

# Cromoterapia Windows IoT

Juan Manuel Rodríguez Pérez



## Contenidos

1. Introducción .....	2
2. Que necesitamos .....	3
2.1 Software .....	3
5.1 Hardware.....	3
10. Instalando el Dashboard.....	4
11. Montando el Hardware.....	6
11.1 Sección Arduino .....	6
10.1.1 Elementos hardware usados .....	6
10.1.2 Diagrama circuito electrico .....	6
11.2 Sección Raspberry Pi .....	7
10.1.1 Elementos hardware usados .....	7
10.1.2 Diagrama circuito eléctrico .....	7
12. Creando el Software.....	8
12.1 Sección Arduino .....	8
12.2 Sección Raspberry Pi .....	10
13. Imágenes del proceso.....	11
13.1 Arduino .....	11
13.2 Raspberry Pi.....	12
14. Bibliografía.....	13
15. Control de versiones .....	14

## 1. Introducción

Proyecto para usar Raspberry Pi, Arduino y Windows IoT

## 2. Que necesitamos

### 2.1 Software

1. Windows 10 IoT Core Dashboard
2. Windows IoT Remote Client (Windows Store)
3. Windows 10 IoT Core Creators Update ISO for Raspberry Pi 2 & 3.
4. Arduino IDE
5. Fritzing

