INTRODUCTION To IoT

Spring 2023 LAB2

Introduction:

In this lab session, you will be interfacing a DHT11 temperature-humidity sensor with the ESP32 and write the outputs to the serial monitor. Once that is done, you will be working on setting a threshold value to make decisions to drive a servo motor.

Part-1) DHT11 Temperature & Humidity sensor on ESP32 using Arduino IDE

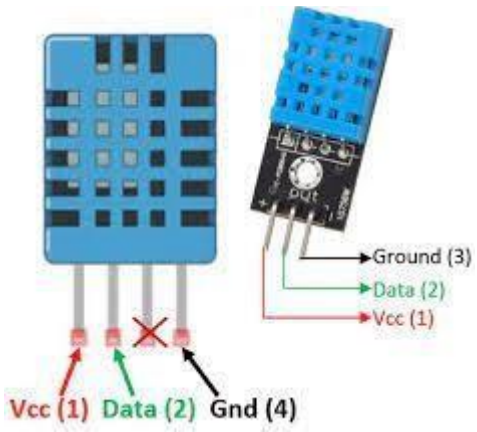
a) Introduction and understanding the DHT11 sensor:

i. The DHT11 is a commonly used Temperature and humidity sensor. The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data.

ii. The DHT11 is chosen because it is lab calibrated, accurate and stable and its signal output is digital.

iii. Most important of all, it is relatively inexpensive for the given performance.

iv. Below is the pinout diagram of the DHT11 sensor



v. DHT11 Specifications:

1) Operating Voltage: 3.5V to 5.5V
2) Operating current: 0.3mA (measuring) 60uA (standby)
3) Output: Serial data
4) Temperature Range: 0°C to 50°C
5) Humidity Range: 20% to 90%
6) Resolution: Temperature and Humidity both are 16-bit
7) Accuracy: ±1°C and ±1%

vi. Please refer to the data sheet for more information about the DHT11 sensor. Link to the data sheet here.

b) Components required for this exercise:

i) DHT11 sensor
ii) ESP32

iii) Breadboard
iv) Few jumper wires

c) Circuit Diagram:

i. Data pin of DHT11 is connected to any digital GPIO pin of the ESP32.
ii. The +VCC of DHT11 goes into +3.3V of the ESP32. iii. GND of the DHT11 goes into Ground Pin (GND) of the ESP32.

d) Library Files:

Following two libraries will be required to run this code. Download the zip file, extract the same and copy this to your Arduino library folder.

i) Library 1: You may download library file from here. ii)
Library 2: You may download library file from here.

e) Pseudocode:

i. Include the library of the DHT11 temperature and humidity sensor using #include "DHT.h"
ii. Define the DHTtype and digital pin, in this case type is DHT11 and digital pin(any digital pin) using #define DHTTYPE and #define dht_dpin.
iii. Instantiate the DHT dht(dht_dpin, DHTTYPE)
iv. In the setup block, use dht.begin(), Serial.begin() to begin the serial communication. Use delay() to give an appropriate delay.
v. In the loop block, use float variables to read the temperature and humidity values from the sensor using dht.readHumidity(), dht.readTemperature(); Serial.print()/Serial.println() to print the values on the serial monitor and give an appropriate delay using delay() according to the frequency of sensor. vi. You can refer to the example codes in the library you already downloaded for the syntaxes as well as some additional points.

f) Expected output:

i. The output values should be the temperature and humidity values measured by the DHT11 sensor at each instant it is sensing corresponding to your delay.
ii. After getting the values plot the mean of these values every minute on the serial plotter
iii. Compare your values with the room temperature and verify if the output temperature value is not too far from it, else your sensor is not working.
iv. Try playing around with the sensor a little like blowing on to it etc. which should increase the temperature and verify.

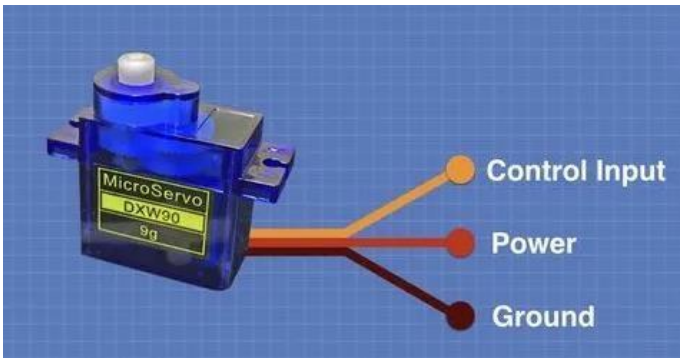
Part-2) Controlling the servo motor by setting threshold:

a) Introduction and understanding the servo motor:

i. Servo motors are geared DC motors that have an integrated servomechanism with a feedback loop to allow precise positioning of the motor shaft. A high gear ratio allows a small servo to have an impressive torque rating.

ii. Most servos are limited in rotation to either 180 or 270 degrees, with 180-degree servo motors being more common. There are specially modified servo motors that can rotate beyond 360-degrees, but we won't be working with those today.

iii. Servo motors come in a wide range of sizes and can be controlled either with an analog PWM signal or with a digital I/O signal. The inexpensive servos we use for hobbyist applications are usually analog servo motors.



b) Components required for this exercise:

i. DHT11 sensor

ii. ESP32

iii. Servo motor iv. Breadboard

v. Few jumper wires

c) Circuit Diagram:

i. The control pin of servo is connected to any pwm pin of the ESP32.

ii. The +VCC of servo goes into +3.3V of the ESP32. iii. GND of the servo goes into Ground Pin (GND) of the ESP32.

d) Library Files:

The ESP32 Arduino Servo Library makes it easier to control a servo motor with your ESP32, using the Arduino IDE. Follow the next steps to install the library in your Arduino IDE:

1. Click here to download the ESP32_Arduino_Servo_Library. You should have a .zip folder in your Downloads folder
2. Unzip the .zip folder and you should get *ESP32-Arduino-Servo-Library-Master* folder
3. Rename your folder from ~~*ESP32-Arduino-Servo-LibraryMaster*~~ to *ESP32_Arduino_Servo_Library*
4. Move the *ESP32_Arduino_Servo_Library* folder to your Arduino IDE installation libraries folder
5. Finally, re-open your Arduino IDE

e) Pseudocode:

i. Include the library of the servo motor using #include "Servo.h"

ii. Create a servo object and declared the initial angle of the servo motor. iii. Adjust the baud rate in the setup function Serial.begin().

iv. Define the pin at which the servo motor is connected to the ESP32

v. Inside the main loop, write the condition for setting thresholds for temperature and humidity values (the threshold values depend on your respective environmental conditions)

vi. Inside the above loop apply the condition in between 0 and 180 degrees, the servo motor's angle need to be changed. (The change in angle of servo is up to you)

vii. You can refer to the example codes in the library you already downloaded for the syntaxes as well as some additional points.

f) Expected output:

i. The output values should be the temperature and humidity values measured by the DHT11 sensor at each instant it is sensing corresponding to your delay.

ii. If the temperature and humidity values meet the thresholds, then the servo motor angle should be changed.

iii. After that add at least one more threshold on the values and map the servo hand to different angles.