

Component List:

- 1 Arduino Nano.
- 1 Beam Splitter (50-50)
- 2 Photo resistors.
- 1 Protoboard.
- 1 Cable arduino-PC or arduino-Smartphone.
- 1 Bluetooth device (optional)



Arduino Nano

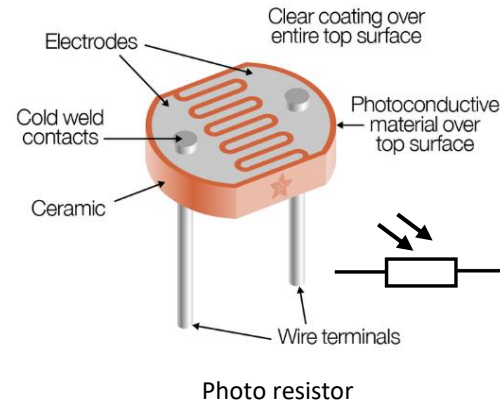
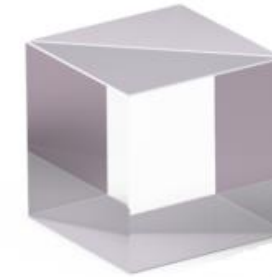
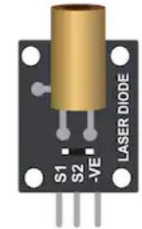


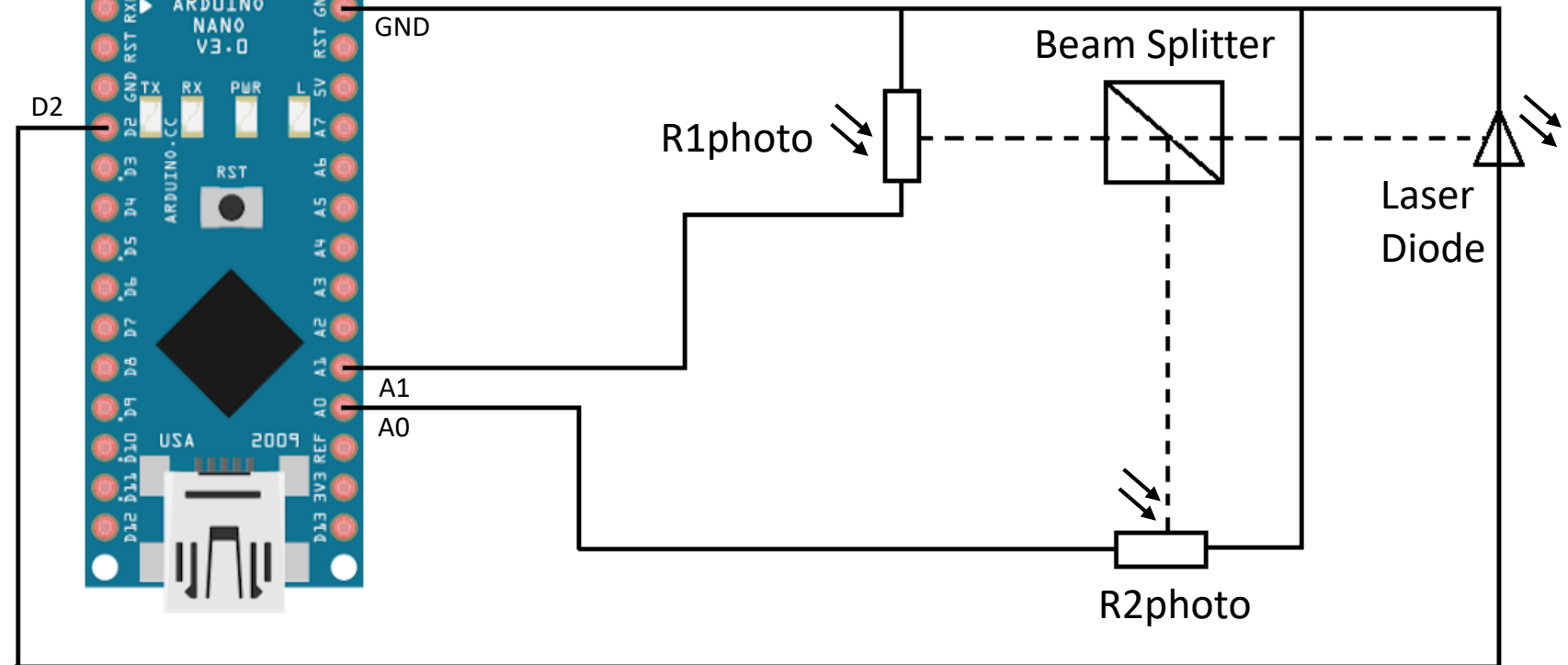
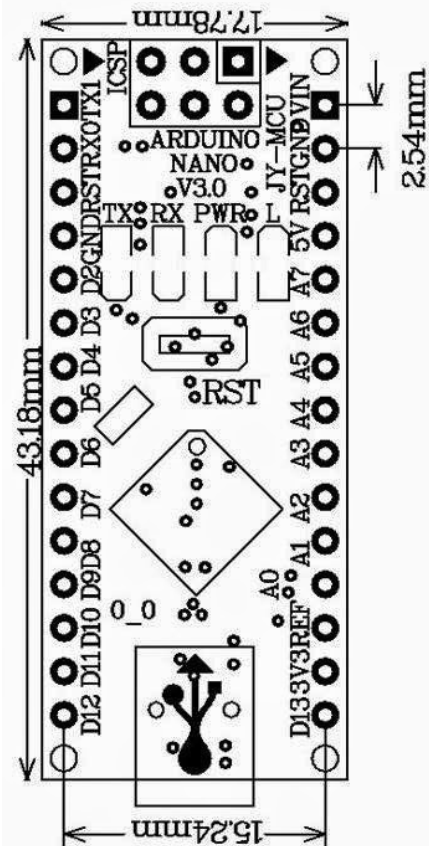
Photo resistor