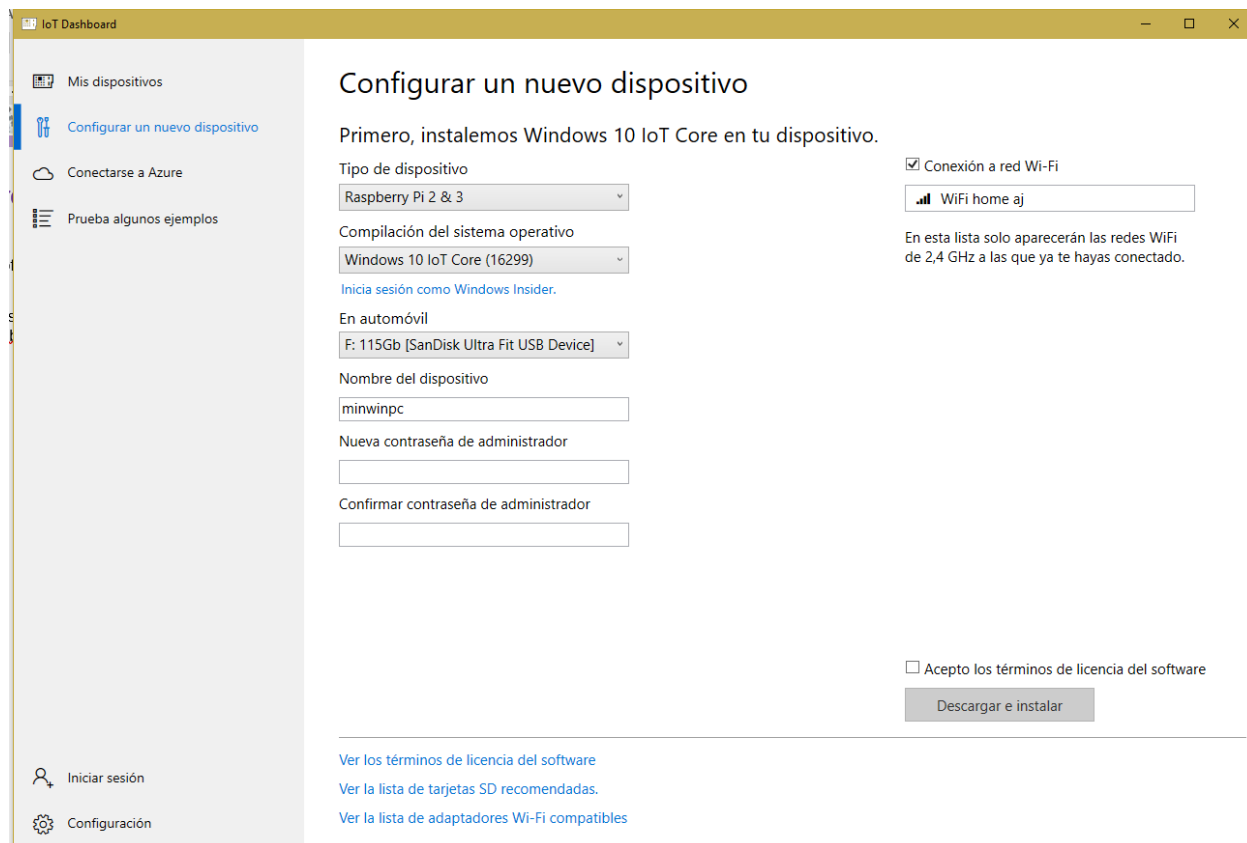
### 5.1 Hardware

1. Raspberry Pi 3 model B
2. Arduino (Placa tipo UNO o MEGA)
3. RGB Strip Led
4. Arduino Proto Shield
5. Bluetooth (HC-05 o HC-06)
6. PortaBaterias
7. Interruptor
8. Cables
9. IRFZ44N IRFZ44 TO-220 Mosfet Transistor

## 10. Instalando el Dashboard

Iniciamos el proyecto instalando Windows 10 IoT Dashboard una vez descargado de la página de Microsoft y siguiendo el asistente.

Una vez instalado deberemos descargar la imagen ISO de Windows IoT Core y usando la aplicación Dashboard instalaremos el sistema operativo en la Raspberry Pi 3 model B.



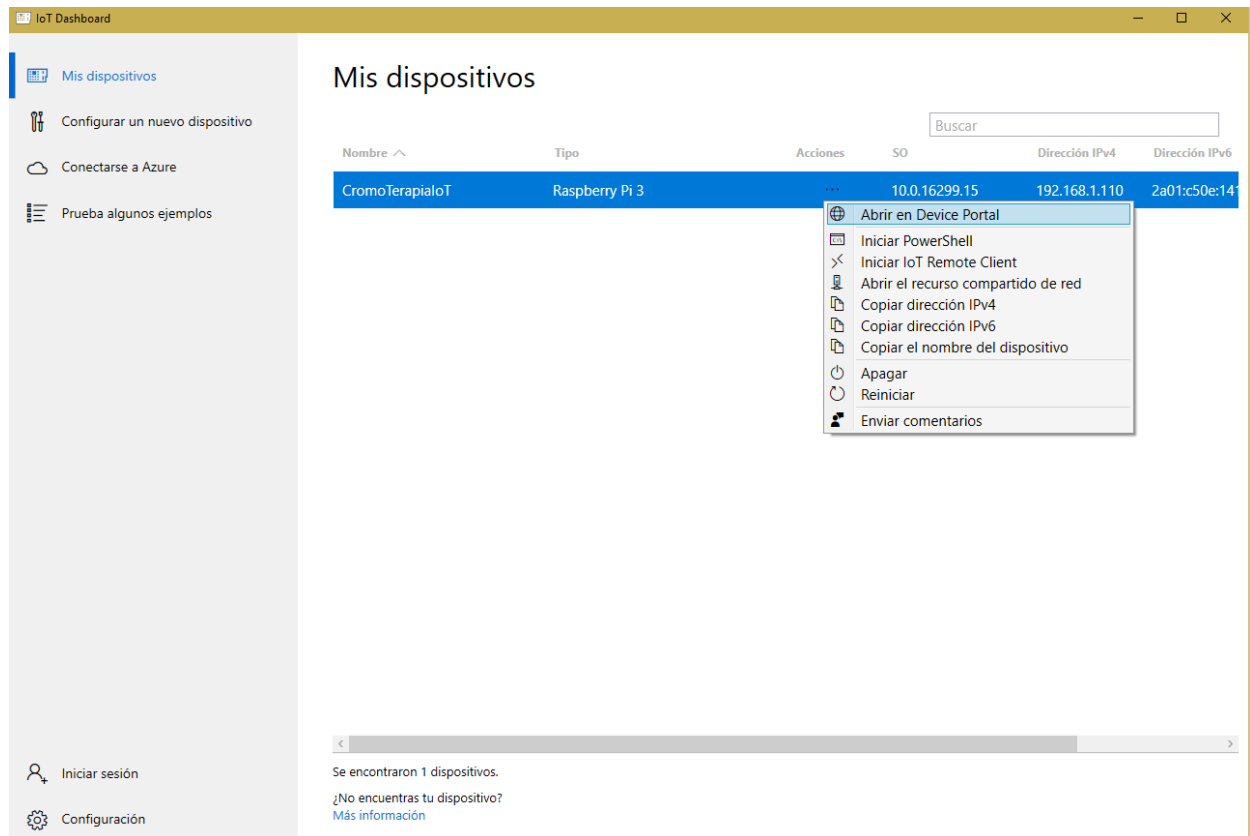
En este punto debemos tener en cuenta que podemos usar una tarjeta SD con adaptador USB.

