**Program: ESE 4009\_2**

**INSTRUCTOR:** Prof**.** Mike Aleshams

# **Group 4**

|  |  |  |
| --- | --- | --- |
| Student Name | Student ID | Signature\* |
| Dawinder Kaur | C0765505 | D.K. |
| Gopi Krishnan K. GopalaKrishnan | C0753811 | G.K. |
| Manpreet Kaur | C0757604 | M.K. |
| Raghav Chopra | C0772292 | R.C. |
| Shivinder Kaur | C0751078 | S.K. |

*\*By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a “0” on the work, an “F” in the course, or possibly more severe penalties.*

**Project Proposal**

**Project Title: IoT based smart Garage door opener**

* **Description of a latest similar project :**

**WiFi based smart Garage Door opener using NodeMCU ESP12E**

The door plays an important role in home security. So, providing a secured garage door system for houses has become necessary. This project deals with opening and closing of garage doors by giving commands through webpage. The principle of operation behind the working of this project lies in the functioning of NodeMCU ESP12E. It works as a wifi module that provides access to the webpage through a local web server. The webpage displays one button to open and close the garage door using a relay module and DC motor.

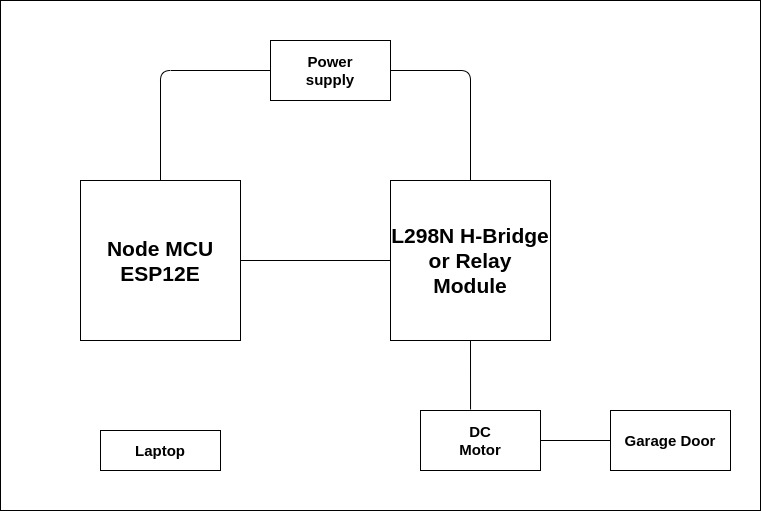


Fig 1.1 Basic block diagram of wifi based smart garage door opener

**Working of the existing project:**

The working principle of the project is giving commands through a web page displaying a message about garage door “open and close” status. So when the garage door is closed then the message is displayed as “The Garage Door is closed” and the OPEN button is shown on the screen. The Node MCU ESP12E is programmed accordingly using an Arduino IDE by providing a wifi username and password in code. When the user opens the webpage using the IP address from the serial monitor window the web page opens. When the user clicks on the open or close button the message goes to Node MCU which passes the command to the relay module to rotate the DC motor “clockwise or anticlockwise” which helps in opening and closing the garage door.

**Limitations of the existing project:**

1. The microcontroller Node MCU ESP12E can only be used as a wifi module
2. The project is not IoT based
3. There are no security features in this project

* **Final Solution:**

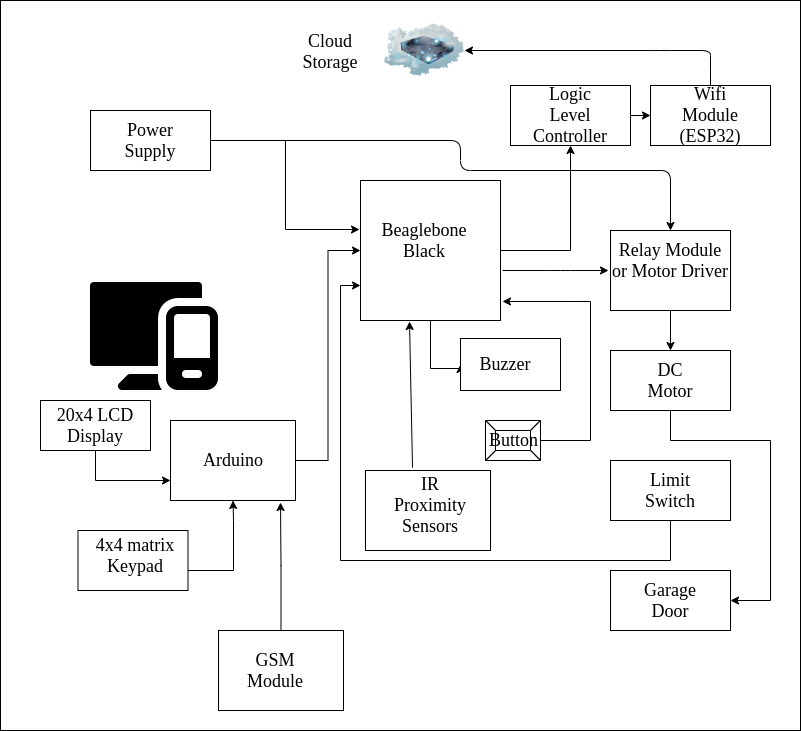


Fig 1.4 Block diagram for the final solution

For the final solution, we will implement the project as we have discussed in the second solution but with some modifications. We will add a GSM module in the project which will be connected with Arduino. This module will be used for sending notifications to the user via SMS in case the door remains open. Also, the keypad will be used to enter a password to open the door but for closing the door we will just need to press enter key.

**Features:**

* Using Beaglebone as the main microcontroller which has high processing speed and more advanced features.
* Using Arduino as a slave device which is supporting the alternate solution of opening and closing the door.
* The project is IoT based as it is connected to Wifi for cloud storage.
* IR Proximity Sensors will be used to detect any object in between the garage door while closing.
* Use of a buzzer if the door remains open for a long time.
* Also a GSM module has been added which will notify the owner via SMS that the door remained open.
* **Hardware and Software Components Used:**

**Hardware Components**

1. Arduino Uno
2. Beaglebone Black
3. ESP32 wifi module
4. 4x4 matrix keypad
5. 20x4 LCD Display
6. SIM800L Module GSM/GPRS
7. Power Supply
8. IR proximity sensors
9. Buzzer Electronic alarm
10. Push button
11. MicroLimit switch
12. DC Motor
13. L298N Motor Driver
14. SanDisk SD card 16 gb
15. Logic level controllers
16. PCB Board
17. 5mm LED light assorted kit

**Software Components**

1. Latest image of debian for beaglebone black
2. Amazon Web Services or ThingSpeak or any other free IoT service provider.
3. C/C++ Programing
4. HTML programming

* **Hardware and Software Tools Used:**

**Hardware Tools**

1. Multimeter
2. Breadboard
3. Jumper Wires/ Silicon hook up wires
4. General purpose tools
5. Soldering Kit
6. Laptop or host machine

**Software Tools**

1. Fritzing for PCB and schematic design
2. Eclipse IDE for programming
3. Compiler
4. Debugger
5. Assembler
6. Arduino IDE

* **Question /Answers**

**Use of various peripherals such as touch screens, cameras, microphones and speakers, GPIOs, timers, GPS modules, Bluetooth, WiFi, and ADC/DACs?**

* + Use of ESP32 module which has Wi-Fi using IEEE802.11 protocol.
  + Use of a DC motor and limit switches interfaced with the relay module for closing and opening of gates.
  + LCD display will display the password being entered as an asterisk and if the password is not correct then it will display as “Wrong Password”.
  + Keypad will be used to enter the password to open the door and the enter key will be used to close the door.
  + Arduino Uno is used to interface LCD and keypad to beaglebone black.
  + GSM module will be interfaced with Arduino Uno and it will communicate with the smartphone if the door remains open for a longer time.
  + Proximity IR sensors are used to detect any object in-between while closing the door.
  + Use of Buzzer when the door remain open for long time or any object is in-between the door
  + Limit switches are used to detect the position of the door.

**Use of I2C, SPI, RS232/RS-485, IrDA infrared, JTAG, USB, Bluetooth, IEEE 802.11 WiFi, IEEE 802.3 Ethernet, CAN and GPS protocols and systems?**

* + Relay Module Motor driver IC can be used to interface a DC motor with BeagleBone Black and it is connected to GPIO pins of BBB.
  + Serial UART interfacing can be done between ESP 32 and BeagleBone Black and Beaglebone Black and Arduino Uno.
  + The Wi-Fi protocol used is IEEE802.11.
  + The protocol used for communicating with the cloud can be HTTP.
  + GPIO pins of BeagleBone Black can be used to interface buzzer and IR Proximity sensors.

