W3PM GPS Referenced DDS VFO

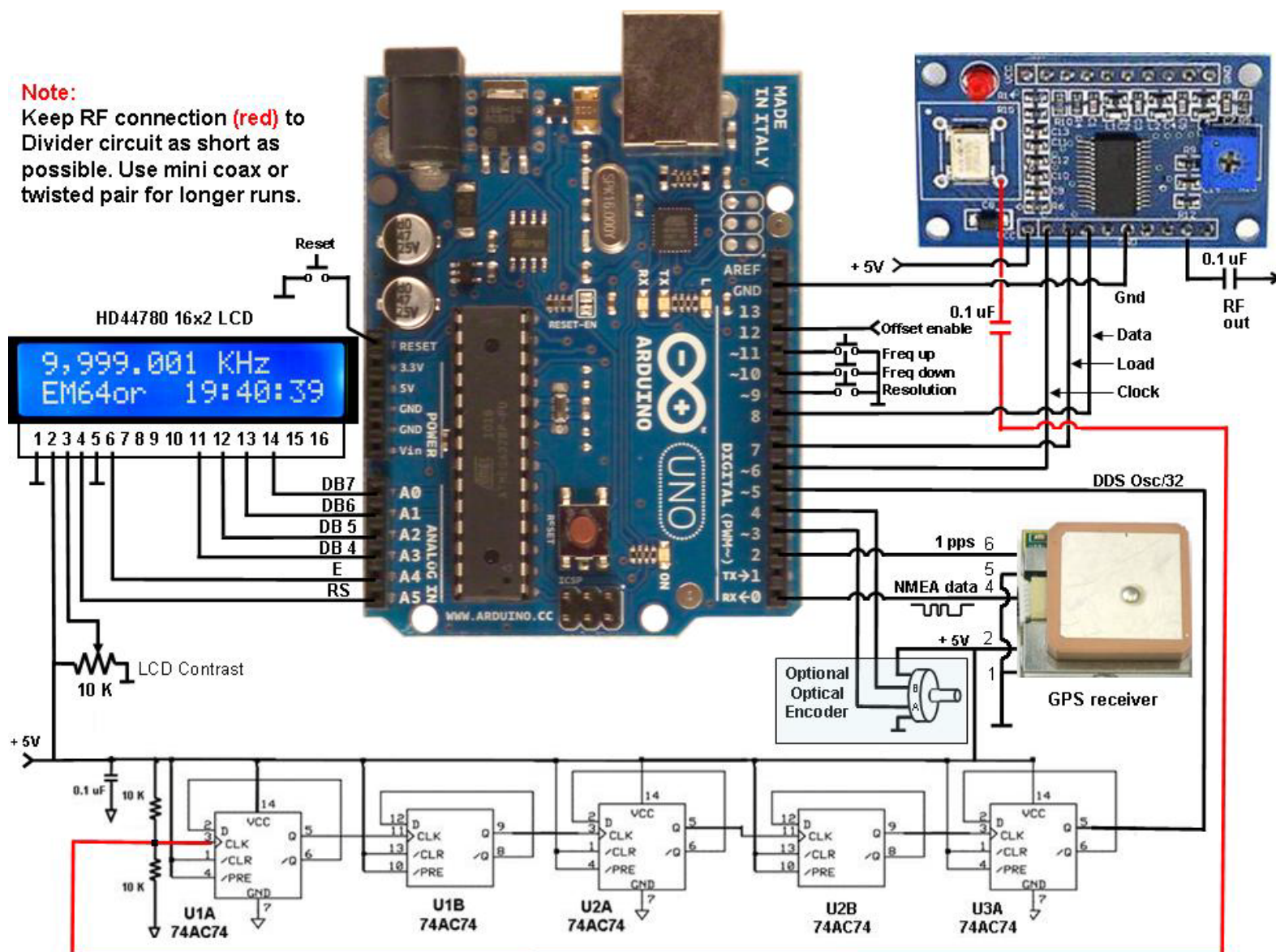**Note:**
Keep RF connection (red) to Divider circuit as short as possible. Use mini coax or twisted pair for longer runs.

Reset

HD44780 16x2 LCD

9,999.001 KHz
EM64or   19:40:39

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

DB7
DB6
DB 5
DB 4
E
RS

LCD Contrast
10 K

+ 5V

0.1 uF   10 K

10 K

U1A
74AC74

U1B
74AC74

U2A
74AC74

U2B
74AC74

U3A
74AC74

AREF
GND
13
12   Offset enable
~11   Freq up
~10   Freq down
~9   Resolution
8
7
~6
~5
4
~3
2
TX→1
RX←0

+ 5V
0.1 uF
Gnd
Data
Load
Clock

0.1 uF
RF out

DDS Osc/32

1 pps 6
5
NMEA data 4
+ 5V 2
1

Optional Optical Encoder

GPS receiver

Arduino Controlled GPS Referenced DDS VFO

Gene Marcus W3PM GM4YRE

Introduction:

Unlike the well known GPS Disciplined Oscillator using a Phased Lock Loop (PLL), this project uses an Arduino UNO (or Nano) as a frequency counter to calculate a correction factor for a Direct Digital Synthesizer (DDS) board. A GPS 1 PPS output signal is used for the frequency counter's one second gate time.

