BITX60ST

```
/*
 BITX program for 60 M (USA allocation) with Shuttle Tuning
 V 1.2   Don Cantrell, ND6T  5 May 2017
 Compiles under etherkit Si5351 library v 2.0.1
 This source file is under General Public License version 3.0
 Reduced unnecessary updating
 Simplified sketch. No metering.

 */
#include <si5351.h>
Si5351 si5351;
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,10,11,12,13);

  int tune; //Tuning knob position
  unsigned long post;//Timing milepost
  int channel = 1; //Channel number
  long BFO = 11999045;    //My BFO frequency (11999045)
  long LO = BFO + 5330500;  //Local Oscillator for Upper sideband,CH.1
  long frequency;

void setup() {

    lcd.begin(16, 2);
    si5351.init(SI5351_CRYSTAL_LOAD_8PF,25004920,0); //My actual ref osc freq.
    si5351.set_pll(SI5351_PLL_FIXED, SI5351_PLLA);
    pinMode(7, OUTPUT); // If you have CW mod
 }

void loop() {

  digitalWrite(7,LOW); //If you have CW mod

  tune = analogRead(A7);// Read the input on analog pin 7:

//Set switching at knob limits and increment channel selection
    if (tune>1000) {
      ++channel;
      post=millis(); //Stake a time post for channel update limit
    }

    if (channel > 5)channel = 1;
    if (tune < 20)channel = 0;

 switch (channel) {
    case 0:
      ShuttleTuning();
        break;
    case 1:
      frequency = 5330500;
         break;
    case 2:
      frequency = 5346500;
         break;
      case 3:
      frequency = 5357000;
         break;
      case 4:
      frequency = 5371500;
          break;
      case 5:
      frequency = 5403500;
         break;
     }

 if(millis()-post<100){     // Update 5351 only if under 100 ms
  LO = BFO + frequency;
  si5351.set_freq(LO * 100, SI5351_CLK2);  //Program the synthesizer
  show();
  delay(1000); // Slow down tuning
  }
}

//****Functions****

void show() {  //Display function
 lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("#");
  lcd.print(channel);
  lcd.print("=");
  lcd.print ((LO-BFO)/1e6,6);//Calculate & show frequency
  lcd.print(" MHz");
    if (tune>560)lcd.write(">");
    if (tune<464)lcd.write("<");
    if (tune>464 && tune<560)lcd.print("I"); //Idle indicator


}

 void ShuttleTuning() {

 while(tune<1000) {
    tune = analogRead(A7);// Read the input on analog pin 7
    if (tune>560)up();   //Establish tuning direction
    if (tune<464)down();     //and idle zone
    if(millis()-post<2000)show();//Display then freeze
    delay(500);// Slow to ease tuning
    }
 }

void up() {
  LO = LO + round(pow((tune-560)/5,3)/100); //Increase local osc frequency
  si5351.set_freq(LO * 100, SI5351_CLK2);//Program the synthesizer
  post=millis();
  }

void down() {
  LO = LO - round(pow((464-tune)/5,3)/100); //Decrease local osc frequency
  si5351.set_freq(LO * 100, SI5351_CLK2);//Program the synthesizer
  post=millis();
  }
```