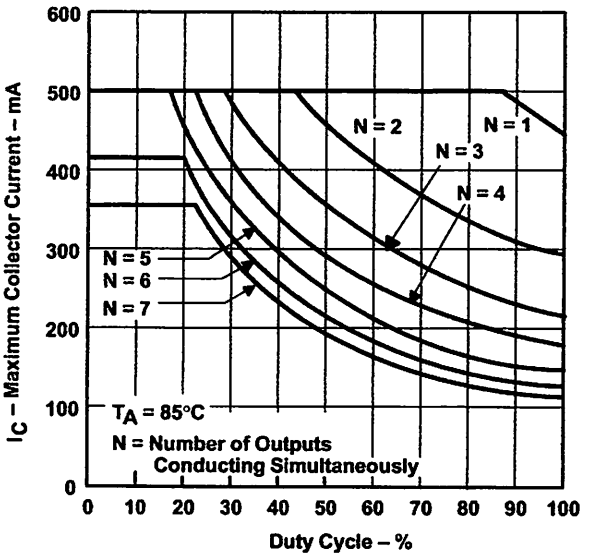


Figure 15

APPLICATION INFORMATION

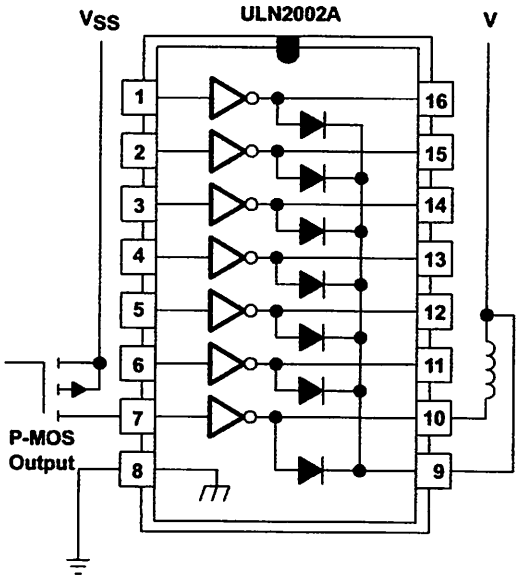


Figure 16. P-MOS to Load

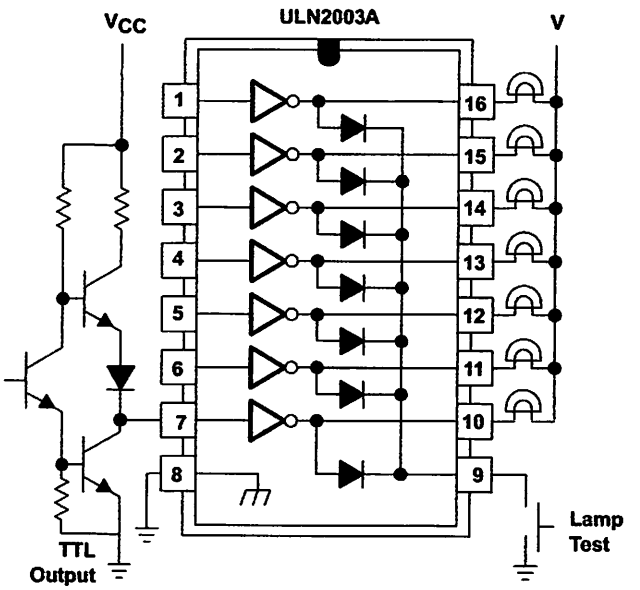


Figure 17. TTL to Load

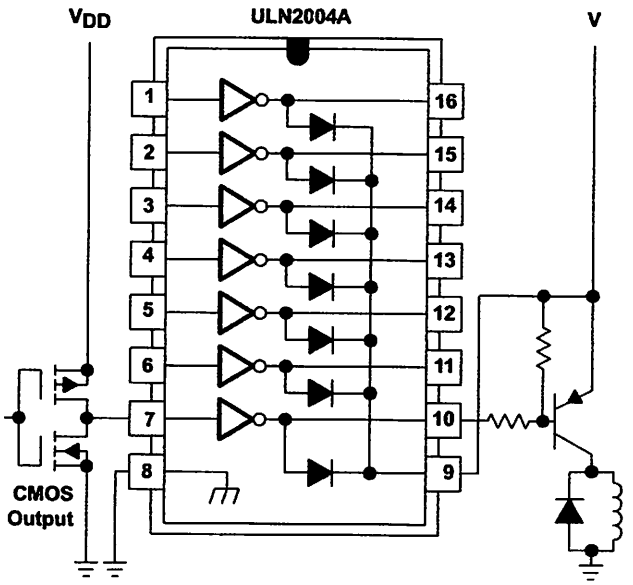


Figure 18. Buffer for Higher Current Loads

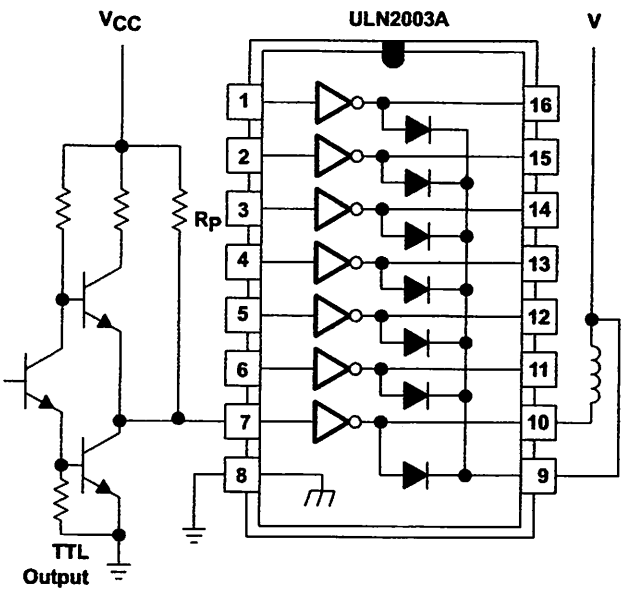


Figure 19. Use of Pullup Resistors to Increase Drive Current

eatures

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

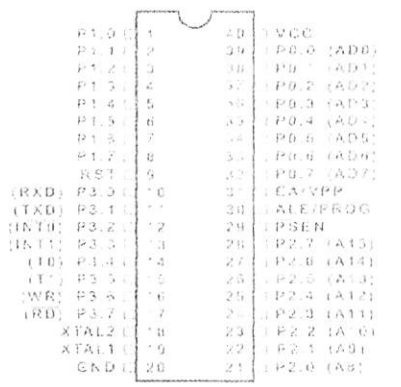
escription

he AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K ytes of Flash Programmable and Erasable Read Only Memory (PEROM). The evice is manufactured using Atmel's high density nonvolatile memory technology nd is compatible with the industry standard MCS-51™ instruction set and pinout. The n-chip Flash allows the program memory to be reprogrammed in-system or by a conentional nonvolatile memory programmer. By combining a versatile 8-bit CPU with ash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which ovides a highly flexible and cost effective solution to many embedded control applitions.

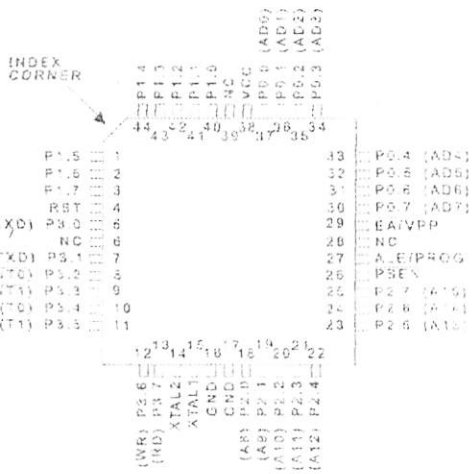
(continued)

