Atlas 210X Digital VFO Modification

Version 2.07
July 22, 2022

Written By:
John Satterfield – KI5IDZ

The following is a link for the files: **https://github.com/JohnWSatterfield/Atlas_210X_Digital_VFO**

Table of Contents

I. OVERVIEW

A. Atlas Radio

The Atlas Radio was developed in the early 70's by Herb Johnson W6QKI who had previously founded Swan Electronics. Perhaps bored, perhaps seeking new challenges in life, Herb Johnson started the Atlas Radio in 1974. His goal was to make an all new solid state design which he called the model 180. In 1975 he introduced the Atlas 210 which remained in production until 1979. He went on to produce over 19,000 of these radios, many of which, are still in operation today. The radio was designed on the KISS principle of keeping everything as simple as possible which makes it an ideal candidate for modification. The various sections of the radio: Audio, 1st IF, 2nd IF, band pass filters, PA, carrier oscillator, VFO were all on separate circuit boards and all solid state. The result was a unique, practical and easy to work on transceiver.

The major issues with the radio are the simplistic string tuning that is always off frequency and the drift of the VFO which was 100 to 300 Hz when new and more today with the 40+ year old radios. The installation of a digital VFO solves both of these problems and gives the operator an easy to read digital interface that drifts less than +/- 5Hz.

The carrier oscillator of this radio used crystals for USB and LSB which are difficult if not impossible to obtain today and suffer from additional issues of drifting. It was for this, the cost and the ease of programming the Si5351 that this modern and easy to obtain signal generator was used for both the digital VFO and the carrier oscillator.

The Si5351 signal generator has 3 different output channels of which CLK0 and CLK1 are used in this project. The original low pass filtering of the Atlas radio was used to filter most of the generated harmonics.

The Interface is a 1.77 inch TFT Module that uses an SPI interface to a 128x160 color display that uses the ST7735 library of commands. The Micro controller selected is an ESP-32 mini "FireBeatle" These items are stacked with a custom board that houses the 3.3VDC power supply, DS3231 clock module and a PFC8574 I2C remote I/O expander module. An Oak-Grigsby 128 bit rotary encoder was used for dialing the VFO and a single SPDT mini momentary toggle switch for operation.

A blank portion of the existing band switch inside the existing VFO was used to signal the Micro controller board which band the radio was in and the existing low pass and band pass filters were used to limit the generated harmonics.

Finally a Drake TR7 2300 Hz crystal filter was used to replace the existing crystal filter that was originally installed in the radio. I paid $23 on Ebay for two new crystal filters. The drop-off of the original filter was so great that even with fine tuning the output of the radio was maxed out at 60 watts on 20m. The original crystal filter is no longer available and is the cause of many of these radios to be retired from service. The radio originally had an output of 90-100 watts. This output was restored with the use of the new crystal filter.

B. Objectives

This document is written to describe the various parts of the new VFO and carrier oscillator, and the software utilized to control these modifications. The challenge of such a project was significant for a 70+ retired engineer but, It kept me busy and occupied over the lock downs that occurred with the Covid-19 epidemic. Some dis-assembly of the radio is needed to accommodate the new components but were done with minimal impact so as to permit restoring the radio back to its original circuitry if things did not turnout as originally planned. I wanted to be able to go back should it not work out. As it turns out, it was not necessary, but I still have the option to restore the radio to original without a lot of work.

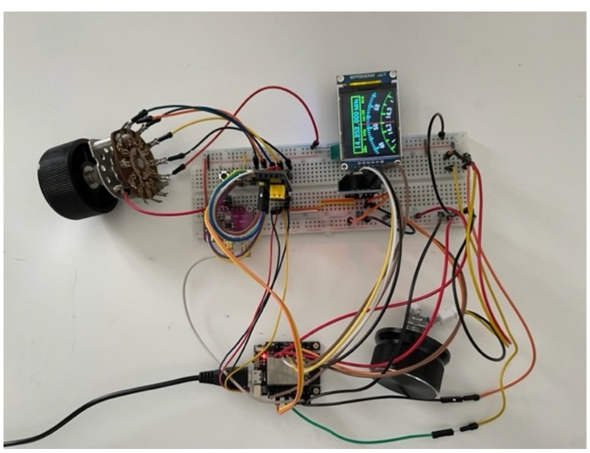
I also wanted to provide documentation for maintenance of the radio in the future by me or anyone that might own it in the future. It seems in all probability the radio will outlive me.

