Electronic Design Workshop

Final Report

Voice Controlled Scrolling LED Display

GROUP NO. 3

PREPARED BY

TANYA SONI – 19070123118
SHARMISHTHA TONGIA – 19070123125
TUSHAR YADAV – 19070123126
VARUN SARATHY – 19070123131

MENTOR

Prof. ADITYA JAIN

TABLE OF CONTENTS

INTRODUCTION

Voice is the most important form of communication, which involves transmitting and receiving data. A Digital LED board is used to display a range of messages in a simple and readable format, and it is used at various places to display messages and attract a crowd. Previous models were like Digital LED boards based on a GSM modem or simple scrolling text models which are not reprogrammable. In this project, we made a digital-led board using the voice-commands technique. We used a NodeMCU microcontroller for controlling the led board, and the microcontroller will take speech as text input and display it on the screen.

DIGITAL LED BOARD DISPLAYING TEXT USING GOOGLE ASSISTANT

OBJECTIVE

The main goal of this system is to create a wireless notice board that displays messages sent by users as well as to design a simple, easy-to-install, user-friendly system that can receive and display messages. In this project, we give voice input through our Android mobile using Google Assistance and display it on the LED Matrix Board.

NEED OF THE PROJECT

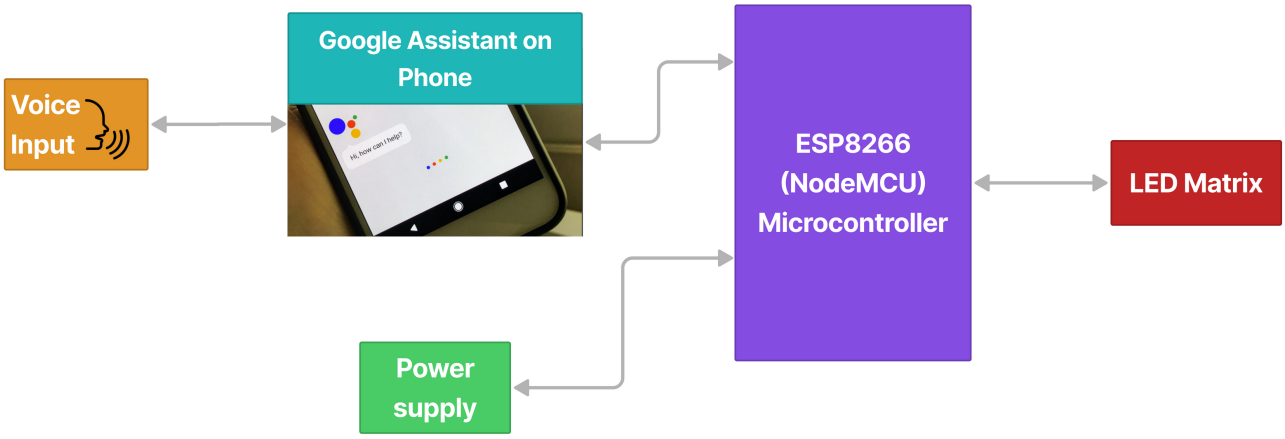
Normally, we use a simple static LED display screen to display a message, and also when we want to display a large message, we change the message every few instances. Nowadays, scrolling displays are more preferred to static ones. So by using a pre-programmed controller, we can make LED displays in a scrolling way. The project would use voice recognition to display messages instead of traditional typing which would help in saving time and energy. And whenever the user needs to change the message on the LED displays, they need to carry a laptop, computer, or keypad, and it can be a hectic task.

So in this project, we can change the messages on the LED displays by just talking to Google Assistant on the phone, which makes our project hassle-free and easy to use. We can use this project on the college notice board and universities for displaying day-to-day information continuously or at regular intervals during working hours. Display devices can be set up at various places on the campus. And this display device can be set up at various other places.

BENEFITS TO SOCIETY:

1- An easy and convenient way to display important notices.
2- Eliminates the need to type.
3- Voice commands through Google Assistant are more time-efficient.