in Configurations

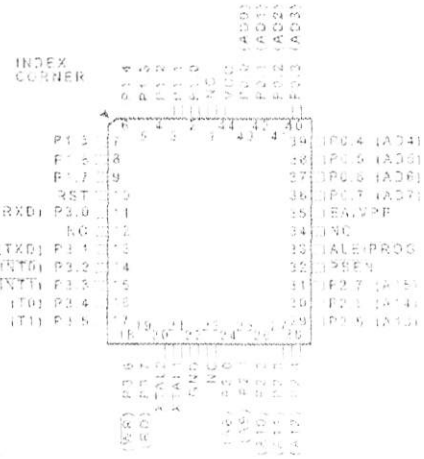
PDIP



PQFP/TQFP



PLCC

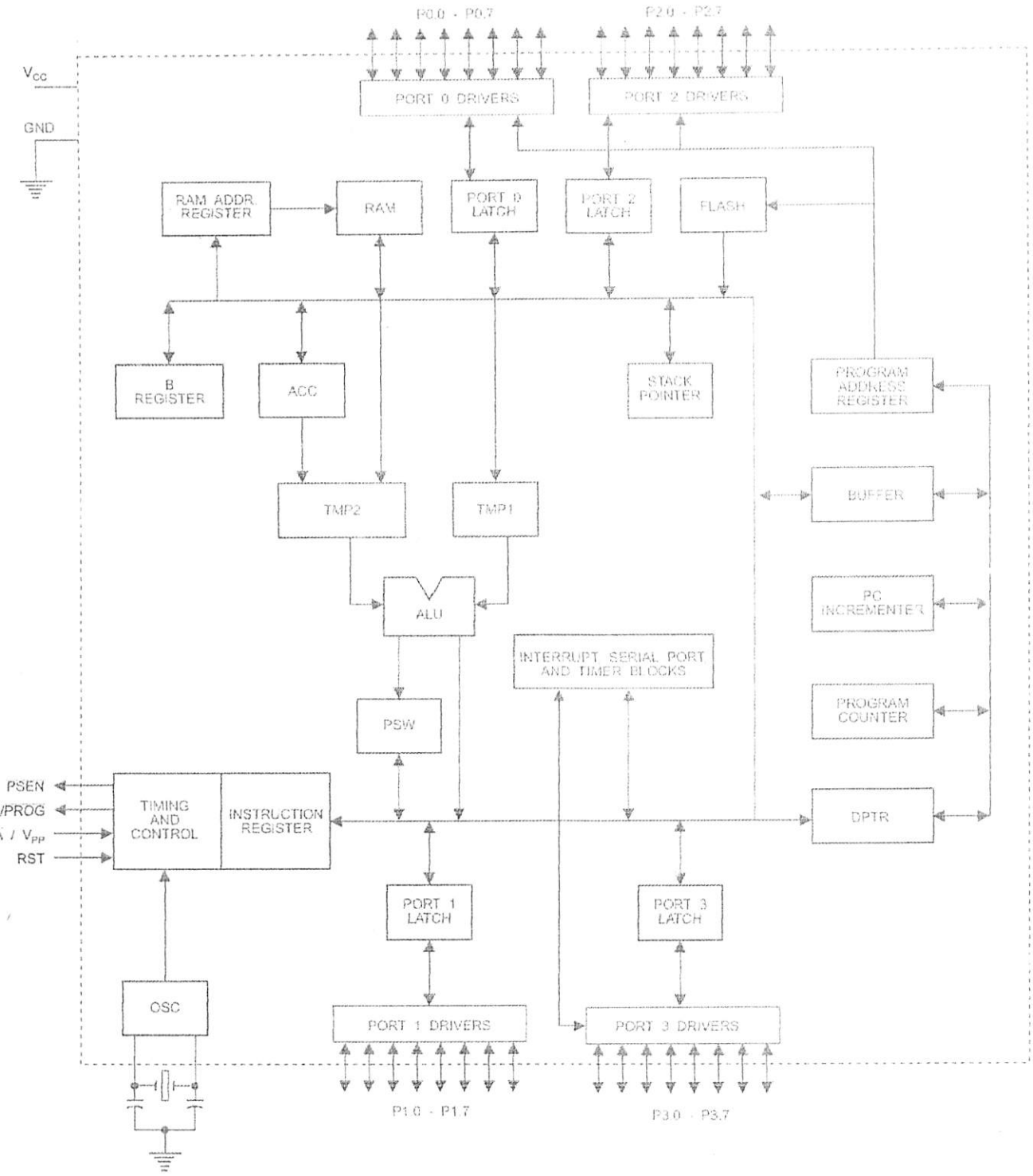


8-Bit Microcontroller with 4K Bytes Flash

AT89C51

0265F-A–12/97

k Diagram


P0.0 - P0.7
P2.0 - P2.7
V_{CC}
GND
PORT 0 DRIVERS
PORT 2 DRIVERS
RAM ADDR. REGISTER
RAM
PORT 0 LATCH
PORT 2 LATCH
FLASH
B REGISTER
ACC
STACK POINTER
PROGRAM ADDRESS REGISTER
BUFFER
TMP2
TMP1
ALU
PC INCREMENTER
INTERRUPT, SERIAL PORT AND TIMER BLOCKS
PROGRAM COUNTER
PSW
PSEN
/PROG
/ V_{PP}
RST
TIMING AND CONTROL
INSTRUCTION REGISTER
DPTR
PORT 1 LATCH
PORT 3 LATCH
OSC
PORT 1 DRIVERS
PORT 3 DRIVERS
P1.0 - P1.7
P3.0 - P3.7

AT89C51 provides the following standard features: 4K ɜs of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit ɜr/counters, a five vector two-level interrupt architecture, ıll duplex serial port, on-chip oscillator and clock cir- ry. In addition, the AT89C51 is designed with static logic operation down to zero frequency and supports two tware selectable power saving modes. The Idle Mode ps the CPU while allowing the RAM, timer/counters, ial port and interrupt system to continue functioning. The ver Down Mode saves the RAM contents but freezes oscillator disabling all other chip functions until the next dware reset.

n Description

c
pply voltage.

ID
ɔund.

rt 0
rt 0 is an 8-bit open drain bidirectional I/O port. As an ıput port each pin can sink eight TTL inputs. When 1s written to port 0 pins, the pins can be used as high- ɔedance inputs.

rt 0 may also be configured to be the multiplexed low- ler address/data bus during accesses to external pro- ım and data memory. In this mode P0 has internal pul- s.

rt 0 also receives the code bytes during Flash program- ıg, and outputs the code bytes during program verifica- ı. External pullups are required during program verifica- ı.

rt 1
rt 1 is an 8-bit bidirectional I/O port with internal pullups. e Port 1 output buffers can sink/source four TTL inputs. ıen 1s are written to Port 1 pins they are pulled high by internal pullups and can be used as inputs. As inputs, rt 1 pins that are externally being pulled low will source rrent (I_{IL}) because of the internal pullups.

rt 1 also receives the low-order address bytes during ısh programming and verification.

rt 2
rt 2 is an 8-bit bidirectional I/O port with internal pullups. e Port 2 output buffers can sink/source four TTL inputs. ıen 1s are written to Port 2 pins they are pulled high by internal pullups and can be used as inputs. As inputs, rt 2 pins that are externally being pulled low will source rrent (I_{IL}) because of the internal pullups.

rt 2 emits the high-order address byte during fetches m external program memory and during accesses to ternal data memory that use 16-bit addresses (MOVX @ PTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3
Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and verification.