Se debe conectar la tarjeta SD al PC que usamos para realizar dicha instalación de momento no nos hace falta la Raspberry Pi.

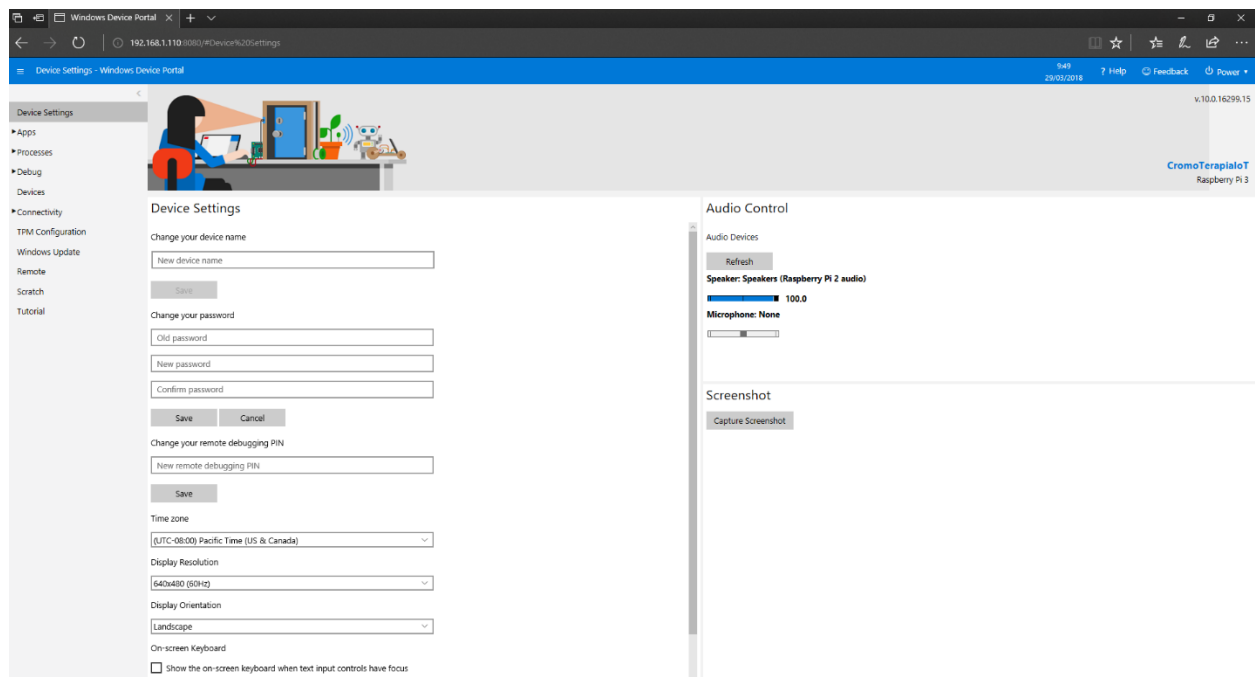
Colocamos la tarjeta SD en la Raspberry Pi en mi caso es la 3 model B y tendremos el lector de la tarjeta SD en la parte inferior de la placa.

La Raspberry Pi debe estar conectada a la red indicada en las opciones de la configuración.

Una vez que la detecte podremos ver el dispositivo y el menú de opciones para el mismo.



Abrimos el Device Portal, nos pedirá el user y el pass (ojo el user puede ser Administrator) una vez logueados nos mostrará la siguiente pantalla:



Desde el Dashboard podremos cambiar el nombre al dispositivo, controlar los drivers, las conexiones, etc.