Beam Splitter 50/50 (BS)



Laser Diode



QRNGv1.0.ino

Software
Program to arduino nano.

```
/* OpenQbitQRNG Firmware V1.0
*Author: Guillermo Vidal
*Copyright © 2020 OpenQbit, Inc.
*License: MIT
*/
```

```
int triggerQ = 2; // This pin will pulse our quantum circuit
int QuA0Pin = A0; // This pin measures the horizontal polarized photons
int QuA1Pin = A1; // This pin measures the vertically polarized photons
float Qu0 = 0;
float Qu1 = 0;

void setup() {
  // Just setting up triggerPin and serial connection
  pinMode(triggerQ, OUTPUT); // sets the digital pin 2 as output
  Serial.begin(9600);
}

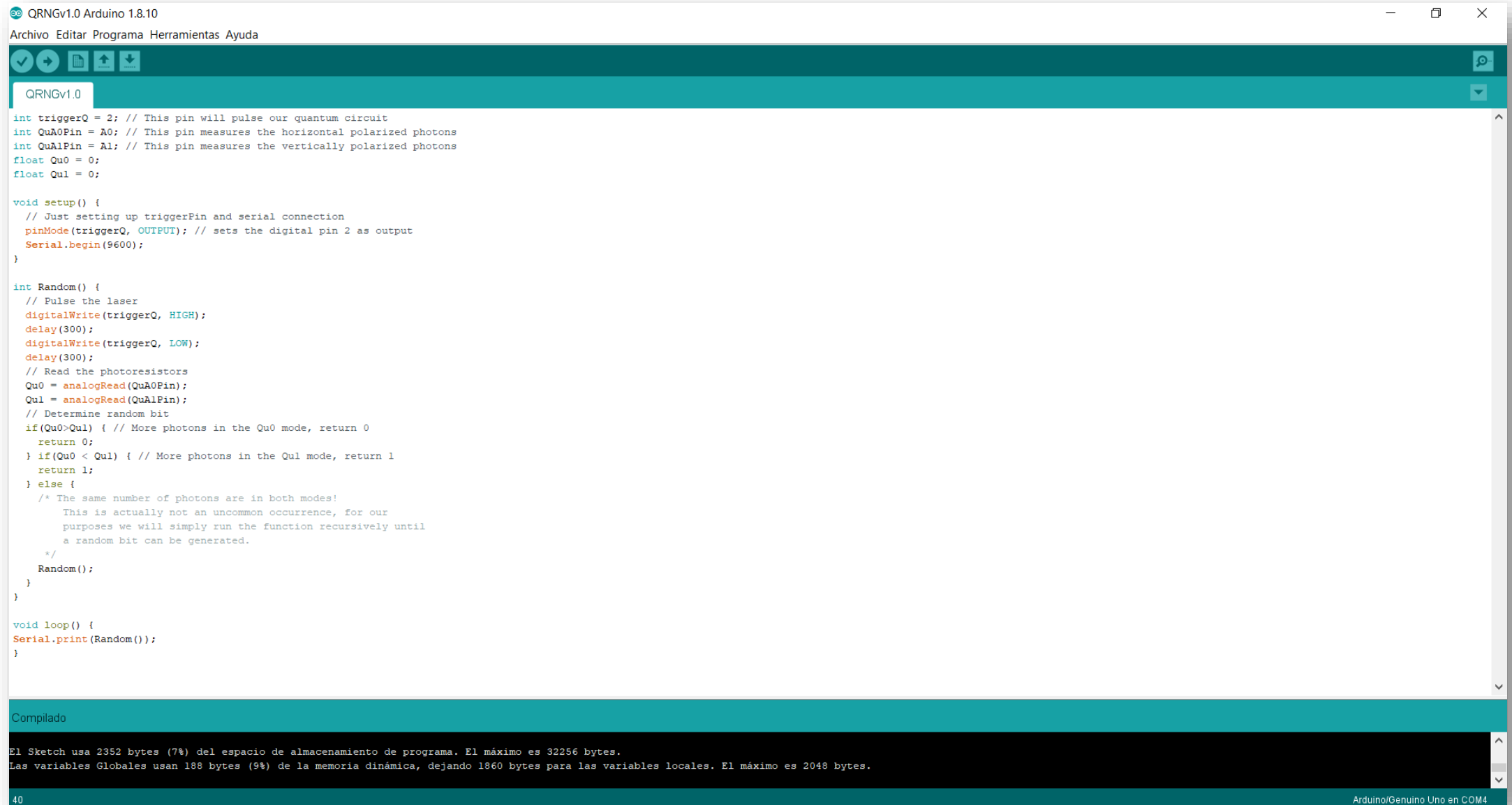
int Random() {
  // Pulse the laser
  digitalWrite(triggerQ, HIGH);
  delay(300);
  digitalWrite(triggerQ, LOW);
  delay(300);
  // Read the photoresistors
  Qu0 = analogRead(QuA0Pin);
  Qu1 = analogRead(QuA1Pin);
  // Determine random bit
  if(Qu0>Qu1) { // More photons in the Qu0 mode, return 0
    return 0;
  } if(Qu0 < Qu1) { // More photons in the Qu1 mode, return 1
    return 1;
  } else {
    /* The same number of photons are in both modes!
    This is actually not an uncommon occurrence, for our
    purposes we will simply run the function recursively until
    a random bit can be generated.
    */
    Random();
  }
}

void loop() {
  Serial.print(Random());
}
```

Output console

0010110101011110101011010.....

