

SKRIPSI

PERENCANAAN DAN PEMBUATAN PENGONTROL MODUL DIRECT DIGITAL SYNTHESIZER AD9850 SEBAGAI PEMBANGKIT FREKUENSI BERBASIS ARDUINO UNO



Disusun Oleh :

Wilfridus Maria Halek

12.12.210

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2016**

LEMBAR PERSETUJUAN

**PERENCANAAN DAN PEMBUATAN PENGONTROL
MODUL DIRECT DIGITAL SYNTHESIZER AD9850 SEBAGAI
PEMBANGKIT FREKUENSI BERBASIS ARDUINO UNO**

SKRIPSI

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

Disusun Oleh :

WILFRIDUS MARIA HALEK

NIM. 12.12.210

Diperiksa dan Disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Dr. F. Yudi Limpraptono, ST, MT
NIP.Y. 1039500274

Dr. Eng. I Komang Somawirara ST, MT
NIP.P. 1030100361

Mengetahui,

Ketua Program Studi Teknik Elektro S-1

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

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

2016

**PERENCANAAN DAN PEMBUATAN PENGONTROL MODUL DIRECT
DIGITAL SYNTHESIZER AD9850 SEBAGAI PEMBANGKIT FREKUENSI
BERBASIS ARDUINO UNO**

Wilfridus M. Halek, NIM 1212210

Dosen Pembimbing : Dr. F. Yudi Limpraptono, ST, MT dan
Dr. Eng. I Komang Somawirata, ST, MT

Jurusan Teknik Elektro S-1, Konsentrasi Teknik Elektronika
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km. 2 Malang
Email : fridushalek@yahoo.com

ABSTRAK

Direct Digital Synthesizer (DDS) menerapkan metode pembangkit gelombang analog secara digital dengan cara membangkitkan sinyal digital yang berubah-ubah terhadap waktu kemudian diubah kedalam bentuk analog menggunakan *digital to analog converter (DAC)*. IC AD9850 merupakan pembangkit gelombang analog yang menerapkan metode DDS, dimana frekuensi yang dibangkitkannya dapat diubah sesuai kebutuhan penggunaannya. Arduino Uno berfungsi memproses data dari keypad sebagai unit masukan (*input*) untuk selanjutnya mengirimkan data hasil perhitungan dan pemrosesan tersebut ke modul DDS AD9850. Perencanaan dan pembuatan alat ini bertujuan untuk pengontrol modul *Direct Digital Synthesizer* AD9850 sebagai pembangkit frekuensi menggunakan Arduino Uno dan frekuensi dapat diatur sesuai yang diinginkan dengan menggunakan Keypad Matriks dan LCD akan menampilkan nilai dari frekuensi yang dibangkitkan. Hasil pengujian pembangkit frekuensi ini dengan rentang frekuensi dari 1 Hz hingga 42 MHz berupa gelombang sinus dengan nilai selisih rata-rata sebesar 0,00057 %.

Kata Kunci : *Direct Digital Synthesizer (DDS)*, AD9850, Arduino Uno, Keypad Matriks

KATA PENGANTAR

Puji Syukur kehadiran Tuhan Yang Maha Kuasa atas berkat dan rahmat-Nya, sehingga kami selaku penyusun dapat menyelesaikan Laporan Skripsi ini yang berjudul **“ PERENCANAAN DAN PEMBUATAN PENGONTROL MODUL DIRECT DIGITAL SYNTHESIZER AD9850 SEBAGAI PEMBANGKIT FREKUENSI BERBASIS ARDUINO UNO ”** dapat terselesaikan.

Adapun maksud dan tujuan dari penulisan laporan ini merupakan salah satu syarat untuk dapat menyelesaikan studi dan mendapatkan gelar Sarjana Jurusan Teknik Elektro S-1, Konsentrasi Teknik Elektronika ITN Malang.

Sebagai pihak penyusun penulis menyadari tanpa adanya kemauan dan usaha serta bantuan dari berbagai pihak, maka laporan ini tidak dapat diselesaikan dengan baik. Oleh karena itu, penyusun mengucapkan terima kasih kepada yang terhormat :

1. Dr. Ir. Lalu Mulyadi, MT selaku Rektor Institut Teknologi Nasional Malang.
2. Ir. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. M. Ibrahim Ashari, ST, MT selaku Ketua Jurusan Teknik Elektro S-1 Institut Teknologi Nasional Malang.
4. Dr. F. Yudi Limpraptono, ST, MT selaku Dosen Pembimbing Satu Skripsi.
5. Dr. Eng. I Komang Somawirata, ST, MT selaku Dosen Pembimbing Dua Skripsi.
6. Sahabat-sahabat dan rekan-rekan yang tidak dapat disebutkan satu persatu, yang telah membantu baik dari segi teknis maupun dukungan moral dalam terselesaikannya skripsi ini.

Usaha telah kami lakukan semaksimal mungkin, namun jika ada kekurangan dan kesalahan dalam penyusunan, kami mohon saran dan kritik yang sifatnya membangun. Begitu juga sangat kami perlukan untuk menambah kesempurnaan laporan ini dan dapat bermanfaat bagi rekan-rekan mahasiswa pada khususnya dan pembaca pada umumnya.

Malang, Agustus 2016

Penyusun

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
ABSTRAK.....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	vi
DAFTAR TABEL.....	vii
BAB I PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Batasan Masalah.....	2
1.5 Metodologi Masalah.....	2
1.6 Sistematika Penulisan.....	3
BAB II DASAR TEORI.....	5
2.1 Arduino Uno.....	5
2.1.1 Board Arduino Uno memiliki fitur – fitur baru sebagai berikut.....	5
2.1.2 Arsitektur Arduino Uno R3.....	6
2.1.3 Arsitektur ATmega328.....	7
2.1.4 Deskripsi Arduino Uno.....	7
2.1.5 Software Arduino.....	10
2.2 Sistem DDS (Direct Digital Synthesizer).....	12
2.2.1 Kelebihan dan Flesibilitas DDS.....	12
2.3 Module Direct Digital Synthesizer AD9850.....	12
2.3.1 Spesifikasi Module DDS AD9850.....	14
2.3.2 Pengaturan Frekuensi Keluaran DDS.....	16
2.4 LCD 16x2 (Liquid Crystal Display).....	19
2.4.1 Fitur LCD 16x2.....	19
2.4.2 Fungsi pin-pin LCD.....	20
2.5 Keypad Matriks 4x4.....	21
2.5.1 Proses Scaning.....	22
BAB III PERENCANAAN DAN PEMBUATAN ALAT.....	24
3.1 Pendahuluan.....	24
3.2 Perancangan Sistem.....	24

3.2.1	Perencanaan Blok Diagram Sistem Secara keseluruhan	24
3.2.2	Prinsip Kerja	25
3.3	Perancangan Mekanik	26
3.4	Perancangan Perangkat Keras	27
3.4.1	Perancangan Rangkaian Arduino Uno	27
3.4.2	Perancangan Rangkaian LCD (<i>Liquid Crystal Display</i>) 16 x 2	28
3.4.3	Perancangan Rangkaian Keypad Matriks 4x4	29
3.4.4	Perancangan Rangkaian Modul AD9851	29
3.5	Perancangan Flowchart & Perangkat Lunak Sistem	30
3.5.1	Flowchart	31
3.5.2	Perancangan Perangkat Lunak (Software)	32
BAB IV PENGUJIAN DAN PEMBAHASAN SISTEM		33
4.1	Pendahuluan	33
4.2	Pengujian Rangkaian Mikrokontroler Arduino UNO	33
4.2.1	Peralatan yang digunakan	33
4.2.2	Langkah-langkah yang dilakukan	34
4.2.3	Hasil pengujian	34
4.3	Pengujian Rangkaian LCD 16 x 2	34
4.3.1	Peralatan yang digunakan	34
4.3.2	Langkah-langkah yang dilakukan	35
4.3.3	Hasil pengujian	35
4.4	Pengujian Koneksi Keypad 4 x 4	36
4.4.1	Hasil pengujian	36
4.5	Pengujian Modul DDS AD9850	37
4.5.1	Peralatan yang digunakan	37
4.5.1	Langkah-langkah yang dilakukan	37
4.5.2	Hasil pengujian	38
4.5	Analisa Pengujian Keseluruhan Sistem Pembangkit Frekuensi DDS AD9850	38
4.5.1	Analisa Hasil Pengujian	44
5.1	Kesimpulan	48
5.2	Saran	48
DAFTAR PUSTAKA		49

DAFTAR GAMBAR

Gambar 2.1 Fisik Board Arduino Uno	5
Gambar 2.2 Diagram Blok Atmega 328 pada Arduino Uno R3	6
Gambar 2.3 Konfigurasi Pin Pada IC Atmega 328.....	7
Gambar 2.4 Tampilan IDE Arduino	11
Gambar 2.5 Fisik Module Direct Digital Synthesizer AD9850	14
Gambar 2.6 Rangkaian DDS AD9850.....	15
Gambar 2.7 Blok diagram fungsi AD9850.....	15
Gambar 2.8 Runtunan data serial yang dimuat ke <i>data input register</i> DDS AD9850....	17
Gambar 2.9 Pin Konfiguration IC AD 9850.....	17
Gambar 2.10 Tampilan fisik LCD 16x2.....	20
Gambar 2.11 Pin-Pin modul LCD.....	20
Gambar 2.12 Konstruksi Matrix Keypad 4×4.....	22
Gambar 2.13 Keypad Matriks 4x4	23
Gambar 3.1 Diagram Blok Sistem.....	25
Gambar 3.2 Box bagian Atas.....	26
Gambar 3.3 Box bagian samping belakang	26
Gambar 3.4 Rangkaian Schematik Arduino Uno.....	27
Gambar 3.5 Rangkaian Koneksi LCD 16 x 2 Karakter ke Arduino.....	28
Gambar 3.6 Rangkaian Koneksi Keypad 4x4 ke Arduino	29
Gambar 3.7 Rangkaian Koneksi Modul AD9850 ke Arduino	29
Gambar 3.8 Diagram Alir Program Arduino	31
Gambar 3.9 Tampilan Awal <i>Software</i> IDE Arduino.....	32
Gambar 4.1 Skematik pengujian board Arduino	34
Gambar 4.2 Sketch pengujian arduino.....	34
Gambar 4.3 Rangkaian LCD	35
Gambar 4.4 Sketch program LCD.....	35
Gambar 4.5 Output hasil pengujian LCD.....	35
Gambar 4.6 Sketch Program arduino bagian keypad.....	36
Gambar 4.7 Hasil pengujian penekanan tombol pada keypad	37
Gambar 4.8 Pengujian pada modul AD9850.....	37
Gambar 4.9 Sketch Program Arduino pembangkit frekuensi	38
Gambar 4.10 (a) Frekuensi masukan sebesar 1 KHz (b) Frekuensi keluaran sebesar 1 KHz.....	38
Gambar 4.11 (a) Frekuensi masukan sebesar 1 Hz (b) Frekuensi keluaran sebesar 1 Hz.....	40

Gambar 4.12 (a) Frekuensi masukan sebesar 100 Hz (b) Frekuensi keluaran sebesar 100 Hz.....	40
Gambar 4.13 (a) Frekuensi masukan sebesar 1 KHz (b) Frekuensi keluaran sebesar 1 KHz.....	40
Gambar 4.14 (a) Frekuensi masukan sebesar 1 MHz (b) Frekuensi keluaran sebesar 1 MHz.....	41
Gambar 4.15 (a) Frekuensi masukan sebesar 40 MHz (b) Frekuensi keluaran sebesar 40 MHz.....	41
Gambar 4.16 (a) Frekuensi masukan 100 Hz (b) Bentuk gelombang keluaran AD9850.....	41
Gambar 4.17 (a) Frekuensi masukan 10 KHz (b) Bentuk gelombang keluaran AD9850.....	42
Gambar 4.18 (a) Frekuensi masukan 1 MHz (b) Bentuk gelombang keluaran AD9850.....	42
Gambar 4.19 (a) Frekuensi masukan 5 MHz (b) Bentuk gelombang keluaran AD9850.....	42

DAFTAR TABEL

Tabel 2.1 Deskripsi Arduino Uno	7
Tabel 2.2 Runtunan data untuk frekuensi 10 MHz.....	16
Tabel 2.3 Pin Function Descriptions	18
Tabel 4.1 Data hasil pengujian pembangkit frekuensi menggunakan Modul AD9850 dengan rentang frekuensi dari 1 Hz hingga 42 Mhz (Gelombang Sinus),	43

BAB I

PENDAHULUAN

1.1 Latar belakang

Teknologi terus berkembang dan selalu diupayakan untuk memiliki fungsi yang lengkap tetapi sederhana. Sehingga tidak mengherankan jika banyak alat yang sudah ada diperbaharui dengan alasan menyederhanakan komponen tetapi dapat mempertahankan atau bahkan dapat meningkatkan performa dari suatu alat. Seperti halnya pembangkit frekuensi, alat ini banyak digunakan pada sistem komunikasi antara lain pada *radio receiver*, sistem GPS, *handphone*, *radiophone*, *walkie-talkies*, radio komunikasi untuk CB, *satellite receiver*, *clock generator*, modulasi *FM*, dan masih banyak lagi^[1].

Fungsi pembangkit frekuensi adalah men-scan range frekuensi tertentu yang dapat menghasilkan sinyal analog. harganya yang mahal maka dengan kemampuan DDS untuk membangkitkan sinyal dapat dirancang sebuah pembangkit frekuensi yang memiliki spesifikasi yang tidak jauh berbeda dan hasil keluaran yang diharapkan tidak jauh berbeda dengan pembangkit frekuensi yang ada dari sisi bentuk sinyal keluaran.

Direct Digital Synthesizer (DDS) menerapkan metode pembangkit gelombang analog secara digital dengan cara membangkitkan sinyal digital yang berubah-ubah terhadap waktu kemudian diubah kedalam bentuk analog menggunakan *digital to analog converter* (DAC)^[2]. IC AD9850 merupakan pembangkit gelombang analog yang menerapkan metode DDS, dimana frekuensi yang dibangkitkannya dapat diubah sesuai kebutuhan penggunaannya. DDS (*Direct Digital synthesizer*) merupakan rangkaian *sinthesis oscillator* yang lebih canggih dari PLL (*phase locked loop*). IC DDS banyak jenisnya, tetapi yang paling populer adalah AD9850 yang mampu membangkitkan frekuensi sampai 40 MHz dan kelebihan DDS yaitu memiliki step 1 Hertz, hingga 10 Mhz yang dapat diatur.

Dalam tugas akhir ini, dirancang suatu alat pengontrol modul *direct digital synthesizer* AD9850 sebagai pembangkit frekuensi menggunakan Arduino dan frekuensi akan di tampilkan di LCD. Frekuensi dapat diatur sesuai yang diinginkan dengan menggunakan Keypad Matriks, sehingga DDS AD9850 dapat difungsikan sebagai sebuah pembangkit frekuensi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diinginkan diatas, maka dapat disimpulkan permasalahan yang dituangkan dalam karya ilmiah ini, yaitu :

1. Bagaimana cara kerja *Direct Digital Synthesizer* (DDS) AD9850 ?
2. Bagaimana cara mengontrol *Direct Digital Synthesizer* (DDS) AD9850 menggunakan Arduino Uno ?
3. Sebarapa besar akurasi frekuensi yang bisa di hasilkan oleh modul AD9850 ?

1.3 Tujuan

Perencanaan dan pembuatan alat ini bertujuan untuk pengontrol modul *Direct Digital Synthesizer* AD9850 sebagai pembangkit frekuensi menggunakan Arduino dan frekuensi dapat diatur sesuai yang diinginkan dengan menggunakan Keypad Matriks dan LCD akan menampilkan nilai dari frekuensi yang di bangkitkan.

1.4 Batasan Masalah

Agar perencanaan dan pembuatan alat ini dapat sesuai dengan tujuan yang diharapkan dan tetap fokus pada konsep awal, maka diperlukan beberapa batasan-batasan diantaranya adalah :

1. Arduino Uno digunakan untuk memproses data dan pengontrolan perangkat.
2. Modul AD9850 digunakan sebagai pembangkit frekuensi.
3. Keypad Matriks digunakan untuk memasukan nilai frekuensi yang ingin dibangkitkan.
4. LCD hanya menampilkan nilai frekuensi yang dibangkitkan.
5. Frekuensi yang mampu dibangkitkan sebesar 1 Hz - 42 Mhz.
6. Output yang dikeluarkan hanya berupa gelombang sinus.

1.5 Metodologi Masalah

Agar dalam perancangan skripsi ini dapat berjalan sesuai dengan tujuan dan konsep yang telah direncanakan serta dapat selesai pada waktunya maka metode penelitian yang digunakan adalah :

1. **Kajian Literatur**
Metode untuk pengumpulan data dan informasi yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan pada perancangan sistem.
2. **Perancangan Alat**
Proses untuk melakukan design dan penyusunan konsep mekanik yang akan digunakan agar sesuai dengan perancangan sistem secara keseluruhan.
3. **Pembuatan Alat**
Pada tahap ini realisasi alat yang dibuat, dilakukan perakitan sistem terhadap seluruh hasil rancangan yang telah dibuat.
4. **Perancangan Flowchart**
Proses untuk menentukan alur kerja dari sistem yang akan dibuat sehingga nantinya proses selanjutnya dapat sesuai dengan sistem yang dirancang.
5. **Pembuatan perangkat lunak**
Pembuatan program sesuai dengan flowchart yang telah dibuat.
6. **Pengujian Sistem dan perbaikan**
Proses ujicoba sistem yang telah dibuat untuk mengetahui adanya kesalahan dan melakukan perbaikan terhadap kesalahan agar sesuai dengan perancangan sistem.
7. **Pelaporan dan Kesimpulan**
Proses untuk penyusunan laporan terhadap sistem dan penarikan kesimpulan pada sistem yang telah diuji.

1.6 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, sistematika penulisan disusun sebagai berikut:

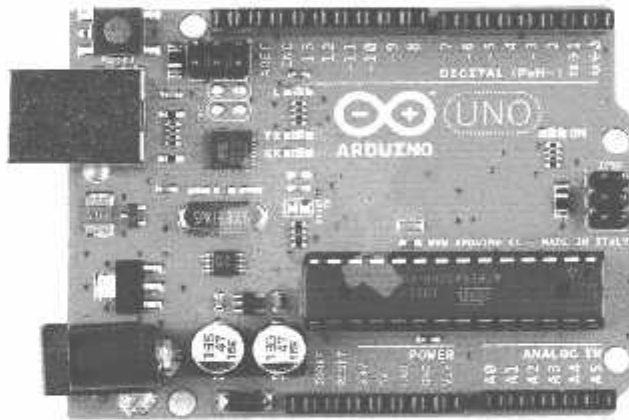
- BAB I : PENDAHULUAN**
Berisi tentang latar belakang rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika penulisan.
- BAB II : KAJIAN PUSTAKA**
Membahas tentang dasar teori mengenai permasalahan yang berhubungan dengan penelitian.
- BAB III : PERENCANAAN DAN PEMBUATAN ALAT**
Bab ini membahas tentang perencanaan dan proses pembuatan meliputi perencanaan, pembuatan alat, cara kerja, dan penggunaan alat.
- BAB IV : PENGUJIAN DAN ANALISA**
Berisi tentang pembahasan dan analisa alat dari hasil yang diperoleh pada pengujian.
- BAB V : PENUTUP**
Berisi tentang semua kesimpulan yang berhubungan dengan penulisan skripsi, dan saran yang digunakan sebagai pertimbangan dalam pengembangan program selanjutnya.
-

BAB II

DASAR TEORI

2.1 Arduino Uno

Arduino Uno adalah board mikrokontroler berbasis ATmega 328. Board ini memiliki 14 digital input/output pin (dimana 6 pin dapat digunakan sebagai output PWM), 6 input analog, 16 MHz osilator kristal, koneksi USB, jack power, ICSP header, dan tombol reset. Pin-pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya. Arduino Uno R3 berbeda dengan semua board sebelumnya karena Arduino Uno R3 ini tidak menggunakan chip driver FTDI USB-to-serial. Melainkan menggunakan fitur dari ATmega 16U2 yang diprogram sebagai converter USB-to-serial [3].



Gambar 2.1 Fisik Board Arduino Uno

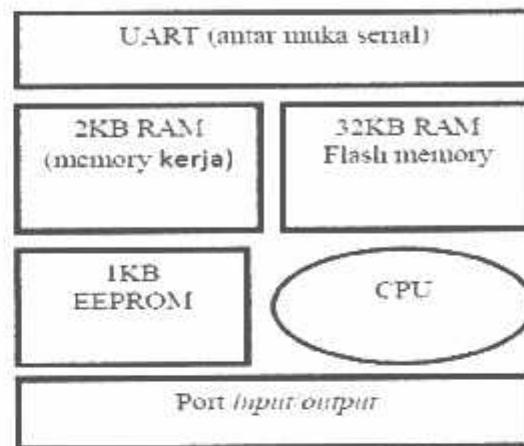
2.1.1 Board Arduino Uno memiliki fitur – fitur baru sebagai berikut :

- Pinout : menambahkan SDA dan SCL pin yang dekat ke pin aref dan dua pin baru lainnya ditempatkan dekat ke pin RESET, dengan I/O REF yang memungkinkan sebagai buffer untuk beradaptasi dengan tegangan yang disediakan dari board sistem. Pengembangannya, sistem akan lebih kompatibel dengan prosesor yang menggunakan AVR, yang beroperasi dengan 5V dan dengan Arduino karena beroperasi dengan 3,3V. Yang kedua adalah pin yang tidak terhubung, yang disediakan untuk tujuan pengembangannya.
- Sirkuit reset.
- ATmega 16U2 ganti 8U yang digunakan sebagai konverter USB-to-serial.

2.1.2 Arsitektur Arduino Uno R3

Arduino uno R3 adalah board arduino revisi terbaru yang merupakan penerus dari arduino uno R3. Arduino uno R3 merupakan board mikrokontroler berdasarkan atmega 328.

Blok diagram dari arduino uno R3, Blok diagram sederhana dari arduino uno dapat dilihat pada gambar 2.2.



Gambar 2.2 Diagram Blok Atmega 328 pada Arduino Uno R3

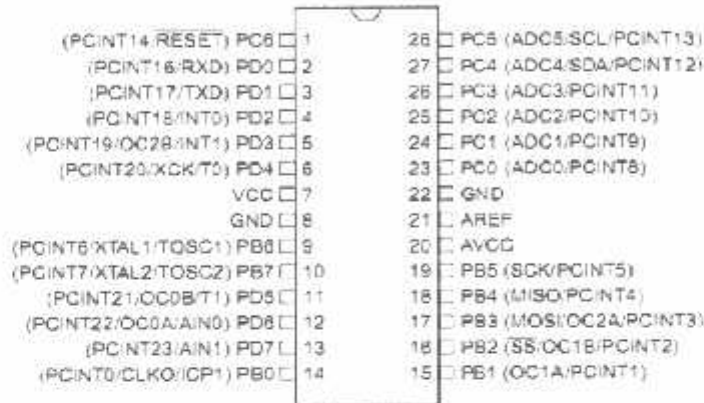
Penjelasan bagian-bagian pada blok diagram sederhana dari arduino uno R3:

1. *Universal Asynchronous Receiver/ Transmitter* (UART) adalah antar muka yang digunakan untuk komunikasi serial seperti RS 232, RS-422, dan RS-485.
2. 2KB RAM pada memori bekerja bersifat volatile (hilang saat daya dimatikan), digunakan oleh variable-variabel di dalam program.
3. 32KB RAM flash memory bersifat non-volatile, digunakan untuk menyimpan program yang dimuat dari komputer. Selain program, flash memori juga menyimpan bootloader. Pengertian bootloader adalah sebuah program inisiasi yang ukurannya kecil, dijalankan oleh CPU saat daya dihidupkan. Setelah bootloader selesai dijalankan, berikutnya program didalam RAM akan dieksekusi.
4. 1KB EEPROM bersifat non-volatile, digunakan untuk menyimpan data yang tidak boleh hilang saat daya dimatikan.
5. *Central Processing Unit* (CPU), bagian dari mikrokontroler untuk menjalankan setiap instruksi dari program.

Port Input/Output, pin-pin untuk menerima data (input) digital atau analog.

2.1.3 Arsitektur ATmega328

Sistem Pada board arduino uno R3 mikrokontroler yang dipakai adalah mikrokontroler Atmega 328. Mikrokontroler atmega 328 adalah buatan Atmel berbasis arsitektur *Reduced Instruction Set Computer (RISC)*, hampir semua instruksi dieksekusi dalam satu siklus clock. Gambar 2.3 di bawah ini merupakan konfigurasi pin dari Atmega 328.



Gambar 2.3 Konfigurasi Pin Pada IC Atmega 328

2.1.4 Deskripsi Arduino Uno

Tabel 2.1 Deskripsi Arduino Uno

Mikrokontroler	ATmega 328
Tegangan Pengoperasian	5 V
Tegangan Input yang disarankan	7 – 12 V
Batas Tegangan Input	6 – 20 V
Jumlah pin I/O digital	14 pin digital (6 diantaranya menyediakan keluaran PWM)
Jumlah pin input Analog	6 pin
Arus DC tiap pin I/O	40mA
Arus DC untuk pin 3,3 V	50mA
Memori Flash	32 KB(ATmega 328) sekitar 0,5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega 328)
EEPROM	1 KB (ATmega 328)
Clock Speed	16 MHz

a. Power

Arduino dapat diberikan power melalui koneksi USB atau power supply. Powernya diseleksi secara otomatis. Power supply dapat menggunakan adaptor DC atau baterai. Adaptor dapat dikoneksikan dengan mencolok jack adaptor pada koneksi port input supply. Board arduino dapat dioperasikan menggunakan supply dari luar sebesar 6 - 20 volt. Jika supply kurang dari 7V, kadangkala pin 5V akan menyuplai kurang dari 5 volt dan board bisa menjadi tidak stabil. Jika menggunakan lebih dari 12 V, tegangan di regulator bisa menjadi sangat panas dan menyebabkan kerusakan pada board. Rekomendasi tegangan ada pada 7 sampai 12 volt.

Penjelasan pada pin power adalah sebagai berikut :

- **Vin**

Tegangan input ke board arduino ketika menggunakan tegangan dari luar (Seperti yang disebut 5 volt dari koneksi USB atau tegangan yang diregulasikan). Pengguna dapat memberikan tegangan melalui pin ini, atau jika tegangan suplai menggunakan power jack, aksesnya menggunakan pin ini.

- **5V**

Regulasi power supply digunakan untuk power mikrokontroler dan komponen lainnya pada board. 5V dapat melalui Vin menggunakan regulator pada board, atau supply oleh USB atau supply regulasi 5V lainnya.

- **3V3**

Suplai 3.3 volt didapat oleh FTDI chip yang ada di board. Arus maksimumnya adalah 50mA.

- **Pin Ground**

berfungsi sebagai jalur ground pada arduino.

- **Memori**

ATmega328 memiliki 32 KB flash memori untuk menyimpan kode, juga 2 KB yang digunakan untuk bootloader. ATmega328 memiliki 2 KB untuk SRAM dan 1 KB untuk EEPROM.

b. Input dan Output

Setiap 14 pin digital pada arduino dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Input/output dioperasikan pada 5 volt. Setiap pin dapat menghasilkan atau menerima maximum 40 mA dan memiliki internal pull-up resistor (disconnected oleh default) 20-50K Ohm.

Beberapa pin memiliki fungsi sebagai berikut :

- Serial : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirim (TX) TTL data serial. Pin ini terhubung pada pin yang koresponding dari USB ke TTL chip serial.
- Interrupt eksternal : 2 dan 3. Pin ini dapat dikonfigurasi untuk trigger sebuah interap pada low value, rising atau falling edge, atau perubahan nilai.
- PWM : 3, 5, 6, 9, 10, dan 11. Mendukung 8-bit output PWM dengan fungsi `analogWrite()`.
- SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mensupport komunikasi SPI, yang mana masih mendukung hardware, yang tidak termasuk pada bahasa arduino.
- LED : 13. Ini adalah dibuat untuk koneksi LED ke digital pin 13. Ketika pin bernilai HIGH, LED hidup, ketika pin LOW, LED mati.

c. Komunikasi

Uno Arduino memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). Firmware Arduino menggunakan USB driver standar COM, dan tidak ada driver eksternal yang dibutuhkan. Namun, pada Windows, file. Ini diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke board Arduino. RX dan TX LED di board akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB ke komputer.

2.1.5 Software Arduino

Arduino Uno dapat diprogram dengan perangkat lunak Arduino . Pada ATmega328 di Arduino terdapat *bootloader* yang memungkinkan Anda untuk meng-*upload* kode baru untuk itu tanpa menggunakan *programmer hardware eksternal*.

IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan Java. IDE Arduino terdiri dari:

1. Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa Processing.
2. Compiler, sebuah modul yang mengubah kode program (bahasa Processing) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa Processing. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya compiler diperlukan dalam hal ini.
3. Uploader, sebuah modul yang memuat kode biner dari komputer ke dalam memory didalam papan Arduino.

Sebuah kode program Arduino umumnya disebut dengan istilah *sketch*. Kata "*sketch*" digunakan secara bergantian dengan "kode program" dimana keduanya memiliki arti yang sama.

Dalam bahasa programan arduino ada tiga bagian utama yaitu struktur, variable dan fungsi: (Artanto, 2012).

1. Struktur Program Arduino

a) Karangka Program

Karangka program arduino sangat sederhana, yaitu terdiri atas dua blok. Blok pertama adalah *void setup()* dan blok kedua adalah *void loop()*.

1) Blok *void setup ()*

Berisi kode program yang hanya dijalankan sekali sesaat setelah arduino dihidupkan atau di-reset, Merupakan bagian persiapan atau instalasi program.

2) Blok *void loop ()*

Berisi kode program yang akan dijalankan terus-menerus. Merupakan tempat untuk program utama.

b) Sintaks Program

Baik blok void setup loop () maupun blok function harus diberi tanda kurung kurawal buka “{“ sebagai awal program di blok itu dan kurung kurawal tutup ”}” sebagai tanda akhir program.

2. Variabel

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas dengan menggunakan sebuah variable.

3. Fungsi

Pada bagian ini meliputi fungsi input output digital, input output analog, advanced I/O, fungsi waktu, fungsi matematika serta fungsi komunikasi.

Pada proses Uploader dimana proses ini merubah bahasa pemrograman yang nantinya decompile oleh avr-gcc (avr-gcc compiler) yang hasilnya akan disimpan kedalam papan arduino.

Avr-gcc compiler merupakan sebuah bagian penting untuk software bersifat open source. Dengan adanya avr-gcc compiler, maka akan membuat bahasa pemrograman dapat dimengerti oleh mikrokontroler. Proses terakhir ini sangat penting, karena dengan adanya proses ini maka akan membuat proses pemrograman mikrokontroler menjadi sangat mudah.



