In the Name of God

Faculty of Computer Engineering

Course: IoT Device Programming

Final Project

Prepared by:

Mohammad Amin Saberi, Majid Soltanian Tiranchi, Mohammad Amin Naqian

Winter 2022

Table of Contents

Introduction

This report provides details on the final project for the IoT Device Programming course. The project will be delivered virtually.

Question Response Status

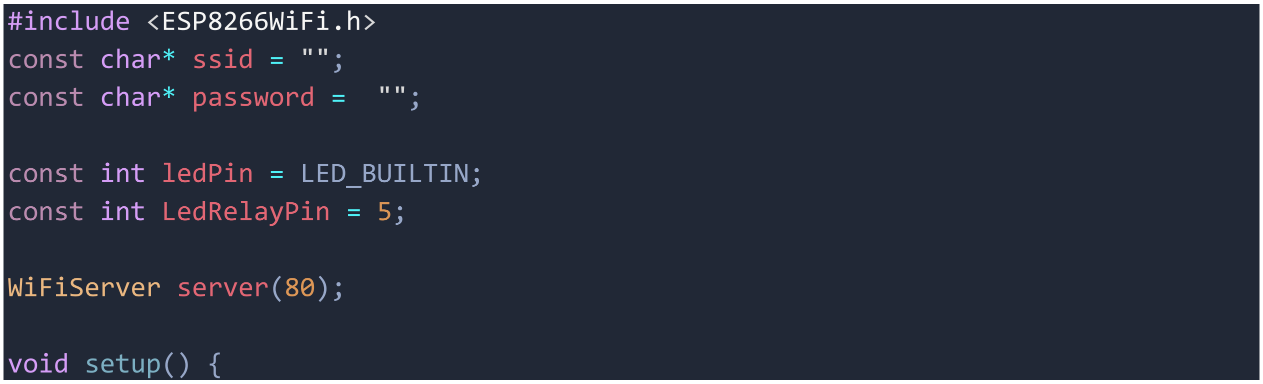
Complete the second column of the table below. Indicate "Completed," "Not Completed," or "Partially Completed" for each question. If the status is "Partially Completed," explain the reason in the "Comments" column.

Question No.	Status	Comments
Question 1	Completed	-
Question 2	Completed	-
Question 3	Completed	-
Question 4	Completed	-
Question 5	Completed	-
Question 6	Completed	-

Explanation of Question 1 Response

Write the changes made to the code and any potential issues encountered here.

Minimal changes were made to the code. The SSID (Wi-Fi name) and password were modified for Wi-Fi connectivity. Below is the updated code:



```
#include <ESP8266WiFi.h>
const char* ssid = "";
const char* password =  "";

const int ledPin = LED_BUILTIN;
const int LedRelayPin = 5;

WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);

  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);

  pinMode(LedRelayPin, OUTPUT);
  digitalWrite(LedRelayPin, LOW);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
    digitalWrite(ledPin, !digitalRead(ledPin));
  }
  Serial.println("");
  Serial.println("WiFi connected");
  // Start the server
  server.begin();
  Serial.println("Server started");
  // Print the IP address
  Serial.print("Use this URL : ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");
}
void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }
  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }
  // Read the first line of the request
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();
  // Match the request
  int value = LOW;

  if((request.indexOf("/LED=ON") != -1)&&(request.indexOf("/LED=OFF") != -1)){
    Serial.println("Error");
  }else{
    if (request.indexOf("/LED=ON") != -1) {
      digitalWrite(ledPin, LOW);
      digitalWrite(LedRelayPin, LOW);
      value = LOW;
    }
    if (request.indexOf("/LED=OFF") != -1){
      digitalWrite(ledPin, HIGH);
      digitalWrite(LedRelayPin, HIGH);
      value = HIGH ;
    }
    // Return the response
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println(""); // do not forget this one
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    client.print("Led pin is now: ");
    if(value == HIGH) {
      client.print("Off");
    } else {
      client.print("On");
    }
    client.println("<br><br>");
    client.println("Click <a href=\"/LED=ON\">here</a> turn the LED on pin 5
ON<br>");
    client.println("Click <a href=\"/LED=OFF\">here</a> turn the LED on pin 5
OFF<br>");
    client.println("</html>");
    delay(1);
    Serial.println("Client disconnected");
    Serial.println("");
  }

}
```

Explanation of Question 2 Response

Explain the method used for internet communication and any potential issues encountered here.

