

Universidad Rey Juan Carlos I

Escuela Técnica Superior de Ingeniería
Informática

Titulación: Ingeniería de Computadores

Asignatura: Diseño de Sistemas Empotrados

Mini Nevera Portátil

Grupo: 13

Alumnos:

Adrián Humia García

Laura Sánchez Muñoz

Hugo García Castanedo

Óscar Utrilla Mora



Índice

1. Introducción	2
2. Casos de uso.....	2
3. Pasos dados	3
4. Reparto de tareas	3
5. Componentes y costes	4
6. Esquemas de montaje	7
7. Problemas y soluciones	9
8. Código	9
9. Fotos sobre el hardware	17
10. Vídeo explicativo del proyecto	18

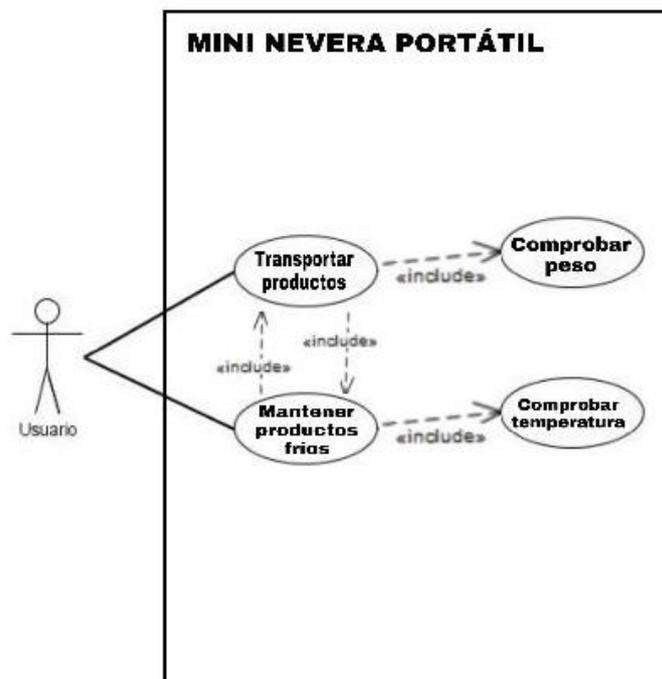
1. Introducción

Este proyecto ha sido desarrollado por el grupo 13, formado por los alumnos Hugo García Castanedo, Adrián Humia García, Laura Sánchez Muñoz y Óscar Utrilla Mora. El proyecto consiste en una nevera portátil controlable mediante una aplicación móvil. La idea surgió a raíz de la visita a un blog en el que se explicaba cómo se había desarrollado una nevera que seguía a un individuo en función de las coordenadas pasadas a través de un módulo bluetooth (referencia al blog: <https://programarfacil.com/podcast/nevera-arduino-perrito-faldero/>).

El funcionamiento de nuestro proyecto es muy similar al que se presenta en este blog, pero nuestra implementación incluye algunas mejoras respecto al otro. En primer lugar, en vez de transmitir coordenadas, damos la posibilidad al usuario de controlar directamente el movimiento de la nevera. En segundo lugar, incluimos un sensor de temperatura que nos permite mostrar la temperatura interior de la nevera a través de un anillo de 12 LEDs colocado en el exterior del chasis. Por último, también se incluye una balanza para mostrar el peso de la nevera en todo momento (también a través de un anillo de 12 LEDs).

2. Casos de uso

Los casos de uso vienen dados en el siguiente diagrama de casos de uso:



3. Pasos dados

Para realizar el proyecto se han seguido una serie de pasos:

Como primer paso, los integrantes del grupo empezamos, una vez teníamos clara la idea acerca de nuestro proyecto, pensando qué materiales y componentes debíamos comprar y cuáles en propiedad del grupo íbamos a aprovechar.

El segundo paso fue anclar los motores, las ruedas, el portador de pilas y el controlador de los motores a la base principal o chasis, y conectar estos componentes a la placa de Arduino.

En el tercer paso procedimos a la búsqueda y descarga de la aplicación móvil para controlar el movimiento de los motores. También conectamos el módulo Bluetooth.

