

**ESCUELA POLITÉCNICA
SUPERIOR DE CÓRDOBA**
Universidad de Córdoba



Internet de las Cosas

*Máster Universitario en Inteligencia Computacional e
Internet de las Cosas*

Práctica 1. Programando un MCU

Carlos Checa Moreno



UNIVERSIDAD DE CÓRDOBA

ÍNDICE GENERAL

EJERCICIO 1A	3
EJERCICIO 1B	8
EJERCICIO 1C	10

ÍNDICE DE FIGURAS

FIGURA 1: DISPOSITIVOS COLOCADOS	4
FIGURA 2: CONEXIONES REALIZADAS	5
FIGURA 3: CÓDIGO INICIAL	6
FIGURA 4: CÓDIGO FINAL	7
FIGURA 5: RESULTADOS SIMULACIÓN	8
FIGURA 6: CÓDIGO EJERCICIO LEDS	9
FIGURA 7: RESULTADOS SIMULACIÓN LEDS	10
FIGURA 8: CÓDIGO EJERCICIO LUZ	11
FIGURA 9: RESULTADOS SIMULACIÓN LUCES	12

Práctica 1

PROGRAMANDO UN MCU

1a – Interactuando con sensores y actuadores

Creemos un esquemático nuevo en blanco. **File -> New (Ctrl + N)**

1) Incorporar los siguientes dispositivos y elementos:

- MCU: **Components -> Boards -> MCU Board**
- Motor: **Components -> Actuators -> Motor**
- Sirena: **End Devices -> Home -> Siren**
- Luz: **End Devices -> Home -> Light**
- LED: **Components -> Actuators -> LED**
- Pulsador: **Components -> Sensors -> Toggle Push Button**
- Potenciómetro: **Components -> Sensors -> Potentiometer**
- Interruptor: **Components -> Sensors -> Rocker Switch**
- Sensor flexible: **Components -> Sensors -> Flex Sensor**

Para empezar, he colocado todos los dispositivos como se indica en el ejercicio

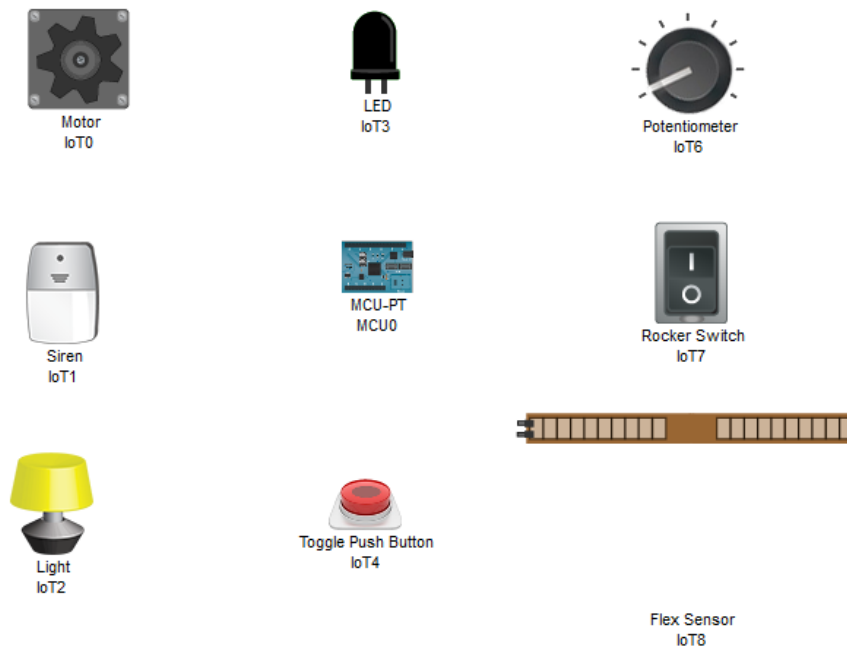


Figura 1: Dispositivos colocados

2) Interconectarlos utilizando **Connections -> IoT Custom Cable** de la siguiente manera:

- Motor Pin A0 -> MCU Pin D5
- Sirena Pin D0 -> MCU Pin D4
- Luz Pin D0 -> MCU Pin D2
- LED Pin D0 -> MCU Pin D3
- Pulsador Pin D0 -> MCU Pin D1
- Potenciómetro Pin A0 -> MCU Pin A0
- Interruptor Pin D0 -> MCU Pin D0
- Sensor flexible Pin A0 -> MCU Pin A1

He realizado la conexiones indicadas.

