Electronics Workshop – I : Final Evaluation

MORSE CODE ENCODER AND DECODER

Sricharan Vinoth Kumar(2024112022)
Saikiran S(2024112007)

Team 23

TA: Vedant Nipane

PROBLEM STATEMENT

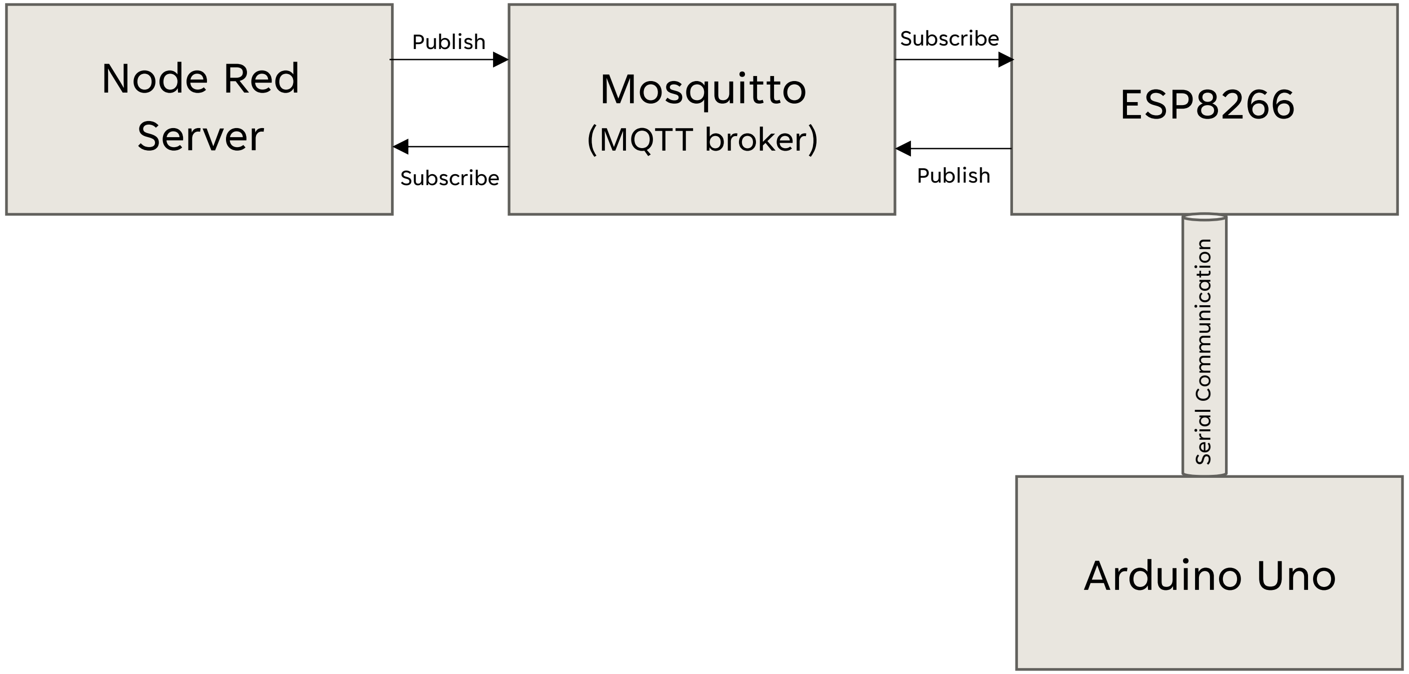
To use Arduino to implement a Morse code encoder and decoder obtaining inputs in various forms and displaying output in various forms.

