

**PERANCANGAN DAN PEMBUATAN APLIKASI
PENGONTROL MODUL DIRECT DIGITAL SYNTHESIZER
AD9850 SEBAGAI PEMBANGKIT FREKUNSI BERBASIS
ANDROID**

SKRIPSI



**Disusun Oleh :
FAHMI NOOR CHOLIK
NIM. 12.12.525**

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

LEMBAR PERSETUJUAN

**PERANCANGAN DAN PEMBUATAN APLIKASI PENGONTROL
MODUL DIRECT DIGITAL SYNTHESIZER AD9850 SEBAGAI
PEMBANGKIT FREKUENSI BERBASIS ANDROID**


SKRIPSI

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Persyaratan
Guna Mencapai Gelar Sarjana Teknik*


**Disusun oleh :
FAHMI NOOR CHOLIK
NIM. 12.12.525**

Diperiksa dan Disetujui,

Dosen Pembimbing I

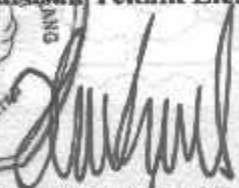

Dr. Ir. F. Yudi Limpraptono, MT
NIP.Y. 1039500274

Dosen Pembimbing II


Ir. Eko Nurcahyo, MT
NIP.Y. 1028700172



**Mengetahui,
Ketua Jurusan Teknik Elektro S-1**


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

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2016**

PERANCANGAN DAN PEMBUATAN APLIKASI PENGONTROL MODUL DIRECT DIGITAL SYNTHESIZER AD9850 SEBAGAI PEMBANGKIT FREKUENSI BERBASIS ANDROID

Fahmi Noor Cholik, NIM 1212525

Dosen Pembimbing : Dr. F. Yudi Limpraptono, ST, MT dan
Ir. Eko Nurcahyo, MT

Konsentrasi Teknik Komputer, Jurusan Teknik Elektro S-1
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
Jl. Raya Karanglo Km.2 Malang
E-mail : fahmisadewo87@gmail.com

ABSTRAK

Direct Digital Synthesizer (DDS) merupakan 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 dibangkitkan dapat diubah sesuai kebutuhan penggunaannya. Aplikasi android frekuensi digunakan sebagai masukan nilai frekuensi yang ingin dibangkitkan kemudian dikirimkan kedalam media Bluetooth, Bluetooth HC05 sebagai penerima dan terhubung pada Arduino Uno, Arduino Uno berfungsi sebagai pemroses data dari android sebagai unit masukan (input) untuk selanjutnya mengirimkan data dari perhitungan dan pemrosesan tersebut ke modul AD9850. Untuk Nilai selisih rata rata dari rentang frekuensi 1Hz hingga 40MHz Frekuensi Dalam keadaan stabil dengan prosentase kesalahan sebesar 0,6936037

Kata Kunci : *DDS, AD9850, Arduino, Android*

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 **“PERANCANGAN DAN PEMBUATAN APLIKASI PENGONTROL MODUL DIRECT DIGITAL SYNTHESIZER AD9850 SEBAGAI PEMBANGKIT FREKUENSI BERBASIS ANDROID”** 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. Ir. Soeparno Djiwo, 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 Pembimbing Ketua Jurusan Teknik Elektro S-1 Institut Teknologi Nasional Malang
4. Dr.F.Yudi Limpraptono, ST, MT selaku Dosen Pembimbing Satu Skripsi.
5. Ir. Eko Nurcahyo, 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 Tabel	v
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan	1
1.4 Batasan Masalah	2
1.5 Metodologi Pemecahan Masalah.....	2
1.6 Sistematika Penulisan	3
BAB II LANDASAN TEORI	
2.1. Android	4
2.1.1 Android versi 1.1	5
2.1.2 Android Versi 1.5 (Cupcake)	5
2.1.3 Android Versi 1.6 (Donuts).....	5
2.1.4 Android Versi 2.0 (Eclair).....	5
2.1.5 Android Versi 2.2 (Froyo).....	6
2.1.6 Android Versi 2.3 (Gingerbread)	6
2.1.7 Android Versi 3.0 (Honeycomb).....	7
2.1.8 Android Versi 4.0 (Ice Cream Sandwich).....	7
2.1.9 Android Versi 4.1 (Jelly Bean).....	8
2.2 Struktur Android	8
2.2.1 Application	8
2.2.2 Application Framework.....	8
2.2.3 Libraries.....	9
2.2.4 Android Runtime.....	10

2.3 Dasar Aplikasi Android	10
2.4 Komponen Aplikasi Android	10
2.4.1 Activities.....	10
2.4.2 Content Provider	11
2.4.3 Broadcast Receiver.....	11
2.4.4 Wireless Fidelity (Wifi).....	12
2.5 Konsep Dasar Pemrograman Android	12
2.5.1 Java.....	12
2.5.2 Java 2 Standart Edition.....	13
2.5.3 Java OSC	13
2.5.4 Eclipse	13
2.5.5 Android Software Development Kit (SDK)	14
2.5.6 Android Development Tools (ADT).....	14
2.5.7 Android APK (Android Package).....	15
2.6 UML (Unfield Modeling Language)	16
2.6.1 Usecase Diagram.....	17
2.6.2 Class Diagram.....	17
2.6.3 Activity Diagram.....	17
2.6.4 Squence Diagram	18
2.6.5 Collaboration Diagram	18
2.6.6 Component Diagram	18
2.6.7 Deployment Diagram	19
2.7 DDS (Direct Digital Synthesizer).....	19
2.7.1 Kelebihan dan Fleksibeitas DDS	19
2.7.2 Modul Direct Digital Synthesizer AD9850.....	20
2.7.3 Spesifikasi Modul DDS AD9850.....	20
2.7.4 Pengaturan Frekuensi Keluaran DDS	21
2.8 Arduino Uno R3	23
2.8.1 Catu Daya Arduino Uno.....	24
2.8.2 Memory ATmega328	25
2.8.3 Komunikasi Arduino Uno	26
2.8.4 Programing Arduino Uno.....	26

2.8.5 Perangkat Lunak (Arduino IDE).....	26
2.8.6 Otomatis Software Reset.....	27
2.9 Bluetooth HC05.....	27
BAB III PERANCANGAN SISTEM	
3.1 Pendahuluan.....	29
3.2 Prinsip Kerja.....	29
3.3 Perancangan Perangkat Keras.....	30
3.3.1 Perancangan Rangkaian Modul AD9850.....	30
3.3.2 Perancangan Rangkaian Bluetooth.....	31
3.4 Perancangan Perangkat Lunak Sistem.....	31
3.4.1 Perancangan dan Program Arduino.....	32
3.5 Perancangan Aplikasi.....	32
3.5.1 Flowchart.....	33
3.5.2 Usecase Diagram.....	34
3.5.3 Activity Diagram.....	35
3.5.4 User Interface.....	37
BAB IV PENGUJIAN DAN PEMBAHASAN SISTEM	
4.1 Pendahuluan.....	38
4.2 Pengujian Minimum Sistem Arduino UNO.....	38
4.2.1 Peralatan yang Dibutuhkan.....	38
4.2.2 Langkah-langkah yang Dilakukan.....	39
4.2.3 Hasil Pengujian.....	39
4.3 Pengujian Aplikasi Android.....	39
4.4 Pengujian Modul Bluetooth Android Ke Bluetooth HC05.....	41
4.5 Pengujian Keseluruhan Sistem.....	43
4.5.1 Analisa Hasil Pengujian.....	47
BAB V PENUTUP	
5.1 Kesimpulan.....	49
DAFTAR PUSTAKA.....	50

DAFTAR GAMBAR

Gambar 2.1	Komponen Utama Sistem Operasi Android.....	8
Gambar 2.2	Blog Diagram Fungsi AD9850	21
Gambar 2.3	Pin Configuration IC AD9850	23
Gambar 2.4	Board Arduino Uno	24
Gambar 2.5	Tampilan Framework Arduino Uno	27
Gambar 2.6	Modul Bluetooth HC05	27
Gambar 3.1	Blok Diagram	29
Gambar 3.2	Rangkaian Koneksi Arduino Dengan AD9850.....	30
Gambar 3.3	Rangkaian Komunikasi Bluetooth.....	31
Gambar 3.4	Flowchart Perancangan Aplikasi.....	33
Gambar 3.5	Flowchart Sistem Aplikasi.....	34
Gambar 3.6	Usecase Sistem	35
Gambar 3.7	Activity Diagram Sistem Koneksi.....	36
Gambar 3.8	Activity Diagram Sistem Aplikasi.....	36
Gambar 3.9	Tampilan Desain Aplikasi.....	37
Gambar 4.1	Skematik Pengujian Board Arduino	39
Gambar 4.2	Script Pengujian Arduino.....	39
Gambar 4.3	Hasil Tampilan Aplikasi Android Frekuensi	40
Gambar 4.4	Hasil Tampilan Pairing Perangkat Bluetooth	40
Gambar 4.5	Hasil Tampilan Pairing Device Bluetooth	41
Gambar 4.6	Hasil Pengujian Frekuensi 1Hz	43
Gambar 4.7	Hasil Pengujian Frekuensi 100Hz	44
Gambar 4.8	Hasil Pengujian Frekuensi 1000Hz	45
Gambar 4.9	Hasil Pengujian Frekuensi 1MHz.....	45
Gambar 4.10	Hasil Pengujian Frekuensi 40MHz.....	46

DAFTAR TABEL

Tabel 2.1 Runtutan Data Frekuensi	22
Tabel 2.2 Deskripsi Arduino Uno	24
Tabel 4.1 Pengujian Pengiriman Bluetooth	41
Tabel 4.2 Pengujian Pengiriman Bluetooth	42
Tabel 4.3 Pengujian Pengiriman Bluetooth	42
Tabel 4.4 Hasil Pengujian Frekuensi	47

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan system operasi handphone telah mengalami kemajuan. Sebagaimana kita kenal saat ini ada *iOs, PalmOs, Symbian, Java*, dan *Android*. Adapun system yang sedang dikembangkan oleh Google adalah *Android*, yang saat ini sangat populer di masyarakat dan mendukung fitur touch screen beserta fitur canggih lainnya.

Synthesizer frekuensi 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].

Saat ini metode sintesa frekuensi menggunakan teknik digital *DirectDigital Synthesis (DDS)* yaitu dengan memori data yang dicuplik dengan kecepatan dan interval tertentu sehingga menghasilkan frekuensi yang diinginkan. Akurasi frekuensi yang dihasilkan sangat penting karena jika terjadi pergeseran frekuensi dari yang diharapkan akan menyebabkan terjadinya gangguan[2],

Maka dari itu untuk memudahkan pengontrolan Pembangkit Frekuensi dengan menggunakan modul *Direct Digital synthesizer AD9850* yang di kendalikan melalui Arduino, diperlukan sebuah aplikasi Android guna untuk mempermudah dalam pengontrolan

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diutarakan diatas, maka dapat disimpulkan permasalahan yang dituangkan dalam Proposal Skripsi ini, yaitu :

1. Bagaimana perancangan dan pembuatan Aplikasi Android Untuk Mengontrol *Direct Digital Synthesizer(DDS) AD9850* pembangkit frekuensi dengan komunikasi melalui bluetooth

1.3 Tujuan Penelitian

Perencanaan dan Pembuatan Pengontrol Modul *Direct Digital Synthesizer AD9850* Pembangkit Frekuensi Berbasis Arduino dan input Frekuensi akan di

tampilkan pada aplikasi android dengan menggunakan media komunikasi melalui bluetooth.

1.4 Batasan Masalah

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

1. Dalam penelitian ini pembangkit frekuensi menggunakan modul *Direct Digital Synthesizer* AD9850 dan Arduino frekuensinya akan di tampilkan pada Aplikasi Android.
2. Aplikasi yang Dirancang Hanya bisa digunakan pada Sistem Operasi Android
3. Komunikasi antara pembangkit frekuensi dengan ponsel android melalui media bluetooth.
4. Output yang dikeluarkan berupa gelombang sinus

1.5 Metodologi Masalah

Metode yang digunakan dalam penyusunan skripsi ini adalah:

1. Kajian Literatur

Pengumpulan data dan informasi yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang berhubungan dengan permasalahan pada perancangan alat dan aplikasi.

2. Perancangan Sistem Elektronika

Pembuatan design rangkaian elektronika seperti: perancangan dan rangkain komunikasi dengan menggunakan bluetooth HC 05, perancangan rangkaian minimum sistem Arduino dengan modul pembangkit frekuensi DDS AD9850, perancangan aplikasi android.

3. Pembuatan Hardware

Pembuatan rangkaian dari hasil perencanaan sistem yang meliputi :

1. Pembuatan rangkaian Arduino dengan modul pembangkit frekuensi DDS AD9850.
 2. Pembuatan sistem komunikasi bluetooth HC05
 3. Proses pengkabelan dan penempatan keseluruhan rangkaian
-

4. Penulisan Program

Program yang dibuat menggunakan bahasa pemrograman yang akan ditanamkan pada Arduino sedangkan untuk program aplikasi android dibuat menggunakan bahasa pemrograman java.

5. Pengujian Sistem

Proses ujicoba rangkaian dan keseluruhan sistem untuk mengetahui adanya kesalahan agar sistem sesuai dengan konsep yang telah dirancang sebelumnya.

6. Pelaporan hasil pengujian dan kesimpulan.

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 : TINJAUAN PUSTAKA

Membahas tentang dasar teori mengenai permasalahan yang berhubungan dengan penelitian.

BAB III : PERANCANGAN

Bab ini membahas tentang perancangan sistem Hardware dan Software serta desain komunikasi

BAB VI : PENGUJIAN

Berisi tentang pembahasan langkah-langkah pengujian alat serta pengujian Aplikasi yang telah dirancang.

BAB V : PENUTUP

Berisi tentang semua kesimpulan yang berhubungan dengan penulisan skripsi, dan saran yang digunakan sebagai pertimbangan dalam pengembangan program selanjutnya.

BAB II LANDASAN TEORI

2.1. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi.

Android merupakan generasi baru platform mobile, platform yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi Android dilisensikan di bawah GNU, General Public Lisensi versi 2 (GPLv2), yang sering dikenal dengan istilah “copyleft” lisensi dimana setiap perbaikan pihak ketiga harus terus jatuh di bawah terms. Android didistribusikan di bawah lisensi Apache Software (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya, komersialisasi pengembang (produsen handset khususnya) dapat memilih untuk mengembangkan platform tanpa harus memberikan perbaikan ke masyarakat open source. Sebaliknya, pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan di bawah lisensi apa pun yang mereka inginkan. Pengembang aplikasi Android diperbolehkan untuk mendistribusikan aplikasi di bawah skema lisensi apapun yang diinginkan.

Pengembang memiliki beberapa pilihan ketika membuat aplikasi yang berbasis Android, sebagian besar pengembang menggunakan Eclipse yang tersedia secara bebas untuk merancang dan mengembangkan aplikasi Android. Eclipse adalah IDE yang paling populer untuk pengembangan Android. Selain itu, Eclipse juga mendapat dukungan langsung dari Google untuk menjadi IDE pengembangan dari aplikasi Android, ini terbukti dengan penambahan plugins di Eclipse untuk membuat project android di mana source software langsung dari situs resminya Google. Akan tetapi, hal di atas tidak menutup kemungkinan untuk menggunakan IDE yang lain seperti Netbeans untuk melakukan pengembangan Android.

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, diperkirakan akhir tahun 2010 hampir semua vendor menggunakan sistem operasi Android. Adapun versi-versi Android yang pernah dirilis adalah sebagai berikut[3] :

2.1.1 *Android Versi 1.1*

Android ini dirilis pada tahun 9 Februari 2009 dalam versi 1.1 user interface yang dimiliki sudah lebih baik dan terdapat beberapa perbaikan aplikasi, misalnya adanya pengelolaan pesan, alarm, voice search, pengiriman pesan dengan Gmail, dan pemberitahuan email.

2.1.2 *Android Versi 1.5 (Cupcake)*

Android ini dirilis pada tanggal 30 April 2009, cupcake atau Android versi 1.5 merupakan perkembangan dari versi 1.1 sehingga memiliki interface yang sudah lebih cantik dan ada penambahan efek animasi. Android versi 1.5 mampu merekam ataupun memutar video serta dapat mengunggah foto di Picasa dan video ke Youtube. Pada versi ini sudah dilengkapi dengan soft keypad dengan fitur auto complete dan mendukung bluetooth A2DP. Selain itu, terdapat pula widget baru yang mengikuti pola pada desktop, serta memiliki fasilitas tambahan, yaitu copy-paste pada internet berbasis webpage

2.1.3 *Android Versi 1.6 (Donuts)*

Android ini dirilis pada tanggal 15 September 2009. Android ini memiliki banyak terobosan terbaru di sistem jaringan dan konektivitas, misalnya adanya penambahan kamera, kamera video, dan galeri foto bahkan terdapat pula penambahan fungsi seleksi multi untuk foto pada fitur galeri. Android versi 1.6 memberikan peningkatan fitur pada voice search dan pengembangan voice dial, selain itu terdapat pula pengembangan fitur pencarian untuk bookmark, kontak dan halaman web plus kecepatan pencarian. Dari segi jaringan pun, terjadi peningkatan dan mendukung layar dengan resolusi WVGA, serta terdapat layanan Android Market.

2.1.4 *Android Versi 2.0/2.1 (Eclair)*

Eclair yang dirilis pada 3 Desember 2009 merupakan lanjutan perkembangan dari Donut. Android Eclair mengalami banyak pembaruan dari segi optimalisasi

hardware dan segi perangkat lunaknya, berikut ini adalah perubahan-perubahan yang terjadi pada Eclipse:

- Penggunaan browser baru dan mendukung HTML5.
- Adanya aplikasi Google Maps 3.1.2.
- Terdapat fasilitas Microsoft Exchange.
- Adanya kamera dengan flash dan zoom digital.
- Pengembangan MotionEvent dari Tracking Multi Touch untuk event.
- Adanya fasilitas keyboard virtual.
- Adanya aplikasi bluetooth v2.1.
- Kecepatan hardware dioptimalkan.
- Rasio B/W pada background lebih tinggi dan mendukung image beresolusi tinggi.

2.1.5 Android Versi 2.2 (Froyo)

Android Froyo merupakan penyempurnaan yang dilakukan android dari generasi sebelumnya (Eclipse). Android Froyo dirilis pada tanggal 20 Mei 2010 dan disempurnakan pada tanggal 20 Juni 2010. Android inilah yang sekarang sangat banyak beredar di pasaran. Berikut ini adalah penyempurnaan-penyempurnaan yang terjadi pada Android v2.2 (Froyo) yaitu:

- Kerangka aplikasi memungkinkan penggunaan dan penghapusan komponen yang tersedia.
- Dalvik Virtual Machine dioptimalkan untuk perangkat mobile. 3. Grafik 2D dan 3D berdasarkan Libraries OpenGL.
- SQLite untuk penyimpanan data
- Mendukung media audio, video, dan berbagai format gambar.
- Kinerja hardware lebih cepat.
- Dapat melakukan instalasi aplikasi ke memori eksternal.
- Adanya dukungan Adobe Flash 10.
- Fasilitas Wi-Fi dapat dijadikan hotspot.
- Adanya fungsi upgrade otomatis di Android Market.
- GSM, bluetooth, EDGE, 3G dan Wifi (hardware independent)
- Adanya fasilitas kamera, Global Positioning System (GPS), kompas, dan accelerometer

2.1.6 Android Versi 2.3 (Gingerbread)

Android Gingerbread merupakan pengembangan lanjutan dari android Froyo. Gingerbread dirilis antara bulan November dan Desember 2010. Terdapat beberapa penambahan dan pengembangan fitur yang diterapkan oleh Android

pada Gingerbread. Berikut ini adalah penambahan dan pengembangan pada Gingerbread:

- Peningkatan dan penambahan user interface
- Input teks lebih cepat dan intuitif
- Proses copy paste lebih mudah dilakukan
- Manajemen energi lebih baik
- Kontrol aplikasi lebih baik dengan aplikasi manage applications yang bisa akses dari menu options 6. Fitur komunikasi dan organizing yang lebih baik.

