

CPSC 359 – Winter 2022
Assignment 3
SNES Controller Device Driver
Weight 8%
Due: February 22nd @ 11:59pm
Maximum points: 100

Objective: Build a simple device driver for a SNES controller. The driver will be used in the next part of the project as the primary input device for your interactive game.

This Assignment Lead TA: Desmond Larsen-Rosner <drlarsen@ucalgary.ca>

Deliverables:

Write a program that can interact with the SNES controller from the Raspberry Pi. Print the creator name(s) at the beginning.

1. Print "Please press button..."
2. Wait for user input.
3. If user presses a button, print a message on the screen indicating the button pressed.
4. Loop back to step 2 if any key other than START is pressed.
5. Pressing the "START" button will end the program displaying an exit message.

Example session:

```
Created by: John Smith and Sarah Smith

Please press a button...      (User Presses Joy-pad RIGHT button)

You have pressed Joy-pad RIGHT

Please press a button...      (User Presses Y button)

You have pressed Y

Please press a button...      (User Presses START button)

Program is terminating...
```

Notes:

1. Use at least the following subroutines:
 - a. Init_GPIO: the subroutine initializes a GPIO line, the line number and function code must be passed as parameters. The subroutine need not be general: it just needs to work for the three SNES lines.
 - b. Write_Latch: writes a bit to the GPIO latch line
 - c. Write_Clock: writes to the GPIO Clock line
 - d. Read_Data: reads a bit from the GPIO data line
 - e. Wait: waits for a time interval, passed as a parameter.
 - f. Read_SNES: main SNES subroutine that reads input (buttons pressed) from a SNES controller. Returns the code of a pressed button in a register.
 - g. Print_Message: prints an appropriate message.

2. Submit a tar-ball of your entire project directory, including a makefile, your source and compiled code, as a file named c359-<student_id>-a3.tar.gz

Grading:

1. Display creator names & messages	6
2. Correctly reading/printing buttons	50
3. Modularity	24
4. Loop back (except for "START")	10
5. Well documented code	10
Total	100 points

Programs that do not compile cannot receive more than **15 points**. Programs that compile, but do not implement any of the functionality described above can receive a maximum of **25 points**.

Teams: You are advised to work with another student in class in order to complete the assignment, but you are not required to do so. Peer evaluation in teams may be conducted.

Submission: Submit a .tar.gz file of your entire project directory (including source code, make file, README, etc) to the appropriate dropbox on Desire2Learn. Only one submission per team is required.

Late Submission Policy: Late submissions will be penalized as follows:

-12.5% for each late day or portion of a day.

Hence, no submissions will be accepted after 8 days (including weekend days) of the announced deadline

Academic Misconduct: Any similarities between submissions will be further investigated for academic misconduct. Your final submission must be your team's original work. While you are encouraged to discuss the assignment with colleagues outside your team, this must be limited to conceptual and design decisions. Code sharing outside your team by any means is prohibited, including looking at someone else's paper or screen. Any re-used code of excess of 5 lines in C and MAL (5 MAL instructions) must be cited and have its source acknowledged. Failure to credit the source will also result in a misconduct investigation.

D2L Marks

Marks posted on D2L are subject to change (up or down).