COM1031 Computer Logic Assessment

Implementation of a Morse Decoder for letters

Deadline	22 November 2024, 4pm
Demo Date	Lab 10 in IFH CS Lab Lab 11 as the back-up date
Feedback & grades	2nd January 2025 via Surreylearn
Module Weight	40%
Academic Misconduct	Coursework will be routinely checked for academic misconduct. Your submission must be your own work. Please read the Student Handbook to ensure that you know what this means. Do not give your code to anyone else, either before or after the coursework deadline.
Versions	1st Version: 19 October 2024
Important:	ALWAYS BACK UP YOUR FILES

1. Morse Code

Morse code is an encoding of letters using short (dots) and long (dashes) signals, see for example http://en.wikipedia.org/wiki/Morse_code. Signals can be given either with short and long acoustic beeps, visual blinks, electric pulses or any other form of transmission as long as short and long signals can be distinguished.

2. Learning Outcomes

a) You will apply the principles of analysis and design of simple circuit systems.
b) You will use number systems to perform simple binary arithmetic.
c) You will understand the basic structures of a computer system and comprehend their function.

3. Details

BELOW IS A LIST OF REQUIREMENTS YOUR IMPLEMENTATION NEEDS TO FULFIL:

1. **Start of the program.** When the program starts, the 7-segment display turns on all LEDs (i.e., showing a '8) for testing then turns them all off. The terminal displays a welcome message.

2. **External Circuitry:** You should connect the components (7-segment display and button) using the lab template (see Figure 1). Make sure there is no short circuit before the program is run.