Although not as accurate as a GPS phase locked loop (PLL) frequency reference, this signal source provides a variable signal source from 100 KHz to 30 MHz with an uncertainty of less than 0.3 Hz at 10 MHz after a 30 minute warm-up. A user programmable offset function allows the unit to be used as a flexible VFO for transceiver operation.

Inexpensive AD9850 DDS boards have become readily available on the internet. When combined with GPS module and a commonly available Arduino microcontroller, the project provides a relatively inexpensive way to build a precision frequency agile signal source.

Theory of Operation:

The AD9850 module's 125 MHz oscillator is divided by 32 to obtain a 3.90626 MHz signal at pin 5 of the Uno. A counter input signal of less than 4 MHz is used due to the Uno's input frequency limitation. The Uno's internal Timer 1 is configured as a counter that is gated by the GPS 1pps signal applied to pin 2. After a count period of 32 seconds the Uno calculates the actual 125 MHz oscillator frequency. This corrected oscillator frequency is used by the microcomputer's algorithms to program the AD9850's output frequency. The AD9850 125 MHz oscillator will experience some frequency drift between the counter's 32 second update intervals. Figures 1 and 2 show the unit's accuracy from a cold start and after a 30 minute warm-up period.

Another small factor contributing to the unit's uncertainty is the AD9850's finite frequency resolution (minimum step size for a given clock frequency). The actual output frequency generated by the AD9850 is not necessarily the frequency the operator requests. This error is primarily due to the limited 40 bit tuning word input to the DDS from the controlling processor that results in an uncertainty of a few millihertz.
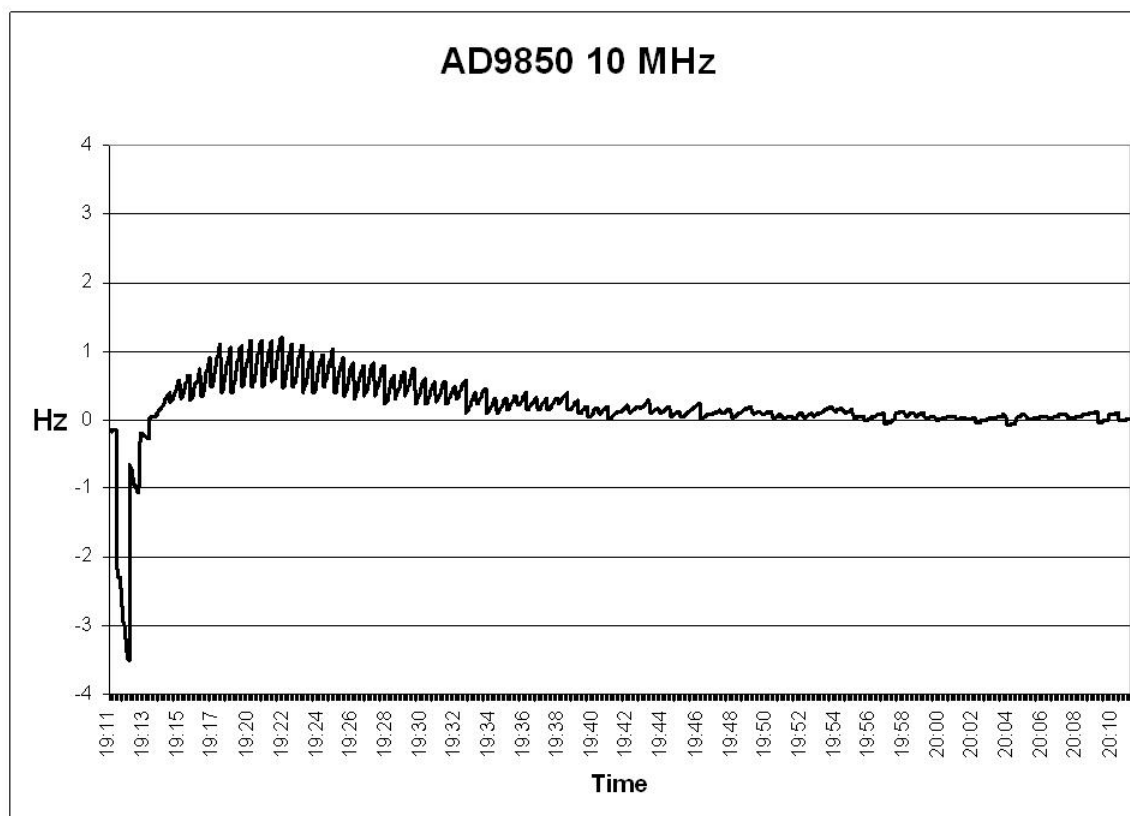

