Indian Institute of Technology, Jodhpur

Lab Manual

Sensors and IoT

LAB - 2

Date: 9 Sept, 2024

Lab Objective

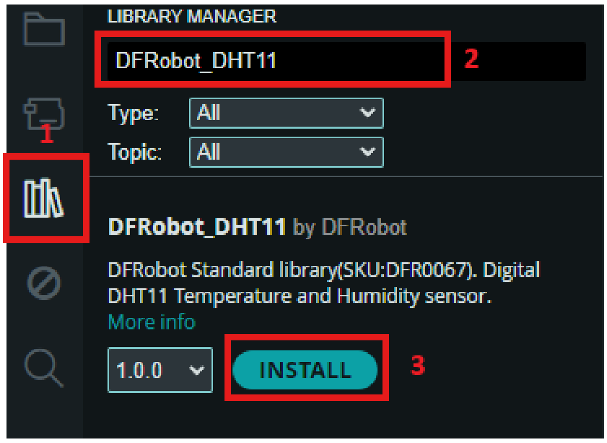
In this lab we will interface with a sensor and explore the support of updating NodeMCU remotely as if it was deployed in an integrated system

Part 1 : Interfacing with sensor using NodeMCU

Using a micro controller allows us to perform interfacing with various devices using the on board pins which can have multiple functionalities mapped to them such as Digital I/O, ADC, DAC etc. on top these pins there are various protocols which are implemented such as I2C, SPI, UART etc.

In this lab we will interface with the DHT11 sensor which provides a readout for environment temperature and humidity. In this case DHT11 uses a proprietary, single-wire communication protocol that's based on timing. Hence we will be using DHT11 library to interface with it.

1. Open the provided DHT11_interfacing.ino file.
2. Open library manager and search for DFRobot_DHT11



DFRobot_DHT11

Type: All

Topic: All

DFRobot_DHT11 by DFRobot

DFRobot Standard library(SKU:DFR0067). Digital DHT11 Temperature and Humidity sensor.

1.0.0

Figure 1: Install DHT11 library

3. Make Connections with the DHT11 using female to female jumper wires as per the following diagram :-

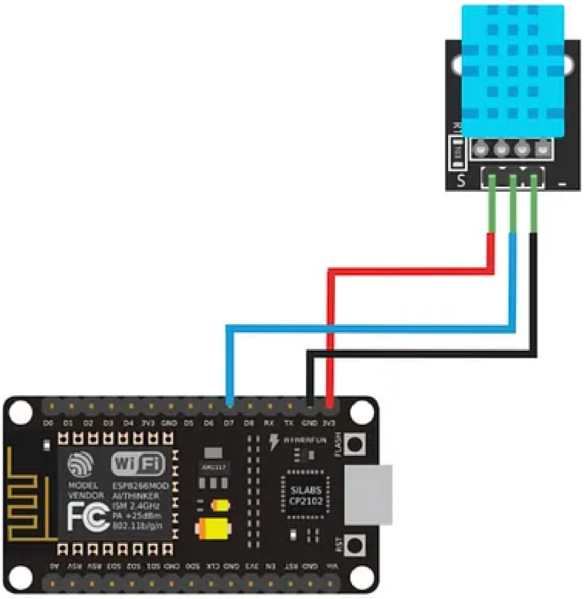
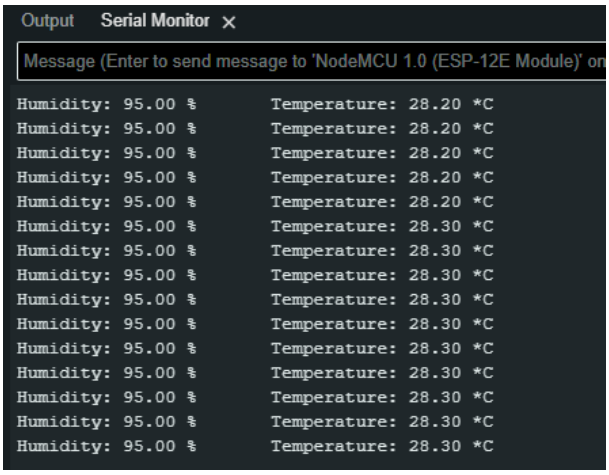


Figure 2: Connect the DATA pin to D7
Connect the vcc pin to 3V
Connect the GND pin to G