RST
Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$
Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{PSEN}$
Program Store Enable is the read strobe to external program memory.

/hen the AT89C51 is executing code from external pro-
ram memory, $\overline{PSEN}$ is activated twice each machine
ycle, except that two $\overline{PSEN}$ activations are skipped during
ach access to external data memory.

$\overline{A}/V_{PP}$
xternal Access Enable. $\overline{EA}$ must be strapped to GND in
rder to enable the device to fetch code from external pro-
ram memory locations starting at 0000H up to FFFFH.
lote, however, that if lock bit 1 is programmed, $\overline{EA}$ will be
nternally latched on reset.

$\overline{A}$ should be strapped to V_{CC} for internal program execu-
ons.

his pin also receives the 12-volt programming enable volt-
ge (V_{PP}) during Flash programming, for parts that require
2-volt V_{PP}.

TAL1
nput to the inverting oscillator amplifier and input to the
nternal clock operating circuit.

TAL2
Output from the inverting oscillator amplifier.

Oscillator Characteristics

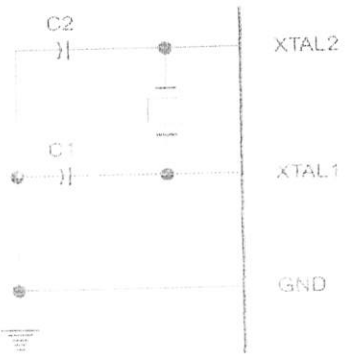
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

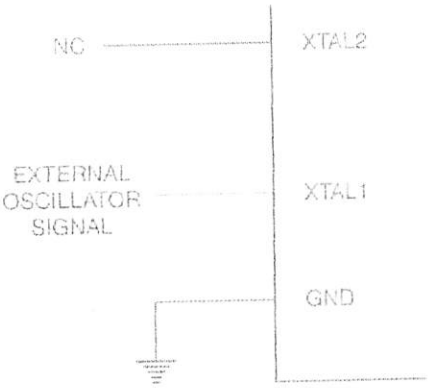
It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Figure 1. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Status of External Pins During Idle and Power Down Modes

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

›wer Down Mode

he power down mode the oscillator is stopped, and the truction that invokes power down is the last instruction ›cuted. The on-chip RAM and Special Function Regis- › retain their values until the power down mode is termi- ed. The only exit from power down is a hardware reset. set redefines the SFRs but does not change the on-chip M. The reset should not be activated before V_{CC} is tored to its normal operating level and must be held ive long enough to allow the oscillator to restart and sta- ze.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

›ck Bit Protection Modes

Program Lock Bits				Protection Type
	LB1	LB2	LB3	
1	U	U	U	No program lock features.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash is disabled.
3	P	P	U	Same as mode 2, also verify is disabled.
4	P	P	P	Same as mode 3, also external execution is disabled.

›ogramming the Flash

› AT89C51 is normally shipped with the on-chip Flash mory array in the erased state (that is, contents = FFH) ı ready to be programmed. The programming interface ›epts either a high-voltage (12-volt) or a low-voltage ›c) program enable signal. The low voltage program- ıg mode provides a convenient way to program the 89C51 inside the user's system, while the high-voltage gramming mode is compatible with conventional third ty Flash or EPROM programmers.

› AT89C51 is shipped with either the high-voltage or -voltage programming mode enabled. The respective -side marking and device signature codes are listed in following table.

	V_{PP} = 12V	V_{PP} = 5V
p-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
gnature	(030H)=1EH (031H)=51H (032H)=FFH	(030H)=1EH (031H)=51H (032H)=05H

› AT89C51 code memory array is programmed byte-by- › in either programming mode. *To program any non- ık byte in the on-chip Flash Memory, the entire memory ›t be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figures 3 and 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}/V_{PP}$ to 12V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89C51 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

ogram Verify: If lock bits LB1 and LB2 have not been grammed, the programmed code data can be read back the address and data lines for verification. The lock bits nnot be verified directly. Verification of the lock bits is nieved by observing that their features are enabled.

ip Erase: The entire Flash array is erased electrically using the proper combination of control signals and by ding ALE/$\overline{PROG}$ low for 10 ms. The code array is written h all "1"s. The chip erase operation must be executed fore the code memory can be re-programmed.

ading the Signature Bytes: The signature bytes are ad by the same procedure as a normal verification of ations 030H,

1H, and 032H, except that P3.6 and P3.7 must be pulled a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 51H indicates 89C51
(032H) = FFH indicates 12V programming
(032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

ode		RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{PP}	P2.6	P2.7	P3.6	P3.7
/rite Code Data		H	L		H/12V	L	H	H	H
ead Code Data		H	L	H	H	L	L	H	H
/rite Lock	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L		H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
hip Erase		H	L	(1)	H/12V	H	L	L	L
ead Signature Byte		H	L	H	H	L	L	L	L

te: 1. Chip Erase requires a 10-ms PROG pulse.

