```
  //TFT LCD radio control sw
#include <UTFT.h>
#include <UTouch.h>
#include <UTFT_Buttons.h>
#include <Encoder.h>
#include "si5351.h"
#include "Wire.h"

// Declare which fonts we will be using
extern uint8_t numeric_xl[];
extern uint8_t numeric_m[];
extern uint8_t SmallFont[];
extern uint8_t BigFont[];
extern uint8_t SevenSegNumFont[];

long enc_old  = 0;
long IF = 19995900; //IF frequency
int cw_shift = 0; //cw offset
long LSB = 19995900; //LSB frequency
long USB = 19998300; //USB frequency
long CW_TX = 19996470; //CW_TX frequency
long vfo = 0;
long vfo_A = 0;
long vfo_B = 0;
long vfoA_buff = 0;
int vfoB_khz_old = 0;
long vfoB_buff = 0;
int vfoA_khz_old = 0;
byte menu_index = 0;
byte mode = 1; //0->LSB 1->USB 2->CW
boolean att = false;
boolean split = false;
char osc = 'A';

int freq_step = 1;
int s_value_old = 0;

byte band_index = 0;
const byte band_data[8] = {3,7,10,14,18,21,24,28};
const long vfo_start[8] = {3500000,7000000,10100000,14000000,18068000,21000000,24890000,28000000};

boolean disp_first = true;
boolean transmit = false;
boolean filt = false;
boolean cwmode = false;
boolean tune = false;




Si5351 si5351;

UTFT myGLCD(ITDB32S,38,39,40,41);   D38 D39 D40 D41
UTouch myTouch(46,45,44,43,42); D46 D45 D44 D43 D42
UTFT_Buttons myButtons(&myGLCD, &myTouch);
Encoder myEnc(18,19); D18 D19


void setup()
{
  //I/O setup
  pinMode(3,OUTPUT); //T/R relay D3
```

```
digitalWrite(3,LOW);
pinMode(A0,INPUT); //s meter input A0
pinMode(6,OUTPUT); //cw filter D6
pinMode(7,OUTPUT); //cw filter neg D7
pinMode(5,OUTPUT); //ATT control D5
pinMode(8,OUTPUT); //band 24-28 D8
pinMode(9,OUTPUT); //band 10 D9
pinMode(10,OUTPUT); //band 7 D10
pinMode(11,OUTPUT); //band 18-21 D11
pinMode(12,OUTPUT); //band 3.5 D12
pinMode(13,OUTPUT); //band 14 D13
pinMode(4,INPUT_PULLUP); //T/R switch D4
digitalWrite(4,HIGH);
pinMode(2,OUTPUT); //during CW mode is high D2
digitalWrite(2,LOW);
pinMode(1,OUTPUT); //mute D1
digitalWrite(1,LOW);
pinMode(0,OUTPUT); //during SSB mode is high it switches the first tx amp and key D0
digitalWrite(1,LOW);



digitalWrite(6,LOW);
digitalWrite(7,HIGH);
digitalWrite(5,LOW);

vfo = vfo_A = vfo_B = vfo_start[band_index];
si5351.init(SI5351_CRYSTAL_LOAD_8PF); D20 SDA Si5351_SDA   D21 SCL Si5351_SCL
si5351.set_pll(SI5351_PLL_FIXED, SI5351_PLLA);
si5351.set_freq(IF+vfo, SI5351_PLL_FIXED, SI5351_CLK0);
si5351.set_freq(LSB, 0, SI5351_CLK1);

myGLCD.InitLCD(LANDSCAPE);
myTouch.InitTouch(LANDSCAPE);
myTouch.setPrecision(PREC_MEDIUM);
myButtons.setTextFont(BigFont);

myGLCD.clrScr();
myGLCD.setFont(BigFont);
myGLCD.print("by HA8SA", CENTER, 100);
delay(1000);
//background painting
    int g = 255;
    for (int y = 0; y<239;y++)
     {
      myGLCD.setColor(0,0,g);
      myGLCD.drawLine(0,y,319,y);
      g--;
     }
 freq_disp_A();
 disp_first = true;
 freq_disp_B();
 band_switch(); //set bpf board

 myGLCD.setFont(BigFont);
 myGLCD.setColor(VGA_WHITE);
 myGLCD.setBackColor(VGA_RED);
 myGLCD.print("A",8,90);