2.1.7 Android Versi 3.0 (Honeycomb)

Android versi 3.0 merupakan versi terbaru dari android dan diberi nama Honeycomb. Android Honeycomb dirilis pada bulan Januari 2011, jauh lebih awal dari perkiraan sebelumnya, yaitu pertengahan tahun 2011. Berikut ini adalah penambahan dan perbaikan yang terjadi pada android v3.0 atau Honeycomb yaitu

- Adanya kemudahan untuk mengakses akun Gmail dengan action bar
- Mendukung grafis tiga dimensi
- Mendukung aplikasi khusus yang diterapkan pada perusahaan tertentu dan memiliki sistem keamanan yang memadai
- Keyboard virtual yang dimiliki lebih komplit dari versi-versi sebelumnya

2.1.8 Android Versi 4.0 (Ice Cream Sandwich)

Android versi 4.0 ini dinamakan ice cream sandwich. Android versi ini dirilis sekitar bulan Oktober tahun 2011, Android 4.0 menjadi OS Google Android pertama yang berjalan di Tablet PC dan smartphone. Dukungan resolusi monitor lebih lebar dan interface tombol dibagian bawah lebih mudah di atur. Berikut adalah fitur-fitur yang baru dari Android versi 4.0

Android versi ini memiliki huruf yang baru bernama Roboto. Bentuknya agak melingkar dan lebih bulat dibandingkan dengan font-font yang sudah ada.

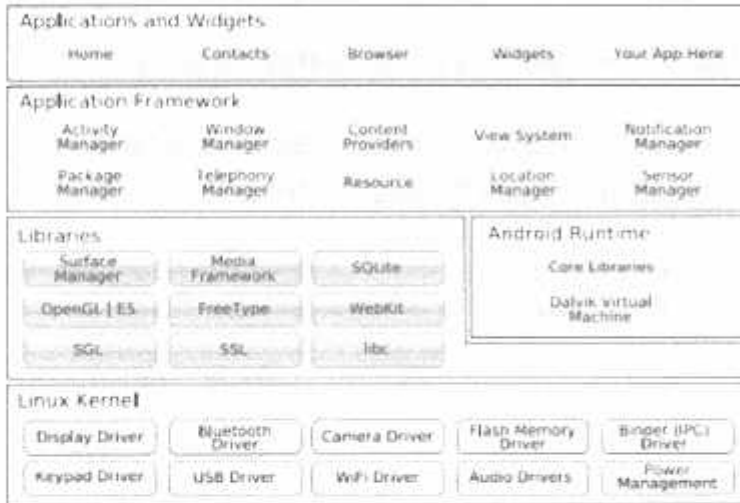
- Beranda, back, menu, dan tombol pencari berubah menjadi tombol sentuh.
 - Perombakan tebesarnya ada di bagian user interface, gambar pada kotak tampil lebih besar dan mampu mengatur ukuran widget pada homescreen, dan adapula panel favorit yang selalu muncul di dasar layar untuk menampilkan daftar nama dan nomor telepon, aplikasi, dan halaman internet secara instant.
 - Aplikasi Video mempunyai pilihan untuk mempercepat dan memperlambat gambar yang sedang diputar
-

2.1.9 Android Versi 4.1 (Jelly Bean)

Android Jelly Bean merupakan versi Android yang terbaru pada saat ini, fitur terbaru dari sistem operasi Android Jelly Bean ini salah satunya adalah peningkatan kemampuan on screen keyboard yang lebih cepat serta responsif, pencarian data kontak dengan fitur voice search dan lain sebagainya.

2.2 Arsitektur Android

Gambar berikut merupakan komponen-komponen utama dari sistem operasi Android yang terbagi menjadi beberapa layer.



Gambar 11.1 Komponen Utama Sistem Operasi Android

2.2.1 Application

Application ini adalah layer dimana kita berhubungan dengan aplikasi saja, dimana terlebih dahulu dilakukan download aplikasi kemudian dilakukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

2.2.2 Application Framework

Android adalah "open Development Platform" yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Application framework menyediakan semua kelas yang digunakan untuk membuat aplikasi Android dan juga sebuah gambaran umum untuk akses hardware, application resources, menjalankan service

background, mengatur alarm, menambah status notification dan sebagainya. Pengembang memiliki akses penuh menuju API framework seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (reuse), sehingga bisa kita simpulkan application framework ini adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti content-providers yang berupa sms dan panggilan telepon.

Berikut adalah sekumpulan services dan sistem yang mendasari semua aplikasi Android[1] :

- sekumpulan dari views yang dapat dikembangkan untuk membuat sebuah aplikasi, termasuk list, grids, text boxes, buttons, dan embeddable browser
- Content providers yang berfungsi untuk mengakses data dari aplikasi lain atau untuk memberikan data dari aplikasi itu sendiri.
- Resource manager yang menyediakan akses pada non-code resources seperti localized strings, graphics, dan layout files.
- Notification manager yang memungkinkan semua aplikasi untuk menampilkan tanda notifikasi tersendiri pada status bar.
- Activity manager untuk mengatur siklus dari aplikasi dan menyediakan sebuah commomnavigation backstack

2.2.3 Libraries

Libraries adalah layer dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya [18] . Berjalan di atas karnel, layer ini meliputi berbagai library C/C++ seperti libc dan secure sockets layer (SSL) yang digunakan oleh berbagai komponen pada sistem Android. Beberapa library lainnya adalah:

- Media library untuk memuar media audio dan video.
 - Surface Manager untuk menyediakan pengaturan tampilan.
 - Graphics library termasuk Scene Graph Library (SGL) dan Open Graphics Library (Open GL) untuk kemampuan menampilkan grafik dua dan tiga dimensi.
 - SQLite untuk dukungan dalam database.
 - SSL dan WebKit untuk web browser yang terintegrasi dan keamanan internet.
-

- Libraries LiveWebcore mencakup modern web browser dengan engine embeded web view
- Libraries 3D yang mencakup implementasi OpenGL ES 1.0 API's

2.2.4 Android Runtime

Android runtime adalah engine yang berfungsi dalam menjalankan berbagai aplikasi. Di dalam android Run Time dibagi menjadi dua bagian yaitu:

1. Core Libraries Aplikasi Android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesinnya bukan Virtual Machine Java, sehingga diperlukan sebuah libraries yang berfungsi untuk menterjemahkan bahasa java/c yang ditangani oleh Core Libraries.
2. Dalvik Virtual Machine Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, di mana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan threading dan manajemen tingkat rendah.

Linux Kernel Linux Kernel adalah layer dimana inti dari operating system dari Android itu berada. Berisi file-file system yang mengatur sistem processing, memory, resource, drivers, dan sistem-sistem operasi 31 android lainnya. Linux Kernel yang digunakan android adalah Linux Kernel release 2.6 [18].

2.3 Dasar Aplikasi Android

Semua aplikasi Android dikembangkan menggunakan bahasa pemrograman java. Kode java akan di kompilasi bersama data dan resource files yang dibutuhkan oleh aplikasi. Semuanya kemudian akan digabungkan oleh Android Asset Packaging Tool (AAPT) menjadi sebuah Android package yang menggunakan format .apk. file berekstensi .apk ini akan menjadi penampung untuk menyimpan aplikasi dan digunakan dalam proses instalasi ke mobile device dari user [9].

2.4 Komponen Aplikasi Android

Komponen aplikasi android adalah bagian penting dari sebuah aplikasi android. Terdapat empat jenis komponen aplikasi dimana setiap jenis mempunyai tujuan dan siklus berbeda yang mendefinisikan bagaimana suatu komponen dibuat dan dihilangkan[2]

2.4.1 Activities

Sebuah activity mewakili satu tampilan untuk user interface. Sebagai contoh, sebuah aplikasi e-mail memiliki activity untuk menunjukkan daftar dari email

baru, membuat email dan membaca email. Walaupun setiap activity bekerja dalam waktu yang sama untuk membentuk suatu aplikasi email yang utuh, masing-masing activity bersifat independen dari yang lainnya. Aplikasi lain juga dapat menggunakan sebuah activity dari aplikasi email tersebut jika memang diberi hak akses. Sebagai contoh, aplikasi kamera dapat menggunakan activity pada aplikasi email untuk membuat email, dengan tujuan untuk mengirim gambar.

Service

Service adalah komponen yang berjalan di background untuk melakukan suatu operasi dalam waktu yang lama. Suatu service tidak memiliki Graphic User Interface (GUI). Sebagai contoh dalam memainkan musik, service mungkin memainkan musik atau mengambil data dari jaringan, tetapi setiap service harus berada dalam kelas induknya. Misalnya, media player sedang memutar lagu dari list yang ada, aplikasi ini akan memiliki dua atau lebih activity yang memungkinkan user untuk memilih lagu misalnya, atau menulis sms sambil player sedang jalan. Untuk menjaga musik tetap dijalankan, activity player dapat menjalankan service. Service dijalankan pada thread utama dari proses aplikasi.

2.4.2 Content Providers

Content providers bertugas mengatur sekumpulan pengiriman data suatu aplikasi. User dapat menyimpan data di sistem file, SQLite database atau lokasi penyimpanan lain yang dapat diakses oleh aplikasi. Melalui content provider, aplikasi lain dapat melakukan query atau memodifikasi data. Sebagai contoh, android menyediakan sebuah aplikasi yang mengatur kontak informasi user. Dengan demikian, aplikasi yang telah mendapat ijin dapat melakukan query dari content provider untuk membaca dan menulis informasi mengenai kontak tersebut. Content provider juga berfungsi dalam membaca dan menulis data yang bersifat private. Contohnya, aplikasi seperti notepad menggunakan content provider untuk menyimpan catatan.

2.4.3 Broadcast Receiver

Broadcast receiver adalah suatu komponen yang merespons terhadap broadcast yang terjadi pada sistem secara keseluruhan. Sebagian besar broadcast berasal dari sistem. Contohnya adalah suatu broadcast yang menginformasikan bahwa tampilan telah dimatikan, keadaan low-battery atau sebuah gambar yang

telah diambil. Aplikasi juga dapat melakukan broadcast, misalnya dengan membiarkan aplikasi lain mengetahui bahwa ada data yang telah selesai di unduh ke device yang sudah digunakan. Walaupun broadcast receiver tidak memiliki user interface, namun tetap ada status bar sebagai notifikasi yang dapat muncul untuk memberi tahu user ketika telah terjadi broadcast event. Secara umum, broadcast receiver hanyalah satu "entry point" ke komponen lain dan ditujukan untuk melakukan pekerjaan yang kecil. Sebagai contoh broadcast receiver dapat membuat suatu service bekerja ketika terjadi event tertentu.

2.4.4 Wireless Fidelity (Wi-Fi)

Wi-fi adalah teknologi Wireless Local Area Network (WLAN) yang menggunakan standar Institute Of Electrical and Electronics Engineer (IEEE) 802.11. Istilah WI-Fi sendiri merupakan merek dagang Wi-Fi alliance yang kemudian dijadikan nama produk[3] . Untuk dapat menggunakan Wi-Fi, suatu device harus memiliki Wireless Network Interface Controller (WNIC) untuk menerima dan mengirim sinyal wireless. Beberapa devices seperti notebook dan smartphone biasanya telah memiliki WNIC tersebut di dalamnya, sedangkan untuk device lain seperti komputer dekstop biasanya terpasang secara terpisah. Device yang menggunakan Wi-Fi dapat mengakses internet selama dalam jangkauan jaringan dan tidak ada gangguan pada sinyal wireless. Area jangkauan jaringan wireless ini disebut dengan istilah Access Point atau Hotspot.

2.5 Konsep Dasar Pemrograman smartphone berbasis Android.

2.5.1 Java

Java merupakan bahasa pemrograman yang dikembangkan oleh James Gosling di Sun Microsystems yang sekarang merupakan bagian dari Oracle Corporation[11] . Aplikasi java biasanya dikompilasi menjadi bytecode (file kelas) yang dapat berjalan di Java Virtual Machine (JVM) manapun walaupun berbeda dalam hal arsitektur komputer. JVM inilah yang membuat java menjadi platform-independent. JVM bisa berbeda tergantung pada platform yang digunakan. Misalnya JVM pada microsoft windows akan berbeda dengan JVM pada Mac OS dan Linux. Setiap komputer dan peralatan elektronik dapat menjalankan program java selama device tersebut memiliki JVM. Java merupakan bahasa pemrograman

paling populer mengalahkan C dan C++[16] . Beberapa kelebihan yang ditawarkan bahasa pemrograman Java:

1. Sederhana Java dirancang untuk mudah digunakan dan mudah dalam proses penulisan, komplikasi, debug dan pembelajaran dibandingkan bahasa pemrograman lain.
2. Object Oriented Programming (OOP) Java bersifat object oriented karena pemrograman di java mengutamakan dalam pembuatan objek, manipulasi objek dan membuat objek bekerja bersama. Hal ini mempermudah pemrograman karena user dapat membuat program modular dan menggunakan kembali kode yang ada.
3. Platform-independent Kelebihan utama java ialah dapat dijalankan di beberapa platform atau sistem operasi. Hal ini sesuai dengan prinsip java yaitu "Write once, run anywhere". Dengan kelebihan ini, programmer cukup menulis sebuah program java dan dikompilasi lalu hasilnya bisa dijalankan di beberapa platform tanpa harus melakukan perubahan.

2.5.2 Java 2 Standard Edition (J2SE)

Merupakan edisi standar dari platform java yang dirancang untuk meningkatkan keamanan serta kemudahan dan merancang aplikasi berforma tinggi untuk dekstop dengan jangkauan yang luas meliputi sistem operasi seperti Mac OS, Linux, Microsoft Windows, dan Sun Solaris[11] .

2.5.3 Java OSC

Java OSC adalah sebuah library untuk menggunakan protokol open sound control (OSC) di java. Java OSC sendiri bukanlah sebuah program, tetapi lebih kepada library yang dirancang untuk mengembangkan program yang memerlukan komunikasi melalaui OSC. OSC adalah sebuah protokol untuk saling berkomunikasi antara komputer, sound synthesizer, dan device multimedia lain yang telah dioptimasi untuk kebutuhan teknologi jaringan modern. Kelebihan OSC meliputi interoperabilitas, ketelitian, fleksibilitas, dan peningkatan kinerja dalam organisasi dan dokumentasi.

2.5.4 Eclipse

Eclipse adalah sebuah kumpulan open source yang membangun sebuah platform untuk merancang dan mengembangkan software yang bersifat terbuka dan multi-bahasa. Eclipse terdiri dari Integrated Development Environment (IDE) dan sistem plug-in. Walaupun sebagian besar ditulis dalam bahasa java, perancangan aplikasi dapat menggunakan bahasa lain dengan penambahan plug-in seperti Ada,

C, C++, PHP, dan sebagainya. Eclipse bersifat terbuka karena mengizinkan ekstensi tambahan dari pihak third-party dan merupakan sebuah IDE karena menyediakan tools untuk mengatur workspace, dapat membangun, menjalankan dan debug aplikasi dan melakukan penyesuaian terhadap pemrograman. Eclipse disebut sebuah platform karena dirancang untuk diperluas tanpa batas dengan penambahan berbagai tools[5]

2.5.5 Android Software Development Kit (SDK)

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman java. Android SDK sebagai kumpulan dari platform, tools, sample code dan dokumentasi yang dibutuhkan untuk mengembangkan aplikasi android[18] . SDK ini merupakan add-on Java Development Kit (JDK) dan memiliki plug-in terintegrasi untuk Eclipse IDE [14] . Tools pada Android SDK dibagi menjadi dua jenis yaitu:

- SDK tools yang tidak bergantung pada platform dan dibutuhkan pada semua platform android.
- Platform tools yang telah disesuaikan untuk mendukung fitur terbaru dari platform android tertentu

Beberapa tools yang disediakan oleh Android SDK :

1. Android Emulator Sebuah implementasi dari android virtual machine yang dirancang untuk berjalan di komputer. Tools ini dapat digunakan untuk melakukan pengujian dan debug aplikasi android.
2. Dalvik debug Moniyoring Service (DDMS) Untuk penggunaan perspektif DDMS dalam memantau dan mengontrol Dalvik VM pada tahap debugging.
3. Android Asset Packaging Tool (AAPT) Untuk membangun android package file (file berekstensi .apk).
4. Android Debug Bridge (ADB) Sebuah aplikasi client-server yang menyediakan link ke emulator sehingga dapat menyalin file instalasi apk, dan menjalankan shell commands.

2.5.6 Android Development Tools (ADT)

ADT adalah sebuah plug-in untuk eclipse IDE yang dirancang untuk memberikan lingkungan terintegrasi dalam membangun aplikasi android, dengan menggunakan ADT untuk eclipse akan dimudahkan dalam membuat aplikasi project android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya,

dan juga dapat melakukan running aplikasi menggunakan android SDK melalui eclipse. Dengan ADT juga dapat melakukan pembuatan package android (.apk) yang digunakan untuk distribusi aplikasi android yang dirancang. Merancang aplikasi menggunakan eclipse dengan ADT sangat direkomendasikan dan merupakan cara tercepat. Dengan adanya project setting, tools integration, custom Extensible Markup Language (XML) editor dan kemampuan untuk debug, ADT memberikan peningkatan yang signifikan dalam perancangan aplikasi android[6].

2.5.7 Android APK (Android Package)

APK adalah paket aplikasi android (Android Package). APK umumnya digunakan untuk menyimpan sebuah aplikasi atau program yang akan dijalankan pada perangkat Android. APK pada dasarnya seperti zip file, karena berisi dari kumpulan file. APK dapat diperoleh melalui berbagai metode, seperti menginstal sebuah aplikasi melalui market, download dari sebuah situs web, atau membuat sendiri dengan bahasa Java[18]. File APK berada di perangkat Android di direktori /data/app/filename.apk. Direktori ini biasanya tidak dapat diakses, untuk melindungi aplikasi berbayar, kecuali rooted ROM digunakan di telepon. Pada rooted ROM, sangat mungkin untuk menyalin aplikasi ke folder ini secara manual dengan menggunakan perintah 'adb push' atau semacamnya. Setelah sebuah file berada di direktori /data/app, maka aplikasi sudah terpasang pada telepon. Selain itu, dimungkinkan juga untuk menginstal aplikasi ke partisi system dari telepon yang sudah diroot. Cara ini memiliki keuntungan yaitu aplikasi tersebut tidak dapat dihapus dari perangkat, sehingga berguna untuk keamanan/anti-pencurian aplikasi seperti WaveSecure dan Mobile Defense. System aplikasi ini akan disimpan dalam /system/app. Aplikasi dalam folder ini biasanya bagian dari sistem operasi Android, dan sebaiknya jangan diubah atau dihapus kecuali diketahui apa yang dilakukan. File APK dapat dihapus dengan banyak cara, antara lain: menggunakan aplikasi Market (jika aplikasi diinstal dari Market), perintah 'uninstall adb apkgname.apk', melalui menu Settings | Applications | Manage Applications, atau dengan menghapus secara manual file-file dari direktori /data/app. Namun biasanya metode yang sering digunakan adalah melalui market atau menu aplikasi[18].

2.6 UML (*Unified Modelling Language*)

UML adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak tergantung pada bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak. Standar UML dikelola oleh OMG (Object Management Group) (Heriyanto, 2004). UML adalah bahasa pemodelan untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan artifak-artifak dari sistem. Kelebihan UML diantaranya:

1. Di dalam (*system intensive process*), metode diterapkan sebagai proses untuk menurunkan atau mengevolusikan sistem.
 2. Sebagai bahasa UML digunakan untuk komunikasi, yaitu alat untuk menangkap pengetahuan (semantik) mengenai suatu subjek dan mengapresiasi yang memperdulikan subjek sebagai suatu komunikasi. Subjek merupakan sistem yang dibahas.
 3. Sebagai bahasa pemodelan, UML fokus pada pemahaman subjek melalui formulasi model dari subjek (dan konsep yang terhubung). Model memuat pengetahuan pada subjek, dan aplikasi dari pengetahuan ini berkaitan dengan intelegensi.
 4. Berkaitan dengan unifikasi, UML memadukan praktek rekayasa terbaik sistem informasi dan industri, meliputi beragam tipe sistem (perangkat lunak dan non lunak), domain (bisnis, perangkat lunak) dan proses siklus hidup.
 5. Ketika diterapkan untuk menspesifikasikan sistem, UML dapat digunakan untuk mengkomunikasikan “apa” yang diperlukan dari sistem dan “bagaimana” sistem dapat direalisasikan.
 6. Ketika diterapkan untuk memvisualisasikan sistem, UML dapat digunakan untuk menjelaskan sistem secara visual sebelum direalisasikan.
 7. Ketika diterapkan untuk membangun sistem, UML dapat digunakan untuk memandu realisasi sistem serupa dengan “blueprint”.
 8. Ketika diterapkan untuk mendokumentasikan sistem, UML dapat digunakan untuk menangkap pengetahuan mengenai sistem pada seluruh siklus hidup
-

menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi ditrigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan perilaku internal sebuah sistem (dan interaksi antar subsistem) secara eksak,

tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

2.6.4 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang menjadi *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

2.6.5 Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama.