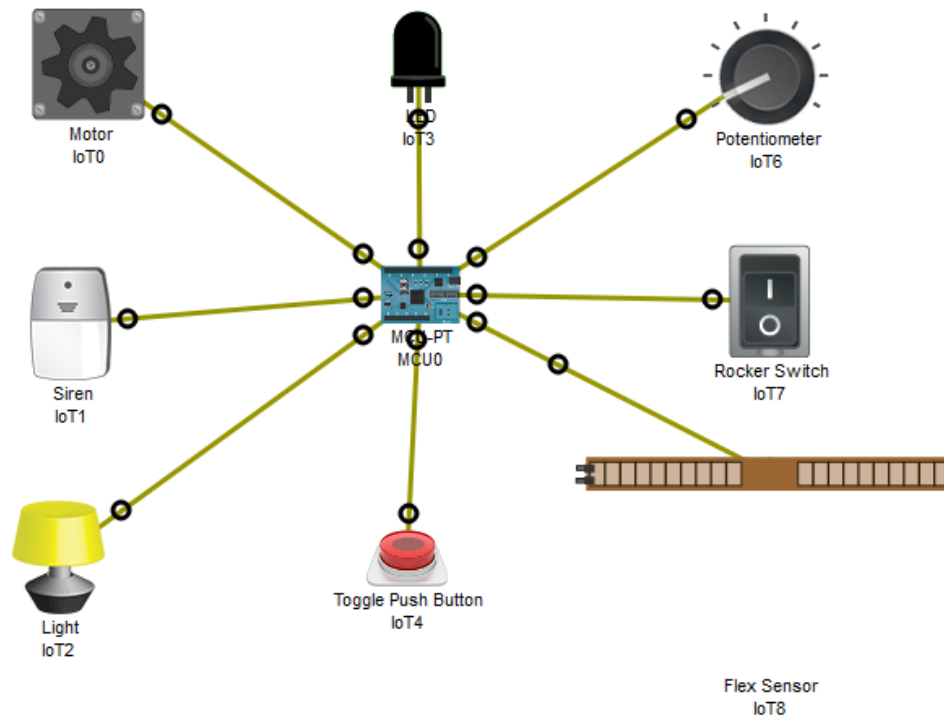


Figura 2: Conexiones realizadas

3) Diseñar el código de programación del MCU (estilo Arduino):

- Abrir el cuadro de diálogo del MCU -> Programming.
 - Pulsar el botón New.
 - Introducir un nombre de proyecto.
 - Seleccionar Template -> Empty Javascript.
- Doble click sobre “main.js” y copiar el siguiente código:

```
function setup() { pinMode(0, INPUT);

    pinMode(1, INPUT);

    pinMode(2, OUTPUT);

    pinMode(3, OUTPUT);

    pinMode(4, OUTPUT);

    pinMode(5, OUTPUT);

}

function loop() {

    // RELLENAR CÓDIGO
```