C. Additional Documents

I have included in the appendix additional documents such as the Arduino Sketch (circuit diagram) used to build the radio and links to the software used. Everything is programmed using C and C++ on the Arduino Development Environment. The ESP-32 micro controller is a must if you want to use the analog dial (developed by T. Uebo – JF3HZB). I used all open sourced software and libraries for controlling the various components and I give all the sources and tribute to the authors in the appendix. The main program and libraries used are all made available for anyone.

D Options

I strongly recommend that anyone attempting this project use a solder-less breadboard to build and program their VFO. I bought the breadboard I used on Amazon for more or less $10: https://www.amazon.com/DGZZI-Electronics-Solderless-tie-Points-Breadboard/dp/B07QKJXZNV/ref=dp_prsubs_3?pd_rd_i=B07QKJXZNV&psc=1



I originally built an external VFO that can easily plug into the back of the Atlas 210X using the 9-pin tube socket provided by the manufacturer of the radio. It allowed me to experiment and develop the operational software that eventually made it into my internal VFO project.

You can easily remove the internal VFO from the radio if you want to recover the space in the radio and house your new VFO there. This is a bit more invasive but it does simplify certain aspects of the project such as tapping into the original band selector spare switches which is a bit of a challenge to get a soldering pencil into the radio if left intact. I did have to remove a couple of components and temporarily move them out of the way to accomplish the tie into this band switch.

Removing the carrier oscillator PC600 and replacing it with CLK1 of the Si5351 generator board is also an option if your current Carrier Oscillator (C.O.) is operational. You will need a C.O. if you build an external VFO as the 9 pin plug only makes an allowance for the external VFO. There are items that could be eliminated on the plug if you wanted to input the C.O. with an external BFO. There are other options not highlighted here that could be implemented.

I chose to change out the original crystal filter with an after market filter made for the Drake radio. The TR-7 2300 filter is a good fit as it has a 5.645 MHz center frequency and fits in the original mounting holes. The original Atlas filter is a 5.645 MHz edge frequency with a 5.646.2 MHz center frequency. The original filter also has a 2700 Hz band width (3300 Hz at -40 db) while the Drake filter I used has a 2300 Hz band width (2800 Hz at -40db). This necessitated modifications to the C.O. software to compensate for the difference.

I also modified the impedance of the output of the 1st IF stage which really only amounted to the addition of a parallel resistor to lower the impedance from 820 ohm to a more reasonable 370 ohms (R146 on PC100). The input of the Drake filter was represented to me to be 50 ohms, although, I suspect it is slightly higher than that. I also added a torid transformer core with 24 bifilar turns of wire (27uh) from the output of the crystal filter to the input of the 2nd IF stage (PC200) and removed the 1k ohm chassis resistor to ground to balance the impedance. The result was a slight improvement in gain which I balanced by adding a 5k ohm resistor on the back of PC200 to replace R203 3.3k ohm resistor.

The result of this modification was a boost in output power from the diminished 40 watts to nearly 100 watts on all bands. As the original Atlas crystal filter was faulty, I had no choice but to replace it and with an easy modification to the C.O. with software using a separate channel on the Si5351, it was a fairly simple task. Figuring out how to adapt the 1st and 2nd IF stages to efficiently use the new crystal filter was a bit more difficult. However, with a little trial and error, I am satisfied with the result. None of this is necessary if your original crystal filter is operating within acceptable parameters.

II. PROJECT SUMMARY

A. Project Tasks

The project tasks involve the following:

1) Purchase parts needed
2) Prepare the radio
3) Build the Power Supply Board
4) Assemble the boards
5) Install software
6) Testing / troubleshooting
7) Install crystal Filter
8) Modify Atlas hardware for new crystal filter
8) Software modification for crystal filter
9) Final testing

B. Parts List

Amazon

The VFO used in the project is a Si5351 I2C 25MHz Programmable Clock Generator Breakout Board 8KHz to 160MHz High Frequency Signal Generator. It uses an Arduino IDE I2C Controller SMA Connector 3.3V LDO. The Si5351 can generate any frequency up to 160 MHz on each of its outputs with an error of 0 ppm. It uses the onboard precision clock to drive multiple PLL's and clock dividers using I2C instructions. By setting up the PLL and dividers you can create precise and arbitrary frequencies. There are three independent outputs, and each one can have a different frequency. Can use 5V and converts the I2C signal to 3.3V.

