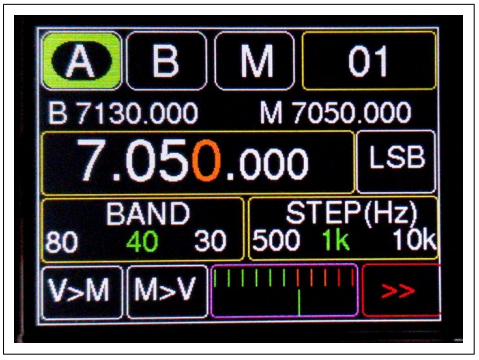
TFT/ Touch VFO/BFO with CAT for uBitx/Bitx (Ver6.x)

SP Bhatnagar (VU2SPF)
vu2spf@gmail.com

(Ver 6.2, 10th Nov 2021)

1. Introduction:



A colored display with touch screen based control of a rig gives user an easy to operate system. Some of the homebrewed rigs and semi-assembled rigs provide limited text LCD displays which are simple looking and sometimes not easy to operate. The TFT with touch screens have become quite easily available at affordable price and along with micro-controllers, like Arduino, provide good combination of looks and control. MCUFriend family of TFTs are designed specifically to physically match with standard Arduino boards(Uno, Mega etc) and provide a single unit to be incorporated in the homebrewed transceivers. We have attempted to create such controllers for Bitx and uBitx.

Our earlier version of this display system was designed to be quite flexible for experimenters and with number of features incorporated, the display became quite congested, button sizes had to be reduced and one really needed a stylus to operate the TFT by touch. A need was felt to incorporate a few more features and to simplify the display, which gave birth to a new User Interface (from ver 5.x) with larger touch buttons and multiple screens to take care of less frequently used features. (Older version link : https://vu2spf.blogspot.com/2018/08/new-version-3.html and http://vu2spf.blogspot.com/2020/04/tft-touch-screen-vfobfo-for-ubitx-with.html). Incorporation of new smooth fonts since ver 6.x makes better looking displays. Several features including the colour scheme can be modified as per user's choice, from within the program (details below in sec 7).

The low cost Si5351 programmable generator ICs were introduced a few years back and experimentation by many hams found these to be suitable for use with Bitx/uBitx. We have used the Si5351 in many versions of Bitx's (more details on vu2spf.blogspot.com).

CAT emulation in the program is helpful for operating the digital modes and computer control of the rig. It was introduced since version 5.x in our program. It emulates Yaesu's new cat commands like in FT-2000 and latest rig models. Only essential / useful CAT commands are available while more can be added in the program as described in a later section. The CAT library is also made available for anyone to use.

It is possible to attach additional physical buttons for many of the general functions(details below) to supplement / substitute the touch buttons.

Interestingly this VFO/BFO system can also be used with single conversion systems like Bitx. The desired parameters can be adjusted / modified from within the program, as discussed below.

1.1 Improvements: The new UI (since ver 5.x) was designed to simplify the display and to make it usable with fingers rather than needing a stylus. The latest version 6.x has better looking smoother fonts and a new option for Tuning Antenna Tuners etc. The UI was also updated and a few additional facilities added. The main screen has only those buttons required for day to day operation of a rig. All

adjustable parameters are shifted to other screens, of course with larger characters and larger easy to operate touch buttons. In addition to existing features, in this new avatar, all the essential parameters stored in memory also can be backed up and restored from another portion of memory. A dump of all parameters in memory, on serial terminal, is another feature which can be used to copy/ store essential settings on a PC. These two features allow experimentation by playing with differen parameters without fear of messing up.

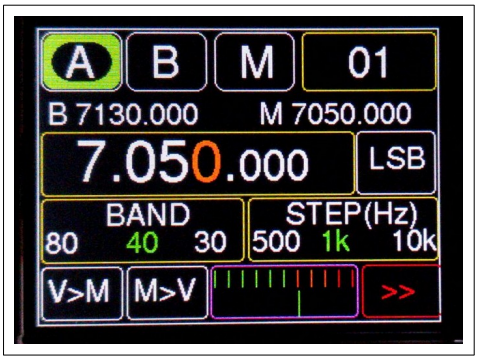
1.2 Types of Touch Button: All touch buttons are surrounded by a colored border. The white borders indicate buttons which are of touch type (like push to on), the yellow bordered buttons are two sided, whose content changes by touching the left or right half of the button (like volume control on cell phones) and those with Green border are Labels which are for information only. The Red bordered buttons are for navigating to next or previous screen. (the colours are not captured correctly in the photos of TFT display, in the photo white looks bluish)

1.3 Screens: We call the four different screens as Screen 0 (Main), Screen 1 (BFOs and Offsets), Screen 2(PTT, Smeter and Touch Sensitivity) and Screen 3 (Memory related Operations and Tune). Each of these are described below.

