|  |
| --- |
|  |
| Google Assistant Based Wireless Notice Board  COMMUNICATION SYSTEM PROJECT |
| |  |  |  | | --- | --- | --- | | ENGR. DURR E JABEEN | ELECTRONIC ENGINEERING DEPARTMENT | SIR SYED UNIVERSITY OF ENGINEERING & TECHNOLOGY | |

**GOOGLE ASSISTANT BASED WIRELESS NOTICE BOARD**

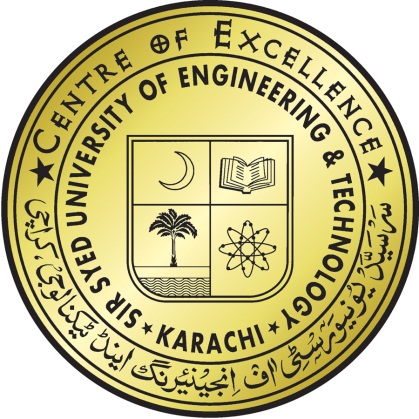
**COMMUNICATION SYSTEMS (EE\_470)**

**Submitted by**

**MUHAMMAD HASSAN 2017-EE-069 AAZIB AHMED ANSARI 2017-EE-078**

**MIRZA HAMZA UMER 2017-EE-079**

**MUAMMAD USMAN 2017-EE-107**



**7th Semester Project Report**

Department of Electronic Engineering

Sir Syed University Of Engineering and Technology, Karachi

**ACKNOWLEDGEMENT**

We would like to express our deepest appreciation to all those who provided us the possibility to complete this project and report.  A special gratitude we give to our project teacher, **ENGR.DURR E JABEEN,** whose contribution in stimulating suggestions and encouragement, helped me to coordinate our project especially in writing this report.

Furthermore we would also like to acknowledge with much appreciation the crucial role of **ENGR. SAMIYA** who gave the guidance to in our project and the necessary materials to complete the project. A special thanks goes to my Group Members who help me to assemble the parts and gave suggestion about the task. Last but not least, many thanks go to the head of the project, whose have invested his full effort in guiding the team in achieving the goal. I have to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comment and advices.

**TABLE OF CONTENTS**

**Page**

**Acknowledgment** 02

**Table of Contents** 03

**List of Figures**  04

**Abstract** 05

1. **INTRODUCTION**

1.1 Introduction 05

Google Based Wireless Notice Board 05

1. **PROJECT BLOCK DIAGRAM**

2.1 Block Diagram 06

2.2 Description of Blocks 06

1. **METHODOLOGY**
   1. Project working
   2. Coding 08
2. **RESULTS** 08
3. **CONCLUSION & FUTURE RECOMMENDATION** 09

**REFRENCES** 10

**APPENDIX A: Cost Analysis of the Project** 10

**LIST OF FIGURES**

**Page**

Figure 1.1 Scrolling Display  05

Figure 2.1 Block Diagram of Circuit 06

Figure 2.2 Node MCU 07

Figure 2.3 Pictures of Circuit 07

Figure 4.1 Observations & Results 09

**ABSTRACT**

This Project represents the idea of a wireless notice board which can be accessed from anywhere of the world having an internet connectivity. The theme of this project is taken from the old traditional notice which is build up Smart. But we took this concept to the next level with **much smaller size** (*around the size of a TV remote*), **much wider coverage area** (*almost from anywhere in the world*) and **much much convenient way of changing the message on it.** (*just say Ok Google, display on the board No Pain No Gain & your task is done*)

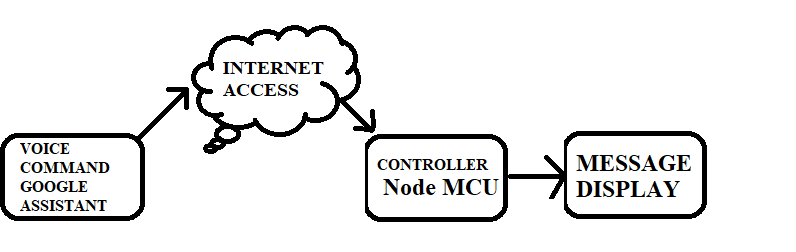
**INTRODUCTION**

* 1. **Introduction:**

This project is based on the concept of the traditional notice board which is made an advance version of the traditional by using the node MCU and with the internet connectivity. We normally use a simple static LED display screen to convey a message. Earlier, when we want to display large data, we used to change message for every few instances. Now scrolling displays are more preferred to static. By using a Node MCU controller, we can make LED display in scrolling way. We can also make LED to adoptable by using [PC controller based](http://www.edgefx.in/computer-controlled-robots-using-pc-for-ece-engineering-students/)system. Simple Outdoor LED Message Moving or Scrolling Sign Board, Electronic projects using LED Scroller Generator for outdoor digital signs, Marketable LED sign board with Message scrolling are the examples of the scrolling LED display.

