SE101—Lab 1

Rollen D'Souza, Patrick Lam, Mahesh Tripunitara

Fall 2016
Last Updated September 22, 2016

Introduction

In the first lab you are required to implement a program that takes as input a series of characters and emits a Morse Code representation of the sequence using the built-in light emitting diode (LED). The microcontroller board houses an LED capable of illuminating in red, green and blue. Take a moment to look at Figure 1 and familiarize yourself with the board layout.

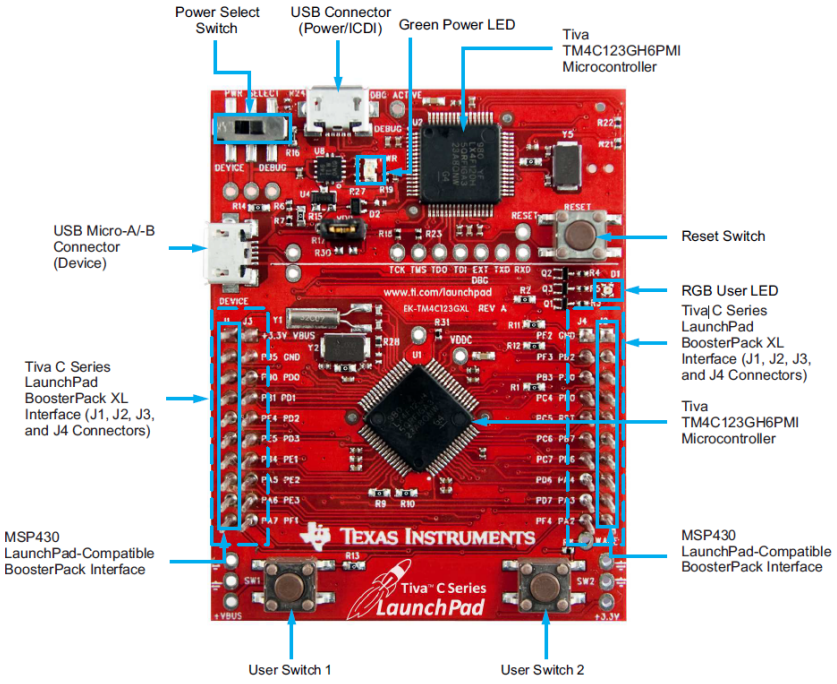

Figure 1: Tiva C-Series TM4C123GXL Board Layout [1]

Morse Code

Morse Code is a method of transmitting characters as a series of alternating tones—traditionally clicks (as in Figure 2), but also lights—that can be understood without the need for special equipment [2]. It is normally represented using a series of dots (short signal) and dashes (long signal). Details on Morse Code, such as the specific mapping between

characters and signal-sequences, can be found at the link provided in the reference. We recommend reading the entire reference. You will be marked against the International Morse Code chart.

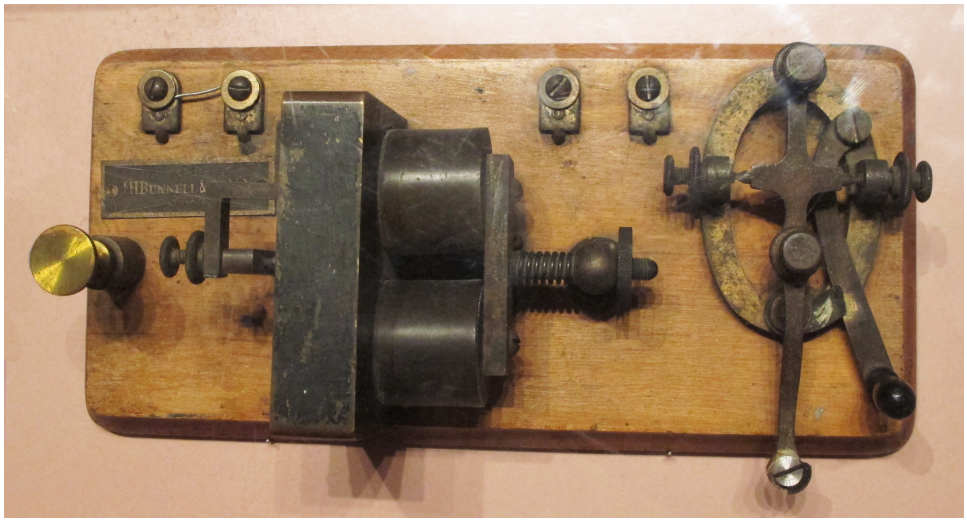

Figure 2: Wright Brothers telegraph key; By Sanjay Acharya—Own work, CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=15733842`

Requirements

The program must accept a line of text from the serial input and transmit the Morse encoding using the *green* LED. A line of text is delimited using the linefeed/newline (`\n`) character. The program must transmit the Morse encoding of the message exactly once and then must continue to accept further messages from the user. You must implement the morse encoding for all 26 english letters, in both upper case and lower case. We will test atleast five characters at random as well as ill-formed input.

A dot-signal lasts one second. A dash-signal lasts three dots. Between dots/dashes there is a *one* dot length wait. Between letters there is a *three* dot length wait. Between words there is a *seven* dot length wait. Punctuation and case should be ignored.

The final submission (commit) must be made by 1130 on Thursday, September 29, 2016. You will demo your solution during that lab period.

Marking Scheme

	Criteria	**Mark Contribution**
Correctness	Correct Morse Code	10
	Handles Ill-Formed Input	5
Style	Naming	1
	Whitespace Usage	1
	Easy to Modify	1
	Use of Control Flow and Types	1
	Use of Source Control	1
	Total	20

Submission

You must submit your assignment using the ECE SVN (subversion) repository. You are graded on your use of Subversion. We expect to see at least two non-trivial commits. Trivial commits are those that consist of at most changes to whitespace.

There will be a separate document on how to use ECE SVN. Please see Piazza for more information.

Please commit your solutions to:

`https://ecesvn.uwaterloo.ca/courses/se101/2016/students/YOUR\_WATID/lab1`

where you replace "YOUR_WATID" with your WatID (e.g. p23lam). Email Patrick Lam if you can't commit to that address.

Tips

- Use Energia's `Serial` library.
- Energia's Serial Monitor supports different line endings. Be sure to select newline only!
- Consider how this lab can be broken into components.
- What we mean by "easy to modify": Can we easily remove some components of your lab for use in a different program?
- GCC supports encoding numbers in binary representation like so: `0b101010111`.
- Make sure to follow Piazza.

References

[1] Texas Instruments. (2014). *Tiva^{TM}C Series TM4C123G Launchpad Evaluation Board User's Guide* [PDF]. Available: `http://www.ti.com/lit/pdf/spmu296`

[2] Wikipedia. (2016). *Morse Code - Wikipedia* [Online]. Available: `https://en.wikipedia.org/wiki/Morse_code`