Prácticas de Tecnologías Emergentes

Elena Cantero Molina Julio Jiménez Nájera

30 de diciembre de 2019

1. Práctica 1. Eye tracker: Sensor de posición de un ojo.

Resumen

En esta práctica realizamos un sensor ocular el cual detecta si el iris se encuentra a la derecha, izquierda o centro del ojo. La idea de la práctica ha sido facilitada por el proyecto realizado en el siguiente enlace.

1.1. Material utilizado

Para realizar este proyecto, hemos utilizado el kit de desarrollo con un ELEGOO Mega 2560, compatible con el Arduino Mega 2560, proporcionado por el centro, un portátil con la IDE de arduino y dos módulos infrarrojos Pololu QTR-1RC. Debido a la popularidad de los kits de desarrollo de Arduino y compatibles, me centraré en explicar el sensor de infrarrojos.

1.1.1. Pololu QTR-1RC

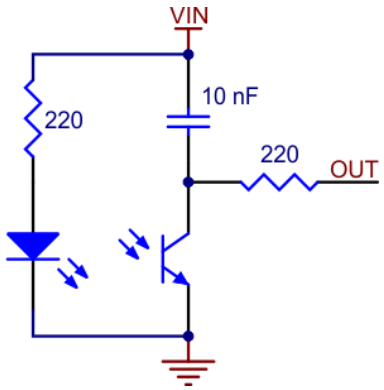


Figura 1: Esquemático del Pololu QTR-1RC

Este módulo permite emitir luz infrarroja a través de un diodo para recoger la cantidad de luz recibida en un fototransistor. Esto nos permite diferenciar si hay un objeto blanco o si hay un objeto oscuro.

1.2. Código utilizado

```
#include <QTRSensors.h>

QTRSensors qtr;

const uint8_t SensorCount = 2;
uint16_t sensorValues[SensorCount];

void setup()
{
  // configure the sensors
  qtr.setTypeAnalog();
  qtr.setSensorPins((const uint8_t[]){A0, A1},
     SensorCount);
  qtr.setEmitterPin(2);

  delay(500);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  digitalWrite(LED_BUILTIN, HIGH); // turn on Arduino'
     s LED to indicate we are in calibration mode

  // analogRead() takes about 0.1 ms on an AVR.
  // 0.1 ms per sensor * 4 samples per sensor read (
     default) * 2 sensors
  // * 10 reads per calibrate() call = ~8 ms per
     calibrate() call.
  // Call calibrate() 1250 times to make calibration
     take about 10 seconds.
  for (uint16_t i = 0; i < 1250; i++)
  {
    qtr.calibrate();
  }
```

```
29    digitalWrite(LED_BUILTIN, LOW); // turn off Arduino's LED to indicate we are through with calibration
30
31    // print the calibration minimum values measured when emitters were on
32    Serial.begin(9600);
33    for (uint8_t i = 0; i < SensorCount; i++)
34    {
35      Serial.print(qtr.calibrationOn.minimum[i]);
36      Serial.print(' ');
37    }
38    Serial.println();
39
40    // print the calibration maximum values measured when emitters were on
41    for (uint8_t i = 0; i < SensorCount; i++)
42    {
43      Serial.print(qtr.calibrationOn.maximum[i]);
44      Serial.print(' ');
45    }
46    Serial.println();
47    Serial.println();
48    delay(1000);
49  }
50
51  void loop()
52  {
53    // read calibrated sensor values and obtain a measure of the line position
54    // from 0 to 5000 (for a white line, use readLineWhite() instead)
55    uint16_t position = qtr.readLineBlack(sensorValues);
56
57    // print the sensor values as numbers from 0 to 1000, where 0 means maximum
58    // reflectance and 1000 means minimum reflectance, followed by the line
59    // position
60    for (uint8_t i = 0; i < SensorCount; i++)
61    {
62      Serial.print(sensorValues[i]);
```

```
63      Serial.print('\t');
64    }
65    Serial.print('\n');
66
67    if(sensorValues[0] > 600){
68      digitalWrite(3, HIGH);
69    }
70    else{
71      digitalWrite(3, LOW);
72    }
73      if(sensorValues[1] > 600){
74      digitalWrite(4, HIGH);
75    }
76    else{
77      digitalWrite(4, LOW);
78    }
79
80    delay(40);
81  }
```

