

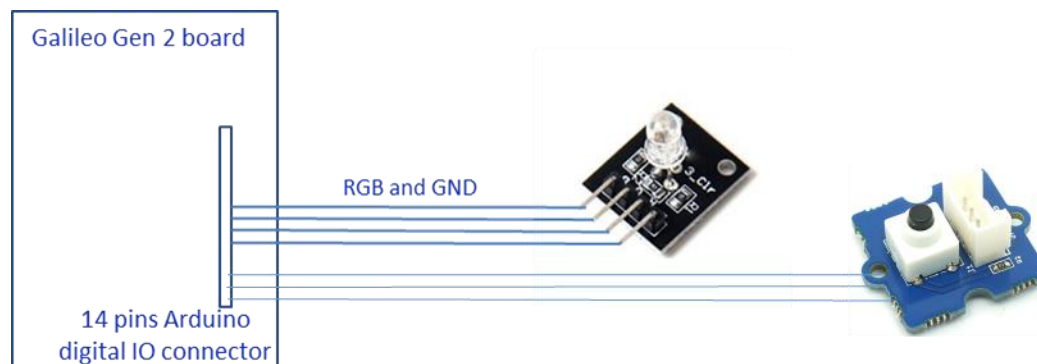
Assignment 1. GPIO control in Linux (100 points)

Assignment Objectives

1. To learn the software development skill for embedded systems.
2. To learn the basic programming technique for GPIO pins in Linux system.
3. To learn IO control in user-level programs.
4. To learn pin multiplexing mechanisms and programming approaches in embedded systems.

Project Assignment

In Linux systems, you can develop various kinds of programs to perform IO operations. In this assignment, you are required to write a user-space program to interface with a RGB LED and a push button. The programs can control LED luminous Intensity, as well as the combination of red, green and blue colors. As shown in the following diagram, the LED and push button are connected to any of 3 GPIO pins of the Arduino digital IO connector of Galileo Gen 2 board.



The input arguments to your programs consist of an integer to specify the percentage of duty cycle of a PWM signal for luminous Intensity. For instance, the input “50” indicates that the LEDs will be on for 50% and off for 50% in every cycle duration. For the assignment, you can set the cycle duration of intensity control to 20ms. The digital IO pins that the RGB LED pins are connected to are fixed at IO0, IO1, and IO2 of the digital connector.

The RGB LED control that your programs must demonstrate is to apply the PWM signal to the combinations of one, two or all three LEDs in an 8-step periodic lighting sequence, where **each step is with a maximal duration of 1 second or until the push button is released (not “pressed”)**. The sequence should be displayed continuously until the program is terminated. An example lighting sequence could be {R, G, B, R&G, R&B, G&B, R&G&B, OFF}. In this lab, you will have to configure IO6, IO7, IO8 as GPIO output pins and connect these terminals to red, green and blue pins of the LED. Also, use IO2 of the shield as the GPIO input pin to read the push button status.

A user-space application for RGB LED display

In this assignment, your team should develop a user-space application program to perform the aforementioned RGB display operations. The main program is named as *RGBLed.c*. If needed, additional source programs and header files can be included in your development.

To control LED intensity and the display duration for each color, you will need to put out either 0 or 1 on the corresponding GPIO pins for certain intervals. For instance, for 50% intensity, the GPIO pin to red LED should be switched to OFF 10ms after it is switched to ON. To perform such timed actions, you can

consider to use `usleep` or `nanosleep` function, or a POSIX timer. In addition, please make sure to use macros for any literal constants or flags, e.g., `#define cycle_duration 20`.

Due Date

The due date is set to 11:59pm, July 6, tentatively.

What to Turn in for Grading

- Create a working directory to include your source files (.c and .h), makefile(s), readme file. Compress the directory into a zip archive file named **RTES-teamX-assgn01.zip**. Note that any object code or temporary build files should not be included in the submission. Email the zip archive to the instructor (3451675397@qq.com) by the due date and time.
- Please make sure that you comment the source files properly and the readme file includes a description about how to make and use your software. Don't forget to add each team member's name and student id in the readme file.
- There will be 20 points penalty per day if the submission is late. **If you have multiple submissions, only the newest one will be accepted.** If needed, you can send an email to the instructor and TA to drop a submission.
- **Your team must work on the assignment without any help from other teams and is responsible to the submission. No collaboration between teams is allowed, except the open discussion in the class.**
- Here are few general rule for deductions:
 - No make file or compilation error -- 0 point for the part of the assignment.
 - Must have `"-Wall"` flag for compilation -- 5-point deduction for each warning.
 - 10-point deduction if no compilation or execution instruction in README file.
 - Source programs are not commented properly -- 10-20-point deduction.