Lauren Courtney

February, 2020

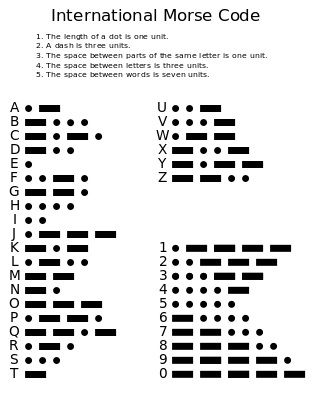
EN 605.715.81 Embedded Systems

**Project 1 – Morse Code LED**

**Requirements**

1. The Arduino shall display a user-typed string as Morse Code using an LED

2. The LED shall blink following the International Morse Code standards as listed below:



3. A user shall be prompted to enter a string, and once they press enter the Arduino will display the string, then the user will be prompted to enter another string, and this will continue in a loop.

i. The maximum length of each entry shall be 100 characters (a constraint I am adding)

4. The program shall exit only upon user input of a sentinel character

i. The sentinel character shall be ‘!’ (I found that Ctrl-Z did not work if a user was trying to use the Serial Monitor that is provided via Arduino Sketch, so I followed the example of [Source 2](https://automaticaddison.com/how-to-display-a-string-as-morse-code-on-an-led-using-arduino/) and used the exclamation point)

5. The program shall implement a Round Robin Design schedule

**Design**

**Hardware**

1 Arduino – Mega 2650 (program should run on any Arduino)

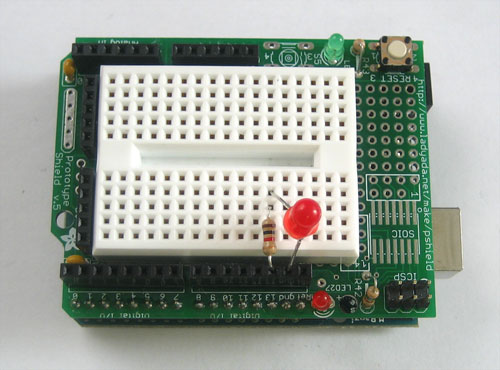
1 LED

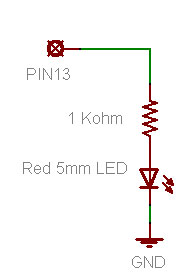
1 KΩ Resistor

1 Breadboard

1 Laptop

1 USB cable

Design schematics and images from [Source 3](http://www.ladyada.net/learn/arduino/lesson3.html) 



Connect resistor to pin 13 (the default LED pin), the use the breadboard to connect the positive end of the LED to the resistor. Connect the negative end of the LED to ground. This will allow the LED to light up any time pin 13 is HIGH. Use the serial port to connect the Arduino to a laptop to allow for user input.

**Software**

The code will be written in C using Arduino Sketch for the Serial Monitor and uploading.

I. Setup:

- Open the Serial port.

- Initialize arrays with hardcoded Morse Code letters and numbers as “..-- “, etc.

- Initialize base time unit to 100 milliseconds

- Create a buffer for string input and initialize a maximum size of 100 chars

II. Loop:

1. Read user input:

-Use the Serial library to take in user input as explained by [Source 1](https://forum.arduino.cc/index.php?PHPSESSID=ujl4fahf22fo321mdu3ag27363&topic=288234.0).

- Display a prompt asking users to enter a string

- Collects input characters in a buffer until the user presses enter.

- If more than 100 characters are entered, the final character will be continuously overwritten

-Alert the user will be alerted each time this happens.

- If the user enters the ‘!’ character:

- Set a “Done” flag to true, then terminate the program by closing the Serial connection.

- Otherwise, once the user hits enter:

- Add a string terminating character to the end of the buffer

- Move to Step 2.

2. Get Morse Character

- Loop through input string buffer until string terminating character

- For each character, determine if it is a letter or number

- For letters, calculate their index into the alphabet by subtracting the char from ‘a’ or ‘A’.

- Use the index to get the Morse Code value of that character from the letter and number arrays 3. Transmit Morse

- Loop through the Morse value of each character

- Turn pin 13 to HIGH

- Delaying one or three time units for a dot or dash

- Turn pin 13 to LOW.

- After each letter delay for one time unit

- After each word it delay an additional two time units (for a 3 unit delay)

- If the character is a space, then delay an extra 4 time units (for a 7 unit delay).

4. Repeat Loop

This processing guarantees that the Arduino will continue to prompt the user and display the string in Morse Code on a loop until the user enters ‘!’.

**Demo**

Demo can be found on YouTube at <https://youtu.be/c25jXkdZB2M>

**References**

Source 1: <https://forum.arduino.cc/index.php?PHPSESSID=ujl4fahf22fo321mdu3ag27363&topic=288234.0>

Source 2: <https://automaticaddison.com/how-to-display-a-string-as-morse-code-on-an-led-using-arduino/>

Source 3: <http://www.ladyada.net/learn/arduino/lesson3.html>