UNIQUE FEATURES OF A 40/20 METER HOMEBREW RECEIVER

LEE SNOOK W1DN

HIGHWAY 285 TECHCONNECT CLUB
TECHFEST
NOVEMBER 3, 2018
PROJECT GOALS

• Simple receiver design with readily available parts
• Adequate specifications for general use
• At home fabrication techniques
• Simple reconfiguration for the Arduino Controller and LCD Display Layout
• 50 ohm Interstage coupling for easy testing of each stage
WILL FOCUS ON THE FOLLOWING TOPICS

- History
- Block Diagram and Overview
- General Specifications
- LO Frequency Selection
- I.F. Crystal Filter Design
- Microprocessor Controlled (ATMEGA 328)
  - Arduino Based Development System
- 7 inch Color LCD Touch Display
  - 4D Systems IDE for display configuration
- At home PCB fabrication
- Single Sided PCB, Maximize Ground Plane
- Rotary Encoder Tuning Knob with selectable increments
- Variable Lowpass Audio Filter
- Two Local Oscillator DDS synthesizers
- Solid state front end switching
- Band Spectrum Display
- 10 User Memories
- Simulated Analog S Meter
- Externally programmable through a serial port
- 50 ohm Interstage coupling impedance
- Passband Offset Tuning
HISTORY

• First Receiver – Foxhole Radio
  • Razor Blade as a crystal detector
  • Of course, 110 turns on a toilet paper roll

• Second Receiver – Dual Slide Tuning
  • 5th Grade
  • 11 years old
HISTORY
MY FIRST MODULAR RECEIVER BREADBOARD. EACH MODULE IS ON ITS OWN PCB WITH 50 OHMS IN AND OUT.
HISTORY
SECOND RECEIVER USING INDIVIDUAL PLUG IN MODULES AND A LCD DISPLAY
7.15 Front End Bandpass Filter

NE602 Mixer

I.F. Filter 10 Mhz

Automatic Gain Control

I. F. Amplifier

Product Detector Mixer JMS-1

2.4 kHz Low Pass Filter

Audio Amp
HISTORY
ITERATION WITH
4D SYSTEMS
COLOR TOUCH 4
INCH DISPLAY
HISTORY

FOURTH RECEIVER (TODAY)
Internal view
GENERAL SPECIFICATIONS

• Sensitivity: -120 dBm, BW = 3200 Hz
• IMD: -42 dBm
• Third Order Intercept: \(\frac{(120-42)}{2}+42 = -3\) dBm
• SFDR: 120-42 = 92 dbm
LOCAL AND PRODUCT DETECTOR FREQUENCY SELECTION TABLE

Select a Local oscillator (LO) such that:
( use the lower cutoff edge since it has the steepest shape factor)
• As the receive frequency increase, the LO increases
• For LSB and USB the sideband falls within the crystal filter passband
40 METERS

RECEIVE 7 MHZ
1st LO = 16.9966 Mhz
I.F. = 16.9966 – 7 = 9.996598 MHZ
NOTE; THE LOWER EDGE OF THE I.F. CRYSTAL FILTER

• **Case 1**
  • As the receive frequency increases
  • the LO frequency increases
  • LO = IF + RF

• **Case 2 (For LSB)**
  • As the Receive Lower Sideband Envelope decreases
  • the IF increases
  • IF = LO - RF
20 METERS

RECEIVE 14 MHZ

1\textsuperscript{st} LO = 4.003402 MHz

IF = RF - LO

\begin{itemize}
  \item **Case 1**
    \begin{itemize}
      \item As the receive frequency increases
      \item the LO frequency increases
      \item LO = IF + RF
    \end{itemize}
  \item **Case 2 (For LSB)**
    \begin{itemize}
      \item the Receive Lower Sideband Envelope decreases
      \item the IF increases
      \item IF = LO - RF
    \end{itemize}
\end{itemize}
EXAMPLES (CONT.)

