

4. Testeando los sensores y LEDs

¿Como leer un sensor analógico?

Para leer un sensor analógico con Arduino, debes seguir los siguientes pasos:

1. Conectar el sensor: Conecta el sensor analógico a un pin analógico en la placa Arduino. Asegúrate de que los pines del sensor estén conectados correctamente al pin analógico de la placa. Además, algunos sensores pueden requerir una conexión adicional de VCC y GND.
2. Leer el valor: Se utiliza la función `analogRead()` para leer el valor del pin analógico. Esta función devuelve un valor entre 0 y 1023 que representa el voltaje en el pin. El valor 0 representa 0 voltios, mientras que el valor 1023 representa la referencia de voltaje de la placa, que en ESP8266 es de 3.3V.
3. Convertir el valor: A veces, el valor devuelto por la función `analogRead()` no es el valor que se necesita para el cálculo. Por ejemplo, algunos sensores pueden requerir una conversión de voltaje a temperatura. En este caso, se puede usar una ecuación de conversión para convertir el valor leído a la unidad de medida necesaria.

Aquí hay un ejemplo de código para leer un sensor analógico conectado al pin A0:

```
int sensorValue = 0; // Variable para almacenar el valor del sensor

void setup() {
  Serial.begin(115200); // Inicia la comunicación serial a una velocidad de 115200 baudios
}

void loop() {
  sensorValue = analogRead(A0); // Lee el valor del sensor en el pin A0
  Serial.println(sensorValue); // Imprime el valor en el monitor serial
  delay(100); // Espera un tiempo para evitar la sobrecarga del puerto serie
}
```

En este ejemplo, la función `analogRead()` se usa para leer el valor del pin A0 y luego se imprime en el monitor serial. El comando `delay()` se utiliza para esperar un tiempo antes de leer el valor del sensor de nuevo y enviarlo al monitor serial para evitar la sobrecarga del puerto serial.

¿Como encender un led con Arduino?

Para encender un LED con Arduino, sigue los siguientes pasos:

1. Conecta un extremo de la resistencia a un pin digital del Arduino y el otro extremo a la protoboard.
2. Conecta el cátodo (pata más corta) del LED a la misma fila de la protoboard en la que conectaste la resistencia.
3. Conecta el ánodo (pata más larga) del LED a otra fila de la protoboard.
4. Conecta un cable desde la fila de la protoboard donde está el ánodo del LED a una fuente de voltaje (por ejemplo, 3.3V).
5. Conecta un cable desde la fila de la protoboard donde está la resistencia a un pin GND del Arduino.
6. En el código de Arduino, usa la función `pinMode` para establecer el pin digital donde conectaste la resistencia como una salida: `pinMode(PIN, OUTPUT);`.
7. En el código de Arduino, usa la función `digitalWrite` para establecer el estado del pin digital donde conectaste la resistencia a HIGH o LOW para encender o apagar el LED, respectivamente: `digitalWrite(PIN, HIGH);`.

Por ejemplo, el siguiente código encendería y apagaría un LED conectado al pin digital D2:

```
void setup() {
  pinMode(D2, OUTPUT); //establecer el pin 13 como salida
}

void loop() {
  digitalWrite(D2, HIGH); //encender el LED
  delay(1000); //esperar un segundo
  digitalWrite(D2, LOW); //apagar el LED
  delay(1000); //esperar un segundo
}
```

¿Como leer el sensor DHT11 con la librería Adafruit y Arduino?

Para leer el sensor de temperatura y humedad DHT11 con la librería de Adafruit y Arduino, debes seguir los siguientes pasos:

1. Descarga e instala la librería de Adafruit DHT desde el administrador de librerías de Arduino. Puedes hacerlo yendo al menú "Sketch -> Incluir librería -> Administrar librerías" y buscando "DHT".
2. Conecta el pin de datos del sensor DHT11 a uno de los pines digitales de la placa Arduino. En el ejemplo a continuación, se utiliza el pin D7.
3. En el código, primero debes incluir las librerías necesarias:

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
```

4. Luego, define el pin de datos y el tipo de sensor que estás utilizando:

```
#define DHTPIN D7 // Pin digital conectado al sensor DHT
#define DHTTYPE DHT11 // DHT 11

DHT_Unified dht(DHTPIN, DHTTYPE);
```

5. En la función `setup()`, inicializa el dispositivo DHT y realiza una impresión por consola de los detalles del sensor:

```
void setup() {
  dht.begin(); // Inicializar dispositivo DHT
  Serial.begin(115200);
  Serial.println(F("Sensor DHT11"));
  // Imprime los detalles del sensor de temperatura.
  sensor_t sensor;
  dht.temperature().getSensor(&sensor);
  Serial.println(F("-----"));
  Serial.println(F("Sensor de Temperatura"));
  Serial.print (F("Valor Max: ")); Serial.print(sensor.max_value);
  Serial.println(F("°C"));
  Serial.print (F("Valor Min: ")); Serial.print(sensor.min_value);
  Serial.println(F("°C"));
  Serial.print (F("Resolucion: ")); Serial.print(sensor.resolution);
  Serial.println(F("°C"));
  // Imprime los detalles del sensor de humedad.
  dht.humidity().getSensor(&sensor);
  Serial.println(F("Sensor de Humedad"));
  Serial.print (F("Valor Max: ")); Serial.print(sensor.max_value);
  Serial.println(F("%"));
  Serial.print (F("Valor Min: ")); Serial.print(sensor.min_value);
  Serial.println(F("%"));
  Serial.print (F("Resolucion: ")); Serial.print(sensor.resolution);
  Serial.println(F("%"));
}
```