ure 3. Programming the Flash

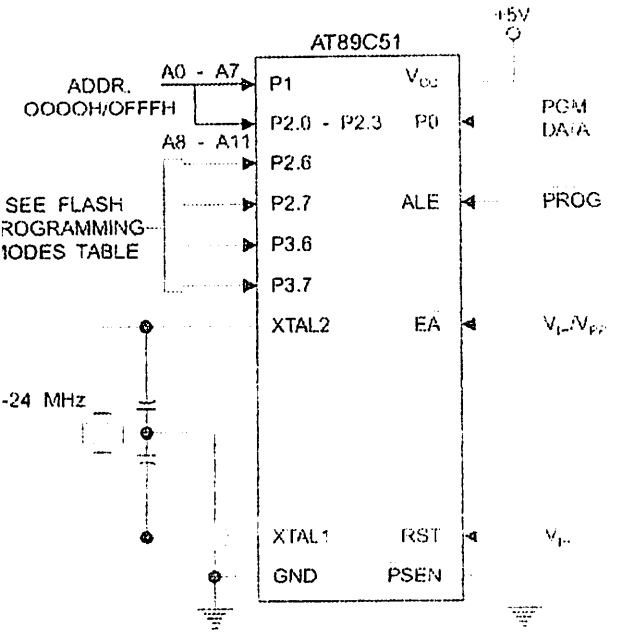
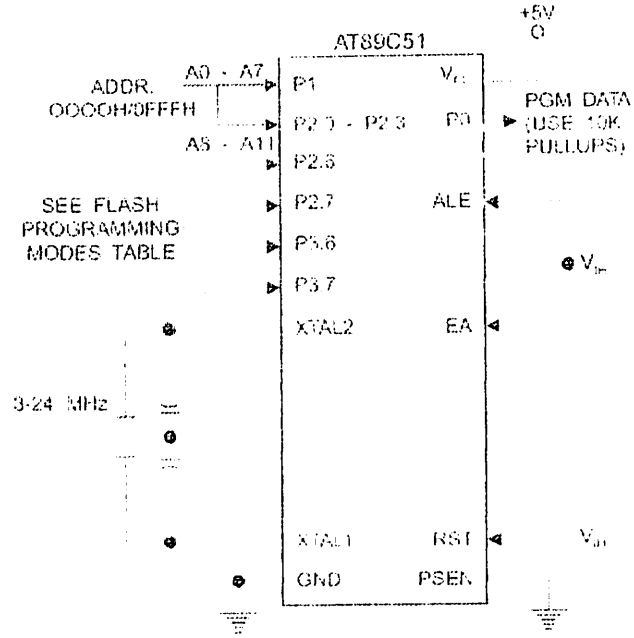


Figure 4. Verifying the Flash



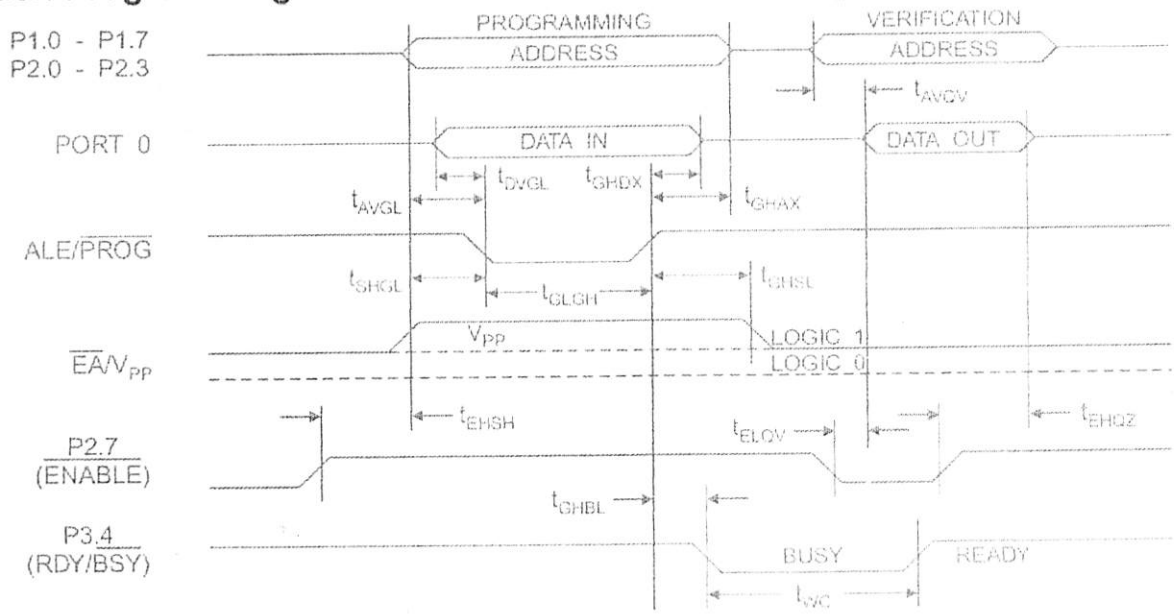
ash Programming and Verification Characteristics