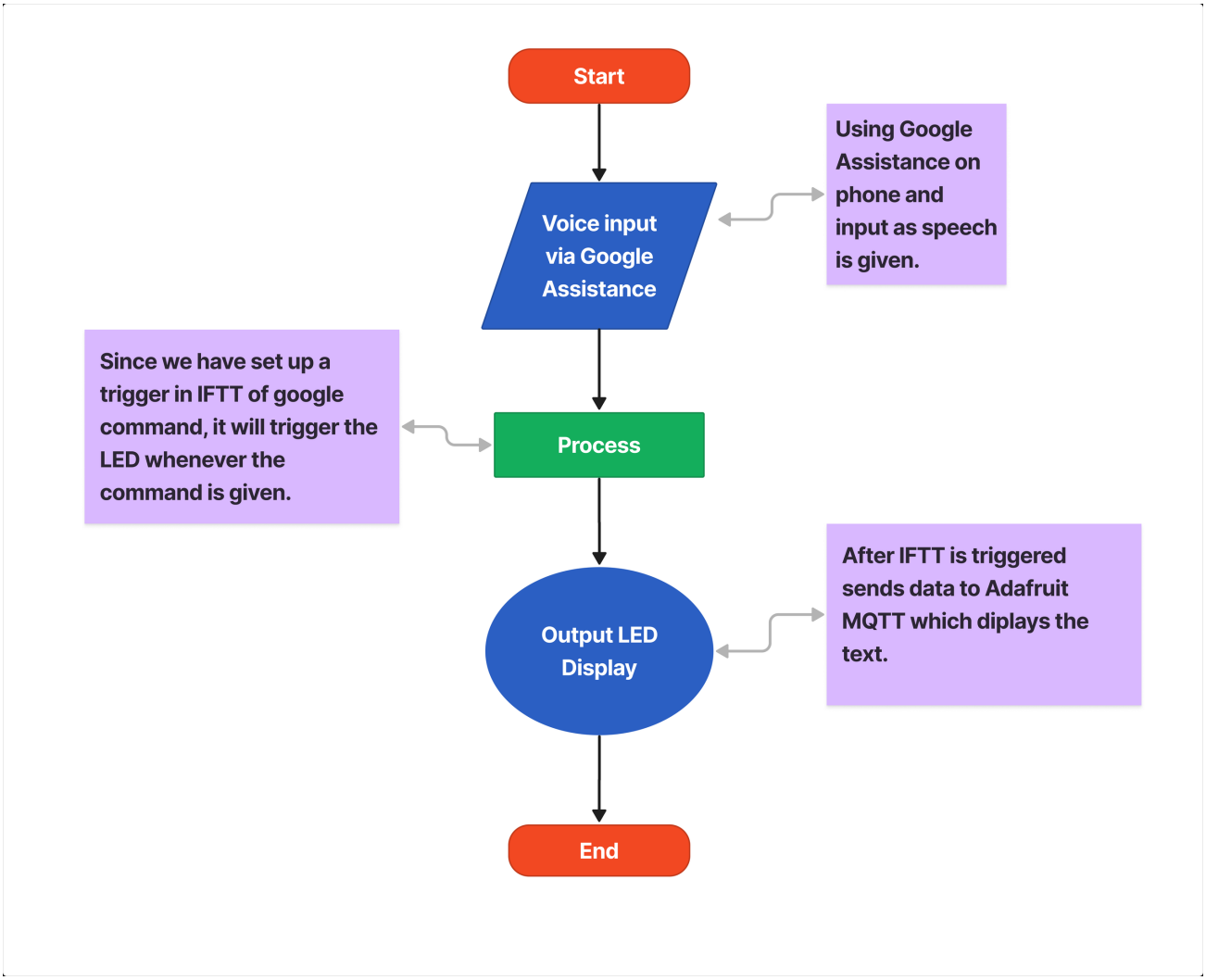
BLOCK DIAGRAM



A voice input is received by the microphone on the Android phone. This input is then relayed onto the NodeMCU, where the code that has been programmed converts the speech into text and displays it onto the LED matrix display in a continuous fashion.

Voice Controlled Scrolling LED Display

FLOWCHART



HARDWARE COMPONENTS

LED matrix display:

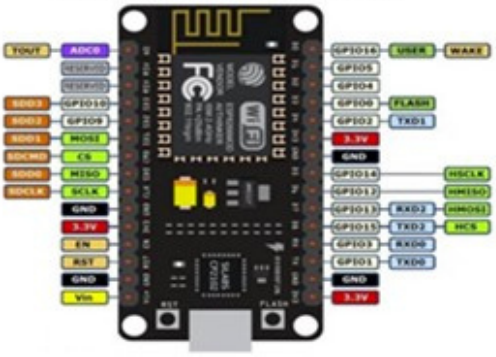
We will use an LED matrix display in this project to display the information. They are also available in different dimensions like 5 x 7, 8 x 8, 16 x 16, 32 x 32 etc. Based on the arrangement of the LEDs in the matrix, an LED matrix can be either a common row anode or a common row cathode. A typical MAX7219 module includes an 8-dot matrix display and a MAX7219 LED display driver.

LED matrix display

NodeMCU module:

ESP8266 is a very low-cost and user-friendly Wi-Fi module that develops a simple TCP/IP connection. It is designed by Espressif Systems. It operates on a +3.3v power supply. And it can be programmed within the Arduino IDE.

Pinout Diagram of ESP8266 NodeMCU



NodeMCU module

Voice Controlled Scrolling LED Display

Female to Female Wires:

To connect to the microcontroller.

Female to Female Wires

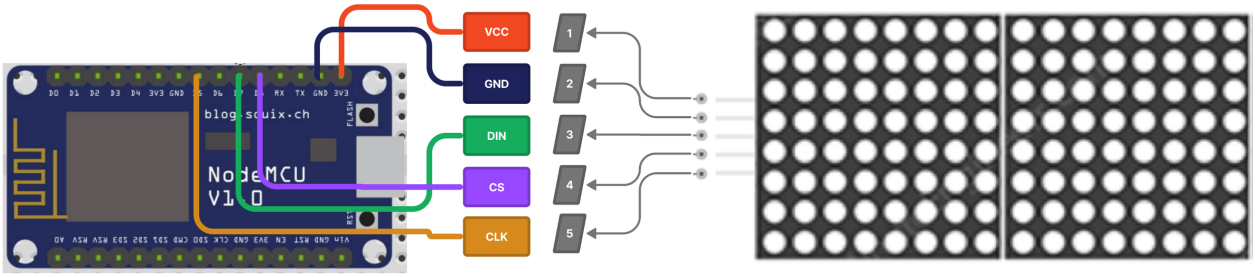
USB Cable:

To connect ESP8266 to a computer.

