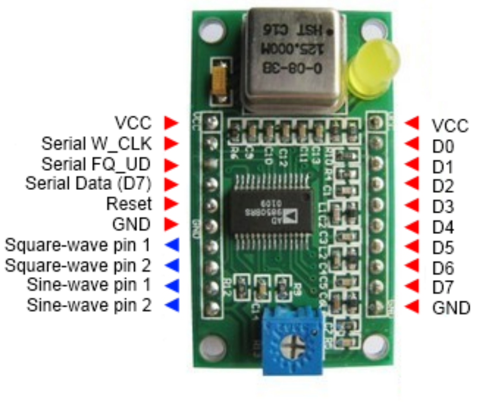
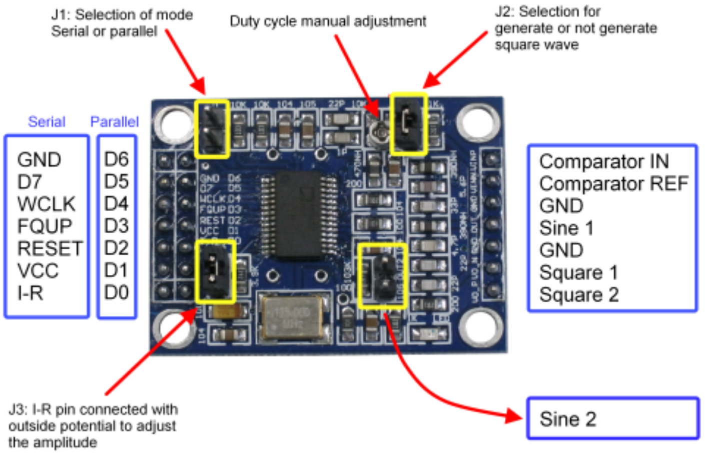
HELP ADS9850 Module & Library

ADS9850.h is a new library for the AD9850 125MHz module - output up to 30MHz
ADS9851.h is a new library for the AD9851 module with 30MHz crystals (internal 180MHz) output up to 70MHz

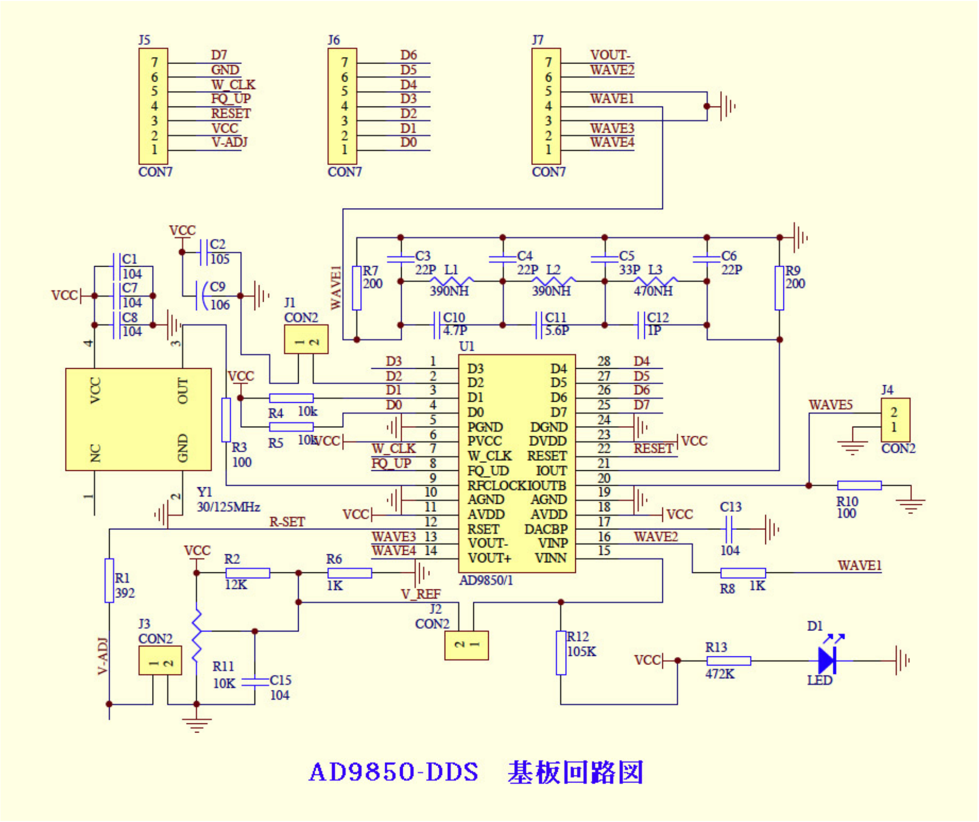
Pin out of AD9850 & 51 modules is the same.