Como cuarto paso, procedimos a conectar el sensor de temperatura y la balanza.

En el quinto paso conectamos los dos anillos de 12 LEDs y comprobamos que todos los componentes funcionaban correctamente.

El sexto paso fue diseñar y crear la estructura de cartón para poder colocar la nevera encima, permitiendo a la balanza medir el peso correctamente. Además, adquirimos una caja de mascarillas para utilizarla como nevera y la forramos de papel de aluminio.

Por último y para embellecer nuestra mini nevera portátil, forramos con goma eva negra toda la estructura de cartón.

Cabe destacar que, a la vez que se realizaban todos estos pasos respectivos al hardware, se iba implementando el código correspondiente.

4. Reparto de tareas

El reparto de tareas se ha realizado de forma equitativa. Los cuatro miembros del grupo hemos participado en todas y cada una de las tareas, aunque, en ocasiones, algún miembro se encargaba más en profundidad de alguna parte en concreto.

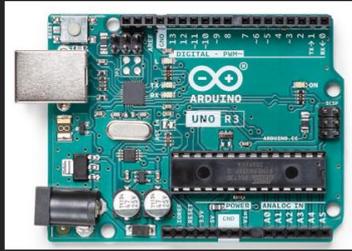
Esto ha sido posible gracias a la realización del proyecto en el horario de clase y en salas en grupo facilitadas por la URJC. Allí, mientras dos miembros del

grupo se encargaban del hardware, los otros dos miembros se encargaban del código.

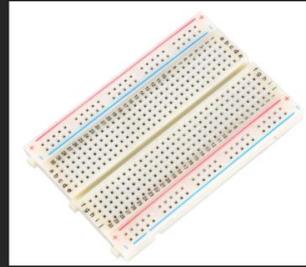
5. Componentes y costes

Placa Arduino uno	1	Proporcionado por la universidad
Protoboard	1	En propiedad del grupo
Tornillos y Tuercas	Varias unidades	En propiedad del grupo
Pilas AA 1.5V	8	1.60€
Porta pilas 4 pilas AA	1	5.20€
Chasis de plástico	1	En propiedad del grupo
Interruptor	1	3.50€
Ruedas y motores reductores para coche	2 ruedas 2 motores	9.90€
Controlador de movimiento L298N	1	7.99€
Rueda giratoria	1	1.57€
Báscula HX711 con conversor analógico digital	1	7.99€
Sensor de temperatura y humedad DHT11	1	6.29€
Anillos RGB LED de 12 Diodos cada uno	2	8.66€
Módulo bluetooth HC-05	1	8.45€
Coste Total		65.85€

- Placa Arduino Uno



- Protoboard



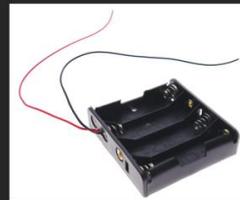
- Tornillos y tuercas



- 8 pilas AA de 1,5 V



- Portapilas



- Interruptor



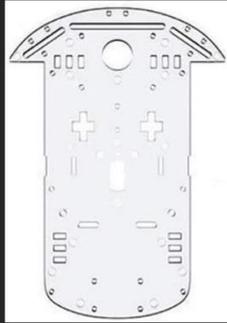
- 2 ruedas y 2 motores con reductor para coche



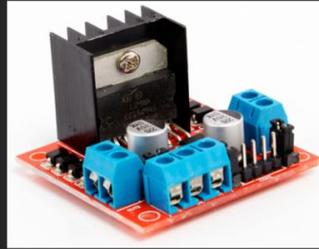
- Rueda giratoria



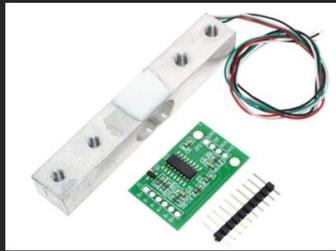
- Base principal



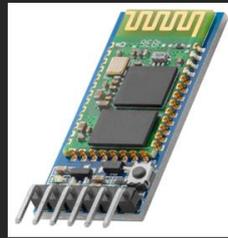
- Controlador de movimiento L298N



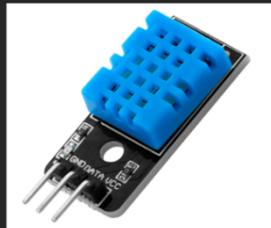
- Sensor de peso y módulo conversor analógico digital HX711