**Use of preemptive versus cooperative scheduler operation; tick rate and time slicing; critical code; fixed, dynamic and hybrid task priority allocation; application-specific considerations; power management tactics; semaphores, mutexes and queues; debugging strategies; performance estimation?**

* + The system will use Real time concepts such as the preemptive scheduler operation where the priorities to each task will be assigned according to their functioning but the higher priority will be given to opening and closing the door.
  + The IR sensors will be performing an interrupt if anything comes in between the door while closing.
  + All the data like car entry, exit or status of the garage will be stored in the cloud which is a real time approach.

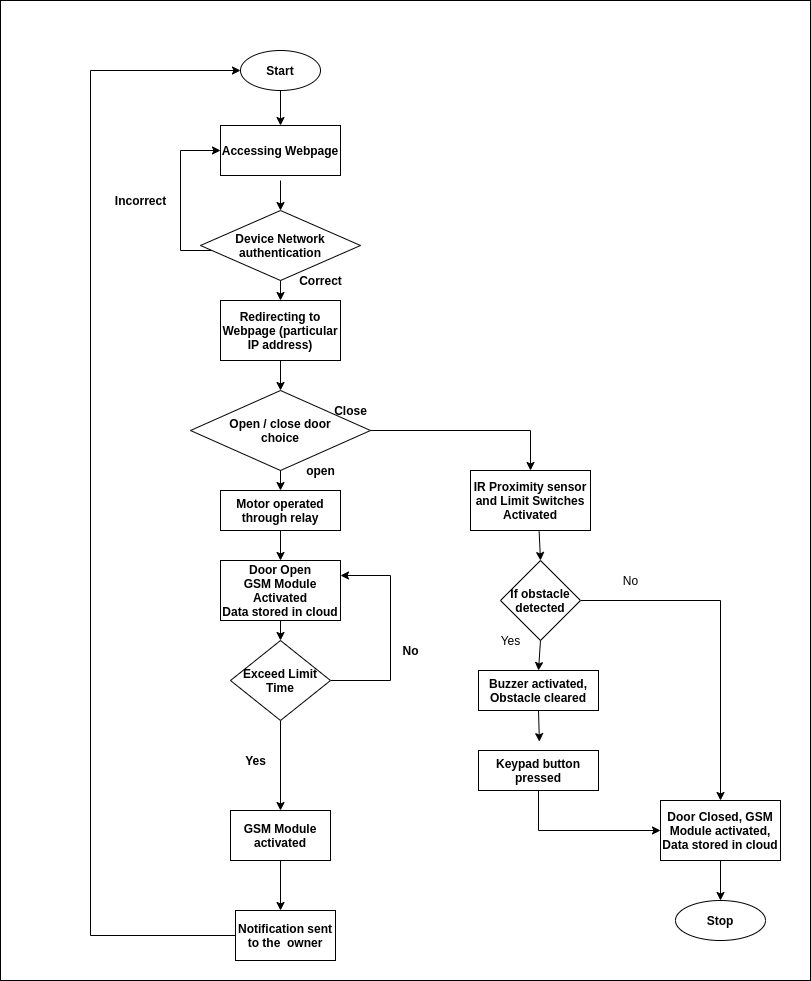
**Priority Scheduling strategy while opening a door:**

* Task 1 Priority : Open a door - Activation of motor drivers (through keypad or wifi module)
* Task 2 Priority : Activation of GSM Module to send Notification
* Task 3 Priority : Activation of Buzzer - if door remain open for long time
* Task 4 Priority : Disabling Buzzer through Push Button

**Priority Scheduling strategy while closing a door:**

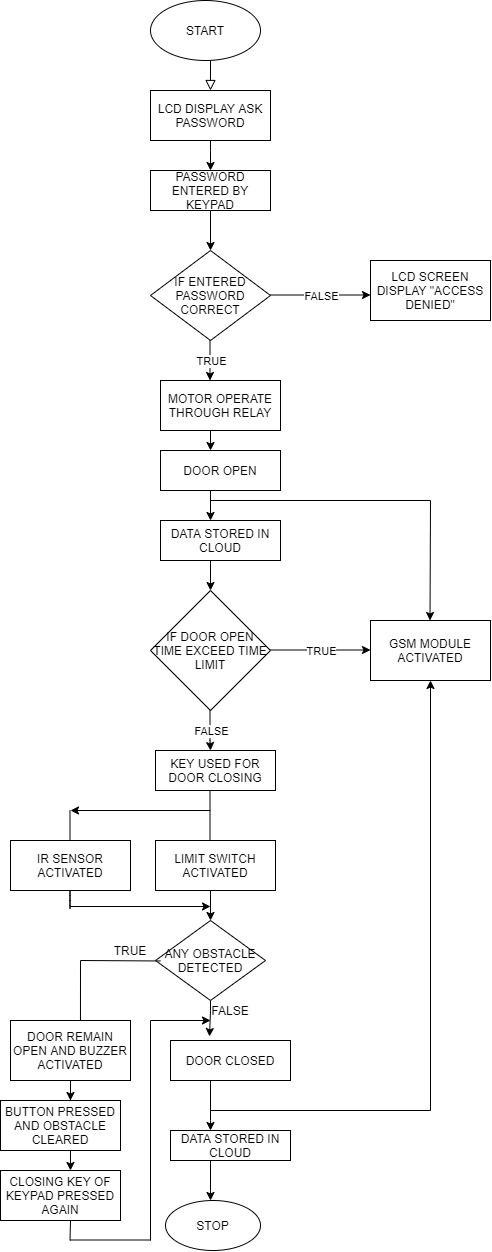
* Task 1 Priority : Activation of IR Proximity sensors (sensing obstacle)
* Task 2 Priority : Activation of Buzzer
* Task 3 Priority : Disabling the Buzzer through Push Button
* **Flowchart:**

1. **Flow Chart for IoT based smart garage door opening and closing using Webpage:**

****

**Explanation:**

First the local server is started and the IP address is entered to open the login page for the user. If the correct login id and password details are given the provision will be provided to open or close the garage door. If the user enters the incorrect details the login page access will be denied. Once the correct login details are entered a choice will be given for opening or closing the door. If the user chooses the open icon, an input will be given for the relay to operate which will in turn enable the rotation of the dc motor and the garage door will be opened. After the door is opened a notification is sent to the user with the help of a GSM module and the door opening time gets recorded in the cloud. If the user forgets to close the door for a set period of time, a notification will be sent to the user to remind them that the door is not closed. The next stage of this project is if the choice is made by the user for closing the garage door. In our project for ensuring a secure and safe closing of the door an IR proximity sensor along with limits switch is used. In case an obstacle for example: a kid comes in between when the door is being closed the two components prevent the door from closing. If an obstacle is detected, a buzzer will be activated which will notify the user of the potential occurrence of a danger. Once the obstacle is cleared by the user, the closing of the door will be possible by pressing the key of the keypad. If an obstacle is not detected it will close normally as designed. Once the door is closed, a message will be sent to the user to inform the user about the closing of the door and a record of the door closing time will also be stored in the cloud.

1. **Flow Chart for IoT based smart garage door opening and closing using keypad and LCD screen:** 

**Explanation:**

At the initial stage of the flow chart, when the vehicle will come to enter the garage, the keypad on the door will ask the password to open the door. If the password entered by the user is correct then the motor drivers will start operating through the relay module and the door will open but if the password will be incorrect then the LCD display will show the “ACCESS DENIED”. After this stage, all the information regarding the vehicle user and the timing will be saved in the cloud for future use or for security purposes as well. If the door will open for so long then the GSM module will activate and it will send the notification to the user through sms that the door is open, if not then a key is used to close the door. After that, the IR proximity sensor and the limit switch will start functioning and they will detect the obstacle at the time of closing the door. If they sense any obstacle, then the buzzer will start beeping and the door will remain open and then a button will be pressed to close the door again or the closing will be pressed. If there will be no object or obstacle then the door will close and all the data regarding the entry and exit of the object will be saved in the cloud storage. Gsm module will enable at the closing of the door as well, if there will be any suspicious activity that will happen to inform the user.