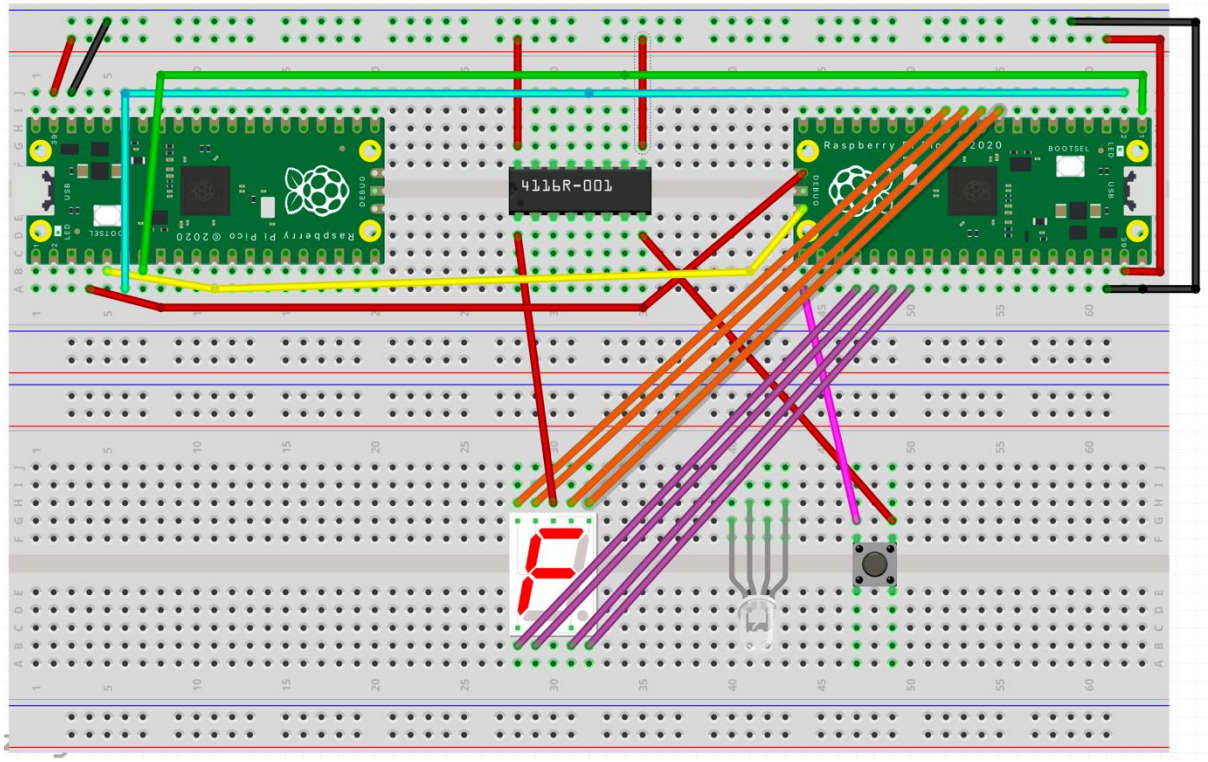

Figure 1. Circuit connection for the 7-segment display and the button

3. **Timing of Duration of Button Presses:** The duration of a button press should be recognised as a dot when it is shorter than 250ms and as a dash otherwise. This timing corresponds to a typical duration of a dot of 100ms and of a dash of 300ms. This is considerably slower than for professional (or even amateur) radio operators (at about 20-50ms for a dot), but this will – with some trial and error – also allow the inexperienced (like you or me) to enter morse codes.

4. **Timing of Gaps between Morse Signals:** We will assume that an inter-signal gap is about 100ms, and an inter-letter gap about 700ms. Taking the average as the boundary to discriminate between short and long gaps, everything shorter than 400ms should be counted as an inter-signal gap and everything longer as an inter-letter gap.

5. **End of a letter** If, after a button release, the button is not pressed again within 400ms, the program should display the recognised letter. This is because 400ms is our boundary for decision whether we have an inter-signal gap where other signals are to follow for this letter, or a letter-gap with no more morse signals to follow.

6. **Letters on the seven-segment display.** A seven-segment display only has limited means of representing letters A–Z. You are suggested to use the following alphabet

displays for the coursework:

Component Data Sheets
Seven-segment datasheet:
http://www.kingbrightusa.com/images/catalog/SPEC/SA08-11EWA.pdf

7. **Error handling** Error messages can be displayed at the terminal if there are any.

Some guidelines for the flow of the program:

1. Detect if the input is a dot, dash or an error
2. Record user's inputs
3. Decide if it is the end of the letter input
4. Branch logic to determine which letter to show given the input
5. Check and display error messages if there are any
6. Display the letter on the 7-segment display and serial console
7. Repeat from point 1

Challenging parts using the buzzer, RGB LED & potentiometer

Ensure the tasks above are completed and tested correctly before you attempt the following tasks. You can use any available GPIO pins for connecting the buzzer, RGB LED, button, potentiometer.

a) **Use the buzzer to give audio feedback in real time** – short beep for a dot, longer beep for a dash. The buzzer also gives negative feedback if the user fails to input within the allotted time; it gives a negative feedback if the inputs cannot be decoded. Make up a different buzz sound for negative.

b) **Use the RGB LED to give visual feedback in real time** – Green for a valid input (an input can be validly decoded into a letter); red for an invalid input.

c) **Use the potentiometer to set time limit** - When your program starts, it prompts the user to tune the potentiometer to set a time limit for inputs. The time limit should be set to 4 seconds by default.
The user can turn one potentiometer to set the overall time limit for each letter input (e.g., 3 seconds per letter or 2 second for faster operators). The terminal should display the newly set time limit.
If the user doesn't input valid dots or dashes (which can be decoded) within this time, the RGB LED will then display red.

d) Your program should count how many correct letters have been put in, ignore any incorrect inputs, if the number of correct letters reaches four, the buzzer plays a short tune, the program displays a decoded message on the console. The program then prompts the user to decide whether they want to continue or exit the program.
    a. Left button = "Yes" to reset all components and continue;
    b. Right button = "No" to turn off all components and terminate the program;
    c. The RBG LED lights up briefly when the user is making the choice (red = terminate, green = continue).

4. Groups

Groups are composed of 3 or 4 students. You are free to form your own group. Inform the module leader if there is any change of groups. The mark you receive for the coursework is the coursework group's mark. **The markers reserve their right to adjust individual group member's marks, for example, for lack of engagement with the group work or non-attendance at the lab demonstration, a mark may be reduced to 0.**

5. Resources

The coursework is challenging. However as you progress through the labs you can acquire the skills you need for certain parts of the coursework.