USB Cable

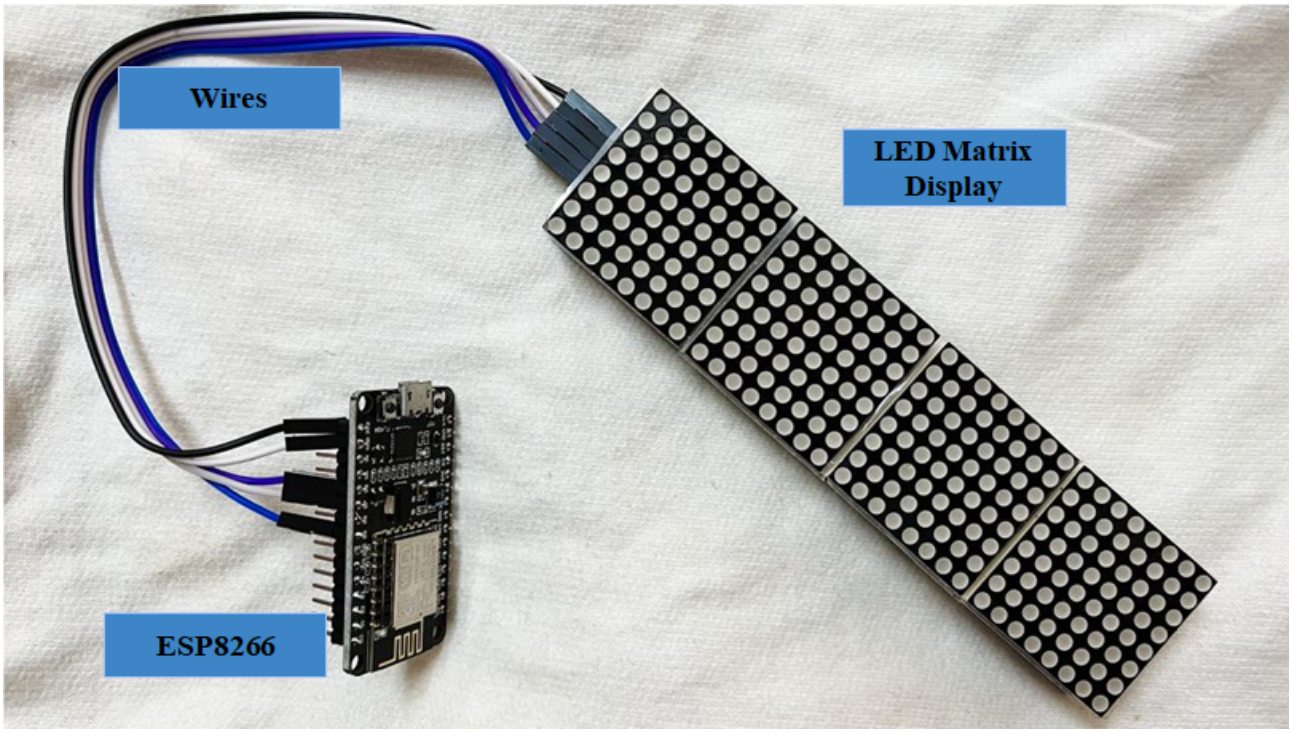
An android phone with Google Assistant.

CONNECTIONS



- VCC is connected to 5V.
- GND is connected to the common ground.
- DIN is the Data In. It is connected to the D7 digital pin of the microcontroller.
- CS is Chip Select. It is connected to the D8 digital pin of the microcontroller.
- CLK is the Clock pin. It is connected to the D5 digital pin of the microcontroller.

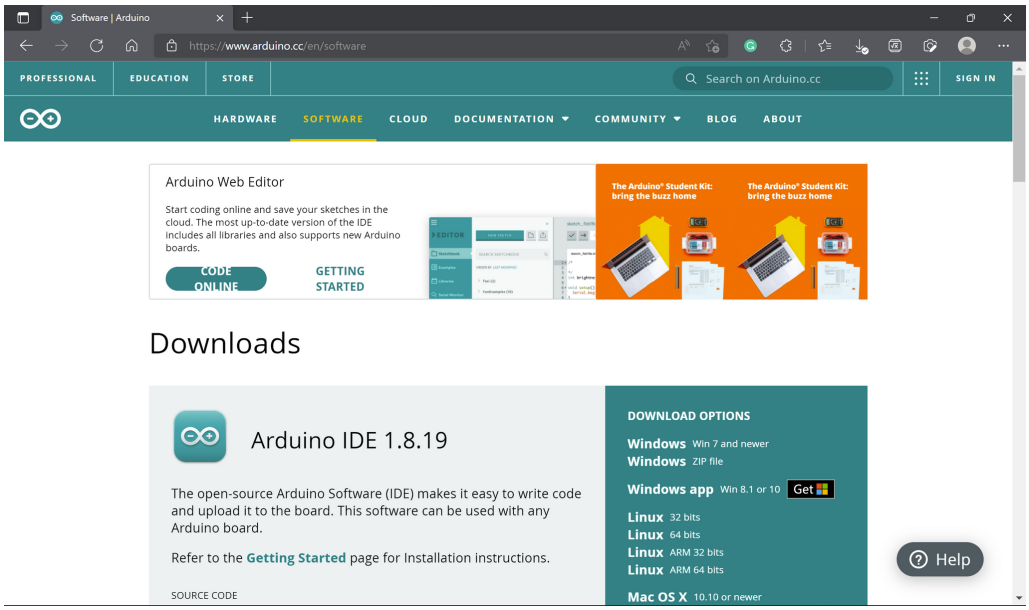
PURCHASED COMPONENTS



SOFTWARE USED

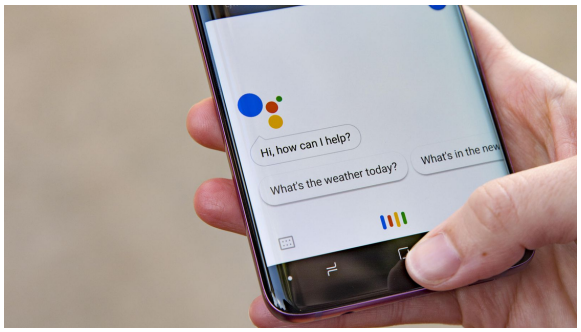
Arduino IDE:

It is used to program the ESP8266 microcontroller.



Android Phone with Google Assistance:

Google Assistant is a virtual assistant powered by artificial intelligence that is primarily available on mobile and smart home devices. All you have to do is say "OK Google" or "Hey Google" to wake it up. Google Assistant can handle a wide range of tasks, like answering questions, providing weather updates, playing music, discovering movies, and even translating languages.