## 11. Montando el Hardware

### 11.1 Sección Arduino

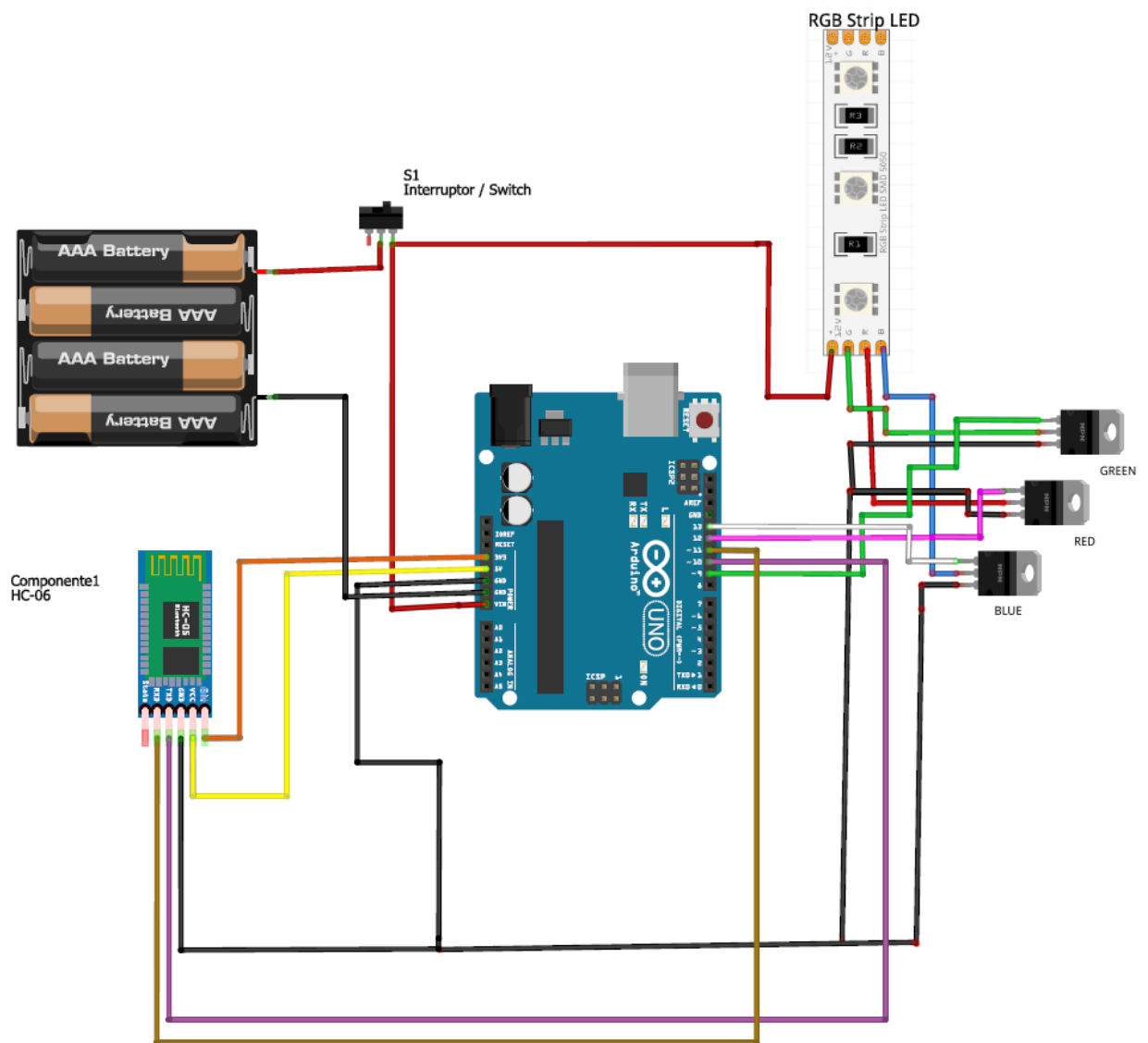
#### 10.1.1 Elementos hardware usados

Para la parte de Arduino de nuestro proyecto necesitaremos:

- PortaBaterias
- Interruptor
- RGB Strip Led
- Componente Bluetooth (en mi caso he usado un HC-06, podría valer el HC-05)
- Placa Arduino (en mi caso es la UNO)
- Mosfet Transistor
- Cables

#### 10.1.2 Diagrama circuito electrico

Usando la aplicación **fritzing** creamos nuestro prototipo de circuito.



fritzing

## 11.2 Sección Raspberry Pi

### 10.1.1 Elementos hardware usados

Para la parte de RaspberryPi de nuestro proyecto necesitaremos:

- a. Placa Raspberry Pi (en mi caso Model B)
- b.