The micro controller is a Mini 32 ESP32-WROVER-B PSRAM 520 KB SRAM WiFi Bluetooth Module Development Board,for QSPI Flash 4MB/PSRAM 8MB,for IPv4,IPv6,SSL,TCP/UDP. It designed to use 2.7 to 3.6 VDC although it also has a 5V, 1A voltage regulator built into the chip. It has a separate I2C interface plug that uses the typical ESP-32 IO21 and IO22. It can operate up to 4 separate I2C channels and two SPI channels.

Operator I/O is with a PCF8574T PCF8574 IO Expansion Board Module I2C IO Expander Module. The module is based on I2C interface of the I/O expansion module. The I2C address can be changed using the three pin plugs on the face of the module. Can be used on 5V or 3.3V. The project uses 3.3V to be compatible with the ESP-32 inputs.

A DS3231 AT24C32 IIC RTC 2 PCS Clock Module Real Time Clock Module is used to display the time on the display. I set it to UTC time for easy logging. It uses a rechargeable CR2032 battery and can be used on 5V or 3.3V. I used 3.3V to be compatible with ESP-32 inputs.
The project uses a 5V Regulator Module Mini Voltage Reducer DC 4.5-24V 12V 24V to 5V Adjustable Buck Converter Voltage Regulator Power Supply Transformer Module. I set it to 3.3V and connected it to the regulated 9VDC control voltage of the Atlas 210X.

I used a separate Buck Converter 6-24V to 5V 1.5A Step-Down Regulator Module Power Inverter Volt Stabilizer for the power to the Si5351 module to guarantee the frequency stability of module and connected it to the regulated 9VDC control voltage of the Atlas 210X. The Si5351 has it own 5V to 3.3V regulator to give compatibility with the ESP-32.

The operator interface uses a mini momentary toggle switch SPDT 3 Position 3 Pins (0N)-Off-(ON) Miniature Toggle Switch with waterproof cap MTS-123-MZ.

The following parts were purchased on Amazon although you can also purchase the same items from Ebay, Digi Key or Mouser, etc.

Si5351 signal generator

https://www.amazon.com/gp/product/B08GP8DBM1/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

Mini 32 ESP32 WROVER micro controller

https://www.amazon.com/gp/product/B08CNJDDYZ/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

PCF8574T IO Expansion Board

https://www.amazon.com/gp/product/B07XD2K4GH/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

DS3231 Clock Module

https://www.amazon.com/gp/product/B097PMNMQZ/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

Voltage Regulator 3.3V

https://www.amazon.com/gp/product/B09N6Y8YYN/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

Voltage Regulator 5V

https://www.amazon.com/gp/product/B076P4C42B/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

Ebay

The VFO is interfaced by a Oak / Grigsby Rotary Encoder - 128 Pulse/Rev - 91Q128 - 2 CH Full Quadrature. It satisfied certain requirements for a large number of pulse/rev, a smooth operating feel to the tuning dial and compact size to fit inside the VFO compartment without having to remove the original VFO circuit board.

The display screen used is a 1.77 inch 1.8 TFT Color Display Module Breakout SPI ST7735S. I selected this screen for its compact size which allows it to sit in the original Atlas 210X window and because it can be powered with 3.3V which makes it compatible with the ESP-32 I/O.

Parts purchased from Ebay:

Encoder

https://www.ebay.com/itm/284621140106

Display Screen

https://www.ebay.com/itm/224197366222

Project Assembly

The display screen sits in the original window that displayed the frequency. I mounted it using 1/32” plexi glass plastic that I cut and drilled. I made two supports, one for each side and mounted it through the original display plastic mounting holes. I added 2 holes on the right bracket to support the micro controller, Si5351, and a custom built power supply board with 4 pin I2C plugs for the expansion board and real time clock module. It made things nice and compact so that it would fit between the front panel and the variable tuning capacitor shaft.