A host and domain were set up, where the host operated on Linux with a CPanel control panel. A .ir domain was purchased, and DNS settings were configured to link the host and domain. This process was facilitated by the hosting provider's support team.

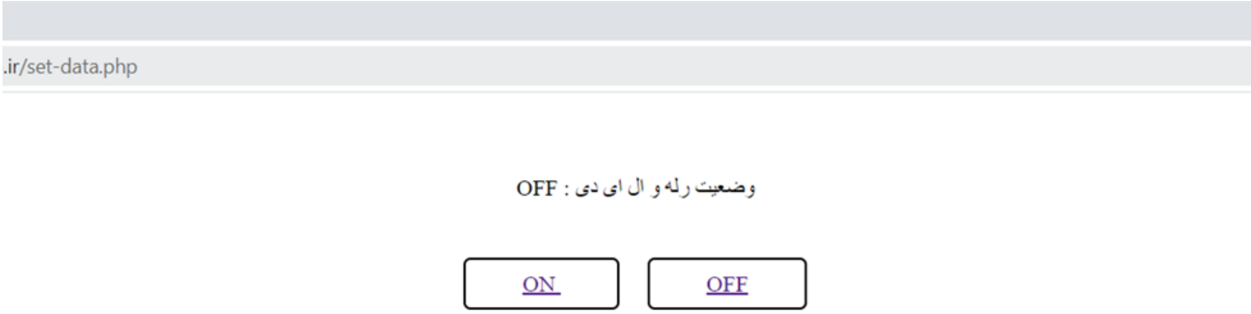
Data regarding the LED's on/off status was stored in a MySQL database, managed via phpMyAdmin. PHP programming and the PDO library were used to interact with the database. The sample code for establishing a database connection is as follows:

```
define('DB_HOST', 'localhost');

define('DB_USER', 'user-name');

define('DB_PASS', 'pass');

define('DB_NAME', 'db-name');


$pdo_conn = new PDO('mysql:host='.DB_HOST.';dbname='.DB_NAME, DB_USER, DB_PASS);
```

A MySQL table named arduino was created with a field named led, storing either on or off. Two PHP files, get-data.php and set-data.php, were created on the server:

1. **get-data.php**: Retrieves the LED status from the database.
2. **set-data.php**: Updates the LED status in the database.

The Arduino was connected to the server to request the get-data.php file using the POST method. The returned value (on or off) was used to turn the LED on or off. The set-data.php file provided an HTML interface for users to update the LED status via mobile, tablet, or computer browsers.