### 10.1.2 Diagrama circuito eléctrico



## 12. Creando el Software

A la hora de desarrollar el código para la parte de Arduino, he buscado información para ver si había un simulador donde pudiéramos crear las conexiones y probar el código realizado.

He encontrado una que esta entre las recomendadas, y de las que he probado me ha gustado más, es haciéndolo online desde la página de **“tinkercad”**.

### 12.1 Sección Arduino

```
#include <SoftwareSerial.h>

//Variables
const byte rxPin = 11;
const byte txPin = 10;
const int GREEN_pin = 3;
const int BLUE_pin = 5;
const int RED_pin = 6;
String RGBColor = "";
String RGBColor_previous = "255.255.255"; // inicializacion a blanco
boolean RGBRead_Completed = false;
long bps = 9600; //bytes por seg

SoftwareSerial myConnect(txPin, rxPin); // RX, TX recordar que se cruzan

void setup()
{
  pinMode(GREEN_pin, OUTPUT);
  pinMode(BLUE_pin, OUTPUT);
  pinMode(RED_pin, OUTPUT);

  Serial.begin(bps);
  myConnect.begin(bps);
}

void loop()
{
  while(myConnect.available())
  {
    char ReadChar = (char)myConnect.read();

    if(ReadChar == ')')
    {
      RGBRead_Completed = true;
    }
    else
    {
      RGBColor += ReadChar;
    }
  }

  if(RGBRead_Completed)
```

```

    {
    if(RGBColor == "ON")
    {
        RGBColor = RGBColor_previous;
        Control_RGB_StripLed();
    }
    else if(RGBColor == "OFF")
    {
        RGBColor = "0.0.0";
        Control_RGB_StripLed();
    }
    else
    {
        Control_RGB_StripLed();
        RGBColor_previous = RGBColor;
    }

    RGBColor = "";
    RGBRead_Completed = false;
    }
}