 myGLCD.setBackColor(VGA_TRANSPARENT);
 myGLCD.print("SPLIT",38,90);
 myGLCD.setBackColor(VGA_TRANSPARENT);
 myGLCD.print("PRE",6,114);
 myGLCD.setBackColor(VGA_TRANSPARENT);
```

```
  myGLCD.print("LSB",70,114);
  myGLCD.drawLine(4,109,116,109);

  s_bar_print();

}


void loop()
{
int A_button, B_button, C_button, D_button, pressed_button;
int jj = 40;
//buttons
myButtons.setButtonColors(VGA_WHITE, VGA_GRAY, VGA_SILVER, VGA_RED, VGA_BLACK);
D_button = myButtons.addButton(239, 189, 80, 50, "SLOW");
C_button = myButtons.addButton(159, 189, 80, 50, "UP");
B_button = myButtons.addButton(79, 189, 80, 50, "DOWN");
A_button = myButtons.addButton(0, 189, 80, 50, "MENU");
myButtons.drawButtons();
while(1)
{
long enc_new = myEnc.read();
 if (enc_new != enc_old)
 {
  if (enc_new < enc_old)
   vfo -= (enc_old-enc_new)*freq_step;
    else
    vfo += (enc_new-enc_old)*freq_step;
  enc_old = enc_new;
   if (osc=='A')
    freq_disp_A();
     else
      freq_disp_B();
  si5351.set_freq(IF+vfo+cw_shift, SI5351_PLL_FIXED, SI5351_CLK0);
 }

//handling T/R switch
 if ((digitalRead(4)==LOW)||(tune==true))
  {
   if (transmit==false)
   {
   if (cwmode==true)
     si5351.set_freq(CW_TX, 0, SI5351_CLK1);  //+700HZ
   digitalWrite(1,HIGH); //mute
   digitalWrite(6,LOW);
   digitalWrite(7,LOW);
   delay(1);
   digitalWrite(3,HIGH); //T/R relay
   transmit = true;
   }
  }
  else
  {
   if (transmit==true)
   {
    si5351.set_freq(LSB, 0, SI5351_CLK1);
    digitalWrite(3,LOW); //T/R relay
    transmit = false;
    delay(50);
       if (filt == false)
          {
           digitalWrite(6,LOW);
           digitalWrite(7,HIGH);
```

```
          }
          else
          {
           digitalWrite(6,HIGH);
           digitalWrite(7,LOW);
          }
     delay(50);
     digitalWrite(1,LOW);
   }
  }
```

```
 //handling touch buttons
   if (myTouch.dataAvailable() == true)
    {
    pressed_button = myButtons.checkButtons();
    if (pressed_button==A_button)
     {
      switch(menu_index)
       {
       case 0:
        menu_index = 1;
        myButtons.relabelButton(B_button,"A/B",true);
        myButtons.relabelButton(C_button,"SPLT",true);
        myButtons.relabelButton(D_button,"MODE",true);
       break;
       case 1:
        menu_index = 2;
        myButtons.relabelButton(B_button,"ATT",true);
        myButtons.relabelButton(C_button,"TUNE",true);
        myButtons.relabelButton(D_button,"FILT",true);
       break;
       case 2:
        menu_index = 0;
        myButtons.relabelButton(B_button,"DOWN",true);
        myButtons.relabelButton(C_button,"UP",true);
        myButtons.relabelButton(D_button,"SLOW",true); freq_step = 1;

       break;
      }
    }

   switch (menu_index)
   {
 case 0:
     if (pressed_button==D_button)
     {
      if (freq_step==1)
      {
       myButtons.relabelButton(D_button,"FAST",true);
       freq_step = 10;
      }
     else
      {
       myButtons.relabelButton(D_button,"SLOW",true);
       freq_step = 1;
