

Arduino Morse Code

-Arpan Thomas & Blesson Thomas

What is Morse Code

Morse code is a method used in telecommunication to encode text characters as standardized sequences of two different signal durations, called dots and dashes or dits and dahs. Morse code is named after Samuel Morse, an inventor of the telegraph.

International Morse Code, also known as Continental Morse Code,[4] encodes the 26 English letters A through Z, some non-English letters, the Arabic numerals and a small set of punctuation and procedural signals (prosigns). There is no distinction between upper and lower case letters. Each Morse code symbol is formed by a sequence of dots and dashes. The dot duration is the basic unit of time measurement in Morse code transmission. The duration of a dash is three times the duration of a dot. Each dot or dash within a character is followed by period of signal absence, called a space, equal to the dot duration. The letters of a word are separated by a space of duration equal to three dots, and the words are separated by a space equal to seven dots.[1] To increase the efficiency of encoding, Morse code was designed so that the length of each symbol is approximately inverse to the frequency of occurrence of the character that it represents in text of the English language. Thus the most common letter in English, the letter "E", has the shortest code: a single dot. Because the Morse code elements are specified by proportion rather than specific time durations, the code is usually transmitted at the highest rate that the receiver is capable of decoding. The Morse code transmission rate (speed) is specified in groups per minute, commonly referred to as words per minute.

Morse code is usually transmitted by on-off keying of an information-carrying medium such as electric current, radio waves, visible light, or sound waves.[6][7] The current or wave is present during the time period of the dot or dash and absent during the time between dots and dashes.

Morse code can be memorized, and Morse code signalling in a form perceptible to the human senses, such as sound waves or visible light, can be directly interpreted by persons trained in the skill.

Because many non-English natural languages use other than the 26 Roman letters, Morse alphabets have been developed for those languages.

In an emergency, Morse code can be generated by improvised methods such as turning a light on and off, tapping on an object or sounding a horn or whistle, making it one of the simplest and most versatile methods of telecommunication. The most common distress signal is SOS – three dots, three dashes, and three dots – internationally recognized by treaty.

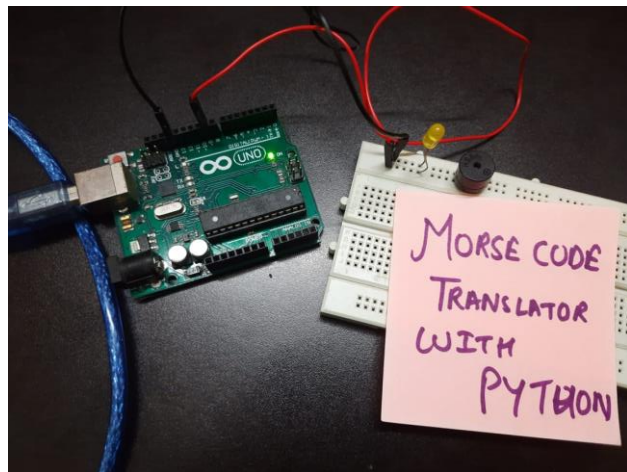
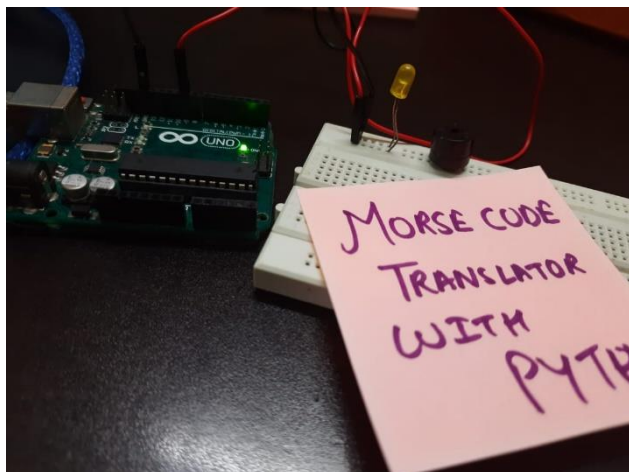
Morse code for various alphabets