2.6.6 Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul

berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time* *link time*, maupun *run time*.

Umumnya komponen terbentuk dari beberapa *class* dan, atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil.

2.6.7 *Deployment Diagram*

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti

keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik. Sebuah node adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

2.7 Sistem DDS (Direct Digital Synthesizer)

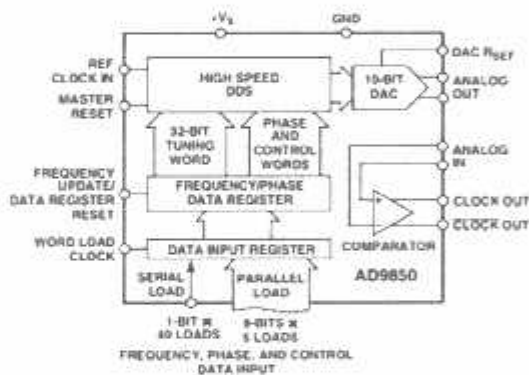
Sistem DDS Direct Digital Synthesizer merupakan salah satu cara untuk menghasilkan sinyal sinusoida secara langsung. Inti dari sistem ini adalah arsitektur akumulator dengan resolusi mencapai mili Hertz dan frekuensi sinyal yang dihasilkan dapat diatur tergantung dari sinyal frekuensi referensi. Keluaran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

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

2.7.1 Kelebihan dan Fleksibilitas DDS

Kelebihan menggunakan system DDS adalah karakteristik system DDS itu sendiri. Dimana keuntungan dari system ini adalah memiliki setting time / kecepatan yang cepat dan memiliki resolusi frekuensi yang halus terhadap frekuensi keluaran, operasi atas suatu spectrum frekuensi yang lebar dan dengan kemajuan dalam desain teknologi proses serta sangat ringkas dan membutuhkan sedikit daya sehingga sangat memungkinkan system DDS bisa lebih dikembangkan untuk desain alat yang berkaitan dengan aplikasi – aplikasi frekuensi hopping serta system – system yang berkaitan dengan peralatan pemancar radio, TV, peralatan test.

- Membangkitkan 2 gelombang sinus (sine wave) dan gelombang persegi (square wave)
- Frekuensi output range 1 Hz – 40 MHz
- Presaisi hingga frekuensi harmonic 20 MHz, max 40 MHz (diatas 20MHz gelombang mengalami istorsi.
- Frekuensi maksimum gelombang persegi 1MHz
- Data masukan dapat dipilih antara parallel dan serial dengan memindahkan jumper.
- Magnitudo gelombang keluaran dapat disetting dengan mudah melalui benchmark pin (pin#12)
- Tegangan referensi untuk pembanding (comprator reference input voltage) dapat diatur dengan menggunakan potensiometer untuk menentukan siklus kerja (duty cycle) pada keluaran gelombang persegi / digital.
- Menggunakan osilator aktif frekuensi tinggi (125MHz) dengan kualitas presisi yang sangat baik (high precision active crystal oscillator)



Gambar 2.2 Blog Diagram Fungsi AD9850

2.7.4 Pengaturan Frekuensi Keluaran DDS

Pengaturan Frekuensi Keluaran DDS Pengaturan frekuensi keluaran misalnya pada frekuensi 77,667 MHz dilakukan dengan mengirimkan data secara serial melalui pin-7 DDS AD9850, data frequency tuning word yang harus dikirimkan dihitung menggunakan persamaan (4) yaitu:

$$M = \frac{77.667 \times 2^N}{180}$$

$$M = 18222068025 \rightarrow \boxed{748} \quad \boxed{118}$$

$$M = 18222068025 \rightarrow 0110111001101011011000100010101_2$$

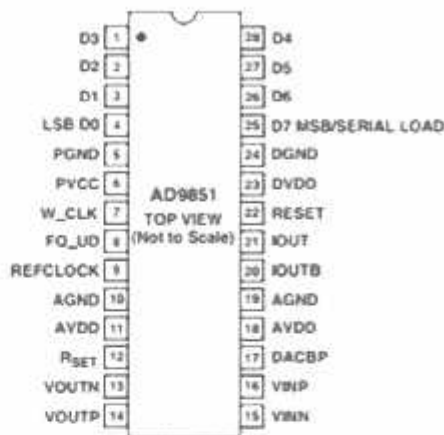
Tabel 2.1 Runtutan Data Frekuensi

- W32 : Pilihan faktor pengali frekuensi referensi
1 frekuensi referensi $6 \times \text{REFCLK}$
0 frekuensi referensi $1 \times \text{REFCLK}$
- W33 : kondisi logic 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°
- W32 : Pilihan faktor pengali frekuensi referensi
1 frekuensi referensi $6 \times \text{REFCLK}$
0 frekuensi referensi $1 \times \text{REFCLK}$
- W33 : 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 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 Gambar 10. 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 MSB. Posisi runtunan bit yang dimuat ke data input register untuk frekuensi 77,667

Data dalam bentuk runtunan bit inilah yang harus dimuat dalam data input register secara serial untuk menentukan frekuensi keluaran DDS dengan panjang 32bit, dengan urutan LSB dan terakhir MSB atau dari kiri ke kanan bermula dari W0 dan berakhir dengan W31, seperti pada tampilan table 2.2 sisa 8 bit yaitu bit

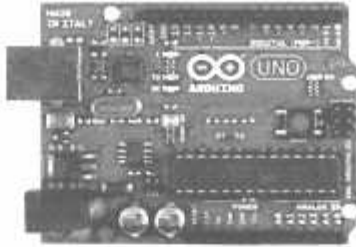
ke 32 sampai bit 39 digunakan untuk mengatur pilihan frekuensi referensi AD9850, Power – down dan berakhir pada bit LSB. Posisi runtutan bit yang dimuat ke data input register untuk frekuensi 10 MHz seperti ditampilkan pada table 2.2



Gambar 2.3 Pin Configuration IC AD9850

2.8 Arduino Uno R3

Uno Arduino Uno Arduino adalah board berbasis mikrokontroler pada ATmega328. 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 listrik 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.[5] Board Arduino Uno memiliki fitur-fitur baru sebagai berikut : - 1,0 pinout: tambah SDA dan SCL pin yang dekat ke pin aref dan dua pin baru lainnya ditempatkan dekat ke pin RESET, dengan IO 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 yang beroperasi dengan 3.3V. Yang kedua adalah pin tidak terhubung, yang di sediakan untuk tujuan pengembangannya.



Gambar 2.4 Board Arduino Uno

Deskripsi Arduino UNO: Tabel 2.2 Deskripsi Arduino Uno

Mikrokontroler	Atmega328
Operasi Voltage	5V
Input Voltage	7-12 V (Rekomendasi)
Input Voltage	6-20 V (limits)
I/O	14 pin (6 pin untuk PWM)
Arus	50 mA
Flash Memory	32KB
Bootloader	SRAM 2 KB
EEPROM	1 KB
Kecepatan	16 Mhz

2.8.1 Catu Daya Arduino Uno

dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber listrik dipilih secara otomatis. Eksternal (nonUSB) daya dapat datang baik dari AC-DC adaptor atau baterai. Adaptor ini dapat dihubungkan dengan cara menghubungkannya plug pusat-positif 2.1mm ke dalam board colokan listrik. Lead dari baterai dapat dimasukkan ke dalam header pin Gnd dan Vin dari konektor Power. Board dapat beroperasi pada pasokan daya dari 6 - 20 volt. Jika diberikan dengan kurang dari 7V, bagaimanapun, pin 5V dapat menyuplai kurang dari 5 volt dan board mungkin tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak board. Rentang yang dianjurkan adalah 7 - 12 volt.

Pin catu daya adalah sebagai berikut:

- VIN. Tegangan input ke board Arduino ketika menggunakan sumber daya eksternal (sebagai lawan dari 5 volt dari koneksi USB atau sumber daya lainnya diatur). Anda dapat menyediakan tegangan melalui pin ini, atau, jika memasok tegangan melalui colokan listrik, mengaksesnya melalui pin ini.

- 5V. Catu daya diatur digunakan untuk daya mikrokontroler dan komponen lainnya di board. Hal ini dapat terjadi baik dari VIN melalui regulator onboard, atau diberikan oleh USB .
- 3,3 volt pasokan yang dihasilkan oleh regulator on-board. Menarik arus maksimum adalah 50 mA.
- GND

2.8.2 Memory ATmega328

ini memiliki 32 KB dengan 0,5 KB digunakan untuk loading file. Ia juga memiliki 2 KB dari SRAM dan 1 KB dari EEPROM 2.1.4 Input & Output Masing-masing dari 14 pin digital pada Uno dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Mereka beroperasi di 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor pull-up internal dari 20-50 K Ω . Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data TTL serial. Pin ini terhubung ke pin yang sesuai dari chip ATmega8U2 USB-to-Serial TTL.
- Eksternal Interupsi: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, tepi naik atau jatuh, atau perubahan nilai. Lihat `attachInterrupt ()` fungsi untuk rincian.
- PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan 8-bit output PWM dengan `analogWrite` fungsi.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI.
- LED: 13. Ada built-in LED terhubung ke pin digital 13. Ketika pin adalah nilai TINGGI, LED menyala, ketika pin adalah RENDAH, itu off.

Uno memiliki 6 input analog, diberi label A0 melalui A5, masing-masing menyediakan 10 bit resolusi yaitu 1024 nilai yang berbeda. Secara default sistem mengukur dari tanah sampai 5 volt.

- TWI: A4 atau SDA pin dan A5 atau SCL pin. Mendukung komunikasi TWI
-

- Aref. Referensi tegangan untuk input analog. Digunakan dengan analogReference.
- Reset.

2.8.3 Komunikasi Arduino Uno

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). Sebuah ATmega16U2 pada saluran board ini komunikasi serial melalui USB dan muncul sebagai com port virtual untuk perangkat lunak pada komputer. Firmware Arduino menggunakan USB driver standar COM, dan tidak ada driver eksternal yang dibutuhkan. Namun, pada Windows, file. Inf 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. ATmega328 ini juga mendukung komunikasi I2C (TWI) dan SPI. Fungsi ini digunakan untuk melakukan komunikasi interface pada sistem.

2.8.4 Programming Arduino Uno

dapat diprogram dengan perangkat lunak Arduino. Pilih Arduino Uno dari Tool lalu sesuaikan dengan mikrokontroler yang digunakan. 13 Para ATmega328 pada Uno Arduino memiliki bootloader yang memungkinkan Anda untuk meng-upload program baru untuk itu tanpa menggunakan programmer hardware eksternal. Ini berkomunikasi menggunakan protokol dari bahas C. Sistem dapat menggunakan perangkat lunak FLIP Atmel (Windows) atau programmer DFU (Mac OS X dan Linux) untuk memuat firmware baru. Atau Anda dapat menggunakan header ISP dengan programmer eksternal .

2.8.5 Perangkat Lunak (Arduino IDE)

Lingkungan open-source Arduino memudahkan untuk menulis kode dan meng-upload ke board Arduino. Ini berjalan pada Windows, Mac OS X, dan Linux. Berdasarkan Pengolahan, avr-gcc, dan perangkat lunak sumber terbuka lainnya.



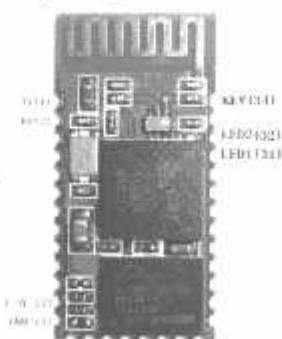
Gambar 2.5 Tampilan Framework Arduino UNO

2.8.6 Otomatis Software Reset

Tombol reset Uno Arduino dirancang untuk menjalankan program yang tersimpan didalam mikrokontroler dari awal. Tombol reset terhubung ke Atmega328 melalui kapasitor 100nf. Setelah tombol reset ditekan cukup lama untuk me-reset chip, software IDE Arduino dapat juga berfungsi untuk meng-upload program dengan hanya menekan tombol upload di software IDE Arduino.

2.9 Modul Bluetooth HC-05

Bluetooth adalah protokol komunikasi *wireless* yang bekerja pada frekuensi radio 2.4 GHz untuk pertukaran data pada perangkat bergerak seperti PDA, laptop, HP, dan lain-lain¹. Salah satu hasil contoh modul *Bluetooth* yang paling banyak digunakan adalah tipe HC-05. modul *Bluetooth* HC-05 merupakan salah satu modul *Bluetooth* yang dapat ditemukan dipasaran dengan harga yang relatif murah. Modul *Bluetooth* HC-05 terdiri dari 6 pin konektor, yang setiap pin konektor memiliki fungsi yang berbeda - beda. Untuk gambar *module bluetooth* dapat dilihat pada gambar 2.3 dibawah ini:



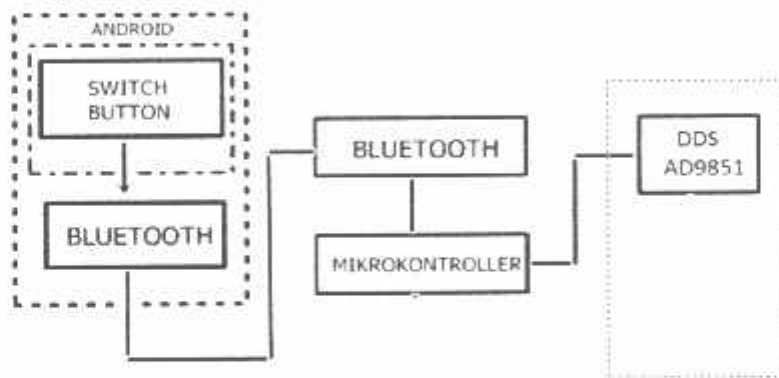
Gambar 2.6 Modul Bluetooth HC-05

Modul *Bluetooth* HC-05 dengan *supply* tegangan sebesar 3,3 V ke pin 12 modul *Bluetooth* sebagai VCC. Pin 1 pada modul *Bluetooth* sebagai transmitter. kemudian pin 2 pada *Bluetooth* sebagai *receiver*.

BAB III PERANCANGAN SISTEM

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 kontrol pembangkit frekuensi yaitu design sistem yang diimplementasikan pada sistem *embedded 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. Berikut adalah desain sistem yang akan diterapkan pada skripsi ini



Gambar 3.1 Diagram Blok Sistem

3.2 Prinsip Kerja

Pada perancangan dan pembuatan alat pengontrol modul DDS AD9850 sebagai pembangkit frekuensi berbasis rduino Uno. Modul Direct Digital Synthesizer (DDS) AD9850 adalah modul pembangkit gelombang yang menggunakan IC AD9850 CMOS 125MHz. AD9850 yang mampu membangkitkan frekuensi sampai 40MHz, Kelebihan DDS yaitu memiliki step 1Hz, hingga 10MHz yang dapat diatur.

Direct Digital Synthesizer (DDS) merupakan 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 dibangkitkan dapat diubah sesuai kebutuhan penggunaanya.

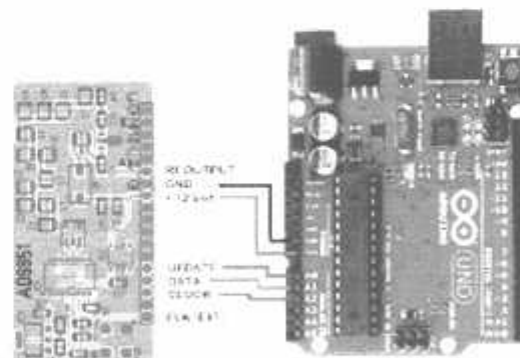
Aplikasi android frekuensi digunakan sebagai masukan nilai frekuensi yang ingin dibangkitkan kemudian dikirimkan kedalam media Bluetooth , Bluetooth HC05 sebagai penerima dan terhubung pada Arduino Uno ,Arduino Uno berfungsi sebagai pemroses data dari android sebagai unit masukan (input) untuk selanjutnya mengirimkan data dari perhitungan dan pemrosesan tersebut ke modul DDS AD9850, pengiriman data secara serial dengan cara mengirimkan ke 40-bit pada biner ari 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 dan selanjutnya AD9850 akan membangkitkan frekuensi secara digital yang berupa gelombang sinus, sesuai dengan data yang dikirim oleh Arduino Uno.

Sistem yang akan dirancang akan dibagi menjadi dua bagian utama yaitu perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*).

3.3 Perancangan sistem perangkat keras

3.3.1 Perancangan Rangkaian Modul AD9850

Rangkaian koneksi antara arduino dengan modul AD9850 ditunjukkan pada gambar 3.2, dalam perancangan ini pin pin yang harus dihubungkan dengan pin arduino antara lain:



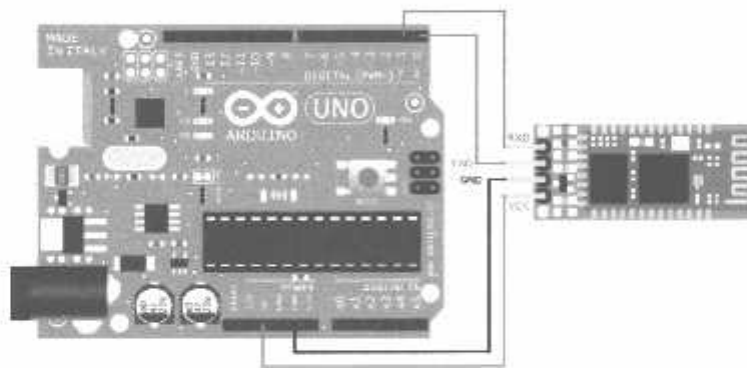
Gambar 3.2 Rangkaian Koneksi Arduino dengan AD9850

Keterangan :

- Pin GND pada modul AD9850 dihubungkan pada pin GND Arduino uno
- Pin RESET pada modul AD9850 dihubungkan pada pin 13 Arduino uno
- Pin Data pada modul AD9850 dihubungkan pada pin12 Arduino uno
- Pin W_CLK pada modul AD9850 dihubungkan pada pin 11 Arduino uno
- Pin FQ_UD pada modul AD9850 dihubungkan pada pin10 Arduino uno
- Pin Vcc pada modul A9850 dihubungkan pada Vcc Arduino 5V

3.3.2 Desain Rangkaian Bluetooth

Komunikasi dalam sistem ini ,yaitu Komunikasi antara android dengan arduino dengan menggunakan media bluetooth



Gambar 3.2 Rangkaian komunikasi bluetooth

Keterangan :

- Pin GND pada Bluetooth HC05 dihubungkan pada pin GND Arduino uno
- Pin VCC pada Bluetooth HC05 dihubungkan pada pin 5v Arduino uno
- Pin RX pada Bluetooth HC05 dihubungkan pada RX Arduino uno
- Pin TX pada Bluetooth HC05dihubungkan pada TX Arduino uno

3.4 Perancangan Perangkat Lunak Sistem

Penuisan program akan dilaksanakan ketika bagian rangkaian elektronika telah selesai dilakukan, penulisan program pada arduino dan pada pengontrol modul DDS AD9850 sebagai pembangkit frekuensi ini dibagi menjadi dua bagian yaitu:

1. Perancangan dan program Arduino untuk memproses data masukan (input) untuk selanjutnya mengirimkan data hasil perhitungan dan pemrosesan tersebut ke modul DDS AD9850 sebagai pembangkit frekuensi.