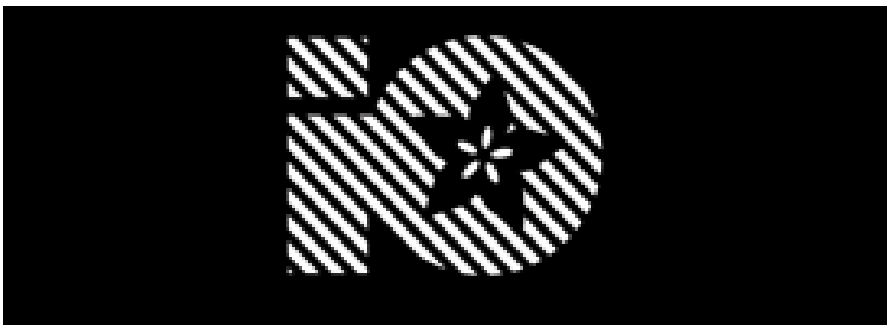
IFTTT App:

Google is collaborating with a variety of companies to make their services available on Google Assistant. IFTTT (If This Then That) is one of these partners, allowing you to create custom behaviors in response to your voice commands.

if this then that

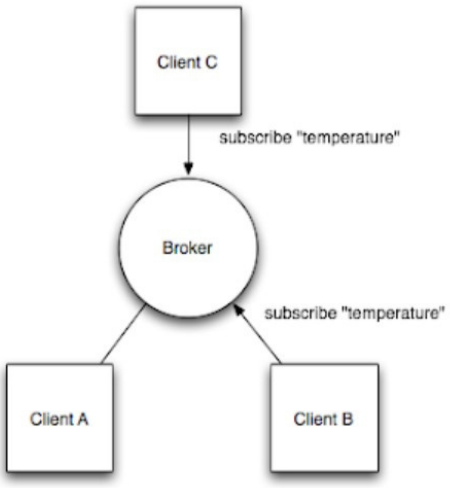
Adafruit IO:

It is an IoT platform based on the MQTT Protocol (Message Queue Telemetry Transport). MQTT is a lightweight protocol that enables several devices to connect to a common server, referred to as the MQTT Broker, and subscribe to or write to user-defined topics. When a device subscribes to a subject, the broker notifies it anytime the topic changes. MQTT is suitable for applications that require modest data rates, have stringent power limits, or have slow Internet connections. Adafruit IO, in addition to providing the MQTT Broker service, allows you to create dashboards that allow you to change or view the current value of any subject directly. Since it can be accessed from a web browser, it makes it the ideal hub for monitoring and controlling all of your various IOT projects. It's the ideal hub for monitoring, and controlling all of your IoT projects because it can be accessed via a web browser.

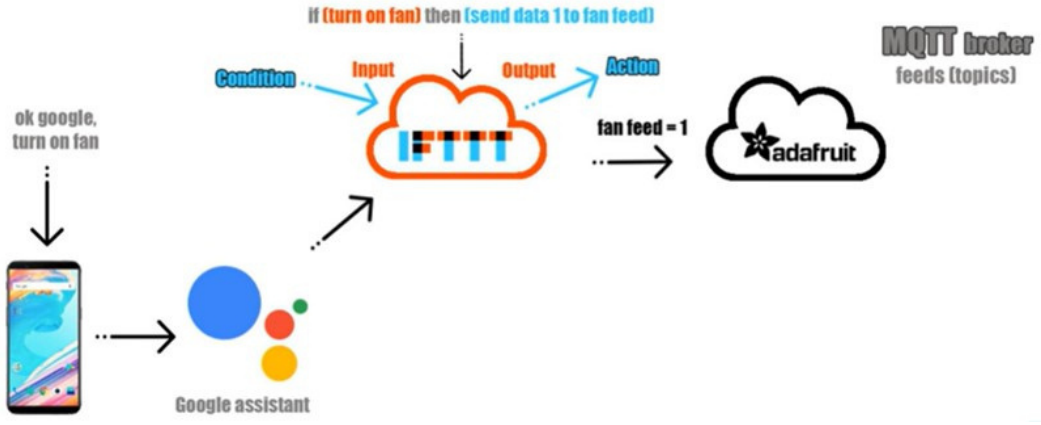


METHODOLOGY

Adafruit MQTT (message queue telemetry transport) and IFTTT are used in this project to allow it to be controlled by Google Assistant from anywhere in the world. The AdafruitMQTT broker allows you to change the message from anywhere on the internet. Because the NodeMCU is operating as an MQTT client, it is always listening to the Adafruit MQTT broker. As a result, any modifications made on the server-side will be reflected on the client-side, i.e. our NodeMCU board. We used a tool called IFTTT (If this then that), to alter our message on the MQTT broker side using Google Assistant. We used an IFTTT to create an applet that connects two services: Google Assistant and Adafruit MQTT. So, with the help of a good applet, we can successfully alter the message on the Adafruit broker side using our phone's Google Assistant.