* **Milestones (Deliverables and Time Schedule) :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sno** | **Task Title** | **Start Date** | **End Date** | **Person In-charge** |
| 1 | Project Proposal | 27 jan 2021 | 12-feb-2021 |  |
| 2 | Deciding and ordering hardware components | 13-feb-2021 | 15-feb-2021 | Gopi Krishnan |
| 3 | Testing hardware components Part 1 | 16-feb-2021 | 22-feb-2021 | Raghav Chopra |
| 4 | Testing hardware components Part 2 | 16-feb-2021 | 22-feb-2021 | Manpreet Kaur |
| 5 | Designing Schematic Diagram | 23-feb-2021 | 26-feb-2021 | Shivinder Kaur |
| 6 | Interfacing beaglebone with logic level controller | 27-feb-2021 | 5-march-2021 | Dawinder Kaur |
| 7 | Connecting LL controller to WIFI Module ESP32 | 27-feb-2021 | 5-march-2021 | Shivinder Kaur |
| 8 | HTML coding for web page | 6-march-2021 | 10-march-2021 | Dawinder Kaur |
| 9 | Interfacing BBB with Motor Driver L298N and connecting it to DC motor | 11-march-2021 | 15-march-2021 | Gopi Krishnan |
| 10 | Interfacing of Limit Switch with BBB | 15-march-2021 | 19-march-2021 | Raghav Chopra |
| 11 | Interfacing BBB with IR sensor | 15-march-2021 | 19-march-2021 | Manpreet Kaur |
| 12 | Interfacing Buzzer with BBB | 20-march-2021 | 26-march-2021 | Raghav Chopra |
| 13 | Interfacing Button with BBB | 23-march-2021 | 28-march-2021 | Shivinder Kaur |
| 14 | Interfacing Cloud services with ESP32 | 28-march-2021 | 2-april-2021 | Dawinder Kaur |
| 15 | Interfacing Arduino with BBB | 28-march-2021 | 2-april-2011 | Gopi krishnan |
| 16 | Interfacing LCD display with arduino | 3-april-2021 | 9-april-2021 | Manpreet Kaur |
| 17 | Connecting Keypad with arduino | 3-april-2021 | 9-april-2021 | Raghav Chopra |
| 18 | Interfacing Arduino with GSM Module | 10-april-2021 | 16-april-2021 | Gopi Krishnan |
| 19 | PCB design using particular software | 10-april-2021 | 16-april-2021 | Manpreet Kaur |
| 20 | Zero PCB Implementation Part 1 | 17-april-2021 | 23-april-2021 | Shivinder Kaur |
| 21 | Zero PCB implementation Part 2 | 17-april-2021 | 23-april-2021 | Dawinder Kaur |
| 22 | Final Report | 24-april-2021 | 25-april-2021 |  |
| 23 | Final Presentation | 27-april-2021 | 28-april-2021 |  |

* **Ethical,Legal and Environment ramification:**

1. **Environment Ramifications:**

In our project, we are using so many components which somehow impacts the environment badly like

* The IR sensors, which emit the infrared rays that can cause harm to mankind and the birds life as well.
* We use a buzzer in the project if it beeps for a long time then it will create noise pollution if we use this component for so long.
* This project includes electrical components which include the minerals such as silicon, uses power often generated by the burning of fossil fuels that also affects the environment. When we purchase the components, the impact on minerals occurs that time.

This project has minimal impact on the environment unless the project gets mass produced. Large scale production of the devices does harm the landfills, consume more power and also affect the ecosystem surrounding those areas.

1. **Ethical Ramifications:**

* As this project is based on Iot, which means it can lead to cyber attacks which involves the stealing of personal information such as, timing of the entry and exit, location, data and so on.
* In this project we are using a keypad to enter the passcode for opening the door and if someone will try to use the keypad in order to access the garage which means that person disobey the ethics because one cannot use anything without the permission of the owner.
* As I described, it is based on Iot which means hackers can easily hack the whole system to get the information which is also an ethical issue because our project is connected with the wifi and it can be hackable.

Privacy is the main concern in all the Iot based projects like how can we protect the personal belongings. These are the ethical issues related to our project.

1. **Legal Ramifications:**

Legally, spying is a crime. If someone checks the password and tries to access the garage in the absence of that person then that person will face some legal terms. Because we cannot use anything without the permission or in the absence of the owner of the garage. Our project has minimal legal ramifications instead of environment and ethical ramifications.

* **Communication Standards for embedded system:**

1. **Interfacing Beaglebone black with ESP32 Wifi module**

The beaglebone black and NodeMCU ESP32 will use UART communication protocol.

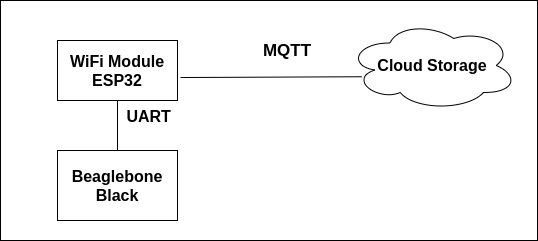
UART pin of beaglebone will be connected with ESP 32 Transmitter and receiver.

Beaglebone acts as master device and Node MCU esp32 act as slave device.

The communication between wifi module and web server is through HTTP protocol.

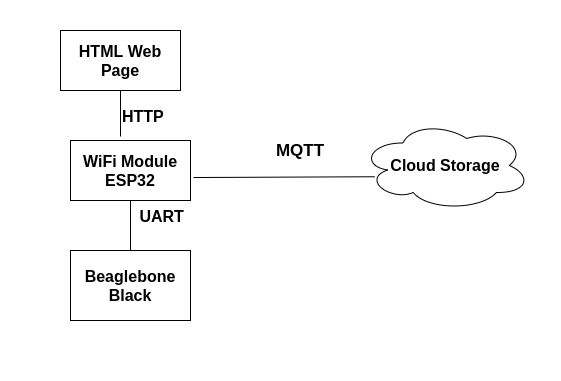
We use a logic level controller to convert 3.3V Vcc from beaglebone to 5V power to operate ESP32 modules.

NodeMCU ESP 32 is programmed using an Arduino IDE.

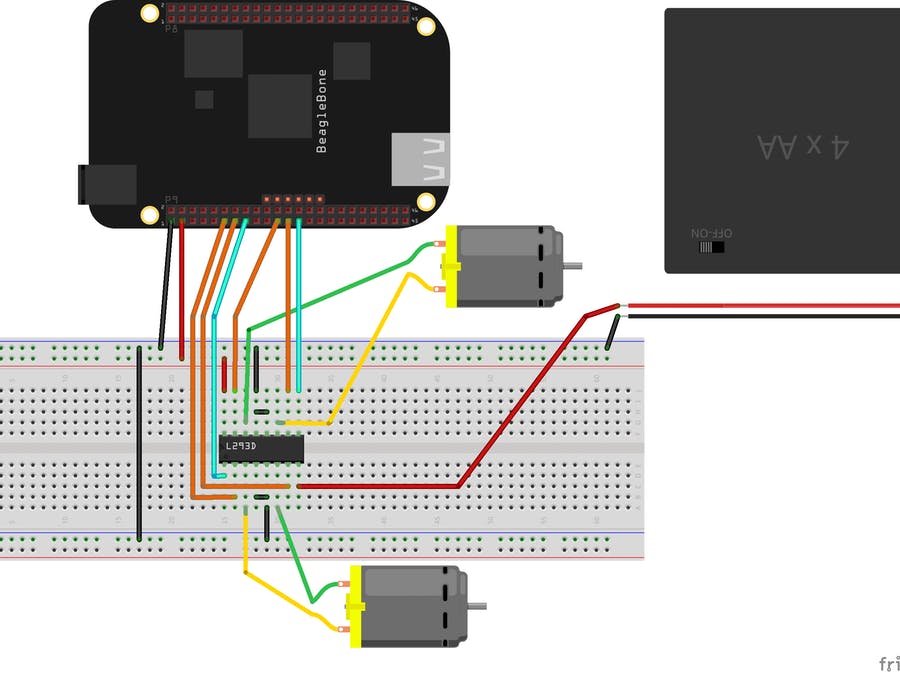


1. **Interfacing Wifi module with the cloud**

To enable the interfacing of the ESP32 wifi module to the cloud we need an MQTT broker. An MQTT broker is a server that receives texts from all the clients and re-routes them to their destination. The protocol used for communicating with the cloud via the ESP32 module is the IEEE802.11 protocol. It is a part of Local Area Network protocols for implementing Local Wireless Area Networks.



1. **Interfacing Beaglebone black with Relay Module H bridge**

****

The L298N is a dual H-Bridge motor driver IC which means that it is capable of driving two motors simultaneously. The GPIO pins are used to provide power to the motor but we also give external power supply because 3.3V (Vcc) power supply is not enough for the DC motor (6V - 12V) to turn on. Enable pins of L298N are made high to enable the motor. GPIO (15) pins of the P9 header are connected to the enable pin on the L298N IC. This pin needs to be high to enable the motor. UART pins (11 and 13) of the P9 header connected to the IN1 and IN2 pins on the IC. The DC motors that we are driving with the L298N are connected to the OUT pins of the IC.

1. **Interfacing Beaglebone black with Limit Switch or Micro switch**

Limit switches are used to detect the position of garage doors (open/close). Interfacing of the limit switch can be done on a GPIO pin.

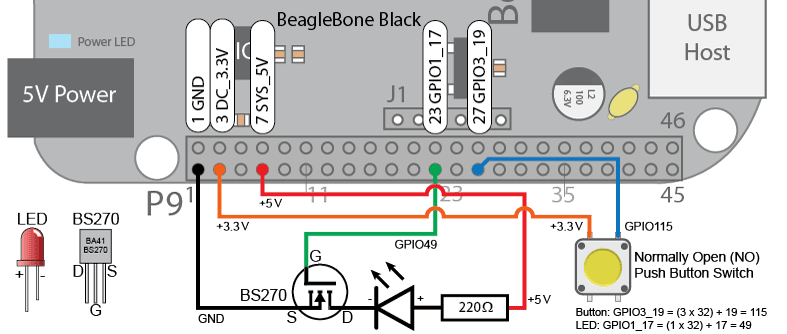
1. **Interfacing Beaglebone black with Buzzer**

Communication between Beaglebone Black and Buzzer is going to be through GPIO pins of beaglebone black. The protocol we will use for this interfacing would be i2c. We will directly connect the buzzer to one of the GPIO pins of beaglebone black and then we will set the GPIO pin as an output for the buzzer.