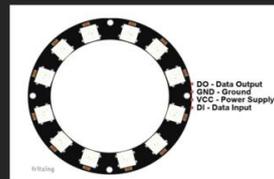
- Módulo Bluetooth HC-05



- Sensor de temperatura y humedad DHT11



- 2 anillos RGB LED de 12 LEDs

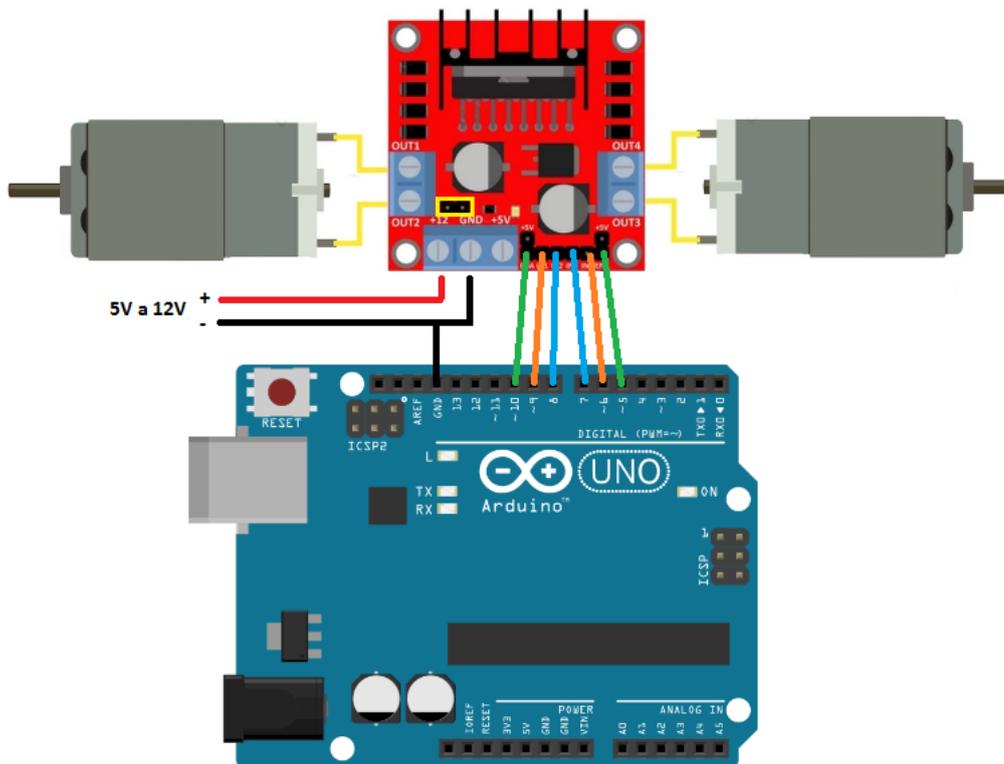


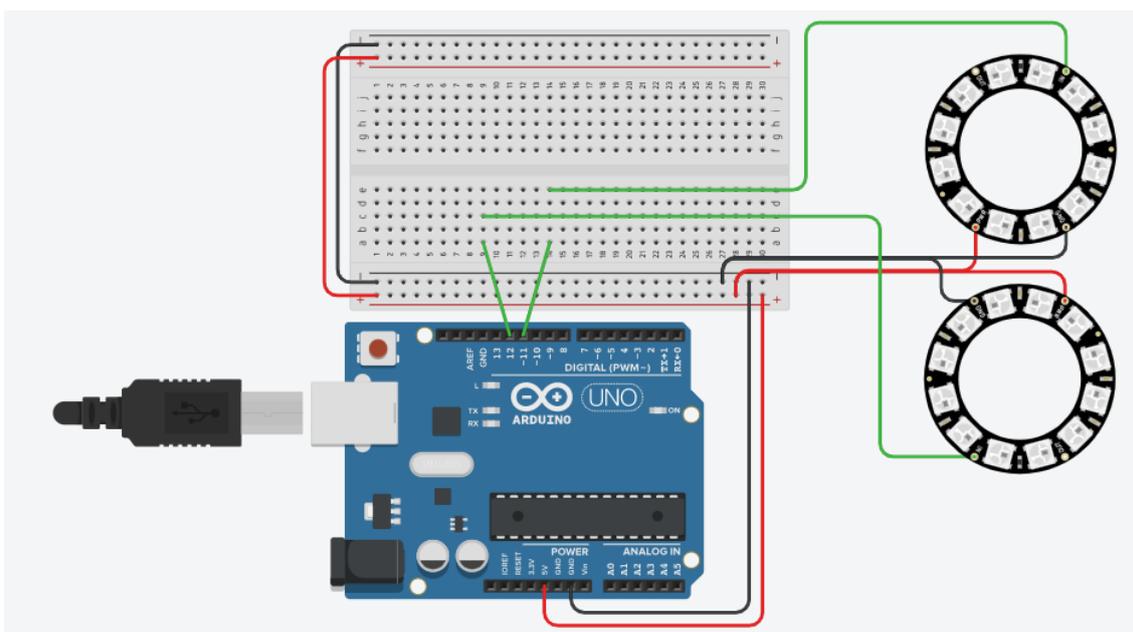
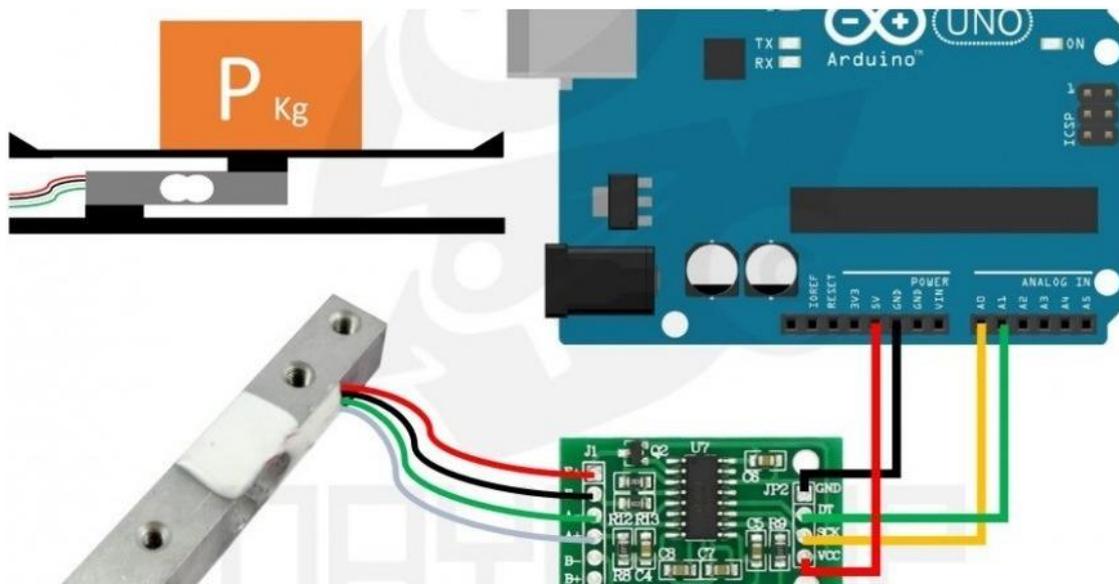
- Polímetro para medición del voltaje de las pilas
- Soldador de estaño



6. Esquemas de montaje

En este apartado indicamos los esquemas de montaje de las partes más complicadas, ya que, algunas como la conexión del módulo DHT11 no son muy complejas.





A la hora de montar todos los componentes, necesitamos usar una protoboard externa para poder conectar varios dispositivos, tanto al pin de 5V, como al de tierra del Arduino.