I purchased all my connectors from Amazon such as the 4 pin I2C Qwiic Cable Kit, wire solderless breadboard jumper wires male to female, PCB Board Prototype Kit for Electronic Projects, 2.54MM 8 Pin Female Single Connector cables, 2.54MM 2 Pin Female Single Connector and 2.54mm 6Pin 8Pin 10Pin Female Socket Pin Headers. These items can be purchased from any good electronics parts house and or you can directly solder all connections. I like using pin headers and connectors to make assembly and dis-assembly easy for future parts replacement. I stacked all the boards on the ESP-32 micro controller by using M2 brass standoff kit hex column spacer standoff screws. By selecting the right spacer, you can trap the jumper wires between the boards to assure that it does not accidentally come apart.

The following is a list of the miscellaneous parts, cables and connectors used in the project:

I2C Quiic Cable

https://www.amazon.com/gp/product/B08HQ1VSVL/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

Jumper Wires

https://www.amazon.com/gp/product/B08151TQHG/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&psc=1

PCB Board

https://www.ebay.com/itm/113735091667?var=413873523311

8 pin connector cable

https://www.amazon.com/gp/product/B01IZALABE/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

2 pin connector cable

https://www.amazon.com/gp/product/B01IZAVWDA/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

Socket pin headers

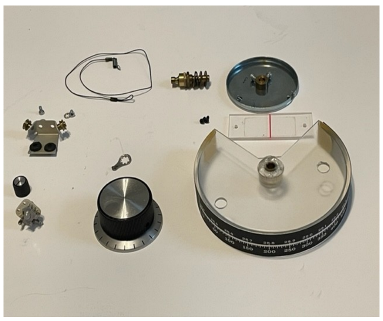
https://www.amazon.com/gp/product/B0778M5P1W/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

Stand off screws and spacers

https://www.amazon.com/gp/product/B07B9X1KY6/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

III PREPARE THE RADIO

The following parts will need to be removed from the Atlas 210X/215X radio:



1) plastic frequency drum (set screw must be loosened)
2) dial cord (spring loaded)
3) dial cord pulley bracket and the two dial lamps (two screws)
4) VFO tuning knob (set screws must be loosened)
5) plastic frequency window (two front panel screws must be removed)
6) metal pulley on the top of the VFO (it is secured to the shaft of the VFO tuning capacitor)
7) bottom cover of the VFO (to access the VFO compartment and band switches
8) VFO front panel vernier (lock nut , to make room for the encoder)
9) Dial Set variable knob and variable capacitor

Disconnect the existing VFO by opening compartment:

10) Orange White wire from VFO PC400 and set aside for VFO input
11) Orange wire from VFO PC400 and set aside to power new Digital VFO
12) 50 ohm coax cable connecting VFO to pin 3 of the 9 pin external socket (both ends)

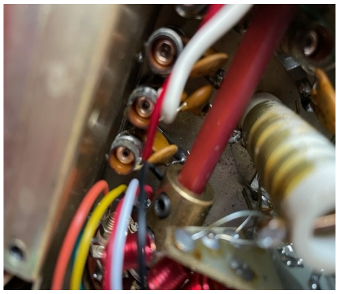
Remove the following if you are replacing the carrier oscillator (CO) PC600:

13) 50 ohm coax cable connecting CO to pin 6 of PC300
14) black/white wire on CO PC600 to be used for the CW input to ESP-32
15) 9V regulated power from P22 of PC200 to the CO PC600 - orange wire
16) 13V unregulated power from CO PC600 to pin 2 of PC300 - red wire

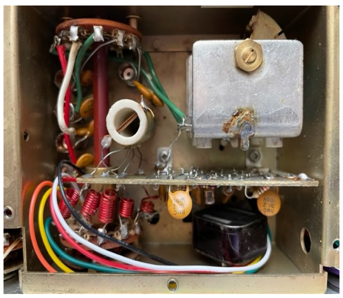
Modifications to Radio

Soldering Band Switch

1) remove screws holding VFO PC400



2) remove screws connecting band switch shaft and slide shaft coupling forward
3) remove band coils on band switch (2 to make room)
4) solder ground connection to first free gang switch 9” wire
5) prepare 8 pin connector cable by cutting length back to 9” long all 8 wires
6) route wires through open hole into the VFO compartment
7) solder 80m band to gang switch using black wire of connector
8) solder 40m band using red wire
9) solder 20m band using white wire