2. Penulisan program komunikasi Bluetooth
3. Perancangan program Aplikasi Android

3.4.1 Perancangan dan program Arduino

Rangkaian Kontrol disini menggunakan Arduino Uno yang akan mengolah data masukan dan keluaran pada setiap pin nya. Arduino yang digunakan dalam perancangan ini merupakan komponen utama karena merupakan komponen yang akan mengatur keseluruhan system agar dapat bekerja dengan baik dan optimal.

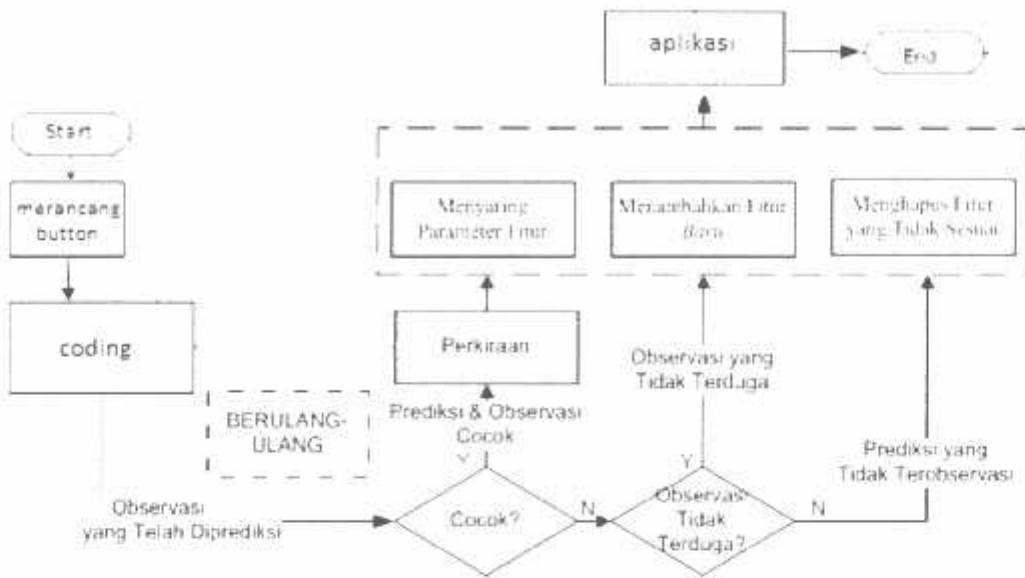
Arduino akan memproses masukan dan keluaran dengan cara mengaktifkan masing – masing pin pada arduino. Untuk mengaktifkan pin- pin yang terdapat di dalam arduino tersebut dilakukan dengan menggunakan perangkat lunak (Software) yang dimiliki oleh arduino. Untuk mengaktifkan Arduino Uno maka perlu diberikan tegangan 5Volt DC yang dapat langsung diambil dari port USB pada computer, atau dapat juga diambil dari supplay tegangan DC lain

3.5 Perancangan Aplikasi

Rancangan Aplikasi Rancangan aplikasi ini akan memberikan gambaran dari aplikasi yang dibuat dengan menggunakan desain sistem yang berorientasi objek, yaitu UML, serta digambarkan menggunakan Star UML. Adapun desain yang dibuat antara lain:

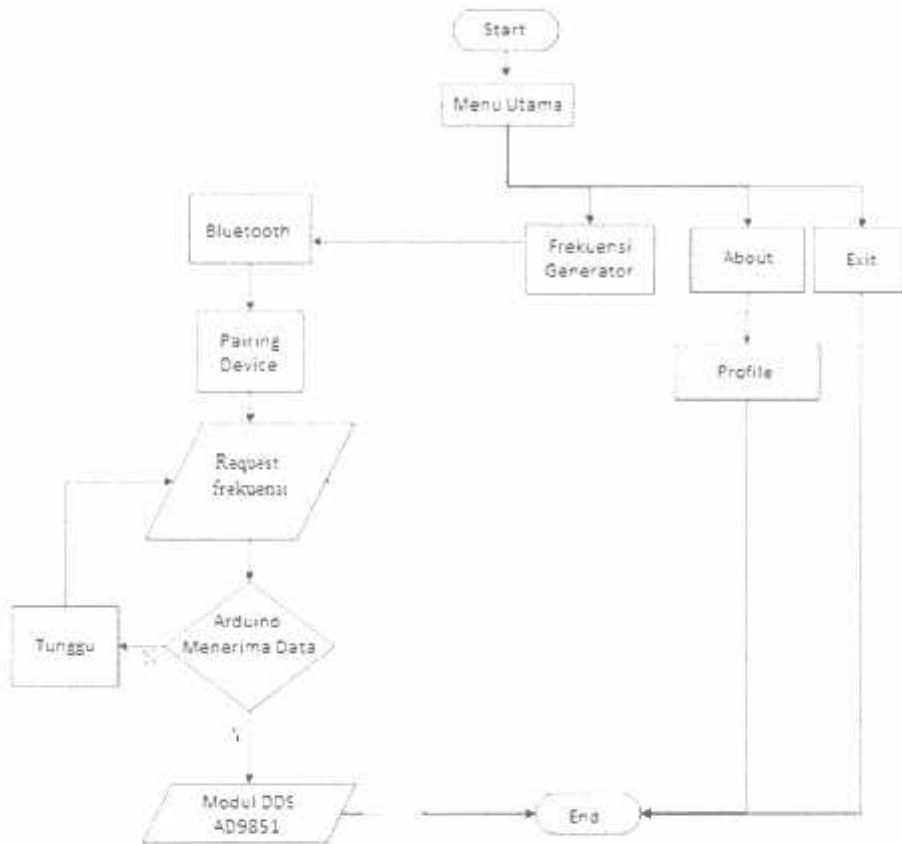
- a. Flowchart
 - b. Use Case Diagram
 - d. Activity Diagram
 - e. User Interface
-

3.5.1 Flowchart



Gambar3.4 flowchart perancangan Aplikasi

Keterangan : Pada flowchart diatas dapat diketahui bahwa alur yang dijadikan pedoman dalam perancangan aplikasi yaitu dimulai dari Start kemudian merancang button yang dibutuhkan dilanjutkan dengan menulis program dari button yang telah dirancang, apabila program yang ditulis sudah sesuai maka dapat dicompile ,jika program tidak sesuai maka dapat menghapusnya dan menambahkan program yang sesuai agar button buton yang dirancang dapat bekerja dengan optimal.

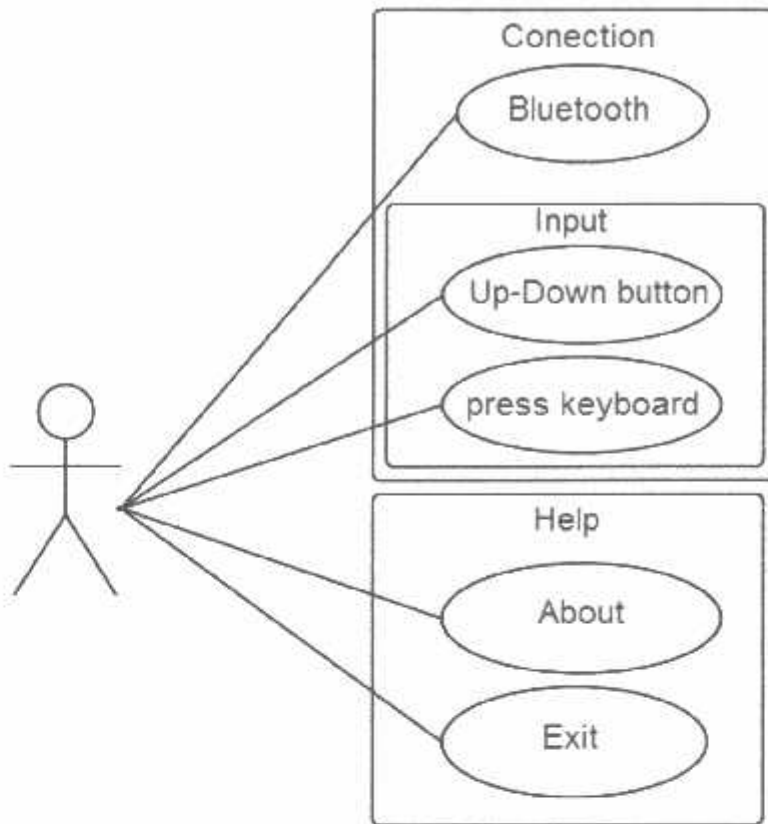


Gambar 3.4 Flowchart Sistem Aplikasi

Keterangan : Pada flowchart diatas dapat diketahui bahwa alur dalam sistem aplikasi yaitu, untuk bisa melakukan input frekuensi harus melakukan koneksi bluetooth , request frekuensi dilanjutkan arduino menerima dan memproses data ,apabila data cocok dapat diteruskan pada modul DDS AD9850 untuk membangkitkan frekuensi.

3.5.2 Perancangan Use Case Diagram

Usecase diagram merupakan salah satu diagram untuk memodelkan aspek perilaku sistem. Masing-masing diagram usecase menunjukkan sekumpulan usecase, aktor dan hubungannya. Diagram usecase digunakan untuk memvisualisasikan, menspesifikasikan dan mendokumentasikan kebutuhan sistem.

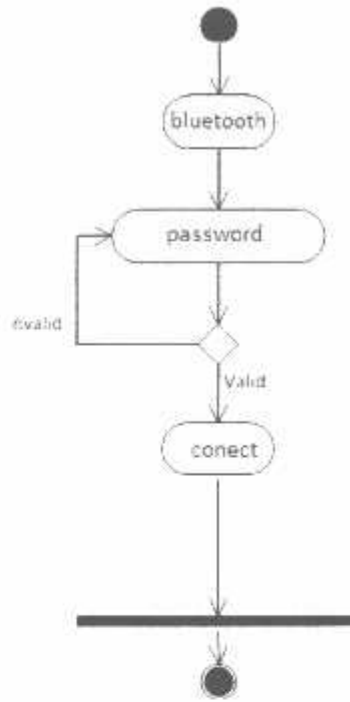


Gambar 4.1 *Use case* Sistem yang diajukan

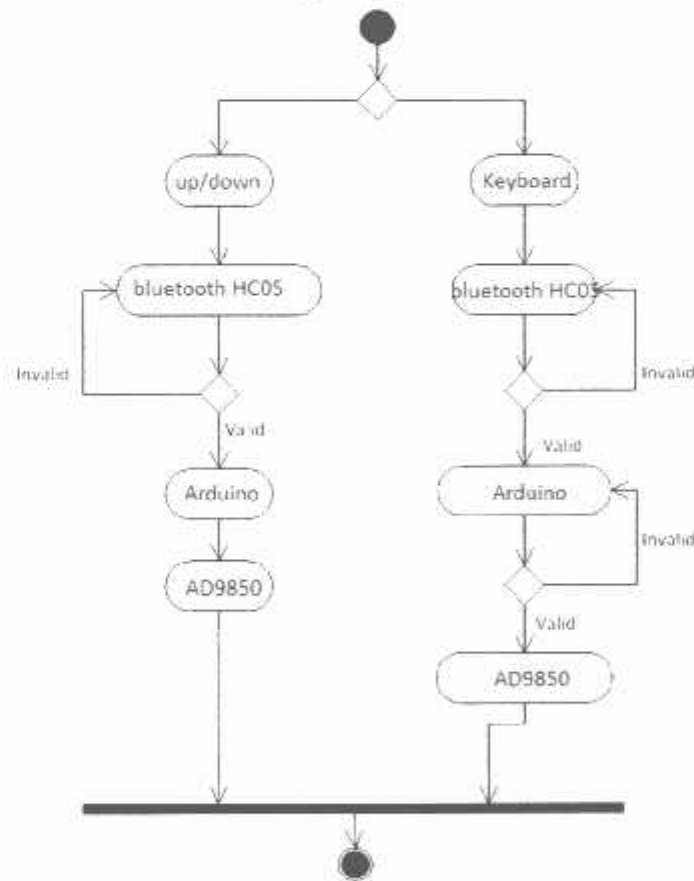
Dari usecase diagram diatas dapat ditunjukkan bahwa kebutuhan user yaitu conection yaitu dengan fitur visualisasi bluetooth, input frekuensi dengan fitur visualisasi Up Down button dan keyboard android, help dengan visualisasi about dan exit.

3.5.3 *Activity Diagram*

Untuk menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, digambarkan *Activity diagram* agar dapat menjelaskan tahapan sistem masing-masing fitur yang dirancang.



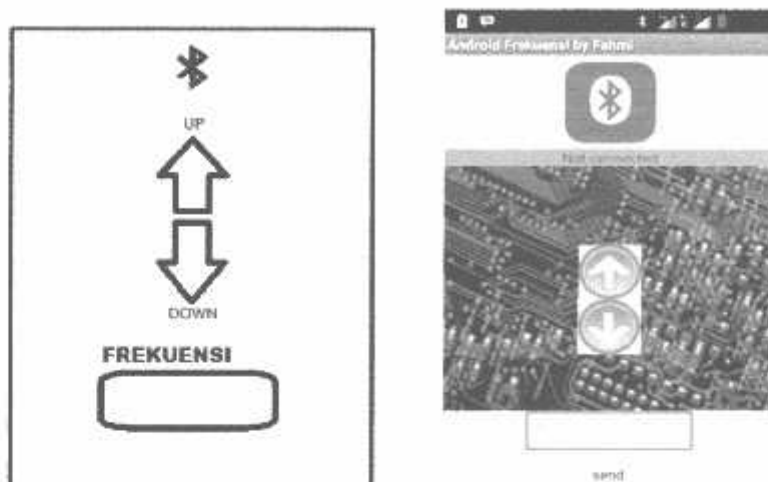
Gambar 4.2 Activity Diagram Sistem Koneksi



Gambar 4.3 Activity Diagram Sistem Aplikasi

3.5.4 User Interface

Aplikasi yang akan dibuat dapat didesain atau dimodelkan sebagai berikut :



Gambar 3.3 tampilan desain aplikasi

Keterangan:

- Button Bluetooth sebagai button untuk koneksi dan pairing Bluetooth
- Button UP-DOWN Untuk menaikkan dan menurunkan frekuensi secara per step
- Keypad Untuk merequest frekuensi dengan cara menginputkan melalui tombol keypad

BAB IV PENGUJIAN DAN PEMBAHASAN SISTEM

4.1 Pendahuluan

Pada bab ini ditunjukkan 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 Aplikasi.
- ✓ Pengujian pada Bluetooth.
- ✓ Pengujian Keseluruhan Sistem Pembangkit Frekuensi

4.2 Pengujian Rangkaian Mikrokontroler Arduino UNO

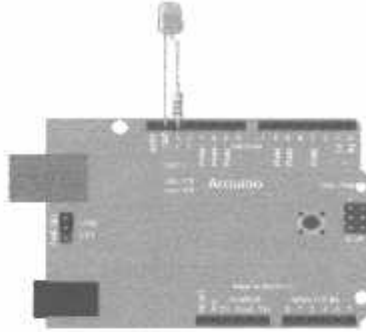
Untuk pengujian modul mikrokontroler Arduino UNO berkerja dengan baik akan dilakukan pengujian pada jalur-jalur *port* yang dimiliki oleh mikrokontroler Arduino UNO. Untuk pengujian modul dilakukan pengisian program terlebih dahulu menggunakan Arduino IDE (Integrated Development Environment). Dengan meng*compile* 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 V
5. Resistor

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 skrip di bawah.

```
sketch_jun06a | Arduino 1.0.5-r2
File Edit Sketch Tools Help
sketch_jun06a $
int led1 = 13; // Lampu led internal pada pin 13 arduino
void setup()
{
  pinMode(led1, OUTPUT); // pin 13 sebagai output
}
void loop()
{
  digitalWrite(led1, HIGH); // led on
  delay(1000); // tunggu 1 detik
  digitalWrite(led1, LOW); // led off
  delay(1000); // tunggu 1 detik
}
```

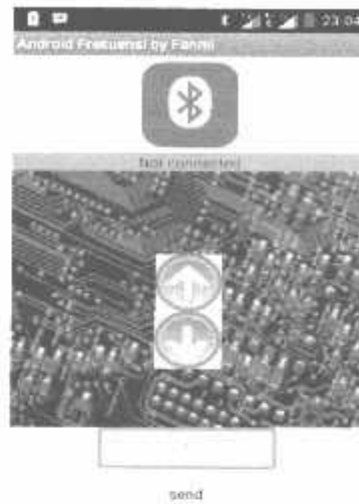
Gambar 4.2 Scrip pengujian arduino

4.2.3 Hasil pengujian

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

4.3 Pengujian Apikasi Android

Hasil pembuatan *apikasi android* frekuensi dengan menu tampilan pada gambar 4.3 berikut ini :



Gambar 4.3 Hasil tampilan aplikasi android frekuensi

Keterangan:

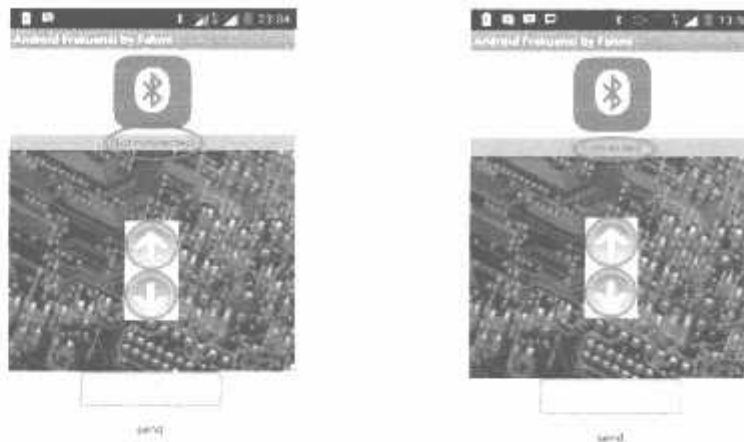
1. Button bluetooth
2. Label status koneksi bluetooth
3. Frekuensi yang dibangkitkan
4. Button up
5. Button down
6. Teks button dengan keypad android
7. Send hasil penulisan dengan keypad

Apabila klik button 1, maka program akan memasuki tampilan seperti pada gambar 4.4 untuk melakukan pencarian bluetooth dan melakukan pairing device.



Gambar 4.4 Hasil tampilan pairing perangkat bluetooth

Apabila perangkat bluetooth sudah ditemukan dan sudah terkoneksi maka akan berubah unuk status koneksinya seperti tampak pada gambar 4.3 berikut ;



Gambar 4.5 Hasil tampilan pairing perangkat bluetooth

Dengan berubahnya status koneksi bluetooth maka dapat disimpulkan pada button bluetooth pada aplikasi android dapat berjalan dengan baik.

4.4 Pengujian Modul Bluetooth android Ke Bluetooth HC-05

Pengujian ini dilakukan untuk mengetahui apakah Bluetooth Slave dapat menerima informasi yang dikirim oleh Bluetooth Master, sejauh mana Bluetooth Master dapat mengirim data pada Bluetooth Slave yang akan dituliskan pada mikrokontroler dalam berbagai kondisi. Data hasil pengujian jarak yang memungkinkan untuk berkomunikasi antar Bluetooth Master dengan Bluetooth Slave ditunjukkan dalam Tabel 1 hingga table 3 berikut ini.

Tabel 4.1. Pengujian Pengiriman Data Antara Bluetooth Master dengan Bluetooth Slave tanpa penghalang

No.	Jarak (cm)	Data Terkirim				% error
		P1	P2	P3	P4	
1	50 cm	Ya	Ya	Ya	Ya	0%
2	330 cm	Ya	Ya	Ya	Ya	0%
3	550 cm	Ya	Ya	Ya	Ya	0%
4	730 cm	Ya	Ya	Ya	Ya	0%
5	950 cm	Ya	Ya	Ya	Ya	0%
6	1030 cm	Ya	Ya	Ya	Ya	0%
7	1040 cm	Ya	Tidak	Ya	Ya	25%
8	1045 cm	Ya	Tidak	Ya	Ya	25%
9	1048 cm	Ya	Tidak	Ya	Tidak	50%
10	1052 cm	Tidak	Tidak	Tidak	Tidak	100%

Tabel pengujian di atas dapat menyimpulkan jika Bluetooth Slave dengan kondisi tanpa penghalang dapat menerima data hingga 1048 cm dengan persentase tidak dapat menerima data 25%. Sedangkan di atas 1048 cm (dalam hal ini 1052 cm), Bluetooth Slave sudah tidak dapat menerima data 100%.

