

Lab 2 – I/O Programming over Raspberry Pi

- 5% of your final grade
- Individual lab, not a team lab, no collaboration is allowed!
- Check out our [Academic Integrity policy](#) here. If two or more students' lab assignments are very similar, all will receive a minimum penalty of a zero on the given lab assignment.

Lab 2 consists of two sections.

Section 1 (50%): This section uses Python as the programming language and require you to complete 4 subtasks.

1. Dynamically and continuously display images over LED matrix. For instance, you could display a walking animal or a swinging pendulum.
 2. There are five modes of joystick operation: left, right, up, down and stay in the middle. You need to perform different reactions over LED matrix for different operations of joystick. For instance, if you move joystick to the left, temperature will be shown; if you move joystick to right, task 1 will be displayed over LED matrix and so on. In short, you should prepare five different reactions to five operations of joystick.
 3. Sense HAT has on-board sensors to detect directions according to g-force. G-force works on three axes, X, Y and Z. If one axis has two (+ and -) directions, there are six directions. Your task is to show reactions towards different directions when you tilt your sense HAT and Pi. For X and Y directions, you need to show a row/column one by one over LED from direction of “-X/-Y” to “+X/+Y” if your Pi is tilting to “+X/+Y” direction and vice versa. You need to use eight different types of color instead of one type of color, each color for one row/column for display. For Z directions, if your Pi is facing up you should display message over LED matrix. You need to print message to shell if your Pi is facing down.
 4. Display oblique lines from top left to bottom right one by one. For instance, the first line is origin of LED matrix and its coordinate is (0, 0); the coordinates for second oblique line are (1, 0) and (0, 1) and finally up to last oblique line with its coordinate as (7, 7). Eight different types of color will be used in this sub task and will be symmetric based on the longest oblique line, which is also antidiagonal of 8 x 8 LED matrix.
- For sub task 3 and 4, you can only use `set_pixel()` function, not `set_pixels()`, to turn on LED. You might consider using nested loop to complete this two sub tasks. 8 x 8 matrix table will be very helpful on finding patterns of coordinates when you get confused.

Section 2 (50%): This section uses C as the programming language.

In this section, you are required to write a system-call-based interrupt over Linux. In completing this section, you need to use timer inside Linux-based operating system to trigger interrupt. When timer expired, as advent of an interrupt, you need to send a signal to kernel to run specific function for handling this expired timer or interrupt.

Some basic technical tutorials about Linux kernel programming will be given in lab lectures by the TA.

Useful library could be `sys/time.h` and `signal.h`

Useful functions could be `memset()`, `setitimer()` and `sigaction()`

Useful structure variables could be `sigaction`, and `itimerval`