Gambar 2.4 Tampilan IDE Arduino

2.2 Sistem DDS (*Direct Digital Synthesizer*)

Sistem DDS (*Direct Digital Synthesizer*) merupakan salah satu cara untuk menghasilkan sinyal sinusoida secara langsung. Inti dari sistem ini adalah arsitektur akumulator dengan resolusi mencapai mili Hertz dan frekuensi sinyal yang di hasilkan dapat diatur tergantung dari sinyal frekuensi referensi dan metode perancangan. Keluaran sistem DDS yang diproses oleh mikrokontroler berupa sinyal digital kemudian menjadi masukan untuk DAC (*D/A converter*) dan LPF (*Low Pass Filter*) untuk menghasilkan sinyal sinusoida yang sempurna.

Semua parameter control sistem DDS berada dalam bentuk digital. Sistem DDS pada dasarnya terdiri atas akumulator fasa, LUT (*Look Up Table*), dan osilator sebagai pembangkit frekuensi referensi (*clock*). Sedangkan DAC (*Digital To Analog Converter*) dan LPF (*Low Pass Filter*) merupakan komponen-komponen penunjang sistem DDS.

2.2.1 Kelebihan dan Flesibilitas DDS

Kelebihan penggunaan Sistem DDS adalah karakteristik sistem DDS itu sendiri, dimana keutamaan dari sistem ini adalah memiliki setting time/kecepatan yang cepat dan memiliki resolusi frekuensi yang halus terhadap frekuensi keluaran, operasi atas suatu sepektrum frekuensi yang lebar dan dengan kemajuan dalam desain teknologi proses. Serta sangat ringkas dan sedikit membutuhkan daya. Sehingga sangat memungkinkan sistem DDS bisa lebih dikembangkan untuk desain alat yang berkaitan dengan aplikasi-aplikasi frekuensi hopping serta sistem-sistem yang berkaitan dengan peralatan pemancar radio, TV, peralatan test, dll.

2.3 Module *Direct Digital Synthesizer* AD9850

Module *Direct Digital Synthesizer* (DDS) AD9850 adalah modul pembangkit gelombang yang menggunakan IC AD9850 CMOS 125 MHz. IC ini menggunakan teknologi *Digital-to-Analog Converter* terkini yang berkinerja tinggi untuk mensintesa frekuensi (*frequency synthesizer*). Dengan osilator kristal aktif berfrekuensi 125 MHz sebagai sumber detak berpresisi sangat tinggi, modul ini membangkitkan gelombang sinus analog spektral murni dengan frekuensi/fasa yang dapat diprogram. Gelombang keluaran

ini dapat digunakan langsung sebagai sumber frekuensi, atau dapat juga dikonversi menjadi gelombang sinus maupun gelombang persegi.

Suatu accumulator/register dengan penambahan data terjadi pada setiap clock dari suatu oscillator, yang menghitung mulai dari 0 sampai dengan seluruh bit dari accumulator I, adalah waktu terbesar atau dengan kata lain frekuensi terkecil yang dapat dihasilkan oleh adanya accumulator tersebut atau dapat juga kita sebut frekuensi resolusi. Lalu bagaimana kita dapat merubah frekuensi dari output DDS. Pengisian data pada accumulator berarti kita memperpendek waktu accumulator penuh (kondisi bit semua 1) atau kita memperbesar frekuensi output, pada data IC kita kenal dengan Δ Phase. Waktu accumulator mencapai penuh merupakan satu perioda, kemudian pembentukan signal digunakan ADC (analog to digital converter)

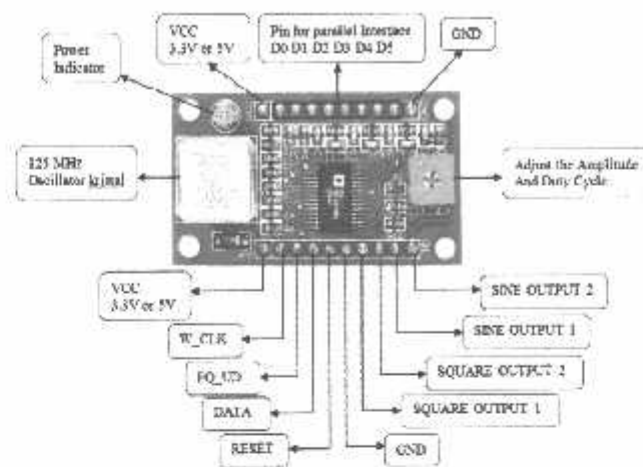
AD9850 yang mampu membangkitkan frekuensi sampai 40 MHz, dengan resolusi frekuensi 0,0291 Hz. Kelebihan DDS yaitu memiliki step 1 Hertz, hingga 10 Mhz yang dapat diatur. DDS AD9850 digunakan untuk membangkitkan frekuensi secara digital yang berupa gelombang sinus dan gelombang kotak. Untuk membangkitkan frekuensi sesuai dengan nilai frekuensi yang diinginkan perlu diatur nilai *Frequency Tunning Word* (FTW) yang dirumuskan sesuai Persamaan (2-1) ^[5].

$$F_{out} = \frac{Freq \cdot 2^N}{Ref\ CLK} \dots\dots\dots(2-1)$$

Keterangan :

- Freq : frekuensi keluran dari DDS
- Fout : frekuensi tuning word
- Ref CLK : referensi internal frekuensi clock
- N : jumlah bit akumulator fasa

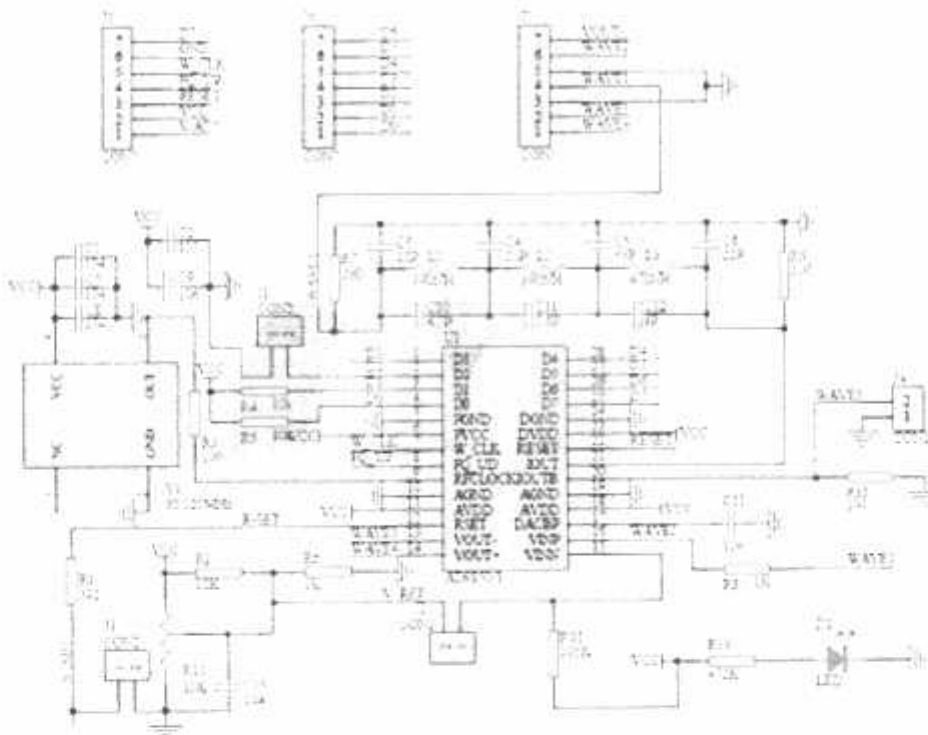
Dengan Freq adalah nilai frekuensi keluaran dari DDS. Fout adalah nilai *frequency tuning word* dari frekuensi yang akan dibangkitkan. Ref CLK adalah referensi internal frekuensi clock (125 Mhz), dan 2^N adalah nilai pembagi dimana N adalah jumlah *bit accumulator fasa* (pada AD9850 jumlah *bit accumulator fasa* yang digunakan adalah 32), dan Rangkaian pembangkit frekuensi seperti pada Gambar 2.6 di bawah ini.



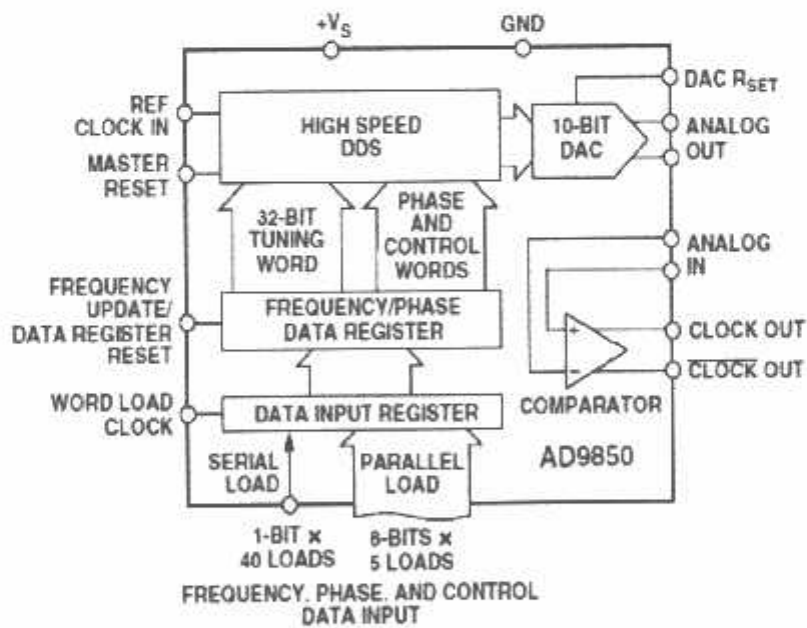
Gambar 2.5 Fisik Module Direct Digital Synthesizer AD9850

2.3.1 Spesifikasi Module DDS AD9850

- Membangkitkan 2 gelombang sinus (*sine wave*) dan 2 gelombang persegi (*square wave*).
- Signal frekuensi output range 0 – 40 MHz.
- Resolusi frekuensi 0,0291 Hz.
- Presisi hingga frekuensi harmonik 20 MHz, max 40 MHz (di atas 20 MHz, gelombang akan mengalami distorsi).
- Frekuensi maksimum gelombang persegi: 1 MHz.
- Data masukan dapat dipilih antara paralel dan serial dengan memindahkan *jumper*.
- Magnitudo gelombang keluaran dapat disetel dengan mudah melalui benchmark pin (pin#12).
- Tegangan referensi untuk pembanding (*comparator reference input voltage*) dapat diatur dengan menggunakan potensiometer untuk menentukan siklus kerja (*duty cycle*) pada keluaran gelombang persegi / digital.
- Menggunakan osilator aktif frekuensi tinggi (125 Mhz) dengan kualitas presisi yang sangat baik (*high precision active crystal oscillator*).



Gambar 2.6 Rangkaian DDS AD9850



Gambar 2.7 Blok diagram fungsi AD9850

Pada Gambar 2.7 terlihat blok diagram fungsi dari DDS AD9850, pengaturan frekuensi dan fase keluaran dapat dilakukan dengan memasukkan data ke *data input*

register secara serial atau parallel. Apabila memasukkan data secara serial maka data yang perlu dimuat adalah 1 bit x 40 sedangkan bila memasukkan data secara parallel 8 bit x 5. Ref Clock In menerima masukan dari osilator kristal.

2.3.2 Pengaturan Frekuensi Keluaran DDS^[6]

Pengaturan frekuensi keluaran DDS AD9850 misalnya jika ingin membangkitkan frekuensi 10 MHz dilakukan dengan mengirimkan data secara serial, data *frequency tuning word* yang harus dikirimkan dihitung menggunakan persamaan(2-1) yaitu :

$$F_{out} = \frac{Freq \cdot 2^N}{RefCLK}$$

$$F_{out} = \frac{10 \cdot 2^{32}}{125}$$

$F_{out} = 343597383_D \rightarrow$ MSB 00010100 01111010 11100001 01000111 LSB $_B$

Tabel 2.2 Runtunan data untuk frekuensi 10 MHz

Serial Load Word																																										
3	3	3	3	3	3	3	3	3	3	3	2	2	5	5	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	1	0	1	0	1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	1
↓				↓				P		c		c		↓																						↓						
M				L				w		o		o		M																						L						
S				S				r		t		t		S																						S						
B				B				D		r		r		B																						B						
Fase				Fase				w		o		o		Frequency tuning word																						Frequency tuning word						
n				n				l		l		B																						B								

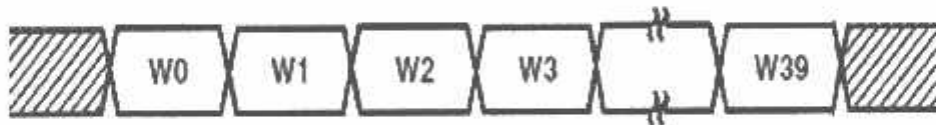
- W32 : control kondisi harus 0
- W33 : control kondisi harus 0
- W34 : 1 Power down ON, tidak ada keluaran sinyal
: 0 Power down OFF, sinyal keluaran dalam kondisi normal
- W35 – W39 : Untuk pengaturan fase dengan penambahan setiap bit 11.25°
- W34 : 1 Power down ON, tidak ada keluaran sinyal

: 0 *Power down OFF*, sinyal keluaran dalam kondisi normal

W35 – W39 : Untuk pengaturan fase dengan kenaikan 11.25°

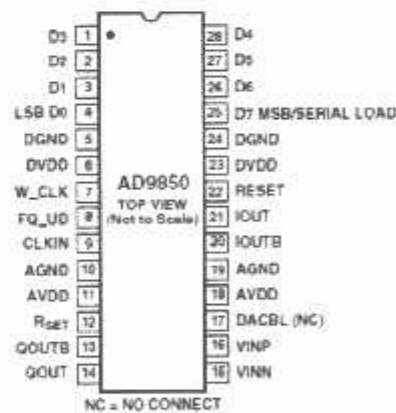
Data dalam bentuk runtunan bit inilah yang harus dimuat dalam *data input register* secara serial untuk menentukan frekuensi keluaran DDS dengan panjang 32 bit, dengan urutan mulai *LSB* dan terakhir *MSB* atau dari kiri ke kanan atau mulai *W0* dan berakhir dengan *W31*, seperti tertampil pada tabel 2.2. Sisa 8 bit yaitu bit ke 32 sampai bit 39 digunakan untuk mengatur pilihan frekuensi referensi AD9850, *power-down* dan fase. Pengaturan fase dimulai dari *LSB* dan berakhir pada bit *LSB*. Posisi runtunan bit yang dimuat ke *data input register* untuk frekuensi 10 MHz seperti ditampilkan pada Tabel 2.2.

Pengaturan frekuensi dan fase keluaran dapat dilakukan dengan memasukkan data ke *data input register* secara serial atau parallel. Apabila memasukkan data secara serial maka data yang perlu dimuat adalah 1 bit x 40 sedangkan bila memasukkan data secara parallel 8 bit x 5. *Ref Clock In* menerima masukan dari osilator kristal. Urutan data yang harus dimuat ke *data input register* seperti tertampil pada Gambar 2.8.



Gambar 2.8

Runtunan data serial yang dimuat ke *data input register* DDS AD9850



Gambar 2.9 Pin Konfigurasi IC AD 9850

Tabel 2.3 Pin Function Descriptions

Pin No.	Mnemonic	Function
4 to 1, 28 to 25	D0 to D7	8-Bit Data Input. This is the 8-bit data port for iteratively loading the 32-bit frequency and the 8-bit phase/ control word. D7 = MSB; D0 = LSB. D7 (Pin 25) also serves as the input pin for the 40-bit serial data-word.
5, 24	DGND	Digital Ground. These are the ground return leads for the digital circuitry.
6, 23	DVDD	Supply Voltage Leads for Digital Circuitry.
7	W_CLK	Word Load Clock. This clock is used to load the parallel or serial frequency/phase/control words.
8	FQ_UD	Frequency Update. On the rising edge of this clock, the DDS updates to the frequency (or phase) loaded in the data input register; it then resets the pointer to Word 0.
9	CLKIN	Reference Clock Input. This may be a continuous CMOS-level pulse train or sine input biased at 1/2 V supply. The rising edge of this clock initiates operation.
10, 19	AGND	Analog Ground. These leads are the ground return for the analog circuitry (DAC and comparator).
11, 18	AVDD	Supply Voltage for the Analog Circuitry (DAC and Comparator).
12	RSET	DAC's External RSET Connection. This resistor value sets the DAC full-scale output current. For normal applications ($I_{S IOUT} = 10 \text{ mA}$), the value for RSET is $3.9 \text{ k}\Omega$ connected to ground. The RSET/IOUT relationship is $I_{OUT} = 32 (1.248 \text{ V}/RSET)$.
13	QOUTB	Output Complement. This is the comparator's complement output.
14	QOUT	Output True. This is the comparator's true output.

15	VINN	Inverting Voltage Input. This is the comparator's negative input.
16	VINP	Noninverting Voltage Input. This is the comparator's positive input.
17	DACBL (NC)	DAC Baseline. This is the DAC baseline voltage reference; this lead is internally bypassed and should normally be considered a no connect for optimum performance.
20	IOUTB	Complementary Analog Output of the DAC.
21	IOUT	Analog Current Output of the DAC.
22	RESET	Reset. This is the master reset function; when set high, it clears all registers (except the input register), and the DAC output goes to cosine 0 after additional clock cycles—see Figure 7.

2.4 LCD 16x2 (*Liquid Crystal Display*)^[7]

LCD (*Liquid Crystal Display*) adalah sesuatu jenis media tampil yang menggunakan Kristal cair sebagai penampil utama. LCD sudah digunakan di berbagai bidang misalnya alat-alat elektronik seperti televisi, kalkulator, atau pun layar computer. Pada postingan aplikasi LCD yang digunakan ialah LCD dot matrik dengan jumlah karakter 2x16. LCD sangat berfungsi sebagai penampil yang nantinya akan digunakan untuk menampilkan status kerja alat.

2.4.1 Fitur LCD 16x2

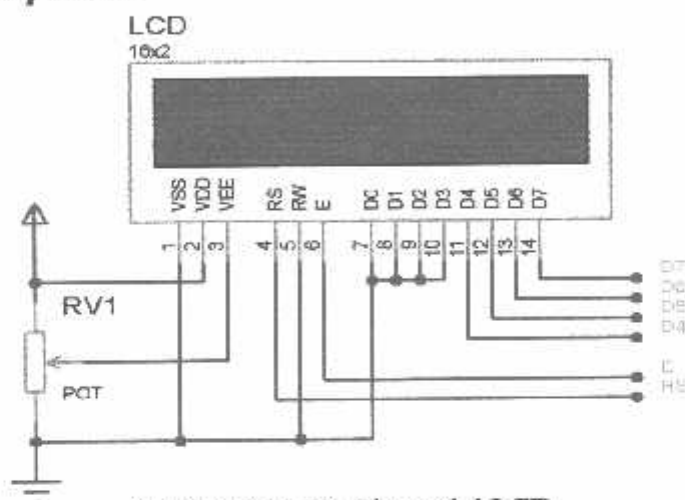
Adapun fitur yang disajikan dalam LCD ini adalah :

- a) Terdiri dari 16 karakter dan 2 baris.
- b) Mempunyai 192 karakter tersimpan.
- c) Terdapat karakter generator terprogram.
- d) Dapat dialamat dengan mode 4-bit dan 8-bit
- e) Dilengkapi dengan back light.



Gambar 2.10 Tampilan fisik LCD 16x2

2.4.2 Fungsi pin-pin LCD



Gambar 2.11 Pin-Pin modul LCD

a. Pin 1 dan 2

Merupakan sambungan catu daya, Vss dan Vdd. Pin Vdd dihubungkan dengan tegangan positif catu daya, dan Vss pada 0V atau ground. Meskipun data menentukan catu 5 Vdc (hanya pada beberapa mA), menyediakan 6V dan 4.5V yang keduanya bekerja dengan baik, bahkan 3V cukup untuk beberapa modul.

b. Pin 3

Pin 3 merupakan pin kontrol Vee, yang digunakan untuk mengatur kontras display. Idealnya pin ini dihubungkan dengan tegangan yang bisa dirubah untuk memungkinkan pengaturan terhadap tingkatan kontras display sesuai dengan kebutuhan, pin ini dapat dihubungkan dengan variable resistor sebagai pengatur kontras.

c. Pin 4

Pin 4 merupakan *Register Select* (RS), masukan yang pertama dari tiga command control input. Dengan membuat RS menjadi high, data karakter dapat ditransfer dari dan menuju modulnya.

d. Pin 5

Read/Write (R/W), untuk memfungsikan sebagai perintah write maka R/W low atau menulis karakter ke modul. R/W high untuk membaca data karakter atau informasi status dari register-nya.

e. Pin 6

Enable (E), input ini digunakan untuk transfer actual dari perintah-perintah atau karakter antara modul dengan hubungan data. Ketika menulis ke display, data ditransfer hanya pada perpindahan *high* atau *low*. Tetapi ketika membaca dari display, data akan menjadi lebih cepat tersedia setelah perpindahan dari low ke high dan tetap tersedia hingga sinyal low lagi.

f. Pin 7 sampai 14

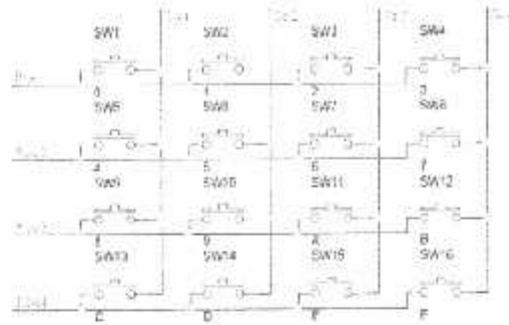
Pin 7 sampai 14 adalah delapan jalur data (D0-D7) dimana data dapat ditransfer ke dan dari display.

g. Pin 15 dan 16

Pin 15 atau A (+) mempunyai level DC +5V berfungsi sebagai LED back light + sedangkan pin 16 yaitu K (-) memiliki level 0 V dan berfungsi sebagai LED back light.

2.5 Keypad Matriks 4x4^[8]

Keypad adalah bagian penting dari suatu perangkat elektronika yang membutuhkan interaksi manusia. Keypad berfungsi sebagai interface antara perangkat (mesin) elektronik dengan manusia atau dikenal dengan istilah HMI (Human Machine Interface). Matrix keypad 4x4 pada artikel ini merupakan salah satu contoh keypad yang dapat digunakan untuk berkomunikasi antara manusia dengan mikrokontroler. Matrix keypad 4x4 memiliki konstruksi atau susunan yang simple dan hemat dalam penggunaan port mikrokontroler. Konfigurasi keypad dengan susunan bentuk matrix ini bertujuan untuk penghematan port mikrokontroler karena jumlah key (tombol) yang dibutuhkan banyak pada suatu sistem dengan mikrokontroler. Konstruksi matrix keypad 4x4 untuk mikrokontroler dapat dibuat seperti pada gambar berikut.



Gambar 2.12 Konstruksi Matrix Keypad 4x4

Konstruksi matrix keypad 4x4 diatas cukup sederhana, yaitu terdiri dari 4 baris dan 4 kolom dengan keypad berupa saklar push button yang diletakan disetiap persilangan kolom dan barisnya. Rangkaian matrix keypad diatas terdiri dari 16 saklar push button dengan konfigurasi 4 baris dan 4 kolom. 8 line yang terdiri dari 4 baris dan 4 kolom tersebut dihubungkan dengan port mikrokontroler 8 bit. Sisi baris dari matrix keypad ditandai dengan nama Row1, Row2, Row3 dan Row4 kemudian sisi kolom ditandai dengan nama Col1, Col2, Col3 dan Col4. Sisi input atau output dari matrix keypad 4x4 ini tidak mengikat, dapat dikonfigurasi kolom sebagai input dan baris sebagai output atau sebaliknya tergantung programernya.

2.5.1 Proses Scaning

Proses scanning untuk membaca penekanan tombol pada matrix keypad 4x4 untuk mikrokontroler diatas dilakukan secara bertahap kolom demi kolom dari kolom pertama sampai kolom ke 4 dan baris pertama hingga baris ke 4. Program untuk scanning matrix keypad 4x4 dapat bermacam-macam, tapi pada intinya sama. Misal kita asumsikan keypad aktif LOW (semua line kolom dan baris dipasang resistor pull-up) dan dihubungkan ke port mikrokontroler dengan jalur kolom adalah jalur input dan jalur baris adalah jalur output maka proses scanning matrix keypad 4x4 diatas dapat dituliskan sebagai berikut.

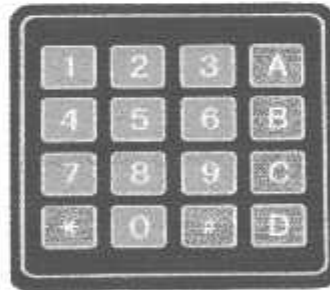
Mengirimkan logika Low untuk kolom 1 (Col1) dan logika HIGH untuk kolom yang lain kemudian membaca data baris, misal tombol SW1 ditekan maka data baris pertama (Row1) akan LOW sehingga data baris yang dibaca adalah 0111, atau tombol yang ditekan tombol SW5 maka data pada baris ke 2 akan LOW sehingga data yang terbaca 1011, atau tombol SW9 yang ditekan sehingga data yang terbaca 1101, atau tombol SW13

yang ditekan maka data yang dibaca adalah 1110 dan atau tidak ada tombol pada kolom pertama yang di tekan maka data pembacaan baris akan 1111.

Kemudian data pembacaan baris ini diolah sebagai pembacaan data penekanan tombol keypad. Sehingga tiap tombol pada matrix keypad 4x4 diatas dengan teknik scanning tersebut akan menghasilkan data penekanan tiap-tiap tombol sebagai berikut.

SW1 = 0111 0111	SW9 = 0111 1101
SW2 = 1011 0111	SW10 = 1011 1101
SW3 = 1101 0111	SW11 = 1101 1101
SW4 = 1110 0111	SW12 = 1110 1101
SW5 = 0111 1011	SW13 = 0111 1110
SW6 = 1011 1011	SW14 = 1011 1110
SW7 = 1101 1011	SW15 = 1101 1110
SW8 = 1110 1011	SW16 = 1110 1110

Data port mikrokontroler, misalkan pada SW2 = 1011 0111 tersebut terbagi dalam nible atas dan nible bawah dimana data nible atas (1011) merupakan data yang kita kirimkan sedangkan data nible bawah (0111) adalah data hasil pembacaan penekanan tombol keypad SW2 pada proses scanning matrix keypad 4x4 diatas.



Gambar 2.13 Keypad Matriks 4x4

BAB III PERENCANAAN DAN PEMBUATAN ALAT

3.1 Pendahuluan

Pada bab ini akan membahas mengenai perencanaan sistem, prinsip kerja dan perancangan perangkat keras (*hardware*) serta perangkat lunak (*software*) yang berkaitan dengan pembangkit frekuensi yaitu mengenai design sistem pengontrol modul DDS AD9850 sebagai pembangkit frekuensi yang diimplementasikan pada Arduino Uno. Pada perancangan ini akan diimplementasikan konsep dan teori dasar yang telah dibahas sebelumnya, sehingga tujuan dari perencanaan dapat tercapai dengan baik. Untuk itu pembahasan difokuskan pada desain yang direncanakan pada diagram blok sistem.

3.2 Perancangan Sistem

Alat yang akan dirancang akan dibagi menjadi dua bagian utama yaitu perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*) pembangkit frekuensi. Perancangan sistem perangkat keras meliputi bagian Input, kontroler, dan output. Pada bagian input terdiri dari rangkaian *Keypad* 4x4 dengan Arduino Uno untuk memasukan nilai frekuensi yang ingin di bangkitkan.

Bagian kontroler terdiri dari Arduino Uno mikrokontroler ATmega 328. Yang berfungsi untuk memproses data dari unit masukan dari keypad (*input*) untuk selanjutnya mengirimkan data hasil perhitungan dan pemrosesan tersebut kepada unit pembangkit frekuensi AD9851 dan menampilkan nilai frekuensi di LCD.

3.2.1 Perencanaan Blok Diagram Sistem Secara keseluruhan

Fungsi Masing-masing Blok Diagram Sistem

1. Keypad Matriks 4x4
Berfungsi untuk memasukan (*input*) nilai frekuensi yang ingin dibangkitkan.
2. Lcd 2x16
Berfungsi sebagai penampil nilai frekuensi agar informasi dari perangkat dapat dengan mudah dilihat dan dibaca.
3. Modul AD9850
Berfungsi untuk membangkitkan frekuensi secara digital yang berupa gelombang

sinus.

4. Arduino Uno

Berfungsi memproses data dari keypad sebagai masukan (*input*) nilai frekuensi yang ingin dibangkitkan, untuk selanjutnya mengirimkan data hasil perhitungan dan pemrosesan tersebut kepada pembangkit frekuensi AD9850.

5. Output

Berupa gelombang sinus, bentuk gelombang dapat di lihat menggunakan Osiloskop.



Gambar 3.1 Diagram Blok Sistem

3.2.2 Prinsip Kerja

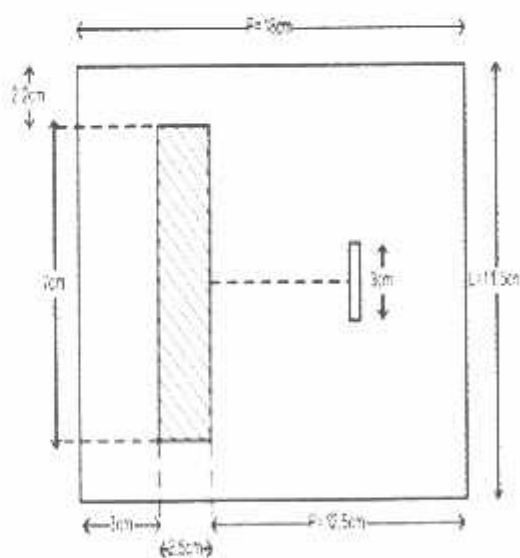
Pada perencanaan dan pembuatan alat pengontrol modul DDS AD9850 sebagai pembangkit frekuensi berbasis Arduino Uno. Module *Direct Digital Synthesizer* (DDS) AD9850 adalah modul pembangkit gelombang yang menggunakan IC AD9850 CMOS 125 MHz. AD9850 yang mampu membangkitkan frekuensi sampai 40 MHz, kelebihan DDS yaitu memiliki step 1 Hertz, hingga 10 Mhz yang dapat diatur.

Direct Digital Synthesizer (DDS) menerapkan metode pembangkit gelombang analog secara digital dengan cara membangkitkan sinyal digital yang berubah-ubah terhadap waktu kemudian diubah kedalam bentuk analog menggunakan *digital to analog converter* (DAC). IC AD9850 merupakan pembangkit gelombang analog yang menerapkan metode DDS, dimana frekuensi yang dibangkitkannya dapat diubah sesuai kebutuhan penggunaanya.

