

**SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)**

**(Established under section 3 of the UGC Act 1956)**

**Re-accredited by NAAC with “A” Grade**

**Department of Electronics & Telecommunication Engineering**

**Academic Year 2025-26**

**Batch 2023-27, Semester - V**

| **Subject: Open Source Technologies (OST)** | **Credits: 3** |
| --- | --- |
| **Semester: V** | **Year: AY 25-26** |

**Title of the Project:** Morse code decoder

**Group Information:**

| **Sr no.** | **PRN** | **Name** |
| --- | --- | --- |
| **1** | 23070123019 | Amey Kumar |
| **2** | 23070123025 | Anushka Shinde |
| **3** | 23070123070 | Jiya Palod |
| **4** | 24070123509 | Maitreyee Gohad |

**1. Working of Project:**

* **Project Title**: Morse Code Decoder.
* **Description**:

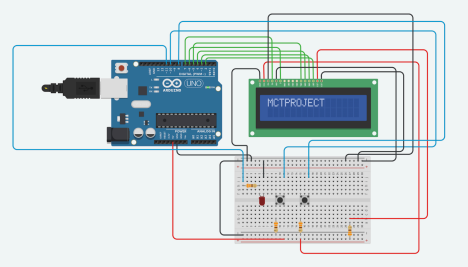
This is an Arduino-based project where the user inputs Morse code (dots “·” and dashes “–”) via two push buttons. The system decodes the Morse sequence into alphanumeric characters and displays them on a 16×2 LCD screen.

* **Tools and Technologies used:**

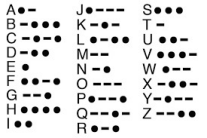
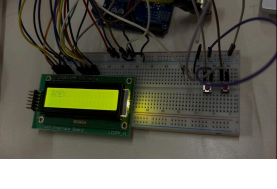
1. **Hardware :** Arduino Uno, push buttons, LEDs, resistors, jumper wires, LCD, breadboard.
2. **Software :** Arduino IDE(C++, liquidCrystal library.
3. **Documentation and repo :** Github

* **Screenshots of Execution**:

**Fig.(a) TinkerCAD Simulation**

****

**Fig.(b)Breadboard Fig(c)Morse Code reference**

****

* **How to Run the Project**:

1. Connect the circuit using the hardware components on breadboard as per the schematic diagram.
2. Open MorseDecoder.ino in **Arduino IDE** (file in repo) .

Github - <https://github.com/jiyapalod/Morse-Code-Decoder>

1. Install LiquidCrystal library.
2. Select **Arduino Uno** board and correct **COM port**.
3. Upload the code.
4. Use dots and dashes using push buttons to enter Morse code.
5. The decode output is displayed on the LCD.

* **Expected vs Actual Output**:

1. Expected Output: When you input a valid Morse code for “A” (·–), the LCD should display “A”.
2. Actual Output: The project works correctly most of the time and decodes Morse inputs accurately. However, it occasionally struggles to distinguish between **letters and words**, especially when the pause between inputs is inconsistent.

**2. Innovation / Novelty / Contribution:**

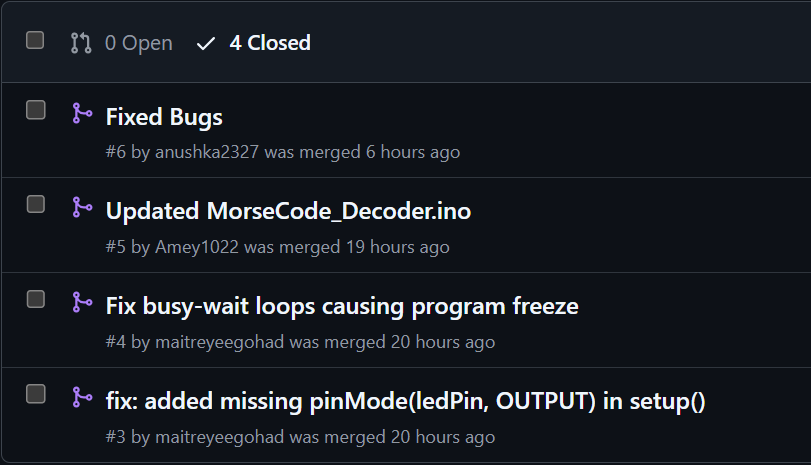
* **Unique Feature or Enhancement**:

→ This project converted combinations of dots and dashes into readable and meaningful letters and words.

1. More time gap between inputs created a differentiation between letters and words so that the combinations were made into meaningful sentences.
2. Two-button integration to make the differentiation between dots and dashes easier.
3. Used an Arduino microcontroller for all the processing, making it inexpensive and affordable.



