

Project Title: Morse Code Translator

Objective:

- i) To design and build a device which converts morse code into letters and numbers.
- ii) To learn basics of Arduino UNO.

Theory: Text characters can be encoded using Morse code, a technique used in communications that uses defined sequences of two distinct signal lengths, known as dots and dashes. For every alphabet and number, a specific set of dots and dashes must be used. When discussing emergency scenarios, Morse code is a useful tool for communication. When telegraph operators first started using electrical pulses to transmit messages, they converted the signals into text using Morse code.

Although technological advancements have made Morse code practically obsolete by improving communication methods, several sectors still utilize it in their workforce. Aviation professionals, like pilots and navigators, can communicate identification letters for station names using Morse code. Furthermore, amateur radio enthusiasts occasionally use the code to start data transfers and convey messages between broadcasts.

We designed and built a device which is able to convert morse code into readable text. The used components are:

1. 1x Arduino UNO R3
2. 1x 16x2 LCD display
3. 6x Push buttons (2 Pin & 4 Pin)
4. Breadboard
5. Connecting wires
6. Screw driver
7. Tapes
8. Scissor

Operation: A code is needed to be programmed in Arduino IDE and then uploaded in the Arduino UNO. Display and buttons are connected with Arduino UNO which would operate the device.

Circuit Diagram:

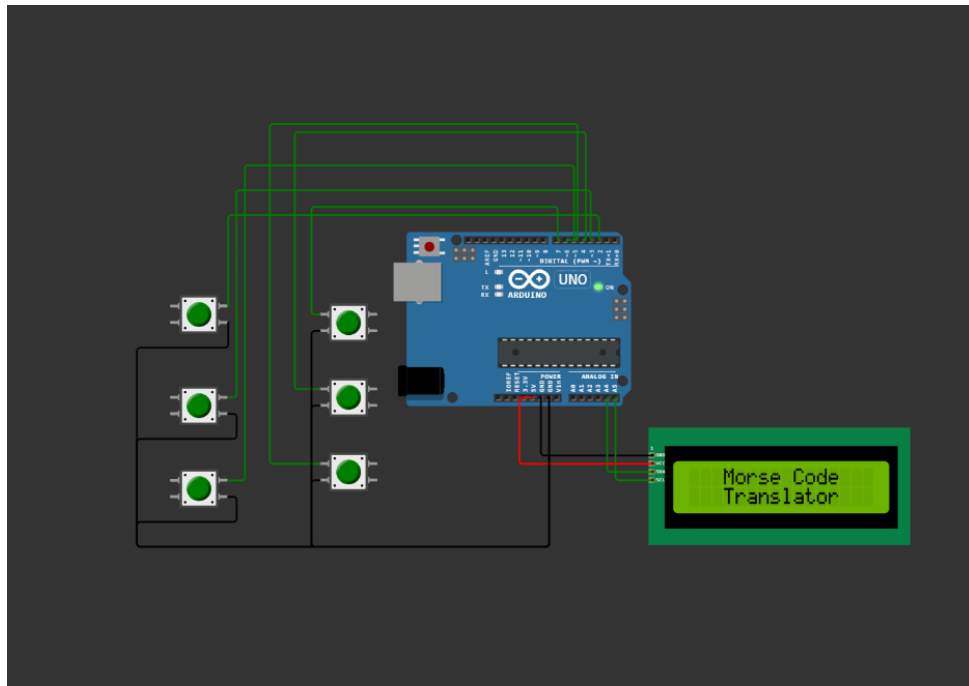


Fig 1: Circuit diagram (In simulation)