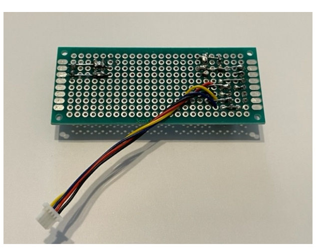


10) solder 15m band using yellow wire
11) solder 10m band using orange wire
12) reassemble components and cover VFO compartment
13) solder top of miniature toggle switch to the green wire (cut to 6” long)
14) solder bottom of toggle switch to the violet wire (cut to 6” long)
15) install miniature toggle switch in the dial set hole green wire up

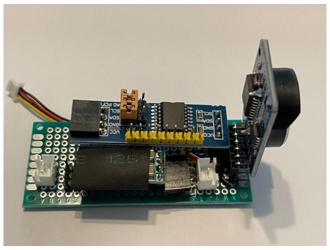
Make brackets for holding the new digital screen into place through the existing VFO dial.
I made a template to scale that you can print out and trace over Polycarbonate - .030" - 1/32" thick. Below is a link to where I purchased the plastic I used but you can also purchase something like this at Lowe’s or Home Depot.

https://www.amazon.com/gp/product/B00520AR9C/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

The two holes on the right side of bracket 2B are for the PC boards. I installed mine using M2 standoffs and screws. First the bracket then 8mm standoff then the micro controller. Second a 6mm standoff and the Si5351 facing away from the micro controller. Then a 10mm standoff and finally the power supply board.



Power supply board is made using a double Sided Universal PCB Proto Prototype Perf Board 3x7 3x7 cm. I mounted the plugs and pins on top and soldered it up under the board. Connect the power, gnd, SCL and SDA wires from the I2C pin plug of the micro controller to the clock board pins and the PCF8574 boards. Pay attention to the connections as the order of each wire is different for each board. Connect the 2 pin male plug to the 4 pin connector for the power module. Orientation of the module is important. Connect wires from the output of the power module to the pins at the foot as well as the additional male plug for power to the display screen.



Solder two female jumper wires cut off to 2 ½” for the SCL and SDA of the SI5351 and connect a minature 1mm 2 wire plug to the power leads of the SI5351. Connect the separate 5V DC power supply to the 9.2V regulated supply of the radio and put a 1mm 2 wire mating plug to the output of the power supply. Pay attention to the polarity of the pins. Red +, black Gnd.

IV. VFO

Assemble the screen to the brackets and connect the micro controller. I used the male pins supplied with the micro controller and connected everything up with female jumper wires. I made up an 8 pin connector cable for the connections to the display screen. I used female jumper wires for the connections to the micro controller and a two pin mating power plug to connect to the power supply board. Again, pay attention to the polarity. I jumper-ed the screen LED (violet wire) to the screen positive (red wire) and then both to the + of the power plug.

Assemble the Si5351 next. Connect 50 ohm coaxial cable to the radio. The Si5351 comes with SMA female plugs that can be soldered directly to the module. You can purchase one long 50 ohm R174 cable with male SMA cables on each end and cutting it to the correct length depending upon how you route the cable and connect each using a 0.1 uf ceramic capacitor. The CLK0 (VFO) cable goes to pin 3 of the 9 pin plug (Ext. Osc. or S1) of the radio used for external VFO. This pin is on the left side of the radio if you look from the front of the radio with the cover off. The CLK1 (C.O.) cable goes to pin 6 of PC300, again using a 0.1 uf ceramic capacitor. On my radio I made up a separate plug board using a piece of 3x4 mm Proto type PC board with two – 2pin male connectors on the Si5351 and two - 2pin female connectors with a 0.1 uf ceramic capacitor connected to the center lead of each and then to the 50 ohm R174 cable I happened to have in my radio shack.

Connect the Encoder to the 3.3V power supply on the PS board and the inputs to pins 4 and pins 25 of the micro controller using female jumper wires. You will have to splice into the Encoder ribbon cable. The black wire on the encoder ribbon cable is the ground wire. The next two wires are Encoder A (IO25) and Encoder B (IO04) inputs to micro controller. The last wire on the opposite side of the ribbon cable from the black wire is the 3.3V input wire.

Follow the Sketch for the Atlas 210x Digital VFO for connections and everything will work when you test it. By the way, I took my assembled VFO and connected it to my prototype board and tested and calibrated it and then installed the finished VFO into the radio. I designed this assembly so that it would be easy to pull in and out of the radio as a unit. You can also program the micro controller while installed in the radio just make certain you disconnect the power of the radio first. The power from the micro USB cable from your programming computer to the micro controller will power the VFO. You will have to jumper the power to Si5351 when you do this in order to calibrate the VFO.

