

```

/**
 * Simple rotary encoder tuned, I2C SSD1306 OLED, Si5351
 * 4 KHz to 100 MHz
 * V 1.0.1 ND6T 23 January 2020
 * Corrections tnx to WA1EDJ
 * This source file is under General Public License version 3.0
 *
 * Pin Connections:
 *   Si5351 & Display: SDA = A4, SCL = A5
 *   Encoder: A = D2, B = D3, switch = D4
 */
#include <Rotary.h> //Available at: https://github.com/brianlow/Rotary
#include <Adafruit_SSD1306.h>
#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);
#include <si5351.h> //Etherkit Si5351 library v 2.0.6
(https://github.com/etherkit/Si5351Arduino)
Si5351 si5351;

Rotary r = Rotary(2,3);//Encoder to pins D2,D3
unsigned char result; //Results
int ind; //Tuning position indicator
int oldind; //Previous tuning indicator
int pos=2; //Cursor position
long incr = 1000; //Initial tuning increment
long upperLimit =15e7; //Upper frequency limit
long oldFQ; //Frequency change reference
long long FQ = 1e7; //Starting frequency

#define sw 4 //Tuning increment switch

void setup(){
  display.begin(0x3C); // initialize with the OLED I2C addr
  r.begin(); //Users that downloaded Rotary library before Dec.2018 should delete this line

  si5351.init(SI5351_CRYSTAL_LOAD_8PF,24999908L,0); //Ref osc freq.
  si5351.set_pll(SI5351_PLL_FIXED, SI5351_PLLA);
  si5351.set_freq(FQ * 100, SI5351_CLK0); //Program the synthesizer

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

  ////////////Splash//////////
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.print("EZgen II Ver 1.0.1");
  display.display();
  delay(3000);
}

void loop(){
  tuning();

  if(FQ!=oldFQ){ //If frequency changed then reprogram
    oldFQ=FQ; //Reset reference
    program(); //Re-program the PLL
  }
}

```

```

//*****FUNCTIONS (subroutines)*****
void show(){ //Show 'em what you have!
    long fq=FQ; //Re-cast to make it printable
    int mhz=(FQ/1e6); //Truncate MHz
    long hz=FQ-(mhz*1000000);
    if(hz==(-1)){hz=999999; //Math patch
    mhz=mhz-1;}
    if(hz==(-2)){hz=999998; // "
    mhz=mhz-1;}
    if(hz==(-3)){hz=999997; // "
    mhz=mhz-1;}
    if(hz==(-4)){hz=999996; // "
    mhz=mhz-1;}
    display.clearDisplay();
    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    if(mhz<10) display.print("0");

    if(FQ>=1e8)display.print(fq); //If over 100MHz shorten display to fit
    else{
        display.print(mhz); //Parse at each 3 digits for easy reading
        if(FQ<1e8)display.print(" ");
        if((hz/1e3)<10) display.print("0");//Insert leading zeros if necessary
        if((hz/1e3)<100) display.print("0");
        display.print((hz)/1e3,3); //Parse final 6 digits
    }
    display.setTextSize(2);
    display.setCursor((ind-1)*12,9);
    display.print("-");
    display.display();
}

void tuning(){
    result = r.process();
    if(digitalRead(sw)==HIGH){ //If tuning knob is not pressed
        if(result == DIR_CW){
            FQ+=incr; //Clockwise. Add the increment
            if(FQ>upperLimit)FQ=upperLimit;//Unless it exceeds upper limit
        }
        if(result == DIR_CCW){ //CounterClockwise subtract
            if(((FQ)-incr)>=4000)FQ-=incr;//But keep it above 4 KHz
        }
    }
    else{ //If the tuning knob is pressed then move the cursor
        if(result == DIR_CW){ //Move cursor right
            if(pos<8)++pos;
        }
        if(result == DIR_CCW){ //Move cursor left
            if(pos>0)--pos;
        }
    }
    ind=pos; //Correlate indicator to position

    if(FQ<1e8){ //If under 100 MHz
        if(ind>2)ind=pos+1; //Compensate for decimal place
        if(ind>6)ind=pos+2;

        if(ind>9)incr=1; //Set increment by cursor position.
        if(ind==9)incr=10;
        if(ind==8)incr=100;
        if(ind==6)incr=1e3;
        if(ind==5)incr=1e4;
        if(ind==4)incr=1e5;
        if(ind==2)incr=1e6;
        if(ind==1)incr=1e7;
    }
}

```

```
if(ind==0)incr=1e8;
}
else{
if(ind==9)incr=1; //If over 100 MHz
if(ind==8)incr=10; //Set increment by cursor position.
if(ind==7)incr=100;
if(ind==6)incr=1e3;
if(ind==5)incr=1e4;
if(ind==4)incr=1e5;
if(ind==3)incr=1e6;
if(ind==2)incr=1e7;
if(ind==1)incr=1e8;
}

if(oldind!=ind)show(); //Display if cursor change
oldind=ind;
}

void program(){
si5351.set_freq(FQ * 100, SI5351_CLK0);//Program the synthesizer
show();
}
```