}

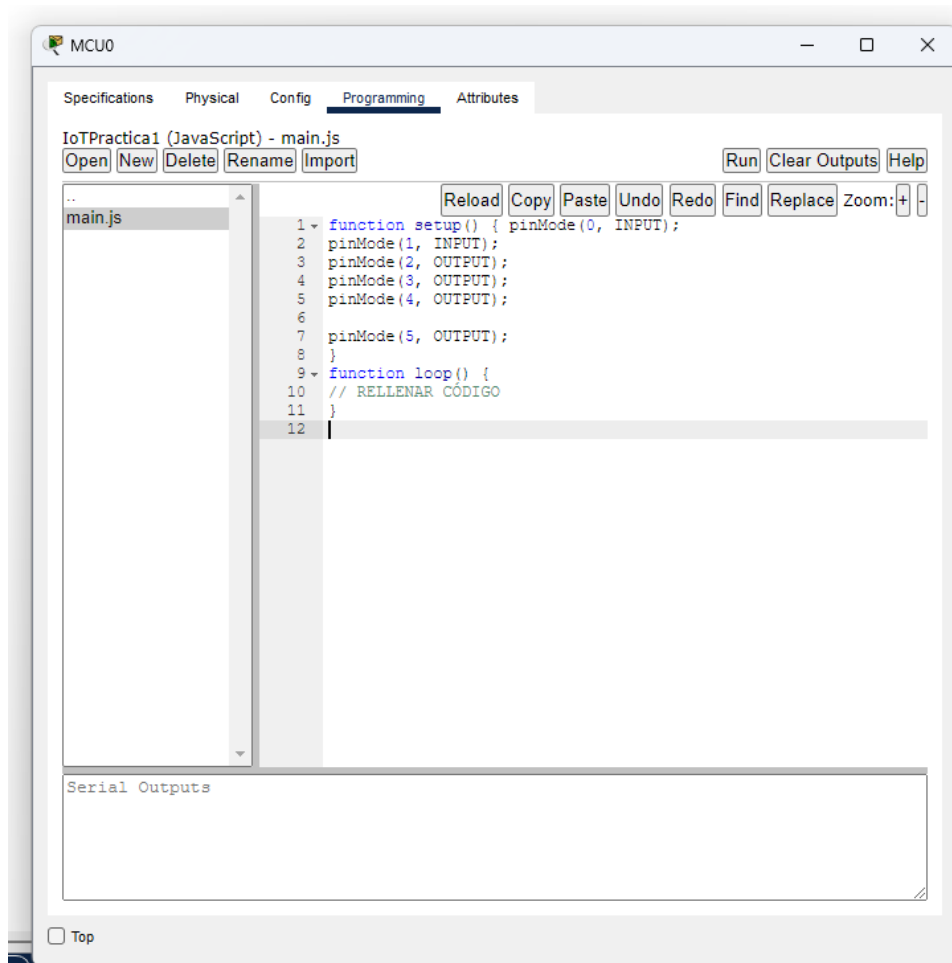
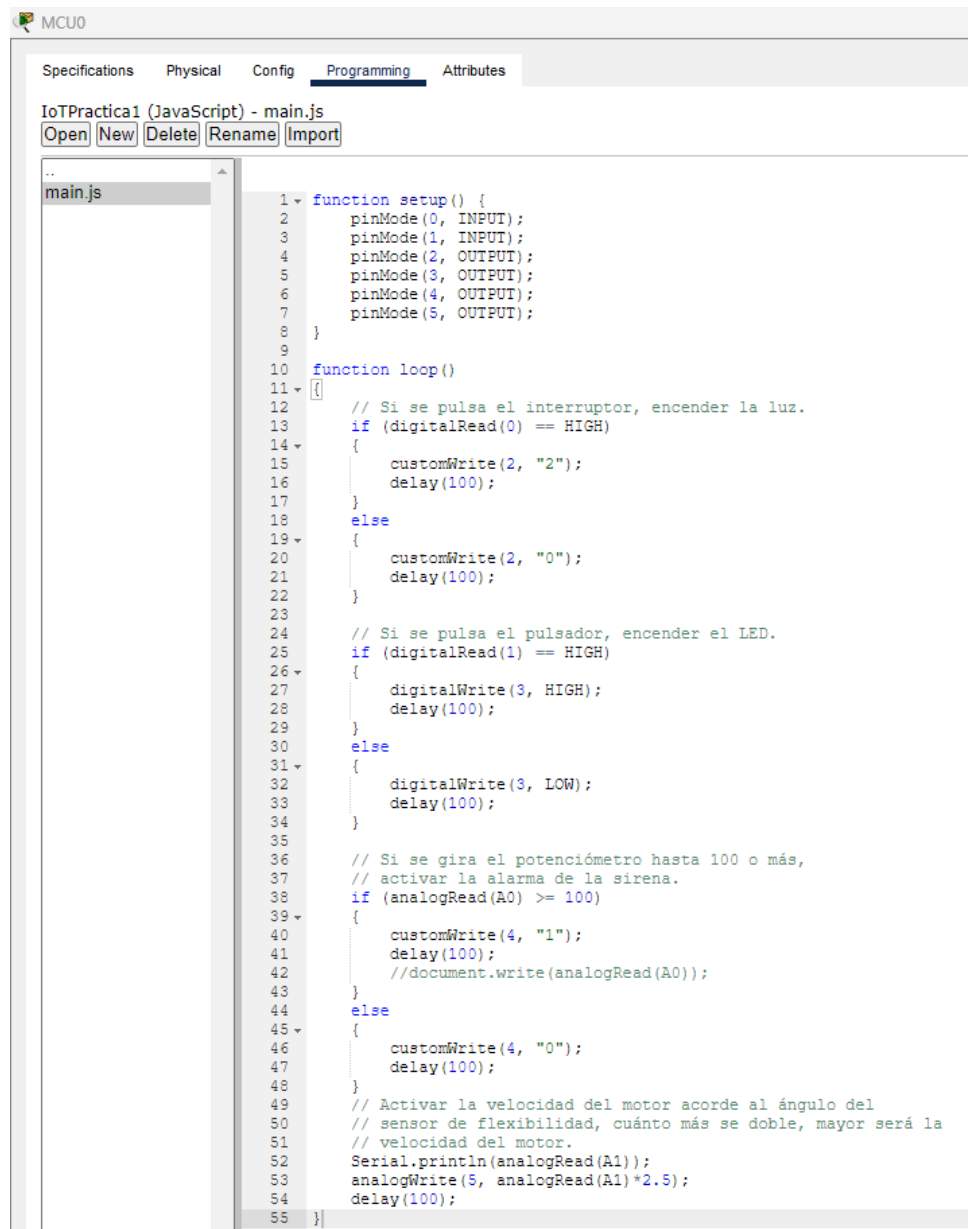