4. Click upload to program the NodeMCU



Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on

```
Humidity: 95.00 %       Temperature: 28.20 *C
Humidity: 95.00 %       Temperature: 28.20 *C
Humidity: 95.00 %       Temperature: 28.20 *C
Humidity: 95.00 %       Temperature: 28.20 *C
Humidity: 95.00 %       Temperature: 28.20 *C
Humidity: 95.00 %       Temperature: 28.30 *C
Humidity: 95.00 %       Temperature: 28.30 *C
Humidity: 95.00 %       Temperature: 28.30 *C
Humidity: 95.00 %       Temperature: 28.30 *C
Humidity: 95.00 %       Temperature: 28.30 *C
Humidity: 95.00 %       Temperature: 28.30 *C
Humidity: 95.00 %       Temperature: 28.30 *C
Humidity: 95.00 %       Temperature: 28.30 *C
Humidity: 95.00 %       Temperature: 28.30 *C
Humidity: 95.00 %       Temperature: 28.30 *C
```

Figure 3: DHT11 Data output

Part 2 : Using OTA to upadte the code

Node MCU has the functionality to update its own firmware using Wi-Fi. This is desired when the possibility to connect to the NodeMCU using wire is not possible

1. Open ArduinoIDE and navigate to File - Examples - ArduinoOTA - BasicOTA

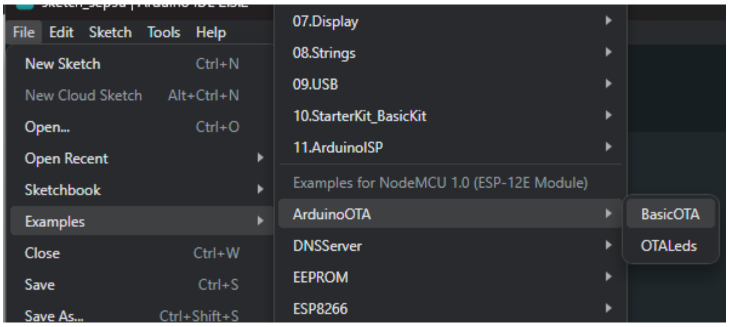
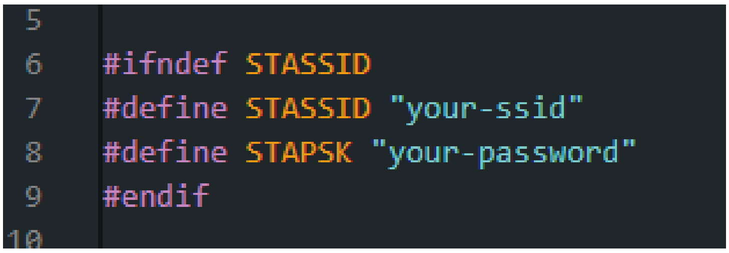


Figure 4: Open OTA example

2. Edit the Wi-Fi configuration as per your system

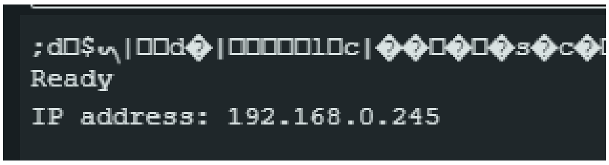


```
#ifndef STASSID
#define STASSID "your-ssid"
#define STAPSK "your-password"
#endif
```

Figure 5: Edit Wi-Fi Configuration

3. Upload the code to enable OTA update on nodeMCU

4. Check the output on Serial Console to verify the uplaod



```
;d□$ჷ|□□d�|□□□□□1□c|��□�□�s�c�
Ready
IP address: 192.168.0.245
```

Figure 6: Ensuring OTA is working

5. Once the output is verified, disconnect the NodeMCU from your system and connect it to your power bank / external power source.

6. Ensure that your Wi-Fi hotspot is on.

7. After a while a new entry should be visible in the devices drop down menu. NOTE:If NodeMCU does not show up, try to use an alternate Wi-Fi network by updating it in the NodeMCU code and connecting your system to the same network.