= 0°C to 70°C, V_{CC} = 5.0 ± 10%

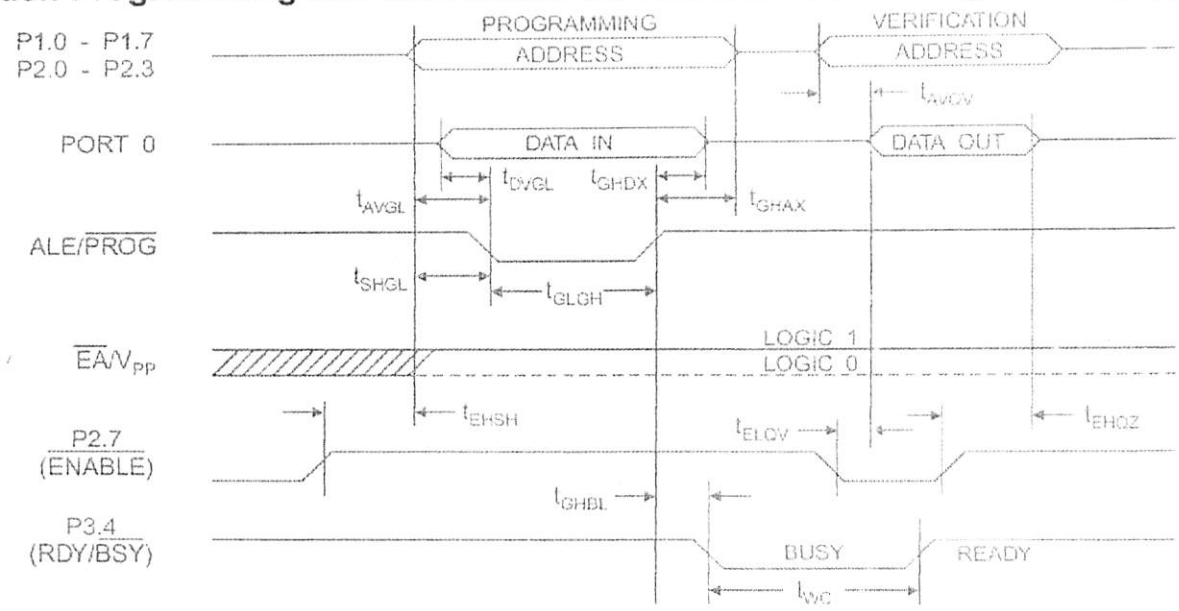
ymbol	Parameter	Min	Max	Units
$_{PP}$(1)	Programming Enable Voltage	11.5	12.5	V
$_{P}$(1)	Programming Enable Current		1.0	mA
$^{t}_{CLCL}$	Oscillator Frequency	3	24	MHz
VGL	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
HAX	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
VGL	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
HDX	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
HSH	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
HGL	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
HSL(1)	V_{PP} Hold After $\overline{PROG}$	10		µs
LGH	$\overline{PROG}$ Width	1	110	µs
VQV	Address to Data Valid		$48t_{CLCL}$	
LQV	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
HQZ	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
HBL	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
VC	Byte Write Cycle Time		2.0	ms

te: 1. Only used in 12-volt programming mode.

ash Programming and Verification Waveforms - High Voltage Mode (V_{PP} = 12V)



ash Programming and Verification Waveforms - Low Voltage Mode (V_{PP} = 5V)



bsolute Maximum Ratings*

)perating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
/oltage on Any Pin vith Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
)C Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

C Characteristics

$_A$ = -40°C to 85°C, V_{CC} = 5.0V ± 20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output Low Voltage(1) (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage(1) (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, VCC = 5V ± 10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
R_{RST}	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power Down Mode(2)	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	µA

otes:
1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port: Port 0: 26 mA
   Ports 1, 2, 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power Down is 2V.

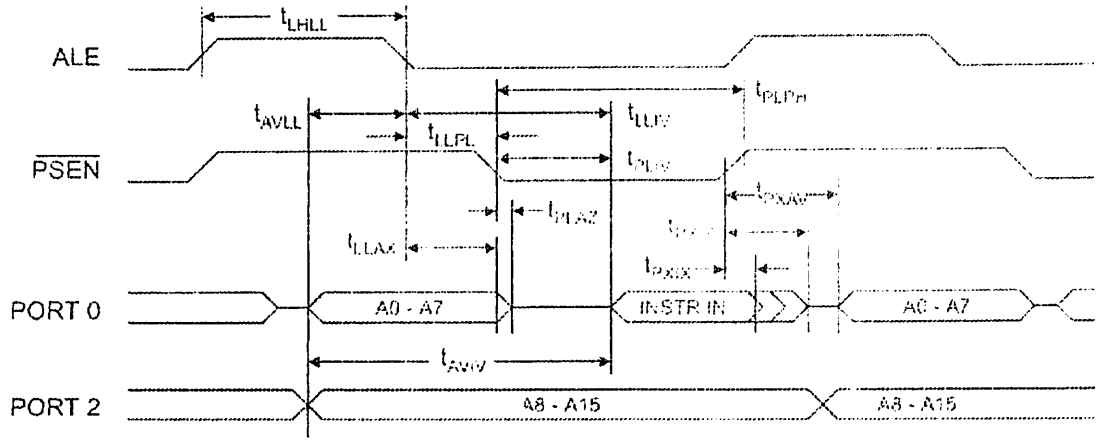
AC Characteristics

(Under Operating Conditions; Load Capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; Load Capacitance for all other outputs = 80 pF)