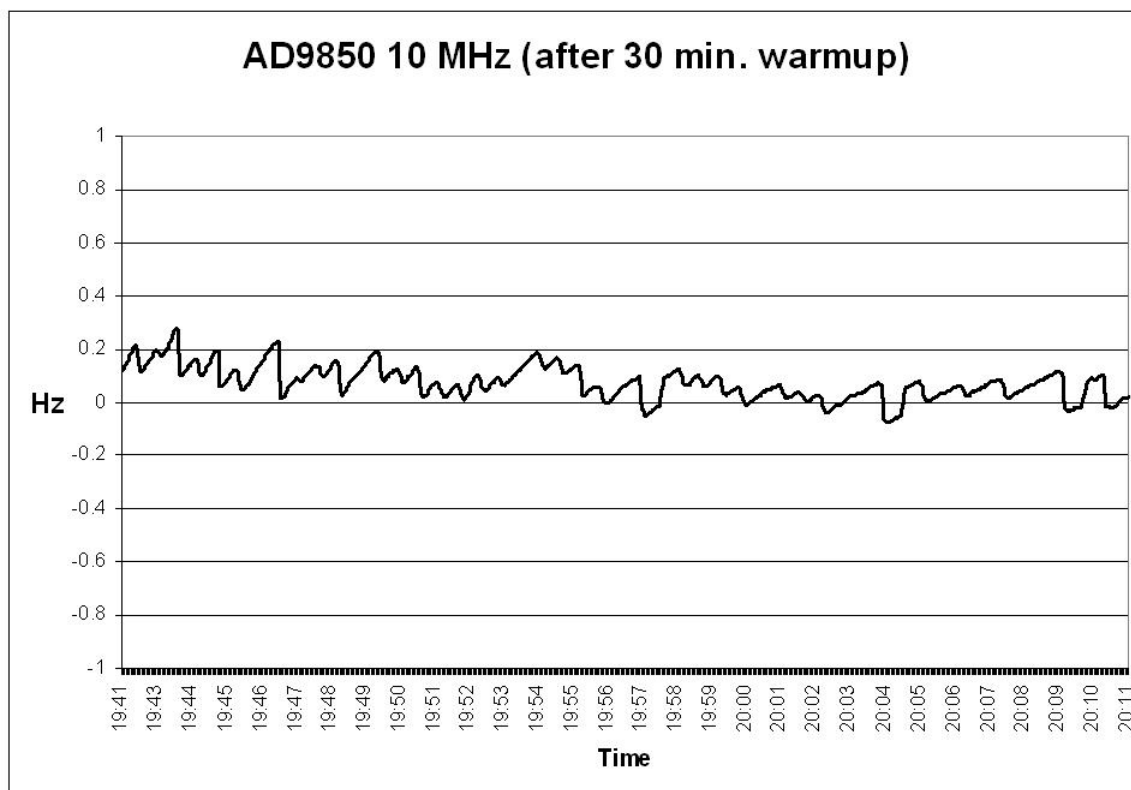Figure 1. Output accuracy from a cold start referenced to a 10 MHz GPS Disciplined Oscillator.

Figure 2. Output accuracy after a 30 minute warm-up referenced to a 10 MHz GPS Disciplined Oscillator.

Construction:

The most critical part in construction is the lead length from the 125 MHz oscillator to the input of the 74AC74 IC dividers. In the photo showing the unit's internal construction you will see a blue 0.1 uF capacitor connecting the 125 MHz oscillator chip to the 74AC74 input. Use either a twisted pair of wire or mini-coax for longer runs.

I used 74AC74 dual D-type flip-flops because I have a number of them in my junk box. Any other high speed method to divide a 125 MHz signal by 32 will work.

The optional optical encoder allows smoother tuning. I have not tried using a mechanical encoder. The use of a mechanical encoder would probably require software modification including a de-bounce routine. If an encoder is not used, the frequency can be changed using the frequency up/down pushbuttons.

Frequency stability between software calibration updates is improved by attaching a heat sink to the 125 MHz oscillator module. Almost any small heat sink will work; even a small coin will suffice. Use super glue to attach the device to the top of the oscillator module. Stability is also improved by placing the unit in an airtight enclosure to eliminate air currents around the oscillator module.