7. Problemas y soluciones

Problema 1: Colocamos de forma errónea los cables del sensor de temperatura, provocando que se estropeará. Esto se debió a la gran similitud entre varios modelos de sensor de temperatura. Al final, tuvimos que comprar uno nuevo.

Problema 2: La báscula mide el peso únicamente en el extremo en vez de en la totalidad de su superficie. Debido a esta forma de funcionar de la báscula, tuvimos que ingeniárnoslas para conseguir medir el peso creando esta estructura:



Problema 3: Algunos componentes requerían ser soldados y las soldaduras tenían que ser muy precisas para evitar cortocircuitos.

8. Código

```
#include <SoftwareSerial.h>
#include <Adafruit_NeoPixel.h>
#include <elapsedMillis.h>
#include "HX711.h"
#include "DHT.h"

#define PIN_TEMP 11 // PIN del led de la temperatura
#define PIN_PESO 12 // PIN del led de la balanza
// #define PIN_SENSOR_TEMP 13 // PIN del entrada de la temperatura
#define NUMPIXELS 12 // Numero de Diodos del led
#define T_MAX 12 // TEMPERATURA MAXIMA EN GRADOS CELSIUS
```

```

#define T_MIN 0
#define P_MAX 1.0 // PESO MAXIMO EN KG
#define P_MIN 0

SoftwareSerial miBT(2, 3); //Instancia la conexion al bluetooth - PIN 2 a TX y PIN
3 a RX
Adafruit_NeoPixel led_temperatura = Adafruit_NeoPixel(NUMPIXELS, PIN_TEMP,
NEO_GRB + NEO_KHZ800); // Led de la temperatura
Adafruit_NeoPixel led_balanza = Adafruit_NeoPixel(NUMPIXELS, PIN_PESO,
NEO_GRB + NEO_KHZ800); // Led de la balanza

HX711 balanza; // Balanza

//DHT sensor_temperatura(PIN_SENSOR_TEMP, DHT11); // Sensor DHT11

elapsedMillis tiempoPasadoTemperatura;
elapsedMillis tiempoPasadoBalanza;

int temp = 6; // Inicializamos la temperatura a 6 grados
const int DOUT=A1; // CONSTANTE DE LA BALANZA
const int CLK=A0; // CONSTANTE DE LA BALANZA

//Entrada de los motores
int motor1A = 5; // INPUT 1
int motor1B = 6; // INPUT 2
int motor2A = 10; // INPUT 3
int motor2B = 9; // INPUT 4

// Dato recibido atraves de la aplicacion movil
char data = 'S';

void setup()
{
  //Definimos las salidas del arduino
  pinMode(motor1A , OUTPUT);
  pinMode(motor1B , OUTPUT);
  pinMode(motor2A , OUTPUT);
  pinMode(motor2B , OUTPUT);

  Serial.begin(9600); //Inicia la comunicaci3n en el monitor serial a 9600 Baudios
  miBT.begin(9600); // inicialmente la comunicacion serial a 9600 Baudios
  (velocidad de convencional)
  // Inicializamos los sensores y leds
  inicializarTemperatura();
  inicializarBalanza();
  //sensor_temperatura.begin();
  Serial.println("M3dulo conectado");
}

```

```

void loop()
{
  //Esperamos hasta que haya un Stream de datos
  if(miBT.available() > 0){
    data = miBT.read();
    Serial.println(data);
  }
  if(data != 'S'){
    switch (data) {
      case 'F':
        avanzar();    // Forward | AVANZAR
        break;
      case 'B':
        retroceder(); // Backward | RETROCEDER
        break;
      case 'L':
        izquierda(); //Left | IZQUIERDA
        break;
      case 'R':
        derecha(); //Right | DERECHA
        break;
      case 'G':
        avanceIzq(); //Stop | Avance Izq
        break;
      case 'I':
        avanceDrcha(); //Stop | Avance Drcha
        break;
      case 'H':
        atrasIzq(); //Stop | Atras Izq
        break;
      case 'J':
        atrasDrcha(); //Stop | Atras Drcha
        break;
    }
    tiempoPasadoBalanza = 0;
  }
  else{
    detener(); //Stop | DETENER
    if(tiempoPasadoTemperatura > 1500){ //Actualizamos la temperatura de la
nevera cada 1'5 segundos
      actualizarTemperatura();
      tiempoPasadoTemperatura = 0;
    }
    if(tiempoPasadoBalanza > 4500){ // Si el coche no se ha movido durante 4'5
segundos, se actualiza la balanza
      actualizarPeso();
      tiempoPasadoBalanza = 0;
    }
  }
}
}