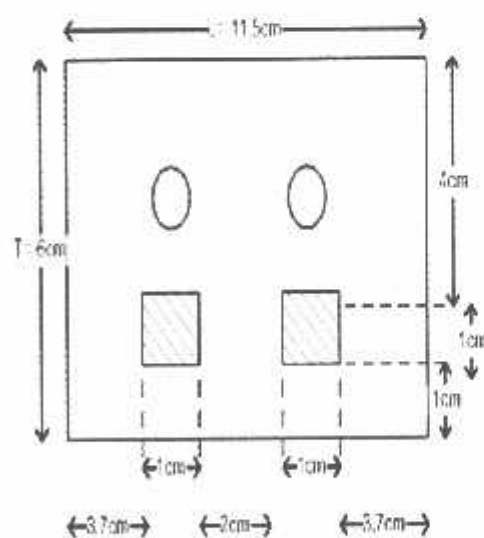
Keypad digunakan sebagai masukan nilai frekuensi yang ingin di bangkitkan. Arduino Uno berfungsi memproses data dari keypad sebagai unit masukan (*input*) untuk selanjutnya mengirimkan data hasil perhitungan dan pemrosesan tersebut ke modul DDS AD9850, pengiriman data secara serial dengan cara mengirimkan ke-40 bit data biner dari nilai FTW ke dalam AD9850, *rising edge* dari DDS_CLK akan menggeser dan memberikan nilai 1 bit pada DDS_SER dan akan langsung mengisi satu nilai di dalam *register* FTW AD9850. Pengiriman data FTW dilakukan dengan mengisi *register* W0 sampai *register* W39. Mode pengisian *register* tersebut dimulai dengan mengisi LSB dan berakhir pada MSB^[5], dan selanjutnya AD9850 akan membangkitkan frekuensi secara digital yang berupa gelombang sinus, sesuai dengan data yang di kirim oleh arduino uno, frekuensi yang dibangkitkan akan ditampilkan pada LCD.

3.3 Perancangan Mekanik

Perancangan mekanik menggunakan box kotak hitam yang sudah jadi, yang mempunyai ukuran sebagai berikut P=18cm, L=11,5cm, T=6cm. Selanjutnya box akan di beri lubang sebagai tempat untuk LCD, USB, Power dan output. Lubang pada box seperti gambar di bawah ini. Yang meliputi gambar bagian atas dan bagian samping belakang, yang samping lainnya tidak di lubangkan.



Gambar 3.2 Box bagian Atas



Gambar 3.3 Box bagian samping belakang

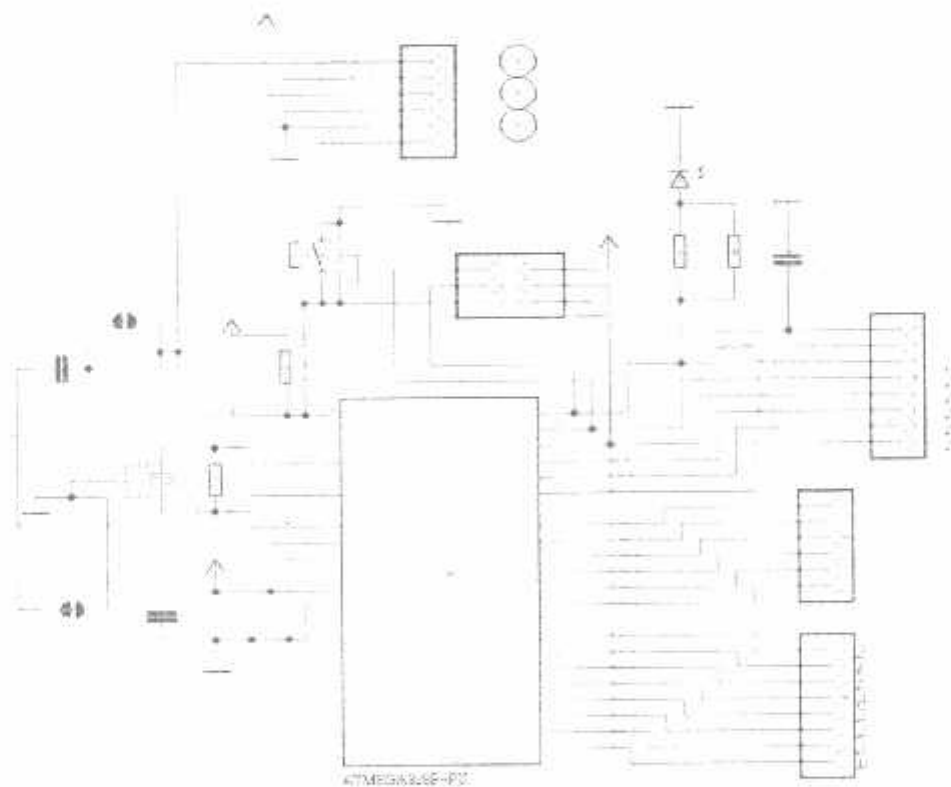
3.4 Perancangan Perangkat Keras

3.4.1 Perancangan Rangkaian Arduino Uno

Rangkaian Kontrol disini menggunakan Arduino Uno yang akan mengolah data masukan dan keluaran pada setiap pinnya. Arduino yang digunakan dalam perancangan ini merupakan komponen utama, karena komponen ini adalah yang akan mengatur keseluruhan system agar dapat bekerja dengan baik dan optimal. Perencanaan rangkaian arduino uno dapat dilihat pada gambar 3.4.

Arduino uno akan memproses masukan dan keluaran yang ada pada peralatan ini, pengontrolan tersebut dilakukan melalui pengaktifan masing-masing pin pada arduino. Untuk mengaktifkan pin-pin yang terdapat didalam arduino tersebut dilakukan dengan menggunakan perangkat lunak (Software) yang dimiliki oleh arduino.

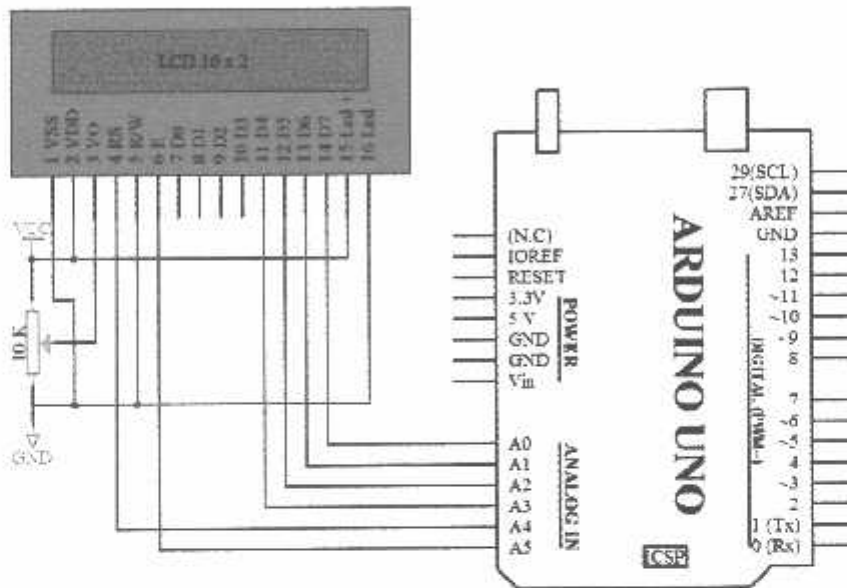
Untuk mengaktifkan arduino uno maka perlu diberikan tegangan supply +5 Volt DC yang dapat langsung diambil dari port USB pada computer, atau dapat juga diambil dari supply tegangan DC lain.



Gambar 3.4 Rangkaian Schematik Arduino Uno

3.4.2 Perancangan Rangkaian LCD (*Liquid Crystal Display*) 16 x 2

Pada perancangan sistem ini digunakan LCD 16 x 2 yang merupakan sebuah modul LCD *dot matrix* yang membutuhkan daya kecil. Rangkaian koneksi LCD karakter 16 x 2 ke Arduino ditunjukkan dalam Gambar 3.5. Dalam perancangan ini, pin-pin yang harus dihubungkan dengan pin Arduino antara lain :



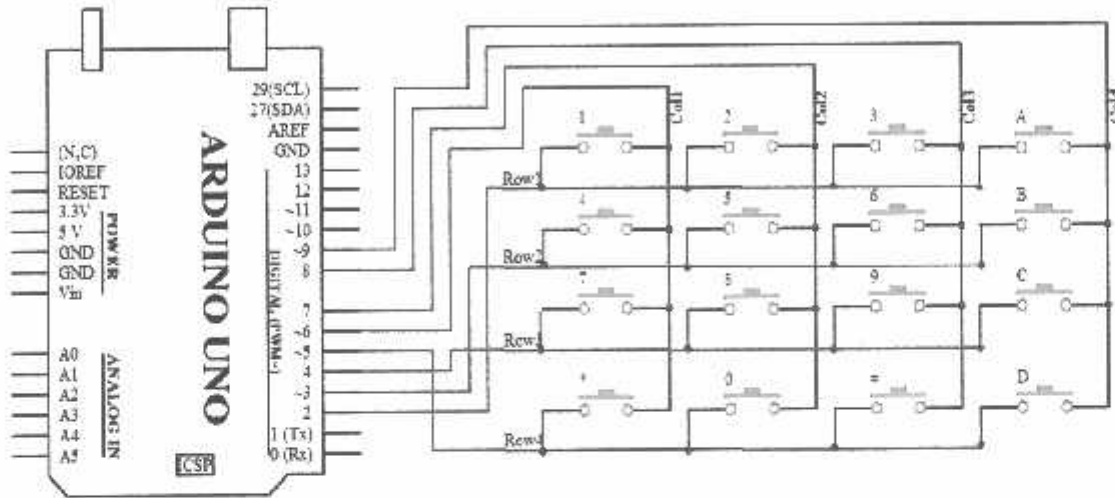
Gambar 3.5. Rangkaian Koneksi LCD 16 x 2 Karakter ke Arduino

- Pin 14 DB7 pada Lcd dihubungkan pada pin A0 Arduino.
- Pin 13 DB6 pada Lcd dihubungkan pada pin A1 Arduino.
- Pin 12 DB5 pada Lcd dihubungkan pada pin A2 Arduino.
- Pin 11 DB4 pada Lcd dihubungkan pada pin A3 Arduino.
- Pin 6 E pada Lcd dihubungkan pada pin A5 Arduino.
- Pin 4 RS pada Lcd dihubungkan pada pin A4 Arduino
- Pin Vcc pada Lcd dihubungkan pada VCC Arduino, 5 V.
- Pin GND pada Modul AD9851 dihubungkan pada pin GND Arduino.

Dalam perancangan ini agar didapat kontras dan *backlight* dari LCD yang sesuai, maka dipasangkan sebuah resistor variabel dengan nilai 10 k Ω pada PIN *contrast*.

3.4.3 Perancangan Rangkaian Keypad Matriks 4x4

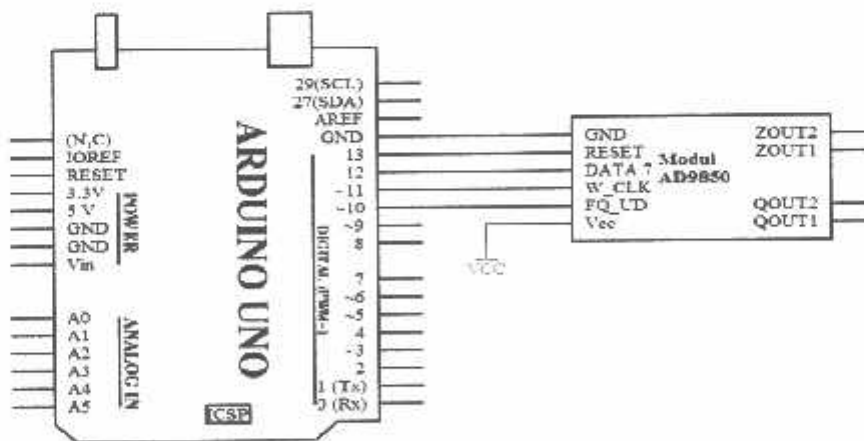
Pada perancangan sistem ini, digunakan keypad dengan dimensi 4 x 4. Port untuk baris dihubungkan pada pin 2, 3, 4, 5 Arduino. Sedangkan untuk port kolom dihubungkan pada pin analog Arduino, yaitu pin 6, 7, 8, dan 9, seperti yang terlihat pada Gambar 3.6.



Gambar 3.6 Rangkaian Koneksi Keypad 4x4 ke Arduino

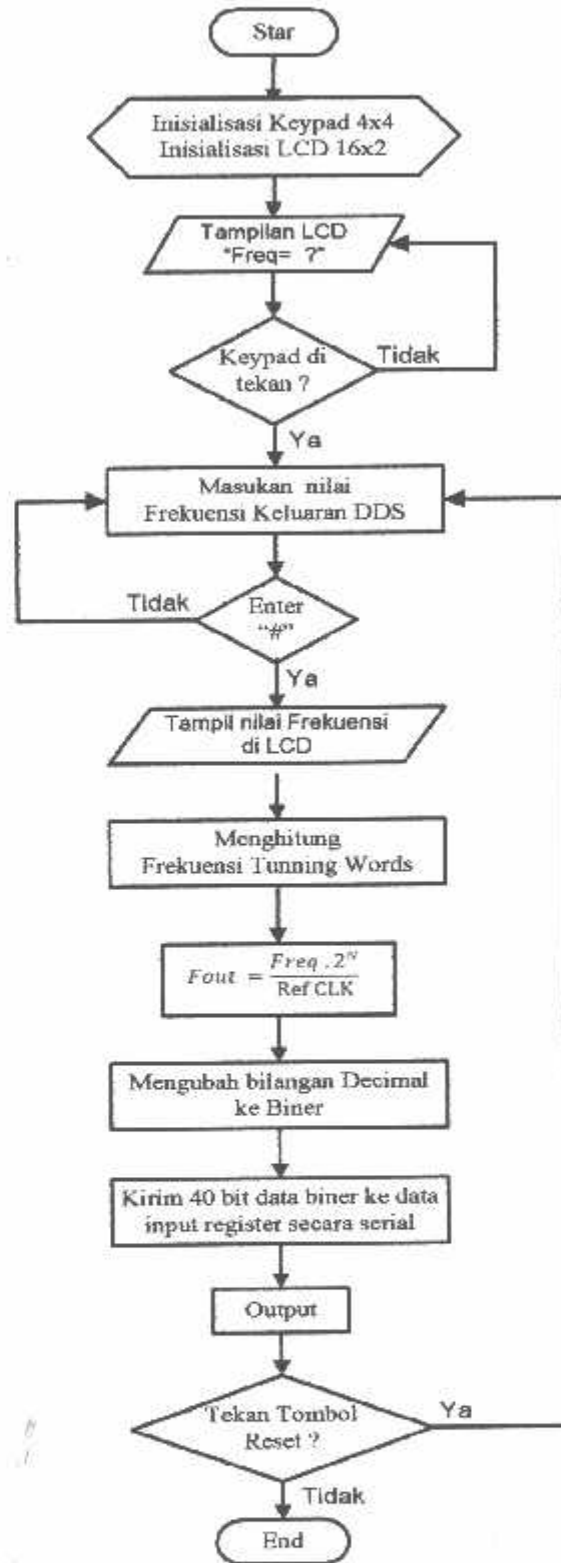
3.4.4 Perancangan Rangkaian Modul AD9850

Rangkaian koneksi antara Arduino dengan Modul AD9850 ditunjukkan pada Gambar 3.7. Dalam perancangan ini, pin-pin yang harus dihubungkan dengan pin Arduino antara lain :



Gambar 3.7 Rangkaian Koneksi Modul AD9850 ke Arduino

3.5.1 Flowchart



Gambar 3.8 Diagram Alir Program Arduino

3.5.2 Perancangan Perangkat Lunak (Software)

Perangkat lunak yang dirancang, dibuat dengan menggunakan bahasa *open source* arduino uno. Algoritma program utama dari perancangan perangkat lunak dapat dilihat dalam diagram alur, yang mana dengan kata lain disebut sebagai diagram yang menyajikan prosedur untuk menjalankan secara berurutan sesuai dengan yang kita harapkan. Pada bagian ini akan diatur mulai dari pendefinisian input-output (inisialisasi pin) hingga mengolah data agar sesuai dengan yang diinginkan.



Gambar 3.9

Tampilan Awal *Software* IDE Arduino

BAB IV PENGUJIAN DAN PEMBAHASAN SISTEM

4.1 Pendahuluan

Pada bab ini ditujukan untuk melakukan pengujian dan pembahasan dari sistem yang telah dirancang sebelumnya agar dapat diketahui bagaimana kinerja dari keseluruhan sistem maupun kinerja masing-masing bagian. Dari hasil pengujian tersebut akan dijadikan dasar untuk menentukan kesimpulan serta point-point kekurangan yang harus segera diperbaiki agar kinerja keseluruhan sistem dapat sesuai dengan perencanaan dan perancangan yang telah dibuat.

Pengujian dan pembahasan sistem ini meliputi beberapa hal antara lain :

- ✓ Pengujian pada Board Arduino uno.
- ✓ Pengujian pada LCD 16 x 2.
- ✓ Pengujian pada keypad 4 x 4.
- ✓ Pengujian Keseluruhan Sistem Pembangkit Frekuensi

4.2 Pengujian Rangkaian Mikrokontroler Arduino UNO

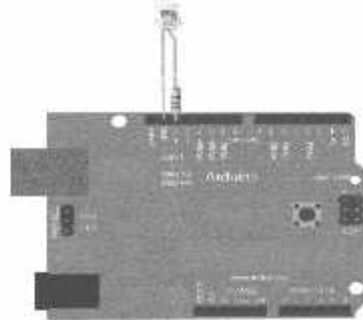
Untuk pengujian modul mikrokontroler Arduino UNO berkerja dengan baik akan dilakukan pengujian pada jalur-jalur *port* yang dimiliki oleh mikrokontroler Arduino UNO. Untuk pengujian modul dilakukan pengisian program terlebih dahulu menggunakan Arduino IDE (Integrated Development Environment). Dengan mengupload program ke mikrokontroler kita dapat mengetahui adanya *error* atau tidak. Untuk menjalankan program, caranya hubungkan langsung antara komputer dengan modul mikrokontroler Arduino UNO. Lalu lihat pada Arduino IDE (Integrated Development Environment) apakah program berhasil terupload. Bila berhasil, berarti modul Arduino UNO dapat digunakan.

4.2.1 Peralatan yang digunakan

1. PC (Personal Komputer)
2. Modul Arduino UNO
3. Software IDE (Integrated Development Environment)
4. LED 5 mm
5. Resistor 220 Ω

4.2.2 Langkah-langkah yang dilakukan

1. Merangkai LED pada board arduino, seperti pada gambar 4.1



Gambar 4.1 Skematik pengujian board Arduino

2. Memberikan program pada board arduino sesuai dengan sketch di bawah.

```

sketch_junibag
int led = 13;
void setup()
{
  pinMode(led, OUTPUT);
  digitalWrite(led, LOW);
}
void loop()
{
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}

```

Gambar 4.2 Sketch pengujian arduino

4.2.3 Hasil pengujian

Dengan mengupload listing program diatas, maka LED akan meyalah selama satu detik dan kemudian padam selama satu detik demikian seterusnya. Maka rangkaian board arduino dalam keadaan baik.

4.3 Pengujian Rangkaian LCD 16 x 2

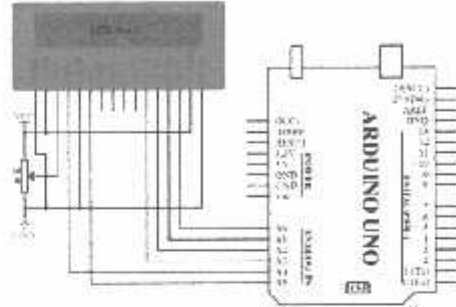
Pengujian ini dilakukan untuk mengetahui apakah port Arduino dapat berfungsi menampilkan 32 karakter (16 karakter di kolom 1 dan 16 karakter di kolom 2)

4.3.1 Peralatan yang digunakan

1. LCD 16 x 2
2. Arduino UNO R 3
3. Variabel resistor 10 K

4.3.2 Langkah-langkah yang dilakukan

1. Rangkaian LCD 16 x 2 variabel resistor 10 K pada Arduino UNO R 3 seperti pada gambar 4.3.



Gambar 4.3 Rangkaian LCD

2. Berikan sketch program seperti terlihat pada gambar di bawah ini.

```
File Edit Sketch Tools Help
HelloWorld $
#include <LiquidCrystal.h>
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
LiquidCrystal lcd(A4, A5, A3, A2, A1, A0);
void setup() {
  Serial.begin(9600);
  lcd.begin(16,2);
}
void loop() {
  lcd.setCursor(0, 0);
  lcd.print("ABCDEFGHIJKLMN");
  lcd.setCursor(0, 1);
  lcd.print("0123456789");
}
```

Gambar 4.4 Sketch program LCD

4.3.3 Hasil pengujian

Dengan mengupload listing program diatas, maka LCD akan menampilkan karakter seperti yang dituliskan dalam program “ ABCDEFGHIJKLMN” pada baris pertama dan pada baris kedua “0123456789”. Seperti pada gambar 4.5



Gambar 4.5 Output hasil pengujian LCD

4.4 Pengujian Koneksi Keypad 4 x 4

Dalam mengkoneksikan Keypad 4x4 pada Arduino memerlukan pengkoneksian awal dengan memprogram Arduino melalui IDE Arduino. Pada gambar 4.6 terdapat perintah `#include <Keypad.h>` ini dimaksudkan untuk menambahkan library keypad pada Arduino dan pin yang digunakan untuk keypad diantaranya pin 2,3,4,5,6,7,8 dan 9 dengan rowPins 6,7,8, dan 9 serta colPins 2,3,4, dan 5



```

keypad4x4_lcd | Arduino IDE v2
File Edit Sketch Tools Help
keypad4x4_lcd
#include <LiquidCrystal.h>
// LiquidCrystal pins
LiquidCrystal lcd(A4, A5, A3, A2, A1, A0);
const int numRows = 4; // Jumlah baris pada Keypad
const int numCols = 4; // Jumlah kolom pada Keypad

const int debounceTime = 100; // Debounce

const int rowPins[numRows] = { 2, 3, 4, 5 }; // Pin x
const int colPins[numCols] = { 6, 7, 8, 9 }; // Pin y

// Keypad Data: 0-9, *, #, dan spasi
const char keypad[numRows][numCols] = {
  { '1', '2', '3', 'A' },
  { '4', '5', '6', 'B' },
  { '7', '8', '9', 'C' },
  { '*', '0', '#', 'D' }
};

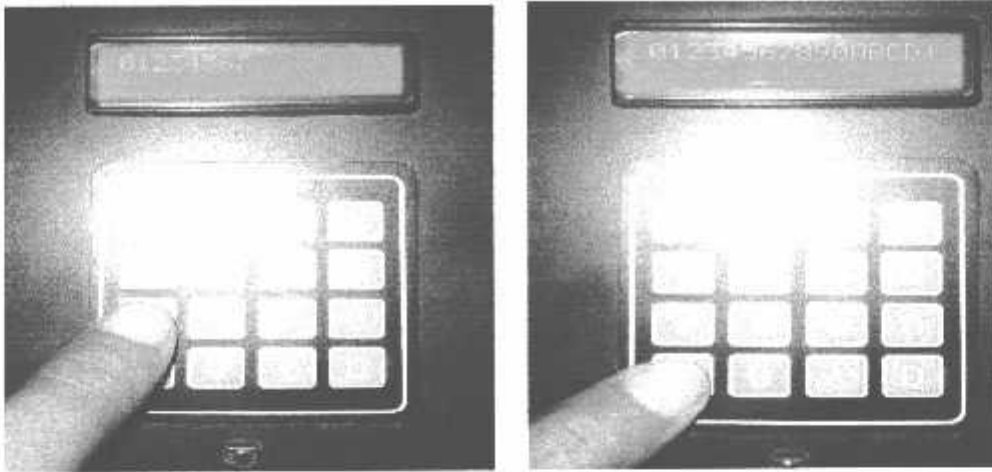
```

Gambar 4.6 Sketch Program arduino bagian keypad

Setelah semuanya sesuai lalu untuk pengujian koneksi keypad dapat dilakukan dengan serial monitor IDE Arduino, dan setiap tombol yang ditekan pada keypad akan terlihat pada LCD.

4.4.1 Hasil pengujian

Dengan mengupload listing program diatas, maka yang perlu dilakukan dengan cara menekan salah satu tombol keypad dan melihat perubahan pada LCD. Jika terjadi proses pembacaan, maka data yang tertampil pada LCD akan berubah sesuai dengan kondisi tombol yang ditekan. Hasil pengujian dapat di lihat pada gambar 4.7 di bawah ini. data yang ditampilkan sesuai dengan kondisi tombol.



Gambar 4.7 Hasil pengujian penekanan tombol pada keypad

4.5 Pengujian Modul DDS AD9850

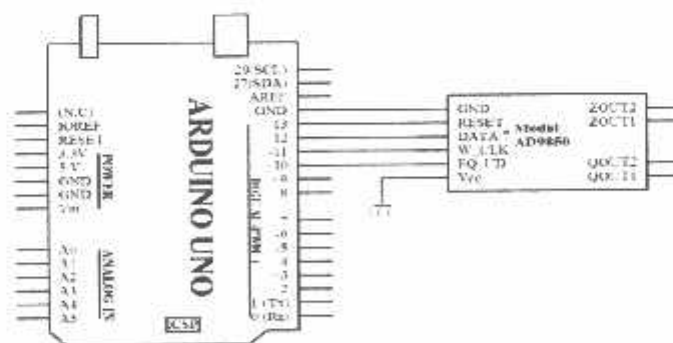
Pada pengujian modul DDS AD9850 ini dilakukan untuk mengetahui apakah modul AD9850 yang digunakan untuk membangkitkan frekuensi dalam keadaan baik atau tidak, dalam pengujian ini akan mencoba membangkitkan frekuensi sebesar 1 KHz.

4.5.1 Peralatan yang digunakan

1. Modul Ad9850
2. Arduino UNO
3. Frekuensi counter
4. Kabel Probe

4.5.2 Langkah-langkah yang dilakukan

1. Rangkaian koneksi antara Arduino dengan Modul AD9850 ditunjukkan pada Gambar 4.7 di bawah ini.



Gambar 4.8 pengujian pada modul AD9850

2. Berikan sketch program seperti terlihat pada gambar di bawah ini.

```

AD9850
File Edit Sketch Tools Help
AD9850
#define VCC 11
#define GND 10
#define DATA 11
#define RESET 13

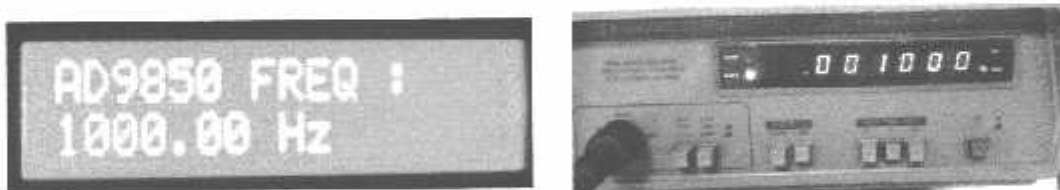
DDS module(DDS module, DDS);
void setup() {
  pinMode(VCC, OUTPUT);
  pinMode(GND, OUTPUT);
  pinMode(DATA, OUTPUT);
  pinMode(RESET, OUTPUT);
}
void loop() {
  digitalWrite(VCC, HIGH);
  digitalWrite(GND, LOW);
  digitalWrite(RESET, LOW);
  digitalWrite(DATA, LOW);
  digitalWrite(VCC, LOW);
  digitalWrite(GND, HIGH);
  digitalWrite(RESET, HIGH);
  digitalWrite(DATA, HIGH);
}

```

Gambar 4.9 Sketch Program Arduino pembangkit frekuensi

4.5.3 Hasil pengujian

Dengan mengupload listing program diatas, maka yang perlu dilakukan yaitu melakukan pengukuran frekuensi menggunakan *GW frequency Counter GFC-8010G* dengan cara menghubungkan kabel probe yang merah pada output 1 dan yang hitam dihubungkan pada ground. Hasil pengujian dapat dilihat pada gambar 4.10 di bawah ini.



Gambar 4.10

- (a) Frekuensi masukan sebesar 1 KHz (b) Frekuensi keluaran sebesar 1 KHz

4.5 Analisa Pengujian Keseluruhan Sistem Pembangkit Frekuensi DDS AD9850

Pada pengujian keseluruhan sistem pembangkit frekuensi ini, untuk mengetahui seberapa besar akurasi pembangkit frekuensi yang bisa di hasilkan oleh modul AD9850, untuk mengetahui apakah perancangan yang sudah di buat sesuai yang di harapkan atau tidak, untuk melakukan pengukuran pembangkit frekuensi ini menggunakan *GW frequency Counter GFC-8010G*. Pada pengujian pembangkit frekuensi ini, diambil beberapa frekuensi yang mewakili rentang frekuensi dari 1 Hz – 42 MHz.

1. Hasil Pengujian Frekuensi 1 Hz

Hasil pengujian untuk pembangkit frekuensi keluaran 1 Hz dari modul AD9850 seperti pada Gambar 4.11. Pada Gambar 4.11.a merupakan hasil pembangkitan frekuensi 1 Hz dari Modul AD9850, Gambar 4.11.b memperlihatkan nilai frekuensi yang diukur menggunakan *GW frequency Counter GFC-8010G* diperoleh nilai sebesar 1 Hz, dengan presentase kesalahannya sebesar 0 %.

2. Hasil Pengujian Frekuensi 100 Hz

Hasil pengujian untuk pembangkit frekuensi keluaran 100 Hz dari modul AD9850 seperti pada Gambar 4.12. Pada Gambar 4.12.a merupakan hasil pembangkitan frekuensi 100 Hz dari Modul AD9850, Gambar 4.12.b memperlihatkan nilai frekuensi yang diukur menggunakan *GW frequency Counter GFC-8010G* diperoleh nilai sebesar 100 Hz, dengan presentase kesalahannya sebesar 0 %.

3. Hasil Pengujian Frekuensi 1 KHz

Hasil pengujian untuk pembangkit frekuensi keluaran 1 KHz dari modul AD9850 seperti pada Gambar 4.13. Pada Gambar 4.13.a merupakan hasil pembangkitan frekuensi 1 KHz dari Modul AD9850, Gambar 4.13.b memperlihatkan nilai frekuensi yang diukur menggunakan *GW frequency Counter GFC-8010G* diperoleh nilai sebesar 1 KHz, dengan presentase kesalahannya sebesar 0 %.

4. Hasil Pengujian Frekuensi 1 MHz

Hasil pengujian untuk pembangkit frekuensi keluaran 1 MHz dari modul AD9850 seperti pada Gambar 4.14. Pada Gambar 4.14.a merupakan hasil pembangkitan frekuensi 1 MHz dari Modul AD9850, Gambar 4.14.b memperlihatkan nilai frekuensi yang diukur menggunakan *GW frequency Counter GFC-8010G* diperoleh nilai sebesar 1,000009 MHz, dengan presentase kesalahannya sebesar 0,0009 %.