Figura 3: Código Inicial

- Rellenar el código de la función loop de tal manera que realicen el siguiente funcionamiento:
 - Si se pulsa el interruptor, encender la luz.
 - Si se pulsa el pulsador, encender el LED.
 - Si se gira el potenciómetro hasta 100 o más, activar la alarma de la sirena.
 - Activar la velocidad del motor acorde al ángulo del sensor de flexibilidad, cuánto más se doble, mayor será la velocidad del motor.
- Realice el programa con el menor número de líneas de código.

He completado el código de la función loop para implementar el comportamiento especificado en el ejercicio. Este código permite que los dispositivos respondan a las entradas: el interruptor enciende la luz, el pulsador activa el LED, el potenciómetro controla la sirena al alcanzar cierto nivel (100 o más), y el sensor de flexibilidad ajusta la velocidad del motor según su ángulo.



```
1 function setup() {  
2   pinMode(0, INPUT);  
3   pinMode(1, INPUT);  
4   pinMode(2, OUTPUT);  
5   pinMode(3, OUTPUT);  
6   pinMode(4, OUTPUT);  
7   pinMode(5, OUTPUT);  
8 }  
9  
10 function loop() {  
11   // Si se pulsa el interruptor, encender la luz.  
12   if (digitalRead(0) == HIGH) {  
13     customWrite(2, "2");  
14     delay(100);  
15   }  
16   else {  
17     customWrite(2, "0");  
18     delay(100);  
19   }  
20  
21   // Si se pulsa el pulsador, encender el LED.  
22   if (digitalRead(1) == HIGH) {  
23     digitalWrite(3, HIGH);  
24     delay(100);  
25   }  
26   else {  
27     digitalWrite(3, LOW);  
28     delay(100);  
29   }  
30  
31   // Si se gira el potenciómetro hasta 100 o más,  
32   // activar la alarma de la sirena.  
33   if (analogRead(A0) >= 100) {  
34     customWrite(4, "1");  
35     delay(100);  
36     //document.write(analogRead(A0));  
37   }  
38   else {  
39     customWrite(4, "0");  
40     delay(100);  
41   }  
42  
43   // Activar la velocidad del motor acorde al ángulo del  
44   // sensor de flexibilidad, cuánto más se doble, mayor será la  
45   // velocidad del motor.  
46   Serial.println(analogRead(A1));  
47   analogWrite(5, analogRead(A1)*2.5);  
48   delay(100);  
49 }  
50  
51  
52  
53  
54  
55 }
```