```

```

    delay(20);
}

void avanzar(){
    // Forward | AVANZAR
    digitalWrite(motor1A, HIGH);
    //digitalWrite(motor1B, LOW);
    digitalWrite(motor2A, HIGH);
    //digitalWrite(motor2B, LOW);

}

void retroceder(){
    // Backward | RETROCEDER
    digitalWrite(motor1A, LOW);
    digitalWrite(motor1B, HIGH);
    digitalWrite(motor2A, LOW);
    digitalWrite(motor2B, HIGH);
}

void derecha(){
    //Right | DERECHA
    digitalWrite(motor1A, HIGH);
    digitalWrite(motor1B, LOW);
    digitalWrite(motor2A, LOW);
    digitalWrite(motor2B, HIGH);
}

void izquierda(){
    //Left | IZQUIERDA
    digitalWrite(motor1A, LOW);
    digitalWrite(motor1B, HIGH);
    digitalWrite(motor2A, HIGH);
    digitalWrite(motor2B, LOW);
}

void detener(){
    //Stop | DETENER
    digitalWrite(motor1A, LOW);
    digitalWrite(motor1B, LOW);
    digitalWrite(motor2A, LOW);
    digitalWrite(motor2B, LOW);
}

void avanceIzq(){
    //Stop | avance Izq
    digitalWrite(motor1A, 100);
    digitalWrite(motor1B, LOW);
    digitalWrite(motor2A, HIGH);
}

```

```

    digitalWrite(motor2B, LOW);
}

void avanceDrcha(){
    //Stop | avance Drcha
    digitalWrite(motor1A, HIGH);
    digitalWrite(motor1B, LOW);
    digitalWrite(motor2A, 100);
    digitalWrite(motor2B, LOW);
}

void atrasIzq(){
    //Stop | atras Izq
    digitalWrite(motor1A, LOW);
    digitalWrite(motor1B, HIGH);
    digitalWrite(motor2A, LOW);
    digitalWrite(motor2B, 100);
}

void atrasDrcha(){
    //Stop | atras Drcha
    digitalWrite(motor1A, LOW);
    digitalWrite(motor1B, 100);
    digitalWrite(motor2A, LOW);
    digitalWrite(motor2B, HIGH);
}
//CODIGO DEL CONTROL DE LA TEMPERATURA//

void inicializarTemperatura(){
    led_temperatura.begin(); //Encendemos el led de la temperatura. Habra que
encender tambien el de la balanza
    led_temperatura.setBrightness(10); // Predeterminado a 50, pero es mucho
ajustarColoresTemperatura(6); // Inicializamos el led
}

int obtenerTemperatura(){
    // Leer la temperatura conlleva unos 250 milisegundos
    //float temp = sensor_temperatura.readHumidity();

    if(temp == T_MAX){
        temp--;
        return temp;
    }
    else if(temp== T_MIN){
        temp++;
        return temp;
    }
    else if((int)random(0,2) == 0){
        temp--;

```

```

    return temp;
}else{
    temp++;
    return temp;
}
}

void actualizarTemperatura(){
    switch(obtenerTemperatura()){
        case 0:
            ajustarColoresTemperatura(12);
            break;
        case 1:
            ajustarColoresTemperatura(11);
            break;
        case 2:
            ajustarColoresTemperatura(10);
            break;
        case 3:
            ajustarColoresTemperatura(9);
            break;
        case 4:
            ajustarColoresTemperatura(8);
            break;
        case 5:
            ajustarColoresTemperatura(7);
            break;
        case 6:
            ajustarColoresTemperatura(6);
            break;
        case 7:
            ajustarColoresTemperatura(5);
            break;
        case 8:
            ajustarColoresTemperatura(4);
            break;
        case 9:
            ajustarColoresTemperatura(3);
            break;
        case 10:
            ajustarColoresTemperatura(2);
            break;
        case 11:
            ajustarColoresTemperatura(1);
            break;
        case 12:
            ajustarColoresTemperatura(0);
    }
}
}