IMPORTANT NOTE: YOU ARE WELCOME TO USE THE PROVIDED LAB CODE AS YOU WISH IN CREATING YOUR COURSEWORK CODE. The given code template is optional for you to use.

Discussions between Groups about the code and the Raspberry Pi Pico boards. Students in one group are encouraged to answer the questions of students in other groups, but are reminded not to actually give away actual solution code (that would count as plagiarism!). Such questions should be as concrete as possible, and answers should be verbal descriptions, not snippets of code.

6. Deliverables

The program including all source code files (*.c and *.h files) should be submitted to Surreylearn. In addition, the code should be documented well, using comments.

Unbalanced Group contribution:
There is a folder titled **"Unbalanced Group Contribution"** within the **"Assignment Folders"**. If your group had members who did not contribute equally, you can upload a text file to this folder. In the text file, please provide the percentage contribution of each member and remember to include your group number.
If all members contributed equally, there is no need to submit anything to this folder. In the absence of a submission, we will assume that all members contributed equally.

Demonstration In addition your group will be required to demonstrate your coursework using the submitted source code with the required Raspberry Pi Pico kits in Lab 10/11 (see front page). **ALL GROUP MEMBERS SHOULD BE PRESENT! If you cannot attend (e.g., illness), please write an email to the module leader and the team members with the reason and indicate clear contribution of the project.**

The source code must be zipped into an archive called group XXX.zip (with XXX replaced with your group number e.g., 004 for group 4) and uploaded to Surrey Learn by the deadline Friday 22/11/2024 at 4pm.
It is sufficient if ONE group member actually uploads the archive file. – However make sure it is named as above to make sure we can easily identify your group.

Marking

The coursework counts 40% to your final mark in COM1031. This coursework is marked out of a maximum 100 marks as follows:

Lab demonstration up to 100 marks, distributed as follows:

Criteria	**Marks**
Start of the program - welcome message [2 marks] - 7-segment display turns on & off [3 marks]	5
External circuit is connected correctly (use the lab template)	5
Program correctly recognises, and displays to the serial console, letters which have a morse code of one signal (that is E and T).	10
Program correctly recognises, and displays to the serial console, letters which have signal length of two A,N,I,M.	10
Program correctly recognises and displays letters with morse code of three signals. (i.e., S, U, R, W, D, K, G, O)	5
Letters up to morse code length 4. (e.g., H,V,F,L,P,J,B,X,C,Y,Z,Q)	5
Error handling - if the button is pressed longer than 700ms, the serial console displays this is an error input; the 7-segment displays '8' for an error. - Similar for morse code that does not correspond to any letter of the alphabet.	5 5
Buzzer to give audio feedback – short beep for a dot, longer beep for a dash; – gives negative feedback buzz if the user fails to input within the allotted time;	3 3

– gives a negative feedback if the inputs cannot be decoded.	3
RGB LED to give visual feedback - green for a valid input (an input can be decoded into a letter); - red for an invalid input;	 3 3
Change the time limit for the inputs using the **potentiometer** (Test this time limit with the buzzer and RGB LED) - the terminal displays the newly set time limit - increase the time limit for the inputs - decrease the time limit for the inputs	 3 5 5
Any four valid attempts can be decoded correctly, - the buzzer plays a short tune, - the program displays a decoded message on the console	 5 3
After 4 valid attempts, the user can decide whether they want to continue or exit the program by pressing the buttons. - Left button = "Yes" to reset all components and continue; - Right button = "No" to turn off all components and terminate the program - The RBG LED lights up briefly when the user is making the choice (red = terminate, green = continue).	 5 5 4

Feedback. Feedback and marks will be provided via SurreyLearn to students before the written examination.

IMPORTANT: please do not post your coursework code online (e.g., social media) and never share your code with other students who are not in your coursework group even after it's been submitted and marked.