WORKING OF MQTT



WORKING OF IFTTT

CODE

```
/********** Necessary Libraries ***********/
#include <MD_MAX72xx.h>
#include <SPI.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

#define SCROLL_DELAY 75

/******** Variables ****************/
char* str;
String payload;
uint32_t present;
bool first_time;
uint16_t scrollDelay;// in milliseconds
#define CHAR_SPACING1 // pixels between characters

// Global message buffers shared by Serial and Scrolling functions
#define BUF_SIZE 75
char curMessage[BUF_SIZE];
char newMessage[BUF_SIZE];
bool newMessageAvailable = false;

ESP8266WiFiMulti WiFiMulti;

#define  MAX_DEVICES   8

/************ Matrix Display Pins *****************/
#define CLK_PIND5 // or SCK
#define DATA_PIN D7 // or MOSI
#define CS_PIN   D8 // or SS
```

```
/************* Adafruit.io Setup ******************/
#define AIO_SERVER"io.adafruit.com"
#define AIO_SERVERPORT 1883                   // use 8883 for SSL
#define AIO_USERNAME    "varunsarathy"
#define AIO_KEY"aio_BTCa50BSh3FuijoAWNceDxnpckLv"


/************ Global State***************/


WiFiClient client;

//WiFiClientSecure client;

// Setup the MQTT client class by passing in the WiFi client and MQTT server and
login details.
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY);

/************ Feeds *****************/

// Setup a feed called 'onoff' for subscribing to changes.
Adafruit_MQTT_Subscribe message = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/Message");

/************* Sketch Code ********************/

void MQTT_connect();

// SPI hardware interface
MD_MAX72XX mx = MD_MAX72XX(CS_PIN, MAX_DEVICES);
// Arbitrary pins
//MD_MAX72XX mx = MD_MAX72XX(DATA_PIN, CLK_PIN, CS_PIN, MAX_DEVICES);

uint8_t scrollDataSource(uint8_t dev, MD_MAX72XX::transformType_t t)
```

```
// Callback function for data that is required for scrolling into the display
{
 static char      *p = curMessage;
 static uint8_t   state = 0;
 static uint8_t   curLen, showLen;
 static uint8_t   cBuf[8];
 uint8_t colData;

 // finite state machine to control what we do on the callback
 switch (state)
 {
  case 0:   // Load the next character from the font table
    showLen = mx.getChar(*p++, sizeof(cBuf) / sizeof(cBuf[0]), cBuf);
    curLen = 0;
    state++;

    // if we reached end of message, reset the message pointer
    if (*p == '\0')
    {
      p = curMessage;          // reset the pointer to start of message
      if (newMessageAvailable)    // there is a new message waiting
      {
        strcpy(curMessage, str);    // copy it in
        newMessageAvailable = false;
      }
    }
  // !! deliberately fall through to next state to start displaying

  case 1:   // display the next part of the character
    colData = cBuf[curLen++];
    if (curLen == showLen)
    {
      showLen = CHAR_SPACING;
      curLen = 0;
      state = 2;
    }
    break;
```

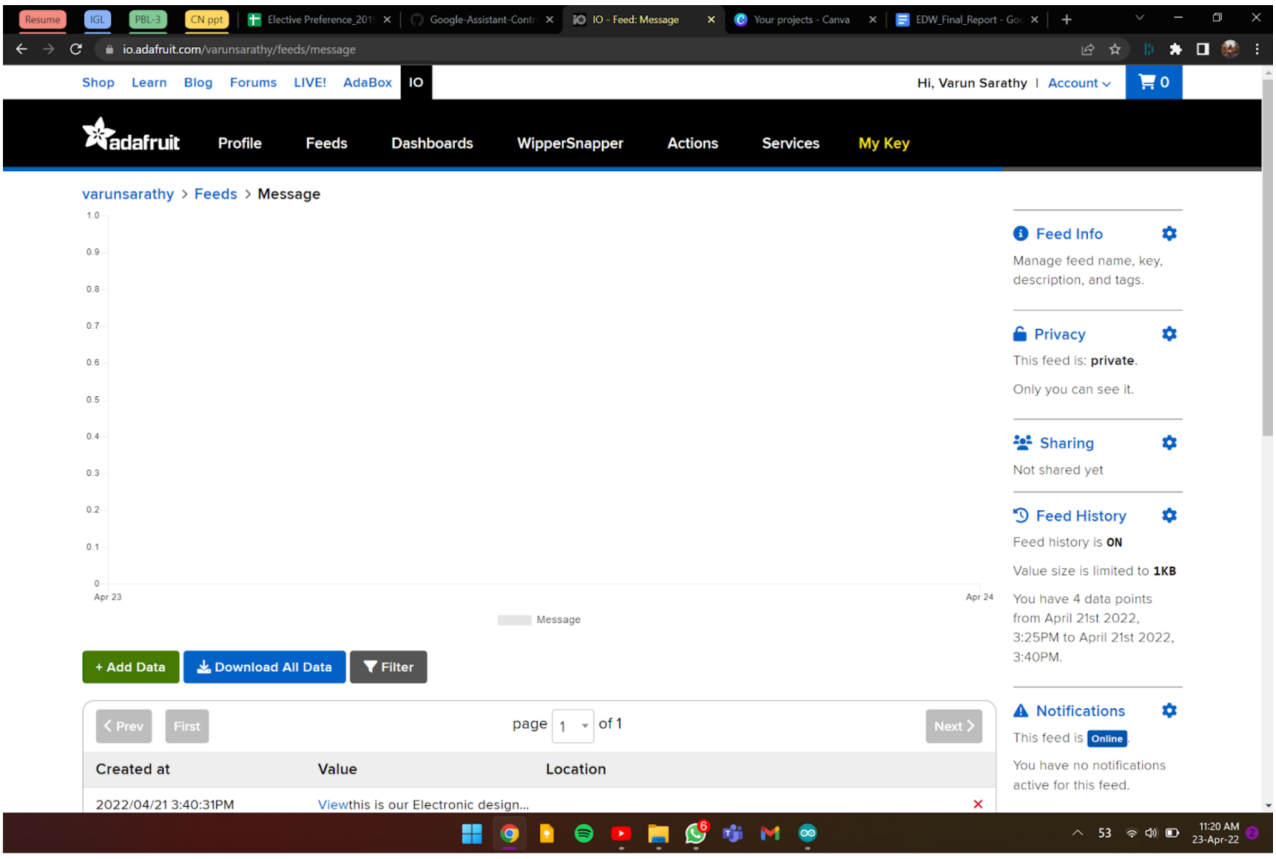
```
    case 2:   // display inter-character spacing (blank column)
      colData = 0;
      curLen++;
      if (curLen == showLen)
        state = 0;
      break;

    default:
      state = 0;
  }

   return (colData);
}

void scrollText(void)
{
  static uint32_t  prevTime = 0;

  // Is it time to scroll the text?
  if (millis() - prevTime >= scrollDelay)
  {
    mx.transform(MD_MAX72XX::TSL);    // scroll along - the callback will load all the
data
    prevTime = millis();          // starting point for next time
  }
}
void no_connection(void)
{
  newMessageAvailable = 1;
  strcpy(curMessage, "No Internet! ");
  scrollText();
}
```