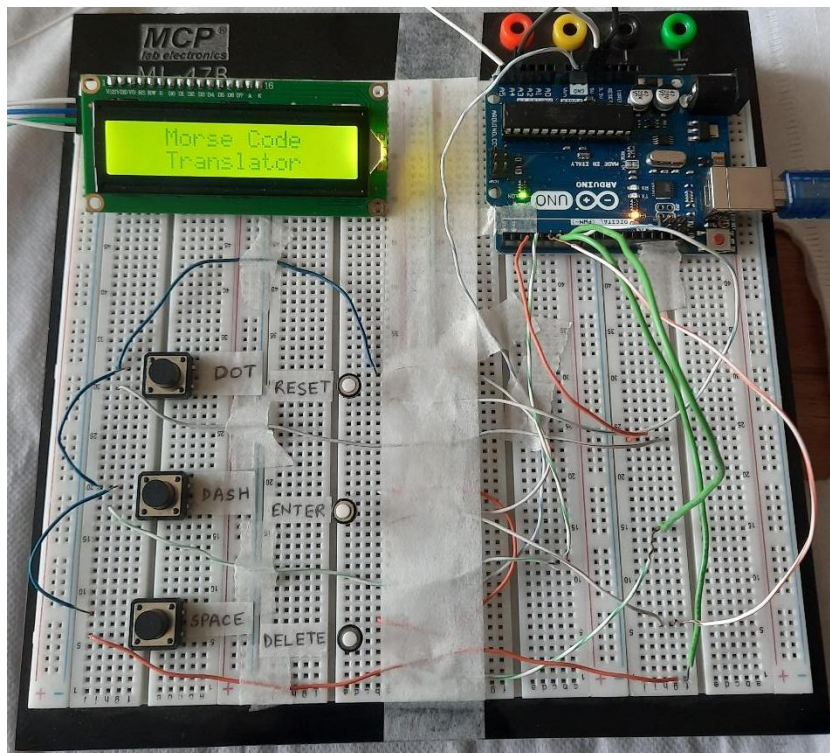


Fig 2: Circuit diagram (In Practical)

Implementation:

1. Connect ground of the display to the ground of Arduino.
2. Connect VCC of display to +5 V of Arduino.
3. Connect SDA of display to A4 of Arduino.
4. Connect SCL of display to A5 of Arduino.
5. Three 4 Pin buttons for DOT, DASH and SPACE connected to pin- 2, pin-3 and pin- 5 of Arduino respectively.
6. Three 2 Pin buttons for ENTER, DELETE and RESET connected to pin- 4, pin- 6 and pin- 7 of Arduino respectively.
7. Connect Arduino with a power source. Power source could be either a battery or a laptop.
8. Install Arduino IDE to write the required code.
9. Install the library “LiquidCrystal_I2C.h” in Arduino IDE since the display used is a 4 pin (12 pin converter) display. If 16 pin display is used, it is not required to install it.
10. Arduino may not find the address of the displaying device. When this issue occurs, the display will look like this:

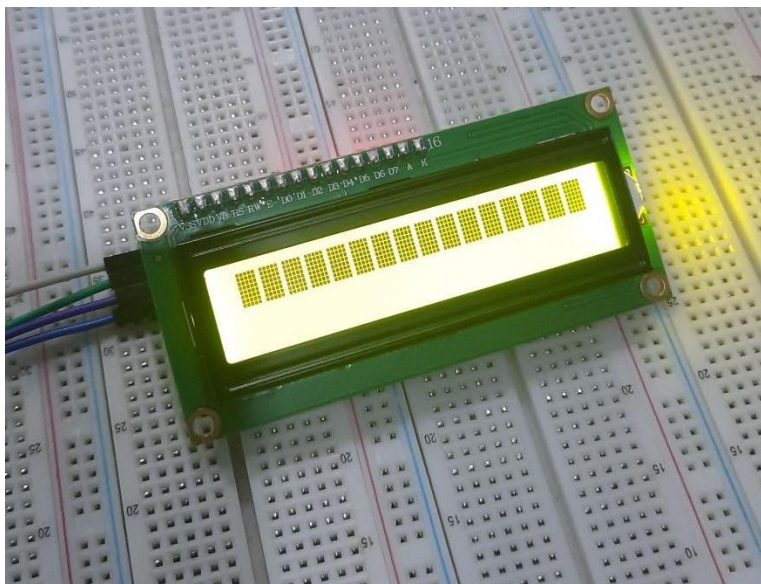


Fig 3: Address of display not found

To solve this issue, use the code given in the link below:

Code link-----[I2C Display Address Finder](#)

The code will provide you the address of the displaying device through which you can have the access to operate the device. The provided address should be copied as it will be needed later.

11. To get better and accurate view of the display, make sure to change the contrast of it. It can be done by rotating the potentiometer at the backside of the display using a screw driver.

