Project 1 Morse Code LED System Development
Douglas Eucken
EN.605.715.81.SP21
Software Development for Real-Time Embedded Systems
January 28, 2021

Problem Statement:

Develop a C or C++ (or whatever other language you prefer) application which executes on an Arduino and displays a user typed string, such as "Hello World", as Morse Code on an LED (or several LEDs), or an LCD. Design using a Round Robbin Design where in a loop it waits for a string, displays it in Morse Code, and only exits the loop if a sentinel is entered, such as ctrl-z

Requirements:

1. The design shall contain an Arduino board with software-controlled LED.
2. The design shall display a text string in Morse code via the software-controlled LED.
3. The design shall implement a Round Robin C or C++ software design where in a loop it waits for a string, displays it in Morse Code, and only exits the loop if a sentinel is entered.
4. (Derived) The design shall display a visual indication at the beginning of each character being transmitted via Morse code.

Design:

The hardware design for this project consists only of my Arduino UNO development board and a USB connected PC. I chose to use the onboard user-controllable LED on the board itself (tied to I/O pin 13) for the Morse code display.
The software design is basic C program which runs a Round Robbin loop (as dictated by the requirements). I used a couple examples I found online and meshed them into one design I liked best (those designs are linked below in the source section). In the beginning of the file there are some variable declarations and setup in order to initialize the LED and open a serial connection to the PC. A welcome message is sent via serial so that the user understands how the program works. There are also three small subroutines that are called by the main loop. Two of them (dot and dash) are used to control the LED and communicate either a single dot or dash. The other (morse) takes each character that is entered by the user and translates it into the correct Morse definition of dots/dashes and then calls the dot and dash subroutines. That definition is linked below in sources as well. The main program loop checks for new serial data and then reads it in one character at a time. If the character is the sentinel value (in this case, it's the "#" character) then the program will enter an infinite loop state. Otherwise, the character is printed on the serial screen and then sent to the morse subroutine for display on the LED.

Implementation/Code:

The completed C code is attached and available for review here:
https://github.com/deucken1/Real_Time_Embedded_Project1/blob/main/morse_code_eucken.ino

Demo Video:

https://livejohnshopkins-my.sharepoint.com/:v:/g/personal/deucken1_jh_edu/EWUSwqihySxBh7GfHVvBpvkB5Xyj5bVtYexeftY7jptByg?e=5QXTXp

Sources:

https://gist.github.com/baojie/4460468
https://automaticaddison.com/how-to-display-a-string-as-morse-code-on-an-led-using-arduino/
http://arduino.cc/forum/index.php/topic,43903.0.html#10
https://en.wikipedia.org/wiki/Morse_code