International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

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

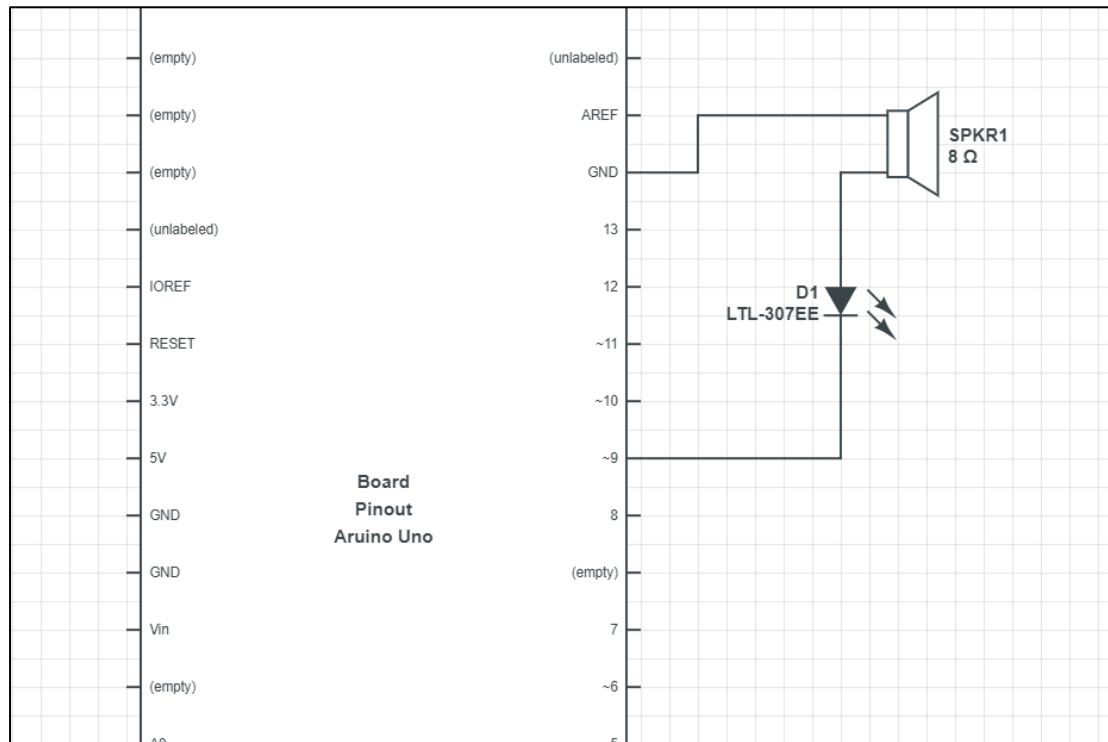
Pictures of project



Python Code

```
from pyfirmata import Arduino, util
board = Arduino('COM5') #COM5 is the port where arduino board is connected
import time
msg = input("Enter your Secret Message: ")
morse_code = {'A': '.-', 'B': '-...', 'C': '-.-.', 'D': '-..', 'E': '.',
              'F': '..-', 'G': '--.', 'H': '....', 'I': '...', 'J': '.---',
              'K': '-.-', 'L': '-.-.', 'M': '--',
              'N': '-.', 'O': '---', 'P': '.---', 'Q': '--.-', 'R': '.-.',
              'S': '...', 'T': '-', 'U': '..-', 'V': '...-', 'W': '-.-',
              'X': '-.-.', 'Y': '-.-.', 'Z': '--..',
              0: '-----', 1: '.-----', 2: '..-----', 3: '...-----',
              4: '....-----', 5: '.....', 6: '-.....', 7: '--.....',
              8: '---.....', 9: '----..'} # python dictionary for the Morse Code Equivalent of
Alphabets and numbers
rate=0.5 #rate of displaying message
for i in msg:
    n = i.capitalize() # the selecting each word from message
    c = morse_code[n] # the morse code output that is "-" and "."
    print(n, end=" ")
    print(c)
    for j in c:
        if j == '.':
            board.digital[9].write(1)
            time.sleep(rate)
            board.digital[9].write(0)
            time.sleep(rate)
        elif j == '-':
            board.digital[9].write(1)
            time.sleep(3*rate)
            board.digital[9].write(0)
            time.sleep(rate)
        elif j==' ':
            time.sleep(7*rate) # wait for 7 unit after any word
            time.sleep(3*rate) # wait of 3 unit after each letter
# code written by Blesson Thomas
# hardware module and logic coding by Arpan Thomas
```

Pin diagram



It is recommended to use place the led and buzzer in parallel so that it looks good. Both can directly be placed in pin 9 by using a breadboard and jumper cable.