```

```

void ajustarColoresTemperatura(int numPixeles){
  for(int i = 0; i < numPixeles; i++){
    led_temperatura.setPixelColor(i, led_temperatura.Color(0, 0, 255));
  }
  for(int i = numPixeles; i < NUMPIXELS; i++){
    led_temperatura.setPixelColor(i, led_temperatura.Color(255, 0, 0));
  }
  led_temperatura.show();          // Actualiza el led para que coincida con el
  valor de la temperatura
}

```

// CODIGO DE LA BALANZA //

```

void inicializarBalanza(){
  balanza.begin(DOUT, CLK);
  Serial.print("Lectura del valor del ADC: ");
  Serial.println(balanza.read());
  Serial.println("No ponga ningun objeto sobre la balanza");
  Serial.println("Destarando...");
  Serial.println("...");
  balanza.set_scale(439430.25); // Establecemos la escala
  balanza.tare(20); //El peso actual es considerado Tara
  led_balanza.begin(); //Encendemos el led de la temperatura. Habra que encender
  tambien el de la balanza
  led_balanza.setBrightness(5); // Predeterminado a 50, pero es excesivo
  ajustarColoresBalanza(0); // Inicializamos el led a Blanco
  Serial.println("Listo para pesar");
}

```

```

float medirPeso(){
  return balanza.get_units(20);
}

```

```

void actualizarPeso(){
  float peso = medirPeso();
  Serial.println(peso);
  if((0 * 0.0833) <= peso <= (0.0833))
    ajustarColoresBalanza(1);
  else if((0.0833) <= peso <= (2*0.0833))
    ajustarColoresBalanza(2);
  else if((2*0.0833) <= peso <= (3*0.0833))
    ajustarColoresBalanza(3);
  else if((3*0.0833) <= peso <= (4*0.0833))
    ajustarColoresBalanza(4);
  else if((4*0.0833) <= peso <= (5*0.0833))
    ajustarColoresBalanza(5);
  else if((5*0.0833) <= peso <= (6*0.0833))
    ajustarColoresBalanza(6);
}