Many factors contribute to the accuracy of the GPS receiver's 1 PPS output. Some sources of error are the receiver's internal clock jitter, the number of satellites in view, multipath distortion, and receiver noise. The level of performance will vary with the GPS receiver that is used. I tried a Rockwell Jupiter and a no name Ebay purchase with good results. Some inexpensive modules such as the VK16E will probably result in inferior performance.

Mount the GPS antenna a few feet or more away from the DDS board. I found that RF noise from the DDS board causes the GPS receiver to de-sense and lose lock.

The unit may be built without the GPS receiver and used as a stand-alone VFO (see below). Operating the VFO in stand-alone mode provides the builder a means to test the unit prior to connecting the GPS related hardware.
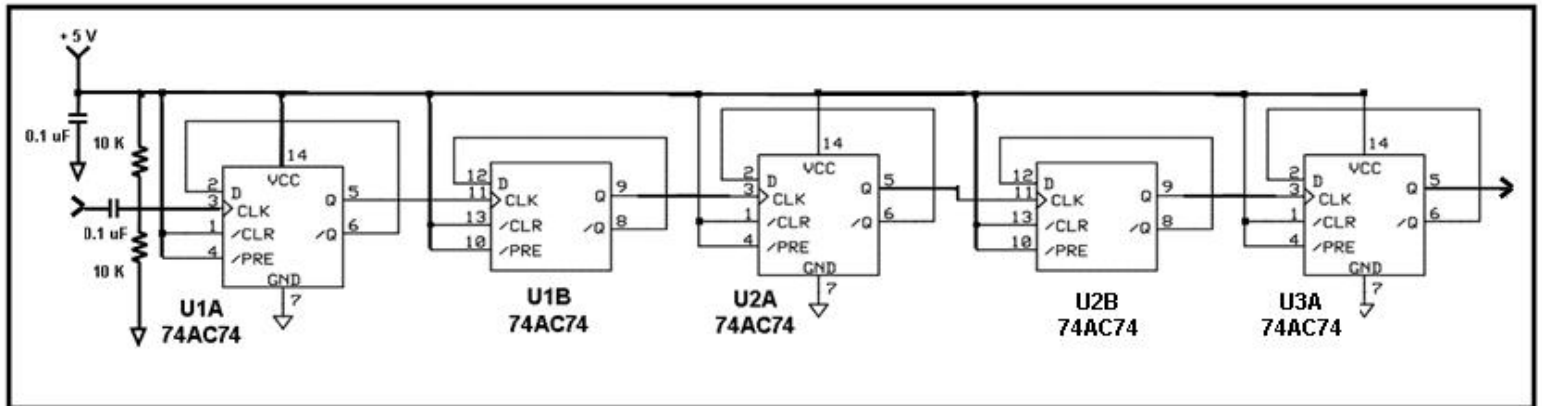

Figure 3. AD9850 125 MHz divide by 32 circuit.

Software:

Important note! The serial input port (pin 0) is used for both GPS and USB serial data. Do not attempt to upload software into the Uno while receiving GPS data. Disconnect the GPS NMEA data line before uploading software.

There are a few user defined variables near the beginning of the Arduino program. The two changes a user may want to make are the start frequency and offset frequency.

//_____Enter start frequency Hz below:_____
uint64_t frequency = 9999000;

//_____Enter offset frequency (Hz) below:_____
int fOffset = -600;

The other configurable variables found in the program were used during development to debug and experiment with other options such as the use of an AD9851. The experimenter may also want to use a four-line LCD to display other parameters such as frequency step resolution, callsign, latitude, longitude and number of satellites in view. Some of the pertinent GPS related variables can be found in the GPSprocess() subroutine.

Operation:

Operation is straightforward. After turning the unit on, allow a few minutes for the GPS receiver to obtain lock and the DDS to stabilize. A reset may be required.

Depress the resolution push button to select the frequency step. When the button is depressed the selected resolution will be displayed for a few seconds on the second line of the LCD. Refer to figure 4. The frequency may be changed either by using the frequency up/down buttons or the optional optical encoder.

The VFO offset is enabled by setting offset enable (pin 9) low.

Stand-alone operation without GPS:

The unit may be placed in stand-alone operation by changing the following user defined variables:

//_____GPS 1pps calibrate flag (0 = NO, 1 = YES)_____
int GPSflag = 0;

//_____If GPSflag = 0 Enter calibration factor below:_____
// Connect the RF output to frequency counter; adjust for 10 MHz on counter; subtract 10 MHz
// from the dial frequency and enter difference.
int fCal = -113;

Figure 4. Frequency step resolution display

Figure 5. Normal display showing frequency, Maidenhead grid locator, and UTC time.

I may be contacted at w3pm at amsat dot org for any questions or comments.