MOTIVATION AND BACKGROUND

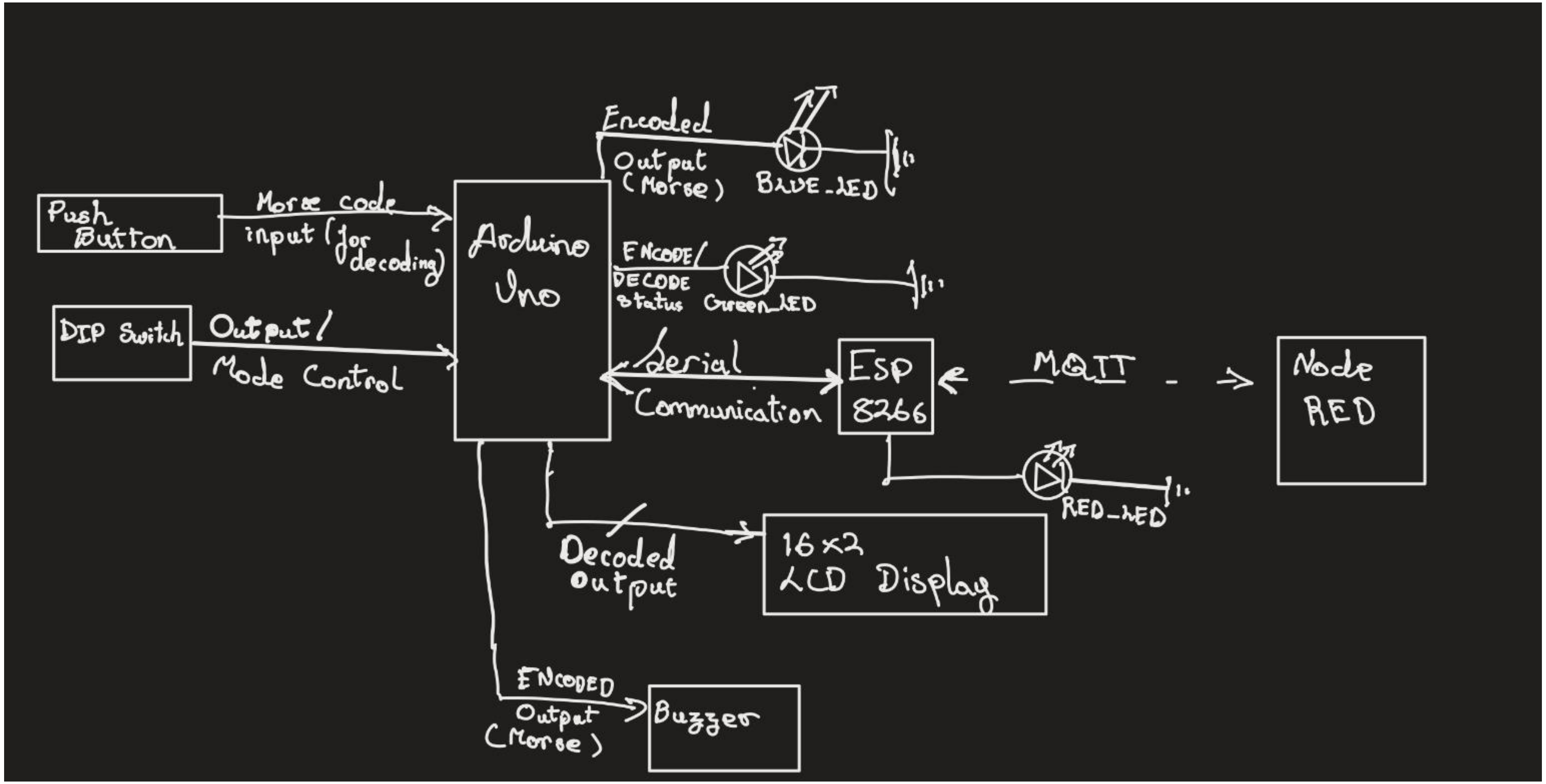
- Morse code is one of the earliest forms of digital communication. Our project aims to culminate this and new developments such as IOT under one project.
- Although there is no practical purpose, this device can be used to communicate within different devices of a large building similar to an intercom.

