Assignment 2 SPI Device Programming and Pulse Measurement (100 points)

Exercise Objectives

1. To learn multi threaded applications in Linux environment.
2. To learn the basic programming technique in Linux spi and gpio devices.
3. To learn the event handling for spi and gpio devices.
4. To develop an application of distance measurement and animation display.

Lab Assignment

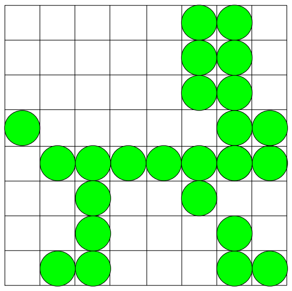
Most of the real world applications (vending machine, digital watch, etc) require some sort of display system to be able to interact with the world. In this lab you will be writing a multi-threaded application that is responsible of handling shared data, communication between different devices.

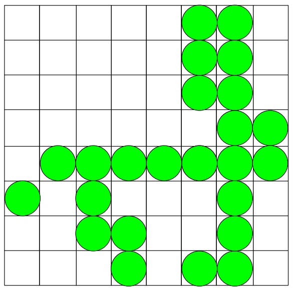
Many display devices are based on raster graphic image which consists of a rectangular matrix of pixels. To display an image, the ON/OFF (or intensity) data for all pixels is saved in a frame buffer and scanned in a fixed frame rate. In this assignment, we will use a simple 8X8 LED matrix to display patterns. The device is equipped with a MAX7219 driver. In addition to source current for LED display, the driver chip can buffer display data and scan digits using an internal oscillator.

Your program needs to create an animation by displaying a sequence of patterns on the LED matrix. To display a patter, 8 bytes of pattern should be written to the driver chip via SPI interface bus. Assume we will use the example definition of a patter:

```
typedef struct
{
        uint8 led[8];     // 8 bytes (64 bits) for 64  leds
} PATTERN;
```

By switching between the following two patterns, the LED matrix should show a running or walking dog. (This is an example display pattern. Please create something interesting, such as bouncing balls, for this assignment.)





The other device you will use in the assignment is a HC-SR04 ultrasonic sensor distance measuring module. After sending a "ping" signal on its trigger pin, the module sends 8 40khz square wave pulses and automatically detect whether receive any echo from a distance object. So the pulse on the echo pin width tells the time of sound traveled from the sensor to measured distance and back, i.e.,

Test distance = (pulse width * ultrasonic spreading velocity in air) / 2

To make your application interesting, you should use the measured distance to determine how fast your display object is moving (from walking to running) and the direction it moves (left or right when the distance object moves to the sensor or away from the sensor).

A user application program for distance-controlled animation

The assignment is to develop a user application consisting of two work threads to control the two devices and to enable a distance-controlled animation. You can use IO4 and IO10 of Galileo board to connect to the trigger and echo pins of HC-SR04. To detect rising and falling edges on the echo signal, you will need to "poll" or "select" GPIO interrupt events in your user-level program. To connect a SPI bus to the display module, you will use SPI_SCK (IO13), SPI1_MOSI (IO11), and IO12 (as SPI_CS) pins on the digital IO connector on Galileo board and spidev driver for data transfer to MAX7219.

Note that it is required to have few separate modules for IO operations, such as:

1. IO initialization
2. Control the display patterns on LED matrix
3. Measure distance from HC-SR04 sensor

Your program should contain the application logic, including calculating distance and controlling display patterns in timely manner. It may be reasonable to have two threads to handle the operations on the two devices. In addition, the program execution terminates gracefully when Ctrl-C is pressed, i.e., any running threads must exit first and the led display must be reset to blank.

Due Date

The due date is set to 11:59pm, July 9, 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-assgn02.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.
- 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.