

```

/*
 * This example shows how to read from a seesaw encoder module.
 * The available encoder API is:
 *   int32_t getEncoderPosition();
 *   int32_t getEncoderDelta();
 *   void enableEncoderInterrupt();
 *   void disableEncoderInterrupt();
 *   void setEncoderPosition(int32_t pos);
 */
#include "Adafruit_seesaw.h"
#include <seesaw_neopixel.h>

#define SS_SWITCH    24
#define SS_NEOPIX    6

#define SEESAW_ADDR    0x36

Adafruit_seesaw ss;
seesaw_NeoPixel sspixel = seesaw_NeoPixel(1, SS_NEOPIX, NEO_GRB + NEO_KHZ800);

int32_t encoder_position;

void setup() {
  Serial.begin(115200);
  while (!Serial) delay(10);

  Serial.println("Looking for seesaw!");

  if (! ss.begin(SEESAW_ADDR) || ! sspixel.begin(SEESAW_ADDR)) {
    Serial.println("Couldn't find seesaw on default address");
    while(1) delay(10);
  }
  Serial.println("seesaw started");

  uint32_t version = ((ss.getVersion() >> 16) & 0xFFFF);
  if (version != 4991){
    Serial.print("Wrong firmware loaded? ");
    Serial.println(version);
    while(1) delay(10);
  }
  Serial.println("Found Product 4991");

  // set not so bright!
  sspixel.setBrightness(20);

```

```

sspixel.show();

// use a pin for the built in encoder switch
ss.pinMode(SS_SWITCH, INPUT_PULLUP);

// get starting position
encoder_position = ss.getEncoderPosition();

Serial.println("Turning on interrupts");
delay(10);
ss.setGPIOInterrupts((uint32_t)1 << SS_SWITCH, 1);
ss.enableEncoderInterrupt();
}

void loop() {
  if (! ss.digitalRead(SS_SWITCH)) {
    Serial.println("Button pressed!");
  }

  int32_t new_position = ss.getEncoderPosition();
  // did we move around?
  if (encoder_position != new_position) {
    Serial.println(new_position);    // display new position

    // change the neopixel color
    sspixel.setPixelColor(0, Wheel(new_position & 0xFF));
    sspixel.show();
    encoder_position = new_position;    // and save for next round
  }

  // don't overwhelm serial port
  delay(10);
}

uint32_t Wheel(byte WheelPos) {
  WheelPos = 255 - WheelPos;
  if (WheelPos < 85) {
    return sspixel.Color(255 - WheelPos * 3, 0, WheelPos * 3);
  }
  if (WheelPos < 170) {
    WheelPos -= 85;
    return sspixel.Color(0, WheelPos * 3, 255 - WheelPos * 3);
  }
}

```

```
WheelPos -= 170;  
return sspixel.Color(WheelPos * 3, 255 - WheelPos * 3, 3);  
}
```