PROJECT OVERVIEW AND CIRCUIT DESIGN

MQTT AND MOSQUITTO - A SIMPLE OVERVIEW



PROJECT DESIGN – BLOCK DIAGRAM



PROJECT DESIGN - WORKFLOW

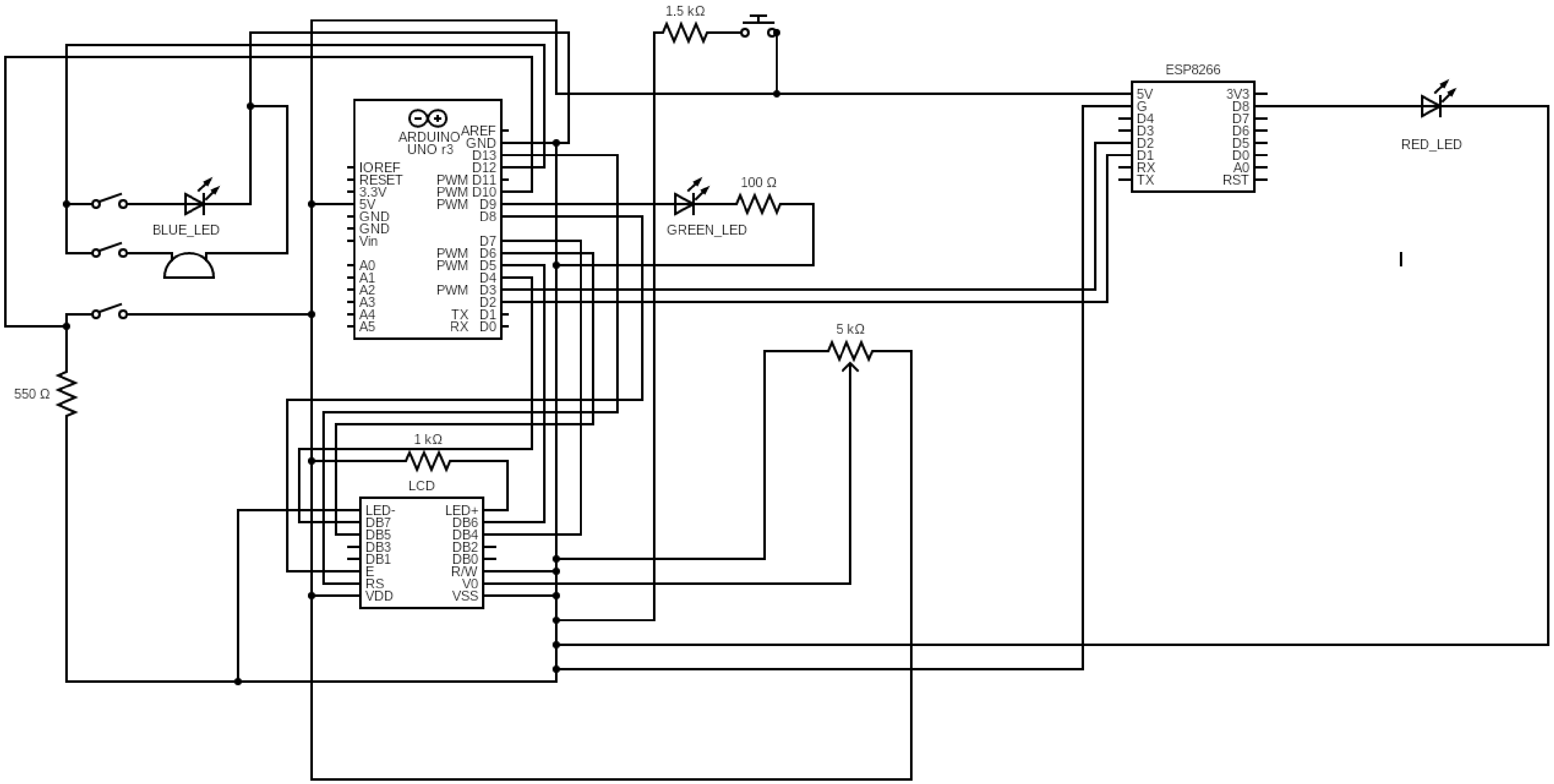
- The user through the dipswitch(5) gets the option to choose between the two modes – encoding and decoding.
- If the user chooses to use the encode a given text to morse code:-
    1. User gets the option to choose the modes of output (Buzzer and LED) through the dipswitch (1 and 2 respectively).
    2. User enters the text to be encoded through node-red dashboard.
    3. The text to be encoded is sent to the Arduino through the ESP module.
    4. The encoded output is visible on the chosen mode of output(s).