Listing 1: Código utilizado

Se puede observar en el código de la placa, que se ha añadido la biblioteca QTRSensors de Pololu para controlar los módulos.
Se configura el sistema para recoger la entrada de dos sensores analógicos. Además, se calibra los sensores durante 10 realizando muestras blancas y oscuras. Esto permite reconocer la pupila del ojo. Se configura el puerto USB para transmitir los datos leidos de los sensores.
En el bucle del programa se va leyendo el valor de los sensores, se apaga el led corresponiente a cada sensor infrarrojo si el valor devuelto es menor a cierto valor umbral y envía los datos recogidos por el sensor al puerto USB.

1.3. Prueba realizada

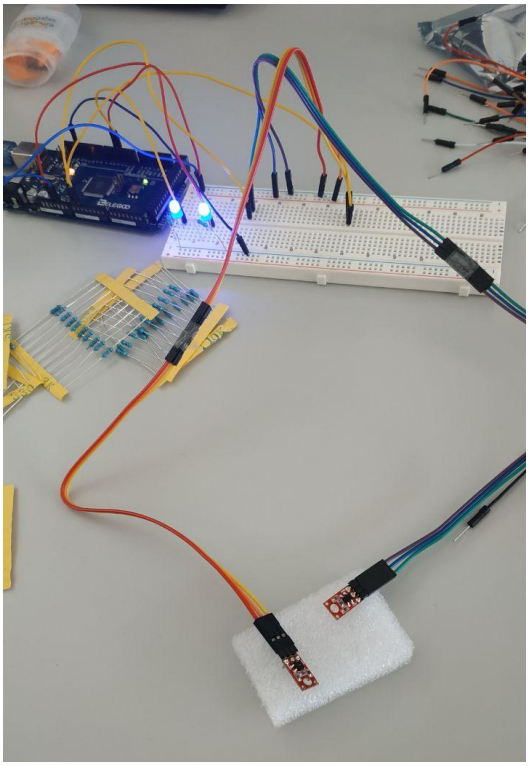


Figura 2: Prototipo montado

Se observa el primer prototipo de Eye Tracker, en él los sensores que irían a cada lado del ojo están situados sobre un trozo de plástico expandido para realizar con mayor facilidad las distintas pruebas. Cada sensor está conectado a la toma de 5 voltios de la placa Arduino Mega 2560, a la toma de tierra de la misma placa y a su pin analógico correspondiente para la lectura del sensor.