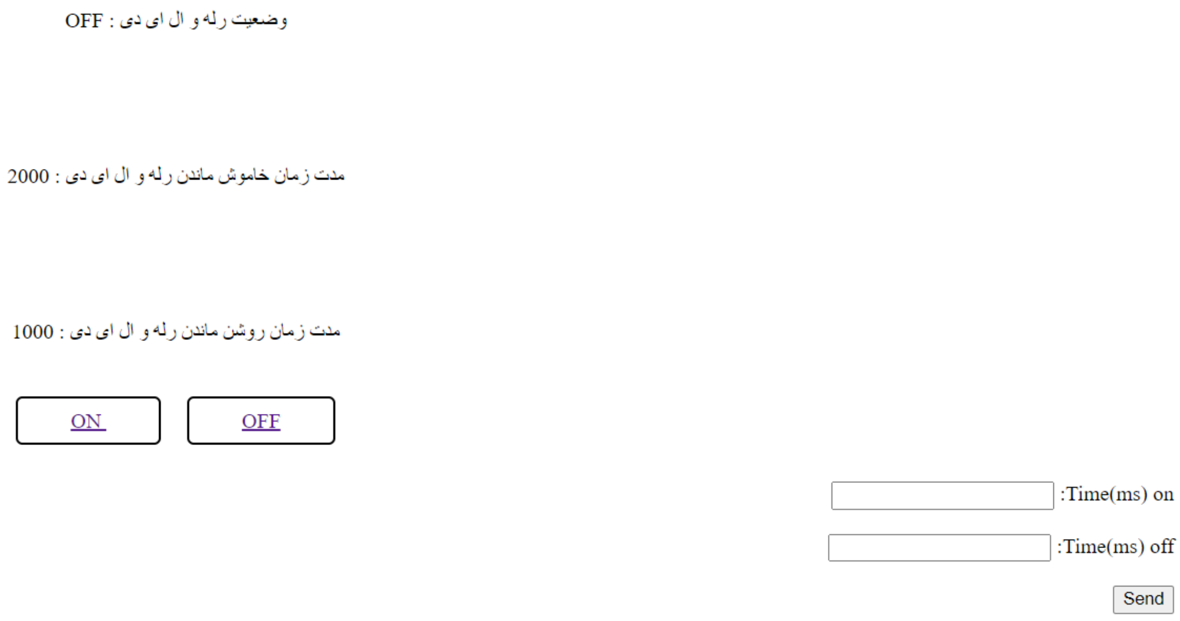
.ir/set-data.php

وضعیت رله و ال ای دی : OFF

Explanation of Question 3 Response

A brief explanation along with screenshots and potential issues encountered.

Two additional fields, ontime and offtime, were added to the MySQL database to store the LED on and off times. When the user clicked "send," these values were sent via the GET method to set-data.php. This file included conditions to check if ontime and offtime parameters were sent. If not, the default page was displayed.



وضعیت رله و ال ای دی : OFF

مدت زمان خاموش ماندن رله و ال ای دی : 2000

مدت زمان روشن ماندن رله و ال ای دی : 1000

Here, to retrieve values from the get-data.php file, we displayed the values as follows. For example, if the value led=on and the duration for being on is 5000 and for being off is 2000, we display it in the following format:

In get-data.php, the response was formatted as follows:

ONt15000t22000

Where:

- ON: LED status.
- t1: Time for the LED to stay on.
- t2: Time for the LED to stay off.