Tabel 4.2. Pengujian Pengiriman Data Antara Bluetooth Master dengan Bluetooth Slave menggunakan penghalang plastik pada Bluetooth Slave.

No.	Jarak (cm)	Data Terkirim				% <i>error</i>
		P1	P2	P3	P4	
1	50 cm	Ya	Ya	Ya	Ya	0%
2	330 cm	Ya	Ya	Ya	Ya	0%
3	550 cm	Ya	Ya	Ya	Ya	0%
4	730 cm	Ya	Ya	Ya	Ya	0%
5	950 cm	Ya	Ya	Ya	Ya	0%
6	1030 cm	Ya	Ya	Ya	Ya	0%
7	1040 cm	Tidak	Ya	Ya	Tidak	50%
8	1045 cm	Tidak	Ya	Tidak	Tidak	75%
9	1048 cm	Tidak	Tidak	Tidak	Tidak	100%
10	1052 cm	Tidak	Tidak	Tidak	Tidak	100%

Pengujian menggunakan penghalang plastik pada Bluetooth Slave memperoleh hasil jika maksimal penerimaan data adalah 1045 cm, dengan catatan pada jarak ini persentase kesalahan yang dapat terjadi adalah 75%.

Tabel 4.3. Pengujian Pengiriman Data Antara Bluetooth Master dengan Bluetooth Slave menggunakan penghalang plastik pada kedua modul Bluetooth.

No.	Jarak (cm)	Data Terkirim				% <i>error</i>
		P1	P2	P3	P4	
1	50 cm	Ya	Ya	Ya	Ya	0%
2	330 cm	Ya	Ya	Ya	Ya	0%
3	550 cm	Ya	Ya	Ya	Ya	0%
4	730 cm	Ya	Ya	Ya	Ya	0%
5	950 cm	Ya	Tidak	Ya	Ya	25%
6	1030 cm	Tidak	Tidak	Ya	Tidak	75%
7	1040 cm	Tidak	Tidak	Tidak	Tidak	100%
8	1045 cm	Tidak	Tidak	Tidak	Tidak	100%
9	1048 cm	Tidak	Tidak	Tidak	Tidak	100%
10	1052 cm	Tidak	Tidak	Tidak	Tidak	100%

Berdasarkan data yang didapatkan di atas, maka penerimaan data Bluetooth Slave dengan kondisi terdapat plastik yang membungkus masing-masing Bluetooth maksimal pada jarak 1030 cm dengan catatan 75% dari percobaan, Bluetooth Slave tidak dapat menerima data.

4.5 Analisa Pengujian Keseluruhan Sistem

Pada pengujian keseluruhan sistem pembangkit frekuensi ini, untuk mengetahui Sebarapa 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*. dan dengan menggunakan Aplikasi Android Frekuensi yang telah dibuat. Pada pengujian pembangkit frekuensi ini, diambil beberapa frekuensi yang mewakili rentang frekuensi dari 1 Hz – 40 MHz.

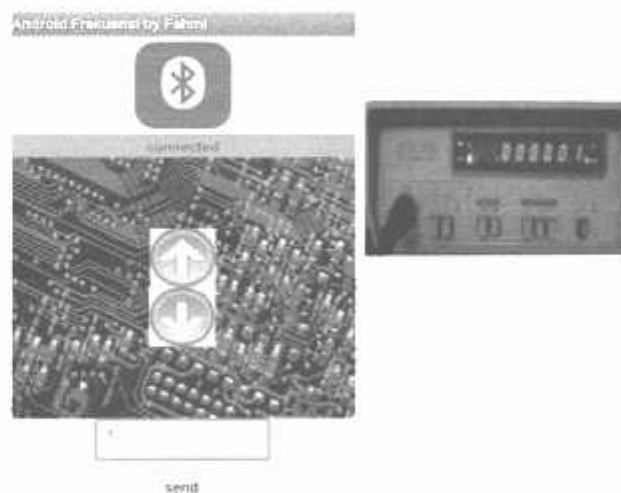
1. Hasil Pengujian Frekuensi 1 Hz

Hasil pengujian untuk pembangkit frekuensi keluaran 1 Hz dari modul AD9850 seperti pada Gambar 4.6. Pada Gambar 4.6.a merupakan hasil pembangkitan frekuensi 1 Hz dari Modul AD9850, Gambar 4.6.b memperlihatkan nilai frekuensi yang diukur menggunakan *GW frequency Counter GFC-8010G* diperoleh nilai sebesar 1 Hz, dengan presentase kesalahannya sebesar 0 %.

Hasil pengujian untuk pembangkit frekuensi menggunakan *GW frequency Counter*

(a) Frekuensi masukan sebesar 1 Hz

(b) Frekuensi keluaran sebesar 1 Hz



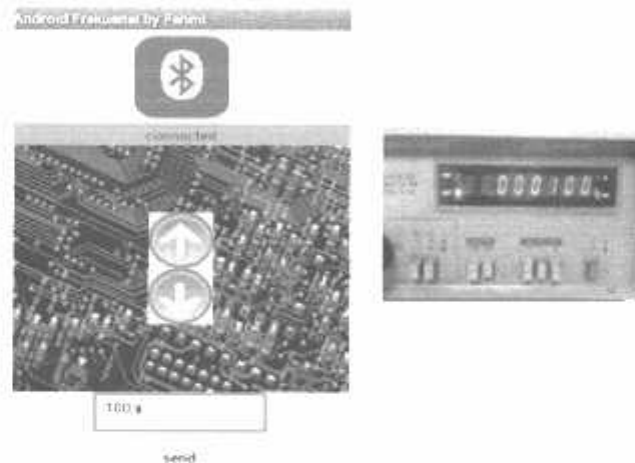
Gambar 4.6 Hasil Pengujian Frekuensi 1 Hz

2. Hasil Pengujian Frekuensi 100 Hz

Hasil pengujian untuk pembangkit frekuensi keluaran 100 Hz dari modul AD9850 seperti pada Gambar 4.7. Pada Gambar 4.7.a merupakan hasil pembangkitan frekuensi 100 Hz dari Modul AD9850, Gambar 4.7.b memperlihatkan nilai frekuensi yang diukur menggunakan *GW frequency Counter GFC-8010G* diperoleh nilai sebesar 100 Hz, dengan presentase kesalahannya sebesar 0 %.

(a) Frekuensi masukan sebesar 100 Hz

(b) Frekuensi keluaran sebesar 100 Hz



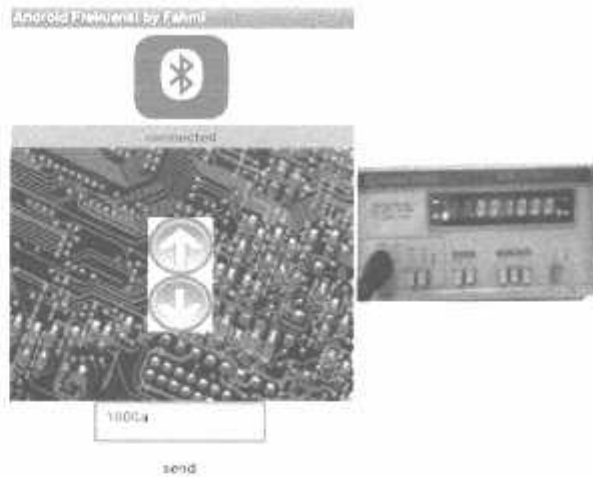
Gambar 4.7 Hasil Pengujian Frekuensi 100 Hz

3. Hasil Pengujian Frekuensi 1 KHz

Hasil pengujian untuk pembangkit frekuensi keluaran 1 KHz dari modul AD9850 seperti pada Gambar 4.8. Pada Gambar 4.8.a merupakan hasil pembangkitan frekuensi 1 KHz dari Modul AD9850, Gambar 4.8.b memperlihatkan nilai frekuensi yang diukur menggunakan *GW frequency Counter GFC-8010G* diperoleh nilai sebesar 1 KHz, dengan presentase kesalahannya sebesar 0 %.

(a) Frekuensi masukan sebesar 1000Hz

(b) Frekuensi keluaran sebesar 1000 Hz

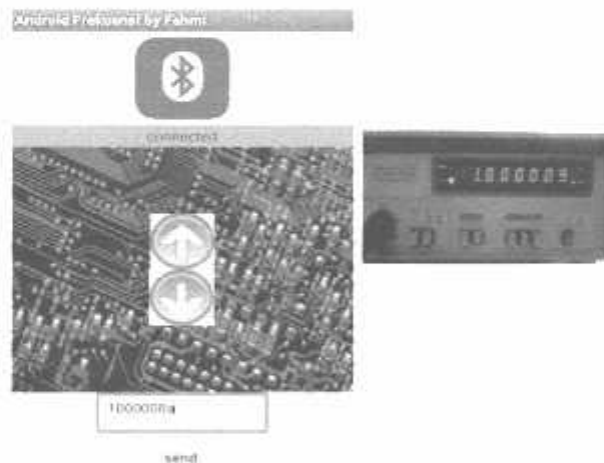


Gambar 4.8 Hasil Pengujian Frekuensi 1000 Hz

4. Hasil Pengujian Frekuensi 1 MHz

Hasil pengujian untuk pembangkit frekuensi keluaran 1 MHz dari modul AD9850 seperti pada Gambar 4.9. Pada Gambar 4.9.a merupakan hasil pembangkitan frekuensi 1 MHz dari Modul AD9850, Gambar 4.9.b memperlihatkan nilai frekuensi yang diukur menggunakan *GW frequency Counter GFC-8010G* diperoleh nilai sebesar 1,000009 MHz, dengan presentase kesalahannya sebesar 0,09 %.

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



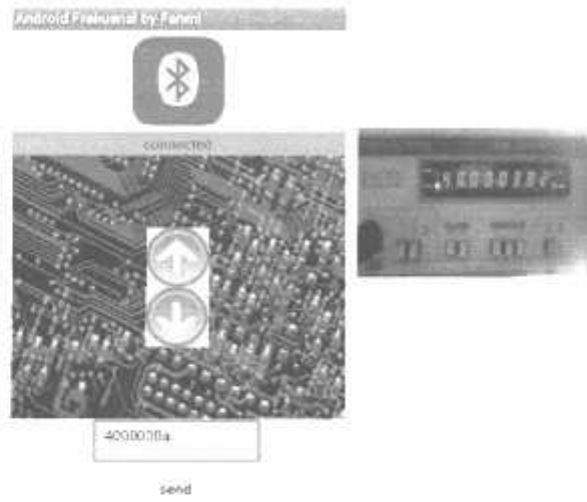
Gambar 4.9 Hasil Pengujian Frekuensi 1 MHz

5. Hasil Pengujian Frekuensi 40 MHz

Hasil pengujian untuk pembangkit frekuensi keluaran 40 MHz dari modul AD9850 seperti pada Gambar 4.10.

(a) Frekuensi masukan sebesar 40MHz

(b) Frekuensi keluaran sebesar 40 KHz



Gambar 4.10 Hasil pengujian 40MHz.

Tabel 4.4 Hasil Pengujian frekuensi

No	AD9850 (Hz)	Freq Counter (Hz)	Selisih (%)	No	AD9850 (Hz)	Freq Counter (Hz)	Selisih (%)
1	1	1	0	26	500000	500004	0,79
2	10	10	0	27	755000	755007	0,9271
3	25	25	0	28	800000	800008	0,1
4	50	50	0	29	1000000	1000009	0,9
5	90	90	0	30	1500000	1500015	0,1
6	100	100	0	31	4000000	4000038	0,95
7	125	125	0	32	5000000	5000048	0,96
8	150	150	0	33	8500000	8500082	0,9640
9	200	200	0	34	10000000	10000097	0,97
10	350	350	0	35	12000000	12000118	0,983
11	500	500	0	36	15000000	15000147	0,98
12	750	750	0	37	17500000	17500171	0,977
13	1000	1000	0	38	19000000	19000184	0,984
14	1500	1500	0	39	20000000	20000193	0,965
15	5000	5000	0	40	21000000	21000204	0,9714
16	10000	10000	0	41	23500000	23500229	0,9744
17	15000	15000	0	42	25000000	25000250	0,1
18	25000	25000	0	43	28000000	28000282	0,9714
19	26500	26500	0	44	30000000	30000301	0,1033
20	50000	50000	0	45	32500000	32500325	0,1
21	60000	60000	0	46	34500000	34500328	0,9507
22	85500	85500	0	47	35000000	35000330	0,9428
23	100000	100000	0	48	36500000	36500345	0,9452
24	125000	125001	0,08	49	39000000	39000371	0,9512
25	250000	250002	0,08	50	40000000	40000383	0,9575

4.5.1 Analisa Hasil Pengujian

Dari data hasil pengujian diatas dapat dihitung persentase selisih pembangkit frekuensi dari modul AD9850 yang diinputkan melalui aplikasi android dan output diukur menggunakan GW Frequency Counter GFC-8010G dengan perhitungan persamaan yang digunakan adalah:

Selisih(%)=

$$\frac{\text{Freq Counter} - \text{inputan pada android}}{\text{freq counter}} \times 100\%$$

Dari hasil pengujian dapat dihitung total selisih dengan menggunakan rumus:

Rata rataselisih (%)

$$= \frac{\text{jumlah semua selisih}}{\text{jumlah percobaan}}$$

$$= \frac{18,7273}{50}$$

Jadi jumlah rata rata selisih adalah =0,374546

- Nilai rata rata selisih dari rentang frekuensi 1Hz hingga 100KHz dari percobaan diatas dengan persamaan :

$$\text{Rata-rata Selisih} = \frac{\text{Total Selisih}}{\text{Jumlah Percobaan}}$$

$$= \frac{0}{23}$$

$$= 0$$

- Nilai rata rata selisih dari rentang frekuensi 100KHz hingga 40MHz dari percobaan diatas dengan persamaan :

$$\text{Rata-rata Selisih} = \frac{\text{Total Selisih}}{\text{Jumlah Percobaan}}$$

$$= \frac{18,7273}{27}$$

$$= 0,6936037$$



BAB V PENUTUP

5.1 Kesimpulan

Setelah dilakukan perancangan, pengujian dan analisa sistem, maka pada perancangan pengontrol modul DDS AD9850 sebagai pembangkit frekuensi dengan menggunakan aplikasi android frekuensi dan pengukuran output menggunakan GW Frequency Counter dapat disimpulkan sebagai berikut :

1. Untuk Nilai selisih rata rata dari rentang frekuensi 1 Hz hingga 100KHz Frekuensi Dalam keadaan stabil dengan prosentase kesalahan sebesar 0
2. Untuk Nilai selisih rata rata dari rentang frekuensi 100 KHz hingga 40MHz Frekuensi Dalam keadaan stabil dengan prosentase kesalahan sebesar 0,6936037
3. Bentuk Gelombang keluaran dari AD9850 dengan rentang frekuensi antara 1Hz – 40 MHz gelombang sinus dalam keadaan sempurna meskipun memiliki prosentase kesalahan



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&mim=11030189>
2. D. Kurniawan, *ATMega8+DDS (Direct Digital Synthesizer), [Online]*. Available: <http://elektronika.web.id/elkav2/index.php?topic=978.msg16112#msg16112>
3. N. Safaa, *Penrograman Aplikasi Mobile Smartphone dan Tablet Berbasis Android*, Informatika, Bandung, 2011.
4. B. Kamboj dan R. Mehra, "Efficient FPGA implementation of direct digital frequency synthesizer for software radios,"
5. Arduino Uno [Online] <http://blog.langitrobotika.com/article/robotika/pengertian-arduino-uno/.html>, diakses 1 April 2016



LAMPIRAN

SURAT PERNYATAAN ORISINALITAS

Yang bertandatangan di bawah ini :

Nama : Fahmi Noor Cholik

NIM : 12.12.525

Program Studi : T.Elektro S-1

Konsentrasi : Teknik Komputer

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

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

Malang, 20 Agustus 2016

Yang membuat Pernyataan,



Fahmi
Fahmi Noor Cholik
NIM : 1212525



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

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

41 (PERSERO) MALANG
JNK NIAGA MALANG

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

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Fahmi Noor Cholik
NIM : 1212525
Program Studi : TEKNIK ELEKTRO S-1
Konsentrasi : TEKNIK KOMPUTER
Judul Skripsi : **PERANCANGAN DAN PEMBUATAN APLIKASI
PENGONTROL MODUL DIRECT DIGITAL
SYNTHESIZER AD9850 SEBAGAI PEMBANGKIT
FREKUENSI BERBASIS ANDROID**

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

Hari : **Senin**
Tanggal : **8 Agustus 2016**
Dengan Nilai : **83,35**

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. Aryuanto Soetedjo, ST, MT
NIP.Y. 1030800417

Penguji II

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



PERSETUJUAN PERBAIKAN SKRIPSI

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

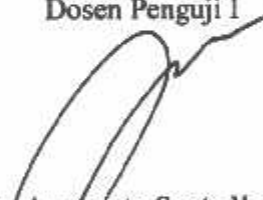
Hari : Senin
Tanggal : 8 Agustus 2016

Telah dilakukan perbaikan skripsi oleh :

Nama : Fahmi Noor Cholik
NIM : 1212525
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer
Judul Skripsi : **Perancangan dan Pembuatan Aplikasi Pengontrol Modul Direct Digital Synthesizer AD9850 Sebagai Pembangkit Frekuensi Berbasis Android**

No	Materi Perbaikan	Keterangan
1.	Menu tampilan android untuk teks disesuaikan ukuranya	✓

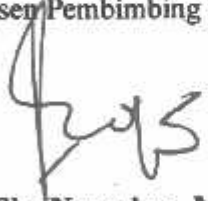
Dosen Penguji I


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

Dosen Pembimbing I


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

Dosen Pembimbing 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 : Senin
Tanggal : 8 Agustus 2016

Telah dilakukan perbaikan skripsi oleh :

Nama : Fahmi Noor Cholik
NIM : 1212525
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer
Judul Skripsi : **Perancangan dan Pembuatan Aplikasi Pengontrol Modul Direct Digital Synthesizer AD9850 Sebagai Pembangkit Frekuensi Berbasis Android**

No	Materi Perbaikan	Keterangan
1.	Daftar pustaka diurutkan abjad	
2.	Tambah kesimpulan	

Dosen Penguji I

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

Dosen Pembimbing I

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

Dosen Pembimbing II

Ir. Eko Nurcahyo, MT
NIP.Y. 1028700172

LAMPIAN SKRIP PROGRAM ARDUINO

```
/*
 * A simple single freq AD9850 Arduino test script
 * Original AD9851 DDS sketch by Andrew Smallbone at
 www.rocketnumbernine.com
 * Modified for testing the inexpensive AD9850 ebay DDS modules
 * Pictures and pinouts at nr80.dhlpilotcentral.com
 * 9850 datasheet at http://www.analog.com/static/imported-
 files/data_sheets/AD9850.pdf
 * Use freely
 */

#define W_CLK 11 // Pin 6 - connect to AD9850 module word load
clock pin (CLK)
#define FQ_UD 10 // Pin 5 - connect to freq update pin (FQ)
#define DATA 12 // Pin 7 - connect to serial data load pin
(DATA)
#define RESET 13 // Pin 4 - connect to reset pin (RST).

#define pulseHigh(pin) {digitalWrite(pin, HIGH); digitalWrite(pin,
LOW); }

char interupsi, data[16] = {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '};
unsigned int baris = 0;
double freq, freq_buffer[16];
bool enter = 0;

// 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);

    Serial.begin(9600);

// pulseHigh(RESET);
// pulseHigh(W_CLK);
// pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet
page 12 figure 10
// sendFrequency(1000000); // freq
}

void loop() {
    while(Serial.available() == 1)
    {
        data[baris] = Serial.read();
        if(data[baris] == 'a')
        {
            enter = 1;