1.4 User Modification of Parameters : Several parametrs can be selected or adjusted by users from within the program. Details are given in sec. 6 below.

2. Main Screen (0)

This is the operational touch screen from where one of the three VFO's may be selected, VFO frequencies adusted, sideband selected, ham bands selected, step size for tuning adusted and exchange between VFO and Memory can be made. Each line/ button is discussed below.



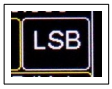
2.1 VFO Buttons: This program provides three VFOs. Switcing between the VFOs is possible by touching appropriate button on screen. As seen in the picture, the three VFOs are labeled as A, B and M (memory or channel). One of the VFO is selected by touching the corresponding button once and this selection is indicated by its Green colour. A second touch on the same VFO button will put the rig in Transmit mode and its button colour becomes Red. This is equivalent of PTT on microphone which may also be connected parallelly. (Note that the A, B and M buttons are white bordered touch buttons). If VFO is changed while in Tx mode, the rig will be first put to Rx mode and then change the VFO. There are 100 memory channels for storing frequently accessed stations. The M VFO button has a channel count displayed on its right which indicates the channel number selected [01 to 100]. The number can be incremented or decremented by touching the elongated channel nuumber button on its right or left half. If the step size [described below in sec 2.5] is 10kHz this channel counter changes in steps of 10, when the step size is less than 10kHz the change is in steps of 1. When the rig is in Tx mode the memory channel can not be changed.

2.2 Unselected VFOs: Currently set frequencies of other two unselected/ inactive VFOs are shown immediately below top line VFO indicators. These are meant to be for information only. This is not a button and has no effect on touching this area.

2.3 Operating frequency (VFO): The current frequency of operation is displayed in larger fonts near the center. The frequency could be changed by touching this double sided button on its left or right half. (Its a yellow bordered button). A rotary encoder connected to proper pins of Arduino also varies the VFO frequency. Depending upon the step size selected the corresponding digit is displayed in a different colour. The colour of this variable digit can be user selected. During Tx, the frequency cannot be altered.

VFO Band Limits: Legal band limits are set inside the program and can be modified by user. Changing the band limits as per requirements of different regions is explained below in sec 7. Initially one frequency within each band is programmed as default start frequency. For changing the default start frequency, tune to the desired frequency and save it using SAVE button on one of the following screens.

2.4 Side Band: Lower or Upper Side-band (LSB / USB) is selected by the button to the right of VFO. Currently only these two are available.



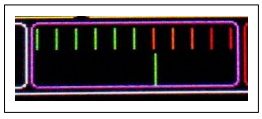
2.5 Band and Step Size : One of the pre-programmed band can be chosen using the two sided Band button. Selected predefined amateur band is displayed in Green at the center of the band button and the next lower or higher band may be selected by touching left or right half of the button near the the white lettered band names, on both sides. This is a round robin type of display and rotates in both the directions.

A special “GC” (General Coverage) band covers full HF (1-30MHz excluding predefined ham bands) and is meant for General coverage receive. Tx is not allowed when GC band is selected / and VFO frequency is not within allowed band limits.

On the right side of Band button a similar display and button is used for selecting the step size which is used for changing the frequencies, either by touch or by encoder. During Tx the band change is not allowed.

2.6 VFO ↔ Memory exchanges: The V> M button copies and saves the currently selected VFO to currently selected memory channel. On the other hand the M>V button will copy (not save) the frequency in current memory channel to the selected VFO. If you want to store your daily net frequency just tune in to the net frequency in VFO A or B, select memory channel where it is to be stored (say ch#10), using the channel counter next to M button, and then touch V>M button. Briefly its colour will change to Red to indicate memory write operation. Next whenever you wish to tune in to that frequency just select channel 10 and VFO M. Alternately it may be copied to currently selected VFO A or B by M>V button. M>V button does not operate during Tx.