Figura 4: Código Final

Resultado de la Simulación

A continuación, en la figura 5 podemos ver la simulación para verificar el funcionamiento de los dispositivos interconectados.

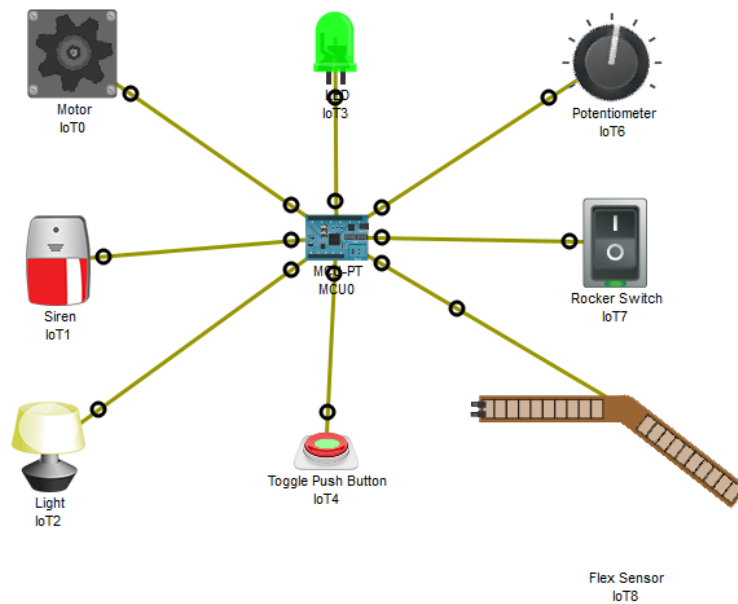


Figura 5: Resultados simulación

1b – Un sensor controlando el comportamiento de múltiples salidas

Diseñe un sistema con 8 LED (Components -> Actuators -> LED) y un pulsador (Components -> Sensors -> Toggle Push Button). Inicialmente estarán todos los LED apagados. Cada vez que se pulse el pulsador se apagará el actualmente iluminado y se encenderá el siguiente LED. Solo habrá un LED encendido cada vez. Cuando esté iluminado el 8º LED, al pulsar el pulsador quedarán los 8 LED apagados, comenzando la iteración de nuevo desde el principio.

En este ejercicio tenemos un sistema de LEDs controlado por un pulsador. Cada pulsación apaga el LED activo y enciende el siguiente, reiniciando el ciclo tras llegar al sexto LED.

IoTPractical1_b (JavaScript) - main.js

Open New Delete Rename Import

```
..
main.js
1  var nLED = -1;
2
3  function setup()
4  {
5      pinMode(0, OUTPUT);
6      pinMode(1, OUTPUT);
7      pinMode(2, OUTPUT);
8      pinMode(3, OUTPUT);
9      pinMode(4, OUTPUT);
10     pinMode(5, OUTPUT);
11     pinMode(6, INPUT); // Toggle Push Button
12 }
13
14 function loop()
15 {
16     // Interruptor -> Luz
17     if (nLED == -1)
18     {
19         for(var i = 0; i < 8; i++)
20         {
21             if (i <= 5)
22                 digitalWrite(i, LOW);
23             else
24                 analogWrite(i, 0);
25         }
26     }
27
28     if (digitalRead(6) > 1000)
29     {
30         if (nLED >= 0)
31             digitalWrite(nLED, LOW);
32
33         // Pasar al siguiente LED
34         nLED++;
35
36         // Volver al primer LED
37         if (nLED > 6)
38             nLED = 0;
39
40         // Encender el siguiente LED
41         if (nLED < 6)
42             digitalWrite(nLED, HIGH);
43
44         Serial.println(nLED);
45     }
46
47     // Esperar a que se libere el botón
48     while (analogRead(6) > 1000)
49     {
50         // Esperar
51     }
52
53     delay(1);
54 }
55 }
```