**PULL REQUESTS:**



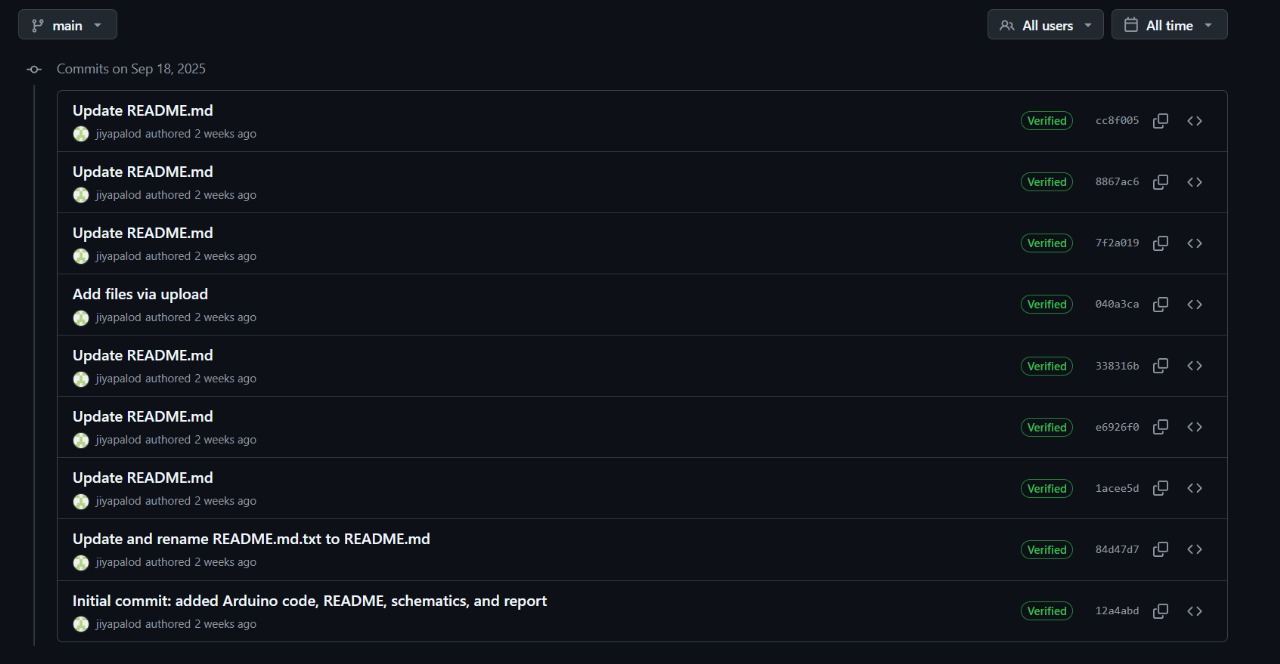
| **Student Name** | **Role/Contribution** | **GitHub Profile** | **Key Commits / Pull Requests** |
| --- | --- | --- | --- |
| Jiya Palod | Backend Design, ReadMe | <https://github.com/jiyapalod> | <https://github.com/jiyapalod/Morse-Code-Decoder/commits/main/> |
| Maitreyee Gohad | Automation | <https://github.com/maitreyeegohad> | <https://github.com/jiyapalod/Morse-Code-Decoder/pull/4> |
| Anushka Shinde | Testing | <https://github.com/anushka2327> | <https://github.com/jiyapalod/Morse-Code-Decoder/pull/6> |
| Amey Kumar | Bug Fixes | <https://github.com/Amey1022> | <https://github.com/jiyapalod/Morse-Code-Decoder/pull/5> |

* **Challenges Faced and Solved**:

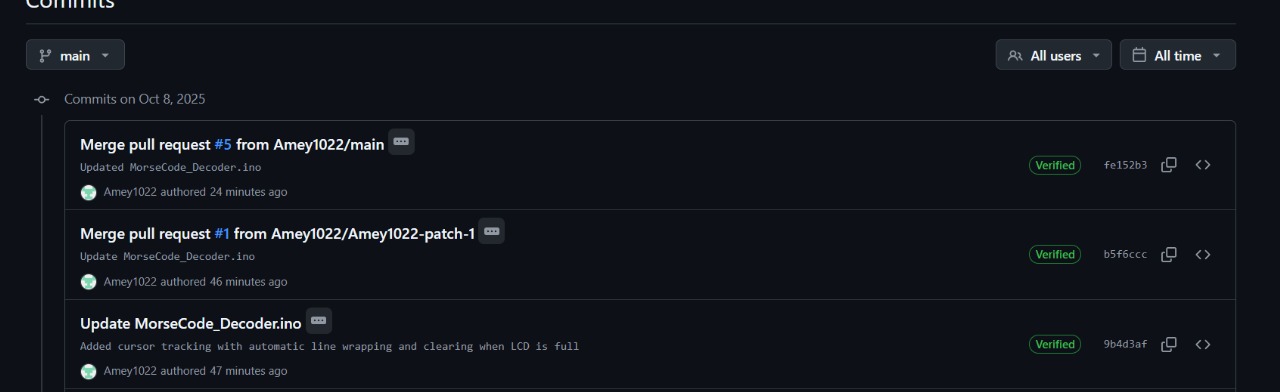
1. LCD Integration: The major challenge we faced was the display not working with the arduino when connected. Fixed the issue by changing the LCD panel, making the connection more secure and setting the regulator on the desired level.
2. Button Recognition: Arduino did not recognize button long press and short press. Fixed the issue by using two buttons- 1 for dot ,and 2 for dash. Also increased the detection duration.

**3. Timely Submission:**

* **Evidence of Progress**: Screenshots of commit history or logs.







**4. Pushing the Project to GitHub:**

* **GitHub Repository Link**: [Morse-Code Decoder](https://github.com/jiyapalod/Morse-Code-Decoder)
* **Issues:**
* **README File**: It contains all the necessary information related to the project repository which explains the objective and working of the project. It helps users understand the reason and operation of the given project.
* **Git commands used:**

1. git init: Initialize a new Git repository
2. git add: Add all project files to staging area
3. git commit -m "initial commit" : Commit the files with a message
4. git branch -M main: Rename the default branch to 'main'
5. git remote add origin [https://github.com/jiyapalod/morse-code-decoder.git :-](https://github.com/jiyapalod/morse-code-decoder.git:-) Connect the local repo to GitHub
6. git push -u origin main: Push the files to GitHub

**5. Conclusion:**

We successfully built a morse code decoder and pushed the project to the github repository. This repository contains all the documents and codes related to the project. We also modified the code to make it more efficient and add new enhancements while solving issues raised by other users.