```

```

else if((6*0.0833) &lt;= peso &lt;= (7*0.0833))
    ajustarColoresBalanza(7);
else if((7*0.0833) &lt;= peso &lt;= (8*0.0833))
    ajustarColoresBalanza(8);
else if((8*0.0833) &lt;= peso &lt;= (9*0.0833))
    ajustarColoresBalanza(9);
else if((9*0.0833) &lt;= peso &lt;= (10*0.0833))
    ajustarColoresBalanza(10);
else if((10*0.0833) &lt;= peso &lt;= (11*0.0833))
    ajustarColoresBalanza(11);
else if(peso >= (12*0.0833))
    ajustarColoresBalanza(12);
else
    ajustarColoresBalanza(0);
}

void ajustarColoresBalanza(int numPixeles){
    int i = 0;
    for(i; i &lt; numPixeles; i++){
        if(i &lt;= 5)
            led_balanza.setPixelColor(i, led_balanza.Color(0, 255, 0)); // VERDE
        else if(i &lt;= 9)
            led_balanza.setPixelColor(i, led_balanza.Color(255, 255, 0)); // AMARILLO
        else
            led_balanza.setPixelColor(i, led_balanza.Color(255, 0, 0)); // ROJO
    }
    for(i; i &lt; NUMPIXELS; i++)
        led_balanza.setPixelColor(i, led_balanza.Color(255, 255, 255)); // BLANCO
    led_balanza.show(); // Actualiza el led para que coincida con el valor
de la temperatura
}

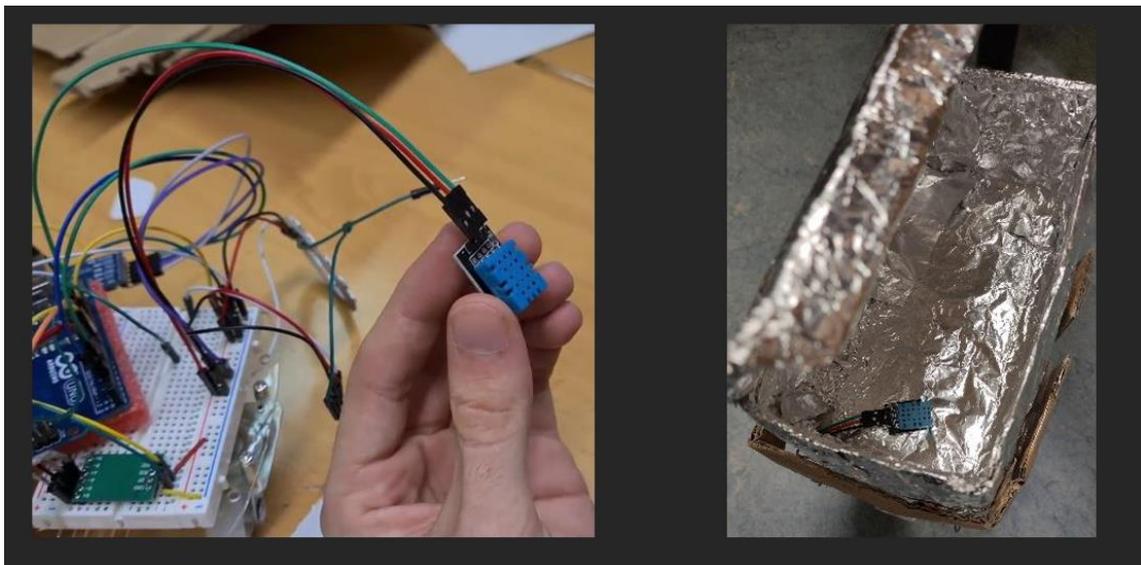
```

9. Fotos sobre el hardware

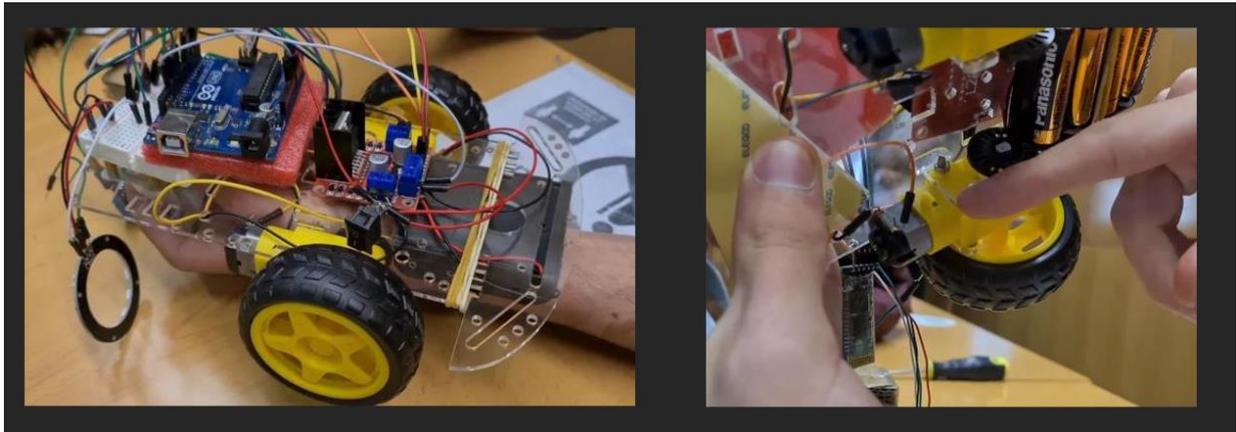
BALANZA



SENSOR DE TEMPERATURA



MOTORES, CONTROLADOR DE MOTORES Y PORTAPILAS



10. Vídeo explicativo del proyecto

A través de este enlace se podrá ver el vídeo en el que se explican el hardware y el código: <https://youtu.be/B4rwYZenjal>