J1: Selection of mode Serial or parallel
Duty cycle manual adjustment
J2: Selection for generate or not generate square wave
Serial
Parallel
GND
D7
WCLK
FQUP
RESET
VCC
I-R
D6
D5
D4
D3
D2
D1
D0
Comparator IN
Comparator REF
GND
Sine 1
GND
Square 1
Square 2
Sine 2
J3: I-R pin connected with outside potential to adjust the amplitude

Right hand board, D7 is DATA connection

Schematic - large board AD9851



Notes for big board

D0 & D1 to VCC via 10k, and D2 J1 = GND gives data input of xxxxx011, which is code to switch to serial mode if W_CLK then FQ_UD pulsed on start-up.

J3 shorts the V-ADJ (known as I-R on the pin out, J5 pin 1) to ground. This sets the DAC current to a value set by the onboard 3k9 resistor. If this is O/C then the output can be adjusted by an external resistor on pin I-R to GND. It is a pity that the on-board resistor is 3k9 as a lower value could be used to give higher output.

The output impedance is 200R, so I have tried a 4:1 (Z) transformer to match 50R using 4+2t/BN43-2402) and a 10k variable resistor on I-R. This gives an output of -15dB/-3dB (38-145mV) adjustable into 50R load, which is not enough. (My target is from < -5dB (0.32mW/0.125mV) to +10 (10mW/0.71V).

Schematic - small board

to be found

Notes - small board

There are 10k resistors from D0 & D1 to VCC on the board and D2 must be connected to GND, allowing switch to serial mode by pulsing W_CLK then FQ_UD,

Only ONE RF output on ZOUT, via the LPF. The other on IOUTB has no filter!

Library ADS9850.h

```
// W_CLK, FQ_UD, DATA, RESET to any pins
// void begin(int W_CLK, int FQ_UD, int DATA, int RESET); intialise pins and reset
// void setFreq(double Hz, double Chz, uint8_t p); set Hz and Chz
// void calibrate(double calHz); change xtal from standard 125MHz to new value
// void down(); power down, power up with setFreq()
// phase coding, 0-180 in 11.25deg steps 0x00, 0x01, 0x02, 0x04, 0x08, 0x10

void begin(int W_CLK, int FQ_UD, int DATA, int RESET);
void setFreq(double Hz, double Chz, uint8_t phase);
void calibrate(double calHz);
void down();
```