There are several GPIO pins on the beaglebone black on both ports P8 and P9. For example, if we consider P9 port of beaglebone black and suppose we are using pin 12 of beaglebone black which is GPIO\_60, to connect to the buzzer. Then we will use the following commands on the terminal going into the beaglebone to set the mode of GPIO\_60 as an output.

1. **Interfacing Beaglebone black with push button**

For this project we are using the push button as an interrupt for the buzzer. An interrupt is a signal that is sent to a microcontroller from an attached hardware device. , software application or circuit that an event has occurred which requires attention. Interrupts are high-priority conditions. For interfacing the push button Pulse Width Modulation(PWM) is being used. As shown in the diagram given below the GPIO pins will be used to connect the push button to the beaglebone black.

****

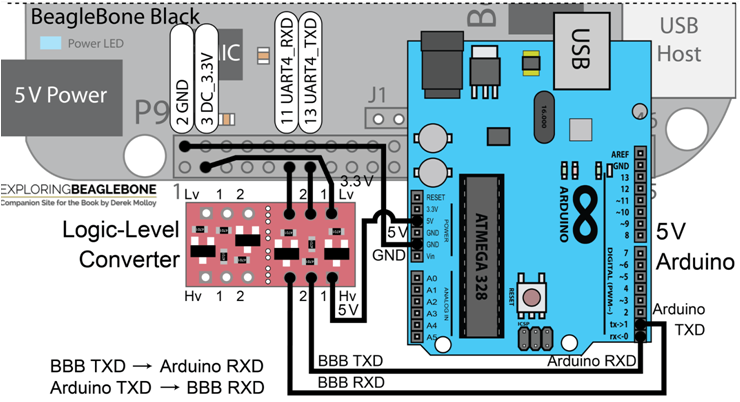
1. **Interfacing Beaglebone black with IR proximity sensor**

IR Sensors are Infrared sensors. We will connect ground(-) of IR sensor to ground pin of bbb and Vcc(+) of IR sensor to Vcc of bbb. The output pin of the IR sensor will be connected to the GPIO pin of bbb. The IR sensor gives digital output only in the form of 0 and 1.

When there is any object present the IR sensor will generate output 0 and otherwise 1. It gives digital output 0 for any obstacle detected and signal led will be on the ir sensor board. The protocol used here is digital GPIO.

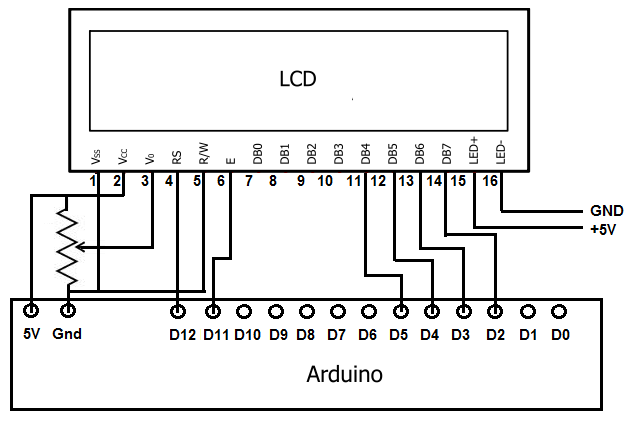
1. **Interfacing between Beaglebone black and Arduino**

Communication between the Beaglebone board and Arduino is through serial ports. It enables communication through a serial and common data structure. Communication between the Beaglebone board and Arduino is enabled through serial ports. The normal GPIO pins of Beaglebone need to be modified as UART pins for enabling this communication. Therefore the interfacing of Beaglebone Black and Arduino falls in Intersystem communication.



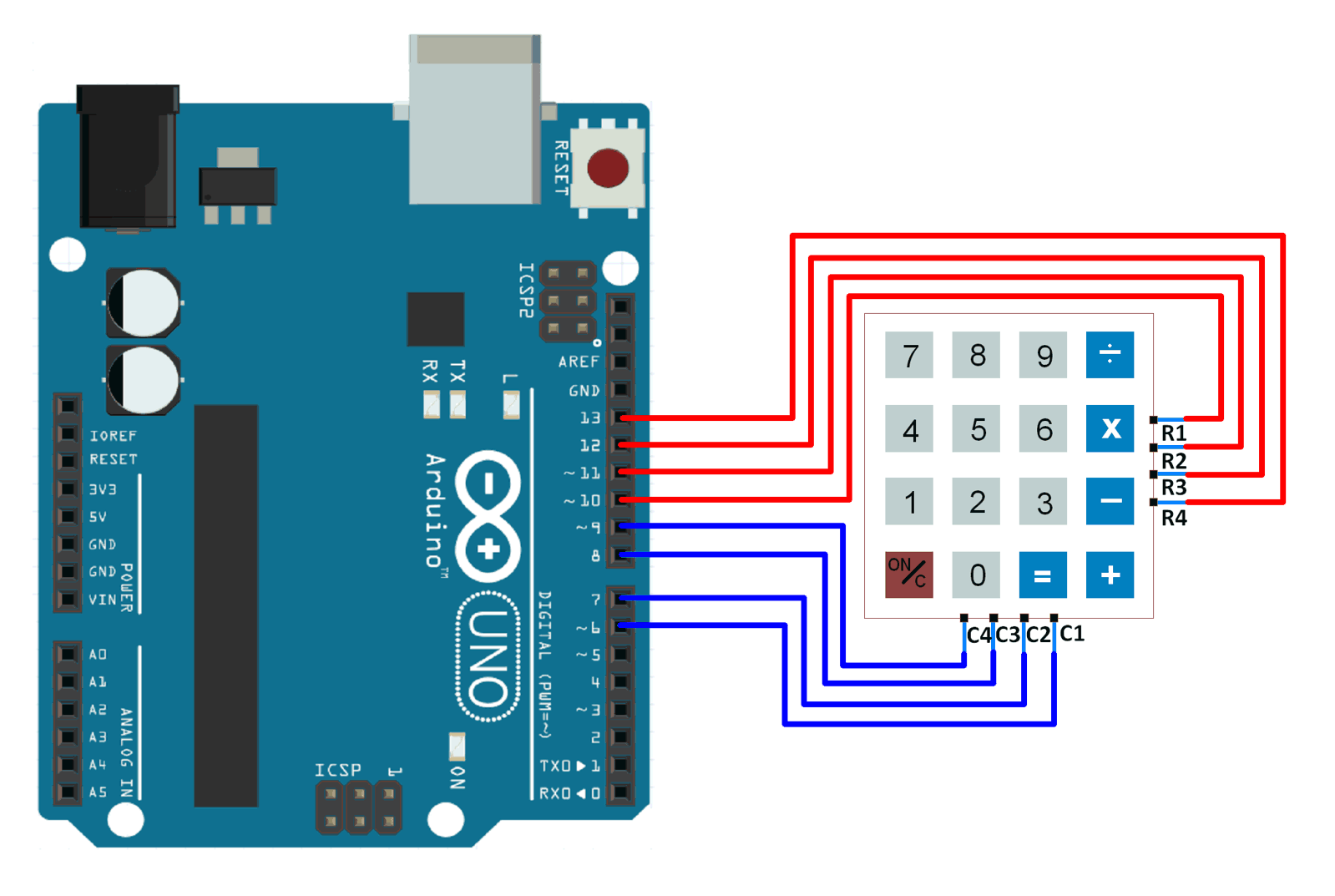
1. **Interfacing between Arduino and 20\*4 LCD Screen**

Interfacing of 20\*4 LCD screen with an Arduino board is done for displaying the reactions to user inputs like a password. The interaction between the Arduino board and the LCD screen is done via I2C communication protocol making it an Intra communication between two components.



1. **Interfacing Between Arduino and 4\*4 Matrix Keypad**

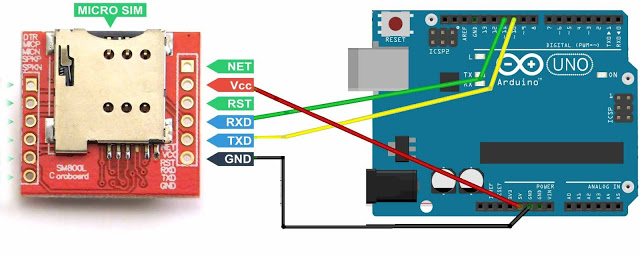
The Arduino and keypad use serial communication to interact with each other. GPIO pins are used to enable Intra serial communication between an Arduino Board and an input device like a keypad.