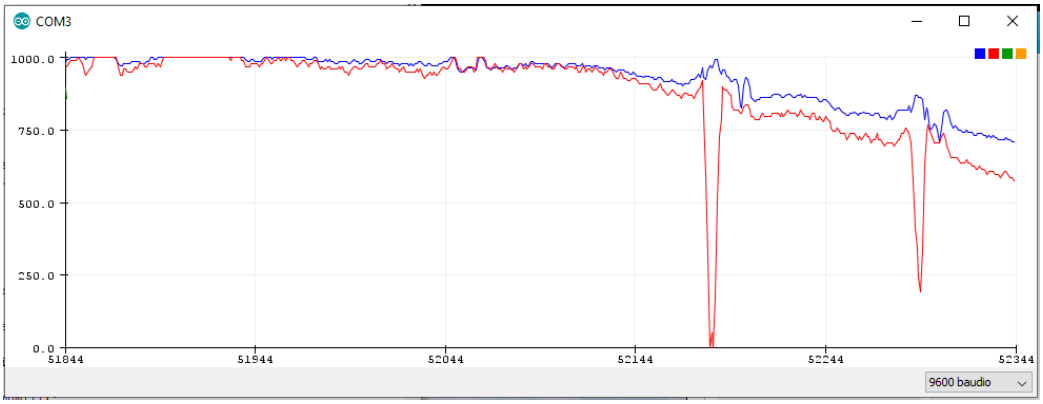


Figura 3: Captura de los sensores

La captura de los valores de los sensores se reciben correctamente en el ordenador y se muestran mediante el Serial Plotter del IDE de arduino. Las bajadas del valor de los sensores se deben al paso de un objeto por el sensor.

La continuación de este proyecto requeriría la correcta sujección de los sensores infrarrojos, lo cual nos es difícil de realizar debido a la falta de materiales como una montura de gafas o de alguna resina adhesiva. Esto es necesario para limitar las variaciones de las capturas de los sensores a las lecturas del ojo del usuario.

2. Práctica 2. XBee ZigBee

Resumen

En esta práctica la idea principal era hacer un prototipo de un detector de movimiento.

2.1. Material utilizado

Para realizar este proyecto, hemos utilizado el XBee ZigBee Mesh Kit, proporcionado por el centro.

Figura 4: XBee ZigBee

2.2. Configuración Módulos

En este apartado se va a mostrar cómo se han configurado los módulos para conectar los dos XBee entre sí. La primero que se muestra es el del coordinador.

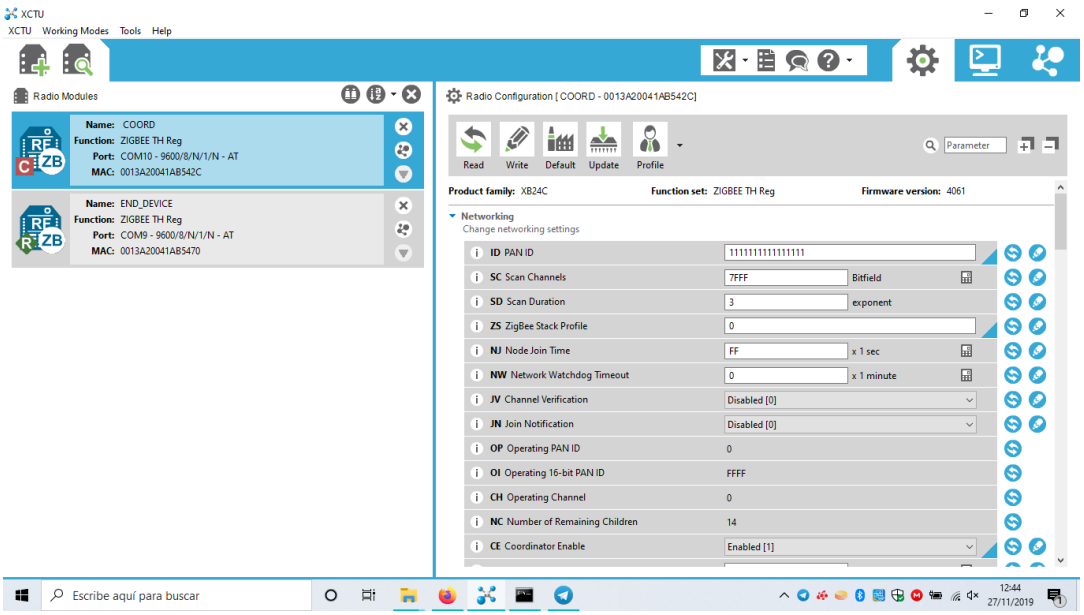


Figura 5: Coordinador

Y la siguiente es la del receptor.