2.7 A simple S-meter shows audio derived signal strength by a moving pointer. This is a relative meter and its upper and lower limits are set from Screen 2.



2.8 Navigation: The red bordered button with “>>” arrow opens next Screen (1).

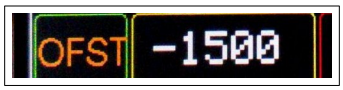
3. BFO / Offset Screen (1)

BFO and Offset adjustment: There are two BFOs in uBitx. Both need to be tuned for getting correct reception and transmission. The changes on this screen are affected by the step size selected in the Main screen (it is easy to step back to main screen using arrow buttons - described below). Fine tuning of BFOs with 1 Hz step is possible. This screen has three main components which are adusted using the double sided touch buttons. It is very illustrative and educative to hear and understand the output while varying the BFOs. When using a single conversion system like bitx, one of the two BFO, is used.

3.1 VFO Frequency: On top line it displays current frequency of operation. This frequency can be adjusted by touch or rotary encoder as in the main screen.

3.2 BFO1 and BFO2 controls: The two BFOs are displayed next. Both BFOs are adjustable with the step size defined in Screen 0. When a station is tuned, the two BFOs are adjusted to get the best audio. In uBitx, BFO1 needs to be **around** 33MHz for LSB and **about** 57MHz for USB while BFO2 is tuned to fixed frequency of 12MHz (actual frequency of BFO2 may be 11.993 MHz). These values are varied to get the best audio for a known frequency station (initially it may be somewhere near the known frequency).

3.3 Offset adjustment: If the station does not tune on a known frequency but somewhere nearby, the offset adjustment will bring it to the correct frequency. Each band has separate offset. Offsets can have both negative or positive values. For adusting the offset tune in to the actual frequecy and incrment or decrement offset by touching the left and right halves of the button to get the best reception. Step size in the previous screen is also effective here.



3.4 Previous & Next screen navigation : The Red bordered left and right buttons are for moving to previous and next screens, (similarly shown in all the following screens).

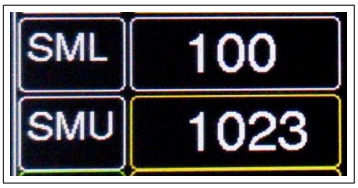
4. Modes and Timeouts Screen (2)

Several parameters are adjusted from this screen. These help in operation of the rig. Those buttons on the left side of display have two parts : the label and its value.

4.1 PTT Modes: There are two possible modes for PTT viz Normal and Toggle PTT. In Normal mode every press of PTT will activate Tx which will remain in Tx as long as the PTT is kept pressed and on releasing the PTT the rig goes back to Rx. Alternatively, in the Toggle mode one short press of the PTT switch will start Tx while the second short press will bring it back to Rx mode. The two modes are selected by touching the button next to PTT label (green bordered). The toggle mode can be combined with a voice recording and playback module to work as voice keyer. Selecting PTT by touch is of Toggle type. Don't forget to SAVE your selection.

4.2 PTT Timeout: Touching the PTO button (white border touch type) activates the automatic time-out for Tx. The time out feature helps to reduce unnecessary heating of the finals. It is specially useful in digital modes and auto cq call using voice keyer. Time out period is displayed in seconds and is adjustable from the button next to PTO. White fonts of Time out period indicates inactive PTO while its font color changes to Red when active.

4.3 S Meter settings: The S-meter display on Screen 0 is audio derived. We need to set its lower and upper limit to match with the approximate S values. The lower limit is set by tuning to a quite spot on band (or disconnect antenna) and touching the SML button. It will read the current S-meter input and displays that value in the box next. Fine adjustments to this value are possible by touching left or right side of this button. Similarly the Upper limit for the S-meter is adjusted by tuning to a strong station and touching SMU button. Absolute minimum and maximum value for these two S-meter limits are 0 and 1023.