5. Hasil Pengujian Frekuensi 40 MHz

Hasil pengujian untuk pembangkit frekuensi keluaran 40 MHz dari modul AD9850 seperti pada Gambar 4.15. Pada Gambar 4.15.a merupakan hasil pembangkitan frekuensi 40 MHz dari Modul AD9850, Gambar 4.15.b memperlihatkan nilai frekuensi yang diukur menggunakan *GW frequency Counter GFC-8010G* diperoleh nilai sebesar 40,000383 MHz, dengan presentase kesalahannya sebesar 0,00096%. Hasil pengujian pembangkit frekuensi menggunakan Modul AD9850 dengan rentang frekuensi dari 1 Hz hingga 40

Mhz (Gelombang Sinus) yang di ukur menggunakan *GW frequency Counter GFC-8010G* seperti pada table 4.1.

Hasil pengujian untuk pembangkit frekuensi menggunakan *GW frequency Counter*



Gambar 4.11

(a) Frekuensi masukan sebesar 1 Hz

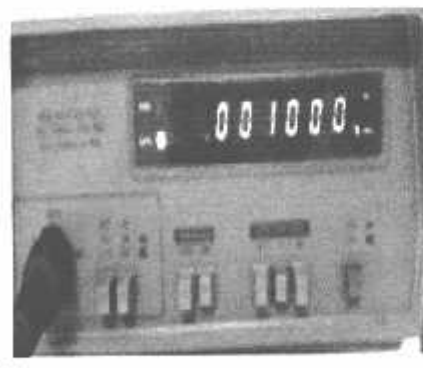
(b) Frekuensi keluaran sebesar 1 Hz



Gambar 4.12

(a) Frekuensi masukan sebesar 100 Hz

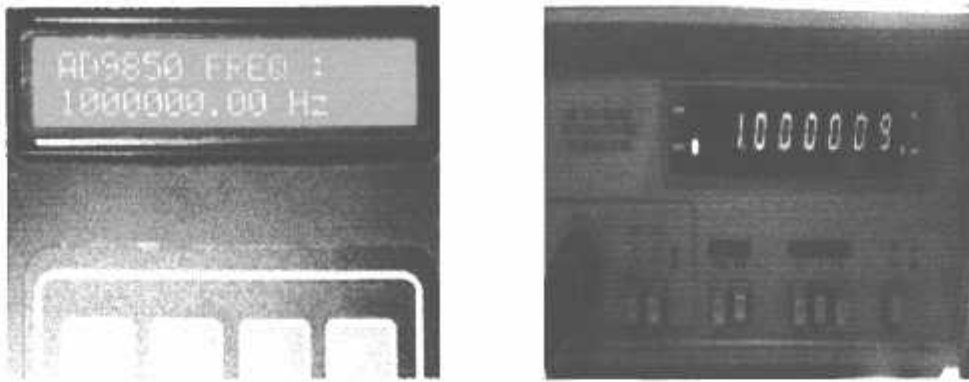
(b) Frekuensi keluaran sebesar 100 Hz



Gambar 4.13

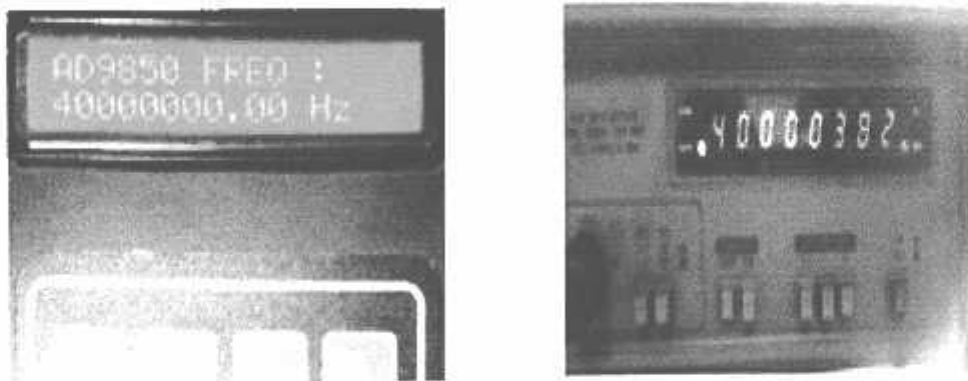
(a) Frekuensi masukan sebesar 1 KHz

(b) Frekuensi keluaran sebesar 1 KHz



Gambar 4.14

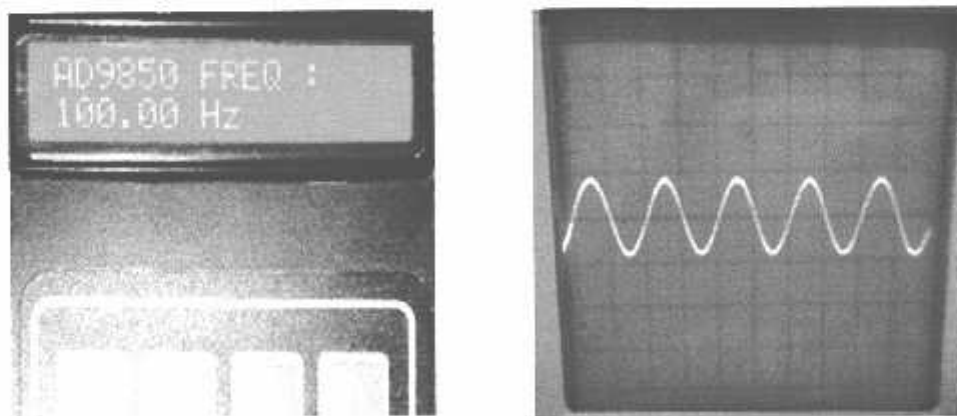
- (a) Frekuensi masukan sebesar 1 MHz (b) Frekuensi keluaran sebesar 1 MHz



Gambar 4.15

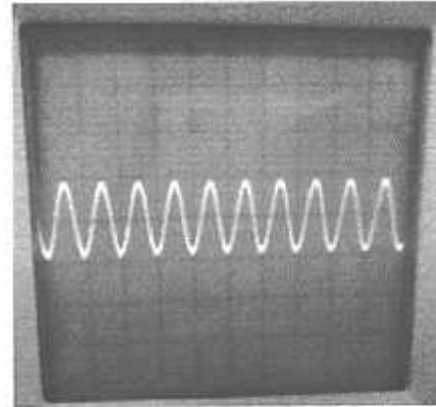
- (a) Frekuensi masukan sebesar 40 MHz (b) Frekuensi keluaran sebesar 40 MHz

Hasil pengujian untuk pembangkitan gelombang sinus menggunakan Osiloskop



Gambar 4.16

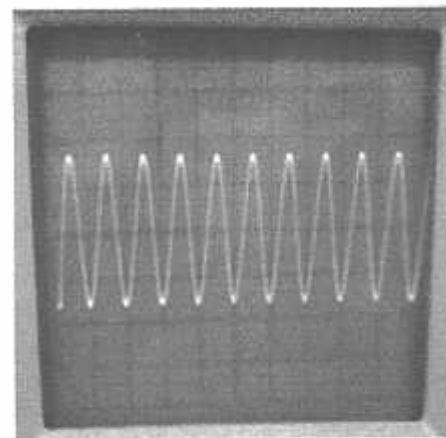
- (a) Frekuensi masukan 100 Hz (b) Bentuk gelombang keluaran AD9850



Gambar 4.17

(a) Frekuensi masukkan 10 KHz

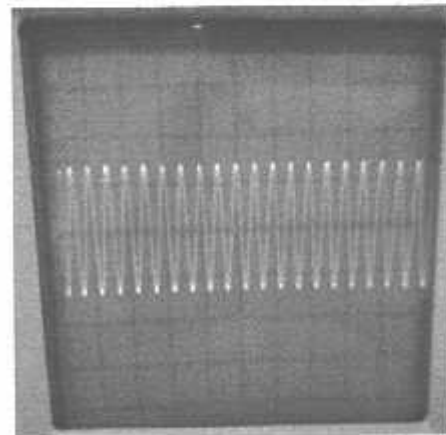
(b) Bentuk gelombang keluaran AD9850



Gambar 4.18

(a) Frekuensi masukkan 1 MHz

(b) Bentuk gelombang keluaran AD9850



Gambar 4.19

(a) Frekuensi masukkan 5 MHz

(b) Bentuk gelombang keluaran AD9850

Tabel 4.1 Data hasil pengujian pembangkit frekuensi menggunakan Modul AD9850 dengan rentang frekuensi dari 1 Hz hingga 42 Mhz (Gelombang Sinus).

No	AD9850 (Hz)	Freq Counter (Hz)	Selisih (%)	No	AD9850 (Hz)	Freq Counter (Hz)	Selisih (%)
1	1	1	0	26	800000	800008	0,001
2	10	10	0	27	1000000	1000009	0,0009
3	25	25	0	28	1500000	1500015	0,001
4	50	50	0	29	4000000	4000038	0,00095
5	100	100	0	30	5000000	5000048	0,00096
6	125	125	0	31	8500000	8500082	0,00097
7	150	150	0	32	10000000	10000097	0,00097
8	200	200	0	33	12000000	12000118	0,00098
9	500	500	0	34	15000000	15000147	0,00098
10	750	750	0	35	17500000	17500171	0,00097
11	1000	1000	0	36	19000000	19000184	0,00096
12	1500	1500	0	37	20000000	20000193	0,00096
13	5000	5000	0	38	21000000	21000204	0,00097
14	10000	10000	0	39	23500000	23500229	0,00097
15	15000	15000	0	40	25000000	25000250	0,001
16	25000	25000	0	41	28000000	28000282	0,001
17	26500	26500	0	42	30000000	30000301	0,001
18	50000	50000	0	43	32500000	32500325	0,001
19	60000	60000	0	44	34500000	34500328	0,00095
20	85500	85500	0	45	35000000	35000330	0,00094
21	100000	100000	0	46	36500000	36500345	0,00095
22	125000	125001	0,0008	47	39000000	39000371	0,00095
23	250000	250002	0,0008	48	40000000	40000383	0,00096
24	500000	500004	0,0008	49	41000000	41000475	0,0011
25	755000	755007	0,0009	50	42000000	42000864	0,0020
Total Selisih							0,02869

4.5.1 Analisa Hasil Pengujian

Dari data hasil pengujian diatas dapat di hitung persentase selisih pembangkit frekuensi dari modul AD9850 yang diukur menggunakan *GW frequency Counter GFC-8010G* dengan perhitungan. Persamaan yang digunakan adalah :

$$\% \text{ Selisih} = \left| \frac{\text{Hasil Pengujian} - \text{Hasil Pengukuran}}{\text{Hasil Pengukuran}} \right| \times 100\%$$

Dari hasil pengujian :

$$1. \quad \% \text{ Selisih} = \left| \frac{125000 - 125001}{125001} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,0008 \%$$

$$2. \quad \% \text{ Selisih} = \left| \frac{250000 - 250002}{250002} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,0008 \%$$

$$3. \quad \% \text{ Selisih} = \left| \frac{500000 - 500004}{500004} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,0008 \%$$

$$4. \quad \% \text{ Selisih} = \left| \frac{755000 - 755007}{755007} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,0009 \%$$

$$5. \quad \% \text{ Selisih} = \left| \frac{800000 - 800008}{800008} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,001 \%$$

$$6. \quad \% \text{ Selisih} = \left| \frac{1000000 - 1000009}{1000009} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,0009 \%$$

$$7. \quad \% \text{ Selisih} = \left| \frac{1500000 - 1500015}{1500015} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,001 \%$$

$$8. \quad \% \text{ Selisih} = \left| \frac{4000000 - 4000038}{4000038} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00095 \%$$

$$9. \ \% \text{ Selisih} = \left| \frac{5000000 - 5000048}{5000048} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00096 \%$$

$$10. \ \% \text{ Selisih} = \left| \frac{8500000 - 8500082}{8500082} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00097 \%$$

$$11. \ \% \text{ Selisih} = \left| \frac{10000000 - 10000097}{10000097} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00097 \%$$

$$12. \ \% \text{ Selisih} = \left| \frac{12000000 - 12000118}{12000118} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00098 \%$$

$$13. \ \% \text{ Selisih} = \left| \frac{15000000 - 15000147}{15000147} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00098 \%$$

$$14. \ \% \text{ Selisih} = \left| \frac{17500000 - 17500171}{17500171} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00097 \%$$

$$15. \ \% \text{ Selisih} = \left| \frac{19000000 - 19000184}{19000184} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00096\%$$

$$16. \ \% \text{ Selisih} = \left| \frac{20000000 - 20000193}{20000193} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00096 \%$$

$$17. \ \% \text{ Selisih} = \left| \frac{21000000 - 21000204}{21000204} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00097 \%$$

$$18. \ \% \text{ Selisih} = \left| \frac{23500000 - 23500229}{23500229} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00097 \%$$

$$19. \% \text{ Selisih} = \left| \frac{25000000 - 25000250}{25000250} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,001 \%$$

$$20. \% \text{ Selisih} = \left| \frac{28000000 - 28000282}{28000282} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,001 \%$$

$$21. \% \text{ Selisih} = \left| \frac{30000000 - 30000301}{30000301} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,001 \%$$

$$22. \% \text{ Selisih} = \left| \frac{32500000 - 32500325}{32500325} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,001 \%$$

$$23. \% \text{ Selisih} = \left| \frac{34500000 - 34500328}{34500328} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00095 \%$$

$$24. \% \text{ Selisih} = \left| \frac{35000000 - 35000330}{35000330} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00094 \%$$

$$25. \% \text{ Selisih} = \left| \frac{36500000 - 36500345}{36500345} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00095 \%$$

$$26. \% \text{ Selisih} = \left| \frac{39000000 - 39000371}{39000371} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00095 \%$$

$$27. \% \text{ Selisih} = \left| \frac{40000000 - 40000383}{40000383} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,00096 \%$$

$$28. \% \text{ Selisih} = \left| \frac{41000000 - 41000475}{41000475} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,0011 \%$$

$$29. \ \% \text{ Selisih} = \left| \frac{42000000 - 42000864}{42000864} \right| \times 100\%$$

$$\% \text{ Selisih} = 0,0020 \%$$

Dari nilai Selisih tersebut dapat diambil nilai rata-rata dari 50 percobaan diatas dengan persamaan :

$$\% \text{ Selisih} = \frac{\text{Total Selisih}}{\text{Jumlah Percobaan}}$$

Dari hasil perhitungan :

$$\% \text{ Selisih} = \frac{0,02869}{50}$$

$$\% \text{ Selisih} = 0,00057\%$$

BAB V PENUTUP

5.1 Kesimpulan

Setelah dilakukan perancangan, pengujian dan analisa sistem, maka pada perancangan pengontrol modul DDS AD9850 sebagai pembangkit frekuensi pada saat melakukan pengukuran menggunakan *GW frequency Counter GFC-8010G* dapat disimpulkan sebagai berikut:

1. Untuk nilai selisih rata-rata dari rentang frekuensi 1 Hz hingga 42 MHz nilai selisih dari 50 percobaan dengan presentase kesalahannya sebesar 0,00057 %.
2. Bentuk gelombang keluaran dari AD9850 dengan rentang frekuensi 1 Hz – 42 MHz, gelombang sinus dalam keadaan sempurna meskipun memiliki presentase kesalahan.

5.2 Saran

Pada perancangan dan pembuatan tugas akhir ini tak lepas dari kekurangan-kekurangan baik dari perancangan sistem maupun peralatan yang digunakan, untuk itu agar sistem dapat menjadi lebih baik maka diperlukan beberapa perbaikan dan penyempurnaan, yaitu :

1. Untuk mengatur selisih keluaran frekuensi lebih baik menggunakan Rotary encoder di bandingkan keypad.
2. Pengkabelan harus dilakukan dengan baik dan benar sehingga tidak menyebabkan error pada saat melakukan pengujian.

DAFTAR PUSTAKA

- [1] Hastuti, Perancangan dan simulasi DDS (Direct Digital Synthesizer) pada aplikasi radar FMCW pada S-band [Online]. Available:
<http://digilab.itelkom.ac.id/index.php?option=comrepository&Itemid34&Task=detail&mid=11030189>.
- [2] D. Kurniawan, ATmega8+DDS (Direct Digital Synthesizer), [Online]. Available:
<http://elektronika.web.id/elkav2/index.php?topic=978.msg16112#msg16112>.
- [3] Arduino uno. [Online].
<http://blog.langitrobotika.com/article/robotika/pengertian-arduino-uno/.html>, diakses 6 April 2016.
- [4] Software Arduino. [Online].
<https://www.arduino.cc/en/Guide/Windows>, diakses 04 April 2016
- [5] CMOS 125 MHz DDS/DAC Synthesizer AD9850, Norwood, USA: Analog Device.
- [6] PRAJITNO, “Perancangan Pembangkit Sinyal Sinusoida 78 Mhz. Dengan Teknik Direct Digital Synthesizer”, *Prosiding Pertemuan dan Presentasi Ilmiah Teknologi Akselerator dan Aplikasinya*, Vol. 12, Yogyakarta, (2010) 90 – 97.
- [7] Liquid Crystal Display (LCD) 16x 2. [Online].
<http://www.leselektronika.com/2012/06/liquid-crystal-display-lcd-16-x-2.html>, diakses 7 April 2016.
- [8] Matrix Keypad 4x4. [Online].
<http://elektronika-dasar.web.id/matrix-keypad-4x4-untuk-mikrokontroler/>, diakses 6 April 2016.

LAMPIRAN



BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

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

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

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Wilfridus M. Halek
NIM : 1212207
Program Studi : TEKNIK ELEKTRO S-1
Konsentrasi : TEKNIK ELEKTRONIKA
Judul Skripsi : **PERENCANAAN DAN PEMBUATAN PENNGONTROL
MODUL DIRECT DIGITAL SYNTHESIZER AD9850
SEBAGAI PEMBANGKIT FREKUENSI BERBASIS
ARDUINO UNO**

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

Hari : Kamis
Tanggal : 4 Agustus 2016
Dengan Nilai : 81,95 (A)

Panitia Ujian Skripsi

Ketua Majelis Penguji

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

Sekretaris Majelis Penguji

Dr. Eng. I Komang Somawirata, ST, MT
NIP.P. 1030100361

Anggota Penguji

Penguji I

Dr. Eng. Arvanto Soetedjo, ST, MT
NIP.Y. 1030800417

Penguji II

Ir. Eko Nurcahyo, MT
NIP.Y. 1028700172



PERSETUJUAN PERBAIKAN SKRIPSI

Dari hasil ujian skripsi Program Studi Teknik Elektro jenjang strata satu (S-1) yang diselenggarakan pada :

Hari : Kamis
Tanggal : 4 Agustus 2016

Telah dilakukan perbaikan skripsi oleh :

Nama : Wilfridus M. Halek
NIM : 1212210
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul Skripsi : **Perencanaan Dan Pembuatan Pengontrol Modul Direct Digital Synthesizer AD9850 Sebagai Pembangkit Frekuensi Berbasis Arduino Uno**

No	Materi Perbaikan	Keterangan
1.	Tambahkan nilai digit dibelakang koma	

Dosen Penguji II

Ir. Eko Nurcahyo, MT
NIP.Y. 1028700172

Dosen Pembimbing I

Dr. F. Yudi Limraptono, ST, MT
NIP.Y. 1039500274

Dosen Pembimbing II

Dr. Eng. I Komang Somawirata, ST, MT
NIP.P. 1030100361



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

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

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417838 Fax. (0341) 417834 Malang

Nomor Surat : ITN-174/EL-FTI/2015
Lampiran : -
Perihal : BIMBINGAN SKRIPSI (Baru)

8 Maret 2016

Kepada : Yth. Bapak/Ibu DR.Ir.F.Yudi Lumpraptono. MT
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa:

Nama : WILFRIDUS M. HALEK
Nim : 1212210
Fakultas : Teknologi Industri
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

“ Semester Genap Tahun Akademik 2015-2016 ”

Demikian atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Ketua Program Studi Teknik Elektro S-1



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



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

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

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

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

Nomor Surat : ITN-174/EL-FTI/2015

8 Maret 2016

Lampiran : -

Perihal : BIMBINGAN SKRIPSI (Baru)

Kepada : Yth. Bapak/Ibu DR.Eng.Komang Somawirata.ST,MT
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa:

Nama : WILFRIDUS M. HALEK
Nim : 1212210
Fakultas : Teknologi Industri
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

" Semester Genap Tahun Akademik 2015-2016 "

Demikian atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui

Ketua Program Studi Teknik Elektro S-1



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



PERMOHONAN PERSETUJUAN SKRIPSI

Yang Bertanda Tangan Dibawah Ini:

Nama : Wilfridus M. Halek
 NIM : 12.12.2.10
 Semester : VII (Delapan)
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-I
 Konsentrasi : ~~TEKNIK ENERGI LISTRIK~~
 TEKNIK ELEKTRONIKA
 TEKNIK KOMPUTER DAN INFORMATIKA
 TEKNIK KOMPUTER
 TEKNIK TELEKOMUNIKASI
 Alamat : Jl. Puncaksono Gang V No. 6 Karangah

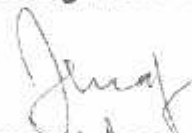
Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat SKRIPSI Tingkat Sarjana. Untuk melengkapi permohonan tersenut, bersama ini kami lampirkan persyaratan-persyaratan yang harus dipenuhi.

Adapun persyaratan- persyaratan pengambilan SKRIPSI adalah sebagai berikut:

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan laporan Praktek Kerja (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
4. Telah menempuh matakuliah > 134 sks dengan IPK > 2 dan tidak ada nilai E (.....)
5. Telah mengikuti secara aktif kegiatan seminar Skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenarannya data tersebut diatas
 Recording Teknik Elektro S-I


 (..... M. Idris)

Disetujui
 Ketua Prodi Teknik Elektro S-I


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

Malang, 15-02.....2016

Pemohon


 (..... Wilfridus M. Halek)

Mengetahui
 Dosen Wali


 (..... M. Ibrahim Ashari, ST, MT)

 NIP. P. 1030100358


Catatan:

Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Prodi T. elektro S-I

1. 1.P. 4.69 / 3-40
2. / 138

BERITA ACARA RAPAT PERSETUJUAN JUDUL/PROPOSAL SKRIPSI
PROGRAM STUDI TEKNIK ELEKTRO S-1
Konsentrasi : *T. ELKA S1*





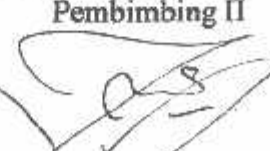
Tanggal : *29-2-2016*

1.	NIM	<i>1212260</i>
2.	Nama	<i>Walfaidus M. Hales</i>
3.	Judul yang diajukan	<i>Perencanaan dan Pembuatan Pengontrol Kecepatan Direct Digital Synthesis Motor AC 3-Phase dengan Menggunakan Teknik Berbasis Arduino Uno</i>
4.	<input checked="" type="checkbox"/> Disetujui/Ditolak	
5.	Catatan:	
6.	Pembimbing yang diusulkan:	<i>1. Dr. F. Yudi L. ST. MT</i> <i>2. Dr. Eng. I. Kang S. ST. MT</i>
Menyetujui		
1. Koordinator Dosen Kelompok Keahlian		
		

* : Coret yang tidak perlu

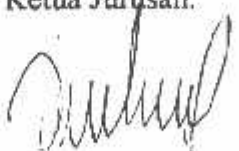
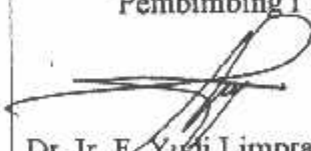



**BERITA ACARA SEMINAR PROPOSAL SKRIPSI
 PROGRAM STUDI TEKNIK ELEKTRO S1**

KONSENTRASI		T. ELEKTRONIKA S1				
1.	Nama Mahasiswa	WILFRIDUS M. HALEK		NIM	1212210	
2.	Keterangan	Tanggal	Waktu	Tempat / Ruang		
	Pelaksanaan	17 Maret 2016				
Spesifikasi Judul (berilah tanda silang *)						
3.	a.	Sistem Tenaga Elektrik	e.	Embbded System	i.	Sistem Informasi
	b.	Konversi Energi	f.	Antar Muka	j.	Jaringan Komputer
	c.	Sistem Kendali	g.	Elektronika Telekomunikasi	k.	Web
	d.	Tegangan Tinggi	h.	Elektronika Instrumentasi	l.	Algoritma Cerdas
4.	Judul Proposal yang diseminarkan Mahasiswa	Perencanaan dan Pembuatan Pengontrolan Modul Direct Digital Synthesizer AD9851 Sebagai Pembangkit Frekuensi Berbasis Arduino UNO				
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian				
6.	Catatan :	<i>ditany dikanti dgn keypad!</i>				
Catatan :						
Persetujuan Judul Skripsi						
7.	Disetujui, Dosen Keahlian I			Disetujui, Dosen Keahlian II		
						
		
Mengetahui, Ketua Jurusan		Disetujui, Calon Dosen Pembimbing				
		Pembimbing I		Pembimbing II		
M. Ibrahim Ashari,ST,MT NIP. P. 1030100358						
		DR.Ir.F.Yudi Limpraptono. MT		DR.Eag.Komang Somawirata.ST,MT		



BERITA ACARA SEMINAR PROGRESS SKRIPSI PROGRAM STUDI TEKNIK ELEKTRO S1

KONSENTRASI		T. Elektronika		
1.	Nama Mahasiswa	Wilfridus M. Halek	NIM	1212210
2.	Keterangan	Tanggal	Waktu	Tempat / Ruang
	Pelaksanaan			
3.	Judul Skripsi	Perencanaan Dan Pembuatan Pengontrol Modul Direct Digital Synthesizer AD9851 Sebagai Pembangkit Frekuensi Berbasis Arduino Uno		
4.	Perubahan Judul		
5.	Catatan: - Flowchart: bcd?, mengatur pilihan frek (Waktu)!			
	- pelajari tentang A.D. 2857, cara mengatur dll.			
6.	Mengetahui, Ketua Jurusan.  <u>M. Ibrahim Ashari, ST, MT</u>	Disetujui, Dosen Pembimbing		
		Pembimbing I  Dr. Ir. E. Yudi Limpraptono, MT	Pembimbing II  Dr. Eng. I Komang Somawirata, ST, MT	



Formulir Perbaikan Ujian Skripsi

Dalam Pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi T.Energi Listrik, /
T. Elektronika, /T. Komputer, / T.Telekomunikasi, Maka Perlu Adanya Perbaikan Skripsi Untuk Mahasiswa:

Nama : Wilfridus A. Halad
NIM : 1212210
Perbaikan Meliputi :

- Batasan masalah lebih jelas.
- Urutay Resaler frekuensi ?
- Pengujian waktu perrosesan
(sinyal keluar) untuk frekuensi yg
berbeda / besa.

Malang.....4/7/16.....20

(.....)