RECEIVE 14 MHZ
1\textsuperscript{st} LO=4.003402 MHz
I.F. = 14-4.003402= 9.996598 MHz
NOTE; THE LOWER EDGE OF THE I.F. CRYSTAL FILTER

- **Case 1;** As the receive frequency increase, the LO frequency increases
- Receive 14.1 Mhz
- 1\textsuperscript{st} LO=4.003402 Mhz
- I.F. = 14.1 – 4.003402 = 10.096598 Mhz
  Note; The Lower Edge of the I.F. Crystal Filter

- **Case 2;** For LSB, the Receive Lower Sideband Envelope decreases, the IF increases
- Receive 6.999 Mhz, 1 khz audio tone
- 1\textsuperscript{st} LO=16.9966 Mhz
- I.F. = 16.9966 – 6.999 = 9.9976 Mhz
  Note; The LSB envelope is within the passband
CRYSTAL FILTER DESIGN
MEASURE A GROUP OF 10 MHZ CRYSTALS AND SELECT 6 THAT HAVE THE CLOSEST PARAMETERS

<table>
<thead>
<tr>
<th>Xtal</th>
<th>RL</th>
<th>Loss in dB</th>
<th>Rm</th>
<th>Reff</th>
<th>fs</th>
<th>fl</th>
<th>fu</th>
<th>deltal f</th>
<th>Cm</th>
<th>Lm</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>12.5</td>
<td>2.62</td>
<td>8.80181</td>
<td>33.80181</td>
<td>9995949</td>
<td>9996160</td>
<td>9996332</td>
<td>172</td>
<td>8.10925E-15</td>
<td>0.03129334</td>
</tr>
<tr>
<td>12</td>
<td>12.5</td>
<td>2.37</td>
<td>7.842787</td>
<td>32.84279</td>
<td>9995918</td>
<td>9996130</td>
<td>9996297</td>
<td>167</td>
<td>8.10348E-15</td>
<td>0.031315828</td>
</tr>
<tr>
<td>18</td>
<td>12.5</td>
<td>2.49</td>
<td>8.299676</td>
<td>33.29968</td>
<td>9995912</td>
<td>9996126</td>
<td>9996289</td>
<td>163</td>
<td>7.80087E-15</td>
<td>0.032530651</td>
</tr>
<tr>
<td>22</td>
<td>12.5</td>
<td>2.97</td>
<td>10.19168</td>
<td>35.19168</td>
<td>9995893</td>
<td>9996113</td>
<td>9996292</td>
<td>179</td>
<td>8.10607E-15</td>
<td>0.031305982</td>
</tr>
</tbody>
</table>

Average: fl = 9996215, fc = 9997815, fh = 9999415

<table>
<thead>
<tr>
<th>Minus 45</th>
<th>45</th>
<th>Plus 45</th>
</tr>
</thead>
<tbody>
<tr>
<td>9995949</td>
<td>9996160</td>
<td>9996332</td>
</tr>
<tr>
<td>9995918</td>
<td>9996130</td>
<td>9996297</td>
</tr>
<tr>
<td>9995912</td>
<td>9996126</td>
<td>9996289</td>
</tr>
<tr>
<td>9995893</td>
<td>9996113</td>
<td>9996292</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>deltal f</th>
<th>Cm</th>
<th>Lm</th>
</tr>
</thead>
<tbody>
<tr>
<td>172</td>
<td>8.10925E-15</td>
<td>0.03129334</td>
</tr>
<tr>
<td>167</td>
<td>8.10348E-15</td>
<td>0.031315828</td>
</tr>
<tr>
<td>163</td>
<td>7.80087E-15</td>
<td>0.032530651</td>
</tr>
<tr>
<td>179</td>
<td>8.10607E-15</td>
<td>0.031305982</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Average:</th>
<th>fl</th>
<th>fc</th>
<th>fh</th>
</tr>
</thead>
<tbody>
<tr>
<td>9996215</td>
<td>9997815</td>
<td>9999415</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fl</th>
<th>fc</th>
<th>fh</th>
</tr>
</thead>
<tbody>
<tr>
<td>9996215</td>
<td>9997815</td>
<td>9999415</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fl</th>
<th>fc</th>
<th>fh</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.02992E-15</td>
<td>0.03161145</td>
<td></td>
</tr>
</tbody>
</table>
CRYSTAL FILTER DESIGN (CONT.)

• For a review of Crystal Filter Design use the following reference
  *Crystal Motional Parameters, A Comparison of Measurement Approaches*
  Jack R. Smith K8ZOA, 11 June 2006

• Using the Freeware Program; DISHAL vers 2.0.5.2;
  Enter Lm, Cm, Cp, BW, # of Xtals, Ripple
Crystal Ladder Filter Calculator "DISHAL" Vers. 2.0.5.2
HF Tools by DJ6EV

Select either Lm or Cm of xtal:
- Lm = 8.03 nF
- Cm = 8.03 nF

Xtal Parameters
- Lm = 31.56kHz
- fs = 9996.000 kHz
- Cm = 8.03 nF
- fp = 10036.054 kHz