```

```

        break;
    }
    convert();
    baris++;
}

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];
    pulseHigh(RESET);
    pulseHigh(W_CLK);
    pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet
page 12 figure 10
    while(1)
    {
        sendFrequency(freq);
        Serial.println(freq);
        if(Serial.available() == 1)
        {
            interupsi = Serial.read();
            if(interupsi == 'u')
            {
                freq++;
                break;
            }

            if(interupsi == 'd')
            {
                freq--;
            }
        }
    }
}

```

```

        break;
    }

    if(interupsi == 'b')
    {
        for(int x = 0; x < 16; x++)
        {
            data[x] = ' ';
        }

        for(int x = 0; x < 16; x++)
        {
            freq_buffer[x] = 0;
        }

        freq = 0;
        baris = 0;
        enter = 0;
        break;
    }
}

}

if(enter == 1 && baris == 2)//ketika nilai frekuensi yg dimasukkan
ada 2 digit
{
    freq = (freq_buffer[0] * 10) + freq_buffer[1];
    pulseHigh(RESET);
    pulseHigh(W_CLK);
    pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet
page 12 figure 10
    while(1)
    {
        sendFrequency(freq);
    }
}

```

```
// Serial.println(freq);
if(Serial.available() == 1)
{
    interupsi = Serial.read();
    if(interupsi == 'u')
    {
        freq++;
        break;
    }

    if(interupsi == 'd')
    {
        freq--;
        break;
    }

    if(interupsi == 'b')
    {
        for(int x = 0; x < 16; x++)
        {
            data[x] = ' ';
        }

        for(int x = 0; x < 16; x++)
        {
            freq_buffer[x] = 0;
        }

        freq = 0;
        baris = 0;
        enter = 0;
        break;
    }
}
}
```

```

}

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];
    pulseHigh(RESET);
    pulseHigh(W_CLK);
    pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet
page 12 figure 10
    while(1)
    {
        sendFrequency(freq);
//        Serial.println(freq);
        if(Serial.available() == 1)
        {
            interupsi = Serial.read();
            if(interupsi == 'u')
            {
                freq++;
            }

            if(interupsi == 'd')
            {
                freq--;
            }

            if(interupsi == 'b')
            {
                for(int x = 0; x < 16; x++)
                {
                    data[x] = ' ';
                }

                for(int x = 0; x < 16; x++)

```

```

        {
            freq_buffer[x] = 0;
        }

        freq = 0;
        baris = 0;
        enter = 0;
        break;
    }
}

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];
    pulseHigh(RESET);
    pulseHigh(W_CLK);
    pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet
page 12 figure 10
    while(1)
    {
        sendFrequency(freq);
//        Serial.println(freq);
        if(Serial.available() == 1)
        {
            interupsi = Serial.read();
            if(interupsi == 'u')
            {
                freq++;
            }

            if(interupsi == 'd')
            {

```

```

        freq--;
    }

    if(interupsi == 'b')
    {
        for(int x = 0; x < 16; x++)
        {
            data[x] = ' ';
        }

        for(int x = 0; x < 16; x++)
        {
            freq_buffer[x] = 0;
        }

        freq = 0;
        baris = 0;
        enter = 0;
        break;
    }
}

}

}

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];
    pulseHigh(RESET);
    pulseHigh(W_CLK);
    pulseHigh(PQ_UD); // this pulse enables serial mode - Datasheet
page 12 figure 10
    while(1)
    {
        sendFrequency(freq);
    }
}

```

```
// Serial.println(freq);
if(Serial.available() == 1)
{
    interupsi = Serial.read();
    if(interupsi == 'u')
    {
        freq++;
    }

    if(interupsi == 'd')
    {
        freq--;
    }

    if(interupsi == 'b')
    {
        for(int x = 0; x < 16; x++)
        {
            data[x] = ' ';
        }

        for(int x = 0; x < 16; x++)
        {
            freq_buffer[x] = 0;
        }

        freq = 0;
        baris = 0;
        enter = 0;
        break;
    }
}
}
```

```
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];  
    pulseHigh(RESET);  
    pulseHigh(W_CLK);  
    pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet  
page 12 figure 10  
    while(1)  
    {  
        sendFrequency(freq);  
//        Serial.println(freq);  
        if(Serial.available() == 1)  
        {  
            interupsi = Serial.read();  
            if(interupsi == 'u')  
            {  
                freq++;  
            }  
  
            if(interupsi == 'd')  
            {  
                freq--;  
            }  
  
            if(interupsi == 'b')  
            {  
                for(int x = 0; x < 16; x++)  
                {  
                    data[x] = ' ';  
                }  
  
                for(int x = 0; x < 16; x++)  
                {
```

```

        freq_buffer[x] = 0;
    }

    freq = 0;
    baris = 0;
    enter = 0;
    break;
}
}
}

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];
    pulseHigh(RESET);
    pulseHigh(W_CLK);
    pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet
page 12 figure 10
    while(1)
    {
        sendFrequency(freq);
//        Serial.println(freq);
        if(Serial.available() == 1)
        {
            interupsi = Serial.read();
            if(interupsi == 'u')
            {
                freq++;
            }

            if(interupsi == 'd')
            {

```

```

        freq--;
    }

    if(interupsi == 'b')
    {
        for(int x = 0; x < 16; x++)
        {
            data[x] = ' ';
        }

        for(int x = 0; x < 16; x++)
        {
            freq_buffer[x] = 0;
        }

        freq = 0;
        baris = 0;
        enter = 0;
        break;
    }
}

}

}

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];

    pulseHigh(RESET);
    pulseHigh(W_CLK);
    pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet
page 12 figure 10

    while(1)
    {

```

```
sendFrequency(freq);  
// Serial.println(freq);  
if(Serial.available() == 1)  
{  
    interupsi = Serial.read();  
    if(interupsi == 'u')  
    {  
        freq++;  
    }  
  
    if(interupsi == 'd')  
    {  
        freq--;  
    }  
  
    if(interupsi == 'b')  
    {  
        for(int x = 0; x < 16; x++)  
        {  
            data[x] = ' ';  
        }  
  
        for(int x = 0; x < 16; x++)  
        {  
            freq_buffer[x] = 0;  
        }  
  
        freq = 0;  
        naris = 0;  
        enter = 0;  
        break;  
    }  
}  
}
```

```

    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];

        pulseHigh(RESET);

        pulseHigh(W_CLK);

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

        while(1)
        {
            sendFrequency(freq);
//            Serial.println(freq);
            if(Serial.available() == 1)
            {
                interupsi = Serial.read();
                if(interupsi == 'u')
                {
                    freq++;
                }

                if(interupsi == 'd')
                {
                    freq--;
                }

                if(interupsi == 'b')
                {
                    for(int x = 0; x < 16; x++)
                    {
                        data[x] = ' ';
                    }

                    for(int x = 0; x < 16; x++)

```

```

        {
            freq_buffer[x] = 0;
        }

        freq = 0;
        baris = 0;
        enter = 0;
        break;
    }
}

}

if(enter == 1 && baris == 10)//ketika nilai frekuensi yg dimasukkan
ada 10 digit
{
    freq = (freq_buffer[0] * 1000000000) + (freq_buffer[1] *
100000000) + (freq_buffer[2] * 10000000) + (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];

    pulseHigh(RESET);
    pulseHigh(W_CLK);
    pulseHigh(FQ_UD); // this pulse enables serial mode - Datasheet
page 12 figure 10
    while(1)
    {
        sendFrequency(freq);
//        Serial.println(freq);
        if(Serial.available() == 1)
        {
            interupsi = Serial.read();
            if(interupsi == 'u')
            {
                freq++;
            }
        }
    }
}

```

```
    if(interupsi == 'd')
    {
        freq--;
    }

    if(interupsi == 'b')
    {
        for(int x = 0; x < 16; x++)
        {
            data[x] = ' ';
        }

        for(int x = 0; x < 16; x++)
        {
            freq_buffer[x] = 0;
        }

        freq = 0;
        baris = 0;
        enter = 0;
        break;
    }
}
}
}
}
}
```

```
void convert()//program untuk mengkonversi data dari string ke integer
{
    if(data[baris] == 48)
    {
        freq_buffer[baris] = 0;
    }
}
```

```
if(data[baris] == 49)
{
    freq_buffer[baris] = 1;
}

if(data[baris] == 50)
{
    freq_buffer[baris] = 2;
}

if(data[baris] == 51)
{
    freq_buffer[baris] = 3;
}

if(data[baris] == 52)
{
    freq_buffer[baris] = 4;
}

if(data[baris] == 53)
{
    freq_buffer[baris] = 5;
}

if(data[baris] == 54)
{
    freq_buffer[baris] = 6;
}

if(data[baris] == 55)
{
    freq_buffer[baris] = 7;
}
```

```
if(data[baris] == 56)
{
    freq_buffer[baris] = 8;
}
```

```
if(data[baris] == 57)
{
    freq_buffer[baris] = 9;
}
```

```
}
```



FEATURES

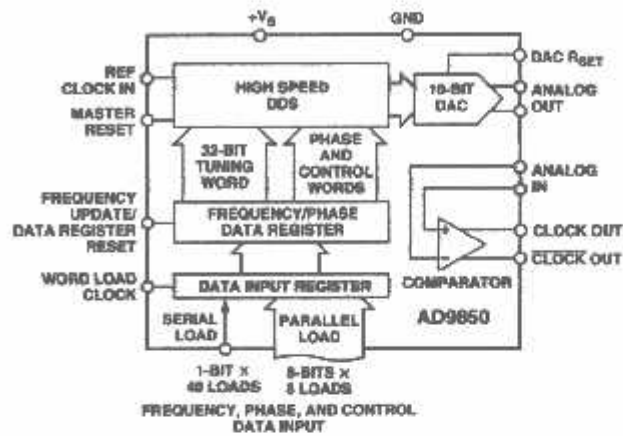
- 125 MHz Clock Rate
- On-Chip High Performance DAC and High Speed Comparator
- DAC SFDR > 50 dB @ 40 MHz A_{OUT}
- 32-Bit Frequency Tuning Word
- Simplified Control Interface: Parallel Byte or Serial Loading Format
- Phase Modulation Capability
- +3.3 V or +5 V Single Supply Operation
- Low Power: 380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

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 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

FUNCTIONAL BLOCK DIAGRAM


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 2-5 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. E

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 which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 World Wide Web Site: <http://www.analog.com>
Fax: 781/326-8703 © Analog Devices, Inc., 1998

D9850—SPECIFICATIONS (V_C = +5 V ± 5% except as noted, R_{SET} = 3.9 kΩ)

Parameter	Temp	Test Level	AD9850BRS			Units
			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
PULSEWIDTH HIGH/LOW						
+5 V Supply	+25°C	IV	3.2			ns
+3.3 V Supply	+25°C	IV	4.1			ns
CLK OUTPUT CHARACTERISTICS						
Full-Scale Output Current						
R _{SET} = 3.9 kΩ	+25°C	V		10.24		mA
R _{SET} = 1.95 kΩ	+25°C	V		20.48		mA
Main Error	+25°C	I	-10		+10	% FS
Main 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		kΩ
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 ± 50 kHz	+25°C	IV		80		dBc
40.13579 MHz ± 200 kHz	+25°C	IV		77		dBc
4.513579 MHz ± 50 kHz/20.5 MHz CLK	+25°C	IV		84		dBc
4.513579 MHz ± 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			kΩ
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
CLK OUTPUT CHARACTERISTICS						
Clock Output Duty Cycle (Clk Gen. Config.)	+25°C	IV		50 ± 10		%

Parameter	Temp	Test Level	AD9850BRS			Units
			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	I	3.0			V
Logic "0" Voltage	+25°C	I			0.4	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

NOTES

*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			Units
			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} min. Pulsewidth High)	Full	IV	3.5			ns
t _{WL} (W _{CLK} min. Pulsewidth 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 _{FD} (FQ_UD Min. Delay After W _{CLK})	Full	IV	7.0			ns
t _{RH} (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

NOTES

*Control functions are asynchronous with CLKIN.

Specifications subject to change without notice.

19850

SOLUTE MAXIMUM RATINGS*

Maximum Junction Temperature	+165°C
Digital Inputs	+6 V
Digital Output Continuous Current	-0.7 V to +V _S
Digital Output Current	5 mA
Digital Output Current	30 mA
Storage Temperature	-65°C to +150°C
Operating Temperature	-40°C to +85°C
Lead Temperature (Soldering 10 sec)	+300°C
Power Dissipation Thermal Impedance	82°C/W

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

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.

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.

Caution 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.



ORDERING GUIDE

Model	Temperature Range	Package Description	Package Option
AD9850BRS	-40°C to +85°C	Shrink Small Outline (SSOP)	RS-28

Table I. Lead Function Descriptions

Pin No.	Mnemonic	Function
4-1, 28-25	D0-D7	8-Bit Data Input. This is the 8-bit data port for iteratively loading the 32-bit frequency and 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 will update 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}	This is the 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	The 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 will go to Cosine 0 after additional clock cycles—see Figure 19.

PIN CONFIGURATIONS



19850—Typical Performance Characteristics

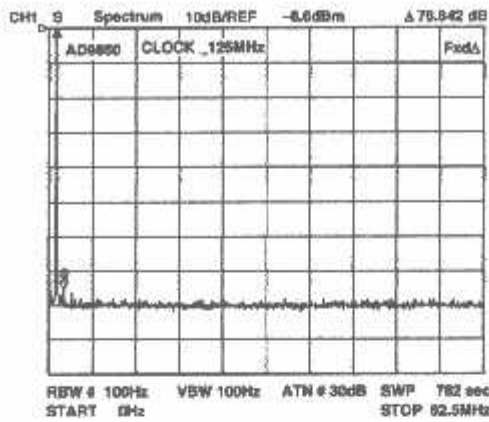


Figure 1. SFDR, $CLKIN = 125\text{ MHz}/f_{OUT} = 1\text{ MHz}$

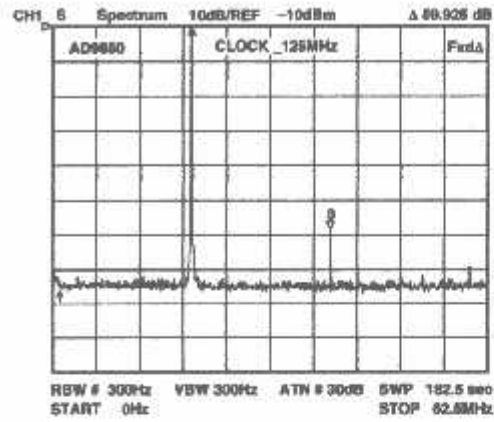


Figure 4. SFDR, $CLKIN = 125\text{ MHz}/f_{OUT} = 20\text{ MHz}$

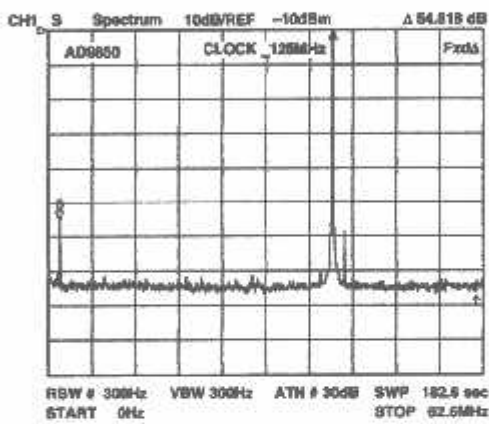


Figure 2. SFDR, $CLKIN = 125\text{ MHz}/f_{OUT} = 41\text{ MHz}$

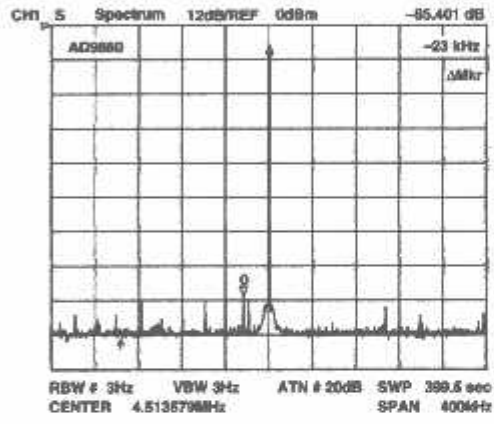


Figure 5. SFDR, $CLKIN = 20.5\text{ MHz}/f_{OUT} = 4.5\text{ MHz}$

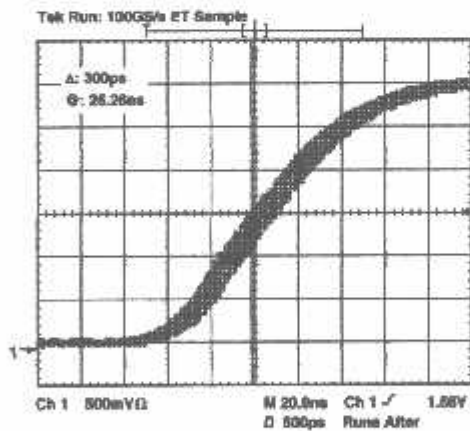


Figure 3. Typical Comparator Output Jitter, AD9850 Configured as Clock Generator w/42 MHz LP Filter (40 MHz A_{OUT} /125 MHz $CLKIN$)

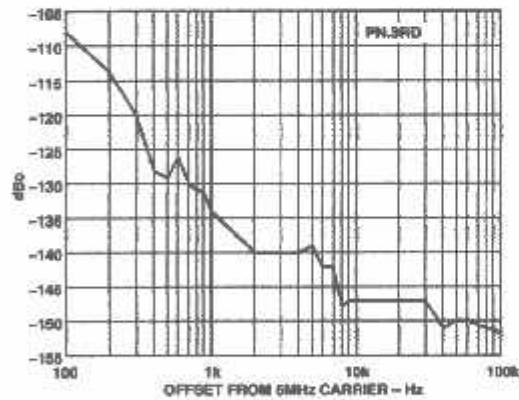


Figure 6. Output Residual Phase Noise (15 MHz A_{OUT} /125 MHz $CLKIN$)

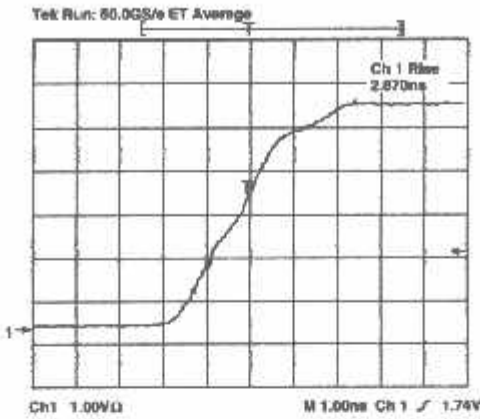


Figure 7. Comparator Output Rise Time (5 V Supply/15 pF Load)

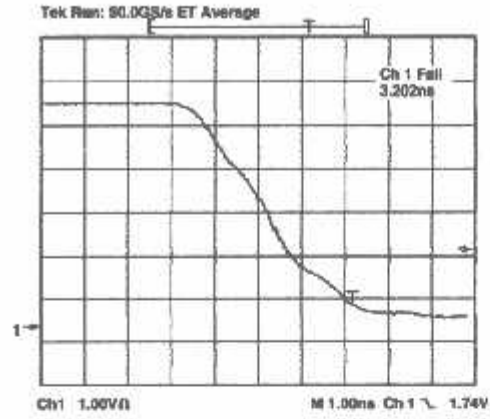


Figure 10. Comparator Output Fall Time (5 V Supply/15 pF Load)

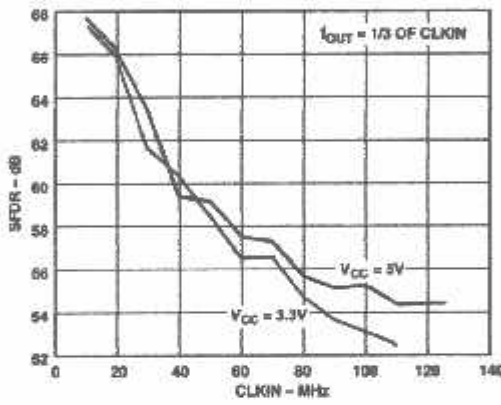


Figure 8. SFDR vs. CLKIN Frequency ($A_{OUT} = 1/3$ of CLKIN)