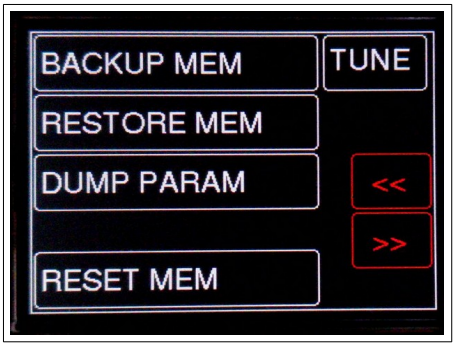
4.4 Touch Sensitivity Control: Different touch screens may have different sensitivities to touch. More sensitive ones respond fast and may need to be slowed down, otherwise they may record multiple touch for each real touch. The TS parameter adjusts this Touch Sensitivity and has to be done by experimenting. Most touch screens tested work well with touch sensitivity values between 80 and 100 .

4.5 Save Parameters: The SAVE button saves all adjustable parameters to memory for use later. The button will change colour while the save operation is in progress.

4.6 Split Mode operation: The SPLT button invokes split mode where VFO A works as Rx frequency and VFO B as Tx frequency. Second touch on SPLT button will deactivate this mode. During the SPLIT mode the text and colour of this button changes and similarly the secondary frequency display on Screen 0 also changes colours (Green Rx and Red Tx). During Split mode if VFO M is selected the split mode is turned off.

5. Memory and Essential Constants Screen (3)

When this program is run for the very first time or when the magic number (line 18 magic_no) is changed the initial values of the various parameters is stored in EEPROM. Also whenever Save button is activated the parameters are refreshed in EEPROM.



5.1 Backup Mem : This button creates a back up of all saved parameters, in a different location within the EEPROM. Usually this should be done once after setting up the radio for the first time or whenever adjustments are made for better signal.

5.2 Restore Mem button: will restore back memory from Backup copy onto original location. Thus while playing with various parameters if any change is made and saved it can be recovered.

5.3 Dump Param button: sends these parameters in memory to a serial terminal, from where these could be printed or stored. Very handy to keep a record of the essential parameter settings.

5.4 Reset Mem button: will reset all parameters in memory to factory settings except the memory channels which a user might have saved.

5.5 Tune Button : Introduced since ver 6.x, this button is activated by first touch and deactivated on second touch. While active, it generates a constant carrier at the dial frequency, which may be used to tune the antenna tuner etc.

6. CAT Emulation / Control

This version is now capable of handling CAT commands for general digital communiction. For using this the rig should be selected as Yaesu FT-2000 and speed set to 38400 baud. WSJTX, FLDIGI, JS8CALL have been tested to work with this program.

Connect using a USB Cable from Arduino USB port to PC's USB port. In the communication package select "Yaesu FT2000" rig, baud 38400, CAT conrol. Test by changing frequency by dial or by program. It should be same on both. Audio between the rig and PC has to be connected using a digital interface. If there is some delay or rig not detected, try to use a good data cable. When initializing, the Arduino will reset, which is normal.

7. User modifications in userdefs.h tab

In this program functions are grouped by their utility. There are several tabs (or files) containing various groups of functions. Several parameters, as listed below, are user controllable. These are listed in userdefs.h tab of the program. Users can edit and modify these as suitable for their rig and their liking.

Tweaking Band Limits: Current band limits are set in the program as per prevailing regulations in VU land. It is easy to change the band limits in the program. The userdefs.h tab of the program contains the following lines :

```
// Band Limits and frequencies and their corresponding display on band button
volatile uint32_t F_MIN_T[9] = {100000UL,  3500000UL, 7000000UL, 10100000UL,
14000000UL, 18068000UL, 21000000UL, 24890000UL, 28000000UL};

volatile uint32_t  F_MAX_T[9] = {30000000UL,  3800000UL, 7200000UL, 10150000UL,
14350000UL, 18168000UL, 21450000UL, 24990000UL, 29700000UL};

String  B_NAME_T[] = { "VFO", "80m", "40m", "30m", "20m", "17m", "15m", "12m",
"10m" };
```

For example to modify the upper limit on 40 m band to say 7300 kHz , change the 7200000UL to 7300000UL in F_MAX_T[9] array. F_MIN_T[9] array has lower band limits.

7.1 Tweaking Step Sizes: The default/ initial step size is 1kHz and a different one can be selected from the userdefs.h file.

```
String step_sz_txt[] = {"  1", " 10", " 50", "100", "500", " 1k", "10k"};   // How
the step size is displayed
uint32_t step_sz[] = {   1,   10,   50,   100,   500,   1000, 10000};   // step
size values
int step_index = 5; // 5 is i1KHz as initially slected step size (6th element in
above array counting from 0)
uint32_t radix = step_sz[step_index];  //index so radix = 1000
```

By changing the value of `step_index` the default step size can be adjusted.