The coding for interfacing of 4\*4 keypad matrix with the Arduino is done in C in the Arduino IDE.

1. **Interfacing between Arduino and GSM Module**

Serial Communication Protocol is used for communication once the SIM800L GSM module is interfaced with Arduino Uno. Here we will use GSM protocol.



The coding standard used for interfacing the ESP32 GSM module with Arduino is C in the Arduino IDE.

* **Coding Standards of Embedded System:**

1. **Coding Standards of Embedded System (C,C++):**

Coding Standards are a set of rules followed by programmers working on Embedded firmware designs Communication protocol for embedded systems tends to deliver mostly short, periodic messages.

**MISRA**® is a set of C and C++ coding standards, developed by the Motor Industry Software Reliability Association (MISRA).

It is widely used by embedded industries — including aerospace and defense, telecommunications, medical devices, and rail.

Most of these industries have a compliance requirement to use a coding standard — such as ISO 26262 for automotive functional safety.

There have been three releases of the MISRA C standard :

* MISRA C:1998 MISRA C:1998 was published in 1998 and remains widely used today. It was written for C90. There are 127 coding rules, including: The statement forming the body of an "if", "else if", "else", "while", "do ... while", or "for" statement shall always be enclosed in braces
* MISRA C:2004 MISRA C:2004 is the second edition of MISRA C, published in 2004. It was written for C90. There are 142 coding rules, including: if (expression) construct shall be followed by a compound statement.
* MISRA C:2012 MISRA C:2012 is the third edition of MISRA C, published in 2012. It was written for C99. There are 143 rules, including: A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand.

Version of MISRA C++ rules available today:

MISRA C++2008 MISRA C++:2008 was published in 2008. It was written for C++03. There are 228 coding rules, including: The condition of an if-statement and the condition of an iteration statement shall have type bool.

1. **Coding Standards of embedded system (HTML):**

HTML is a Hyper Text Markup Language. It is the standard markup language for resting Webpages. It describes the structure of a Webpage consisting of a series of elements.

It was initially developed by W3C and WHATWG in 1993. W3C mainly checks for standards of HTML and other stuff. Since the initial release, lots of versions of HTML have been developed. Initially, its version is HTML 1.0 or simply HTML.

Different versions of HTML:

1. HTML 1.0
2. HTML 2
3. HTML 3.2
4. HTML 4.01
5. HTML 5

HTML 5 is the latest version of HTML language. It was started to use in 2014 and it came up with lots of HTML tags support. It provided support for new form elements like input elements of different types, geolocations, support tags, etc.

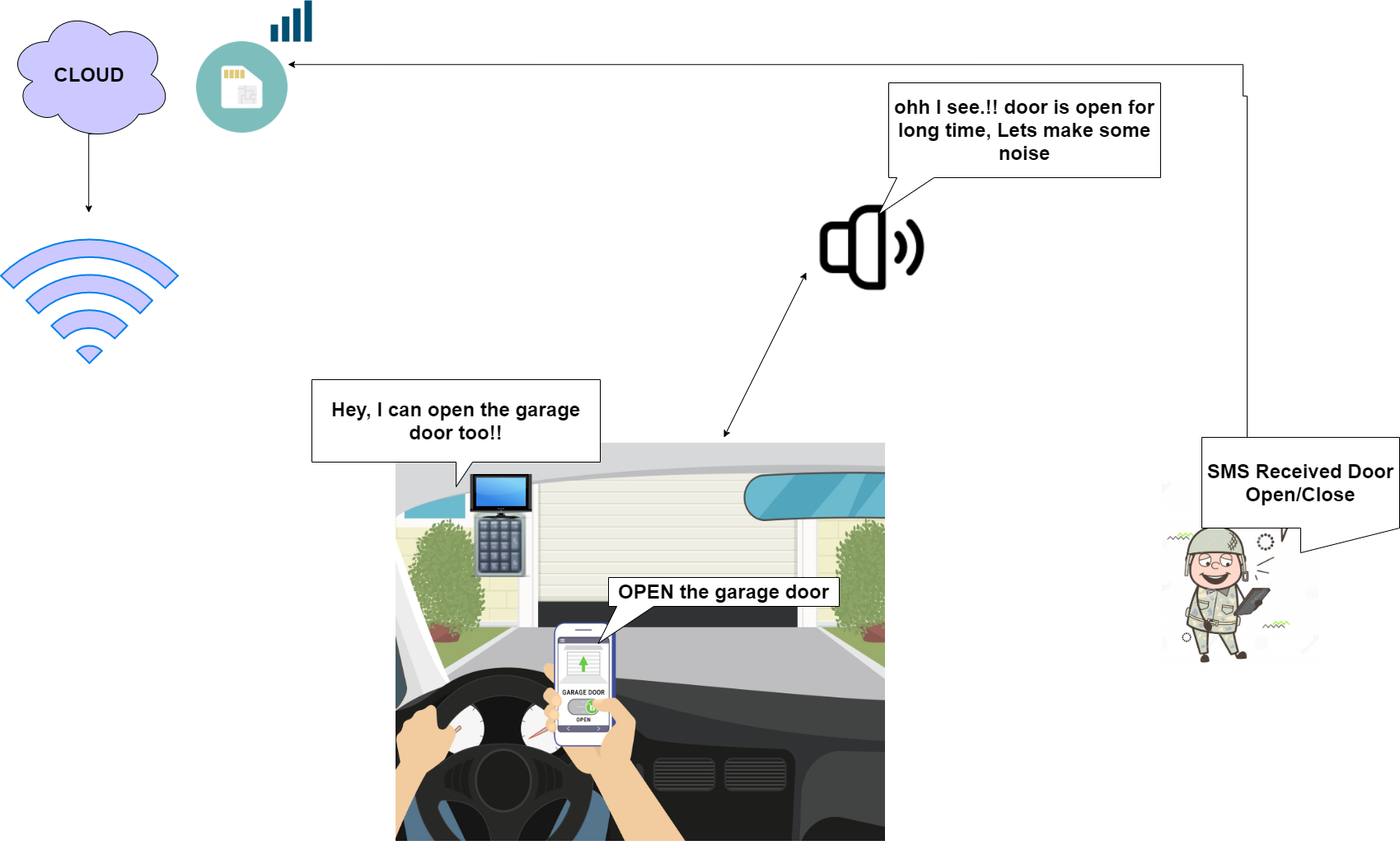
The new tags that has been added to HTML 5 are:

* Email
* Password
* Audio tag
* Semantic tags
* Section tag

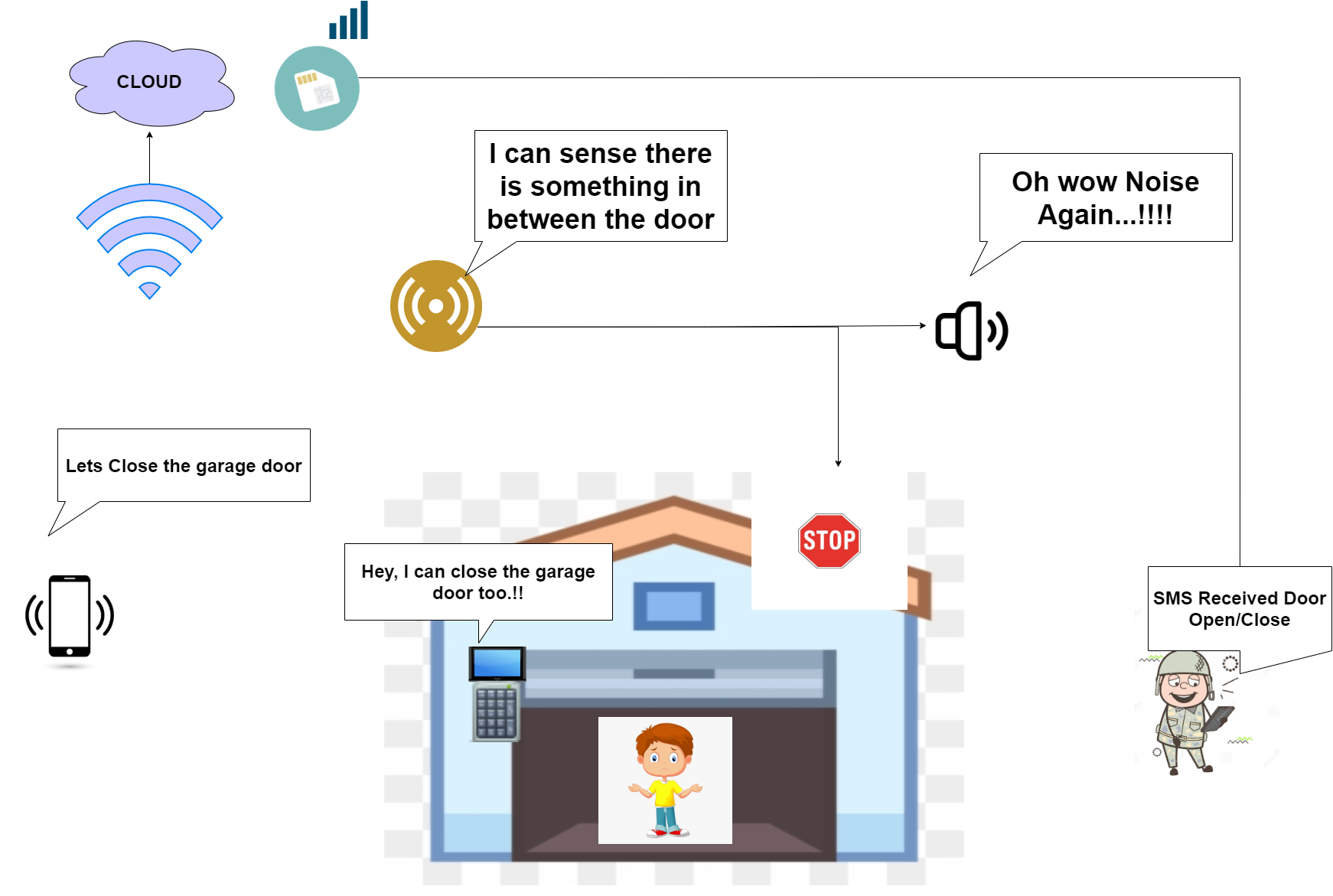
There is a tool known as W3C HTML Validator that has been used to maintain or check whether a piece of HTML tag or HTML is valid or not. An HTML validator follows the standards of W3C to validate an HTML page. It follows the W3C standards.