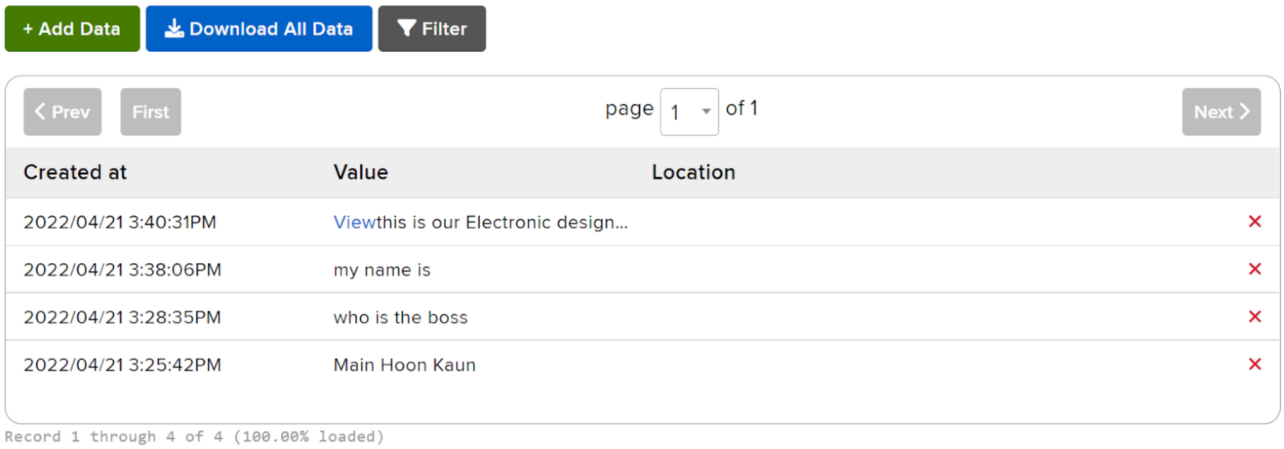
```
void setup()
{
 mx.begin();
 mx.setShiftDataInCallback(scrollDataSource);

 scrollDelay = SCROLL_DELAY;
 strcpy(curMessage, "Hello! ");
 newMessage[0] = '\0';

 Serial.begin(57600);
 Serial.print("\n[MD_MAX72XX Message Display]\nType a message for the scrolling
display\nEnd message line with a newline");

 Serial.begin(115200);
 // Serial.setDebugOutput(true);

 Serial.println();
 Serial.println();
 Serial.println();

 for (uint8_t t = 4; t > 0; t--) {
   Serial.printf("[SETUP] WAIT %d...\n", t);
   Serial.flush();
   delay(1000);
 }

 WiFi.mode(WIFI_STA);
 WiFiMulti.addAP("Vee", "SRLSSLmsl");
 WiFiMulti.addAP("SSID2", "PASS2");
 WiFiMulti.addAP("SSID3", "PASS3");
 Serial.println("Connecting");
 newMessageAvailable = 1;
 present = millis();
 first_time = 1;
   // Setup MQTT subscription for onoff feed.
 mqtt.subscribe(&message);
 str = " WELCOME TO G3 EDW PROJECT ";
}
```

```
void loop()
{
 while (WiFiMulti.run() != WL_CONNECTED) {
   Serial.println("WiFi not connected!");
   delay(1000);
 }

 MQTT_connect();


 Adafruit_MQTT_Subscribe *subscription;
 while ((subscription = mqtt.readSubscription(1))) {
   if (subscription == &message) {
     payload ="";
     Serial.print(F("Got: "));
     Serial.println((char *)message.lastread);
     str = (char*)message.lastread;
     payload = (String) str;
     //payload.toUpperCase();       // Comment this if upper case character is required
     payload += "      ";
     str = &payload[0];
     newMessageAvailable = 1;
   }
 }

 scrollText();
}
```

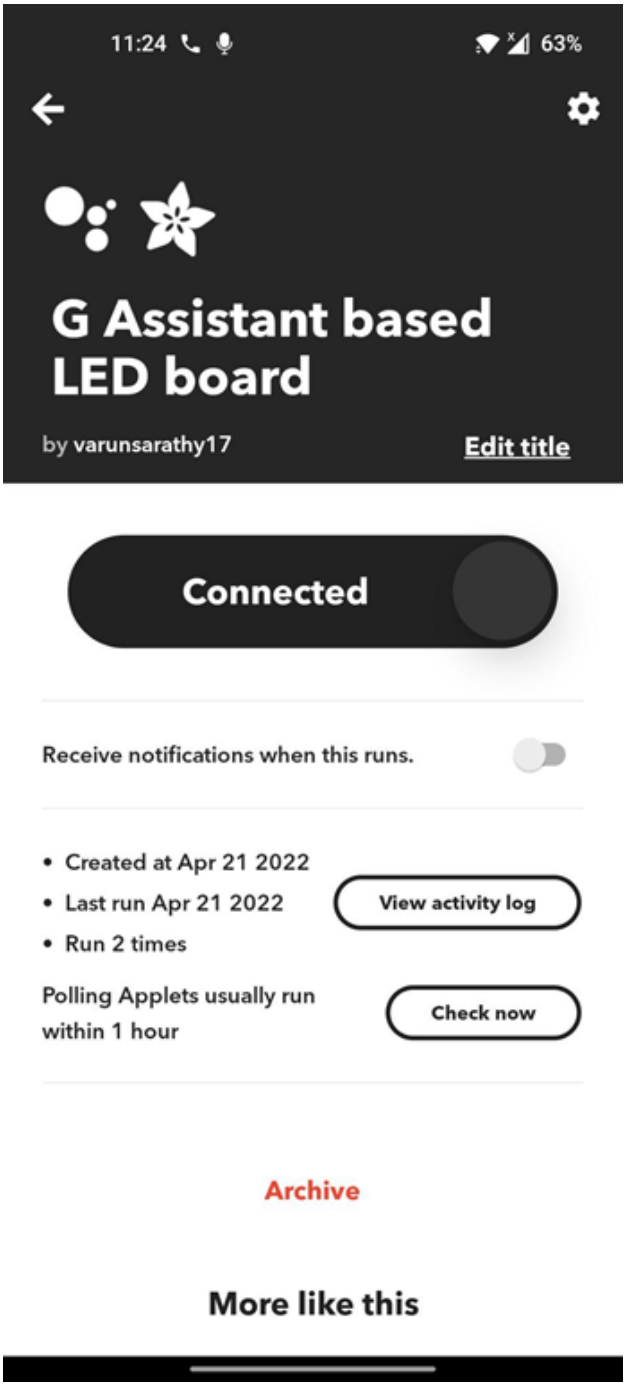
```
void MQTT_connect() {
 int8_t ret;

 // Stop if already connected.
 if (mqtt.connected()) {
   return;
 }

 Serial.print("Connecting to MQTT... ");

 uint8_t retries = 3;
 while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
Serial.println(mqtt.connectErrorString(ret));
      Serial.println("Retrying MQTT connection in 5 seconds...");
      mqtt.disconnect();
      delay(5000); // wait 5 seconds
      retries--;
      if (retries == 0) {
        // basically die and wait for WDT to reset me
        while (1);
      }
 }
 Serial.println("MQTT Connected!");
}
```