```

```

void Control_RGB_StripLed() {
    // get finish position of colors
    int position1RGBColor = RGBColor.indexOf(' ');
    int position2RGBColor = RGBColor.indexOf(' ', position1RGBColor + 1);
    int position3RGBColor = RGBColor.indexOf(' ', position2RGBColor + 1);

    // get colors from string RGBColor with previous positions
    String RedColor = RGBColor.substring(0, position1RGBColor);
    String GreenColor = RGBColor.substring(position1RGBColor + 1, position2RGBColor);
    String BlueColor = RGBColor.substring(position2RGBColor + 1, position3RGBColor);

    //Pass string colors to int and get color variety between 0 - 255
    analogWrite(GREEN_pin, (GreenColor.toInt())); // you can use de analogWrite if you use this
    arduino's conexions
    analogWrite(BLUE_pin, (BlueColor.toInt()));
    analogWrite(RED_pin, (RedColor.toInt()));

    // If you need to invert colors, you could change GreenColor.toInt() by (255 -
    GreenColor.toInt())
}

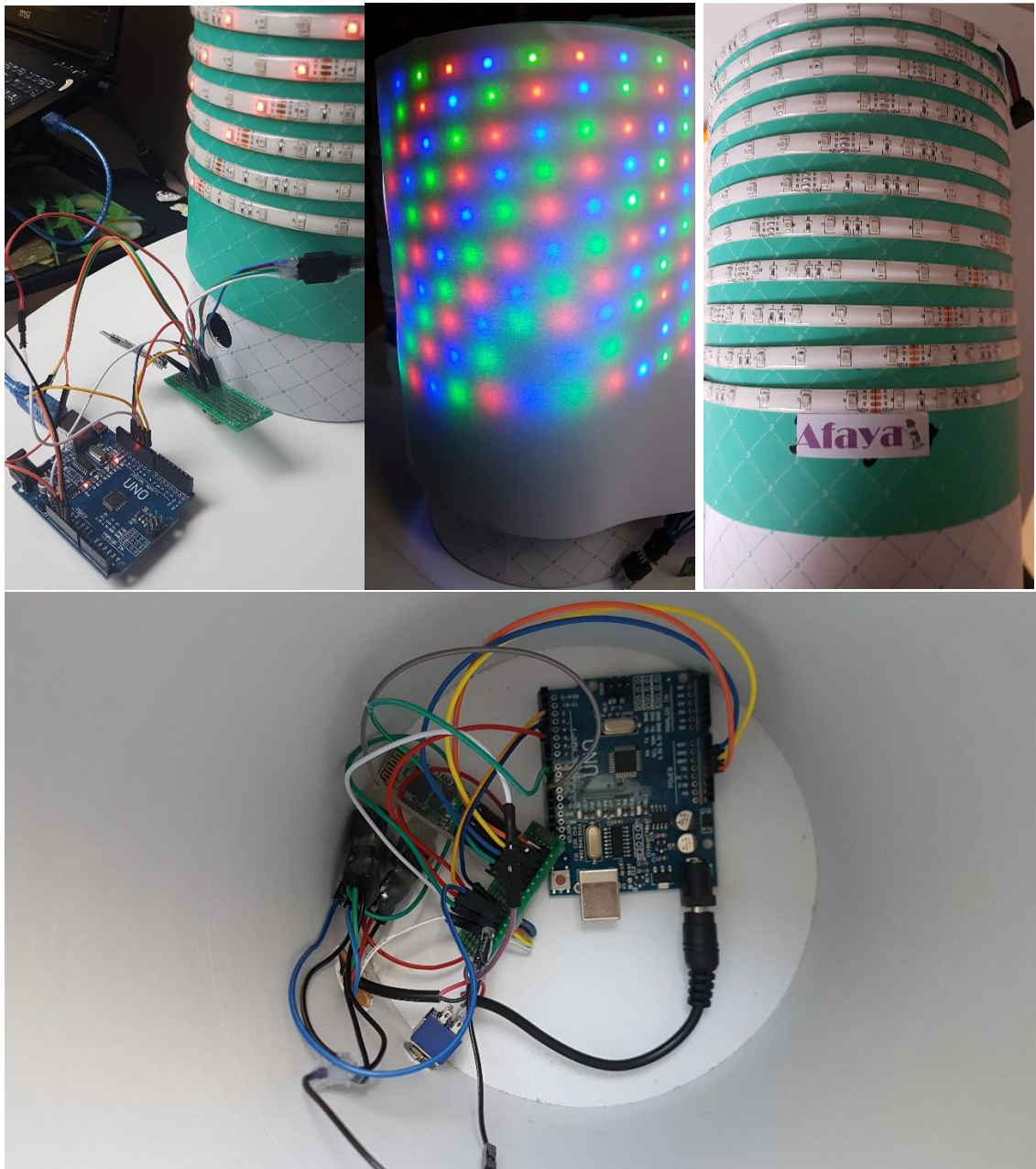
```

}

## 12.2 Sección Raspberry Pi

## 13. Imágenes del proceso

### 13.1 Arduino



## 13.2 Raspberry Pi

## 14. Bibliografía

Enlaces interesantes para tener en cuenta o donde se ha obtenido información.

- Windows IoT: <https://developer.microsoft.com/es-es/windows/iot>
- Windows IoT download: <https://developer.microsoft.com/en-us/windows/iot/Downloads>
- Samples with Windows IoT: <https://www.hackster.io/microsoft/products/windows-10-iot-core?sort=trending>
- Temperature Sample : <https://www.hackster.io/Alirezap/windows-10-iot-temperature-smt172-6cdabe>
- Chromotherapy Sample: <https://www.hackster.io/lentzlive/chromotherapy-with-raspberry-and-arduino-69d11e>
- Weather Station: <https://www.hackster.io/windows-iot/weather-station-67e40d>
- Windows IoT Remote Control: Store de Windows 10
- Programa diseño diagramas eléctricos: <http://fritzing.org/>
- Programa online para testear código y circuito de Arduino: <https://www.tinkercad.com/learn/>
- Arduino cc: <https://www.arduino.cc/>

## 15. Control de versiones

Versión	Descripción	Fecha
<b>1.0</b>	Versión Inicial	28/03/2018
<b>1.1</b>	Añadida Bibliografía	29/04/2018
<b>1.2</b>	Añadido hardware y software de la parte de Arduino	25/07/2018
<b>1.3</b>	Modificación circuito Arduino	16/09/2018