* **Engineering Drawing of Project:**

1. **Engineering drawing of the project (while the door will open)**

****

1. **Engineering drawing of the project (while the door will close)**

****

* **Bill of Material table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **BILL OF MATERIALS (BOM) table** | | | |
| **S.no.** | **Name of the Component** | **Part number** | **How it will help in our project** | **From where we have ordered it** |
| 1. | Beaglebone Black | BBB01-SC-505-ND | Beaglebone is the main MCU of the project and is the heart of the project. All the devices will eventually be connected to this device. | Digikey Electronics,  BEAGLEBONE BLK REV C AM3358BZCZ  <https://www.digikey.ca/en/products/detail/BBB01-SC-505/BBB01-SC-505-ND/6210999?itemSeq=354364471> |
| 2. | Arduino | 1050-1024-ND | Used to interface lcd, keypad and GSM to beaglebone black | Digikey Electronics, ARDUINO UNO R3 ATMEGA328P EVAL  <https://www.digikey.ca/en/products/detail/arduino/A000066/2784006?s=N4IgTCBcDaIIwAYCsCC0iwBZUDkAiIAugL5A> |
| 3. | L298N Motor Driver Controller | IS017 | It will be used to connect a motor to beaglebone, thus whenever the beaglebone gives output to open or close this will direct the dc motor to do so. | Amazon,  PChero 2Packs L298N Motor Drive Controller Board Module, Dual H Bridge  <https://www.amazon.ca/dp/B07C4B3DL4/ref=cm_sw_r_wa_apa_i_4E62PGXXSWMMZ6EYZNCE?psc=1> |
| 4. | DC 6V Gear Motor | Walfront67vb8ogpqz-04 | It will be interfaced with the motor driver and will open and close the garage door. | Amazon,  DC 6V Gear Motor with Long M3\*55MM Lead Screw Thread Output Shaft 30/60/100/150/200/300/400/500RPM(6V 150RPM)  <https://www.amazon.ca/dp/B07552PJPC/ref=cm_sw_r_wa_apa_i_QW44KT7GY6A20N7KX0K6> |
| 5. | Microswitches | iS008 | It will be connected to the garage door directly which will detect the opening and closing of the gate. | Amazon,  PChero 5pcs Microswitches Mechanical Endstop Switches with 70cm Cable RAMPS 1.4 for 3D Printer RepRap Prusa Mendel Makerbot Arduino Mega 2560 CNC  <https://www.amazon.ca/dp/B071JV158X/ref=cm_sw_r_wa_apa_i_T4SVA1ZN2XH2E1WKM99N> |
| 6. | Wifi Module ESP32 | 3320 | We are using Beaglebone black and it does not have an in built wifi module therefore, we are using this module to access wifi services. | Elmwood electronics,  ESP32 WIFI-BT-BLE MCU MODULE / ESP-WROOM-32  <https://elmwoodelectronics.ca/products/esp32-wifi-bt-ble-mcu-module-esp-wroom-32?variant=33713585283> |
| 7. | Logic level Controller | 1568-1209-ND | Logic Level Controller will be interfaced with the arduino and wifi module to change the voltage level they are receiving from the beaglebone black. | Digikey Electronics,  LOGIC LEVEL CONVERTER - BI-DIREC  <https://www.digikey.ca/en/products/detail/sparkfun-electronics/BOB-12009/5673795?s=N4IgTCBcDaIIwFYBsAOAtHMAGAnGgdgCYgC6AvkA> |
| 8. | IR Sensors | Ek1254 | It will help us to detect obstacles if any in between the door | Amazon,  Gikfun IR sensor Obstacles Avoidance module, Case of 5pcs  <https://www.amazon.ca/dp/B07FFM7DYQ/ref=cm_sw_r_wa_apa_i_0MFESTK9QPSRGT214K7N> |
| 9. | Buzzer | CLT1036 | It will help to alert the user by beeping if the door remains open too long and if there is any obstacle in between while the door is closing. | Amazon,  Cylewet 5v active buzzer , pack of 10pcs  <https://www.amazon.ca/dp/B07DDK92F5/ref=cm_sw_r_wa_apa_i_988HAS3XMSSDS9VYRQG9> |
| 10. | Push Buttons | E370-CA | This push button is used to disable the buzzer in case the buzzer goes off for a long period of time and is not required. | Amazon,Gebildet 10 Pcs Waterproof Push Button Momentary Stainless Steel Push Button On Off 12mm 2A 12V/24V/125V/250V(High Head) <https://www.amazon.ca/Gebildet-Waterproof-Button-Momentary-Stainless/dp/B08671VD54/ref=sr_1_2?dchild=1&keywords=push+button&qid=1613076018&sr=8-2> |
| 11. | GSM,GPRS, SMS Module | YWBL-WHxiugqvk870 | It will be used to send notifications to user via SMS of door open and close | Amazon,  GSM Module  <https://www.amazon.ca/dp/B07WZTXBSX/ref=cm_sw_r_wa_apa_i_3E9ADJAP9J1WT8PAS84Y> |
| 12. | 4\*4 Matrix keypad | B086ZGKDK9 | It will be used to enter a passcode to open and also to close the garage door.It will be interfaced with Arduino. | Amazon,  DIYmalls 4x4 Matrix Membrane Keypad 16 Keys Keyboard Module Array Switch for Arduino UNO R3  <https://www.amazon.ca/dp/B086ZGKDK9/ref=cm_sw_r_wa_apa_i_79TG40BP7MPFDC5C7MCA?_encoding=UTF8&psc=1> |
| 13. | 20\*4 Lcd display | WYTP22004YG | It will be used to display a message to the user to enter a passcode to open the garage door. It will be interfaced with Arduino. | Amazon,  WayinTop 20x4 2004 LCD Display Module with IIC/I2C/TWI Serial Interface Adapter for Arduino for Mega 2560 (Yellow Green/2004)  <https://www.amazon.ca/dp/B07TXBV8MS/ref=cm_sw_r_wa_apa_i_40E5B0C5NGJZXYPYCCMW?psc=1> |
| 14. | DC 5V 4A Power Adapter | 237-1385-ND | Power Adapter is use to provide power supply to the beaglebone black | Digikey Electronics,  AC/DC WALL MOUNT ADAPTER 5V 10W  <https://www.digikey.ca/en/products/detail/triad-magnetics/WSU050-2000/3094911?s=N4IgTCBcDa4MwHYC0BGOAOArEgcgERAF0BfIA> |
| 15. | Soldering Kit | B075KLBQCV | It is used to mount the components on PCB. | Amazon,Soldering Iron, Soldering Iron Kit Electronics, 60W Adjustable Temperature [https://www.amazon.ca/dp/B075KLBQCV/ref=cm\_sw\_r\_wa\_api\_i\_7TTFWQ81XFD5Y](https://www.amazon.ca/dp/B075KLBQCV/ref=cm_sw_r_wa_api_i_7TTFWQ81XFD5YAZY15D9)  [AZY15D9](https://www.amazon.ca/dp/B075KLBQCV/ref=cm_sw_r_wa_api_i_7TTFWQ81XFD5YAZY15D9) |
| 16. | Multimeter | B08H8F3YGP | Multimeter is used to check the flow of current and also to check the voltage. We will mainly use it to test the hardware components. | Amazon,XL830L Pocket Digital Multimeter Mini Voltage Tester Home Measuring Tools Multi-Tester Test AC/DC Current, Resistance, Continuity, Frequency Backlight LCD Display with Battery <https://www.amazon.ca/dp/B08H8F3YGP/ref=cm_sw_r_wa_apip_Cm22MKA3Y4xoY> |
| 17. | Sandisk 16 GB Sd Card | SDSQUAR-016G-GN6MA | This will help us to flash latest image of debian | Amazon,SanDisk 16GB Ultra microSDHC UHS-I Memory Card with Adapter - 98MB/s, C10, U1, Full HD, A1, Micro SD Card<https://www.amazon.ca/dp/B073K14CVB/ref=cm_sw_r_wa_apa_i_RRHWKXZC25VFZ82R6NS2> |
| 18. | 22 AWG Silicone hook up wires | B07T4SYVYG | It is used to connect the sensor, buzzer and other components to the main unit. | Amazon,  22 AWG Silicone Hook Up Wire - 22 Gauge Stranded Tinned Copper Wire with Silicone Insulation, 6 Colors (Black, Red, Yellow, Green, Blue, White) 23ft / 7m Each, Hook Up Wire Kit from Plusivo  <https://www.amazon.ca/dp/B07T4SYVYG/ref=cm_sw_r_wa_apa_i_EVVSF849VAEVAA1NXRTQ> |
| 19. | Gikfun 5mm LED light assorted kit | LYSB01ER728F6-ELECTRNCS | To notify the open/close status of the door. | Amazon.  Gikfun 5mm LED Light Assorted Kit DIY LEDs for Arduino (Pack of 100pcs) EK8437  <https://www.amazon.ca/dp/B01ER728F6/ref=cm_sw_r_wa_apa_i_DB50XF0SM7JPTH92CDK8?_encoding=UTF8&psc=1> |
| 20. | PCB Board | LP-23HM-X595 | Used to solder components together and make the project more presentable. | Amazon,  YUNGUI PCB Prototype Circuit Board Universal Single-Sided Circuits Boards for Electronic Project and DIY Soldering with 4 Sizes  <https://www.amazon.ca/dp/B07TTD2HXM/ref=cm_sw_r_wa_apa_i_G3TJ5F7WB37YX3T5ZB99?_encoding=UTF8&psc=1> |
| 21. | Resistors | B0872Y9RXH | It is used to limit or resist the flow of current or voltage through the circuit. | Amazon, LUTER 850pcs 30 Values Resistor Kit 1% 0 Ohm-1M Ohm 1/4W Metal FilmResistors Assortment with a Plastic Box for DIY, Experiment <https://www.amazon.ca/dp/B0872Y9RXH/ref=cm_sw_r_wa_apip_9BQ5zzuVO7xdM> |
| 22. | Breadboard | B0002H4W0K | Breadboard will be used to test the circuit that everything is proper before it is mounted or fixed on PCB | Amazon, Elenco Breadboard-Prototype Design Aid 9440 <https://www.amazon.ca/dp/B0002H4W0K/ref=cm_sw_r_wa_apip_NNGtFsoSx64WW> |
| 23. | SparkFun Beginners Part Kit | B01N1WJ3Z1 | This kit contains various capacitors, diodes and other material which we may need for our project. | Amazon, SparkFun Beginner Parts Kit <https://www.amazon.ca/dp/B01N1WJ3Z1/ref=cm_sw_r_wa_apip_S77JfKjKkdRjA> |