PROGRAM STUDI TEKNIK ELEKTRO S-1

**FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

Kampus II : Jl Raya Karanglo Km. 2 Malang

**MONITORING BIMBINGAN SKRIPSI
SEMESTER GENAP TAHUN AKADEMIK 2015-2016**

Nama Mahasiswa : Wilfridus M. Halek
NIM : 1212210
Nama Pembimbing : Dr. F. Yudi Limpraptono, ST, MT
Judul Skripsi : PERENCANAAN DAN PEMBUATAN PENGONTROL
MODUL DIRECT DIGITAL SYNTHESIZER AD9851
SEBAGAI PEMBANGKIT FREKUENSI BERBASIS
ARDUINO UNO

Minggu Ke-	Hari, Tanggal	Waktu Bimbingan	Materi Bimbingan	Paraf
1	Kelasa 07/4 2016	11.00	Bab I & II	
2	Senin 08/4 2016	16.00	Bab III	
3	Rabu 14/4 2016	13.00	Proyek	
4	Selasa 14/4 2016	12.30	Bab IV & V	
5	Senin 20/4 2016	16.00	Bab IV + V (Review)	
6	Rabu 22/4 2016	16.00	Material Ferriter Lensa	
7	Rabu 29/4 2016	16.00	Review	

Malang, 23/6/2016

Pembimbing

Dr. F. Yudi Limpraptono, ST, MT
NIP.Y. 1039500274



**MONITORING BIMBINGAN SKRIPSI
SEMESTER GENAP TAHUN AKADEMIK 2015-2016**

Nama Mahasiswa : Wilfridus M. Halek
NIM : 1212210
Nama Pembimbing : Dr. Eng. I Komang Somawirata, ST, MT.
Judul Skripsi : PERENCANAAN DAN PEMBUATAN PENGONTROL
MODUL DIRECT DIGITAL SYNTHESIZER AD9851
SEBAGAI PEMBANGKIT FREKUENSI BERBASIS
ARDUINO UNO

Minggu Ke-	Hari, Tanggal	Waktu Bimbingan	Materi Bimbingan	Paraf
1	Selam 5/4	13.30	Konsultasi bab I & II	
2	Jumat. 13/4	11.30	Makalah sumbu	
3	Selam 21/4	10.27	Konsultasi perencanaan	
4	Selam 29/4	10.31	Makalah sumbu - Di manfaatkan/diingat	
5	Rening 20/5	11.12	AEC ujian skripsi	
6				
7				

Malang,

Pembimbing

Dr. Eng. I Komang Somawirata, ST, MT.
NIP. P. 1030100361

SURAT PERNYATAAN ORISINALITAS

Yang bertandatangan di bawah ini :

Nama : Wilfridus M. Halek
NIM : 12.12.210
Program Studi : T. Elektro S-1
Konsentrasi : Teknik Elektronika

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

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

Malang, 28 Agustus 2016

Yang membuat Pernyataan,



Wilfridus M. Halek
NIM : 1212210

```

/*****
* Program Arduino
* Nama      : Wilfridus M. Halek
* NIM      : 12 12 210
* Project   : Tugas Akhir Skripsi 2016
* Judul Skripsi : Perencanaan Dan Pembuatan Pengontrol Modul Direct Digital Synthesizer
                AD9850 Sebagai Pembangkit Frekuensi Berbasis Arduino Uno
*****/

```

```

#include <LiquidCrystal.h>
#include <Keypad.h>
#define W_CLK 11 // Pin 11 - connect to AD9850 module word load clock pin (CLK)
#define FQ_UD 10 // Pin 10 - connect to freq update pin (FQ)
#define DATA 12 // Pin12 - connect to serial data load pin (DATA)
#define RESET 13 // Pin 13 - connect to reset pin (RST).
#define pulseHigh(pin) {digitalWrite(pin, HIGH); digitalWrite(pin, LOW);}

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(A4, A5, A3, A2, A1, A0);

const byte ROWS = 4; //four rows
const byte COLS = 4; //three columns
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {2, 3, 4, 5}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {6, 7, 8, 9}; //connect to the column pinouts of the keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
char key, karakter, text[15] = {' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '};
int baris = 0, kolom = 1;
double freq, freq_buffer[16];
bool enter = 0;
unsigned int add = 1;

// transfers a byte, a bit at a time, LSB first to the 9850 via serial DATA line
void tfr_byte(byte data)
{
  for (int i=0; i<8; i++, data>>=1) {
    digitalWrite(DATA, data & 0x01);
    pulseHigh(W_CLK); //after each bit sent, CLK is pulsed high
  }
}

```

```

    }
// frequency calc from datasheet page 8 = <sys clock> * <frequency tuning word>/2^32
void sendFrequency(double frequency) {
int32_t freq = frequency * 4294967295/125000000; // note 125 MHz clock on 9850
for (int b=0; b<4; b++, freq>>=8) {
tfr_byte(freq & 0xFF);
}
tfr_byte(0x000); // Final control byte, all 0 for 9850 chip
pulseHigh(FQ_UD); // Done! Should see output
}
void setup() {
// configure arduino data pins for output
pinMode(FQ_UD, OUTPUT);
pinMode(W_CLK, OUTPUT);
pinMode(DATA, OUTPUT);
pinMode(RESET, OUTPUT);

// set up the LCD's number of columns and rows:
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.print("AD9850 FREQ :");
Serial.begin(9600);
}
void loop() {
lcd.setCursor(baris, kolom);
key = keypad.getKey();
while(key) //ketika keypad ditekan, maka akan masuk ke statement ini
{
karakter = key;
if(karakter != 'A' && karakter != 'B' && karakter != 'C' && karakter != 'D' && karakter != '*' &&
karakter != '#') //jika tombol 1,2,3,4,5,6,7,8,9,0 ditekan maka akan muncul angka tersebut di
LCD
{
text[baris] = karakter;
convert();
lcd.print(karakter);
baris++;
display_cursor();
break;
}
if(karakter == '*') //jika tombol * ditekan maka akan menghapus angka terakhir
{
baris--;
while(baris < 0)
{
baris = 0;
break;
}
text[baris] = ' ';
lcd.setCursor(baris, kolom);
}
}
}

```

```

    lcd.print(text[baris]);
    break;
}
if(karakter == 'A' && text[baris] < '9' && text[baris] != ' ') //jika tombol A ditekan dan
angkanya kurang dari 9 maka angka akan bertambah
{
    text[baris]++;
    convert();
    lcd.setCursor(baris, kolom);
    lcd.print(text[baris]);
    break;
}
if(karakter == 'A' && text[baris] == '9' ) //jika tombol A ditekan dan angkanya adalah 9
maka angka tidak akan bertambah
{
    break;
}
if(karakter == 'B' && text[baris] > '0' && text[baris] != ' ') //jika tombol B ditekan dan
angkanya lebih dari 0 maka angka akan berkurang
{
    text[baris]--;
    convert();
    lcd.setCursor(baris, kolom);
    lcd.print(text[baris]);
    break;
}
if(karakter == 'B' && text[baris] == '0' ) //jika tombol B ditekan dan angkanya adalah 0
maka angka tidak akan berkurang
{
    break;
}
if(karakter == 'A' && text[baris] == ' ') //jika tombol A ditekan dan tidak ada angka maka
angka tidak akan bertambah
{
    break;
}
if(karakter == 'B' && text[baris] == ' ') //jika tombol B ditekan dan tidak ada angka maka
angka tidak akan bertambah
{
    break;
}
if(karakter == 'C') //jika tombol C ditekan maka pointer berpindah ke arah kiri
{
    baris--;
    while(baris < 0)
    {
        baris = 0;
        break;
    }
    display_cursor();
}

```

```

break;
}
if(karakter == 'D' && text[baris] != ' ') //jika D ditekan maka pointer berpindah ke arah
kanan
{
    baris++;
    display_cursor();
    break;
}
if(karakter == 'D' && text[baris] == ' ') //jika tombol D ditekan tetapi pointer ada di sisi
paling kanan maka pointer tidak berpindah
{
    break;
}
if(karakter == '#' && text[baris] == ' ') //tombol # berfungsi sebagai enter
{
    enter = 1;
    break;
}
if(karakter == '#' && text[baris] != ' ') //tombol # berfungsi sebagai enter
{
    break;
}
}
if(enter == 1 && baris == 0) //ketika nilai frekuensi yg dimasukkan kosong
{
    enter = 0;
}
if(enter == 1 && baris == 1) //ketika nilai frekuensi yg dimasukkan hanya 1 digit
{
    freq = freq_buffer[0];
    lcd.setCursor(0, 0);
    lcd.print("AD9850 FREQ :");
    lcd.setCursor(0, 1);
    lcd.print(freq);
    lcd.print(" Hz");
    pulseHigh(RESET);
    pulseHigh(W_CLK);
    pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet page 12 figure 10
    while(1)
    {
        sendFrequency(freq);
        key = keypad.getKey();
        if(key == 'A')
        {
            lcd.clear();
            freq = freq + add;
            lcd.setCursor(0, 0);
            lcd.print("AD9850 FREQ :");
            lcd.setCursor(0, 1);

```

```

lcd.print(freq);
lcd.print(" Hz");
}
if(key == 'B')
{
lcd.clear();
freq = freq - add;
lcd.setCursor(0, 0);
lcd.print("AD9850 FREQ :");
lcd.setCursor(0, 1);
lcd.print(freq);
lcd.print(" Hz");
}

if(key == 'C')
{
add = add * 10;
}

if(key == 'D')
{
add = add / 10;
}
if(add > 1000)
{
add = 1000;
}
if(add < 1)
{
add = 1;
}
}
}
if(enter == 1 && baris == 2)//ketika nilai frekuensi yg dimasukkan ada 2 digit
{
freq = (freq_buffer[0] * 10) + freq_buffer[1];
lcd.setCursor(0, 0);
lcd.print("AD9850 FREQ :");
lcd.setCursor(0, 1);
lcd.print(freq);
lcd.print(" Hz");
pulseHigh(RESET);
pulseHigh(W_CLK);
pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet page 12 figure 10
while(1)
{
sendFrequency(freq);
key = keypad.getKey();
if(key == 'A')
{

```

```

lcd.clear();
freq = freq + add;
lcd.setCursor(0, 0);
lcd.print("AD9850 FREQ :");
lcd.setCursor(0, 1);
lcd.print(freq);
lcd.print(" Hz");
}
if(key == 'B')
{
lcd.clear();
freq = freq - add;
lcd.setCursor(0, 0);
lcd.print("AD9850 FREQ :");
lcd.setCursor(0, 1);
lcd.print(freq);
lcd.print(" Hz");
}
if(key == 'C')
{
add = add * 10;
}
if(key == 'D')
{
add = add / 10;
}
if(add > 1000)
{
add = 1000;
}
if(add < 1)
{
add = 1;
}
}
if(enter == 1 && baris == 3)//ketika nilai frekuensi yg dimasukkan ada 3 digit
{
freq = (freq_buffer[0] * 100) + (freq_buffer[1] * 10) + freq_buffer[2];
lcd.setCursor(0, 0);
lcd.print("AD9850 FREQ :");
lcd.setCursor(0, 1);
lcd.print(freq);
lcd.print(" Hz");
pulseHigh(RESET);
pulseHigh(W_CLK);
pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet page 12 figure 10
while(1)
{
sendFrequency(freq);
}
}

```

```

key = keypad.getKey();
if(key == 'A')
{
  lcd.clear();
  freq = freq + add;
  lcd.setCursor(0, 0);
  lcd.print("AD9850 FREQ:");
  lcd.setCursor(0, 1);
  lcd.print(freq);
  lcd.print(" Hz");
}
if(key == 'B')
{
  lcd.clear();
  freq = freq - add;
  lcd.setCursor(0, 0);
  lcd.print("AD9850 FREQ:");
  lcd.setCursor(0, 1);
  lcd.print(freq);
  lcd.print(" Hz");
}
if(key == 'C')
{
  add = add * 10;
}
if(key == 'D')
{
  add = add / 10;
}
if(add > 1000)
{
  add = 1000;
}
if(add < 1)
{
  add = 1;
}
}
if(enter == 1 && baris == 4)//ketika nilai frekuensi yg dimasukkan ada 4 digit
{
  freq = (freq_buffer[0] * 1000) + (freq_buffer[1] * 100) + (freq_buffer[2] * 10) +
  freq_buffer[3];
  lcd.setCursor(0, 0);
  lcd.print("AD9850 FREQ:");
  lcd.setCursor(0, 1);
  lcd.print(freq);
  lcd.print(" Hz");
  pulseHigh(RESET);
  pulseHigh(W_CLK);
}

```

```
pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet page 12 figure 10
while(1)
```

```
{
  sendFrequency(freq);
  key = keypad.getKey();
  if(key == 'A')
  {
    lcd.clear();
    freq = freq + add;
    lcd.setCursor(0, 0);
    lcd.print("AD9850 FREQ :");
    lcd.setCursor(0, 1);
    lcd.print(freq);
    lcd.print(" Hz");
  }
  if(key == 'B')
  {
    lcd.clear();
    freq = freq - add;
    lcd.setCursor(0, 0);
    lcd.print("AD9850 FREQ :");
    lcd.setCursor(0, 1);
    lcd.print(freq);
    lcd.print(" Hz");
  }
  if(key == 'C')
  {
    add = add * 10;
  }
  if(key == 'D')
  {
    add = add / 10;
  }
  if(add > 1000)
  {
    add = 1000;
  }
  if(add < 1)
  {
    add = 1;
  }
}
```

```
if(enter == 1 && baris == 5)//ketika nilai frekuensi yg dimasukkan ada 5 digit
```

```
{
  freq = (freq_buffer[0] * 10000) + (freq_buffer[1] * 1000) + (freq_buffer[2] * 100) +
  (freq_buffer[3] * 10) + freq_buffer[4];
  lcd.setCursor(0, 0);
  lcd.print("AD9850 FREQ :");
```

```

lcd.setCursor(0, 1);
lcd.print(freq);
lcd.print(" Hz");
pulseHigh(RESET);
pulseHigh(W_CLK);
pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet page 12 figure 10
while(1)
{
sendFrequency(freq);
key = keypad.getKey();
if(key == 'A')
{
lcd.clear();
freq = freq + add;
lcd.setCursor(0, 0);
lcd.print("AD9850 FREQ:");
lcd.setCursor(0, 1);
lcd.print(freq);
lcd.print(" Hz");
}
if(key == 'B')
{
lcd.clear();
freq = freq - add;
lcd.setCursor(0, 0);
lcd.print("AD9850 FREQ:");
lcd.setCursor(0, 1);
lcd.print(freq);
lcd.print(" Hz");
}
if(key == 'C')
{
add = add * 10;
}
if(key == 'D')
{
add = add / 10;
}
if(add > 1000)
{
add = 1000;
}
if(add < 1)
{
add = 1;
}
}
if(enter == 1 && baris == 6)//ketika nilai frekuensi yg dimasukkan ada 6 digit
{

```

```

freq = (freq_buffer[0] * 100000) + (freq_buffer[1] * 10000) + (freq_buffer[2] * 1000) +
(freq_buffer[3] * 100) + (freq_buffer[4] * 10) + freq_buffer[5];
lcd.setCursor(0, 0);
lcd.print("AD9850 FREQ :");
lcd.setCursor(0, 1);
lcd.print(freq);
lcd.print(" Hz");
pulseHigh(RESET);
pulseHigh(W_CLK);
pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet page 12 figure 10
while(1)
{
sendFrequency(freq);
key = keypad.getKey();

if(key == 'A')
{
lcd.clear();
freq = freq + add;
lcd.setCursor(0, 0);
lcd.print("AD9850 FREQ :");
lcd.setCursor(0, 1);
lcd.print(freq);
lcd.print(" Hz");
}
if(key == 'B')
{
lcd.clear();
freq = freq - add;
lcd.setCursor(0, 0);
lcd.print("AD9850 FREQ :");
lcd.setCursor(0, 1);
lcd.print(freq);
lcd.print(" Hz");
}
if(key == 'C')
{
add = add * 10;
}
if(key == 'D')
{
add = add / 10;
}
if(add > 1000)
{
add = 1000;
}
if(add < 1)
{
add = 1;
}
}

```

```

    }
  }
}
if(enter == 1 && baris == 7)//ketika nilai frekuensi yg dimasukkan ada 7 digit
{
freq = (freq_buffer[0] * 1000000) + [freq_buffer[1] * 100000] + (freq_buffer[2] * 10000) +
(freq_buffer[3] * 1000) + (freq_buffer[4] * 100) + (freq_buffer[5] * 10) + freq_buffer[6];
  lcd.setCursor(0, 0);
  lcd.print("AD9850 FREQ :");
  lcd.setCursor(0, 1);
  lcd.print(freq);
  lcd.print(" Hz");
  pulseHigh(RESET);
  pulseHigh(W_CLK);
  pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet page 12 figure 10
  while(1)
  {
    sendFrequency(freq);
    key = keypad.getKey();
    if(key == 'A')
    {
      lcd.clear();
      freq = freq + add;
      lcd.setCursor(0, 0);
      lcd.print("AD9850 FREQ :");
      lcd.setCursor(0, 1);
      lcd.print(freq);
      lcd.print(" Hz");
    }
    if(key == 'B')
    {
      lcd.clear();
      freq = freq - add;
      lcd.setCursor(0, 0);
      lcd.print("AD9850 FREQ :");
      lcd.setCursor(0, 1);
      lcd.print(freq);
      lcd.print(" Hz");
    }
    if(key == 'C')
    {
      add = add * 10;
    }
    if(key == 'D')
    {
      add = add / 10;
    }
    if(add > 1000)
    {
      add = 1000;
    }
  }
}

```

```

    }
    if(add < 1)
    {
        add = 1;
    }
}
}
if(enter == 1 && baris == 8)//ketika nilai frekuensi yg dimasukkan ada 8 digit
{
    freq = (freq_buffer[0] * 10000000) + (freq_buffer[1] * 1000000) + (freq_buffer[2] * 100000)
+ (freq_buffer[3] * 10000) + (freq_buffer[4] * 1000) + (freq_buffer[5] * 100) + (freq_buffer[6]
* 10) + freq_buffer[7];
    lcd.setCursor(0, 0);
    lcd.print("AD9850 FREQ :");
    lcd.setCursor(0, 1);
    lcd.print(freq);
    lcd.print(" Hz");
    pulseHigh(RESET);
    pulseHigh(W_CLK);
    pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet page 12 figure 10
    while(1)
    {
        sendFrequency(freq);
        key = keypad.getKey();
        if(key == 'A')
        {
            lcd.clear();
            freq = freq + add;
            lcd.setCursor(0, 0);
            lcd.print("AD9850 FREQ :");
            lcd.setCursor(0, 1);
            lcd.print(freq);
            lcd.print(" Hz");
        }
        if(key == 'B')
        {
            lcd.clear();
            freq = freq - add;
            lcd.setCursor(0, 0);
            lcd.print("AD9850 FREQ :");
            lcd.setCursor(0, 1);
            lcd.print(freq);
            lcd.print(" Hz");
        }
        if(key == 'C')
        {
            add = add * 10;
        }
        if(key == 'D')
        {

```

```

add = add / 10;
    }
if(add > 1000)
    {
add = 1000;
    }
if(add < 1)
    {
add = 1;
    }
    }
}

if(enter == 1 && baris == 9)//ketika nilai frekuensi yg dimasukkan ada 9 digit
    {
freq = (freq_buffer[0] * 100000000) + (freq_buffer[1] * 10000000) + (freq_buffer[2] *
1000000) + (freq_buffer[3] * 100000) + (freq_buffer[4] * 10000) + (freq_buffer[5] * 1000) +
(freq_buffer[6] * 100) + (freq_buffer[7] * 10) + freq_buffer[8];
    lcd.setCursor(0, 0);
    lcd.print("AD9850 FREQ :");
    lcd.setCursor(0, 1);
    lcd.print(freq);
    lcd.print(" Hz");
    pulseHigh(RESET);
    pulseHigh(W_CLK);
    pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet page 12 figure 10
    while(1)
        {
sendFrequency(freq);
key = keypad.getKey();
if(key == 'A')
    {
    lcd.clear();
    freq = freq + add;
    lcd.setCursor(0, 0);
    lcd.print("AD9850 FREQ :");
    lcd.setCursor(0, 1);
    lcd.print(freq);
    lcd.print(" Hz");
    }
if(key == 'B')
    {
    lcd.clear();
    freq = freq - add;
    lcd.setCursor(0, 0);
    lcd.print("AD9850 FREQ :");
    lcd.setCursor(0, 1);
    lcd.print(freq);
    lcd.print(" Hz");
    }
if(key == 'C')

```

```

    {
    add = add * 10;
    }
    if(key == 'D')
    {
    add = add / 10;
    }
    if(add > 1000)
    {
    add = 1000;
    }
    if(add < 1)
    {
    add = 1;
    }
    }
}
if(enter == 1 && baris == 10)//ketika nilai frekuensi yg dimasukkan ada 10 digit
{
freq = (freq_buffer[0] * 1000000000) + (freq_buffer[1] * 1000000000) + (freq_buffer[2] *
100000000) + (freq_buffer[3] * 1000000) + (freq_buffer[4] * 100000) + (freq_buffer[5] *
10000) + (freq_buffer[6] * 1000) + (freq_buffer[7] * 100) + (freq_buffer[8] * 10) +
freq_buffer[9];
    lcd.setCursor(0, 0);
    lcd.print("AD9850 FREQ :");
    lcd.setCursor(0, 1);
    lcd.print(freq);
    lcd.print(" Hz");
    pulseHigh(RESET);
    pulseHigh(W_CLK);
    pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet page 12 figure 10
    while(1)
    {
    sendFrequency(freq);
    key = keypad.getKey();
    if(key == 'A')
    {
    lcd.clear();
    freq = freq + add;
    lcd.setCursor(0, 0);
    lcd.print("AD9850 FREQ :");
    lcd.setCursor(0, 1);
    lcd.print(freq);
    lcd.print(" Hz");
    }
    if(key == 'B')
    {
    lcd.clear();
    freq = freq - add;
    lcd.setCursor(0, 0);

```

```

lcd.print("AD9850 FREQ :");
lcd.setCursor(0, 1);
lcd.print(freq);
lcd.print(" Hz");
}
if(key == 'C')
{
add = add * 10;
}
if(key == 'D')
{
add = add / 10;
}

if(add > 1000)
{
add = 1000;
}
if(add < 1)
{
add = 1;
}
}
}
}
void display_cursor()
{
lcd.setCursor(baris, kolom);
lcd.print(text[baris]);
delay(200);
lcd.setCursor(baris, kolom);
lcd.print(" ");
delay(200);
lcd.setCursor(baris, kolom);
lcd.print(text[baris]);
delay(200);
lcd.setCursor(baris, kolom);
lcd.print(" ");
delay(200);
lcd.setCursor(baris, kolom);
lcd.print(text[baris]);
}
}
void convert()//program untuk mengkonversi data dari string ke integer
{
if(text[baris] == 48)
{
freq_buffer[baris] = 0;
}
if(text[baris] == 49)
{

```

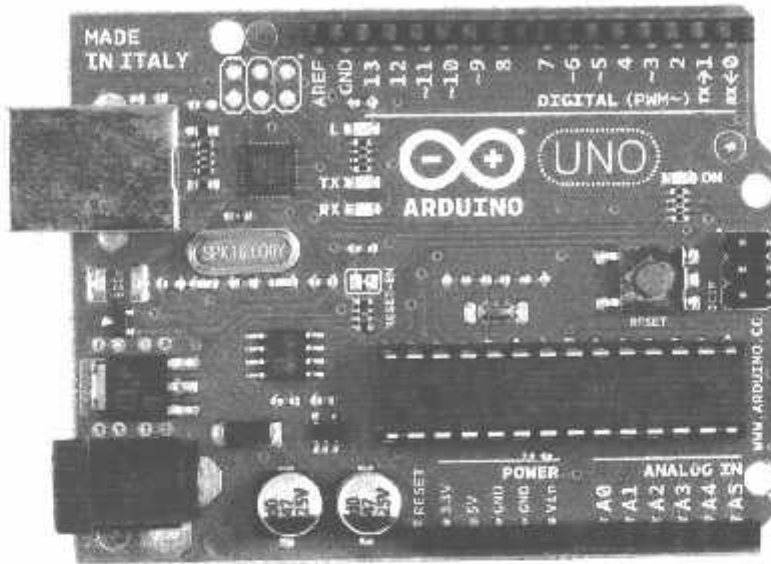
```
freq_buffer[baris] = 1;
    }
    if(text[baris] == 50)
        {
        freq_buffer[baris] = 2;
        }
    if(text[baris] == 51)
        {
        freq_buffer[baris] = 3;
        }
    if(text[baris] == 52)
        {
        freq_buffer[baris] = 4;
        }
    if(text[baris] == 53)
        {
        freq_buffer[baris] = 5;
        }
    if(text[baris] == 54)
        {
        freq_buffer[baris] = 6;
        }
    if(text[baris] == 55)
        {
        freq_buffer[baris] = 7;
        }
    if(text[baris] == 56)
        {
        freq_buffer[baris] = 8;
        }
    if(text[baris] == 57)
        {
        freq_buffer[baris] = 9;
        }
    }
```

DATA SHEET

Beberapa Foto dari Alat Pembangkit Frekuensi AD9850



Arduino UNO



Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Index

Technical Specifications	Page 2
How to use Arduino Programming Enviroment, Basic Tutorials	Page 6
Terms & Conditions	Page 7
Enviromental Policies half sqm of green via Impatto Zero®	Page 7



radiospores

RADIONICS



Technical Specification

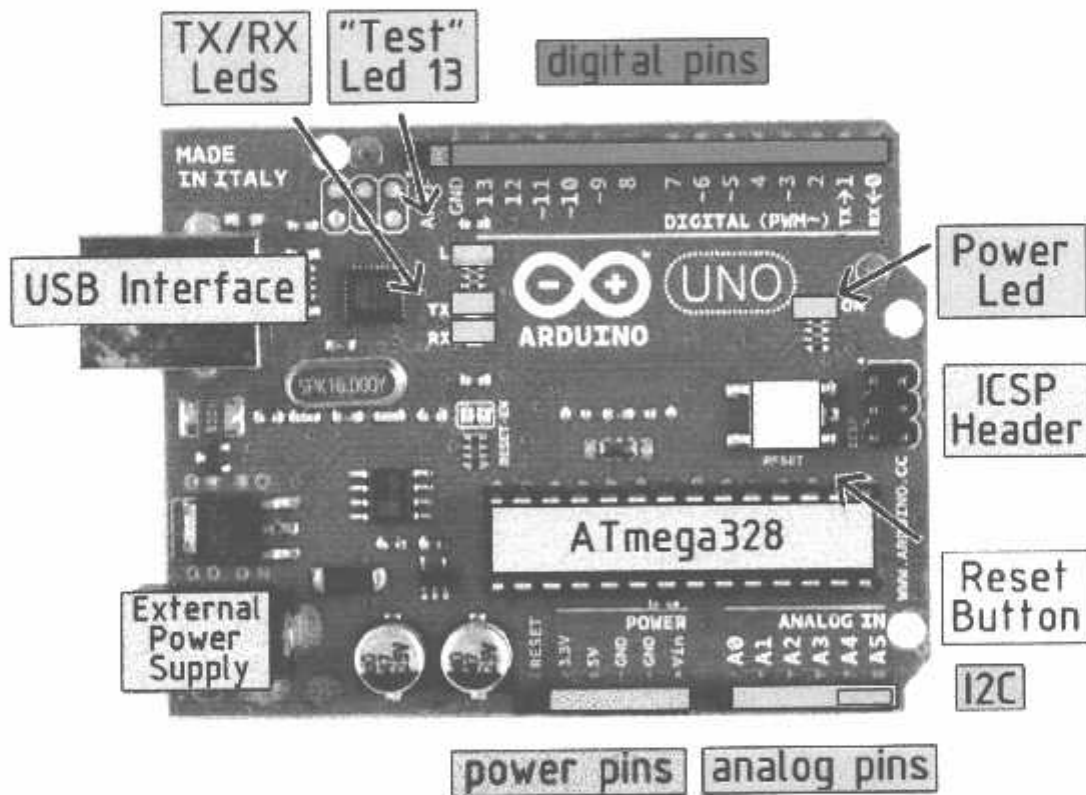


EAGLE files: <http://arduino.cc/wiki/boards/uno-design.zip> Schematic: <http://arduino.cc/wiki/boards/uno/schematic>

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



Radiospere

RAMONICS



Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.



radiospaces

RADIONICS



The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with `analogReference()`.
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and Atmega328 pins](#).

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).



ADIDISGORES

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

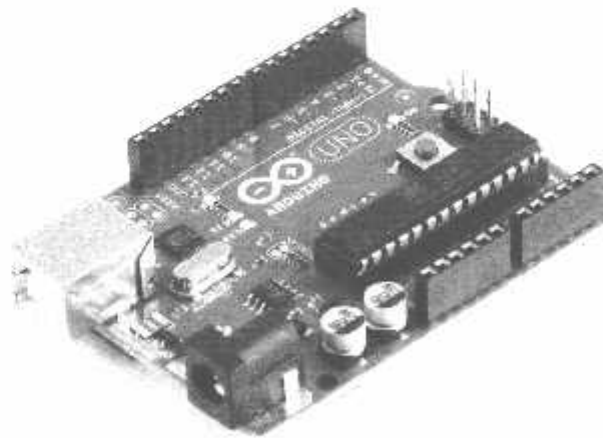
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



radiospares

RADIONICS



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](http://arduino.cc/en/Guide/HomePage) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

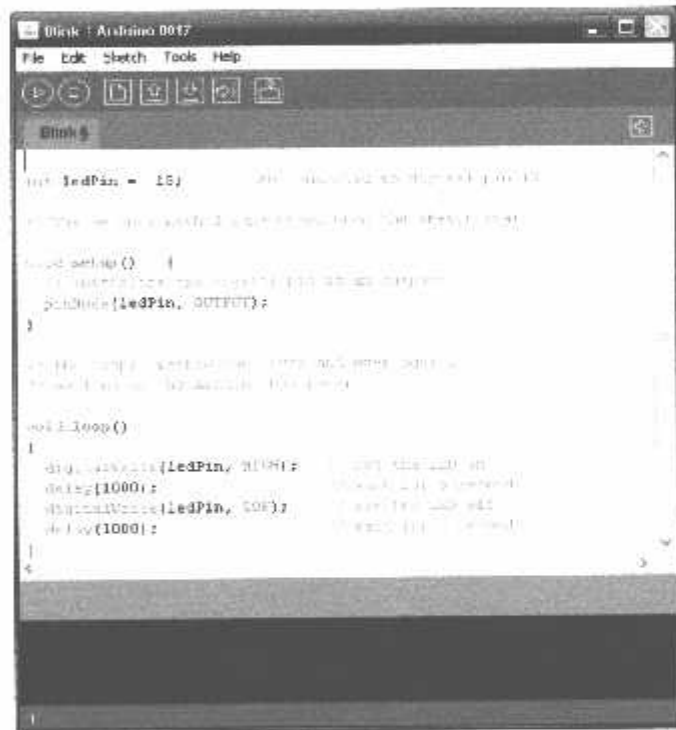
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.



Done compiling

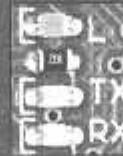
Press Compile button
(to check for errors)



Upload



TX RX Flashing



Blinking Led!

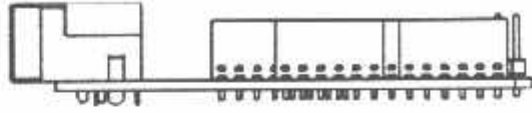


radiospares

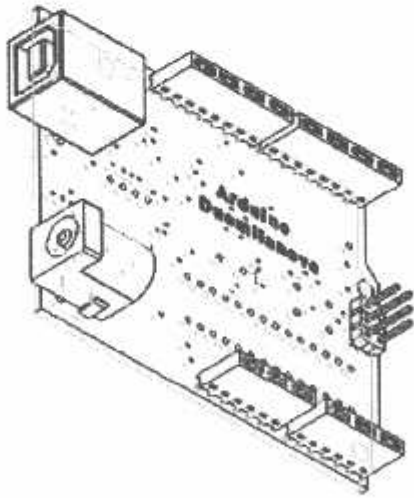
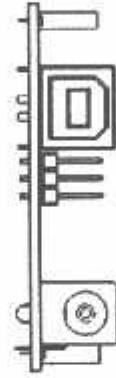
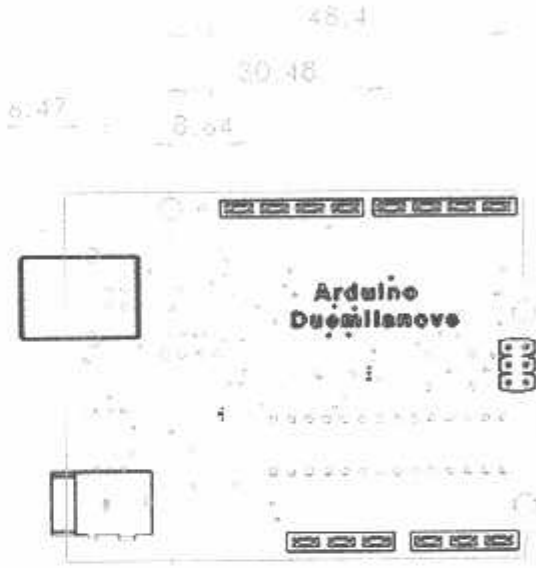
RADIONICS



Dimensioned Drawing



10.01



RADIOSPARRES

RADNOMICS



Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



radiospares

RADIONICS



EIModule

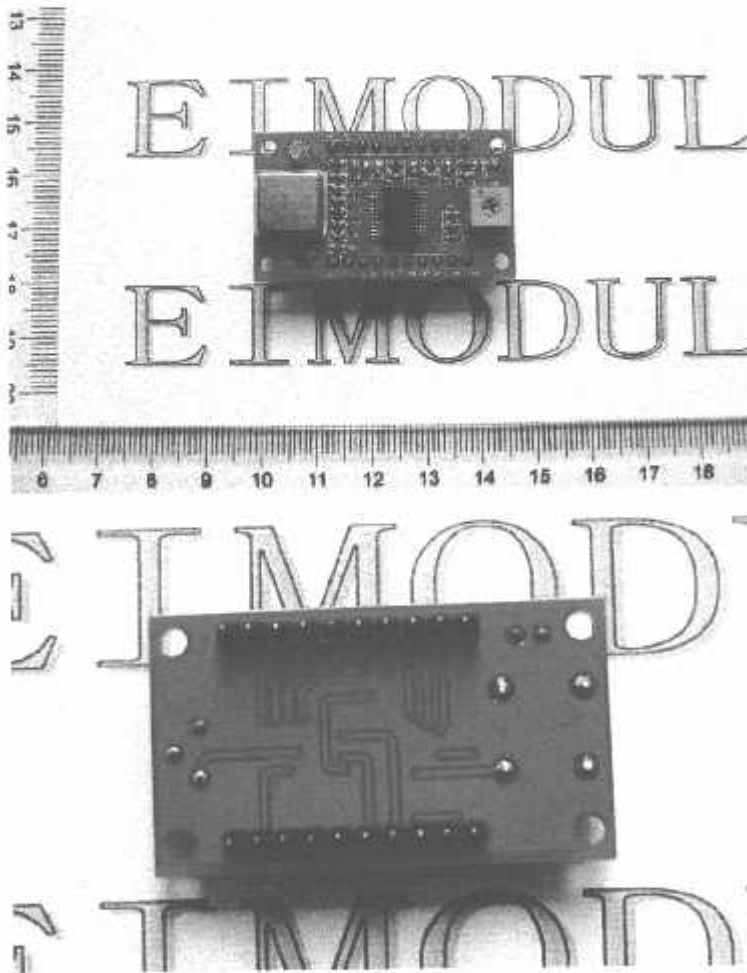
ADS9850 Signal Generator Module

1. Introduction

This module described here is based on ADS9850, a CMOS, 125MHz, and Complete DDS Synthesizer.