7.2 Select your colors on display: Since ver 6.x the background colour, colours of buttons, their outlines and texts can be selected by user, again from usrdefs.h. In the section on button color parameters, select your preferred colours and enjoy.

7.3 Using with other rigs: Currently only ubitx and bitx are defined and more can be added. Select the rig type (ubitx or bitx) by removing "//" (unselect the other one by putting "//")

```
#define ubitx
//#define bitx
```

For bitx Select the IF the rig uses from the ststements below (any one) b removing "//".

```
//#define IF_12MHZ
//#define IF_10MHZ
```

For other rigs using different IF / BFO/ VFO the actual values may be defined in the lines below these.

7.4 Factory Reset: If sometimes a factory reset is needed, just by changing the value of the variable "magic_no" to any other value, than what is set currently. For example if currently

`magic_no = 03;` then making it `magic_no = 04;`

and reloading the program will reset everything to initial state.

Version_Info: If the PTT is kept pressed at power up, a banner with Version information is displayed, for as long as PTT remains pressed.

8. Hardware

Details published earlier at : https://vu2spf.blogspot.com/2018/05/digital-inerface-card-for-ubitx.html

Arduino Mega 2560 with a Mcufriend type TFT display is needed to control the Si5351 generator. The TFT shield sits on top of Arduino, thus making it one unit to handle. Mounting on front panel of the rig is also therefore easy. A small PCB is designed to connect the uBitx to Arduino. (Si5351 breakout boards can also be connected to Arduino and to uBitx.) On this PCB two additional circuits for Tx POP reduction and audio derived S-meter are also included for the users of earlier versions of uBitx. This small PCB plugs into the uBitx board at one end and the connector on the other end is connected to Arduino with parallel wires. Currently this replaces the Raduino on uBitx.

Display: Various TFT screens of MCUfriend type are useful for this application. The button and text sizes are automatically scaled to match the 2.4" to 3.5" displays. In case a TFT is not displaying the screens properly or the touch buttons are not working as per design the following procedure is to be followed.

1. From MCUFriend_kbv library (https://github.com/prenticedavid/MCUFRIEND_kbv) load and run the various utilities in examples folder to test and check the TFT display and touch.
2. Load the Touch_shield_new sketch in Arduino and carry out the calibration of touch panel. Please open the Serial Monitor when running this program. It will display (i) Display size ,
e.g. LANDSCAPE CALIBRATION 480 x 320

(ii) touch screen pin connections :

Output of Touch_shield_new sketch	To be written in our program as (around line 330)
Touch Pin Wiring XP=8 XM=A2	XM = A2, XP = 8 ;
YP=A3 YM=9	YP = A3, YM = 9;

(sometimes A1 is reported as 54, A2 as 55, A3 as 56 etc)

(iii) touch coordinate conversion expressions :

Output of Touch_shield_new sketch	To be written in our program as (around line 580)
x = map(p.y, LEFT=956, RT=97, 0, 480)	xpos = map(tp.y, TS_LEFT = 956, TS_RT = 97, 0, 480);
y = map(p.x, TOP=921, BOT=126, 0, 320)	ypos =map(tp.x, TS_TOP = 921, TS_BOT = 126, 0, 320);