* **Debug Cycle implementation:**

Debugging is the process of finding bugs or errors in the computer program or the developed software. For the proper functioning and successful implementation of the project, It is essential to get the code running, in turn, it is to be free from errors. Most of the popularly used IDEs in embedded system designing these days have an inbuilt compiler, linker, assembler, and debugger which helps in getting the project working. Among all these unavoidable software development tools debuggers point to the errors and line of the code where the error is found, thus it makes the rectification of the errors relatively easy.

There are various approaches during the edit, test and debug cycle of the development process:

1. Using a target system
2. Using an emulator for target system
3. Using a simulator for hardware
4. Using IDE’s
5. **Target Machine Terminal -nano**

We will also directly write code in the terminal nano file on the laptop

and then after execution and debugging of code on linux machine we will transfer the file to beaglebone black using sftp transfer protocol. In this we will use debugger and compiler compatible with beaglebone black and linux OS such as ARM gdb debugger.

1. **Integrated Development Environment(IDE)**

An IDE or Integrated Development Environment is a software application that enables the most popular software development tools like source code editor, automation tools, debugger, etc for the programmer in one place. It enables modifiable, flexible, and non-erroneous programming which is very critical in developing embedded system applications.

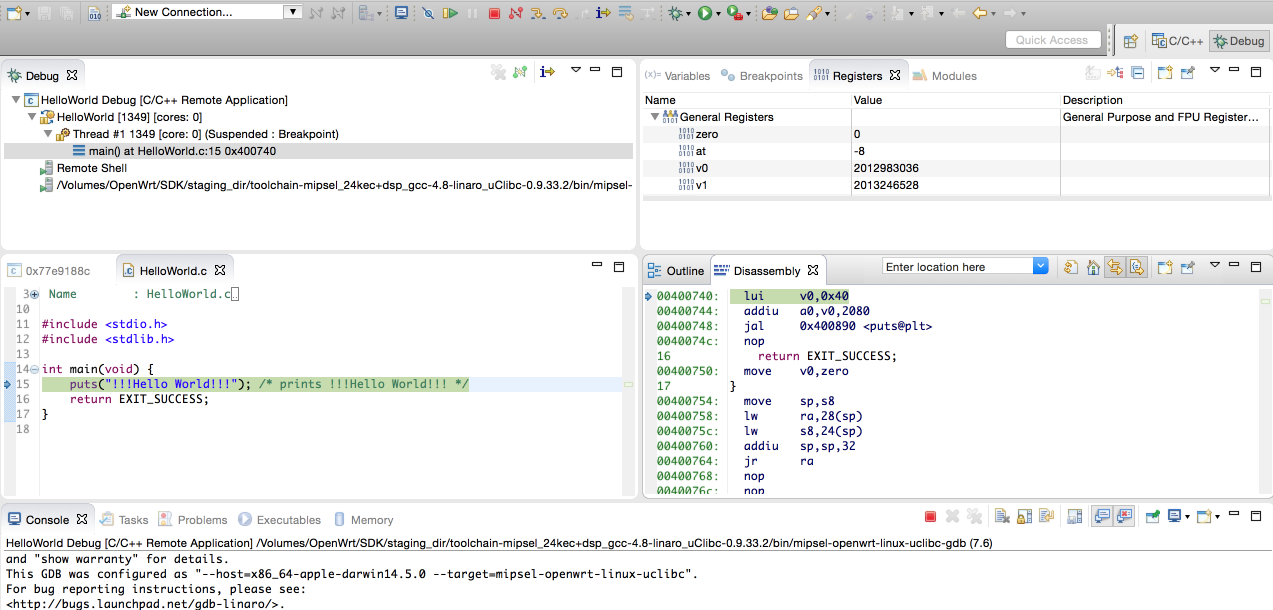
* **Eclipse IDE**
* **Arduino IDE**

**Eclipse IDE**

Eclipse IDE is the most popular IDE used in the field of embedded system application development. It enables programming in almost all the programming languages used in the software field, predominantly C/C++, JAVA, Python, Ruby, and many more.

We will use eclipse ide for debugging and execution of C/C++ code. Debugging of code can be done by using eclipse cross toolchain (GCC) debugger. As we are using linux OS and beaglebone black these both are compatible with gdb debugger.

Building and Debugging in Eclipse IDE



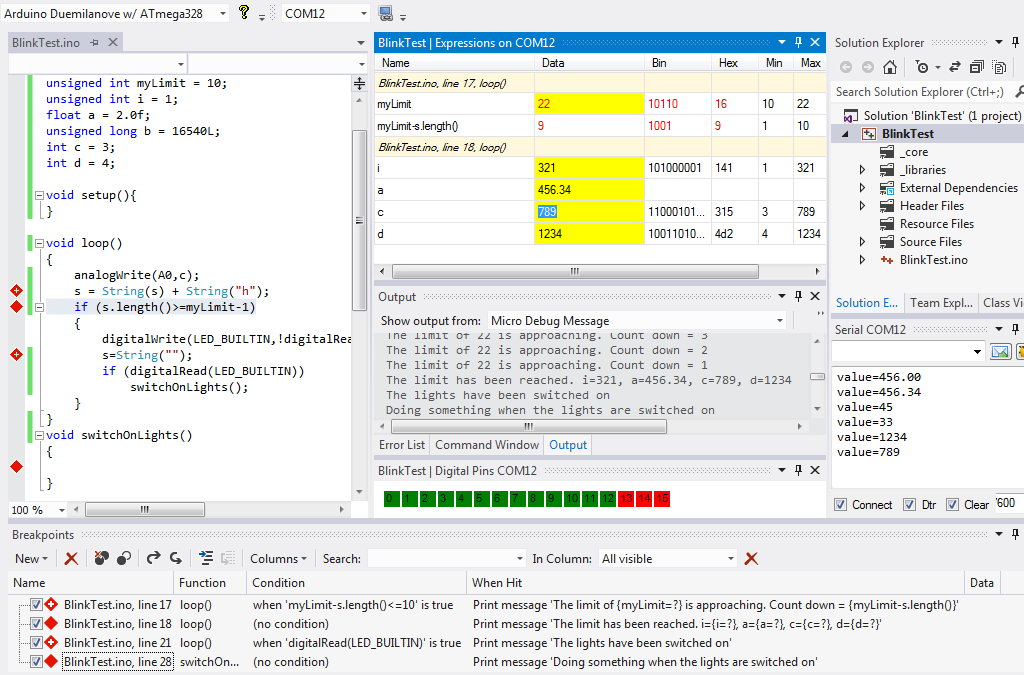
**Arduino IDE**

Arduino Integrated Development Environment is another approach used in embedded systems. It is the cross-platform application for Windows, Linux, and macOS that is written in the functions from C and C++. Arduino doesn’t have an onboard debugger, unlike the other software.