Calibration of the Si5351 is straight forward. You will need an accurate frequency meter to measure the output. I used the 20m band with the VFO set to 14.250 MHz. The output of the VFO will be 8,605,000 Hz when using stock crystal filter and 8,606,400 with Drake TR-7 crystal filter. I powered my VFO to the USB of the computer, jumper-ed the Si5351 5V supply from the micro controller and let it set 24 hours before beginning calibration. You change the value of define CORRECTION 70780ULL near the top of the program. Increasing the CORRECTION value decreases the output frequency of the Si5351 and vise-verse increases the output frequency. Trial and error will get you to the correct " CORRECTION' value in a few tries. Test the memory select (3 memories A, B and C) by grounding the green wire tied to the PCF8574 or the dial step change with by grounding violet wire from the PCF8574. Test the bands and the Encoder and you are ready to connect the VFO to your radio.

If you are using CLK1 for your carrier oscillator then you should see 5,645,000 if you are using the stock crystal filter or 5,643,600 if you are using the Drake TR-7 crystal filter. When the VFO frequency is correct then the carrier oscillator is correct.

V. SOFTWARE

The software used is a combination of C and C++. I downloaded everything from the internet, primarily from Github. Almost every program and library was modified a little to accommodate the Atlas 210X radio. The main program VFOsystemSwitch06 and 07 were written by me although I did get the analog dial routines from T. Uebo – JF3HZB. He is a ham operator and university professor in Japan. I also got some ideas from Julio Ceasar of CeasarSound. He is a Brazilian business man and ham operator. Feel free to modify and or improve the software and any of the libraries. You are supposed to publish on the internet any changes you make to the programs or make them available upon request if you modify them. I am trying to get permission on Github to publish my changes. It should make things easier to download and you won't have publish anything as I have already done it.

I used the Arduino programming software for everything. It is available on internet for free. Just download the version you need for the computer you use (microsoft, linux or Mac). Just put all of the programs into the same sub directory that has the same name as the .ino file and load it with the Arduino software. Set up the correct micro controller board "FireBeatle-ESP32". Plug in the usb cable to the micro controller. Select the correct port. Compile the program and upload it to the micro controller. You may have to short pin 2 of the micro controller to ground to get the software to load. Once loaded, the screen will light up and show the dial and frequency.

When connected to the radio, it will go to the band you have selected on the band switch. If you connected it to your prototype board it will go the band selected on the prototype band switch. You can eliminate the prototype band switch by using a jumper wire to ground. Just ground the band input wire you want the VFO to use. You will have to ground IO pin 34 (VN) of the micro controller and connect pin 33 to 3.3V supply to get everything to boot up in default position. When you connect the Encoder you will be able to vary the output of the VFO.

It defaults to 1KHz steps which is very practical as almost everyone in the US tries very hard to transmit on frequency. Change the step to 500 Hz, 100 Hz or 10 Hz to fine tune the radio. The step frequencies can be modified in the void setstep1() sub routine in the main program.

The clock can be calibrated by removing the comment // marks in front of the following line in void setup() subroutine (set the date and time you want to see):

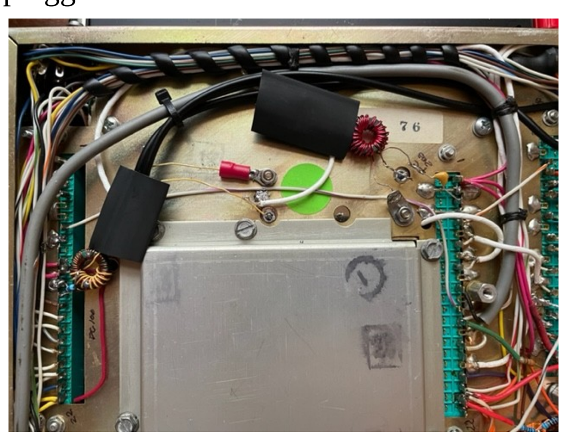
//rtc.adjust(DateTime(2022, 7, 17, 1, 07, 10)); // (yyyy, m, d, h, m, s)

Compile and Upload to calibrate and then put the comment // marks back in front of the line, compile again and upload again. Otherwise, if you leave the comment // marks off in the program it will calibrate the clock every time you boot the program.