      }
     }
     if (pressed_button==C_button) //band up
```

```
{
 if (band_index < 7)
  {
   band_index++;
   band_switch();
   vfo = vfo_A = vfo_B = vfo_start[band_index];
   disp_first = true;
   freq_disp_A();
   disp_first = true;
   freq_disp_B();
   si5351.set_freq(IF+vfo+cw_shift, SI5351_PLL_FIXED, SI5351_CLK0);
  }
}
if (pressed_button==B_button) //band down
{
 if (band_index > 0)
  {
   band_index--;
   band_switch();
   vfo = vfo_A = vfo_B = vfo_start[band_index];
   disp_first = true;
   freq_disp_A();
   disp_first = true;
   freq_disp_B();
   si5351.set_freq(IF+vfo+cw_shift, SI5351_PLL_FIXED, SI5351_CLK0);
  }
}
break;
case 1:
 if (pressed_button==B_button) //a/b
   {
    if (osc=='A')
     {
      osc = 'B';
      vfo_A = vfo;
      vfo = vfo_B;
      si5351.set_freq(IF+vfo+cw_shift, SI5351_PLL_FIXED, SI5351_CLK0);
      myGLCD.setFont(BigFont);
      myGLCD.setColor(VGA_WHITE);
      myGLCD.setBackColor(VGA_GREEN);
      myGLCD.print("B",8,90);
     }
     else
     {
      osc = 'A';
      vfo_B = vfo;
      vfo = vfo_A;
      si5351.set_freq(IF+vfo+cw_shift, SI5351_PLL_FIXED, SI5351_CLK0);
      myGLCD.setFont(BigFont);
      myGLCD.setColor(VGA_WHITE);
      myGLCD.setBackColor(VGA_RED);
      myGLCD.print("A",8,90);
     }
   }
 if (pressed_button==C_button) //split
   {
    if (split==true)
     {
      split = false;
      myGLCD.setFont(BigFont);
      myGLCD.setColor(VGA_WHITE);
      myGLCD.setBackColor(0,0,160);
      myGLCD.print("SPLIT",38,90);
```

```
      }
      else
      {
       split = true;
       myGLCD.setFont(BigFont);
       myGLCD.setColor(VGA_WHITE);
       myGLCD.setBackColor(VGA_RED);
       myGLCD.print("SPLIT",38,90);
      }
    }
  if (pressed_button==D_button) //mode
    {
     switch (mode)
      {
       case 0: //LSB
        cwmode = false;
        cw_shift = 0;
        digitalWrite(2,LOW);
        digitalWrite(0,HIGH);
        si5351.set_freq(LSB, 0, SI5351_CLK1);
        myGLCD.setFont(BigFont);
        myGLCD.setColor(VGA_WHITE);
        myGLCD.setBackColor(0,0,136);
        myGLCD.print("LSB",70,114);
        mode++;
       break;
       case 1: //USB
        si5351.set_freq(USB, 0, SI5351_CLK1);
        myGLCD.setFont(BigFont);
        myGLCD.setColor(VGA_WHITE);
        myGLCD.setBackColor(0,0,136);
        myGLCD.print("USB",70,114);
        mode++;
       break;
       case 2: //CW
        cwmode = true;
        cw_shift = 570;
        si5351.set_freq(LSB, 0, SI5351_CLK1);
        myGLCD.setFont(BigFont);
        myGLCD.setColor(VGA_WHITE);
        myGLCD.setBackColor(0,0,136);
        myGLCD.print(" CW ",62,114);
        digitalWrite(2,HIGH);
        digitalWrite(0,LOW);
        mode=0;
       break;
      }
    }
 break;
case 2:
  if (pressed_button==B_button) //ATT
    {
     if (att==true)
      {
      digitalWrite(5,LOW);
      att=false;
      myGLCD.setFont(BigFont);
      myGLCD.setColor(VGA_WHITE);
      myGLCD.setBackColor(0,0,136);