Functions of Arduino IDE:

* It is used to write and upload programs to Arduino compatible boards.
* It converts the executable code into a text file that will upload to the Arduino board by the loader program.
* It has inbuilt libraries and it is possible to import the libraries and expand them.

Debugging in Arduino IDE:



1. **Emulator**

An emulator typically enables the host system to run the software. The programmer uses the emulator to load the programs into the embedded system and run them step by step. In our project, we are not using any emulator for this project as beaglebone black and Ubuntu OS both using linux distribution and getting an emulator for linux isdifficult.

1. **Simulator**

Code is tested for the system by simulating it on the host machine that is used for the code development. It’s main function is to monitor the detailed information of the source code as the execution goes on step by step.As there are no libraries available for simulation of beaglebone black. So we are not using a simulator. We are using arduino for transmit and receive so we don’t require any simulator software as well.

**References:**

1. Retrieved from IoTDesignPro (2019, Nov 07) *IoT Wi-Fi Garage Door Opener using NodeMCU ESP12E* <https://iotdesignpro.com/projects/iot-based-garage-door-opener>
2. Retrieved from PROJECT HUB (2018, Oct 27) *RFID and Keypad Based Door Lock Using Arduino* [https://create.arduino.cc/projecthub/muhammad-aqib/rfid-and-keypad-based-door-lock-using-arduino-89e1d5?ref=tag&ref\_id=home%20security&offse=6](https://create.arduino.cc/projecthub/muhammad-aqib/rfid-and-keypad-based-door-lock-using-arduino-89e1d5?ref=tag&ref_id=home%20security&offset=6)
3. Retrieved from *Robu.in*, Tuesday May19,2020, Priyanka D. URL: <https://robu.in/ir-sensor-working/>
4. Retrieved from *IndustrialShields.com*, 2020, URL:<https://www.industrialshields.com/blog/arduino-industrial-1/post/how-to-use-the-gprs-module-sim800l-library-129>
5. Retrieved from *Cuidevices.com, basics technologies and buzzers and tone driving circuits,* Bruce R,2021 URL:<https://www.cuidevices.com/blog/buzzer-basics-technologies-tones-and-driving-circuits>
6. Retrieved from *Lastminuteengineers.com, Arduino keypad tutorial*, 2021, URL: <https://lastminuteengineers.com/arduino-keypad-tutorial/>
7. Retrieved from *lastminuteengineers.com L298N with DC Motor* ,2021 <https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/>
8. Retrieved from *thomas.net, Instruments control limit switche*s, 2021 URL: <https://www.thomasnet.com/articles/instruments-controls/limit-switches/>
9. Retrieved from *Instructables.com, interfacing lcd 20\*4 display with Arduino*,URL: <https://www.instructables.com/Interfacing-20x4-LCD-with-Arduino/>
10. Retrieved from *Industrialshields.com, module esp32,* 2020, URL: <https://www.industrialshields.com/blog/arduino-industrial-1/post/wifi-bluetooth-module-esp32-133>
11. Retrieved from TEPTechnopreneur (2016, Nov 13) *6 Necessary Embedded Tools* URL:[https://www.theengineeringprojects.com/2016/11/necessary-embedded-tools.](https://www.theengineeringprojects.com/2016/11/necessary-embedded-tools.html#:~:text=These%20embedded%20tools%20include%20editors%2C%20compilers%2C%20assemblers%2C%20debuggers%2C,the%20embedded%20tools%20are%20necessary%20and%20work%20together)
12. Retrieved from Robin Mitchell (2018, Nov 05) *An introduction to circuit design using Fritzing* URL:<https://maker.pro/custom/tutorial/an-introduction-to-circuit-design-with-fritzing>
13. Retrieved from *educba.com, What is eclipse IDE?* URL:<https://www.educba.com/what-is-eclipse-ide/>
14. Retrieved from *filehippo.com*(2019, Jan 30) URL:<https://filehippo.com/download_tera_term/?ex=CORE-190.2>
15. Retrieved from *github.com* (2020, Sept 24) URL:<https://github.com/arduino/arduino-pro-ide>
16. Retrieved from *geeksforgeeks.org* (2019, Aug 26) *Difference between Compiler and assembler* URL:<https://www.geeksforgeeks.org/difference-between-compiler-and-assembler/>
17. Retrieved from *geeksforgeeks.org*(2019, Nov 21) *Linker* URL:<https://www.geeksforgeeks.org/linker/>
18. Retrieved from *wikipedia, Arduino IDE* URL:<https://en.wikipedia.org/wiki/Arduino_IDE>
19. Retrieved from *educba.com, What is eclipse IDE?* URL: <https://www.educba.com/what-is-eclipse-ide/>
20. Retrieved from *electronics-notes.com, USB operation: Protocol, Data transfer & packets,* URL <https://www.electronics-notes.com/articles/connectivity/usb-universal-serial-bus/protocol-data-transfer.php>
21. Retrieved from *teachmemicro.com*, *BEAGLEBONE BLACK SERIAL CONNECTION TO ARDUINO,* URL <https://www.teachmemicro.com/beaglebone-black-serial-arduino/>
22. Retrieved from *Trevors Electronic Blog*, *How to get serial UART port working on the beaglebone black.* URL<https://electronics.trev.id.au/2018/02/09/get-uart-serial-ports-working-beaglebone-black/>
23. Retrieved from *electronics-lab.com, Using a 20\*4 I2C Character LCD With Arduino UNO.* URL <https://www.electronics-lab.com/project/using-20x4-i2c-character-lcd-display-with-arduino-uno/>
24. Retrieved from *instructables.com, Interfacing 20\*4 LCD with Arduino.* URL: <https://www.instructables.com/Interfacing-20x4-LCD-with-Arduino/>
25. Retrieved from *robu.in*(2020, Jan 9), *Interfacing Sim 800L with Arduino.* URL: <https://robu.in/sim800l-interfacing-with-arduino/>
26. Retrieved from *github.com*(2020, Nov 17), *Arduino-SIM800L- driver* URL: <https://github.com/ostaquet/Arduino-SIM800L-driver>
27. Retrieved from *electronicwings.com*, *4\*4 Keypad interfacing with Arduino Uno* URL: <https://www.electronicwings.com/arduino/4x4-keypad-interfacing-with-arduino-uno>
28. Retrieved from *instructables.com, Connect ESP32 to AskSensors over https*. URL: <https://www.instructables.com/How-to-Connect-an-ESP32-to-the-IoT-Cloud/>
29. Retrieved from *derekmolloy.ie*, *Writing a Kernel Linux module- Part 3: Buttons and LED* URL: <http://derekmolloy.ie/kernel-gpio-programming-buttons-and-leds/>
30. Retrieved from beagleboard.org (2016,June 5) URL: <https://beagleboard.org/p/hoffmanjon/controlling-a-motor-with-an-h-bridge-fd13b2>
31. Retrieved from, learn.adafruit.com, Using an ir sensor URL:<https://learn.adafruit.com/ir-sensor/using-an-ir-sensor>
32. Retrieved from instructables.com (2018, May 6) *ESP32: Internal Details and Pinout* <https://www.instructables.com/ESP32-Internal-Details-and-Pinout/>
33. Retrieved from Perforce.com (2021) <https://www.perforce.com/resources/qac/misra-c-cpp#why>
34. Retrieved from steemit.com (2018) *Interfacing SIM 800l GSM Module with Arduino* <https://steemit.com/utopian-io/@kimp0gi/interfacing-sim-800l-gsm-module-with-arduino>
35. Retrieved from w3schools.com, HTML introduction URL: <https://www.w3schools.com/html/html_intro.asp>
36. Retrieved from educba.com, Versions of HTML, URL: <https://www.educba.com/versions-of-html/>