Filter Parameters
- Type: Chebychev
- PB-Ripple: 1db
- Impedance [Ohm]: 191.1
- # of Xtras: 6
- Center Frequency [kHz]: 9997.174
- BW (6db): 2.05 kHz
- BW (60db): 4.14 kHz
- BW (20db): 2.33 kHz
- BW (80db): 5.89 kHz
- BW (40db): 3.01 kHz
- BW (100db): 8.57 kHz

Coupling (Shunt) Capacitances [pF]
- Ck12 = 59.7
- Ck23 = 71.1
- Ck34 = 72.6
- Ck45 = ----

Tuning (Series) Capacitances [pF]
- Cs1 = 71.1
- Cs3 = 333.5
- Cs4 = ----
- Cs5 = ----
- Cs6 = ----
- Cs7 = ----

Calculate
LOG→Lin

[Graph showing frequency response with markers for attenuation and center frequency]
MEASURED RESULT

BANDWIDTH FOR THIS FILTER = 3 kHz
MICROPROCESSOR (ATMEGA 328)

• The Arduino Uno prototyping board

• **Arduino Uno** is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

• Off-Boarding the ATMEG328 requires changing the pin assignments

• For instance; Programming a command to Pin 13 on the Uno Prototype board must now be changed to Pin 19 for the stand alone ATMEGA328. See the diagame below.
MICROPROCESSOR PROGRAMMING

• Programming either the Uno board or the stand alone processor requires connecting a serial USB cable to the Uno USB port or, for the stand alone case, using a MAX232 chip to convert the RS232 voltages to 5 volts.

• I use an USB to RS232 converter cable which can be purchased online at;

MICROPROCESSOR PROGRAMMING (CONT.)

• A simple programming example:

• Make an LED blink. Connect an LED to Pin 13 of the Arduino Uno Proto board with the appropriate series dropping resistor of 330 ohms.

• Note: The maximum current sourcing pin current is = 40 ma.

• An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.