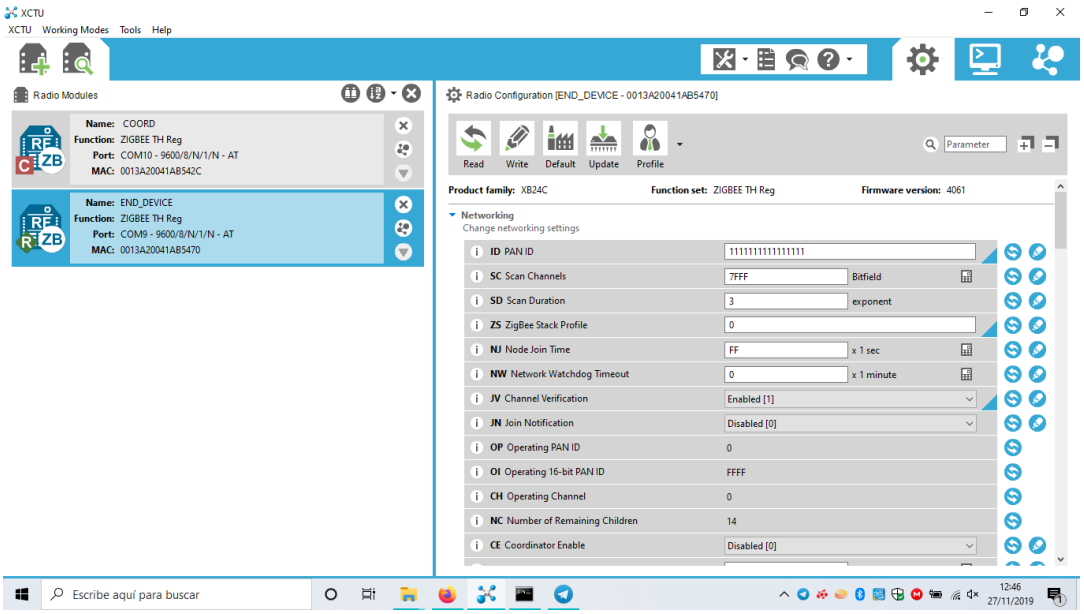


Figura 6: Receptor

Y por último se muestra cómo se buscan ambos módulos para establecer la conexión entre ellos.

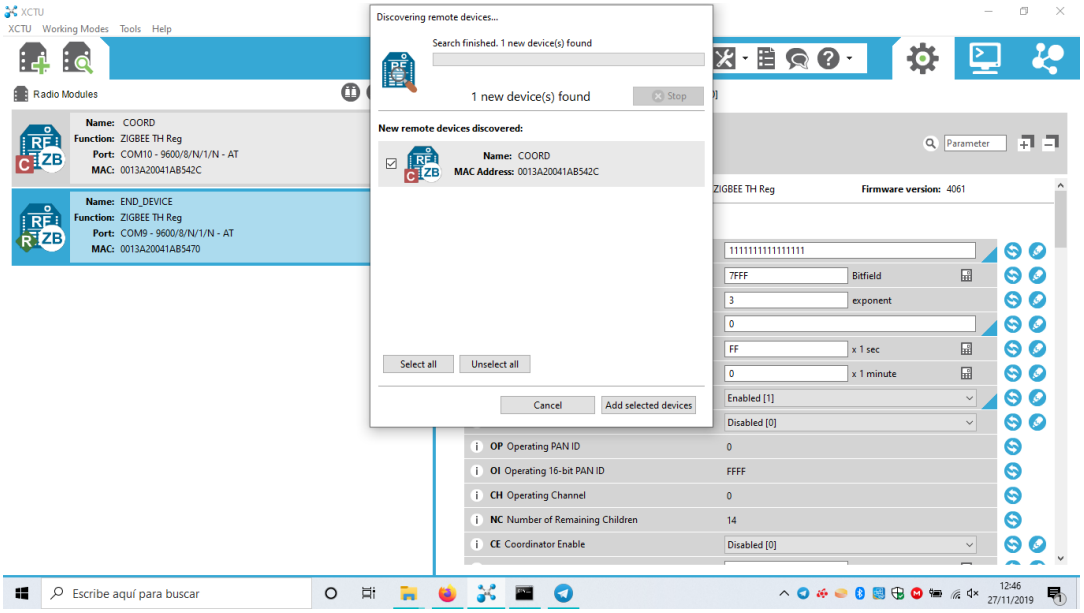


Figura 7: Conexión

2.3. Prueba realizada

Para comprobar que el receptor le envía datos al coordinador hemos usado la propia terminal que nos proporciona la XCTU.