      myGLCD.print("PRE",6,114);
      }
      else
       {
```

```
            digitalWrite(5,HIGH);
            att=true;
           myGLCD.setFont(BigFont);
           myGLCD.setColor(VGA_WHITE);
           myGLCD.setBackColor(VGA_RED);
           myGLCD.print("PRE",6,114);
            }
         }
        if (pressed_button==C_button) //tune
         {
          if (tune==false)
           {
            tune = true;
            cwmode = true;
            digitalWrite(2,HIGH);
            digitalWrite(0,HIGH);
           }
           else
           {
            tune = false;
            cwmode = false;
            digitalWrite(2,LOW);
            digitalWrite(0,LOW);
           }
         }
        if (pressed_button==D_button) //FILT
         {
          if (filt == false)
          {
           digitalWrite(6,HIGH);
           digitalWrite(7,LOW);
           filt = true;
          }
          else
          {
           digitalWrite(6,LOW);
           digitalWrite(7,HIGH);
           filt = false;
          }

         }
      break;

    } //end of switch
   }
  //S meter
   s_bar(analogRead(A0)/25);

  }
}

void band_switch()
{
digitalWrite(8,LOW);digitalWrite(9,LOW);digitalWrite(10,LOW);digitalWrite(11,LOW);digitalWrite(12,LOW);digital
Write(13,LOW);
switch (band_index)
 {
 case 0: digitalWrite(12,HIGH);break;
 case 1: digitalWrite(10,HIGH);break;
 case 2: digitalWrite(9,HIGH);break;
 case 3: digitalWrite(13,HIGH);break;
 case 4: digitalWrite(11,HIGH);break;
 case 5: digitalWrite(11,HIGH);break;
```

```
 case 6: digitalWrite(8,HIGH);break;
 case 7: digitalWrite(8,HIGH);break;
 }
}

void freq_disp_A()
{
int vfoA_khz = 0;
//write band char
 myGLCD.setFont(numeric_xl);
 myGLCD.setColor(VGA_WHITE);
 myGLCD.setBackColor(0,0,80);
 if (band_index<2)
  {
   if (disp_first==true)
   {
      myGLCD.setColor(0,0,80);
      myGLCD.fillRoundRect2(5,5,314,79); //A vfo window
      myGLCD.setColor(VGA_WHITE);
      myGLCD.fillRoundRect(69,65,77,73);
      myGLCD.fillRoundRect(200,65,208,73);
      myGLCD.printNumI(band_data[band_index],28,10,1);
      disp_first=false;
   }
   vfoA_buff = vfo-band_data[band_index]*1000000;
   vfoA_khz = vfoA_buff / 1000;
   if (vfoA_khz!=vfoA_khz_old)
   {
    myGLCD.printNumI(vfoA_khz,79,10,3,'0');
    vfoA_khz_old = vfoA_khz;
   }
   myGLCD.printNumI((vfoA_buff % 1000)/10,210,10,2,'0');
  }
  else
  {
   if (disp_first==true)
   {
     myGLCD.setColor(0,0,80);
     myGLCD.fillRoundRect2(5,5,314,79); //A vfo window
     myGLCD.setColor(VGA_WHITE);
     myGLCD.fillRoundRect(89,65,97,73);
     myGLCD.fillRoundRect(220,65,228,73);
     myGLCD.printNumI(band_data[band_index],8,10,2);
     disp_first=false;
   }
   vfoA_buff = vfo-band_data[band_index]*1000000;
   vfoA_khz = vfoA_buff / 1000;
   if (vfoA_khz!=vfoA_khz_old)
   {
    myGLCD.printNumI(vfoA_khz,99,10,3,'0');
    vfoA_khz_old = vfoA_khz;
   }
   myGLCD.printNumI((vfoA_buff % 1000)/10,230,10,2,'0');
  }
}

void freq_disp_B()
{