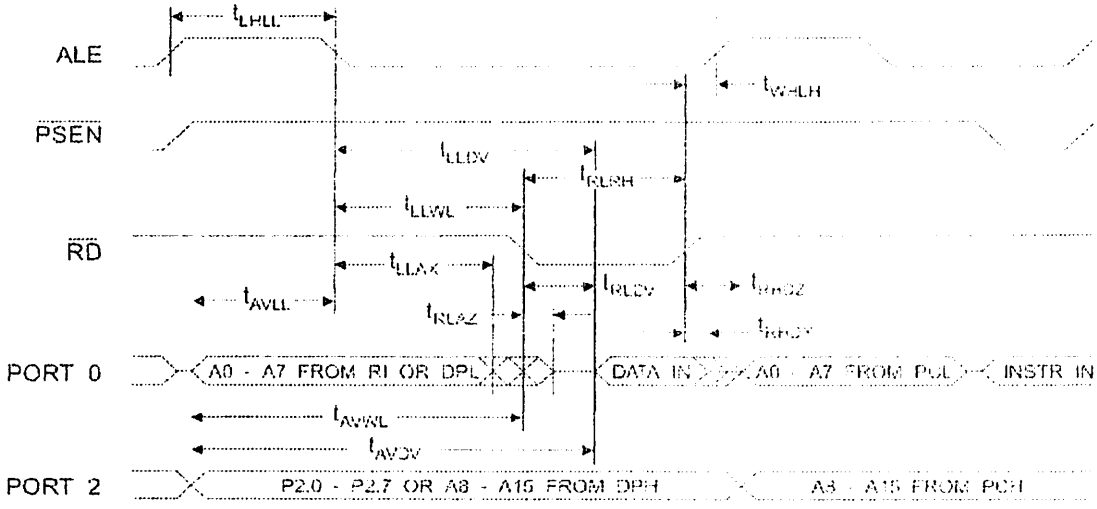
External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-13		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-20		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}$-65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		t_{CLCL}-13		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}$-20		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}$-45	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		t_{CLCL}-10	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}$-55	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}$-90	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}$-150	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}$-165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		t_{CLCL}-20		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}$-120		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		t_{CLCL}-20		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	t_{CLCL}-20	t_{CLCL}+25	ns