• Download the Arduino IDE at https://www.arduino.cc/en/Main/Software
Code;
// The setup function runs when you press reset or power the board
// Code in the setup(), Intitalizes Pins as Input or Output.
// Setup only runs Once on startup
// The void keyword is used only in function declarations. It indicates that the function is expected to return
// no information
// to the function from which it was called.
// The following code is for the UNO proto board, Pin 13 would be changed to pin 19 for the off boarded
// processor.

```cpp
void setup()
{
  pinMode(Pin 13, OUTPUT); // initialize digital pin 13 as an output.
}

void loop() // The loop function runs over and over again forever
{
  digitalWrite(Pin 13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(500); // wait for a second
  digitalWrite(Pin 13, LOW); // turn the LED off by making the voltage LOW
  delay(500); // wait for a second
}```
PROGRAMMING THE 4D SYSTEMS LCD DISPLAY

Backside view
• The microUSB Programming Adaptor (uUSB-PA5 and uUSB-PA5-II) is a USB to Serial-TTL UART bridge converter

• Insert the micro SD card into the PC USB port (use micro to mini USB adapter card if necessary)

• Connect uUSB-PA5 from +5V, TX, RX, GND, Reset connector to PC USB port.
BOOT up the IDE program
Select the ViSi Genie Display Type and Orientation
Select Programming Environment; By simply laying the display out with the wanted objects and setting the events to drive them, the code is written automatically. ViSi-Genie provides the latest rapid development experience from 4D Labs.
Control Objects that can be used
Start the LCD layout by selecting and placing the “Objects” onto the workspace.
Example: In this example we will place a Button on the screen and note the Object Inspector.
Remember to Select the Comms Tab and Comm Port, it will turn Blue when connected.
Name and Index number.
Under Object Inspector on the right side of the screen, select the Events Tab and select “Report Message”.

![Object Inspector screenshot](image)
EVENT HANDLER EXAMPLE

The Arduino Code (in the Event Handler) will look for a Winbutton Event with an Index of 0 and execute the user code when this button is pressed.
The programming language used for Arduino is based on C language. The syntax is almost identical to that of C or C++.

Void Setup()
{
    Serial.begin(115200);
genie.Begin(Serial);
genie.AttachEventHandler(myGenieEventHandler);
delay (2000); //let the display start up
genie.WriteContrast(brightness);}
EVENT HANDLER EXAMPLE (CONT.)

Void Loop()
{
    genie.DoEvents();
}

// Event Handler-------------------------------------------------------------
void myGenieEventHandler(void)
{
    genieFrame Event;
    genie.DequeueEvent(&Event);

    if(Event.reportObject.cmd == GENIE_REPORT_EVENT)
    {
        if (Event.reportObject.object == GENIE_OBJ_WINBUTTON)
        {
            // If Winbutton Pressed, Do Something-----------------------------------
            if (Event.reportObject.index == 0)
            {// If Button(0)
                //Do Something
            }
        }
    }
}
I use PCB Artist for Schematic Capture and Board Layout

The Receiver Layout in PCB Artist:
PCB FABRICATION PROCESS (CONT.)

- Using an Ink Jet printer,
  - print the Top Copper side of the Layout
  - High Contrast Black
  - onto rough side of transparent film
- Remember, under printer settings, Flip the print under the printer setup so that the rough side is facing the photo sensitive side of the PCD board.
PCB FABRICATION PROCESS (CONT.)

- Remove the light protective film from the photo sensitive PCB board
- Insert the photo sensitive board under the transparent film and expose in a lightbox
PCB FABRICATION PROCESS (CONT.)

• Heat Developer and Ferric Chloride to ~90 degrees F.

• Exposed PCB into glass tray with PCB Developer
  • DATAK Cat. No 12-404 Positive Type PCB Developer
  • Developer concentrate diluted 1:10 with water.

• Agitate until the Light Sensitive Film (Green color) is dissolved between traces.

• Wash the board
PCB FABRICATION PROCESS (CONT.)

- Board into Ferric Chloride Etchant Solution bath
  - MG Chemicals #415-1L, 945 ml bottles Ferric Chloride
  - Typically 15 minutes

- Periodically agitate

- Inspect and remove when the copper between traces has been dissolved

- Wash off Ferric Chloride with fresh water
PCB FABRICATION PROCESS (CONT.)
QUADRATURE ROTARY ENCODER
SPARKFUN ELECTRONICS; PN COM-11102

Rotation (Clockwise)
The routine samples Channel A every loop cycle every 15 ms.
The software looks for a negative slope on channel A.
(Compare A(sample 1) to A(sample 2)..................)
If A(1) is greater than A(2), then we have a negative slope.

If a negative slope is found on A, then look at the value of B
If B is 0, then the wheel is turning Clockwise and the increment is positive.
An RC time network was added to Ports A and B for Debounce. R=10k and C=.01\mu f.
In the following code:
A1=Sample 1 of Port A
A2=Sample 2 of Port A
B=Sample Port B

//-----------------------------Rotarty Encoder-----------------------------

A1=digitalRead(Encoder_Pin1);
delay(15); //Delay next sample by 15 ms
A2=digitalRead(Encoder_Pin1);
B=digitalRead(Encoder_Pin2);
if (A1>A2) //A1(sample 1) is greater than A2(sample 2)
{
  if (B==HIGH)
  {
    f_rx=f_rx+increment[inc_pointer]; //Increase frequency by increment
    freq_low=(f_rx%10000);
    freq_high=(f_rx-freq_low)/10000;
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 2, freq_low);