**FIGURE 1.1 Scrolling Display**

**PROJECT BLOCK DIAGRAM**

* 1. **Block Diagram:**

**FIGURE 2.1 Block Diagram of the circuit**

* 1. **Description of Project:**

**HARDWARE & SOFTWARE DESCRIPTION**

This project uses two services to make it control through Google Assistant from anywhere in the world, [Adafruit MQTT](http://io.adafruit.com/) and [IFTTT](http://www.ifttt.com/). Adafruit MQTT broker allows to change the message from any internet connected device globally. Here the NodeMCU is acting as a MQTT client hence it is constantly listening to Adafruit MQTT broker. So if any changes occurs in the server side, the same changes will be observed on the client side i.e. on our NodeMCU board. And to change our message on the MQTT broker side via Google assistant, we are using one service called IFTTT. In IFTTT, we are making an applet in which we can connect two services, Google Assistant and Adafruit MQTT. So by making proper applet, we can successfully update the message on the adafruit broker side with our google assistant on our phone.

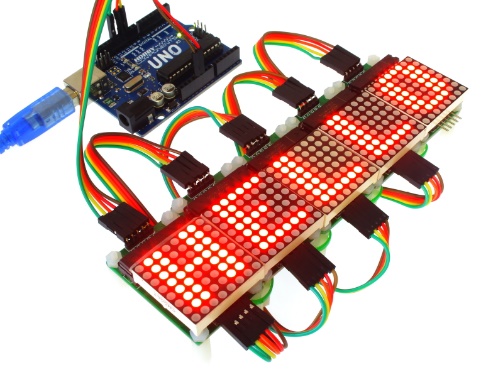
1. **NODE MCU:**

NodeMCU is a low-cost open source IoT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module.

**FIGURE 2.2: Node MCU**

1. **DOT MARIX DISPLAY 8\*8:**

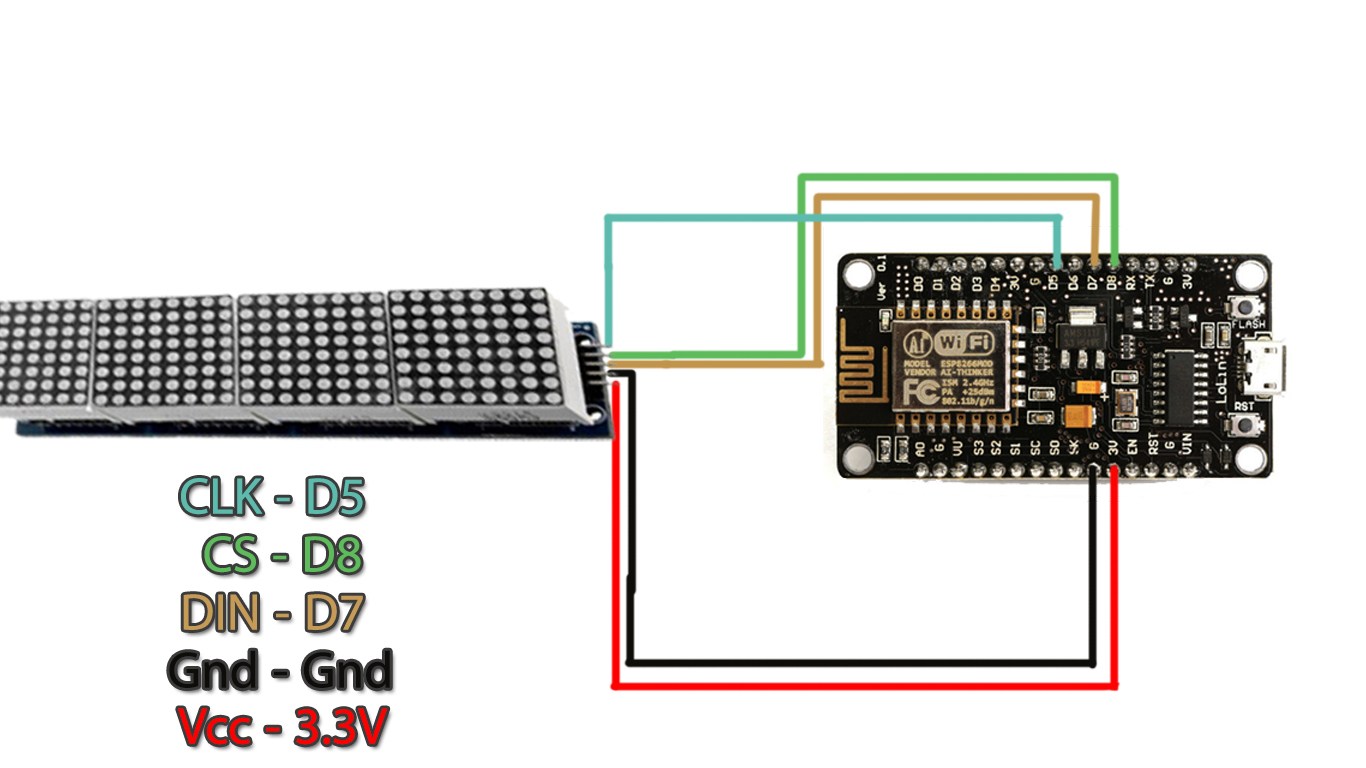
The **dot matrix** that we're going to use in this guide is a **8**×**8 matrix** which means that it has **8** columns and **8** rows, so it contains a total of 64 LEDs. The MAX7219 chip makes it easier to control the **dot matrix**, by just using 3 digital pins of the Arduino board.