PROJECT DESIGN - WORKFLOW

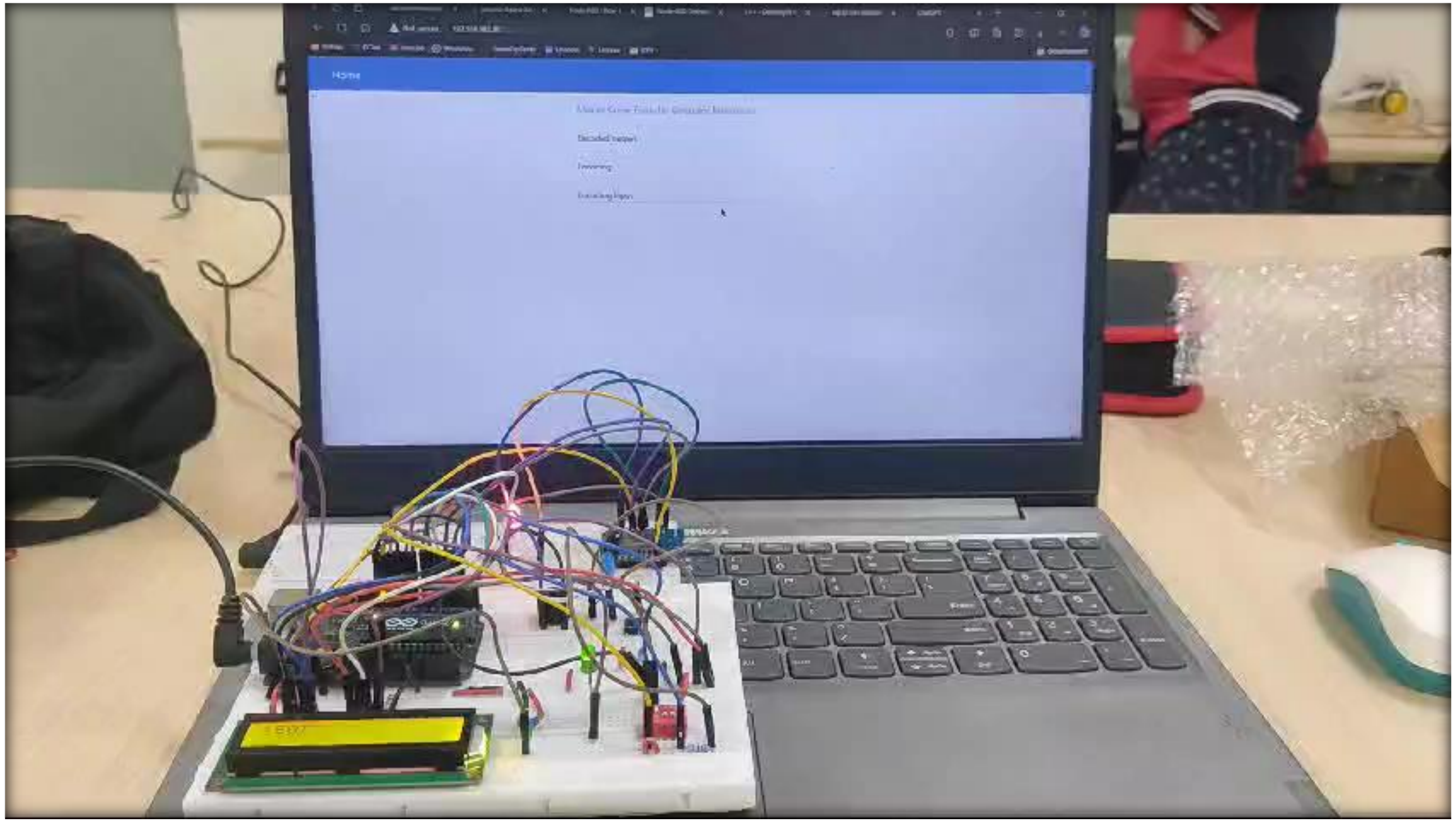
- If the user chooses to give morse input through pushbutton and wishes to decode:-