Figura 6: Código ejercicio LEDs

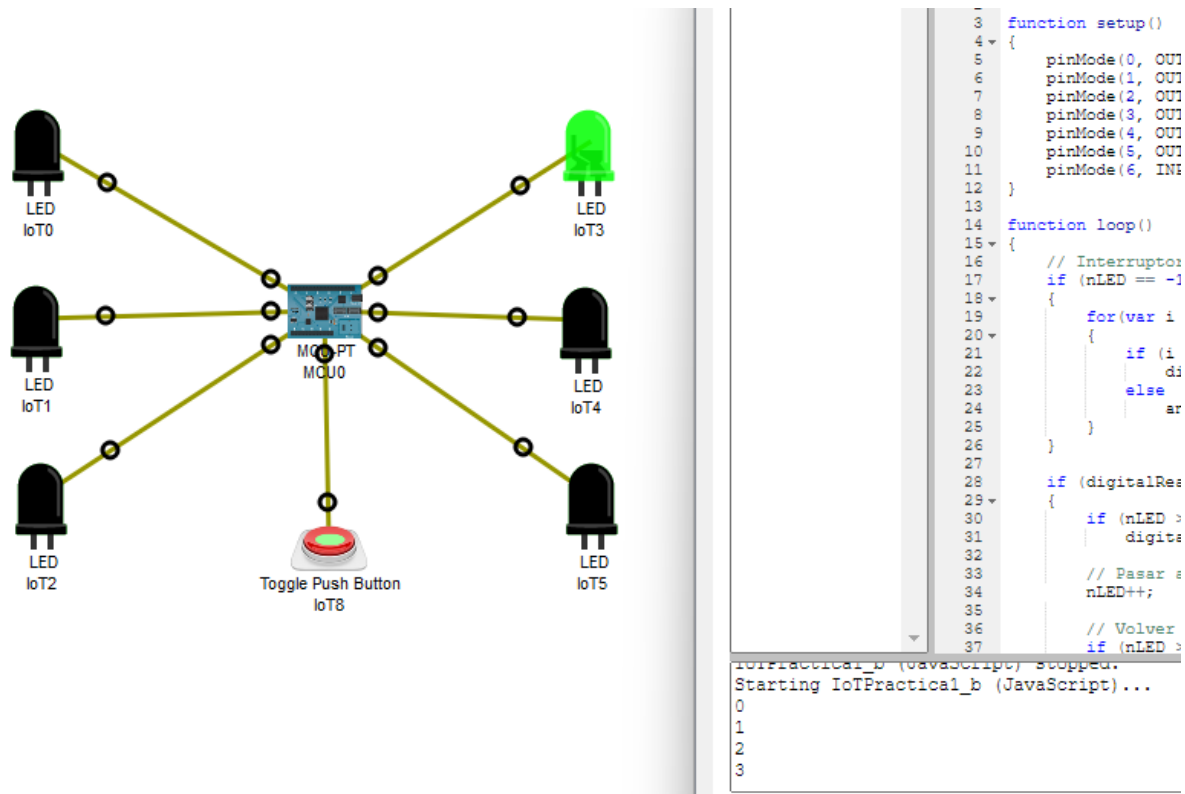


Figura 7: Resultados simulación LEDs

1c – Controlar una salida con dos parámetros mediante dos sensores diferentes

Diseñe un sistema con una lámpara (End Devices -> Home -> Light), un pulsador (Components

-> Sensors -> Toggle Push Button) y un potenciómetro (Components -> Sensors -> Potentiometer). La lámpara tiene 3 posibles valores de iluminación (0, apagado; 1, luz suave; 2, luz intensa). Seleccionar mediante el potenciómetro el valor de iluminación y mientras se mantenga presionado el pulsador, la luz de la lámpara se iluminará con el valor indicado en el potenciómetro.