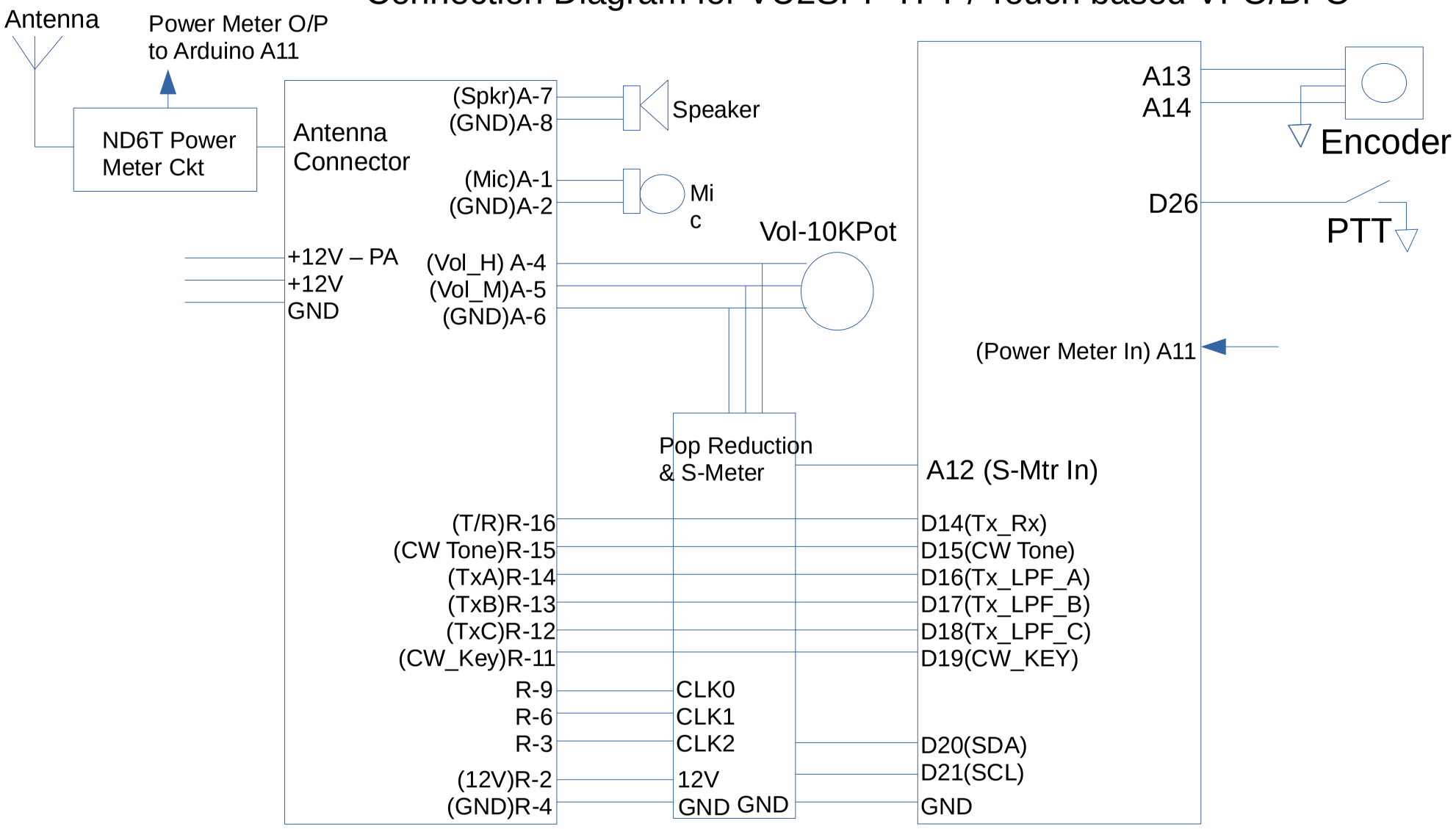
These details must be put in the program at around line numbers 330 and 580.
Most of the time the display and touch work well with these data. In case the touch is not at proper location, one has to try playing with these coordinates of left, right, top and bottom. If touch is inverted in x (touching right affects button on left) exchange xpos and ypos names. If the y is inverted (touch in upper half affects button in lower half) exchange tp.x and tp.y in these expressions.
identifier == 0x9320 is the type of controller reported by one of the example programs. If your display identifier does not exist in the program one of the existing identifiers may be substituted along with other details.

Acknowledgement: The initial development of this project was with very active support of OM **Joe Basque, (**VE1BWV, joeman2116@gmail.com), without whose help this project would be still languishing. I thank Joe for his help and very quick feedbacks.

Many Hams and other enthusiasts have opened up their creations for the whole community and society's benefit. We thank them for their gracious efforts and sharing their creations. We are highly motivated by their generousness. Many Hams have provided great feedback on the previous versions of this program. Thanks are due to all of them too. Please keep sending your suggestions and comments for improving these small efforts

73's

Connection Diagram for VU2SPF TFT / Touch based VFO/BFO



R-x uBitx Raduino connector pins,
A-x uBitx Audio Connector pins