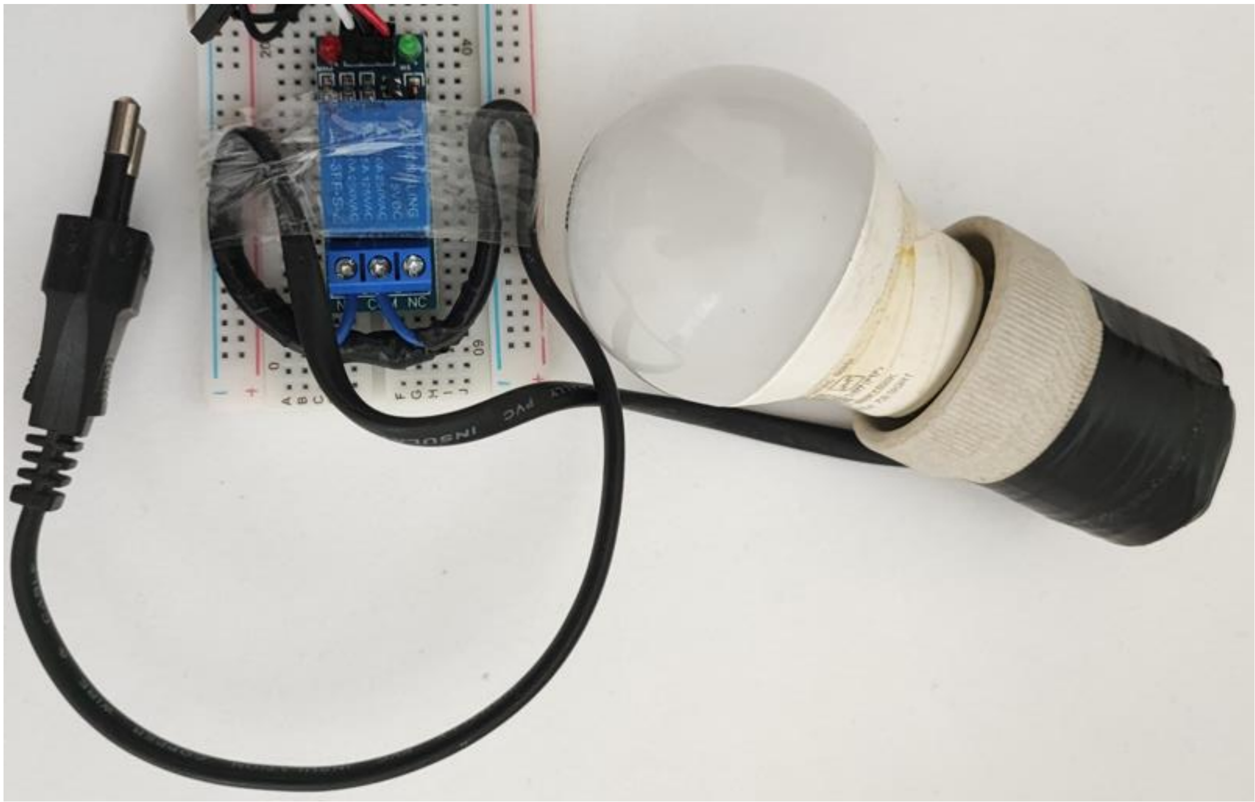
In Arduino, the values were extracted using substring functions. For example:

```
startStr = OnTimeStr.indexOf("t1") + 2;
endStr = OnTimeStr.indexOf("t2");
OnTimeStr = OnTimeStr.substring(startStr , endStr);
```

Explanation of Question 4 Response

The LED was replaced with a 5V relay module, allowing control of a lamp using a socket. A two-wire cable was cut, and one wire was connected to the relay's NO and COM terminals. The relay acted as a switch. The other relay terminals were connected as follows:

- VCC: 5V power from ESP module.
- GND: Ground.
- IN: Input signal to control the relay.



Explanation of Question 5 Response

A host and domain setup were explained in Question 2. Using PHP programming, the LED status was periodically updated from the server and used to control the Arduino.

Explanation of Question 6 Response

A Telegram bot was developed using PHP and Telegram's standard API. Steps:

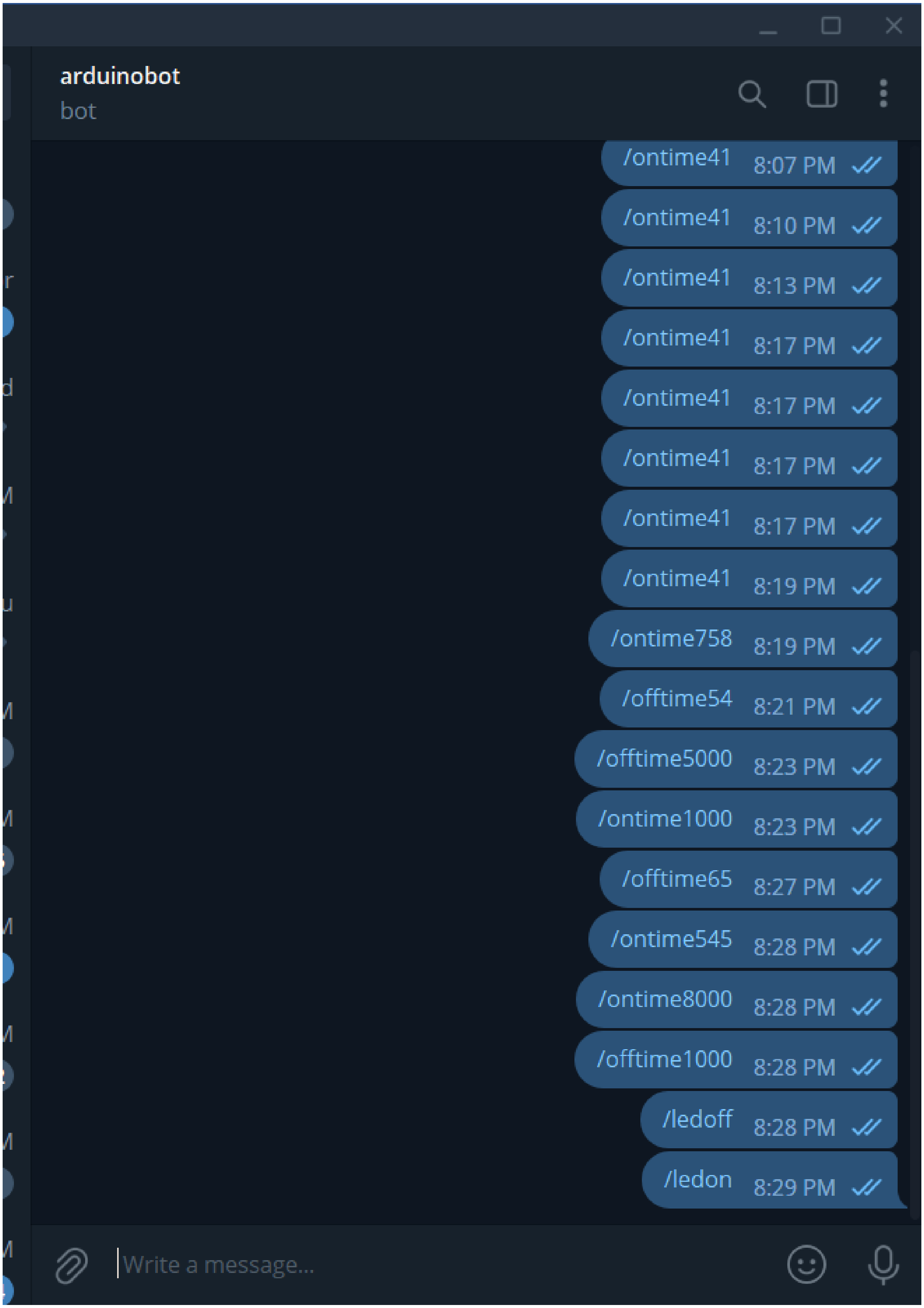
1. Registered the bot on Telegram using BotFather and received a token.
2. Set up a webhook:

https://api.telegram.org/bot[token]/setwebhook?url=

Here, [token] was replaced with the bot token, and url with the server domain and file location (set-data.php).

Bot Commands:

- **Turn LED on**: /ledon
- **Turn LED off**: /ledoff
- **Set on-time**: /ontime [value]
- **Set off-time**: /offtime [value]



arduinobot
bot

```
/ontime41      8:07 PM
/ontime41      8:10 PM
/ontime41      8:13 PM
/ontime41      8:17 PM
/ontime41      8:17 PM
/ontime41      8:17 PM
/ontime41      8:17 PM
/ontime41      8:19 PM
/ontime758     8:19 PM
/offtime54     8:21 PM
/offtime5000   8:23 PM
/ontime1000    8:23 PM
/offtime65     8:27 PM
/ontime545     8:28 PM
/ontime8000    8:28 PM
/offtime1000   8:28 PM
/ledoff        8:28 PM
/ledon         8:29 PM
```