The AD9850 is a highly integrated device that uses advanced DDS technology coupled with an internal high speed, high performance, D/A converter and comparator, to form a complete digitally programmable frequency synthesizer and clock generator function.

All the external components which are needed are integrated on the board and the designer don't need to care more about the detailed design of ADS9850. The designer only needs to add the power and control signals to this module to driver this module



<http://www.eimodule.com>

EIModule

2. FEATURES and APPLICATIONS

2.1 FEATURES:

- **Signal Frequency output range: 0-40MHz**
- **4 Signal outputs:**
 - 2 sine wave outputs and 2 square wave outputs**
- **DAC SFDR > 50 dB @ 40 MHz AOUT**
- **32-Bit Frequency Tuning Word**
- **Simplified Control Interface: Parallel Byte or Serial Loading Format**
- **Phase Modulation Capability**
- **+3.3 V or +5 V Single Supply Operation**
- **Low Power: 380 mW @ 125 MHz (+5 V)**
- **Low Power: 155 mW @ 110 MHz (+3.3 V)**
- **Power-Down Function**

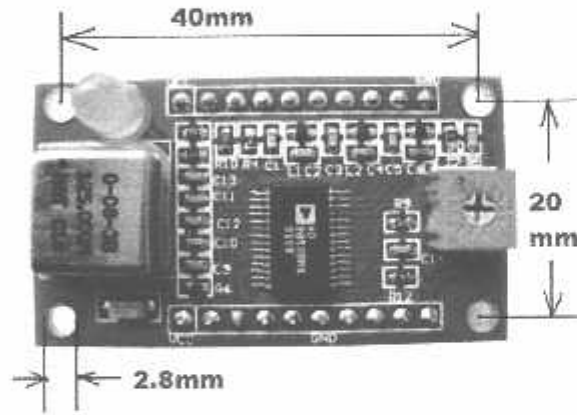
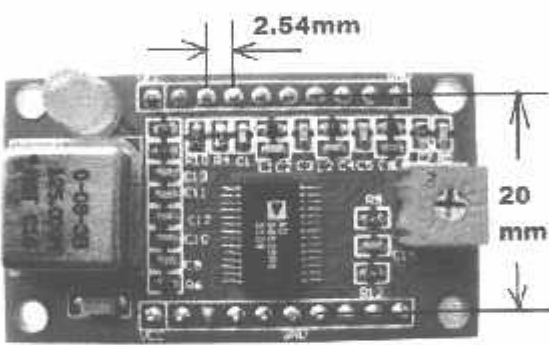
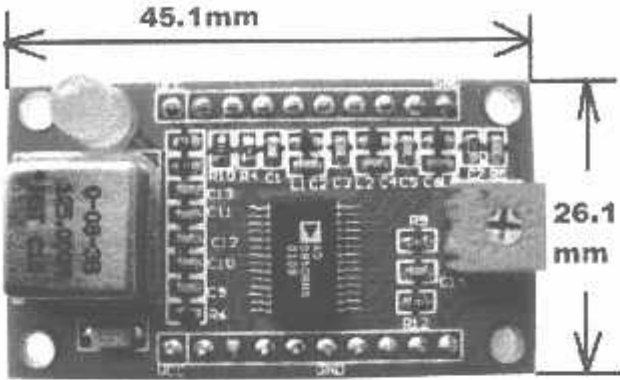
2.2 APPLICATIONS

- **Frequency/Phase-Agile Sine-Wave Synthesis**
- **Clock Recovery and Locking Circuitry for Digital Communications**
- **Digitally Controlled ADC Encode Generator**
- **Agile Local Oscillator Applications**

<http://www.eimodule.com>

EIModule

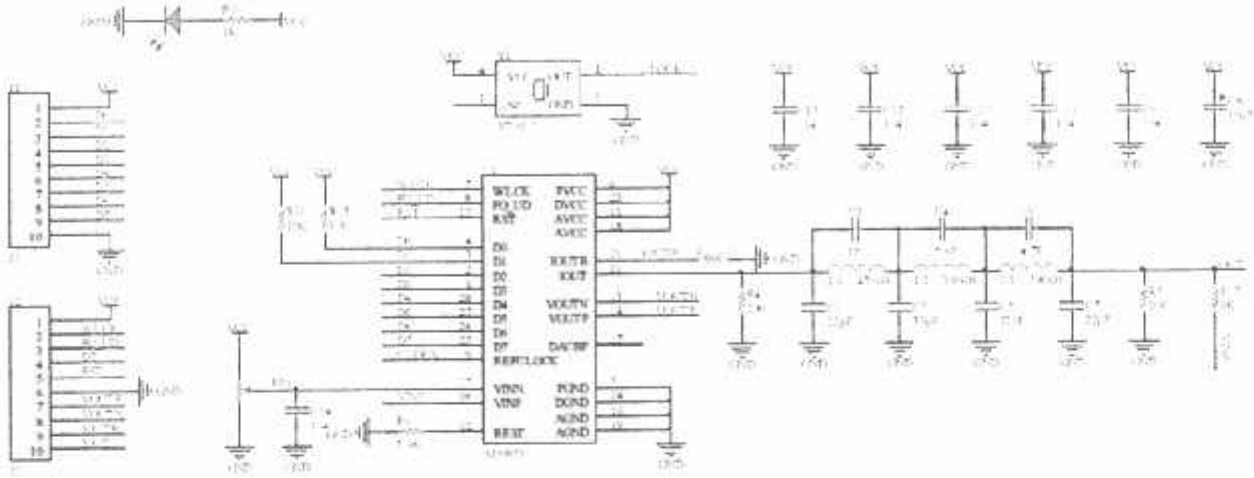
3. The assembly drawing



<http://www.eimodule.com>

EIModule

4. Schematic

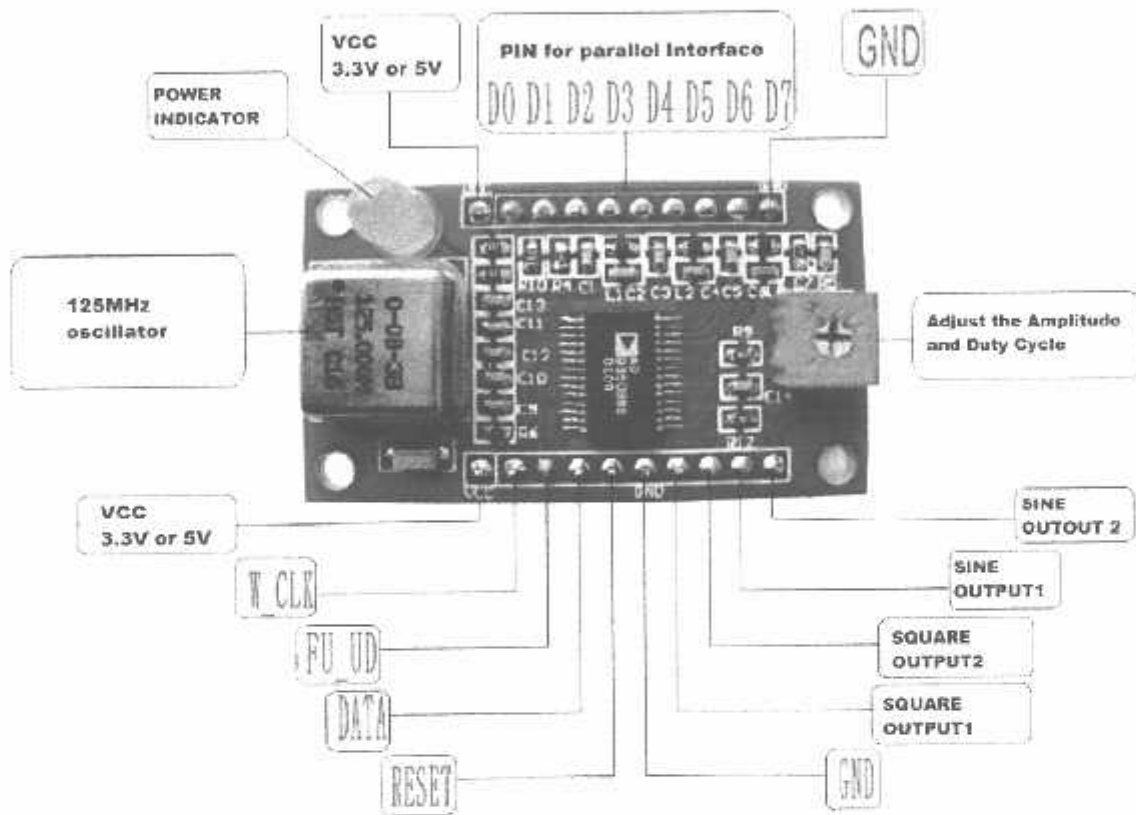


<http://www.eimodule.com>

EIModule

4. How to drive this module

1) Pin definition



Symbol	Type	Function
VCC	P	This is a voltage supply pin. 3.3V or 5V power input
GND	P	This is a ground pin.
W_CLK	I	Word Load Clock. This clock is used to load the parallel or serial frequency/phase/control words
FQ_UD	I	Frequency Update. On the rising edge of this clock, the DDS will update to the frequency (or phase) loaded in the data input register, it then resets the pointer to Word 0
DATA	I	D7, Serial Load
RESET	I	Reset. This is the master reset function; when set high it clears all registers (except the input register) and

<http://www.eimodule.com>

EIModule

		the DAC output will go to Cosine 0 after additional clock cycles
D0-D7	I	8-Bit Data Input. This is the 8-bit data port for iteratively loading the 32-bit frequency and 8-bit phase/28-25 control word. D7 = MSB; D0 = LSB. D7 (Pin 25) also serves as the input pin for the 40-bit serial data word.
Square Wave Output1	O	This is the comparator's true output
Square Wave Output1	O	This is the comparator's complement output.
Sine Wave Output1	O	Analog Current Output of the DAC.
Sine Wave Output1	O	The Complementary Analog Output of the DAC.

2) DC Characteristics

- 1) Power supply: 3.3V or 5.0V
- 2) Interface voltage: 3.3V if power supply is 3.3V.
5.0V if power supply is 5.0V

Characteristics	Symbol	Min	Typ	Max	Unit
Logic "1" Voltage +5 V Supply	V_{IH}	3.5	5.0	6	V
Logic "1" Voltage +3.3 V Supply	V_{IH}	3.0	3.3	6	V
Logic "0" Voltage	V_{IL}		-	0.4	V

EIModule

3) Software description

The AD9850 contains a 40-bit register that is used to program the 32-bit frequency control word, the 5-bit phase modulation word and the power-down function. This register can be loaded in a parallel or serial mode.

In the parallel load mode, the register is loaded via an 8-bit bus; the full 40-bit word requires five iterations of the 8-bit word. The `W_CLK` and `FQ_UD` signals are used to address and load the registers. The rising edge of `FQ_UD` loads the (up to) 40-bit control data word into the device and resets the address pointer to the first register. Subsequent `W_CLK` rising edges load the 8-bit data on words [7:0] and move the pointer to the next register. After five loads, `W_CLK` edges are ignored until either a reset or an `FQ_UD` rising edge resets the address pointer to the first register.

In serial load mode, subsequent rising edges of `W_CLK` shift the 1-bit data on Lead 25 (D7) through the 40 bits of programming information. After 40 bits are shifted through, an `FQ_UD` pulse is required to update the output frequency (or phase).

For detailed information for Programming the AD9850, please download the following two documents on our website: www.eimodule.com (Search EIM377)

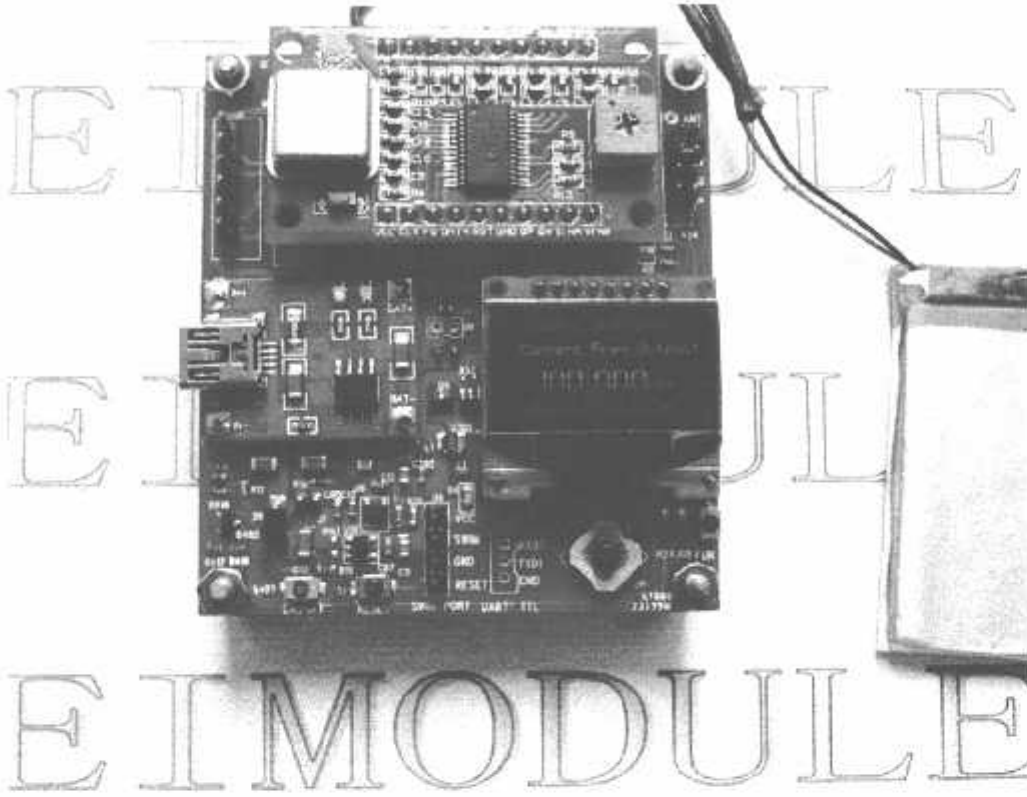
1. AD9850_specification.pdf (From Page 9)
2. AD9850_SW_samples.zip

We also have demo board for this module for you. See the picture on the next page.

<http://www.eimodule.com>

EIModule

5. Pictures for working module

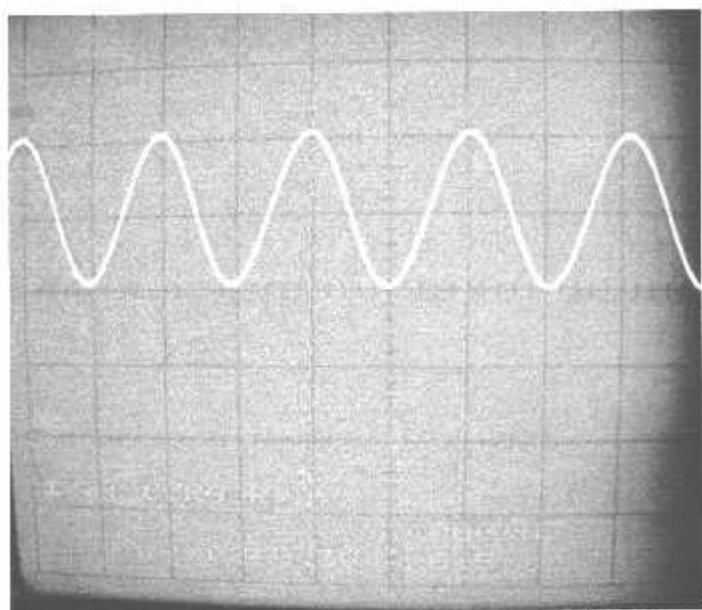


<http://www.eimodule.com>

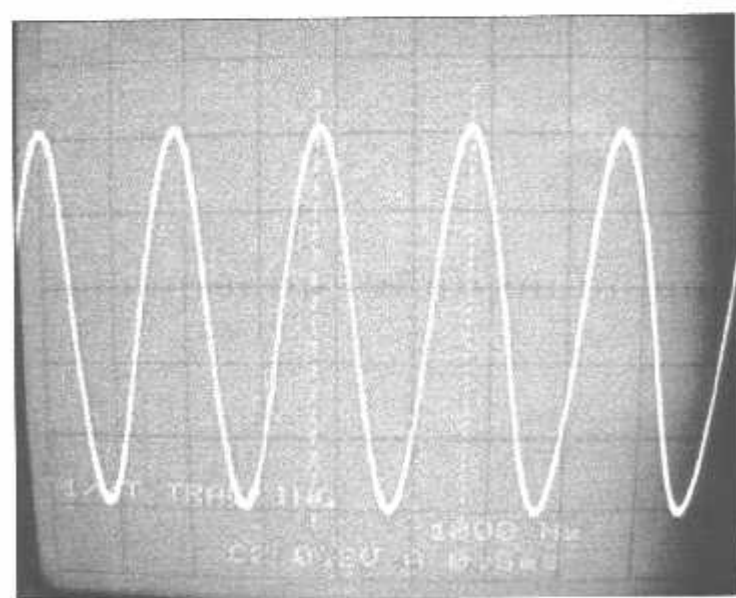
EIModule

6. Test results

1MHz 正弦波输出图:



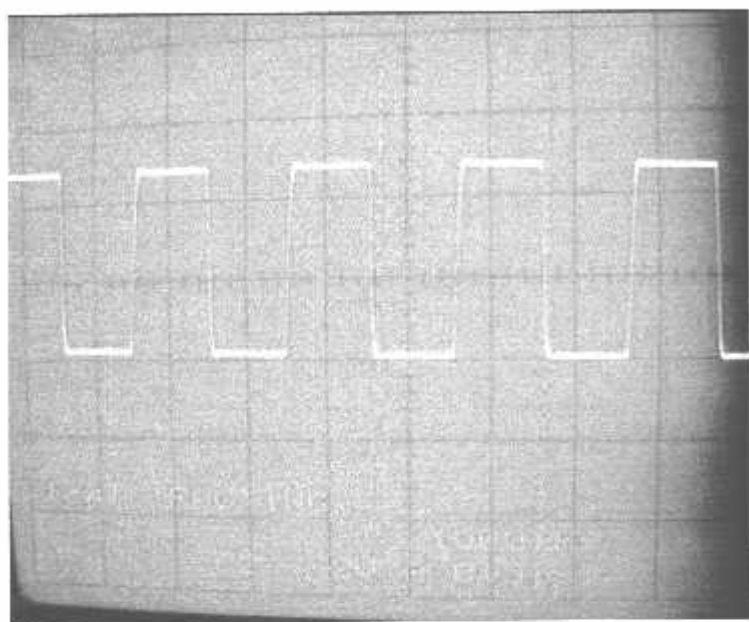
1kHz 正弦波输出图:



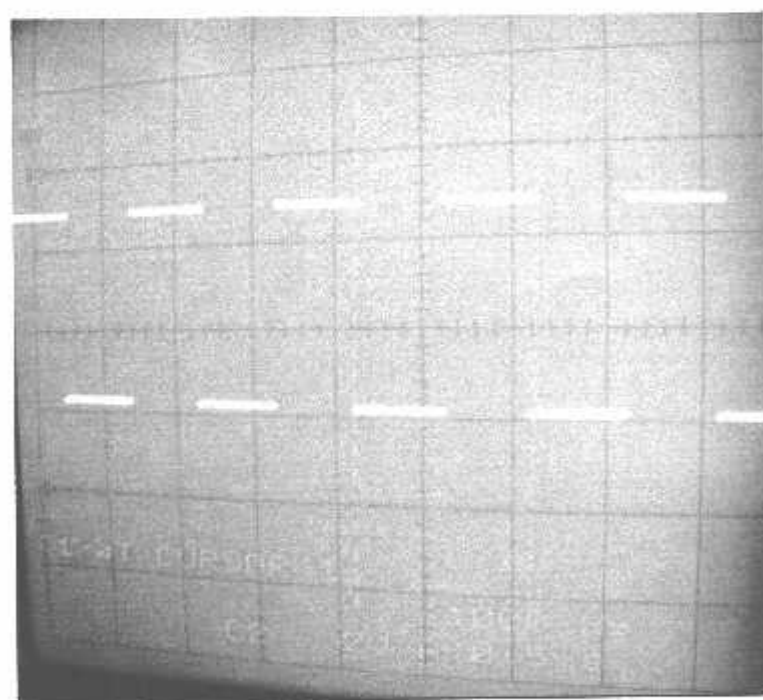
<http://www.eimodule.com>

EIModule

1MHZ 方波输出图:



1KHZ 方波输出图:



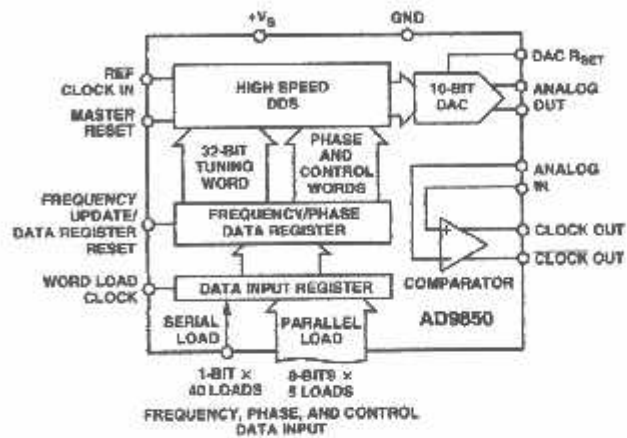
<http://www.eimodule.com>

FEATURES

125 MHz Clock Rate
On-Chip High Performance DAC and High Speed Comparator
DAC SFDR > 50 dB @ 40 MHz A_{OUT}
32-Bit Frequency Tuning Word
Simplified Control Interface: Parallel Byte or Serial Loading Format
Phase Modulation Capability
3.3 V or 5 V Single-Supply Operation
Low Power: 380 mW @ 125 MHz (5 V)
155 mW @ 110 MHz (3.3 V)
Power-Down Function
Ultrasmall 28-Lead SSOP Packaging

APPLICATIONS

Frequency/Phase—Agile Sine Wave Synthesis
Clock Recovery and Locking Circuitry for Digital Communications
Digitally Controlled ADC Encode Generator
Agile Local Oscillator Applications

FUNCTIONAL BLOCK DIAGRAM

GENERAL DESCRIPTION

The AD9850 is a highly integrated device that uses advanced DDS technology coupled with an internal high speed, high performance D/A converter and comparator to form a complete, digitally programmable frequency synthesizer and clock generator function. When referenced to an accurate clock source, the AD9850 generates a spectrally pure, frequency/phase programmable, analog output sine wave. This sine wave can be used directly as a frequency source, or it can be converted to a square wave for agile-clock generator applications. The AD9850's innovative high speed DDS core provides a 32-bit frequency tuning word, which results in an output tuning resolution of 0.0291 Hz for a 125 MHz reference clock input. The AD9850's circuit architecture allows the generation of output frequencies of up to one-half the reference clock frequency (or 62.5 MHz), and the output frequency can be digitally changed (asynchronously) at a rate of up to 23 million new frequencies per second. The device also provides five bits of digitally controlled phase modulation, which enables phase shifting of its output in increments of 180°, 90°, 45°, 22.5°,

11.25°, and any combination thereof. The AD9850 also contains a high speed comparator that can be configured to accept the (externally) filtered output of the DAC to generate a low jitter square wave output. This facilitates the device's use as an agile clock generator function.

The frequency tuning, control, and phase modulation words are loaded into the AD9850 via a parallel byte or serial loading format. The parallel load format consists of five iterative loads of an 8-bit control word (byte). The first byte controls phase modulation, power-down enable, and loading format; Bytes 2 to 5 comprise the 32-bit frequency tuning word. Serial loading is accomplished via a 40-bit serial data stream on a single pin. The AD9850 Complete DDS uses advanced CMOS technology to provide this breakthrough level of functionality and performance on just 155 mW of power dissipation (3.3 V supply).

The AD9850 is available in a space-saving 28-lead SSOP, surface-mount package. It is specified to operate over the extended industrial temperature range of -40°C to +85°C.

REV. H

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
 Tel: 781/329-4700 www.analog.com
 Fax: 781/326-8703 © 2004 Analog Devices, Inc. All rights reserved.

AD9850—SPECIFICATIONS ($V_S = 5\text{ V} \pm 5\%$ except as noted, $R_{SET} = 3.9\text{ k}\Omega$)

Parameter	Temp	Test Level	AD9850BRS			Unit
			Min	Typ	Max	
LOCK INPUT CHARACTERISTICS						
Frequency Range						
5 V Supply	Full	IV	1		125	MHz
3.3 V Supply	Full	IV	1		110	MHz
Pulse Width High/Low						
5 V Supply	25°C	IV	3.2			ns
3.3 V Supply	25°C	IV	4.1			ns
AC OUTPUT CHARACTERISTICS						
Full-Scale Output Current						
$R_{SET} = 3.9\text{ k}\Omega$	25°C	V		10.24		mA
$R_{SET} = 1.95\text{ k}\Omega$	25°C	V		20.48		mA
Gain Error	25°C	I	-10		+10	% FS
Gain Temperature Coefficient	Full	V		150		ppm/°C
Output Offset	25°C	I			10	μA
Output Offset Temperature Coefficient	Full	V		50		nA/°C
Differential Nonlinearity	25°C	I		0.5	0.75	LSB
Integral Nonlinearity	25°C	I		0.5	1	LSB
Output Slew Rate (50 Ω , 2 pF Load)	25°C	V		400		V/ μs
Output Impedance	25°C	IV	50	120		k Ω
Output Capacitance	25°C	IV			8	pF
Voltage Compliance	25°C	I			1.5	V
Spurious-Free Dynamic Range (SFDR)						
Wideband (Nyquist Bandwidth)						
1 MHz Analog Out	25°C	IV	63	72		dBc
20 MHz Analog Out	25°C	IV	50	58		dBc
40 MHz Analog Out	25°C	IV	46	54		dBc
Narrowband						
40.13579 MHz \pm 50 kHz	25°C	IV		80		dBc
40.13579 MHz \pm 200 kHz	25°C	IV		77		dBc
4.513579 MHz \pm 50 kHz/20.5 MHz CLK	25°C	IV		84		dBc
4.513579 MHz \pm 200 kHz/20.5 MHz CLK	25°C	IV		84		dBc
COMPARATOR INPUT CHARACTERISTICS						
Input Capacitance	25°C	V		3		pF
Input Resistance	25°C	IV	500			k Ω
Input Current	25°C	I	-12		+12	μA
Input Voltage Range	25°C	IV	0		V_{DD}	V
Comparator Offset*	Full	VI	30		30	mV
COMPARATOR OUTPUT CHARACTERISTICS						
Logic 1 Voltage 5 V Supply	Full	VI	4.8			V
Logic 1 Voltage 3.3 V Supply	Full	VI	3.1			V
Logic 0 Voltage	Full	VI			0.4	V
Propagation Delay, 5 V Supply (15 pF Load)	25°C	V		5.5		ns
Propagation Delay, 3.3 V Supply (15 pF Load)	25°C	V		7		ns
Rise/Fall Time, 5 V Supply (15 pF Load)	25°C	V		3		ns
Rise/Fall Time, 3.3 V Supply (15 pF Load)	25°C	V		3.5		ns
Output Jitter (p-p)	25°C	V		80		ps
LOCK OUTPUT CHARACTERISTICS						
Clock Output Duty Cycle (Clk Gen. Config.)	25°C	IV		50 \pm 10		%

Parameter	Temp	Test Level	AD9850BRS			Unit
			Min	Typ	Max	
CMOS LOGIC INPUTS (Including CLKIN)						
Logic 1 Voltage, 5 V Supply	25°C	I	3.5			V
Logic 1 Voltage, 3.3 V Supply	25°C	IV	2.4			V
Logic 0 Voltage	25°C	IV			0.8	V
Logic 1 Current	25°C	I			12	μA
Logic 0 Current	25°C	I			12	μA
Input Capacitance	25°C	V		3		pF
POWER SUPPLY (A_{OUT} = 1/3 CLKIN)						
+V_S Current @						
62.5 MHz Clock, 3.3 V Supply	Full	VI		30	48	mA
110 MHz Clock, 3.3 V Supply	Full	VI		47	60	mA
62.5 MHz Clock, 5 V Supply	Full	VI		44	64	mA
125 MHz Clock, 5 V Supply	Full	VI		76	96	mA
P_{DISS} @						
62.5 MHz Clock, 3.3 V Supply	Full	VI		100	160	mW
110 MHz Clock, 3.3 V Supply	Full	VI		155	200	mW
62.5 MHz Clock, 5 V Supply	Full	VI		220	320	mW
125 MHz Clock, 5 V Supply	Full	VI		380	480	mW
P_{DISS} Power-Down Mode						
5 V Supply	Full	V		30		mW
3.3 V Supply	Full	V		10		mW

*Tested by measuring output duty cycle variation.
Specifications subject to change without notice.

TIMING CHARACTERISTICS* (V_S = 5 V ± 5% except as noted, R_{SET} = 3.9 kΩ)

Parameter	Temp	Test Level	AD9850BRS			Unit
			Min	Typ	Max	
t _{DS} (Data Setup Time)	Full	IV	3.5			ns
t _{DH} (Data Hold Time)	Full	IV	3.5			ns
t _{WH} (W_CLK Minimum Pulse Width High)	Full	IV	3.5			ns
t _{WL} (W_CLK Minimum Pulse Width Low)	Full	IV	3.5			ns
t _{WD} (W_CLK Delay after FQ_UD)	Full	IV	7.0			ns
t _{CD} (CLKIN Delay after FQ_UD)	Full	IV	3.5			ns
t _{FH} (FQ_UD High)	Full	IV	7.0			ns
t _{FL} (FQ_UD Low)	Full	IV	7.0			ns
t _{CF} (Output Latency from FQ_UD)						
Frequency Change	Full	IV	18			CLKIN Cycles
Phase Change	Full	IV	13			CLKIN Cycles
t _{UD} (FQ_UD Minimum Delay after W_CLK)	Full	IV	7.0			ns
t _{RR} (CLKIN Delay after RESET Rising Edge)	Full	IV	3.5			ns
t _{RL} (RESET Falling Edge after CLKIN)	Full	IV	3.5			ns
t _{RS} (Minimum RESET Width)	Full	IV	5			CLKIN Cycles
t _{OL} (RESET Output Latency)	Full	IV	13			CLKIN Cycles
t _{RR} (Recovery from RESET)	Full	IV	2			CLKIN Cycles
Wake-Up Time from Power-Down Mode	25°C	V		5		μs

*Control functions are asynchronous with CLKIN.
Specifications subject to change without notice.