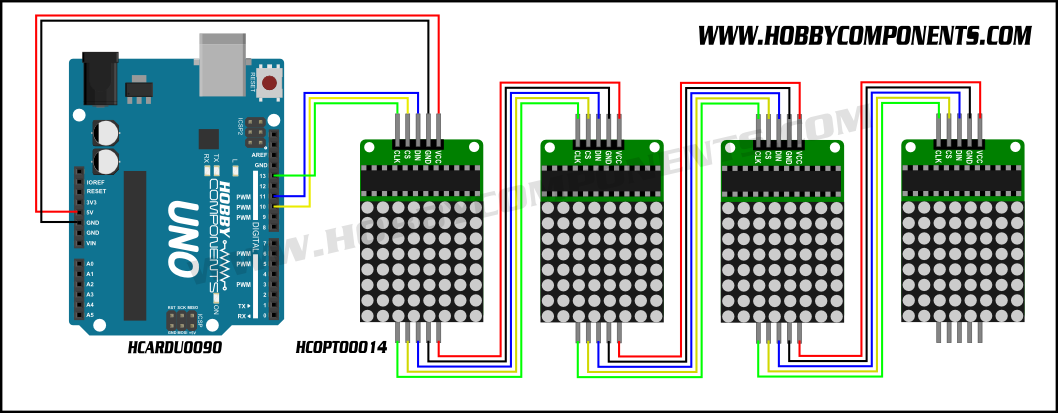
****

**FIGURE 2.3 PICTURES OF CIRCUIT**

**METHODOLOGY**

**3.1 Method:**

The node MCU is the controller for the scrolling text display the VCC and the GND of the dot matrix display is connected to the VCC of node MCU at 5V and the connect the CLK to D5 CS to D8 DIN to D7 and Vcc and GND pins to ESP8266 Board and the data is transferred serially.

****

**3.2 Coding:**

#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 130

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Variables \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

char\* str;

String payload;

uint32\_t present;

bool first\_time;

uint16\_t scrollDelay; // in milliseconds

#define CHAR\_SPACING 1 // 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 the number of devices we have in the chain and the hardware interface

// NOTE: These pin numbers will probably not work with your hardware and may

// need to be adapted

#define MAX\_DEVICES 8

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Matrix Display Pins \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define CLK\_PIN D5 // 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 "User"

#define AIO\_KEY "From adafruit"

/\*\*\*\*\*\*\*\*\*\*\*\* Global State (you don't need to change this!) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Create an ESP8266 WiFiClient class to connect to the MQTT server.