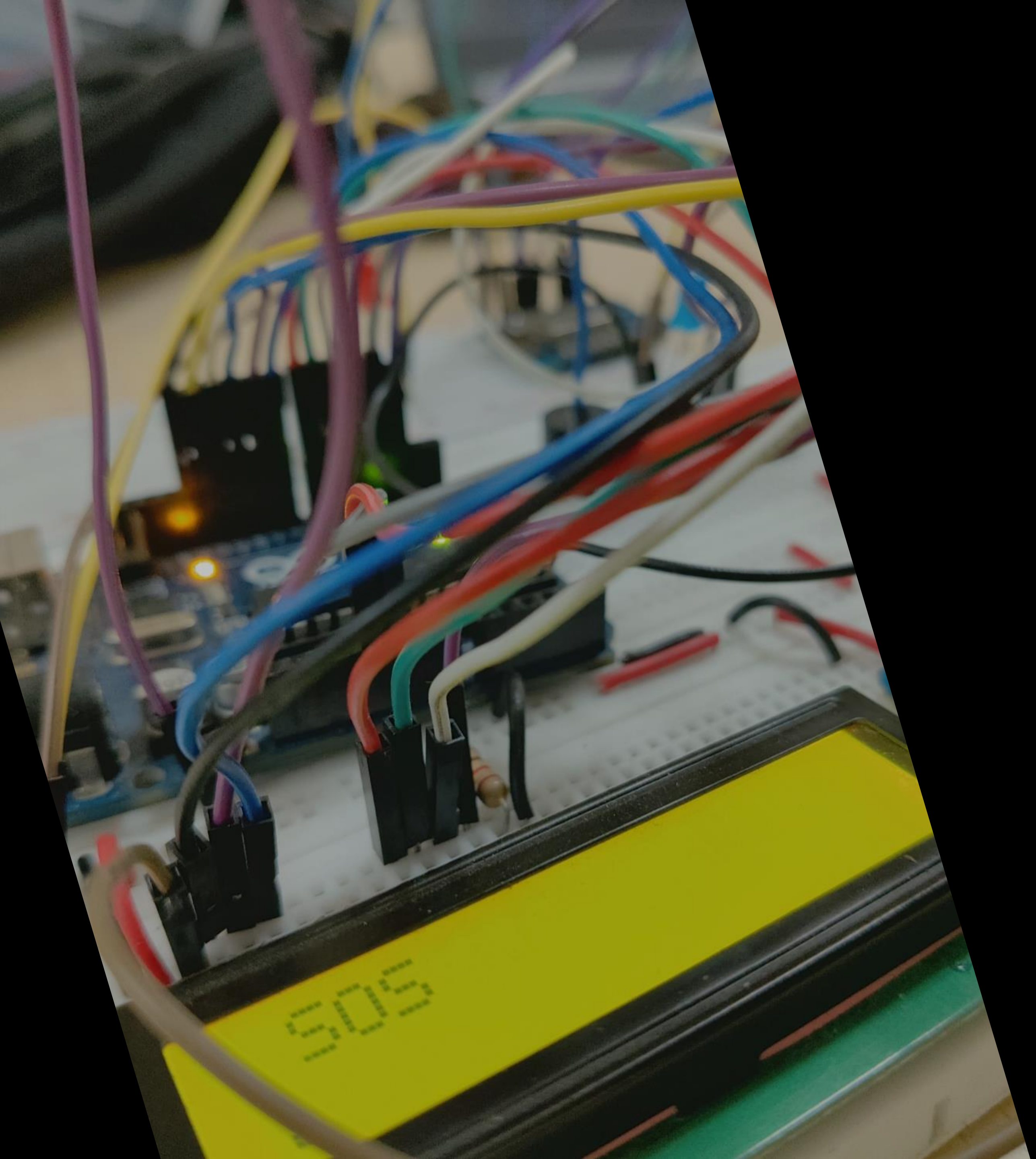
  1. The user enters the morse code in the form of dits and dahs through the pushbutton.
  2. The dits and dahs are decoded by the Arduino and the output is sent to both the LCD display and Node-RED server.
  3. The decoded output is visible on both the Node-RED dashboard and LCD display.

PROJECT DESIGN – CIRCUIT DIAGRAM



PROJECT DESIGN - DEMO





SOME KEY POINTS

This section delves into the efficiency of the code used and ease of operation for the user.

