

Project Report

Kiran Gani

2581349, s8kigani@stud.uni-saarland.de

Mahima Mahabaleshwar Acharya

7003016, maac00001@stud.uni-saarland.de

1 BASIC PIPELINE : MORSE TO HEX CODE CONVERSION SYSTEM

We implemented an interactive system which converts given Morse input to corresponding Hexadecimal output. We have used a single push button to provide Morse input, where small press is considered as dot and long press is considered as dash. Then we process the provided input using WeMos D1 mini microcontroller and we decode the input into a hexadecimal code using a Servo motor, which rotates to corresponding hexadecimal as shown here <https://drive.google.com/file/d/1U2toRIrgiAYNLJHqZlpbeS-KzVjvWZ3s/view?usp=sharing>.

Mandatory alternative encoding method:

Here we have implemented a DIY capacitive touch sensor¹ using Aluminum foil and resistor as input device instead of push button as shown in the end part of this link <https://drive.google.com/file/d/1U2toRIrgiAYNLJHqZlpbeS-KzVjvWZ3s/view?usp=sharing>. This method was much easier to give input compared to push button.

2. ALTERNATIVE HARDWARE DECODING

Here we used 7 segment display to show the hexadecimal output. We also used two separate DIY capacitive touch sensor using Aluminum foil as input for dot and dash and it was comfortable to provide input. As we couldn't buy 7 segment display as shown in the previous concept report, we used one of the digit from 4-digit 7 segment display. Refer the provided link to see the implementation: <https://drive.google.com/file/d/1UJVS5H5DHXCytutj18hTlSvOcd6GZkQv0/view?usp=sharing>.

3. ARDUINO-UNITY3D COMMUNICATIONS

We have established communication between Unity3D and Arduino. Arduino interacts with Unity3D whenever the Morse code is given using a push button and Unity3D displays corresponding ASCII values. Similarly, we control an LED using a push button game object. Video link: <https://drive.google.com/file/d/1UM4CFyxm9DmYFvpsZ9aU6-kr9E46uV6k/view?usp=sharing>.

4. 3D VISUALIZATION IN UNITY3D SCENE

Visualization of the basic pipeline includes push button, a board with hexadecimal numbers and a pointer to read the output. Pressing a valid Morse code using a push button makes the pointer to indicate corresponding hexadecimal value.

Link : <https://drive.google.com/file/d/1UVA6q35bA40WmYDEY9QN1kQX2qn3ieV5/view?usp=sharing>.

5. REMOTE COMMUNICATION

The communication between two microcontrollers is established using NRF24L01² digital transceivers. The bidirectional remote communication is shown in this link : <https://drive.google.com/file/d/1UIU9ZG7-4Gg4ukpCYTcwMVMdLK9UDMOj/view?usp=sharing>.

Components List: Push Button, LEDs, WeMos D1 mini, Jump wires, 220KΩ and 100KΩ resistors.

Additional : 1. 9g Servo motor, Aluminum foil, 4.7KΩ resistor. 2. Two Aluminum foils, Arduino Uno, 4-digit 7 segment display, 22KΩ resistors. 5. NRF24L01 digital transceivers.

References:

1. <https://www.arduino.cc/reference/en/libraries/capacitivesensor/>
2. <https://github.com/nRF24/RF24>