Operation

The module is used in the Serial mode. The example sketch is

```
// using ADS9850 library
// output a frequency

#include "ADS9850.h"

// pins
#define W_CLK 8
#define FQ_UD 9
#define DATA 10
#define RESET 11

// freqs are floating point
```

```
#define Hz 7100000.0
#define cHz 146.0
#define calFreq 124999000.0

// create ads object
ADS9850 ads;

void setup() {
  ads.begin(W_CLK, FQ_UD, DATA, RESET);

  ads.calibrate(calFreq); // calibrated XTAL freq

  ads.setFreq(Hz, cHz, 0); // on & output 7100001.46 Hz, 0 phase

  delay(30000); // 30sec

  ads.down(); // off
}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Comments

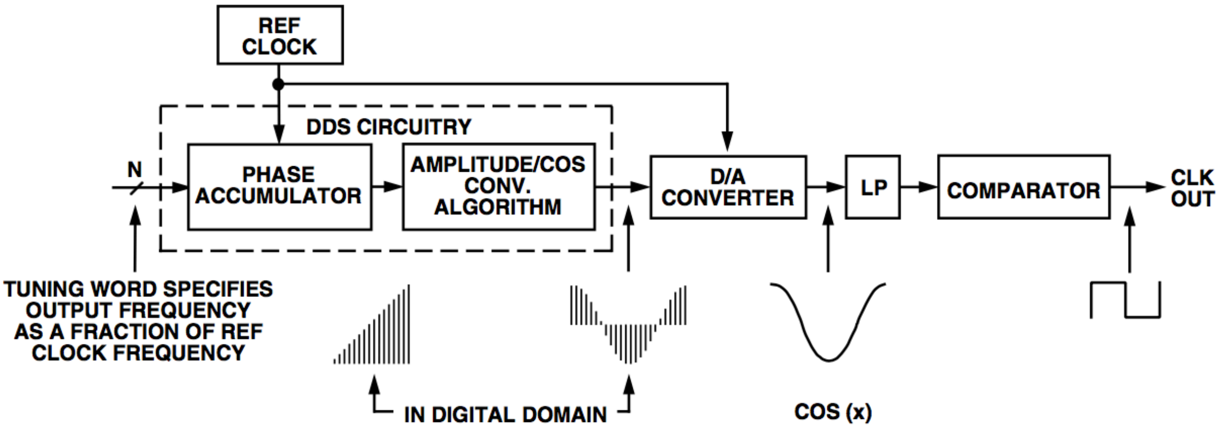
1. Include the library `ADS9850.h or AD9851.h`
2. Create an object `ads`
3. Initialise the AD9850/1 and set the pin numbers
4. Calibrate by entering the actual Xtal frequency (or x6 if AD9851 REFCLK used)
5. Output a frequency, giving two double floating point numbers, the frequency in Hz and the centi-frequency in cHz (100cHz = 1Hz)

The phase is controlled by 5 bits

```
7 6 5 4 3 2 1 0
4 3 2 1 0 P C C  C = control, P = power down, 0-4 phase bits

Load a 5bit word and <<3 times to shift up to b3-b7
0000 0000 0x00 0
0000 0001 0x01 11.5
0000 0010 0x02 22.5
0000 0100 0x04 45
0000 1000 0x08 90
0001 0000 0x10 180
```

Analog sine wave output is generated from a digital signal by a 10bit D/A converter


REF CLOCK
DDS CIRCUITRY
N
PHASE ACCUMULATOR
AMPLITUDE/COS CONV. ALGORITHM
D/A CONVERTER
LP
COMPARATOR
CLK OUT
TUNING WORD SPECIFIES OUTPUT FREQUENCY AS A FRACTION OF REF CLOCK FREQUENCY
IN DIGITAL DOMAIN
COS (x)

See ADS9850.cpp & .h for more info

AD9851

- 180Mhz xtal, or modules using 30MHz and with the internal x6 multiplier (REFCLK = 1)
- 14bit DAC for lower harmonic output

Operation

freqTune_word = freq_wanted * 2^32 / clock_freq
Resolution: freqTune_word increment of 1, gives +0.0419Hz change
down() sets freq = 0 and PD = 1, REFCLK = 1 (5th byte 0b00000101)

Tuning 4 bytes for freq control and one for chip configuration (3bits) & phase (5bits)

```
LSB                            MSB  LSB              MSB
f0-7, f8-f15, f16-f23, f24-f31, REFCLK ALWAYS0 PD p0-p4
0x00  0x00    0x00     0x00     0      0       0  00000
```

REFCLK = 1 uses x6 multiplier, for 30MHz external xtals to give internal reference of 180MHz
ALWAYS0 = 0 *must* be zero, it is a factory config mode.
PD = 1 shuts the power down.

setFreq(...) with freq and phase f(Hz), cf(cHz), p = phase (as above)

See ADS9851.cpp & .h for more info