Compiling and uploading for arduino nano...



```
QRNGv1.0 Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda

QRNGv1.0

int triggerQ = 2; // This pin will pulse our quantum circuit
int QuA0Pin = A0; // This pin measures the horizontal polarized photons
int QuA1Pin = A1; // This pin measures the vertically polarized photons
float Qu0 = 0;
float Qu1 = 0;

void setup() {
  // Just setting up triggerPin and serial connection
  pinMode(triggerQ, OUTPUT); // sets the digital pin 2 as output
  Serial.begin(9600);
}

int Random() {
  // Pulse the laser
  digitalWrite(triggerQ, HIGH);
  delay(300);
  digitalWrite(triggerQ, LOW);
  delay(300);
  // Read the photoresistors
  Qu0 = analogRead(QuA0Pin);
  Qu1 = analogRead(QuA1Pin);
  // Determine random bit
  if(Qu0>Qu1) { // More photons in the Qu0 mode, return 0
    return 0;
  } if(Qu0 < Qu1) { // More photons in the Qu1 mode, return 1
    return 1;
  } else {
    /* The same number of photons are in both modes!
       This is actually not an uncommon occurrence, for our
       purposes we will simply run the function recursively until
       a random bit can be generated.
    */
    Random();
  }
}

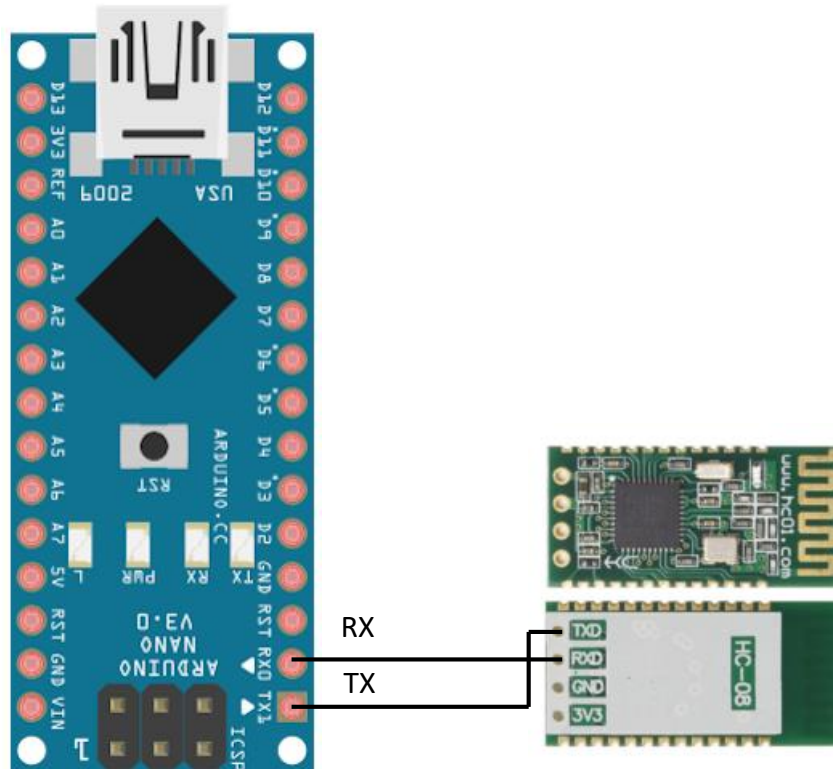
void loop() {
  Serial.print(Random());
}
```

Compilado

El Sketch usa 2352 bytes (7%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables Globales usan 188 bytes (9%) de la memoria dinámica, dejando 1860 bytes para las variables locales. El máximo es 2048 bytes.

40 Arduino/Genuino Uno en COM4

Another way to connect to Arduino Nano is through an external Bluetooth device.



We can test the correct operation using the following links where you can find examples, comments or answers to questions. This is through block programming in App inventor or similar from android.

http://kio4.com/appinventor/9A0_bluetooth_RXTX.htm

<http://kio4.com/appinventor/index.htm#bluetooth>

<https://community.appinventor.mit.edu/>