D9850

ABSOLUTE MAXIMUM RATINGS*

Maximum Junction Temperature	150°C
Maximum Input Voltage	6 V
Digital Inputs	-0.7 V to +V _{DD}
Digital Output Continuous Current	5 mA
AC Output Current	30 mA
Storage Temperature	-65°C to +150°C
Operating Temperature	-40°C to +85°C
Soldering Temperature (Soldering 10 sec)	300°C
θ _{JA} Thermal Impedance	82°C/W

Absolute maximum ratings are limiting values, to be applied individually, and depend on which the serviceability of the circuit may be impaired. Functional reliability under any of these conditions is not necessarily implied. Exposure of absolute maximum rating conditions for extended periods of time may affect device reliability.

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the AD9850 features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

Application Note: Users are cautioned not to apply digital input signals prior to power-up of this device. Doing so may result in a latch-up condition.

EXPLANATION OF TEST LEVELS

Test Level

- I 100% Production Tested.
- III Sample Tested Only.
- IV Parameter is guaranteed by design and characterization testing.
- V Parameter is a typical value only.
- VI All devices are 100% production tested at 25°C. 100% production tested at temperature extremes for military temperature devices; guaranteed by design and characterization testing for industrial devices.



ORDERING GUIDE

Model	Temperature Range	Package Description	Package Option
D9850BRS	-40°C to +85°C	Shrink Small Outline Package (SSOP)	RS-28
D9850BRS-REEL	-40°C to +85°C	Shrink Small Outline Package (SSOP)	RS-28
D9850BRSZ*	-40°C to +85°C	Shrink Small Outline Package (SSOP)	RS-28
D9850BRSZ-REEL*	-40°C to +85°C	Shrink Small Outline Package (SSOP)	RS-28
D9850/CGPCB		Evaluation Board Clock Generator	
D9850/FSPCB		Evaluation Board Frequency Synthesizer	

* = Pb-free part.

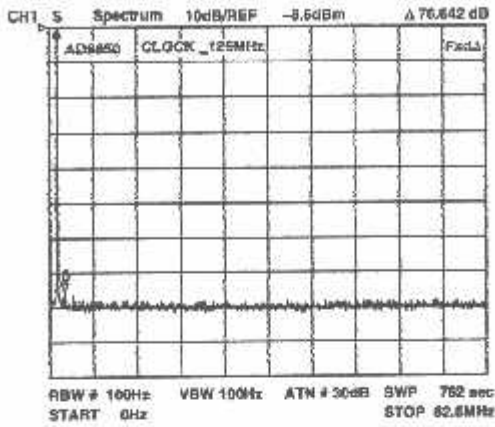
PIN CONFIGURATION



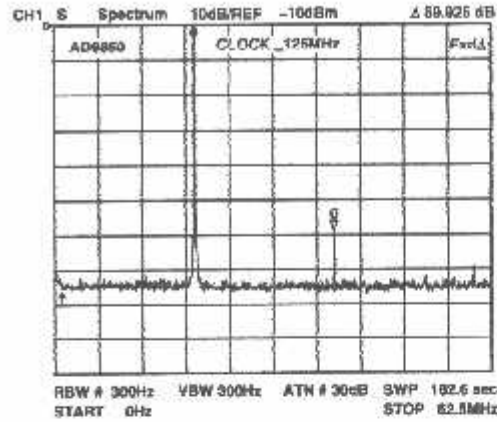
Table I. PIN FUNCTION DESCRIPTIONS

Pin No.	Mnemonic	Function
4 to 1, 28 to 25	D0 to D7	8-Bit Data Input. This is the 8-bit data port for iteratively loading the 32-bit frequency and the 8-bit phase/control word. D7 = MSB; D0 = LSB. D7 (Pin 25) also serves as the input pin for the 40-bit serial data-word.
5, 24	DGND	Digital Ground. These are the ground return leads for the digital circuitry.
6, 23	DVDD	Supply Voltage Leads for Digital Circuitry.
7	W_CLK	Word Load Clock. This clock is used to load the parallel or serial frequency/phase/control words.
8	FQ_UD	Frequency Update. On the rising edge of this clock, the DDS updates to the frequency (or phase) loaded in the data input register; it then resets the pointer to Word 0.
9	CLKIN	Reference Clock Input. This may be a continuous CMOS-level pulse train or sine input biased at 1/2 V supply. The rising edge of this clock initiates operation.
10, 19	AGND	Analog Ground. These leads are the ground return for the analog circuitry (DAC and comparator).
11, 18	AVDD	Supply Voltage for the Analog Circuitry (DAC and Comparator).
12	R _{SET}	DAC's External R _{SET} Connection. This resistor value sets the DAC full-scale output current. For normal applications ($F_S I_{OUT} = 10 \text{ mA}$), the value for R _{SET} is 3.9 k Ω connected to ground. The R _{SET} /I _{OUT} relationship is $I_{OUT} = 32 (1.248 \text{ V}/R_{SET})$.
13	QOUTB	Output Complement. This is the comparator's complement output.
14	QOUT	Output True. This is the comparator's true output.
15	VINN	Inverting Voltage Input. This is the comparator's negative input.
16	VINP	Noninverting Voltage Input. This is the comparator's positive input.
17	DACBL (NC)	DAC Baseline. This is the DAC baseline voltage reference; this lead is internally bypassed and should normally be considered a no connect for optimum performance.
20	IOUTB	Complementary Analog Output of the DAC.
21	IOUT	Analog Current Output of the DAC.
22	RESET	Reset. This is the master reset function; when set high, it clears all registers (except the input register), and the DAC output goes to cosine 0 after additional clock cycles—see Figure 7.

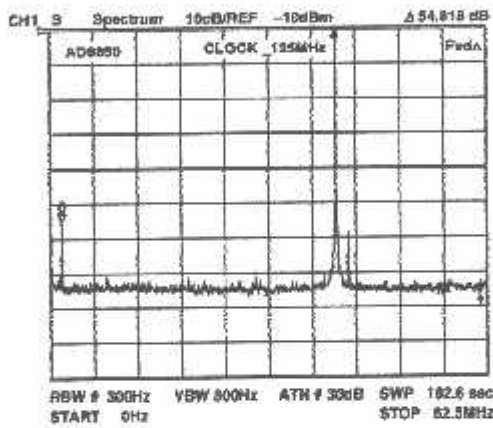
D9850—Typical Performance Characteristics



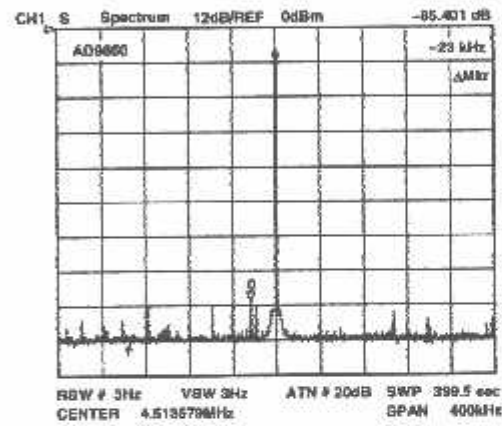
TPC 1. SFDR, CLKIN = 125 MHz/ f_{OUT} = 1 MHz



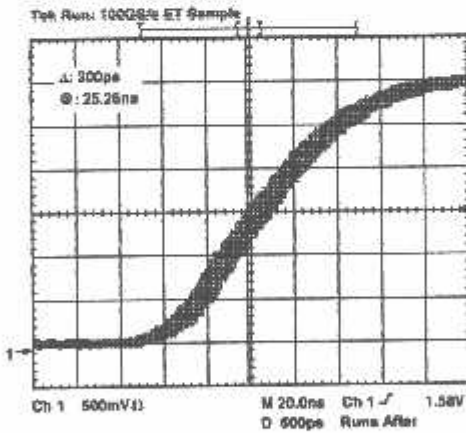
TPC 4. SFDR, CLKIN = 125 MHz/ f_{OUT} = 20 MHz



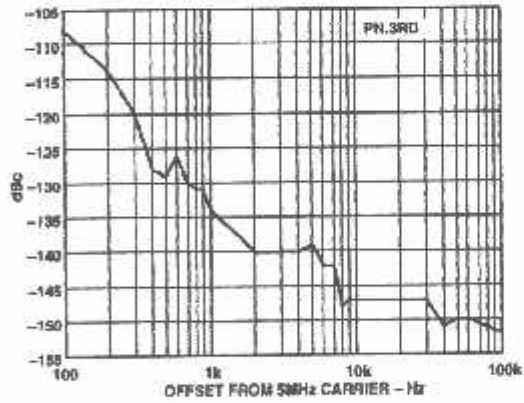
TPC 2. SFDR, CLKIN = 125 MHz/ f_{OUT} = 41 MHz



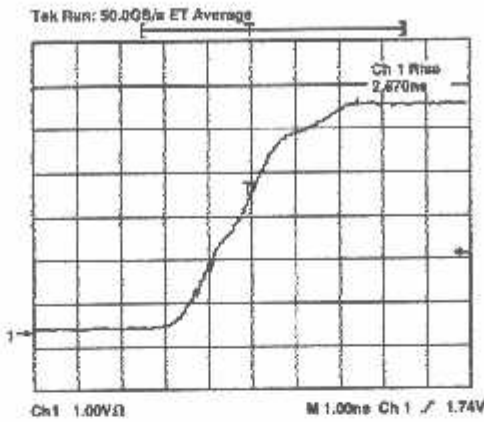
TPC 5. SFDR, CLKIN = 20.5 MHz/ f_{OUT} = 4.5 MHz



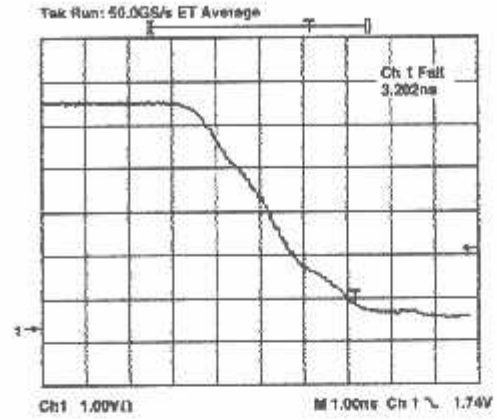
TPC 3. Typical Comparator Output Jitter, AD9850 Configured as Clock Generator with 42 MHz LP Filter (40 MHz A_{OUT} /125 MHz CLKIN)



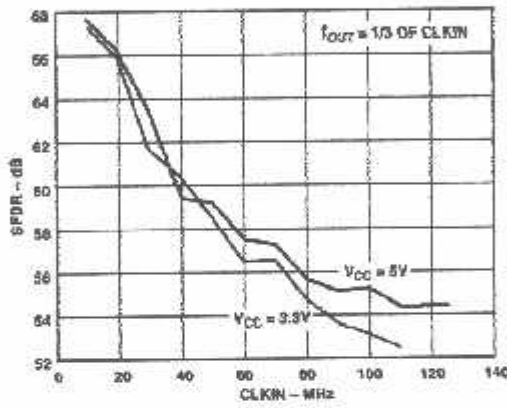
TPC 6. Output Residual Phase Noise (5 MHz A_{OUT} /125 MHz CLKIN)



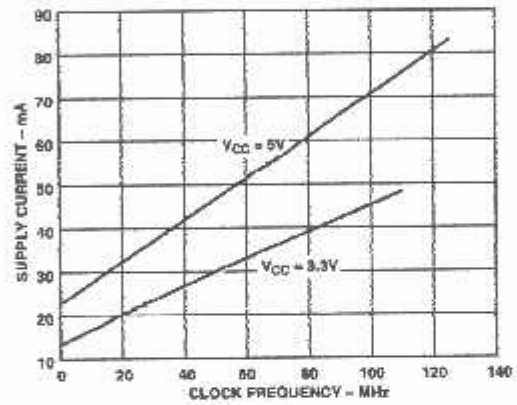
TPC 7. Comparator Output Rise Time (5 V Supply/15 pF Load)



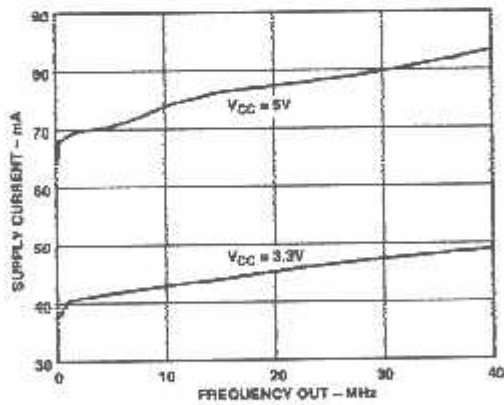
TPC 10. Comparator Output Fall Time (5 V Supply/15 pF Load)



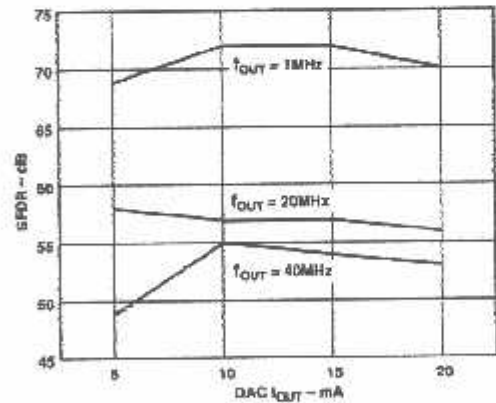
TPC 8. SFDR vs. CLKIN Frequency ($A_{OUT} = 1/3$ of CLKIN)



TPC 11. Supply Current vs. CLKIN Frequency ($A_{OUT} = 1/3$ of CLKIN)



TPC 9. Supply Current vs. A_{OUT} Frequency (CLKIN = 125/110 MHz for 5 V/3.3 V Plot)



TPC 12. SFDR vs. DAC I_{OUT} ($A_{OUT} = 1/3$ of CLKIN)

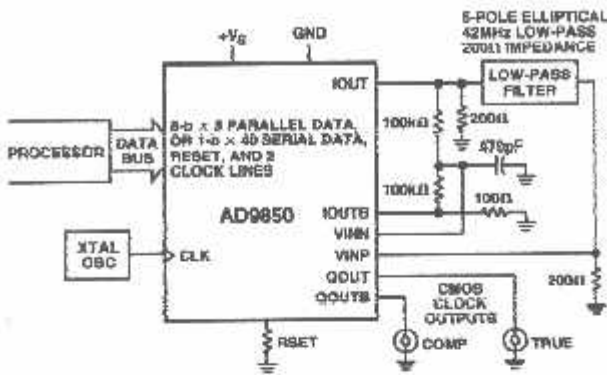


Figure 1. Basic AD9850 Clock Generator Application with Low-Pass Filter

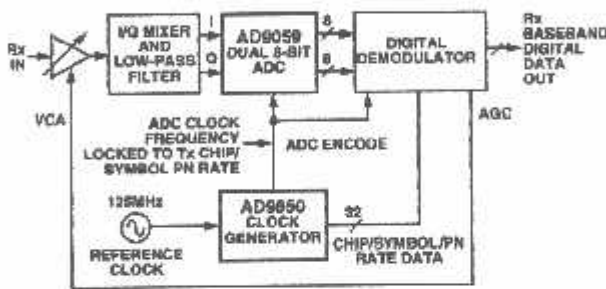


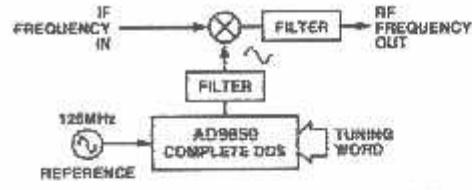
Figure 2. AD9850 Clock Generator Application in a Spread-Spectrum Receiver

THEORY OF OPERATION AND APPLICATION

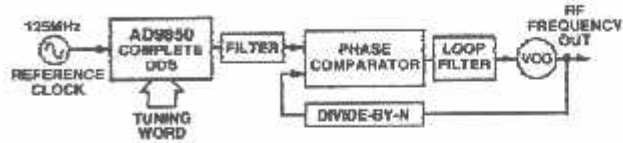
The AD9850 uses direct digital synthesis (DDS) technology, in the form of a numerically controlled oscillator, to generate a frequency/phase-agile sine wave. The digital sine wave is converted to analog form via an internal 10-bit high speed D/A converter, and an on-board high speed comparator is provided to translate the analog sine wave into a low jitter TTL/CMOS compatible output square wave. DDS technology is an innovative circuit architecture that allows fast and precise manipulation of its output frequency under digital control. DDS also enables very high resolution in the incremental selection of output frequency; the AD9850 allows an output frequency resolution of 0.0291 Hz with a 125 MHz reference clock applied. The AD9850's output waveform is phase continuous when changed.

The basic functional block diagram and signal flow of the AD9850 configured as a clock generator is shown in Figure 4.

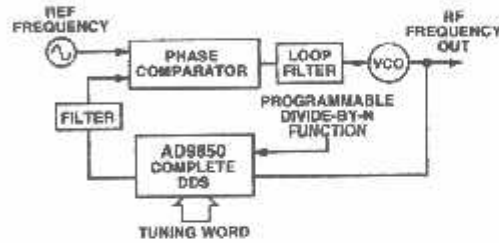
The DDS circuitry is basically a digital frequency divider function whose incremental resolution is determined by the frequency of the reference clock divided by the 2^N number of bits in the tuning word. The phase accumulator is a variable-modulus counter that increments the number stored in it each time it receives a clock pulse. When the counter overflows, it wraps around, making the phase accumulator's output contiguous.



3a. Frequency/Phase-Agile Local Oscillator



3b. Frequency/Phase-Agile Reference for PLL



3c. Digitally-Programmable Divide-by-N Function in PLL

Figure 3. AD9850 Complete DDS Synthesizer in Frequency Up-Conversion Applications

The frequency tuning word sets the modulus of the counter, which effectively determines the size of the increment (Δ Phase) that is added to the value in the phase accumulator on the next clock pulse. The larger the added increment, the faster the accumulator overflows, which results in a higher output frequency. The AD9850 uses an innovative and proprietary algorithm that mathematically converts the 14-bit truncated value of the phase accumulator to the appropriate COS value. This unique algorithm uses a much reduced ROM look-up table and DSP techniques to perform this function, which contributes to the small size and low power dissipation of the AD9850. The relationship of the output frequency, reference clock, and tuning word of the AD9850 is determined by the formula

$$f_{OUT} = (\Delta \text{ Phase} \times CLKIN) / 2^{32}$$

where:

- Δ Phase is the value of the 32-bit tuning word.
- CLKIN is the input reference clock frequency in MHz.
- f_{OUT} is the frequency of the output signal in MHz.

The digital sine wave output of the DDS block drives the internal high speed 10-bit D/A converter that reconstructs the sine wave in analog form. This DAC has been optimized for dynamic performance and low glitch energy as manifested in the low jitter performance of the AD9850. Because the output of the

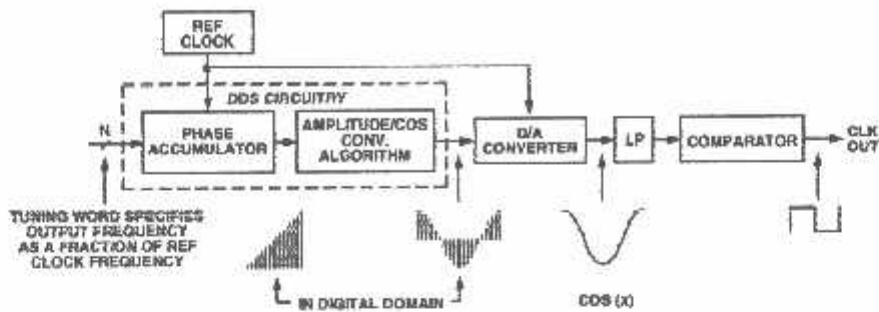


Figure 4. Basic DDS Block Diagram and Signal Flow of AD9850

AD9850 is a sampled signal, its output spectrum follows the Nyquist sampling theorem. Specifically, its output spectrum contains the fundamental plus aliased signals (images) that occur at multiples of the reference clock frequency \pm the selected output frequency. A graphical representation of the sampled spectrum, with aliased images, is shown in Figure 5.

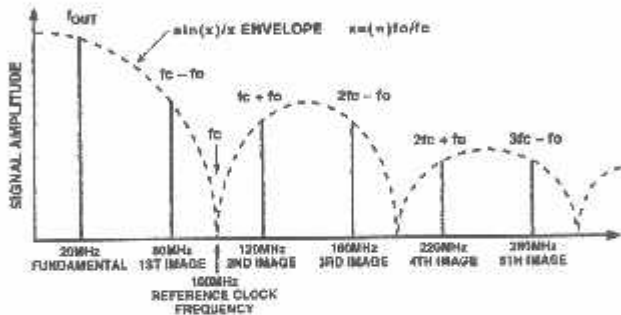


Figure 5. Output Spectrum of a Sampled Signal

In this example, the reference clock is 100 MHz and the output frequency is set to 20 MHz. As can be seen, the aliased images are very prominent and of a relatively high energy level as determined by the $\sin(x)/x$ roll-off of the quantized D/A converter output. In fact, depending on the f_o /reference clock relationship, the first aliased image can be on the order of -3 dB below the fundamental. A low-pass filter is generally placed between the output of the D/A converter and the input of the comparator to further suppress the effects of aliased images. Obviously, consideration must be given to the relationship of the selected output frequency and the reference clock frequency to avoid unwanted (and unexpected) output anomalies.

To apply the AD9850 as a clock generator, limit the selected output frequency to $<33\%$ of reference clock frequency, and thereby avoid generating aliased signals that fall within, or close to, the output band of interest (generally dc-selected output frequency). This practice eases the complexity (and cost) of the external filter requirement for the clock generator application.

The reference clock frequency of the AD9850 has a minimum limitation of 1 MHz. The device has internal circuitry that senses when the minimum clock rate threshold has been exceeded

and automatically places itself in the power-down mode. When in this state, if the clock frequency again exceeds the threshold, the device resumes normal operation. This shutdown mode prevents excessive current leakage in the dynamic registers of the device.

The D/A converter output and comparator inputs are available as differential signals that can be flexibly configured in any manner desired to achieve the objectives of the end system. The typical application of the AD9850 is with single-ended output input analog signals, a single low-pass filter, and the generation of the comparator reference midpoint from the differential DAC output as shown in Figure 1.

Programming the AD9850

The AD9850 contains a 40-bit register that is used to program the 32-bit frequency control word, the 5-bit phase modulation word, and the power-down function. This register can be loaded in a parallel or serial mode.

In the parallel load mode, the register is loaded via an 8-bit bus; the full 40-bit word requires five iterations of the 8-bit word. The W_CLK and FQ_UD signals are used to address and load the registers. The rising edge of FQ_UD loads the (up to) 40-bit control data-word into the device and resets the address pointer to the first register. Subsequent W_CLK rising edges load the 8-bit data on words [7:0] and move the pointer to the next register. After f_{res} loads, W_CLK edges are ignored until either a reset or an FQ_UD rising edge resets the address pointer to the first register.

In serial load mode, subsequent rising edges of W_CLK shift the 1-bit data on Pin 25 (D7) through the 40 bits of programming information. After 40 bits are shifted through, an FQ_UD pulse is required to update the output frequency (or phase).

The function assignments of the data and control words are shown in Table III; the detailed timing sequence for updating the output frequency and/or phase, resetting the device, and powering up/down, are shown in the timing diagrams of Figures 6 through 12.

Note: There are specific control codes, used for factory test purposes, that render the AD9850 temporarily inoperable. The user must take deliberate precaution to avoid inputting the codes listed in Table II.

Table II. Factory Reserved Internal Test Control Codes

Loading Format	Factory Reserved Codes
Parallel	1) W0 = XXXXXX10 2) W0 = XXXXXX01
Serial	1) W32 = 1; W33 = 0 2) W32 = 0; W33 = 1 3) W32 = 1; W33 = 1

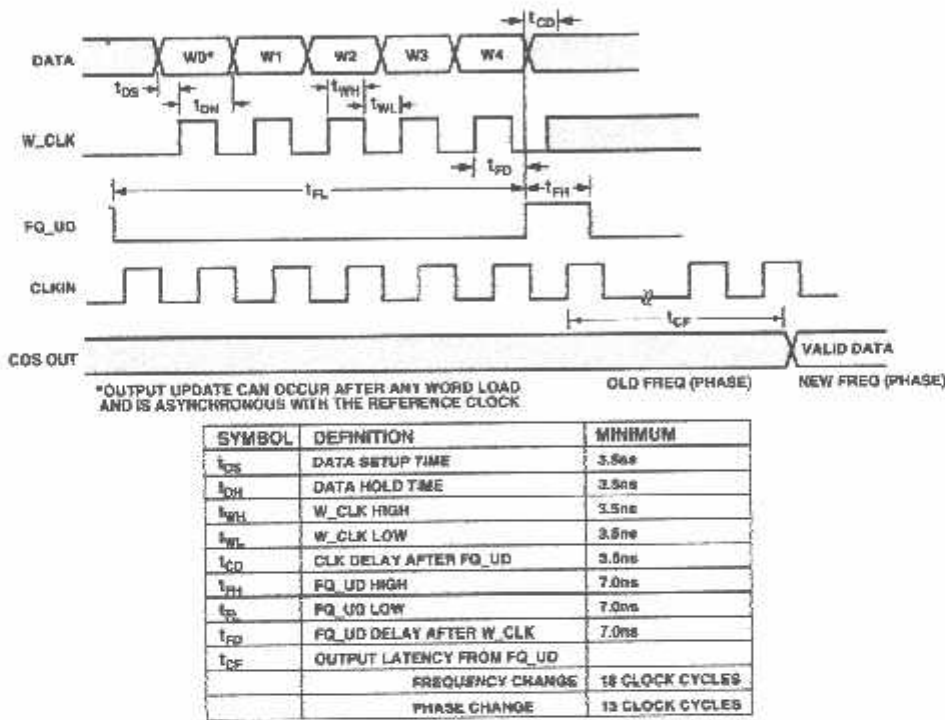
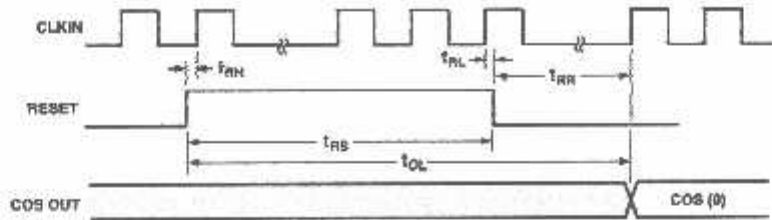


Figure 6. Parallel Load Frequency/Phase Update Timing Sequence

Table III. 8-Bit Parallel Load Data/Control Word Functional Assignment

ord	Data[7]	Data[6]	Data[5]	Data[4]	Data[3]	Data[2]	Data[1]	Data[0]
1	Phase-b4 (MSB)	Phase-b3	Phase-b2	Phase-b1	Phase-b0 (LSB)	Power-Down	Control	Control
1	Freq-b31 (MSB)	Freq-b30	Freq-b29	Freq-b28	Freq-b27	Freq-b26	Freq-b25	Freq-b24
2	Freq-b23	Freq-b22	Freq-b21	Freq-b20	Freq-b19	Freq-b18	Freq-b17	Freq-b16
3	Freq-b15	Freq-b14	Freq-b13	Freq-b12	Freq-b11	Freq-b10	Freq-b9	Freq-b8
4	Freq-b7	Freq-b6	Freq-b5	Freq-b4	Freq-b3	Freq-b2	Freq-b1	Freq-b0 (LSB)



NOTE: THE TIMING DIAGRAM ABOVE SHOWS THE MINIMAL AMOUNT OF RESET TIME NEEDED BEFORE WRITING TO THE DEVICE. HOWEVER, THE MASTER RESET DOES NOT HAVE TO BE SYNCHRONOUS WITH THE CLKIN IF THE MINIMAL TIME IS NOT REQUIRED.

