Final code

```
//NEOPIXELCODE
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
 #include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif

// Which pin on the Arduino is connected to the NeoPixels?
#define PIN        6 // On Trinket or Gemma, suggest changing this to 1

// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS 1 // Popular NeoPixel ring size

// When setting up the NeoPixel library, we tell it how many pixels,
// and which pin to use to send signals. Note that for older NeoPixel
// strips you might need to change the third parameter -- see the
// strandtest example for more information on possible values.
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_RGB + NEO_KHZ800);

#define DELAYVAL 500 // Time (in milliseconds) to pause between pixels


//FINALCODE

#include <Arduino.h>
#include <U8g2lib.h>

#ifdef U8X8_HAVE_HW_SPI
#include <SPI.h>
#endif
#ifdef U8X8_HAVE_HW_I2C
#include <Wire.h>
#endif



U8G2_SSD1327_EA_W128128_1_HW_I2C u8g2(U8G2_R0, /* reset=*/ U8X8_PIN_NONE); /*
Uno: A4=SDA, A5=SCL, add "u8g2.setBusClock(400000);" into setup() for speedup if possible
*/
```

```
// End of constructor list


// This example shows a scrolling text.
// If U8G2_16BIT is not set (default), then the pixel width of the text must be lesser than 128
// If U8G2_16BIT is set, then the pixel width an be up to 32000



//Click here to get the library: http://librarymanager/All#SparkFun_VCNL4040
#include "SparkFun_VCNL4040_Arduino_Library.h"
#define PIN_ANALOG_IN A0
VCNL4040 proximitySensor;

long startingProxValue = 0;
long deltaNeeded = 0;
boolean nothingThere = false;

u8g2_uint_t offset;     // current offset for the scrolling text
u8g2_uint_t width;      // pixel width of the scrolling text (must be lesser than 128 unless
U8G2_16BIT is defined
const char *text = "XXXXXX";  // scroll this text from right to left







void setup(void) {

//NEOPIXEL
 // These lines are specifically to support the Adafruit Trinket 5V 16 MHz.
  // Any other board, you can remove this part (but no harm leaving it):
#if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)
  clock_prescale_set(clock_div_1);
#endif
  // END of Trinket-specific code.

  pixels.begin(); // INITIALIZE NeoPixel strip object (REQUIRED)
```

```
/* U8g2 Project: SSD1306 Test Board */
//pinMode(10, OUTPUT);
//pinMode(9, OUTPUT);
//digitalWrite(10, 0);
//digitalWrite(9, 0);

/* U8g2 Project: T6963 Test Board */
//pinMode(18, OUTPUT);
//digitalWrite(18, 1);

/* U8g2 Project: KS0108 Test Board */
//pinMode(16, OUTPUT);
//digitalWrite(16, 0);

Serial.begin(9600);
Serial.println("SparkFun VCNL4040 Example");

Wire.begin(); //Join i2c bus

if (proximitySensor.begin() == false)
{
  Serial.println("Device not found. Please check wiring.");
  while (1); //Freeze!
}

//Set the current used to drive the IR LED - 50mA to 200mA is allowed.
proximitySensor.setLEDCurrent(200); //For this example, let's do max.

//The sensor will average readings together by default 8 times.
//Reduce this to one so we can take readings as fast as possible
proximitySensor.setProxIntegrationTime(8); //1 to 8 is valid

//Take 8 readings and average them
for (byte x = 0 ; x < 8 ; x++)
{
  startingProxValue += proximitySensor.getProximity();
}
startingProxValue /= 8;

deltaNeeded = (float)startingProxValue * 0.05; //Look for 5% change
if (deltaNeeded < 5) deltaNeeded = 5; //Set a minimum
```

```
  u8g2.begin();

  u8g2.setFont(u8g2_font_roentgen_nbp_h_all); // set the target font to calculate the pixel width
  //width = u8g2.getUTF8Width(text);    // calculate the pixel width of the text
width = 500;
 u8g2.setFontMode(0);    // enable transparent mode, which is faster
}


void loop(void) {

//NEO

pixels.clear(); // Set all pixel colors to 'off'






//OLED
  u8g2_uint_t x;
  u8g2.firstPage();
  unsigned int proxValue = proximitySensor.getProximity();

  int value = analogRead(PIN_ANALOG_IN);

  if (proxValue > (startingProxValue + deltaNeeded))
  {
    Serial.print("Something is there!");
    nothingThere = false;
  }
  else
  {
    if (nothingThere == false) Serial.print("I don't see anything");
    nothingThere = true;
  }

  if (nothingThere == false) {

    if (value <= 40) {
      Serial.println("Quiet.");
      //      text = "ON";
      text ="BREATHE";
      pixels.setPixelColor(0, pixels.Color(150, 0, 0));



    //delay(DELAYVAL); // Pause before next pass through loop
    }
    else if ( (value > 40) && ( value <= 60) )
    {
      Serial.println("Moderate.");
        text ="RELAX";
        pixels.setPixelColor(0, pixels.Color(0, 150, 0));



    //delay(DELAYVAL); // Pause before next pass through loop
    }
    else if (value > 60)
    {
      Serial.println("Loud.");
      text ="STAY CALM";
      pixels.setPixelColor(0, pixels.Color(0, 0, 150));



    //delay(DELAYVAL); // Pause before next pass through loop
    }

pixels.show();

    do {

      // draw the scrolling text at current offset
      x = offset;
      u8g2.setFont(u8g2_font_roentgen_nbp_h_all);    // set the target font
      do {                // repeated drawing of the scrolling text...
        u8g2.drawUTF8(x, 30, text);     // draw the scolling text
        x += width;           // add the pixel width of the scrolling text
      } while ( x < u8g2.getDisplayWidth() );   // draw again until the complete display is filled

      u8g2.setFont(u8g2_font_roentgen_nbp_h_all);   // draw the current pixel width
      u8g2.setCursor(0, 58);
      u8g2.print(width);          // this value must be lesser than 128 unless U8G2_16BIT is set

    } while ( u8g2.nextPage() );

    offset -= 50;             // scroll by one pixel
    if ( (u8g2_uint_t)offset < (u8g2_uint_t) - width )
      offset = 0;             // start over again

    delay(10);              // do some small delay
  } else if (nothingThere == true) {
    Serial.println("off");
    text = "";
    do {

      // draw the scrolling text at current offset
      x = offset;
      u8g2.setFont(u8g2_font_roentgen_nbp_h_all);    // set the target font
      do {                // repeated drawing of the scrolling text...
        u8g2.drawUTF8(x, 30, text);     // draw the scolling text
        x += width;           // add the pixel width of the scrolling text
      } while ( x < u8g2.getDisplayWidth() );   // draw again until the complete display is filled

      u8g2.setFont(u8g2_font_roentgen_nbp_h_all);   // draw the current pixel width
      u8g2.setCursor(0, 58);
      u8g2.print(width);          // this value must be lesser than 128 unless U8G2_16BIT is set

    } while ( u8g2.nextPage() );

    offset -= 8;            // scroll by one pixel
    if ( (u8g2_uint_t)offset < (u8g2_uint_t) - width )
      offset = 0;             // start over again

    delay(10);              // do some small delay
  }




}
```