Adafruit:


io.adafruit.com/varunsarathy/feeds/message
Shop Learn Blog Forums LIVE! AdaBox IO
Hi, Varun Sarathy | Account 0
adafruit Profile Feeds Dashboards WipperSnapper Actions Services My Key
varunsarathy > Feeds > Message
Feed Info
Manage feed name, key, description, and tags.
Privacy
This feed is: private.
Only you can see it.
Sharing
Not shared yet
Feed History
Feed history is ON
Value size is limited to 1KB
You have 4 data points from April 21st 2022, 3:25PM to April 21st 2022, 3:40PM.
Notifications
This feed is Online
You have no notifications active for this feed.
Apr 23
Apr 24
Message
+ Add Data
Download All Data
Filter
Prev
First
page 1 of 1
Next
Created at
Value
Location
2022/04/21 3:40:31PM
Viewthis is our Electronic design...


+ Add Data
Download All Data
Filter
Prev
First
page 1 of 1
Next
Created at
Value
Location
2022/04/21 3:40:31PM
Viewthis is our Electronic design...
2022/04/21 3:38:06PM
my name is
2022/04/21 3:28:35PM
who is the boss
2022/04/21 3:25:42PM
Main Hoon Kaun
Record 1 through 4 of 4 (100.00% loaded)

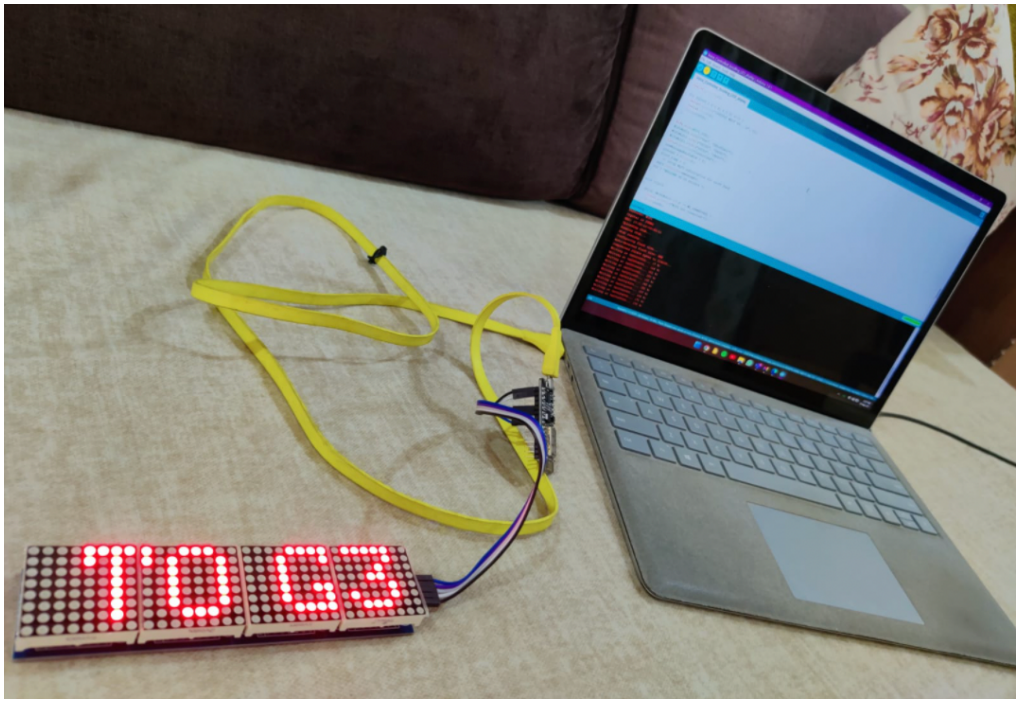
IFTTT:


11:24
63%
G Assistant based LED board
by varunsarathy17
Edit title
Connected
Receive notifications when this runs.
Created at Apr 21 2022
Last run Apr 21 2022
Run 2 times
View activity log
Polling Applets usually run within 1 hour
Check now
Archive
More like this