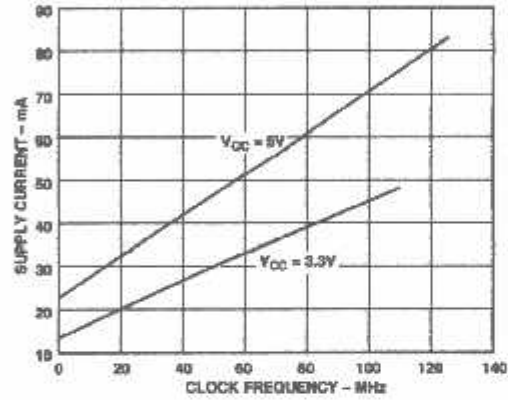


Figure 11. Supply Current vs. CLKIN Frequency ($A_{OUT} = 1/3$ of CLKIN)

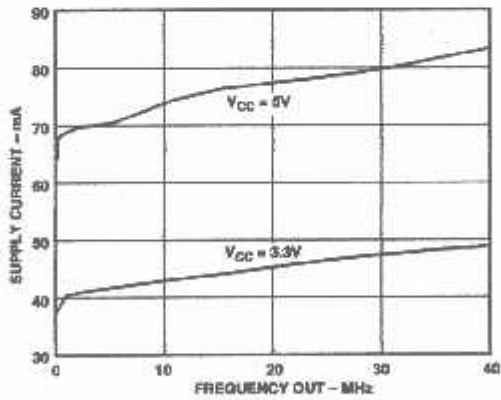


Figure 9. Supply Current vs. A_{OUT} Frequency (CLKIN = 125/110 MHz for 5 V/3.3 V Plot)

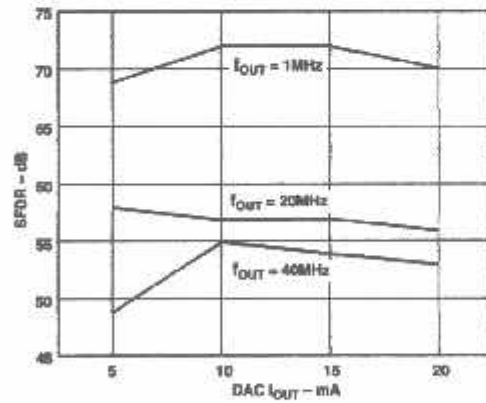


Figure 12. SFDR vs. DAC I_{OUT} ($A_{OUT} = 1/3$ of CLKIN)

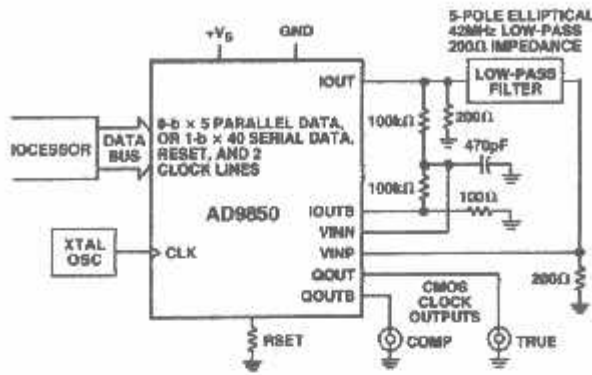


Figure 13. Basic AD9850 Clock Generator Application with Low-Pass Filter

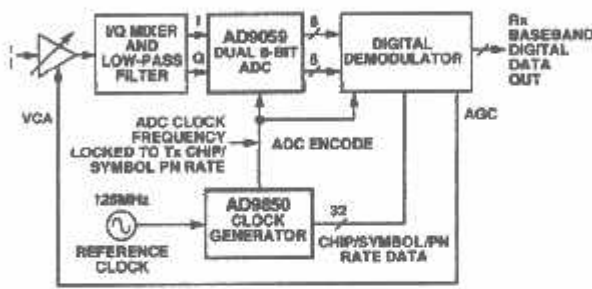


Figure 14. AD9850 Clock Generator Application in a Spread-Spectrum Receiver

THEORY OF OPERATION AND APPLICATION

AD9850 uses direct digital synthesis (DDS) technology, in 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 onboard 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 output frequency under full digital control. DDS also allows 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 and can be changed.

The basic functional block diagram and signal flow of the AD9850 configured as a clock generator is shown in Figure 16.

DDS circuitry is basically a digital frequency divider function where incremental resolution is determined by the frequency of 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

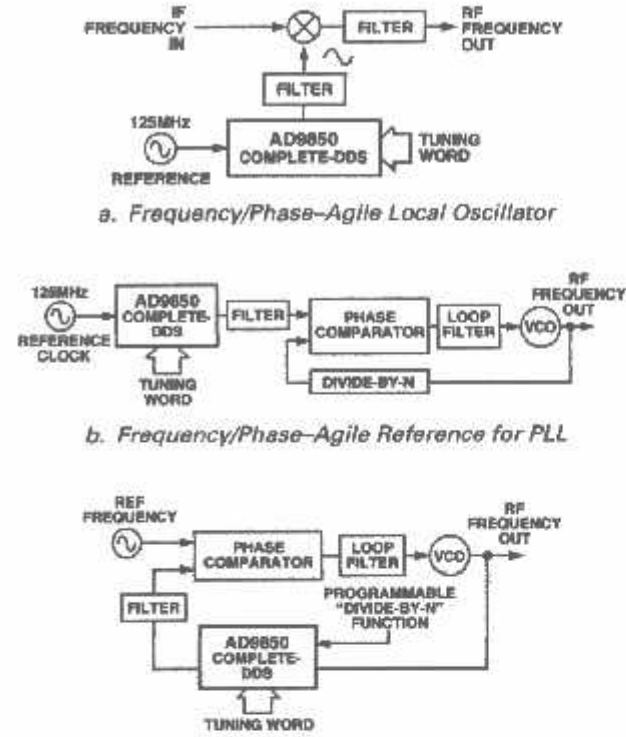


Figure 15. AD9850 Complete-DDS Synthesizer in Frequency Up-Conversion Applications

receives a clock pulse. When the counter overflows it wraps around, making the phase accumulator's output contiguous. The frequency tuning word sets the modulus of the counter that effectively determines the size of the increment (Δ Phase) that gets 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 = value of 32-bit tuning word
 CLKIN = input reference clock frequency in MHz
 f_{OUT} = 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

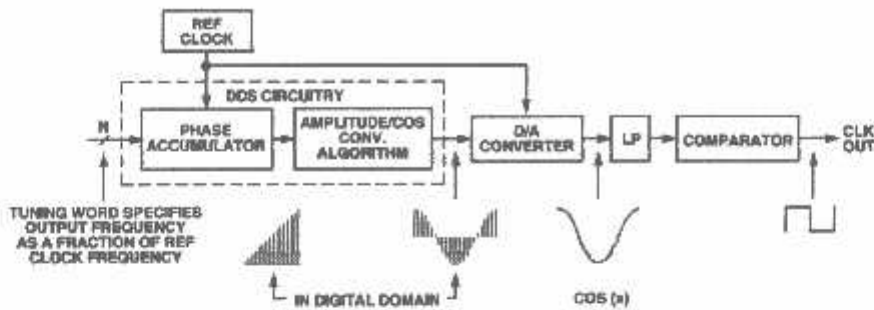


Figure 16. Basic DDS Block Diagram and Signal Flow of AD9850

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. Since the output of the 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 17.

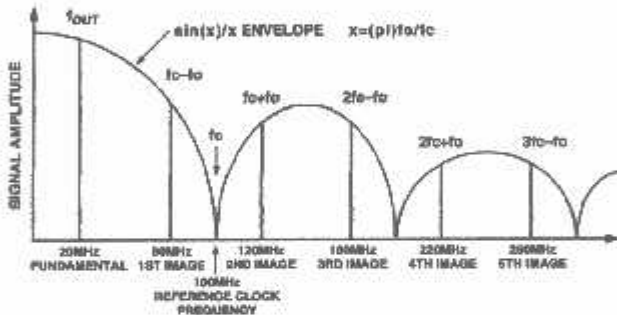


Figure 17. Output Spectrum of a Sampled Signal

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 $f_o/\text{Ref Clk}$ 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.

A good rule-of-thumb for applying the AD9850 as a clock generator is to limit the selected output frequency to $<33\%$ of Reference Clock frequency, thereby avoiding generating aliased signals that fall within, or close to, the output band of interest (generally de-selected output frequency). This practice will ease 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 generating the comparator reference midpoint from the differential DAC output as shown in Figure 13.

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 *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).

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 18–24.

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

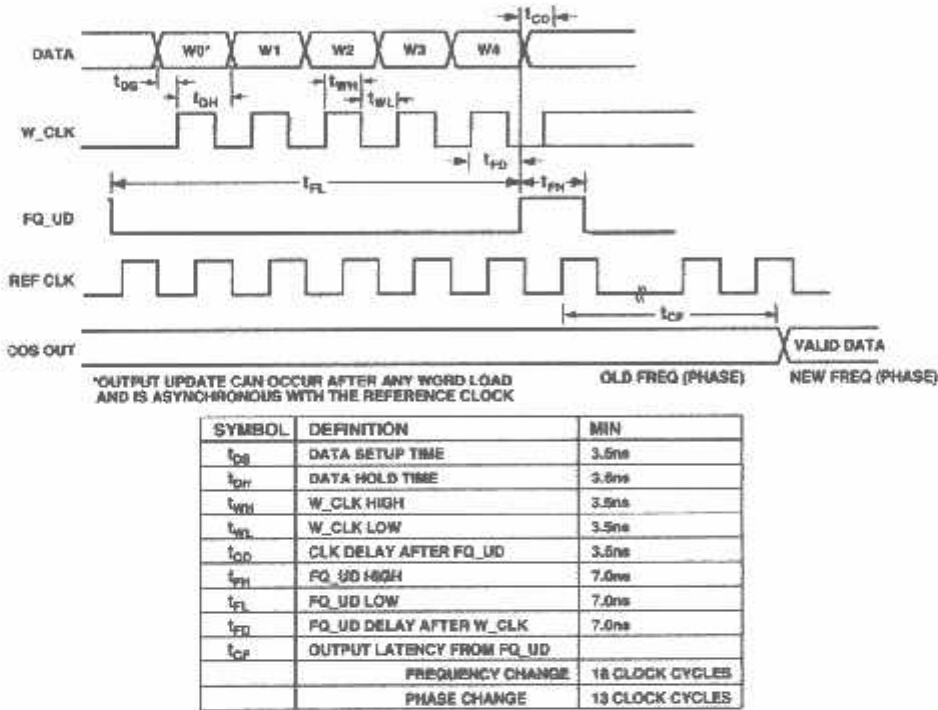
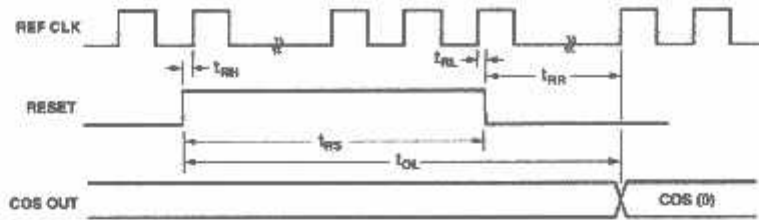


Figure 18. Parallel-Load Frequency/Phase Update Timing Sequence

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

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



SYMBOL	DEFINITION	MIN SPEC
t_{RH}	CLK DELAY AFTER RESET RISING EDGE	3.8ns
t_{RL}	RESET FALLING EDGE AFTER CLK	3.5ns
t_{WR}	RECOVERY FROM RESET	2 CLK CYCLES
t_{RS}	MINIMUM RESET WIDTH	5 CLK CYCLES
t_{OL}	RESET OUTPUT LATENCY	13 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 19. Master Reset Timing Sequence

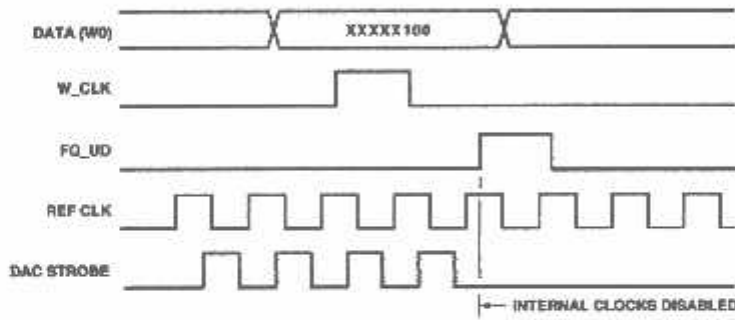


Figure 20. Parallel-Load Power-Down Sequence/Internal Operation

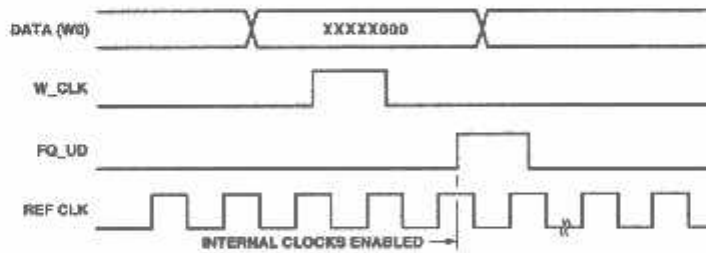


Figure 21. Parallel-Load Power-Up Sequence/Internal Operation

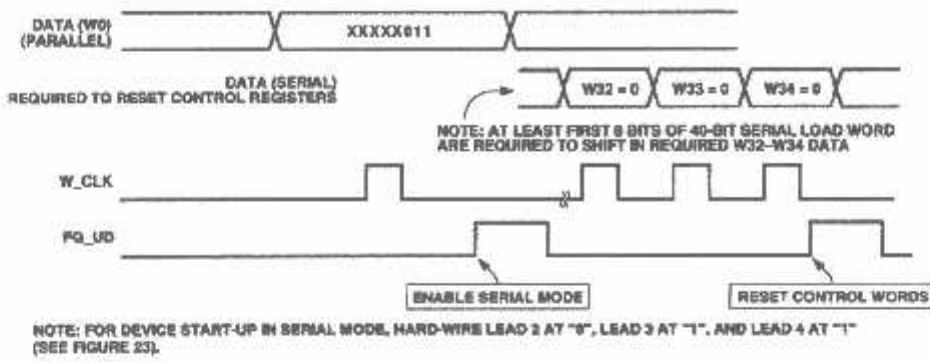


Figure 22. Serial-Load Enable Sequence

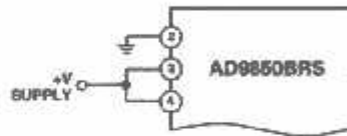


Figure 23. Leads 2-4 Connection for Default Serial-Mode Operation

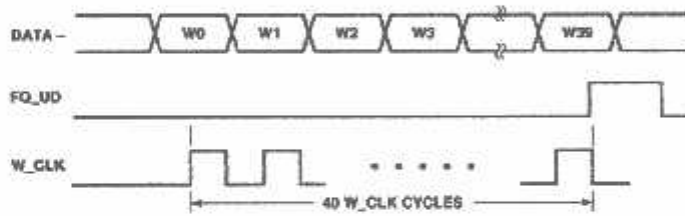


Figure 24. Serial-Load Frequency/Phase Update Sequence

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		

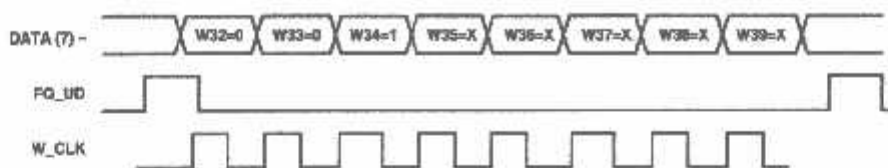


Figure 25. Serial-Load Power-Down Sequence

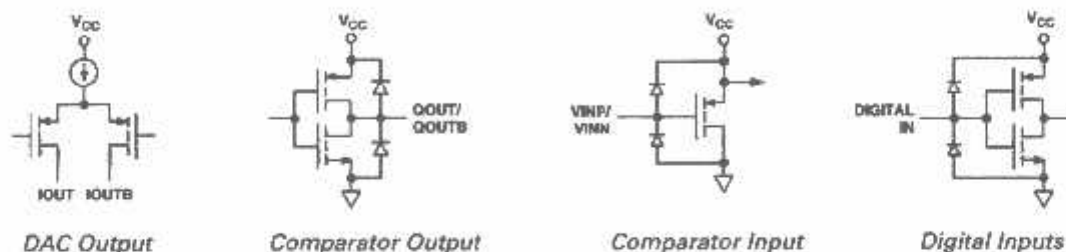


Figure 26. AD9850 I/O Equivalent Circuits

PCB LAYOUT INFORMATION

The AD9850/CGPCB and AD9850/FSPCB evaluation boards (Figures 27–30) 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 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 as these will 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 will reduce 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.

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 intended for applications where the device will primarily be used as 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 28 for electrical schematic of AD9850/FSPCB). The AD9850/CGPCB is intended for 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 29 for electrical schematic of AD9850/CGPCB).

Both versions of the AD9850 evaluation boards 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" floppy provided with the evaluation board contains an executable file that loads and displays the AD9850 function-selection screen. The evaluation board may 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 onboard crystal clock source is used, remove R2.

All trademarks are the property of their respective holders.

9850

9850 Evaluation Board Instructions

Required hardware/software:

1 compatible computer operating in a Windows environment
1 printer port, 3.5" floppy drive and Centronics compatible
printer cable.

1 AL 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.

1 9850 evaluation board software disk and AD9850/FSPCB or
AD9850/CGPCB evaluation board.

1 V_{DD} voltage supply

Step 1:

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

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

Apply power to AD9850 evaluation board. The AD9850 is
powered separately from the connector marked "DUT +V."
AD9850 may be powered with 3.3 V to +5 V.
Connect external 50 ohm clock or remove R2 and apply a high
input clock such as a crystal "can" oscillator.

Execute the file called 9850REV2.EXE and execute that program.
The monitor should display a "control panel" to allow operation of
the AD9850 evaluation board.

Step 2:

On the control panel, locate the box called "COMPUTER I/O."
Click it to and click the selection marked LPT1 and then point to
"TEST" box and click. A message will appear telling you if
your choice of output ports is correct. Choose other ports as
necessary to achieve a correct setting. If you have trouble getting
your computer to recognize any printer port, try the following:
1. Connect three 2K pull-up resistors from Pins 9, 8 and 7 of U3 to
ground. This will assist "weak" printer port outputs in driving the
high capacitance load of the printer cable. If troubles persist,
try a different printer cable.

Click the "MASTER RESET" button with the mouse and
execute. This will reset the AD9850 to 0 Hz, 0 degrees 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 you 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 your 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 associ-
ated with the on-chip comparator and the onboard, 5th order,
200 ohm input/output Z, elliptic 45 MHz low-pass filter. Jumper-
ing (soldering a wire) E1 to E2, E3 to E4, and E5 to E6 connects
the onboard filter and the midpoint switching voltage to the
comparator. Users may elect to insert their own filter and com-
parator threshold voltage by removing the jumpers and inserting
a filter between J7 and J6 and then providing a threshold voltage
at E1.

