```
/**
 * Simple rotary encoder tuned for Raduino
 * V 1.1.3 ND6T  4 October 2017
 * Compiles under Etherkit Si5351 library v 2.0.6
 * This source file is under General Public License version 3.0
 *  Pin Connections:  *
 *      Encoder A = D2
 *      Encoder B = D3
 *      Encoder switch = D4
 */
#include <Rotary.h> //Available at:https://github.com/brianlow/Rotary
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,10,11,12,13);
#include <si5351.h>
Si5351 si5351;

  Rotary r = Rotary(2,3); //Encoder to pins 2,3
  byte result;
  int ind;             //Tuning position indicator
  long incr = 1000;    //Initial tuning increment
  long BFO= 11999855;  //Actual measured frequency
  long LO = BFO -7.2e6;//Initial frequency
  long oldLO;          //Old LO change reference
  float FQ;            //Operating frequency
  long long post;      //Time post for sensitive delays

void setup(){
  PCICR |= (1 << PCIE2);//Interrupt setup
  PCMSK2 |= (1 << PCINT18) | (1 << PCINT19);//Matrix "state machine" decode
  r.begin();   //Users that downloaded Rotary library before Dec.2018 should delete this line

  lcd.begin(16, 2);
  lcd.clear();

  si5351.init(SI5351_CRYSTAL_LOAD_8PF,25005200L,0); //Ref osc freq.
  si5351.set_pll(SI5351_PLL_FIXED, SI5351_PLLA);
  si5351.set_freq(LO * 100, SI5351_CLK2);   //Program the synthesizer LO

  pinMode(4,INPUT_PULLUP);//Tuning increment switch

  lcd.setCursor(0,0); ///////////Splash/////////////
  lcd.print("EZRaduino");
  lcd.setCursor(0,1);
  lcd.print("ver. 1.1.3");
  delay(2000);
  post=millis();
  }

void loop(){
 ind=(int(log10(FQ)))-(int(log10(incr)))+1;//Calculate indicator position
    if(incr>100)ind-=1;       //Compensate for decimal place
 if(millis()-post<50) show(); //Display if changed less than 50 ms ago

 if(LO!=oldLO){ //If frequency changed then reprogram
  oldLO=LO;     //Reset reference
  program();    //Re-program the LO
  }
}
```

```
//*****************FUNCTIONS (subroutines)***********************************
void show(){          //Display routine
  FQ=BFO-LO;
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(FQ/1000,3);//Parse the display for easy reading
  lcd.print(" KHz");
  lcd.setCursor(ind,0);//Indicator position
  lcd.cursor();        //Tuning increment indicator
  delay(100);          //To prevent flicker
}

void program(){
 si5351.set_freq(LO * 100, SI5351_CLK2);//Program the synthesizer
 post=millis();                         //Return and display
}

ISR(PCINT2_vect){      //Interrupt service routine
  result = r.process();
  if(digitalRead(4)==HIGH){       //If tuning knob is not pressed
   if(result == DIR_CW)LO-=incr;  //Clockwise subtract it.
   if(result == DIR_CCW) LO+=incr;//CounterClockwise. Add the increment
   post=millis();                 //Return and display
   }
  else{            //If the tuning knob is pressed then move the cursor
    if(result == DIR_CW){ //Move cursor right
      incr=incr/10;
        if(incr<1)incr=1; //Lower limit
    }
    if(result == DIR_CCW){//Move cursor left
      incr=incr*10;
        if((log10(incr))>((log10(FQ))-1))incr=incr/10;//Upper limit
        }
    }
  post=millis(); //Return and display
}
```