Por último, para este ejercicio se ha desarrollado el sistema de control de la lámpara. El código permite ajustar la intensidad de la luz mediante el potenciómetro y activarla con el pulsador.

```
IoTPractical1_c (JavaScript) - main.js
Open New Delete Rename Import

main.js
1 function setup()
2 {
3   pinMode(1, INPUT);
4 }
5
6 function loop()
7 {
8
9   // Valores 0 - 1023
10  if (digitalRead(0) == HIGH)
11  {
12    if (analogRead(A0) < 1023 / 3) // Apagada de 0 a 1023 / 3
13    {
14      Serial.println("Apagada");
15      customWrite(A1, "0");
16    }
17    else if (analogRead(A0) < 1023 / 3 * 2) // Luz suave de 1023 / 3 a 1023 / 3 * 2
18    {
19      Serial.println("Luz Suave");
20      customWrite(A1, "1");
21    }
22    else // Encendida de 1023 / 3 * 2 a 1023
23    {
24      Serial.println("Encendida");
25      customWrite(A1, "2");
26    }
27  }
28  else
29  {
30    Serial.println("Apagada");
31    customWrite(A1, "0");
32  }
33 }
```

Figura 8: Código ejercicio Luz

En la figura 9 se encuentra simulado el sistema de la lámpara para comprobar que el control de la iluminación funciona correctamente según el diseño, desde la luz apagada para valor bajos del potenciómetros hasta la luz intensa para los mayores valores del potenciómetro.

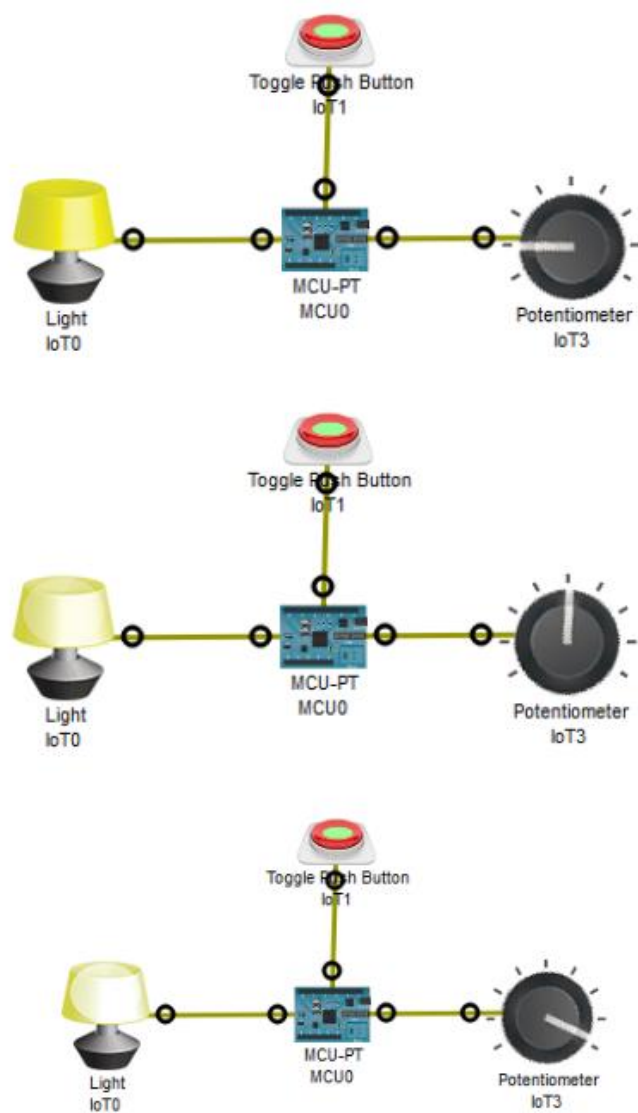


Figura 9: Resultados simulación luces

Práctica 2