SYMBOL	DEFINITION	MINIMUM
t_{FFH}	CLK DELAY AFTER RESET RISING EDGE	3.5ns
t_{RL}	RESET FALLING EDGE AFTER CLK	3.5ns
t_{RR}	RECOVERY FROM RESET	2 CLK CYCLES
t_{RS}	MINIMUM RESET WIDTH	5 CLK CYCLES
t_{CL}	RESET OUTPUT LATENCY	10 CLK CYCLES

- RESULTS OF RESET:
- FREQUENCY/PHASE REGISTER SET TO 0
 - ADDRESS POINTER RESET TO W0
 - POWER-DOWN BIT RESET TO 0
 - DATA INPUT REGISTER UNEFFECTED

Figure 7. Master Reset Timing Sequence

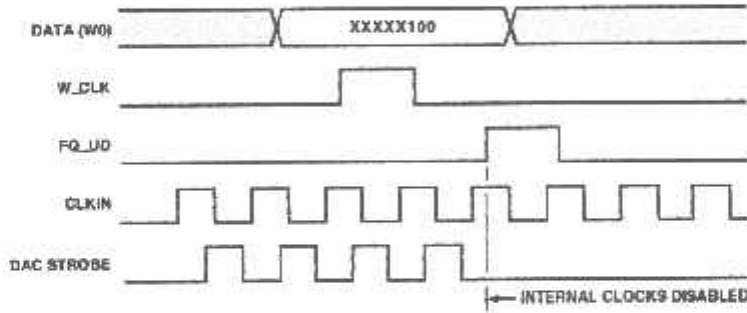


Figure 8. Parallel Load Power-Down Sequence/Internal Operation

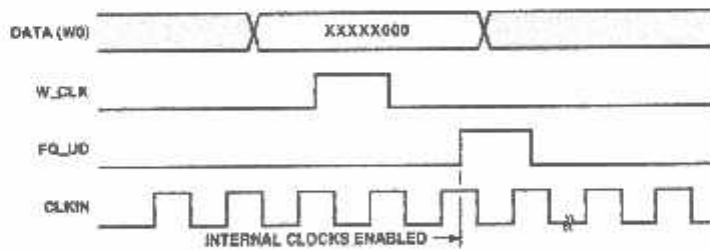


Figure 9. Parallel Load Power-Up Sequence/Internal Operation

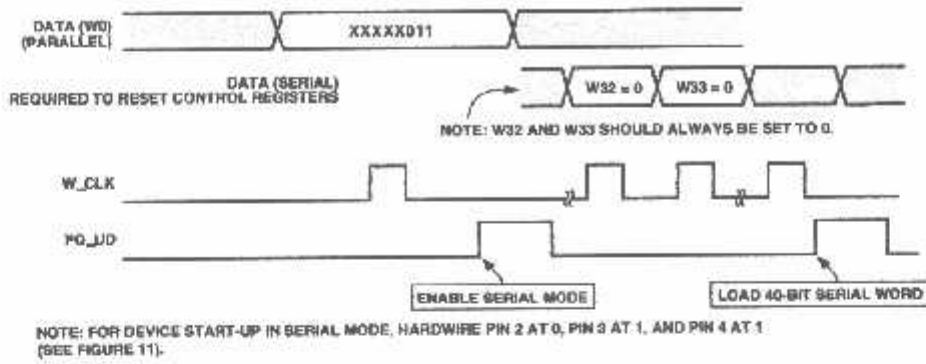


Figure 10. Serial Load Enable Sequence

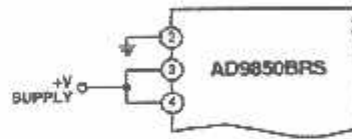


Figure 11. Pins 2 to 4 Connection for Default Serial Mode Operation

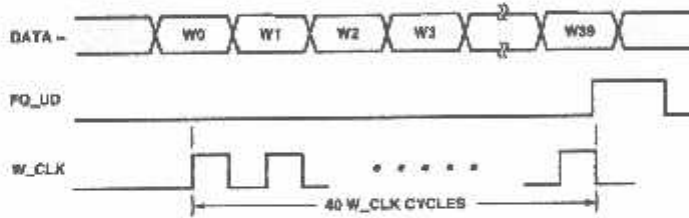


Figure 12. Serial Load Frequency/Phase Update Sequence

Table IV. 40-Bit Serial Load Word Function Assignment

W0	Freq-b0 (LSB)	W14	Freq-b14	W28	Freq-b28
W1	Freq-b1	W15	Freq-b15	W29	Freq-b29
W2	Freq-b2	W16	Freq-b16	W30	Freq-b30
W3	Freq-b3	W17	Freq-b17	W31	Freq-b31 (MSB)
W4	Freq-b4	W18	Freq-b18	W32	Control
W5	Freq-b5	W19	Freq-b19	W33	Control
W6	Freq-b6	W20	Freq-b20	W34	Power-Down
W7	Freq-b7	W21	Freq-b21	W35	Phase-b0 (LSB)
W8	Freq-b8	W22	Freq-b22	W36	Phase-b1
W9	Freq-b9	W23	Freq-b23	W37	Phase-b2
W10	Freq-b10	W24	Freq-b24	W38	Phase-b3
W11	Freq-b11	W25	Freq-b25	W39	Phase-b4 (MSB)
W12	Freq-b12	W26	Freq-b26		
W13	Freq-b13	W27	Freq-b27		

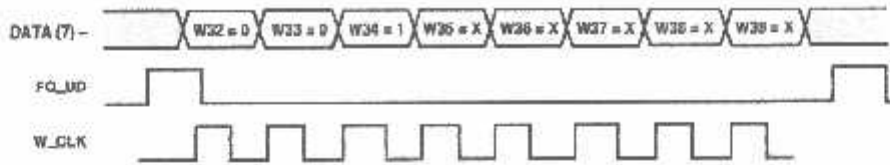


Figure 13. Serial Load Power-Down Sequence

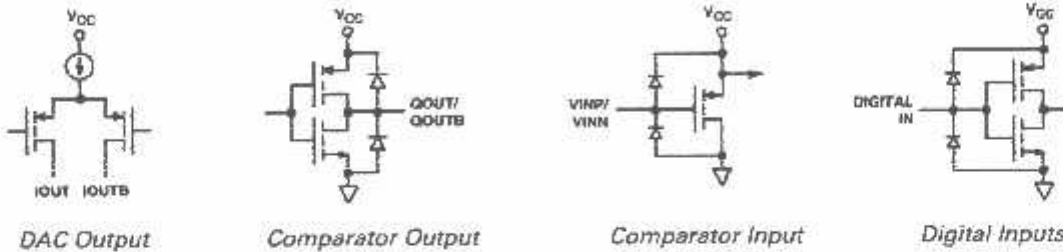


Figure 14. AD9850 I/O Equivalent Circuits

PCB LAYOUT INFORMATION

The AD9850/CGPCB and AD9850/FSPCB evaluation boards (Figures 15 through 18) represent typical implementations of the AD9850 and exemplify the use of high frequency/high resolution design and layout practices. The printed circuit board that contains the AD9850 should be a multilayer board that allows dedicated power and ground planes. The power and ground planes should be free of etched traces that cause discontinuities in the planes. It is recommended that the top layer of the multilayer board also contain an interspatial ground plane, which makes ground available for surface-mount devices. If separate analog and digital system ground planes exist, they should be connected together at the AD9850 for optimum results.

Avoid running digital lines under the device because these couple noise onto the die. The power supply lines to the AD9850 should use as large a track as possible to provide a low impedance path and reduce the effects of glitches on the power supply line. Fast switching signals like clocks should be shielded with ground to avoid radiating noise to other sections of the board. Avoid crossover of digital and analog signal paths. Traces on opposite sides of the board should run at right angles to each other. This reduces the effects of feedthrough through the circuit board. Use microstrip techniques where possible.

Good decoupling is also an important consideration. The analog (AVDD) and digital (DVDD) supplies to the AD9850 are independent and separately pinned out to minimize coupling between analog and digital sections of the device. All analog and digital supplies should be decoupled to AGND and DGND, respectively, with high quality ceramic capacitors. To achieve best performance from the decoupling capacitors, they should be placed as close as possible to the device, ideally right up against the device. In systems where a common supply is used to drive both the AVDD and DVDD supplies of the AD9850, it is recommended that the system's AVDD supply be used.

Analog Devices, Inc. applications engineering support is available to answer additional questions on grounding and PCB layout. Call 1-800-ANALOGD or contact us at www.analog.com/dda.

Evaluation Boards

Two versions of evaluation boards are available for the AD9850, which facilitate the implementation of the device for bench-top analysis and serve as a reference for PCB layout. The AD9850/FSPCB is used in applications where the device is used primarily as a frequency synthesizer. This version facilitates connection of the AD9850's internal D/A converter output to a 50 Ω spectrum analyzer input; the internal comparator on the AD9850 DUT is not enabled (see Figure 15 for an electrical schematic of AD9850/FSPCB). The AD9850/CGPCB is used in applications using the device in the clock generator mode. It connects the AD9850's DAC output to the internal comparator input via a single-ended, 42 MHz low-pass, 5-pole elliptical filter. This model facilitates the access of the AD9850's comparator output for evaluation of the device as a frequency- and phase-agile clock source (see Figure 17 for an electrical schematic of AD9850/CGPCB).

Both versions of the AD9850 evaluation board are designed to interface to the parallel printer port of a PC. The operating software runs under Microsoft® Windows® and provides a user-friendly and intuitive format for controlling the functionality and observing the performance of the device. The 3.5 inch floppy provided with the evaluation board contains an executable file that loads and displays the AD9850 function-selection screen. The evaluation board can be operated with 3.3 V or 5 V supplies. The evaluation boards are configured at the factory for an external reference clock input; if the on-board crystal clock source is used, remove R2.

D9850

D9850 Evaluation Board Instructions

Required Hardware/Software

IBM compatible computer operating in a Windows environment.
Printer port, 3.5 inch floppy drive, and Centronics compatible printer cable.

XTAL clock or signal generator—if using a signal generator, dc offset the signal to one-half the supply voltage and apply at least 3 V p-p signal across the 50 Ω (R2) input resistor. Remove R2 for high Z clock input.

AD9850 evaluation board software disk and AD9850/FSPCB or AD9850/CGPCB evaluation board.

5 V voltage supply.

Setup

Copy the contents of the AD9850 disk onto your hard drive (there are three files).

Connect the printer cable from your computer to the AD9850 evaluation board.

Apply power to AD9850 evaluation board. The AD9850 is powered separately from the connector marked DUT +V.

The AD9850 may be powered with 3.3 V to 5 V.

Connect external 50 Ω clock or remove R2 and apply a high Z input clock such as a crystal can oscillator.

Locate the file called 9850REV2.EXE and execute that program. Monitor should display a control panel to allow operation of the AD9850 evaluation board.

Verification

At the control panel, locate the box called COMPUTER I/O. Point to and click the selection marked LPT1 and then point to the TEST box and click. A message will appear telling users if their choice of output ports is correct. Choose other ports as necessary to achieve a correct setting. If they have trouble getting their computer to recognize any printer port, they should do the following: connect three 2 k Ω pull-up resistors from Pins 8, 9, and 7 of U3 to 5 V. This will assist weak printer port outputs in driving the heavy capacitance load of the printer cable. If troubles persist, try a different printer cable.

Click the MASTER RESET button with the mouse and click. This will reset the AD9850 to 0 Hz, 0° phase. The output should be a dc voltage equal to the full-scale output of the AD9850.

Locate the CLOCK box and place the cursor in the frequency box. Type in the clock frequency (in MHz) that the user will be applying to the AD9850. Click the LOAD button or press enter on the keyboard.

Move the cursor to the OUTPUT FREQUENCY box and type in the desired output frequency (in MHz). Click the LOAD button or press the enter key. The BUS MONITOR section of the control panel will show the 32-bit word that was loaded into the AD9850. Upon completion of this step, the AD9850 output should be active and outputting the user's frequency information.

Changing the output phase is accomplished by clicking on the down arrow in the OUTPUT PHASE DELAY box to make a selection and then clicking the LOAD button.

Other operational modes (frequency sweeping, sleep, serial input) are available to the user via keyboard/mouse control.

The AD9850/FSPCB provides access into and out of the on-chip comparator via test point pairs (each pair has an active input and a ground connection). The two active inputs are labeled TP1 and TP2. The unmarked hole next to each labeled test point is a ground connection. The two active outputs are labeled TP5 and TP6. Unmarked ground connections are adjacent to each of these test points.

The AD9850/CGPCB provides BNC inputs and outputs associated with the on-chip comparator and the on-board, fifth-order, 200 Ω input/output Z, elliptic, 45 MHz, low-pass filter. Jumping (soldering a wire) E1 to E2, E3 to E4, and E5 to E6 connects the on-board filter and the midpoint switching voltage to the comparator. Users may elect to insert their own filter and comparator threshold voltage by removing the jumpers and inserting a filter between J7 and J6 and then providing a threshold voltage at E1.

If users choose to use the XTAL socket to supply the clock to the AD9850, they must remove R2 (a 50 Ω chip resistor). The crystal oscillator must be either TTL or CMOS (preferably) compatible.

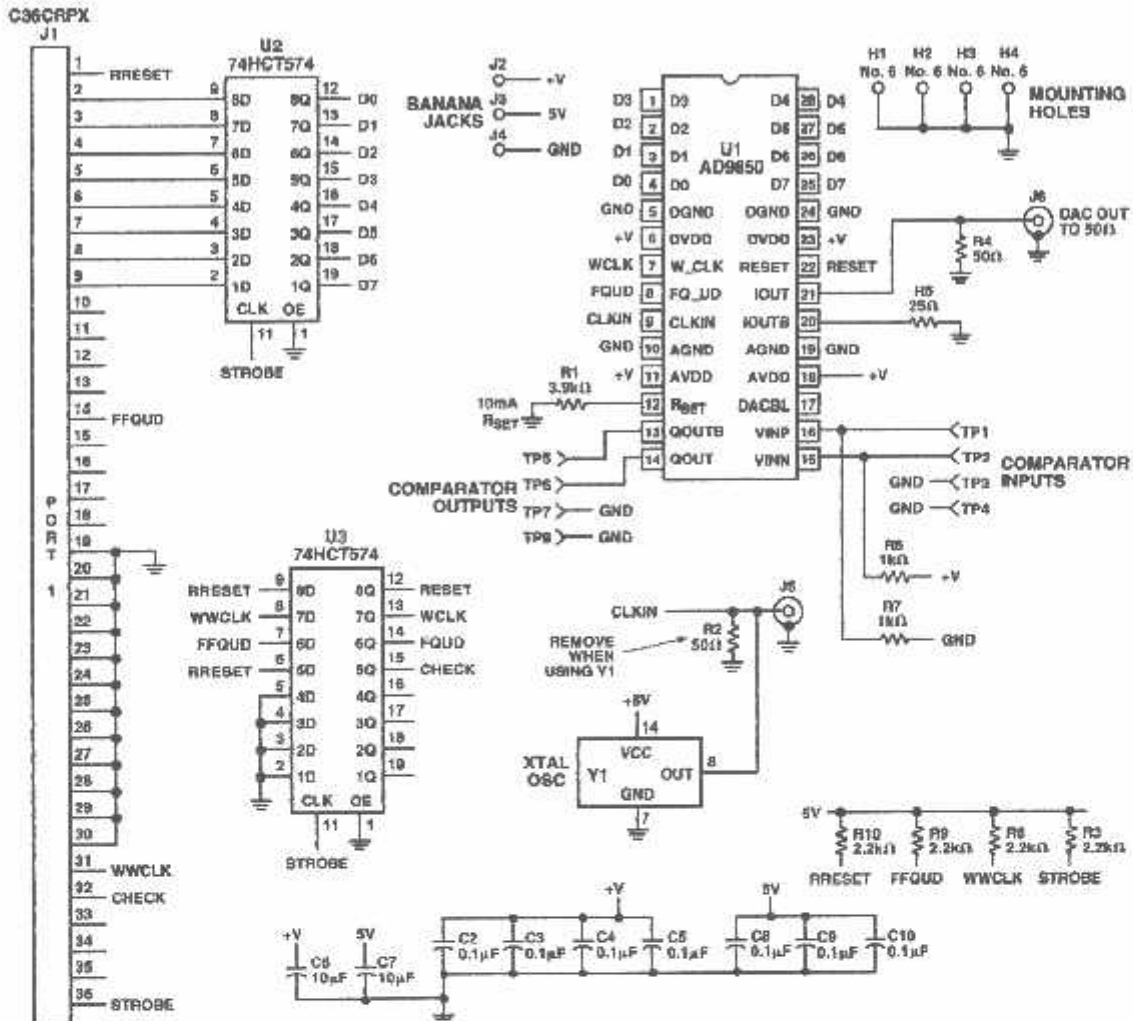


Figure 15. AD9850/FSPCB Electrical Schematic

COMPONENT LIST

Integrated Circuits

U1

AD9850BRS (28-Lead SSOP)

U2, U3

74HCT574 H-CMOS Octal Flip-Flop

Capacitors

C2 to C5, C8 to C10

0.1 μF Ceramic Chip Capacitor

C6, C7

10 μF Tantalum Chip Capacitor

Resistors

R1

3.9 kΩ Resistor

R2, R4

50 Ω Resistor

R3, R8, R9, R10

2.2 kΩ Resistor

R5

25 Ω Resistor

R6, R7

1 kΩ Resistor

Connectors

J1

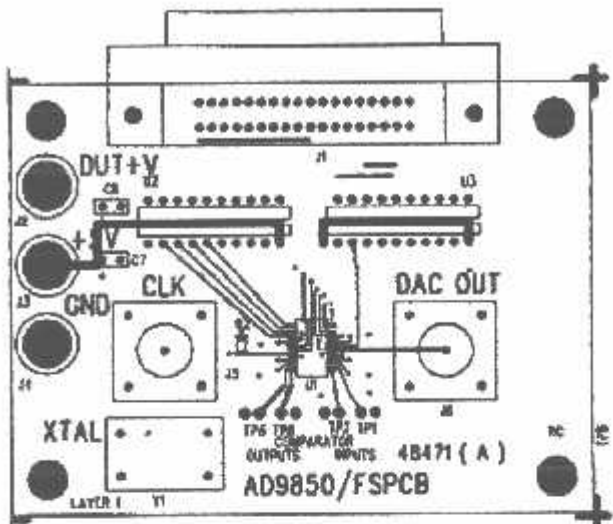
36-Pin D Connector

J2, J3, J4

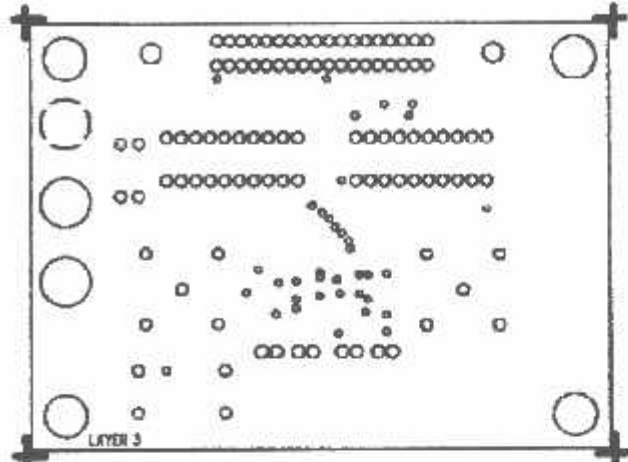
Banana Jack

J5, J6

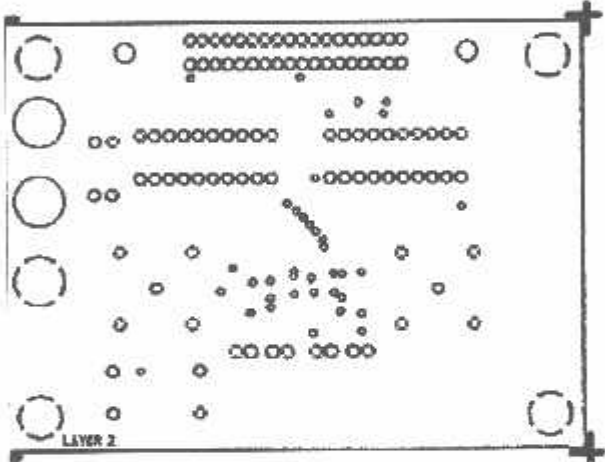
BNC Connector



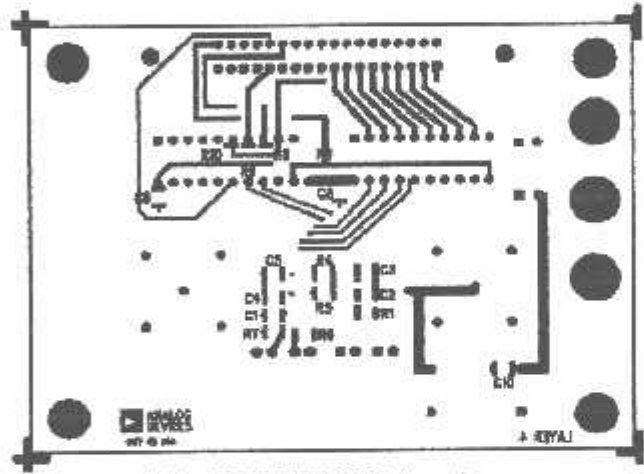
16a. AD9850/FSPCB Top Layer



16c. AD9850/FSPCB Power Plane



16b. AD9850/FSPCB Ground Plane



16d. AD9850/FSPCB Bottom Layer

Figure 16. AD9850/FSPCB Evaluation Board Layout

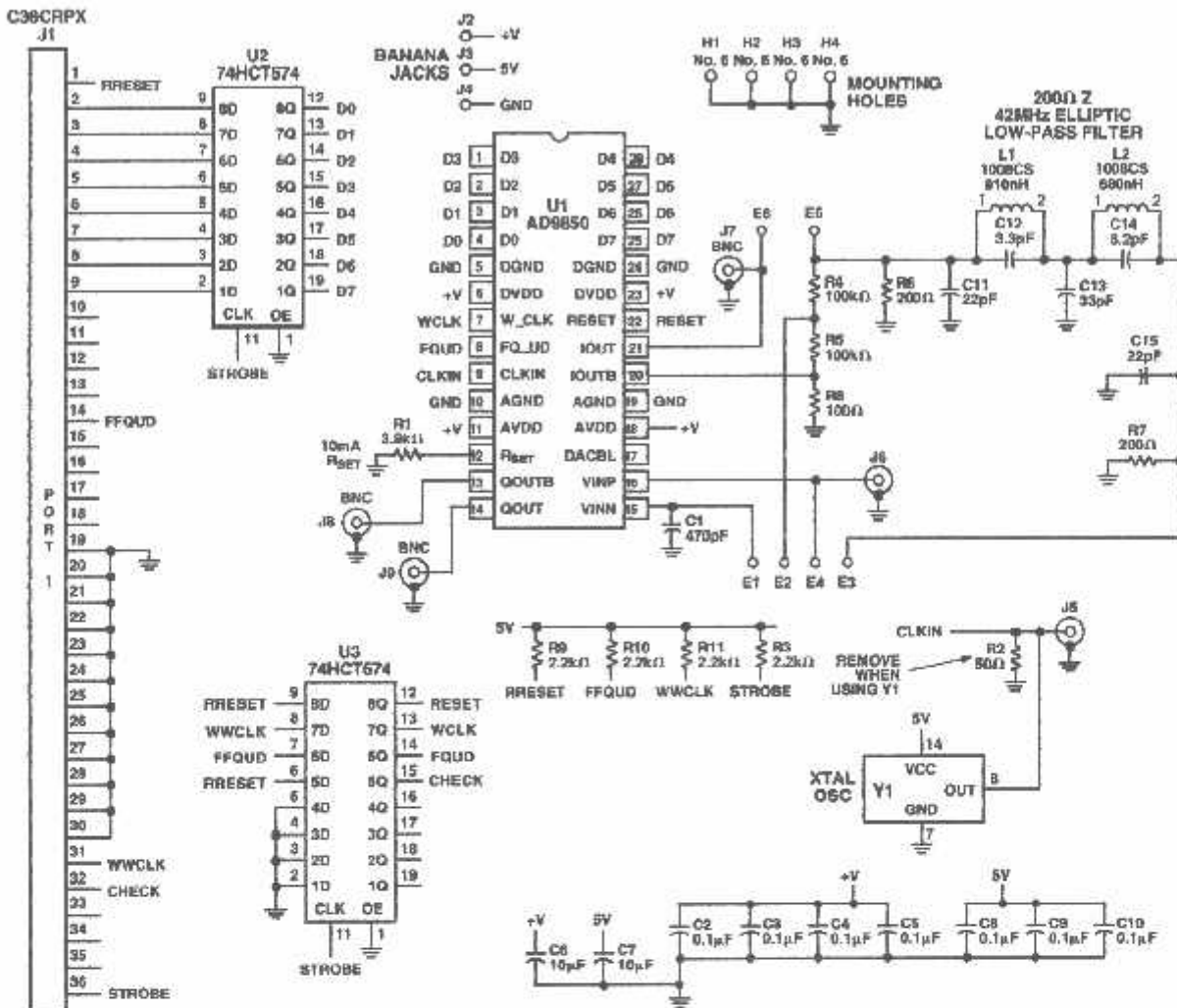


Figure 17. AD9850/CGPCB Electrical Schematic

COMPONENT LIST

Integrated Circuits

- U1 AD9850BRS (28-Lead SSOP)
- U2, U3 74HCT574 H-CMOS Octal Flip-Flop

Capacitors

- C1 470 pF Ceramic Chip Capacitor
- C2 to C5, C8 to C10 0.1 μF Ceramic Chip Capacitor
- C6, C7 10 μF Tantalum Chip Capacitor
- C11 22 pF Ceramic Chip Capacitor
- C12 3.3 pF Ceramic Chip Capacitor
- C13 33 pF Ceramic Chip Capacitor
- C14 8.2 pF Ceramic Chip Capacitor
- C15 22 pF Ceramic Chip Capacitor

Resistors

- R1 3.9 kΩ Resistor
- R2 50 Ω Resistor
- R3, R9, R10, R11 2.2 kΩ Resistor
- R4, R5 100 kΩ Resistor
- R6, R7 200 Ω Resistor
- R8 100 Ω Resistor

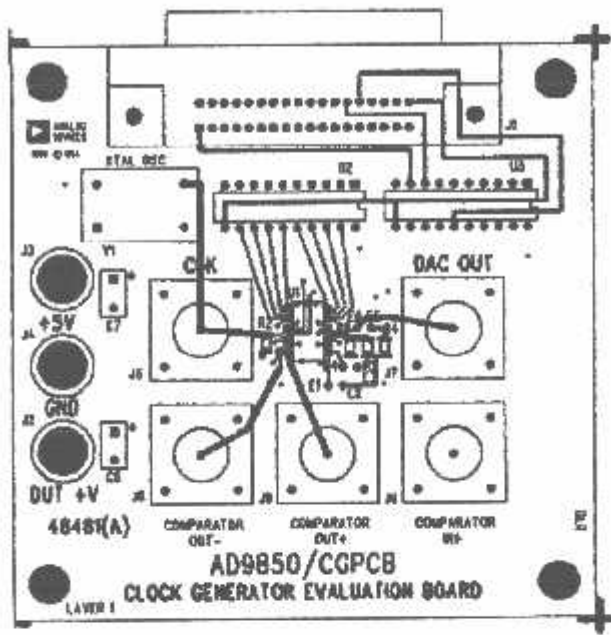
Connectors

- J2, J3, J4 Banana Jack
- J5 to J9 BNC Connector

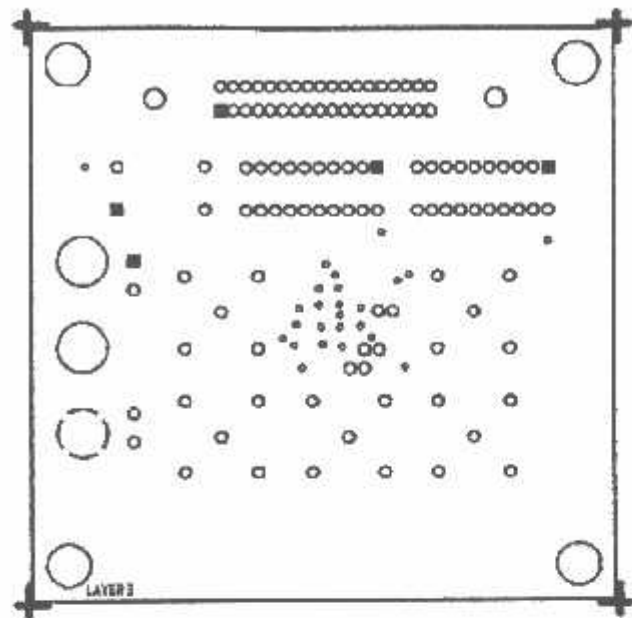
Inductors

- L1 910 nH Surface Mount
- L2 680 nH Surface Mount

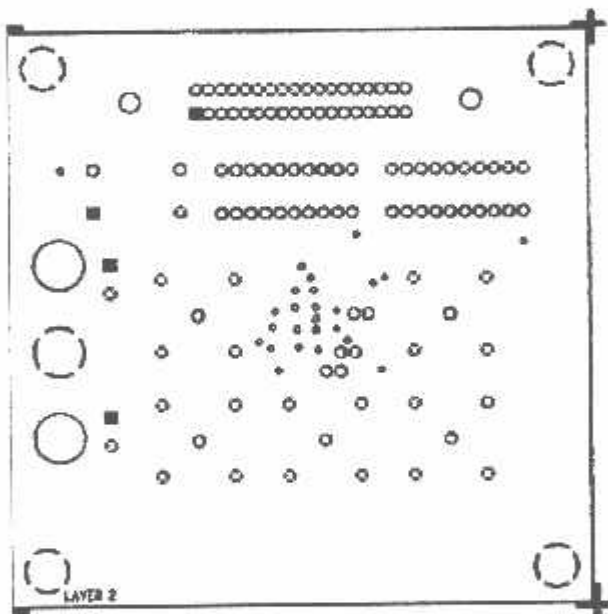
D9850



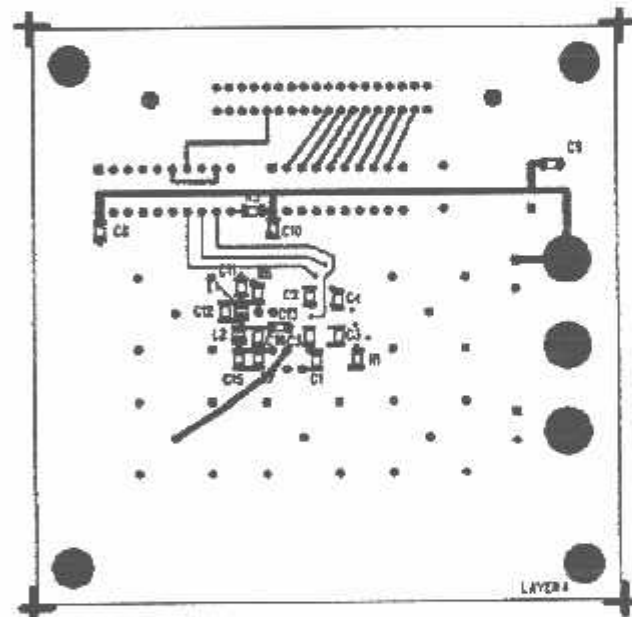
18a. AD9850/CGPCB Top Layer



18c. AD9850/CGPCB Power Plane



18b. AD9850/CGPCB Ground Plane



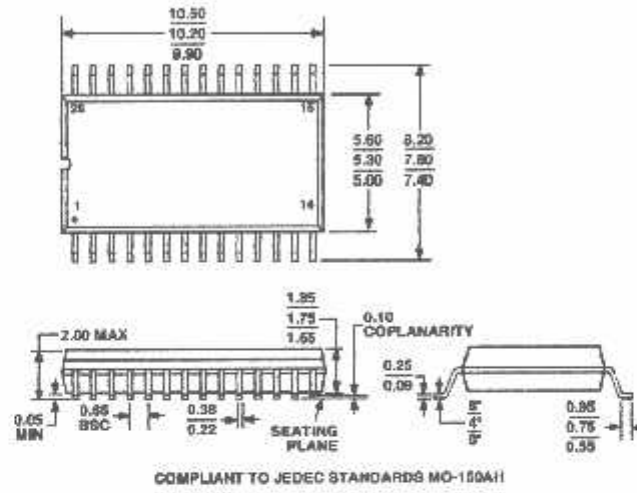
18d. AD9850/CGPCB Bottom Layer

Figure 18. AD9850/CGPCB Evaluation Board Layout

OUTLINE DIMENSIONS

28-Lead Shrink Small Outline Package [SSOP]
(RS-28)

Dimensions shown in millimeters



D9850

Revision History

Revision	Description	Page
04	Data Sheet changed from REV. G to REV. H. Changes to SPECIFICATIONS	3
03	Data Sheet changed from REV. F to REV. G. Changes to SPECIFICATIONS	3
	Changes to Table I	5
03	Data Sheet changed from REV. E to REV. F. Numbered figures and TPCs	Universal
	Changes to SPECIFICATIONS	2
	Changes to ABSOLUTE MAXIMUM RATINGS	3
	Updated ORDERING GUIDE	4
	Updated OUTLINE DIMENSIONS	9

C00632-0-2/04(H)