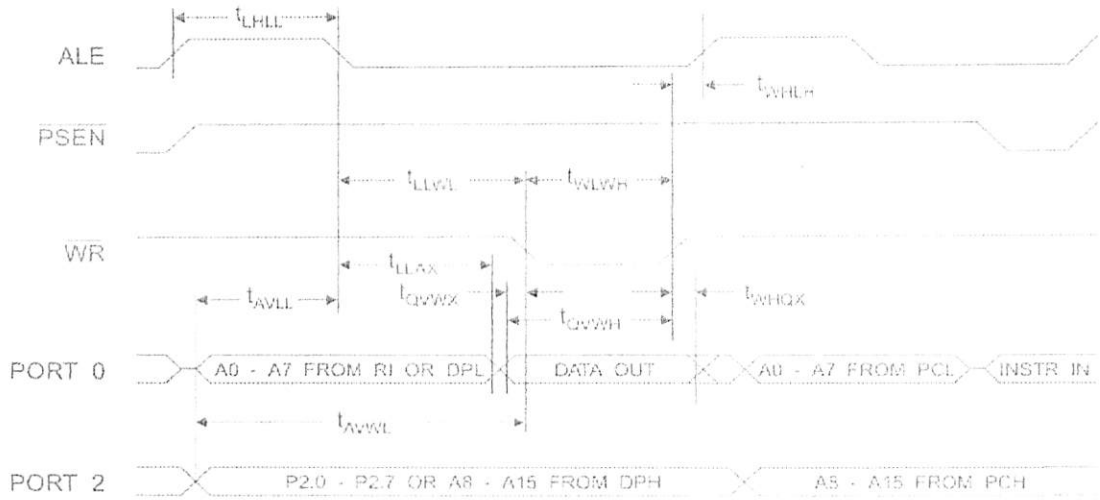
xternal Program Memory Read Cycle



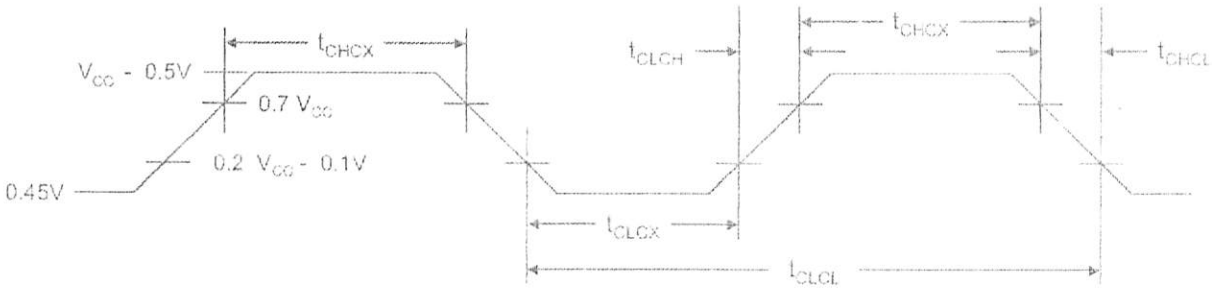
xternal Data Memory Read Cycle



xternal Data Memory Write Cycle



xternal Clock Drive Waveforms



xternal Clock Drive

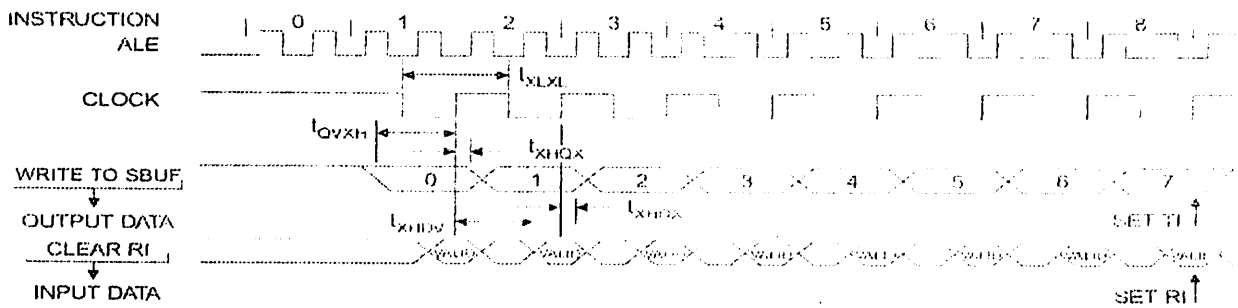
ymbol	Parameter	Min	Max	Units
/tCLCL	Oscillator Frequency	0	24	MHz
CLCL	Clock Period	41.6		ns
CHCX	High Time	15		ns
CLCX	Low Time	15		ns
CLCH	Rise Time		20	ns
CHCL	Fall Time		20	ns

ɜrial Port Timing: Shift Register Mode Test Conditions

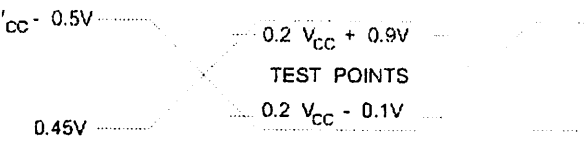
ɔc = 5.0 V ± 20%; Load Capacitance = 80 pF)

ymbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
LXL	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
VXH	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
HQX	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-117		ns
HDX	Input Data Hold After Clock Rising Edge	0		0		ns
HDV	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

ıift Register Mode Timing Waveforms

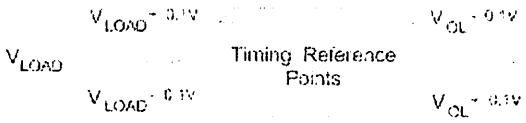


ɔ Testing Input/Output Waveforms(1)



ɜ: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms(1)



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

dering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ± 20%	AT89C51-12AC	44A	Commercial
		AT89C51-12JC	44J	(0°C to 70°C)
		AT89C51-12PC	40P6	
		AT89C51-12QC	44Q	
		AT89C51-12AI	44A	Industrial
		AT89C51-12JI	44J	(-40°C to 85°C)
		AT89C51-12PI	40P6	
		AT89C51-12QI	44Q	
		AT89C51-12AA	44A	Automotive
		AT89C51-12JA	44J	(-40°C to 105°C)
		AT89C51-12PA	40P6	
		AT89C51-12QA	44Q	
16	5V ± 20%	AT89C51-16AC	44A	Commercial
		AT89C51-16JC	44J	(0°C to 70°C)
		AT89C51-16PC	40P6	
		AT89C51-16QC	44Q	
		AT89C51-16AI	44A	Industrial
		AT89C51-16JI	44J	(-40°C to 85°C)
		AT89C51-16PI	40P6	
		AT89C51-16QI	44Q	
		AT89C51-16AA	44A	Automotive
		AT89C51-16JA	44J	(-40°C to 105°C)
		AT89C51-16PA	40P6	
		AT89C51-16QA	44Q	
20	5V ± 20%	AT89C51-20AC	44A	Commercial
		AT89C51-20JC	44J	(0°C to 70°C)
		AT89C51-20PC	40P6	
		AT89C51-20QC	44Q	
		AT89C51-20AI	44A	Industrial
		AT89C51-20JI	44J	(-40°C to 85°C)
		AT89C51-20PI	40P6	
		AT89C51-20QI	44Q	

dering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	5V ± 20%	AT89C51-24AC	44A	Commercial
		AT89C51-24JC	44J	(0°C to 70°C)
		AT89C51-24PC	44P6	
		AT89C51-24QC	44Q	
		AT89C51-24AI	44A	Industrial
		AT89C51-24JI	44J	(-40°C to 85°C)
		AT89C51-24PI	44P6	
		AT89C51-24QI	44Q	

Package Type	
A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)