

## Hardware libre + Cerradura en arduino

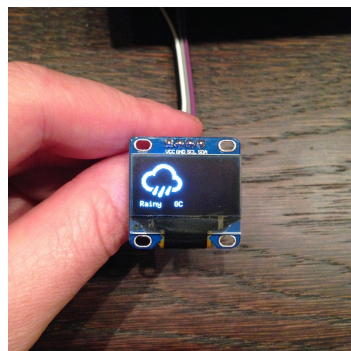
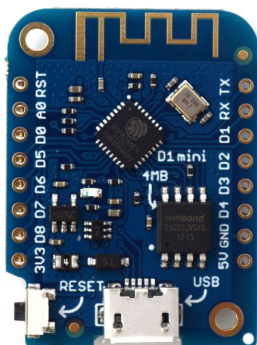
Aunque arduino nos permite hacer muchos proyectos interesantes, no es uno de las placas más avanzadas que hay. Ya que hemos visto algo de electrónica, conviene explicar algunos otros componentes que pueden ser interesantes y nos permiten avanzar en nuestro estudio de la electrónica.

¿Te suenan otros dispositivos a parte de el arduino? ¿Cuáles?

Existen varios tipos, depende de la potencia y el objetivo de nuestros proyectos. Uno de los más conocidos es **Raspberry Pi**. ¡La RPI en realidad es un ordenador completo! pero igual que nuestro arduino, nos permite conectar cosas, ampliarlo, modificarlo, etc. Permite hacer infinidad de proyectos. Como ha tenido tanto éxito, tiene varias copias, como la “Orange Pi” o la Pine 64.



A parte, existen componentes parecidos a Arduino pero menos potentes y mucho más pequeños. WEMOS



¿Para qué crees que pueden usarse estos componentes menos potentes y más pequeños?

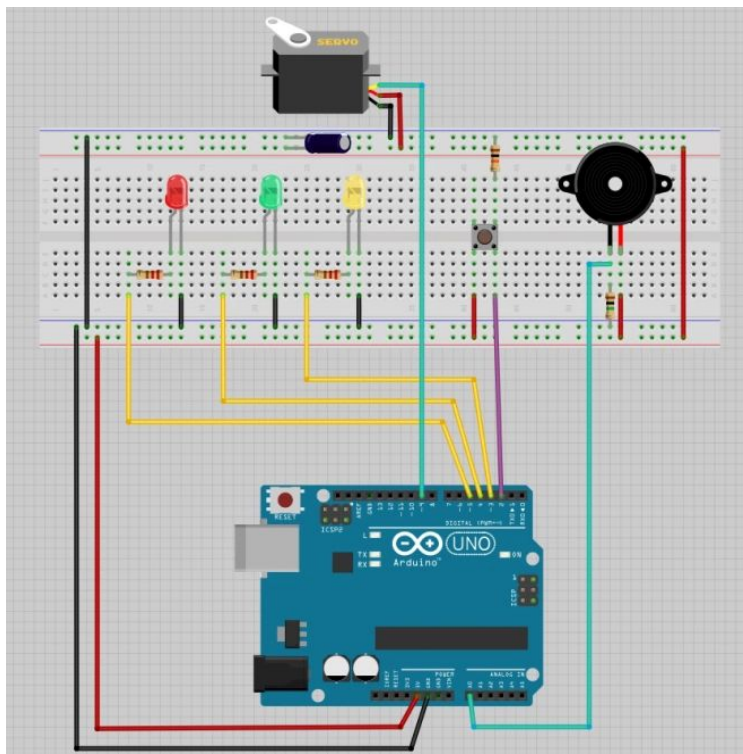
Existe algo llamado **IoT** (Internet of Things) que nos permite añadirle conexión a internet a electrodomésticos y a nuestra vida diaria, con la intención de mejorarla. Estos pequeños dispositivos nos permiten hacerlo en nuestra casa, open hardware.

¿Podrías dar un ejemplo de utilidad de IoT en varias situaciones? Nuestra vida diaria, una tienda, un museo...

Para acercarnos a un poco más de complejidad en la electrónica vamos a probar un proyecto muy completo, relacionado con seguridad, un cerrojo. Podemos consultar nuestro libro, página 125. Para ello vamos a utilizar:

- Piezo
- Servomotor
- Interruptor (botón)
- leds rojo, amarillo y verde
- resistencia de 10KOhm (naranja-negra-marrón)
- resistencia de 20KOhm (marrón-rojo-rojo)
- resistencia de 1 MegOhm (verde-negra-marrón)
- condensador de 100uF
- 3 clavijas

Lo montaremos así:



Completa el código:

```
#include <Servo.h>
Servo myServo;
const int piezo = A0;

//completa desde aqui
const int switchPin = ;
const int yellowLed = ;
const int greenLed = ;
const int redLed = ;
//hasta aqui

int knockVal;
int switchVal;
const int quietKnock = 10;
const int loudKnock = 100;

//¿Sabes que es un boolean?
boolean locked = false;

int numberOfKnocks = 0;
void setup(){

    myServo.attach(9);
    pinMode(yellowLed, OUTPUT);

    //completa desde aquí
    pinMode(redLed, );
    pinMode( , OUTPUT);
    //hasta aquí

    pinMode(switchPin, INPUT);
    Serial.begin(9600);
    digitalWrite(greenLed, HIGH);
    myServo.write(0);
    Serial.println("Cerrojo desbloqueado!");

} //Fin de la función setup.
void loop(){

    if(locked == false){
        switchVal = digitalRead();
    if(switchVal == HIGH){
        locked = true;
        digitalWrite(greenLed, LOW);
        digitalWrite(redLed, HIGH);
    }
}
```

```
        myServo.write(90);  
        Serial.println("Cerrojo bloqueado!");  
        delay(1000);  
    }  
}  
  
if(locked == true){if(numberOfKnocks >= 3){  
    locked = false;  
    myServo.write(0);  
    delay(20);  
    digitalWrite(greenLed, HIGH);  
    digitalWrite(redLed, LOW);  
    Serial.println("Cerrojo desbloqueado");  
}  
}  
  
} //Fin de la función loop.  
  
    knockVal = analogRead(piezo);  
if(numberOfKnocks < 3 && knockVal > 0){  
    if(checkForKnock(knockVal) == true){  
        numberOfKnocks++;  
    }  
    Serial.print("Golpes restantes: ");  
    Serial.println(3-numberOfKnocks);  
}  
  
boolean checkForKnock(int value){  
if(value > quietKnock && value < loudKnock){  
digitalWrite(yellowLed, HIGH);  
    delay(50);  
    digitalWrite(yellowLed, LOW);  
    Serial.print("Valor del golpe correcto: ");  
    Serial.println(value);  
return true;  
}  
else {  
    Serial.print("Valor del golpe incorrecto: ");  
    Serial.println(value);  
    return false;  
}  
  
} //Fin de la funcion checkForKnock.
```

Para probarlo debemos subir el programa (comprobando que todo está bien), abrir el Serial Monitor y probar nuestro proyecto, ¿Cómo funciona?

¿Qué hay de diferente en este código con respecto a otros que hemos visto?