COMPONENTS USED

- Push Button
- Arduino(with USB cable)
- Breadboard
- 2x16 LCD display
- Buzzer
- LED
- Jumper Wires
- Some Resistors
- ESP 8266
- Dipswitch
- Potentiometer

KEY POINTS – ALGORITHMIC EFFICIENCY

- **Encoding**:
  - The encoding employs an O(n^2) algorithm where the Arduino reads from its serial monitor for input.
  - Each letter of this input is mapped to one of two functions – numFlash and charFlash based on the character’s ascii value.
  - These functions get the corresponding ascii code of the given character from the respective reference arrays and pass it onto the Flash function.
  - The flash function passes a high or low value on the corresponding DIO pin based on the received morse code from one of the above two functions and adjusts the delay between the pulses sent.

KEY POINTS – ALGORITHMIC EFFICIENCY

- **Decoding**:
  - The decoding also employs an $O(n^2)$ algorithm where the Arduino reads from the DIO pin connected to the pushButton for the input.
  - The function determines how long the pushbutton is pressed and based on this maps the given input to a dash or a dot.
  - These dashes and dots are accumulated in the string variable str which is then checked with the letters and numbers character arrays to find a match.
  - The index of the corresponding match is added with a certain offset and the character is accumulated in the string variable curr_val.
  - The character is printed onto the display and the curr_val is printed onto the serial monitor.

KEY POINTS – HARDWARE COMPLEXITY

- The current implementation of the project uses two power heavy components – LCD Display and ESP 8266.
- The implementation of serial communication between the ESP module and the Arduino adds to the complexity of the project requiring more wiring.
- The ESP module's connection with the Node-RED server through the MQTT protocol involves additional code for the ESP module, the setup of the mosquitto broker and setup of nodes in Node-RED – which adds to the complexity.
- The use of the LCD display to display the decoded output along with the current input also requires additional wiring adding to the complexity.

KEY POINTS – DEPLOYMENT AND EASE OF USE

- The setup in any environment is easy only requiring access to a power socket and a 12V AC power adapter for the Arduino.

- The user interface in Node-red is also simple with clearly labelled instructions.

- It also allows multiple users to send request to the Arduino for encoding through connection of their device to the Node-RED server.

- The availability of an LCD screen also allows the user to Locally verify the decoded output being sent to the server.

KEY POINTS – COST EFFICIENCY

Actuals

Item	Price	Qty	Cost
Arduino Uno	400	1	400
ESP8266	390	1	390
16X2 LCD display	250	1	250
Miscelleaneous costs	50	-	50
Buzzer	30	1	30
LED	10	3	30
Dipswitch 5 way	30	1	30
Potentiometer (5k) small	20	1	20
		TOTAL	**1200**

KEY POINTS – COST EFFICIENCY

- The project is not very cost efficient when it comes to the use of the ESP module for transmitting and receiving data from the server.
- However, it is more cost efficient in using the ESP8266 module instead of the ESP32 module as the ESP8266 module has lesser number of features compared to the 32 but is sufficient for the requirements for this project.

CONTRIBUTIONS

CONTRIBUTIONS BY TEAM MEMBERS

Sricharan

- Encoding function for the morse code
- Interfacing ESP8266 with Arduino
- Interfacing the LCD display with Arduino

Saikiran S

- Decoding function for the morse code
- Setting up Node-RED server and dashboard
- Setting up MQTT and mosquitto.

REMARKS AND LEARNINGS FROM THE PROJECT

REMARKS

- The use of devices like the raspberry pi as a server is much more feasible due to its low cost and enough capability.
- Switches can be implemented in Node-RED to implement mode control and output selection instead of dipswitch.
- Implementation of additional components is hard due to the limited power output of the Arduino.

LEARNINGS FROM THE PROJECT

- We learnt how to work together as a team to solve problems apart from learning other skills that helped solve our problem.
- We learnt how to interface Node-RED with the serial monitor of the Arduino to extract information from the serial monitor and vice versa.
- We learnt how to work with MQTT, Mosquitto, the ESP 8266 module and the interaction with each component with one another.
- We learnt how to use MQTT to establish communication with the Node-RED server.
- We learnt how to encode and decode morse code using Arduino and get inputs from and send outputs to different devices respectively.

THANK YOU!!