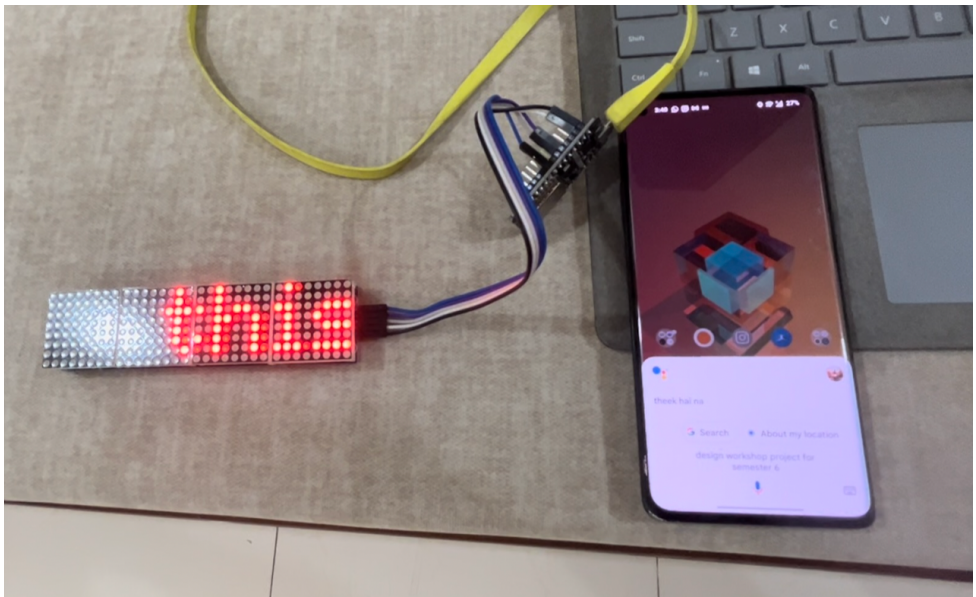

11:25
63%
Edit Applet
Created by varunsarathy17
If
Say a phrase with a text ingredient
Then
Send data to Adafruit IO
Update

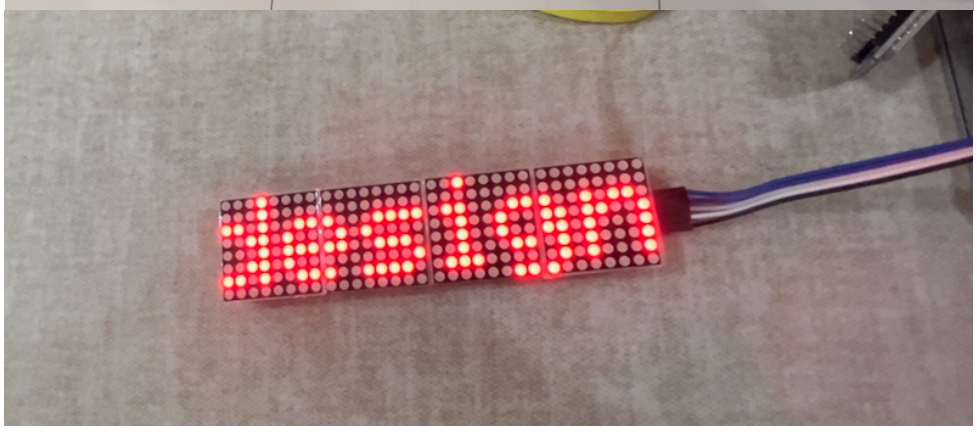
RESULTS

Welcome Message



Displaying Message by giving input through Google Assistance





ADVANTAGES

- Low cost
- Low power consumption
- Flexible
- The older version of display boards made use of wired technology for communication. Here we will be using wireless technology wi-fi for communication.
- This system will be more portable, an android phone is required instead of carrying a computer or keyboard every time to change the messages to be displayed on the LED matrix display.

LIMITATIONS

- The size of the board.
- The need for WiFi throughout the working of the system to display.

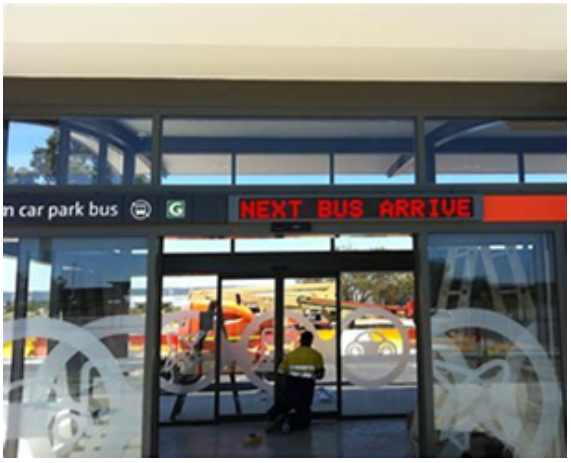
BILL OF MATERIALS

S.no.	Component	Quantity	Price (Rs.)
1	LED matrix display	4	425
2	ESP8266 (NodeMCU module)	1	400
3	Female to Female Wire	5	10
4	USB Cable	1	150
			Total = Rs. 985

APPLICATIONS

This display device can be set up at various other places such as -

- Educational institutes and organizations.
- Hospitals.
- Railway stations.
- Bus stations.
- Shopping malls.
- Scoreboard.
- Advertisements.
- Stock exchanges.
- Highways signboards



FUTURE PLANS

We may improve our idea by using solar panels, adding effects to the LED display, using colourful LED boards, and using a larger LED board. The future scope of this project is immeasurable as it is versatile and can be used in various DOMAINS. It can be built in various ways by adding additional features like , animations instead of text and by adding security features which would also have facial recognition before the modification of a text. It could further be enhanced by having a moving model

FUTURE INNOVATIONS AND USAGE OF THE PROJECT

CONCLUSION

As a result, We have successfully implemented a Voice controlled scrolling display by the use of which we can alter the text on the LED displays by simply talking to Google Assistant on our phone, making our project hassle-free and simple to use. We can utilize this project to display day-to-day information on college notice boards and universities throughout working hours, either constantly or at regular intervals. Display devices can be placed in a variety of locations around campus. Around hospitals ,commercial areas , businesses and shopping malls to attract a crowd and can also be used to display valuable offers/information in various sites .