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

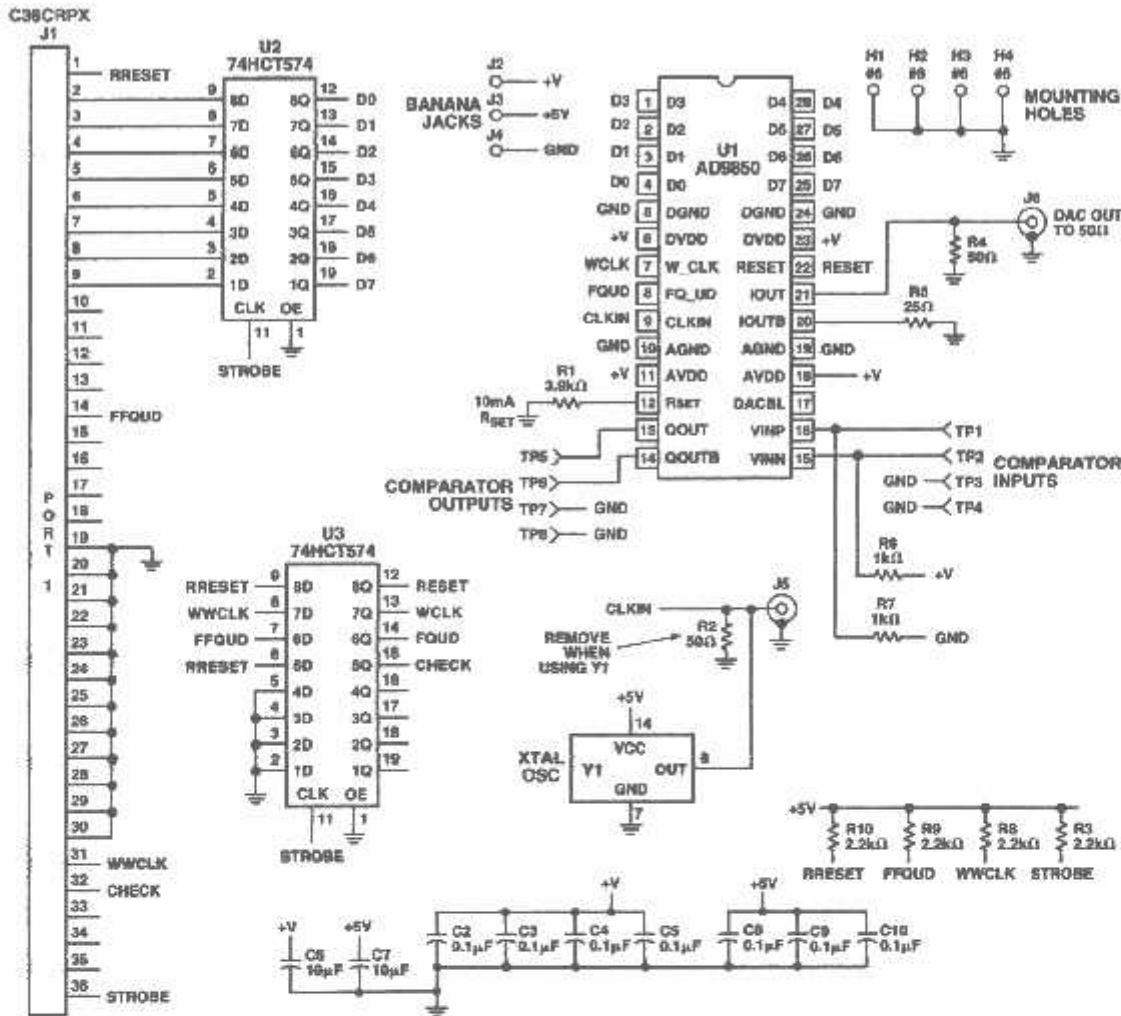


Figure 27. AD9850/FSPCB Electrical Schematic

COMPONENT LIST

Integrated Circuits

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

Capacitors

- C2-C5, C8-C10 0.1 μ F Ceramic Chip Capacitor
- C6, C7 10 μ F Tantalum Chip Capacitor

Resistors

- R1 3.9 k Ω Resistor
- R2, R4 50 Ω Resistor
- R3, R8, R9, R10 2.2 k Ω Resistor
- R5 25 Ω Resistor
- R6, R7 1 k Ω Resistor

Connectors

- J1 36-Pin D Connector
- J2, J3, J4 Banana Jack
- J5, J6 BNC Connector

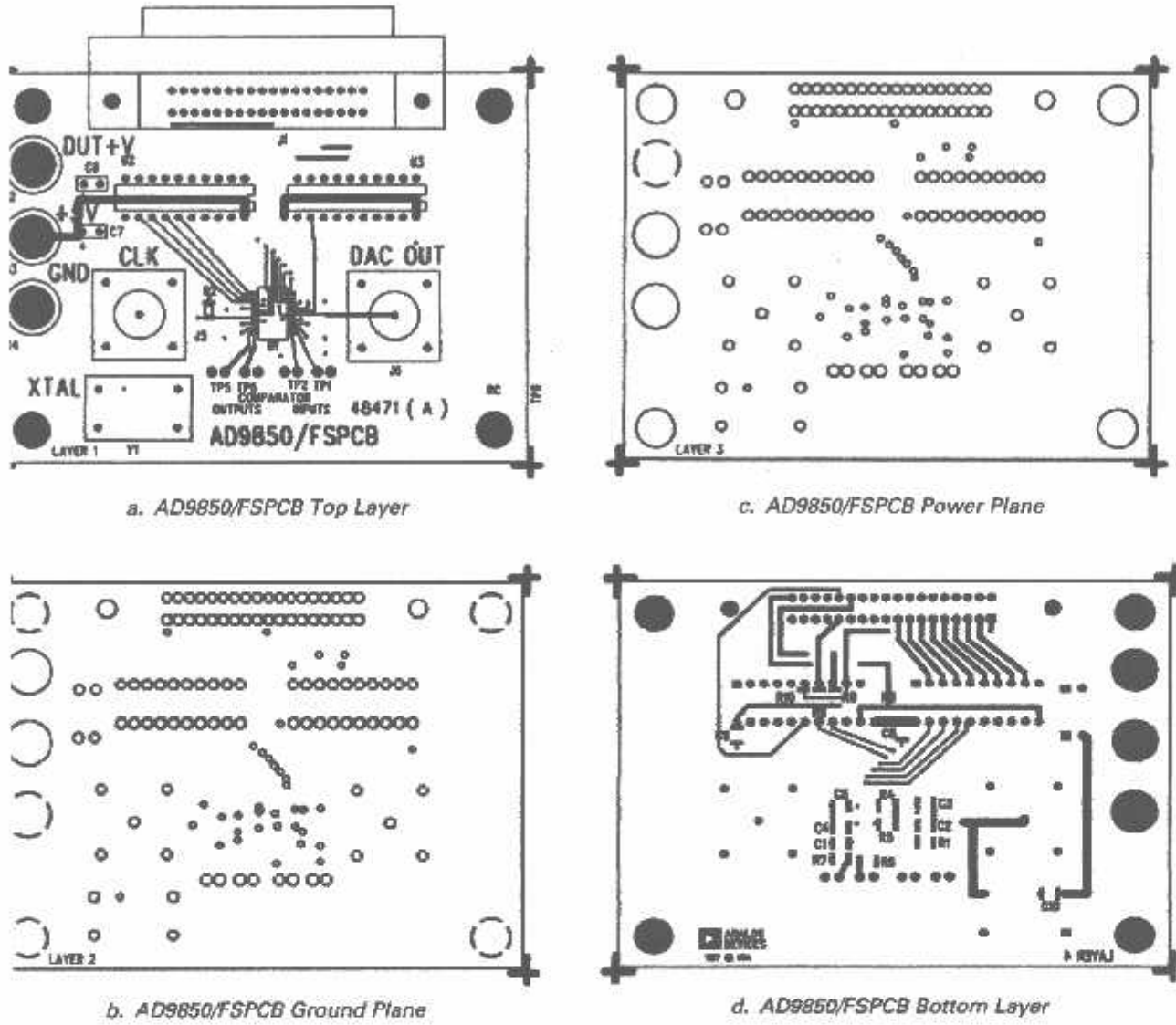


Figure 28. AD9850/FSPCB Evaluation Board Layout

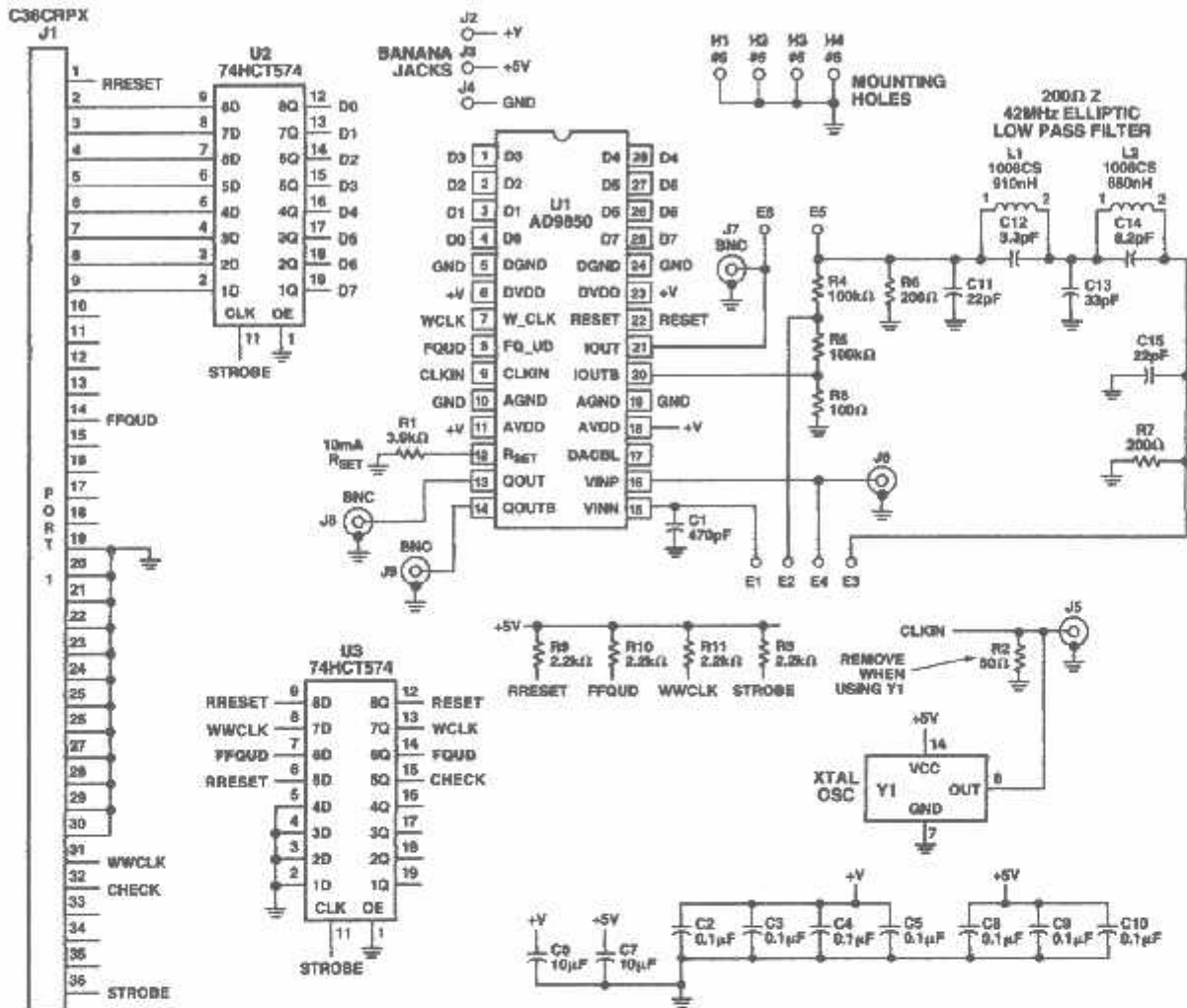


Figure 29. AD9850/CGPCB Electrical Schematic

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-C5, C8-C10 0.1 μ F Ceramic Chip Capacitor
- C6, C7 10 μ F 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

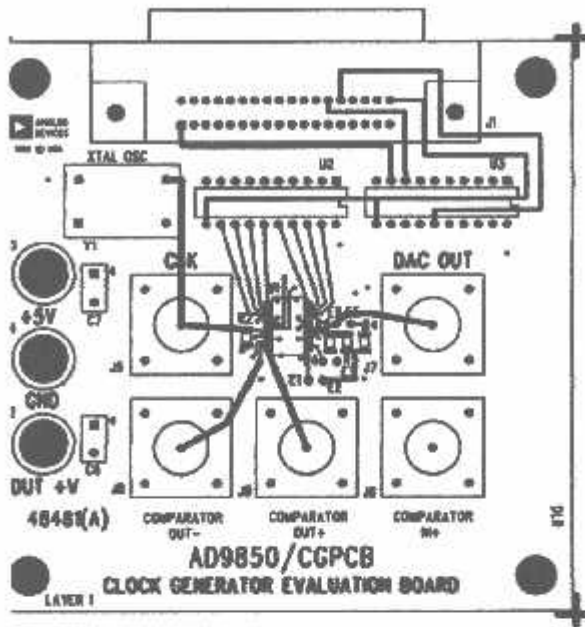
- R1 3.9 k Ω Resistor
- R2 50 Ω Resistor
- R3, R9, R10, R11 2.2 k Ω Resistor
- R4, R5 100 k Ω Resistor
- R6, R7 200 Ω Resistor
- R8 100 Ω Resistor

Connectors

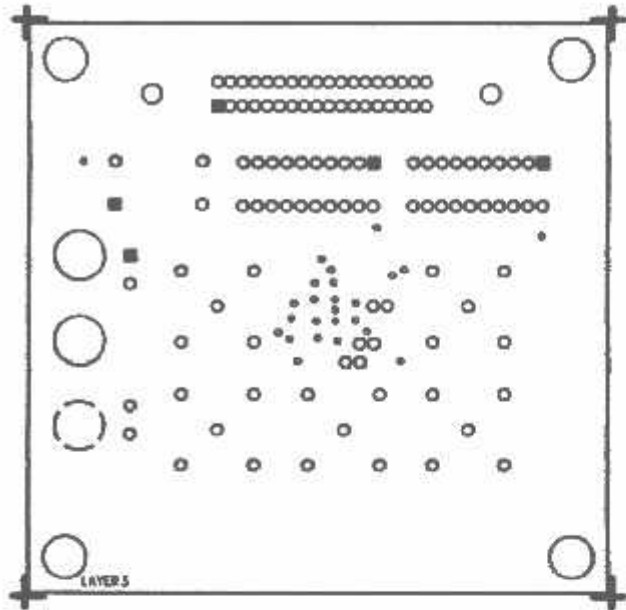
- J2, J3, J4 Banana Jack
- J5-J9 BNC Connector

Inductors

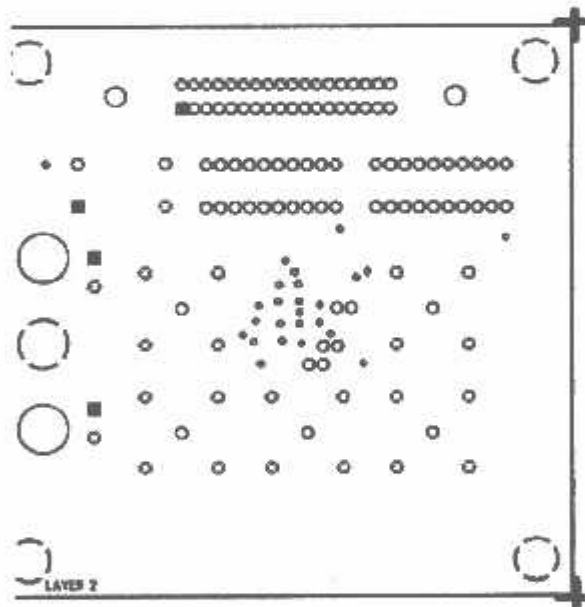
- L1 910 nH Surface Mount
- L2 680 nH Surface Mount



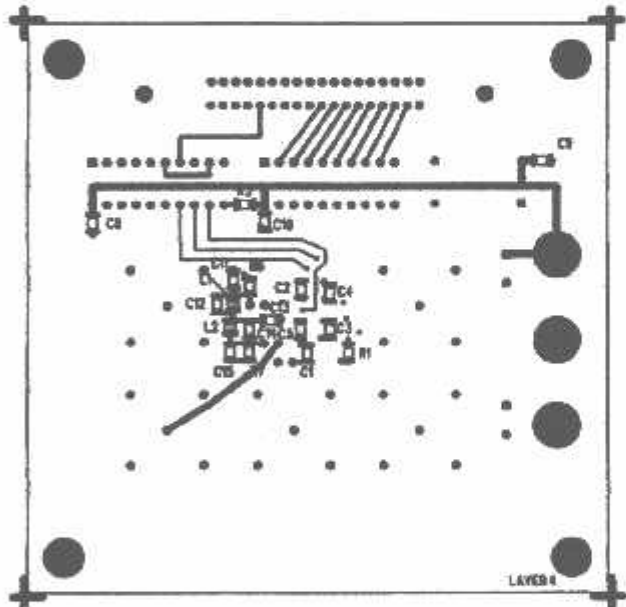
a. AD9850/CGPCB Top Layer



c. AD9850/CGPCB Power Plane



b. AD9850/CGPCB Ground Plane

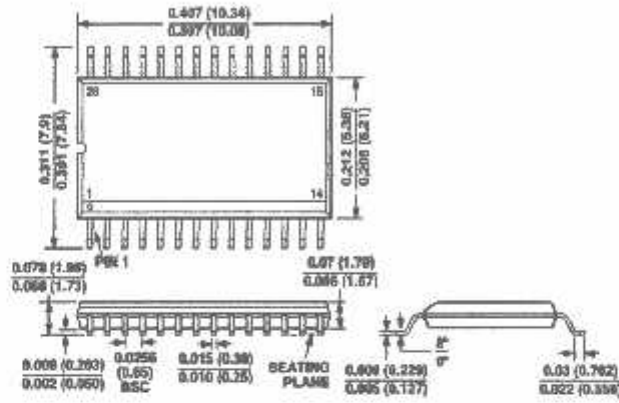


d. AD9850/CGPCB Bottom Layer

Figure 30. AD9850/CGPCB Evaluation Board Layout

OUTLINE DIMENSIONS
 Dimensions shown in inches and (mm).

**28-Lead Shrink Small Outline Package
 (RS-28)**



C2155e-0-5/89

PRINTED IN U.S.A.

LAMPIAN SKRIP PROGRAM ANDROID

```
<manifest
  <uses-permission android:name="android.permission.ANDROID_FREKUENSI"/>
  <uses-permission android:name="android.permission.BLUETOOTH"/>

  <application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme">

    <activity
      android:name=".MainActivity"
      android:configChanges="orientation|keyboardHidden"
      android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>

    <activity
      android:name=".DeviceListActivity"
      android:configChanges="orientation|keyboardHidden"
      android:label="@string/select_device"
      android:theme="@android:style/Theme.Holo.Dialog"/>

  </application>
</manifest>

<item
  android:id="@+id/secure_connect_scan"

  android:icon="@drawable/ic_action_device_access_bluetooth_searching"
  android:showAsAction="ifRoom"
  android:title="@string/secure_connect"/>

<item
  android:id="@+id/insecure_connect_scan"
  android:showAsAction="never"
  android:title="@string/insecure_connect"/>

<item
  android:id="@+id/discoverable"
  android:showAsAction="never"
  android:title="@string/discoverable"/>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  >

  <TextView
    android:id="@+id/title_paired_devices"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#666"
```

```

        android:paddingLeft="5dp"
        android:text="@string/title_paired_devices"
        android:textColor="#fff"
        android:visibility="gone"
    />

<ListView
    android:id="@+id/paired_devices"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:stackFromBottom="true"
/>

<TextView
    android:id="@+id/title_new_devices"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#666"
    android:paddingLeft="5dp"
    android:text="@string/title_other_devices"
    android:textColor="#fff"
    android:visibility="gone"
/>

<ListView
    android:id="@+id/new_devices"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:stackFromBottom="true"
/>

<Button
    android:id="@+id/button_scan"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/button_scan"
/>
</LinearLayout>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/sample_main_layout">

    <ViewAnimator
        android:id="@+id/sample_output"
        android:layout_width="match_parent"
        android:layout_height="0px"
        android:layout_weight="1">

        <ScrollView
            style="@style/Widget.SampleMessageTile"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                style="@style/Widget.SampleMessage"
                android:layout_width="match_parent"

```

```
        android:layout_height="wrap_content"
        android:paddingLeft="@dimen/horizontal_page_margin"
        android:paddingRight="@dimen/horizontal_page_margin"
        android:paddingTop="@dimen/vertical_page_margin"
        android:paddingBottom="@dimen/vertical_page_margin"
        android:text="@string/intro_message" />
</ScrollView>

<fragment
    android:name="com.example.android.common.logger.LogFragment"
    android:id="@+id/log_fragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

</ViewAnimator>

<View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:background="@android:color/darker_gray" />

<FrameLayout
    android:id="@+id/sample_content_fragment"
    android:layout_weight="2"
    android:layout_width="match_parent"
    android:layout_height="0px" />

</LinearLayout>
```