B40all

```
/*
 Shuttle Tuned VFO for BITX-40 Raduino with speed control CW + SK
 Don Cantrell,ND6T  v 1.5.1   2 September 2017
 Compiles under Etherkit Si5351 library v 2.0.6
 This source file is under General Public License version 3.0
 BITX hardware modifications on http://bitxhacks.blogspot.com  25,Feb,2017
 Added RF power monitor 29,March,2017
 S meter to A1, Reverse power to A2, Forward power to A3, Keying speed to A6
 */
#include <si5351.h>
Si5351 si5351;
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,10,11,12,13);

  float QSK=1.5;  //Delay (in seconds)for semi-QSK
  int offset=700; // CW offset
  int wpm ;       // CW speed
  int p ;         //Timing period (milliseconds) for keyer function
  int tune;       //Tuning knob position
  float sm;       //"S" meter value
  float F = 0;    //Forward RF output (PEP watts RMS)
  float R = 0;    //Reverse RF output (PEP watts RMS)
  long count = 0; // Timeout counter
  unsigned long post;      // Time post
  float BFO = 11.999038e6; //My I.F. frequency
  float LO = BFO -7.2e6;   //I.F. minus starting frequency

void setup() {
    lcd.begin(16, 2);

    si5351.init(SI5351_CRYSTAL_LOAD_8PF,24999020L,0); //My actual ref osc freq.
    si5351.set_pll(SI5351_PLL_FIXED, SI5351_PLLA);
    si5351.set_freq(LO * 100, SI5351_CLK2);//Program the synthesizer

    pinMode(4, INPUT_PULLUP); //Dash input on Plug 3 pin 4
    pinMode(5, INPUT_PULLUP); //Dot input on Plug 3 pin 3
    pinMode(6, OUTPUT);       //Sidetone from Plug 3 pin 2
    pinMode(7, OUTPUT);       //T/R keying for CW Plug 3 pin 1

    lcd.setCursor(0,0); ///////////Splash/////////////
    lcd.print("B40all");
    lcd.setCursor(0,1);
    lcd.print("ver. 1.5.1");
    delay(2000);
    post=millis();    //Set starting time post
}

void loop() {
  //Read peak Forward RF power
   if(analogRead(A3)/(3e4/analogRead(A3)+1)>F)F=(analogRead(A3)/
(3e4/analogRead(A3)+1));
  //Read peak Reverse RF power
   if(analogRead(A2)/(3e4/analogRead(A2)+1>R))R=(analogRead(A2)/
(3e4/analogRead(A2)+1));

  if(((digitalRead(4)==LOW)&&(analogRead(A6)>=300)) || (digitalRead(5)==LOW))
{CW();}//Is the key active?
  digitalWrite(7, LOW); // Restore T/R relays from CW mode
  tune = analogRead(A7);// Read the tuning input on analog pin 7:

    if (tune>560)up();   //Establish tuning direction
    if (tune<464)down();

  sm=analogRead(A1);// Read signal level

  if (millis() < 100){  //Prevent blank display on initialization.
      show();
      si5351.set_freq(LO * 100, SI5351_CLK2);//Program the synthesizer
      }
  if(tune<464||tune>560){//If tuning then display
      show();
      delay(300); //To ease tuning
      }
  if ((millis() - post)> 2000){  //If idle, display each 2 sec
    show();
    F=0;                //Clear power readings
    R=0;
    post=millis();
  }
}

//****Functions****

void show() {  //Display function
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print ((BFO-LO)/1e3,3);//Calculate & show frequency
   if(tune>560)lcd.print(" >");
   if(tune<464)lcd.print(" <");
       if (tune>464 && tune<560)lcd.print(" I"); //Idle indicator
  lcd.print(" KHz");
  lcd.setCursor(14,1);
  lcd.print((analogRead(A6)/30));//Display keying speed
    if(analogRead(A6)<300){      //or Straight Key mode
      lcd.setCursor(14,1);
      lcd.print("SK");}
  lcd.setCursor(0,1);
    if(F>=1){  //If RF power present replace S meter with power.
      lcd.print(F,0);
  lcd.print("W SWR=");
    lcd.print((1+sqrt(R/F))/(1-sqrt(R/F)),1);
    lcd.print(":1");
    lcd.setCursor(14,1);
    }
    else{     //Otherwise display S meter
    if (sm>=110){lcd.print("S9+20");}
    if (sm>=80&&sm<110){lcd.print("S9+10");}
    if (sm>=45&&sm<80){lcd.print("S9");}
    if (sm<45){lcd.print("S");
    lcd.print(sm/5,0);
    }
   }
 }
```

```
void down() {
  if (LO >= BFO-7e6)LO = BFO-7e6;        //Lower frequency limit
  LO = LO + (pow((464 - tune)/5,3)/100); //Increase local osc frequency (decreases
T/R freq)
  si5351.set_freq(LO * 100, SI5351_CLK2);//Program the synthesizer
}

void up() {
  if (LO <= BFO-7.3e6)LO = BFO-7.3e6;    //Upper frequency limit
  LO = LO - (pow((tune - 560)/5,3)/100); //Decrease local osc frequency (increases
T/R freq)
  si5351.set_freq(LO * 100, SI5351_CLK2);//Program the synthesizer
}

void CW(){ //CW modes
  digitalWrite(7,HIGH); // Key T/R relays and do the setup while they activate
  wpm = analogRead(A6)/30; //Read CW speed pot and set WPM rate
  p = 1200/wpm;   // convert speed to milliseconds

  if (wpm < 10)sk();// Read speed control to switch to Straight Key mode
  if (wpm < 10)return; //To hasten the return to SSB

  //Iambic keyer
  while (count < (QSK*6e4)) { // Delay time after last action to return to normal
SSB
    if(digitalRead(4)==LOW)dah();
    if(digitalRead(5)==LOW)dit();
    count++;} //Increment time-out for CW routine
    count=0;  // Reset the CW timeout
}

void dit() {   //Send a dot and an element space
   si5351.set_freq(((BFO-LO)-offset) * 100 , SI5351_CLK1); //Key on CW transmit
frequency
    tone(6,offset); //Sidetone
    delay(p);
    noTone(6);
  si5351.output_enable(SI5351_CLK1, 0);    // Unkey transmit
    delay(p);
  count=0;  //Reset counter
}

void dah() {  //Send a dash and an element space
   si5351.set_freq(((BFO-LO)-offset) * 100 , SI5351_CLK1); //Key on CW transmit
frequency
    tone(6,offset); //Sidetone
    delay(3*p);
    noTone(6);
    si5351.output_enable(SI5351_CLK1, 0); // Unkey transmit
    delay(p);
   count=0;  //Reset counter
}
```

```
void sk() {  //Straight Key mode
  while (count < 2000) { // Delay time after last action to return to normal SSB
   if(digitalRead(5)==LOW)post=millis();     //Set post for display timing
     while(digitalRead(5)==LOW){
      si5351.set_freq(((BFO-LO)-offset) * 100 , SI5351_CLK1); //Key down
      tone(6,offset); //Sidetone
    if(millis()-post>500){ //If keyed for more than half second, read power
     F=analogRead(A3)/(3e4/analogRead(A3)+1); //Read Forward RF power
     R=analogRead(A2)/(3e4/analogRead(A2)+1); //Read Reverse RF power
    show();
    delay(100);
    }
    count=0; //Reset counter
  }
    si5351.output_enable(SI5351_CLK1, 0); // Unkey transmit
     noTone(6);
    count++;
  }
  count=0; //Reset counter
}
```