```
Serial.println(F("-----"));  
}
```

6. En la función `loop()`, puedes obtener los valores de temperatura y humedad del sensor DHT11 y mostrarlos por consola:

```
void loop() {  
  sensors_event_t event;  
  dht.temperature().getEvent(&event);  
  if (isnan(event.temperature)) {  
    Serial.println(F("¡Error al leer la temperatura!"));  
  }  
  else {  
    Serial.print(F("Temperatura: "));  
    Serial.print(event.temperature);  
    Serial.println(F("°C"));  
  }  
  dht.humidity().getEvent(&event);  
  if (isnan(event.relative_humidity)) {  
    Serial.println(F("¡Error al leer la humedad!"));  
  }  
  else {  
    Serial.print(F("Humedad: "));  
    Serial.print(event.relative_humidity);  
    Serial.println(F("%"));  
  }  
  delay(2000);  
}
```

¡Ahora es tu turno! Realiza el código completo con los siguientes ingredientes hardware para que todo el sistema funcione.

El programa deberá utilizar un sensor DHT11 para medir la temperatura y humedad del ambiente y un sensor LDR para medir la intensidad de luz. También incluir dos LEDs que parpadean de manera alterna cada segundo.

En la función `setup()`, se inicializan los puertos de entrada y salida para los sensores y los LEDs, se imprime información detallada acerca del sensor DHT11 y se configura la velocidad de comunicación del puerto serie a 115200 bauds.

En la función `loop()`, se leen los valores del sensor LDR y se imprimen en el monitor serie junto con los valores de temperatura y humedad del sensor DHT11, si están disponibles. Los LEDs parpadean

alternadamente cada segundo.

Código Completo

Solución

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN D7 // Pin digital conectado al sensor DHT
#define DHTTYPE DHT11 // DHT 11
DHT_Unified dht(DHTPIN, DHTTYPE);

#define LED_ROJO D5
#define LED_VERDE D6

#define LDR_PIN A0
int sensorLDRValue = 0; // variable para almacenar el valor proveniente del sensor LDR

// la función de setup() se ejecuta una vez cuando presionas el botón de reset o enciendes
// la placa
void setup() {

  Serial.begin(115200); // inicializa el puerto serie (USB - COM) en una velocidad
// de comunicación de 115200 bauds
  Serial.println("\n\n");

  pinMode(LED_ROJO, OUTPUT); // inicializa el pin digital LED_ROJO como salida.
  pinMode(LED_VERDE, OUTPUT); // inicializa el pin digital LED_VERDE como salida.

  dht.begin(); // Inicializar dispositivo DHT
  Serial.println(F("Sensor DHT11"));
// Imprime los detalles del sensor de temperatura.
  sensor_t sensor;
  dht.temperature().getSensor(&sensor);
```

```

Serial.println(F("-----"));
Serial.println(F("Sensor de Temperatura"));
Serial.print (F("Valor Max:  ")); Serial.print(sensor.max_value);
Serial.println(F("°C"));
Serial.print (F("Valor Min:  ")); Serial.print(sensor.min_value);
Serial.println(F("°C"));
Serial.print (F("Resolucion:  ")); Serial.print(sensor.resolution);
Serial.println(F("°C"));
// Imprime los detalles del sensor de humedad.
dht.humidity().getSensor(&sensor);
Serial.println(F("Sensor de Humedad"));
Serial.print (F("Valor Max:  ")); Serial.print(sensor.max_value);
Serial.println(F("%"));
Serial.print (F("Valor Min:  ")); Serial.print(sensor.min_value);
Serial.println(F("%"));
Serial.print (F("Resolucion:  ")); Serial.print(sensor.resolution);
Serial.println(F("%"));
Serial.println(F("-----"));
}

// la función de loop() se ejecuta una y otra vez para siempre
void loop() {
digitalWrite(LED_ROJO, HIGH); // enciende el LED (HIGH es el nivel alto de voltaje -
3.3V)
digitalWrite(LED_VERDE, LOW); // apaga el LED haciendo que el voltaje sea bajo (LOW)
delay(1000); // espera un segundo - 1000 ms (milisegundos)
digitalWrite(LED_ROJO, LOW); // apaga el LED haciendo que el voltaje sea bajo (LOW)
digitalWrite(LED_VERDE, HIGH); // enciende el LED (HIGH es el nivel alto de voltaje -
3.3V)
delay(1000); // espera un segundo - 1000 ms (milisegundos)

// read the value from the sensor:
sensorLDRValue = analogRead(LDR_PIN);
Serial.print(F("LDR:  "));
Serial.print(sensorLDRValue);

mostrar_datos_sensor_DHT11();

Serial.println();

```

```
}

void mostrar_datos_sensor_DHT11(){
    // Obtenga si hay evento de temperatura e imprima su valor.
    sensors_event_t event;
    dht.temperature().getEvent(&event);
    if (isnan(event.temperature)) {
        Serial.print(F("\t"));
        //Serial.println(F("; Error al leer la temperatura!"));
    }
    else {
        Serial.print(F("\t"));
        Serial.print(F("Temperatura: "));
        Serial.print(event.temperature);
        Serial.print(F("°C"));
    }
    // Obtenga si hay evento de humedad e imprima su valor.
    dht.humidity().getEvent(&event);
    if (isnan(event.relative_humidity)) {
        Serial.print(F("\t"));
        //Serial.println(F("; Error al leer la humedad!"));
    }
    else {
        Serial.print(F("\t"));
        Serial.print(F("Humedad: "));
        Serial.print(event.relative_humidity);
        Serial.print(F("%"));
    }
}
}
```

[Descargar el código de ejemplo completo](#)

Revision #13

Created 29 April 2023 17:46:29 by Rafa

Updated 11 May 2023 16:56:45 by Rafa