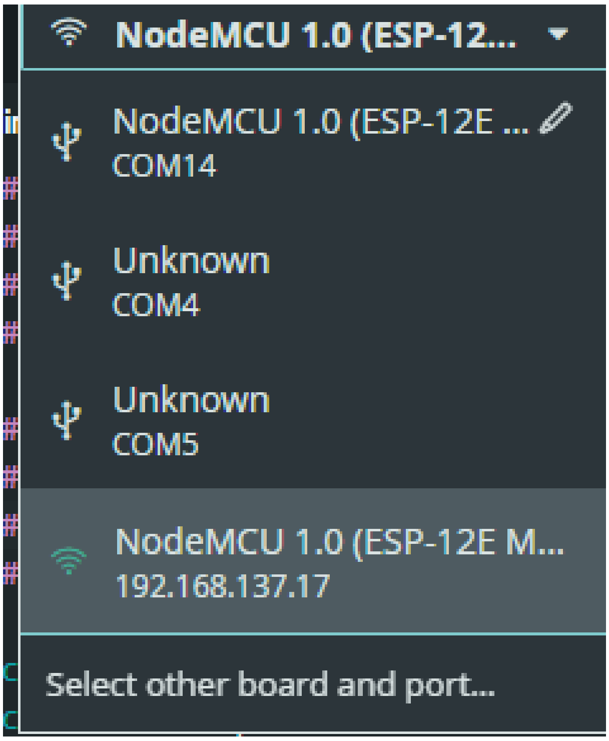
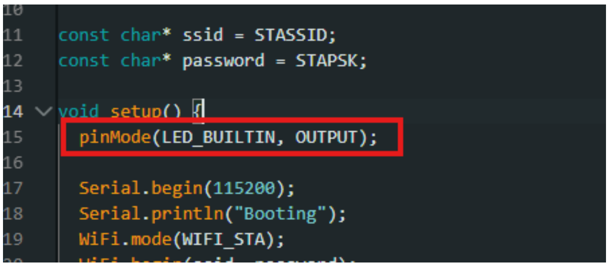


Figure 7: Select the NodeMCU with wireless symbol

8. Once the NodeMCU is visible modify the code by adding the blink logic to the corresponding code blocks

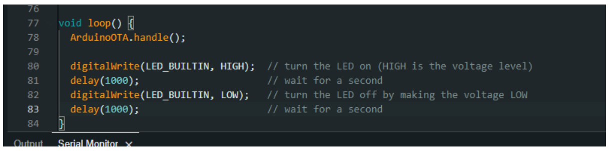


```
const char* ssid = STASSID;
const char* password = STAPSK;

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);

  Serial.begin(115200);
  Serial.println("Booting");
  WiFi.mode(WIFI_STA);
```

Figure 8: Edit the setup function

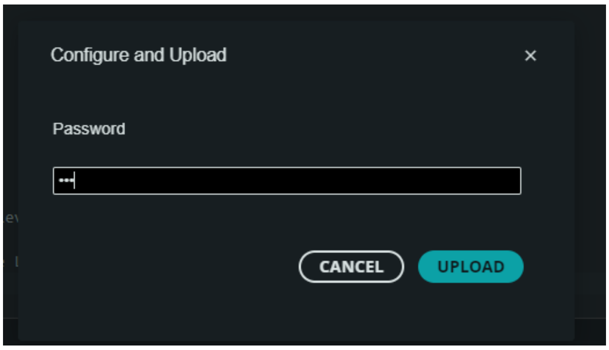


```
void loop() {
  ArduinoOTA.handle();

  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```

Figure 9: edit the loop function

9. Once done click upload. It will ask for password enter ”123” as password and click upload



Configure and Upload

Password

•••

Figure 10: Upload Dialog

10. Once done NodeMCU’s on board LED should start blinking.

Assignment : Fetch data from DHT11 sensor and upload it to ThingSpeak

- As part of lab submission take the data from the DHT11 sensor and upload the data to Thingspeak.
- The code has to be updated using OTA and not via direct connection using USB.
- Crete a new channel as YOURNAME_ROLLNUMBER_DHT11
- Upload Temperature and Humidity and the random number as part of 3 different graphs.
- The number within the random function needs to be the product of your admission year and the numeric part of your roll number
  - EG - X22XXX002, then the input to your function should be 22 X 002 = 44
  - If your roll number creates an output of less than 20 use 100 as the input number of your function
- Create the report using overleaf with the help of provided template.
- For any help please refer to the references section of this document.
- Submit a report consisting of LAB examples as well as assignment before the due date.
- Plagiarism will be checked and defaulters will be penalized.