WiFiClient client;

// or... use WiFiFlientSecure for SSL

//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 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Bug workaround for Arduino 1.6.6, it seems to need a function declaration

// for some reason (only affects ESP8266, likely an arduino-builder bug).

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("Hamza Connect", "Usman157");

WiFiMulti.addAP("Aazib\_Ansari", "engr.aazib1808");

WiFiMulti.addAP("SSID3", "PASS3");

Serial.println("Connecting");

newMessageAvailable = 1;

present = millis();

first\_time = 1;

// Setup MQTT subscription for onoff feed.

mqtt.subscribe(&message);

str = " Notice Board ";

}

void loop()

{

while (WiFiMulti.run() != WL\_CONNECTED) {

Serial.println("WiFi not connected!");

delay(1000);

}

// Ensure the connection to the MQTT server is alive (this will make the first

// connection and automatically reconnect when disconnected). See the MQTT\_connect

// function definition further below.

MQTT\_connect();

// this is our 'wait for incoming subscription packets' busy subloo

// try to spend your time here

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 += " ";

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!");

}

**Results:**

**4.1 Results:**

By this project we have displayed our message on the electronic display system using a Node MCU controller which is interfaced by the google assistant using IFTT which uses the ch-308 to communicate with the display.

****

**FIGURE 4.1 RESULTS**

**Conclusions & Future Recommendations:**

* 1. **Conclusions & Future Recommendation:**

There are many Applications of Electronic Notice Board Control By Human Voice, few of them are listed below:

1. An advanced wireless notice board can be used in public transportation areas like Bus stations, Railway stations and even at Airports.
2. Voice operated electronic notice board finds its main application in educational premises like schools, colleges, university campuses. It can be used to display information like exam schedule, notice, event notification, and exam result announcement.
3. The main Advantage of Speech controlled electronic notice board is that the Wireless Notice board is easy to install and easy to use.
4. Speech controlled rolling display is really helpful for disabled people or handicapped people
5. We can add a feedback system in the Android app. So that user can get feedback of the action
6. We can implement a password so that any other person can not control the system.

**REFRENCES**

**[1]**[https://neelpmehta.wordpress.com/analog-PIC Micro-Controller-control-using-op- amps/?fbclid=IwAR21gfcrOejXRKZkJPOSoin9geFHaMaxskdBxyx\_7v5IlXw8UJg9p59MAWM](https://neelpmehta.wordpress.com/analog-pid-control-using-op-%20%20%20%20amps/?fbclid=IwAR21gfcrOejXRKZkJPOSoin9geFHaMaxskdBxyx_7v5IlXw8UJg9p59MAWM)

**[2] Book: Muhammad Ali Mazidi, Janice GillispieMazidi, Rolin D. McKinlay**

“The PIC MICRO-CONTROLLER & Embedded Systems”, Pearson Education Inc. 2nd Edition, 2008. pg. 123-138

**[3] Conference Paper from the Internet**

Stipanicev D., Marasovic J., “NODE MCU ”, Proceedings of 2003 IEEE Conference on Control Applications, Vol.2, June, pp.1350-1355, 2003. [Online]. Available: IEEE Xplore, http://www.ieee.org. [Accessed: 10 Sept. 2010].

**[4] Whole Internet Site**

[1]<http://aggie-horticulture.tamu.edu/ornamental/a-reference-guide-to-plant-care-handling-and-> merchandising/

[2] <http://www.controlledenvironments.org/Growth_Chamber_Handbook/Ch03.pdf>

[3] For pH circuit- [www.66pacific.com/ph/simplest\_ph.aspx](http://www.66pacific.com/ph/simplest_ph.aspx)

[4] <https://www.edgefx.in/scrolling-led-display-circuit/>

**For more material type visit:** <https://libguides.murdoch.edu.au/IEEE/all>

**APPENDIX-A**

**COST ANALYSIS OF THE PROJECT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MAJOR EQUIPMENT SPECIFICATION & COST** | | | | |
| **S.No** | **Component Name** | **Description** | **QTY** | **Cost** |
| **1** | **Node MCU** | **Controller** | **1** | **400** |
| **2** | **DOT MATRIX** | **Display** | **4** | **250 x 1** |
| **Total Cost of the Project** | | | | **1400/=** |