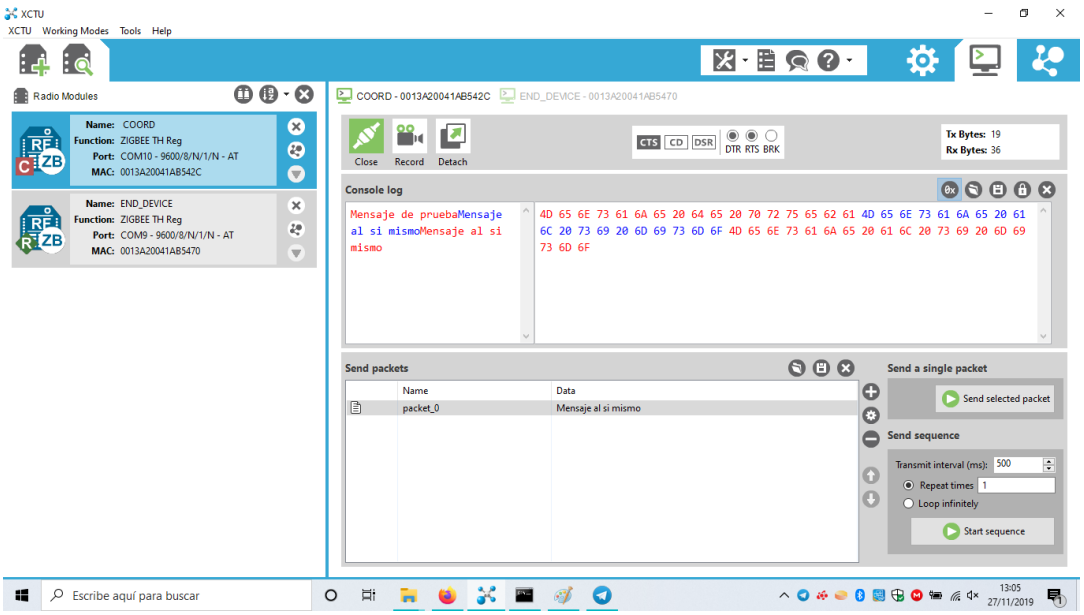
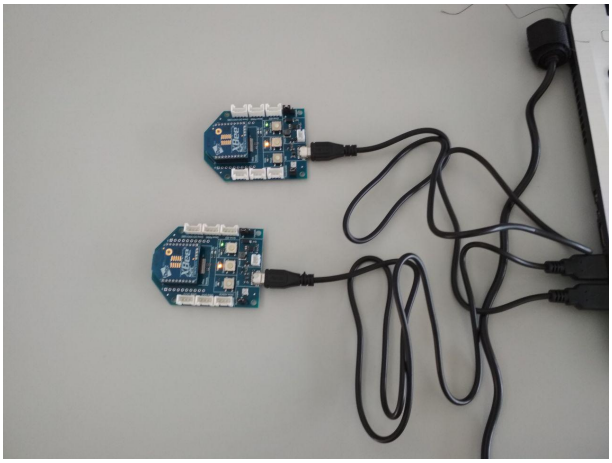


Figura 8: Conexión

En la imagen de arriba se muestra en azul lo recibido del receptor y en rojo el mismo mensaje que envía el coordinador sobre sí mismo para confirmar que le ha llegado información del receptor.

2.3.1. XBee



Este es el resultado, el cual no lleva conectado el módulo infrarrojo porque no fuimos capaces de conectarle nada estos XBee y por lo tanto el proyecto se quedó así.