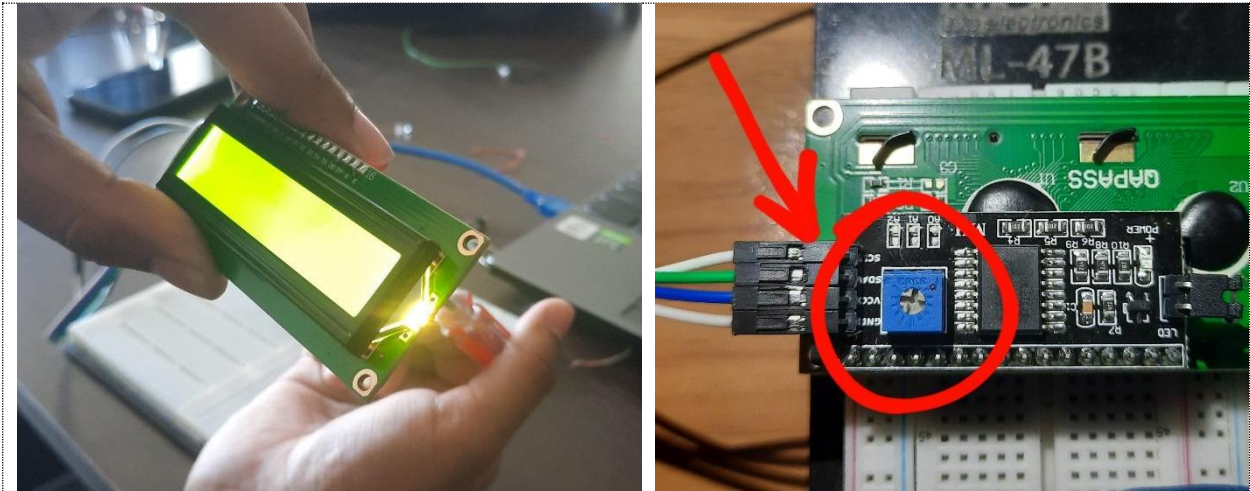


Fig 4: Adjusting the contrast

12. Upload the following code to the Arduino.

Code link----- [Morse Code](#)

13. Follow the given combinations of morse code (International Morse Code) to get the desired result.

A ● -	J ● - - -	S ● ● ●	1 ● - - - -
B - ● ● ●	K - ● -	T -	2 ● ● - - -
C - ● - - ●	L ● - ● ●	U ● ● -	3 ● ● ● -
D - ● ●	M - -	V ● ● ● -	4 ● ● ● ● -
E ●	N - ●	W ● - -	5 ● ● ● ●
F ● ● - ●	O - - -	X - ● ● -	6 - ● ● ●
G - - ●	P ● - - ●	Y - ● - -	7 - - ● ●
H ● ● ● ●	Q - - ● -	Z - - ● ●	8 - - - ● ●
I ● ●	R ● - ●		9 - - - - ●
			0 - - - - -

Fig 5: International Morse Code

Note: The characters are only displayed on the top half of the display. If the first row is filled, then please RESET.

Budget:

Serial	Components	Quantity	Price Per Quantity (BDT)	Total Price (BDT)
01	Arduino UNO R3	1	1,100	1,100
02	16X2 Serial LCD	1	340	340
03	Push Button Switch (2 Pin)	5	5	25
04	Push Button Switch (4 Pin)	5	7	35
				Total= 1500

User Manual:

1. There are six buttons, named
 - a) DOT
 - b) DASH
 - c) SPACE
 - d) ENTER
 - e) DELETE
 - f) RESET
2. Press RESET to start/ clear the display.
3. Pressing DOT inputs a dot and pressing DASH inputs a dash. Certain combinations of dots and dashes will show the corresponding letters or numbers.
4. ENTER must be pressed after each combination otherwise the characters won't show.
5. When SPACE is pressed, it works like a normal space button and produces a gap between two letters.
6. DELETE can be used to delete the last letter written. ENTER must be pressed after DELETE to make it work. DELETE can be pressed as many times as required, and after that if ENTER is pressed, those characters will be deleted together.
7. The buttons are slightly sensitive. So, make sure you don't hold the button for long, otherwise it will take that as multiple inputs.
8. Enjoy our product.

Discussion:

First, we designed a circuit diagram and checked if it worked using a simulation software. As the simulation was successful, we ordered the necessary components online. After we received those, we started to work on the project. We did every step as mentioned in the "Implementation" section. We faced a number of problems. First, we had a problem with our display. The Arduino was unable to find the address of the display, so we could not access it. We used a code which provided us the address, and the problem was solved. We had to adjust the contrast to our preference. The contrast of display can be controlled by controlling a

potentiometer which is at the back side of the display. Then we faced problems with the wires. So, all the wires were checked and then connections were given. We had lots of difficulties writing down the correct code. We took help from online and modified the code to our requirements so that it could work. Finally, we were successful with the code. The buttons are a bit sensitive. We tried using resistors to solve this, but it didn't help. Now, we just have to press the buttons for a very short amount of time to make the translator work perfectly fine.

Reference: Inspired By,

1. <https://www.youtube.com/watch?v=CJriiDdaXP0>
2. https://github.com/tesar-tech/arduino_and_morse_code/blob/master/script.ino
3. https://github.com/SABIT-HASAN-117/Morse-Code-Translator-with-Arduino/blob/main/I2C_Display_Address_Finder.ino