VI. CRYSTAL FILTER

As I indicated in the intro letter, changing the crystal filter is not for the faint of heart. Having said that, this is a procedure for changing the stock 2.7KHz 5.645MHz crystal filter for a Drake TR-7 5.645 MHz 2.3Khz crystal filter. Typical crystal filters are specified by the center frequency. The atlas crystal filter was specified at the edge of the band pass frequency at about -40db. Measuring the filter gives a center frequency of 5.6466MHz with a shape factor of 1.6 also the standard filter was 2.7K wide at -6dB and 3.3K at -40dB. The Drake filter is specified at 5.645MHz center frequency with a shape factor of 1.7. I made the assumption that the Drake filter was 3.0KHz wide at -40 db which gave me the IF offset frequency of 5.6435MHz.

Using a Drake crystal Filter necessitates the use of CLK1 as the carrier oscillator as the stock C.O. PC600 cannot be tuned to this frequency and the OPP crystal would be further off frequency. I pulled the coax off pin 17 of PC300 and connected CLK1 of the Si5351 to the same pin tuned to 5.645MHz with the original crystal filter and everything just worked. I have since connected my other radio to CLK1 and am very happy with the solution. I made an input to the micro controller for the CW position and programmed a 1200Hz offset for CW an transmitter output testing. I also made an input for Norm/Opp positions for the SSB switch with a 3300Hz offset for the stock filter and a 3000Hz offset for the Drake filter. When I plugged in these numbers to the Si5351 I found that the radio was right on frequency.



Dealing with the impedance difference is another matter. The stock filter is about 800 ohm impedance and the Drake filter has a 50 ohm impedance. A ½” diameter toroid with a primary of 40 turns and a secondary of 10 turns (4:1) is used. This gives an impedance ratio of 16:1. One lead of the primary is connected to pin 13 of PC 120 and the other end in series with a 1000 ohm resistor to ground. The secondary is connected to the input of the crystal filter and the other end to ground. An acceptable match from the filter to PC-200 is acomplished by using a torid core with 24 bifilar turns of wire (27uh). This is similar to the torid described by Clint in his Engineering Supplement but with more turns. Make sure you adjust the transmitter as described in the Atlas Service Bulletins and tune the IF coils. Works like a champ!

The results received were higher gain than the stock filter and of course a narrower bandwidth. Most contacts made were received with good audio reports. One report was received from someone that noticed the narrower bandwidth but he also indicated that the audio was good. I have noticed from my end that the wider bandwidth with the stock filter on a different radio gives a more robust signal on receive but I also hear more more interference from adjacent stations. I suppose this is to be expected.

In the program the variables ifFreq, oppFreq and cwFreq contain the following values:

Stock Filter Norm: IF offset 5645000Hz	Drake Filter Norm: IF offset 5643500Hz
Stock Filter OPP: IF offset 5648300Hz	Drake Filter OPP: IF offset 5646500Hz
Stock Filter CW: IF offset 5646200Hz	Drake Filter CW: IF offset 5644700Hz

There is also a #define IF for calculating the VFO frequency:

Stock Filter #define IF 5645000	Drake Filter #define IF 5643500

VII. USER INTERFACE

Once installed in the radio the user interface is as easy as the Atlas 210X to operate. Change the band position switch and the band will change on the dial. Toggle the mini toggle switch up and you will change the memory to A, B or C. The program will remember the memories if

you switch to another band and come back as long as you leave memory A displayed. Toggle the mini toggle switch down and you will change the step frequency the encoder advances on the dial. I like the default value of 1KHz and only change this value if someone is operating off frequency to fine tune.

Unplug the 9 pin plug on the back of the radio and the display will stop changing with the encoder and will display off rather than MHz on the digital display. This will permit you to plug in an external VFO to control the radio. You won't see the frequency on the radio and it remains where it was when you disconnected the plug.

If you are using the Si5351 for your carrier oscillator then when you switch the radio mode switch to CW it should work as normal and and transmit with a carrier at 1200 Hz above the normal frequency.

Leave the SSB switch to Norm and you get USB for 14MHz and above and LSB for 7.3MHz and below. Switch to OPP and you will get the opposite side band for the above.

That's it, have fun!

John – KI5IDZ

APPENDIX