int vfoB_khz = 0;
//write band char
 myGLCD.setFont(numeric_m);
 myGLCD.setColor(VGA_WHITE);
 myGLCD.setBackColor(0,0,80);
```

```
 if (band_index<2)
  {
   if (disp_first==true)
   {
      myGLCD.setColor(0,0,80);
      myGLCD.fillRoundRect2(128,85,314,135); //A vfo window
      myGLCD.setColor(VGA_WHITE);
      myGLCD.fillRoundRect(169,125,174,130);
      myGLCD.fillRoundRect(250,125,255,130);
      myGLCD.printNumI(band_data[band_index],143,90,1);
      disp_first=false;
   }
   vfoB_buff = vfo-band_data[band_index]*1000000;
   vfoB_khz = vfoB_buff / 1000;
   if (vfoB_khz!=vfoB_khz_old)
   {
    myGLCD.printNumI(vfoB_khz,176,90,3,'0');
    vfoB_khz_old = vfoB_khz;
   }
   myGLCD.printNumI((vfoB_buff % 1000)/10,257,90,2,'0');
  }
  else
  {
   if (disp_first==true)
   {
      myGLCD.setColor(0,0,80);
      myGLCD.fillRoundRect2(128,85,314,135); //A vfo window
      myGLCD.setColor(VGA_WHITE);
      myGLCD.fillRoundRect(179,125,184,130);
      myGLCD.fillRoundRect(258,125,263,130);
      myGLCD.printNumI(band_data[band_index],131,90,1);
      disp_first=false;
   }
   vfoB_buff = vfo-band_data[band_index]*1000000;
   vfoB_khz = vfoB_buff / 1000;
   if (vfoB_khz!=vfoB_khz_old)
   {
    myGLCD.printNumI(vfoB_khz,186,90,3,'0');
    vfoB_khz_old = vfoB_khz;
   }
   myGLCD.printNumI((vfoB_buff % 1000)/10,265,90,2,'0');
  }
}

void s_bar(int value)
{
 if (value > s_value_old)
  {
   for (int j = s_value_old;j < value;j++)
   {
     if (j<25)
      myGLCD.setColor(VGA_WHITE);
       else
        myGLCD.setColor(VGA_RED);
     myGLCD.fillRect(38+j*6,158,41+j*6,166);
   }
   s_value_old = value;
  }
  else
  {
   myGLCD.setColor(0,0,80);
   myGLCD.fillRect(38+value*6,158,41+s_value_old*6,166);
   s_value_old = value;
```

```
  }
}

void s_bar_print()
{
  const char* s_mero[] = {"1","3","5","7","9","+20","+40","+60db"};
  const char* p_mero[] = {" 0","0.2","0.5"," 1"," 2","5W"};



   myGLCD.setFont(BigFont);
   myGLCD.setColor(VGA_WHITE);
   myGLCD.setBackColor(VGA_TRANSPARENT);
   myGLCD.print("S",15,144);
   myGLCD.print("Po",4,163);
   myGLCD.setFont(SmallFont);
   myGLCD.drawLine(37,155,181,155);
   myGLCD.setColor(150,200,255);
   myGLCD.drawLine(183,155,282,155);
   myGLCD.setColor(VGA_WHITE);
   myGLCD.drawLine(37,169,282,169);

   for (int i=0;i<16;i++)
   {
    if (i<10)
      myGLCD.setColor(VGA_WHITE);
      else
       myGLCD.setColor(150,200,255);
   if (i%2==0)
   {
      myGLCD.drawLine(39+i*16,154,39+i*16,152);
      myGLCD.drawLine(40+i*16,154,40+i*16,152);
   }
    else
    {
      myGLCD.drawLine(39+i*16,154,39+i*16,150);
      myGLCD.drawLine(40+i*16,154,40+i*16,150);
      if (i<10)
       myGLCD.print(s_mero[i/2],37+i*16,138);
       else
        myGLCD.print(s_mero[i/2],30+i*16,138);

    }
   if (i%3==0)
     {
      myGLCD.setColor(VGA_WHITE);
      myGLCD.drawLine(39+i*16,170,39+i*16,174);
      myGLCD.drawLine(40+i*16,170,40+i*16,174);
      myGLCD.print(p_mero[i/3],29+i*16,176);
     }

   }
}
```