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 3, freq_high);
    load1();
  }
}

else //A(sample 1) is less than A(sample 2)
{
  f_rx=f_rx-increment[inc_pointer]; //Decrease frequency by increment
  freq_low=(f_rx%10000);
  freq_high=(f_rx-freq_low)/10000;
  genie.WriteObject(GENIE_OBJ_LED_DIGITS, 2, freq_low);
  genie.WriteObject(GENIE_OBJ_LED_DIGITS, 3, freq_high);
  load1();
}

More work is required, sometimes the encoder will skip a count when turning the wheel too fast.
VARIABLE BANDPASS AUDIO FILTER

• MAX7403 is a 8th-order, lowpass, elliptic, switched-capacitor filters

• These devices draw 2mA of supply current and allow corner frequencies from 1Hz to 10kHz

• Two clocking options are available
  • self-clocking through the use of an external capacitor
  • external clocking for tighter cutoff-frequency control

• Cutoff Frequency = Clock/100. Fosc(kHz) = (38*10000)/Cosc; Cosc is in pf.

• For my application, I used external capacitors (self clocking) switched with a solid state single pole three throw switch (TS5A3357)
VARIABLE BANDPASS AUDIO FILTER (CONT.)
VARIABLE BANDPASS AUDIO FILTER (CONT.)

• Audio Lowpass Filter Frequency Response
• Traditional designs of high bandwidth frequency synthesizers employ the use of a phase-locked-loop (PLL).

• A direct digital synthesizer (DDS) provides many significant advantages over the PLL approaches
  • Fast settling time
  • Sub-Hertz frequency resolution
  • Continuous-phase switching response
  • Low phase noise
DIRECT DIGITAL SYNTHESIZERS (CONT.)

• A DDS produces a sine wave at a given frequency

• The frequency depends on two variables
  • Reference-clock frequency
  • Binary number programmed into the frequency register (tuning word)
DIRECT DIGITAL SYNTHESIZERS (CONT.)

• Binary number in frequency register provides main input to phase accumulator

• Sine look-up table used

• Phase accumulator computes phase (angle) address for look-up table
  • Outputs the digital value of amplitude
  • Corresponding to the sine of that phase angle—to DAC

• DAC converts number to a corresponding value of analog voltage or current
DIRECT DIGITAL SYNTHESIZERS (CONT.)

- To generate a fixed-frequency sine wave, constant value (the phase increment— from binary number added to the phase accumulator with each clock cycle.
  - If the phase increment is large, the phase accumulator will step quickly through the sine look-up table and thus generate a high frequency sine wave.
  - If the phase increment is small, the phase accumulator will take many more steps, accordingly generating a slower waveform

![Digital Phase Wheel](image)

\[ f_o = \frac{M \cdot f_c}{2^n} \]

<table>
<thead>
<tr>
<th>n</th>
<th>Number of Points = (2^n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>256</td>
</tr>
<tr>
<td>12</td>
<td>4,096</td>
</tr>
<tr>
<td>16</td>
<td>65,536</td>
</tr>
<tr>
<td>20</td>
<td>1,048,576</td>
</tr>
<tr>
<td>24</td>
<td>16,777,216</td>
</tr>
<tr>
<td>28</td>
<td>268,435,456</td>
</tr>
<tr>
<td>32</td>
<td>4,294,967,296</td>
</tr>
<tr>
<td>48</td>
<td>281,474,976,710,656</td>
</tr>
</tbody>
</table>

*Figure 3: Digital Phase Wheel*
Consider the case for \( n = 32 \), and \( M = 1 \). The phase accumulator steps through each of 232 possible outputs before it overflows and restarts. The corresponding output sinewave frequency is equal to the input clock frequency divided by \( 2^{32} \). If \( M=2 \), then the phase accumulator register "rolls over" twice as fast, and the output frequency is doubled. This can be generalized as follows.

For an \( n \)-bit phase accumulator (\( n \) generally ranges from 24 to 32 in most DDS systems), there are \( 2^n \) possible phase points. The digital word in the delta phase register, \( M \), represents the amount the phase accumulator is incremented each clock cycle. If \( f_c \) is the clock frequency, then the frequency of the output sinewave is equal to:

\[
 f_{\text{OUT}} = \frac{(\Delta \text{Phase} \times \text{CLKIN})}{2^{32}} \\
\Delta \text{Phase} = \frac{f_{\text{OUT}} \times 2^{32}}{\text{CLKIN}}
\]
DIRECT DIGITAL SYNTHESIZERS

- 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.
- Premounted on a small PCB online from either Amazon or Digikey.
DIRECT DIGITAL SYNTHESIZER (AD9850)

- The frequency tuning, control, and phase modulation words are loaded into the AD9850 via
  - Parallel byte or
  - Serial loading format (I used the serial loading format)

- In serial load mode, subsequent rising edges of W_CLK shift the 1-bit data on Pin 25 (D7) through the 40 bits of programming information.

- After 40 bits are shifted through, an FQ_UD pulse is required to update the output frequency (or phase).
The Arduino Code for loading the DDS synthesizer:

```c
long function_delta_phase1() {
    freq_dds1 = f_rx + if_center - offset;

    // From the equation above \( \Delta \text{Phase} = (fOUT \times 2^{32}) / CLKN \)
    long delta_phase1 = (((freq_dds1) * pow(2, 32)) / clock);

    return (delta_phase1);
}

void load1() {
    long delta_phase1 = function_delta_phase1();
    digitalWrite (LOAD1_Pin, LOW);
    shiftOut(DATA_Pin, CLOCK_Pin, LSBFIRST, delta_phase1);
    shiftOut(DATA_Pin, CLOCK_Pin, LSBFIRST, delta_phase1 >> 8);
    shiftOut(DATA_Pin, CLOCK_Pin, LSBFIRST, delta_phase1 >> 16);
    shiftOut(DATA_Pin, CLOCK_Pin, LSBFIRST, delta_phase1 >> 24);
    shiftOut(DATA_Pin, CLOCK_Pin, LSBFIRST, 0x0);
    digitalWrite (LOAD1_Pin, HIGH);
}
```