ES6152- EMBEDDED SOFTWARE DEVELOPMENT PROJECT

Submitted by

Swarna Kamakshi Jayaraman G1601351B

Janani Namashivayam G1601363E

Table of Contents

Objective

To implement a traffic light system for pedestrian crossing, using event-driven programming approach.

Tools

Hardware:

- Raspberry Pi with SD Card installed with Raspbian
- One Ubuntu based PC
- One Ethernet Cable
- One Micro-USB Cable (for supplying power to RPi board by the PC)
- Pibrella Board

Software:

- WiringPi GPIO Access Library, usable with C and C++ programming languages
- PUTTY for SSH Remote access

Hardware Set-up Overview

Raspberry Pi is an embedded board with ARM11 based processor. This board runs embedded version of the Linux OS.

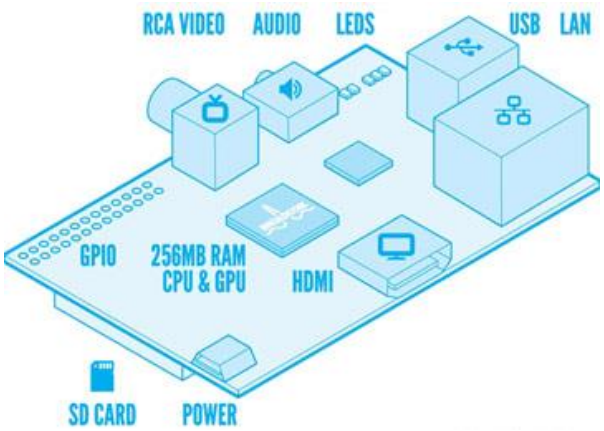


Fig 1: Schematic representation of RPi

The Pibrella board is interfaced with GPIO Pins of RPi. It has the following parts on-board:

- Tactile switch
- Red, green, and amber LEDs
- Piezo speaker
- 4 protected inputs
- 4 high-power outputs

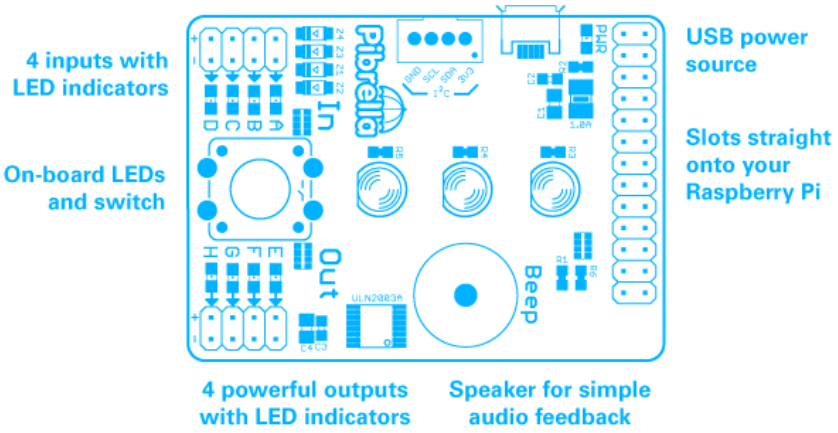


Fig 2: Schematic representation of Pibrella

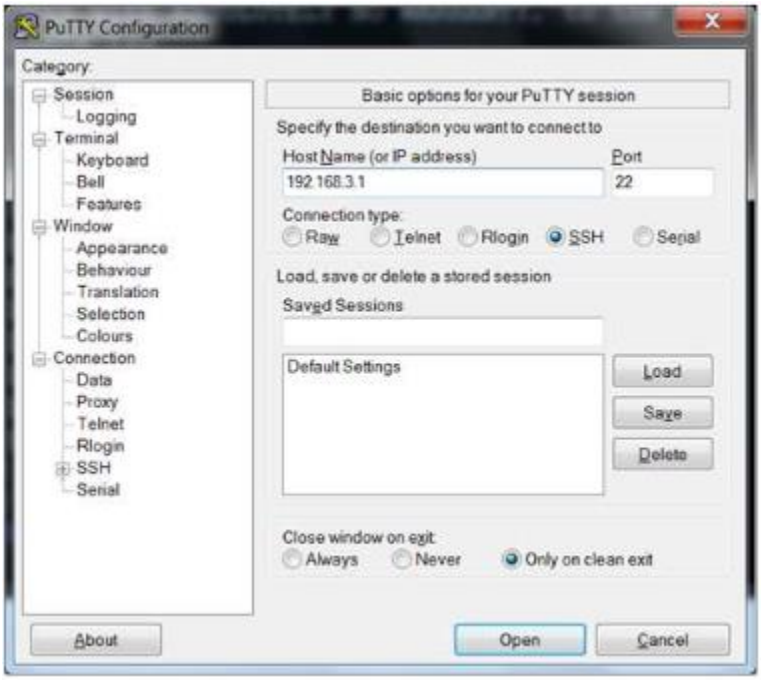
The figure 3 illustrates the set- up of the two boards.

Fig 3: Interfacing Pibrella with RPi

An Ethernet cable is connected from the Ethernet port of the RPi to the System that has a Linux OS to enable SSH connection through PUTTY.

Software Set-up

The following options should be given in PUTTY for SSH Connection.



Product working specification

Specification	Operation
1.	• Blinks 3 LEDs (Red, Green and Yellow) once for 1s when the system is powered. • Rings buzzer for 1s simultaneously with the previous LED blink
2.	Turns red LED ON and monitors Push button press indefinitely.
3.	When push-button is pressed, red LED is switched OFF after 2s.
4.	After red LED is OFF, green LED is turned ON for 5s
5.	• Green LED blinks at 1Hz for 5s. • Buzzer ON-OFF at 1Hz
6.	• Green LED blinks at 2Hz for 5s.

	• Buzzer ON-OFF at 2Hz
7.	• Yellow LED is turned ON for 2s • Buzzer rings for 2s
8.	Yellow LED is turned OFF and red LED is turned ON.
9.	When push-button is pressed, specifications 3-8 repeat.
10.	Between 3 and 8, the push button press shall have no effect.

Background Information

WiringPi library allows programming the RPi in C. Pibrella board is now compatible with the WiringPi PIN configurations.

PIN No.	**Component**	**Pin type**
7	Red LED	O/P
0	Yellow LED	O/P
2	Green LED	O/P
14	Push button	I/P
1	Buzzer	PWM O/P

The operations in specification 1, 5, 6 and 7 are to be performed concurrently. Hence, the concept of multithreading is used to achieve the same. As pthread does not support multiple arguments to a function, the arguments are packed in a structure and sent by the calling function(main). To ensure that the main program does not terminate before the thread execution is complete, pthread_join() is used in appropriate place.

Special features:

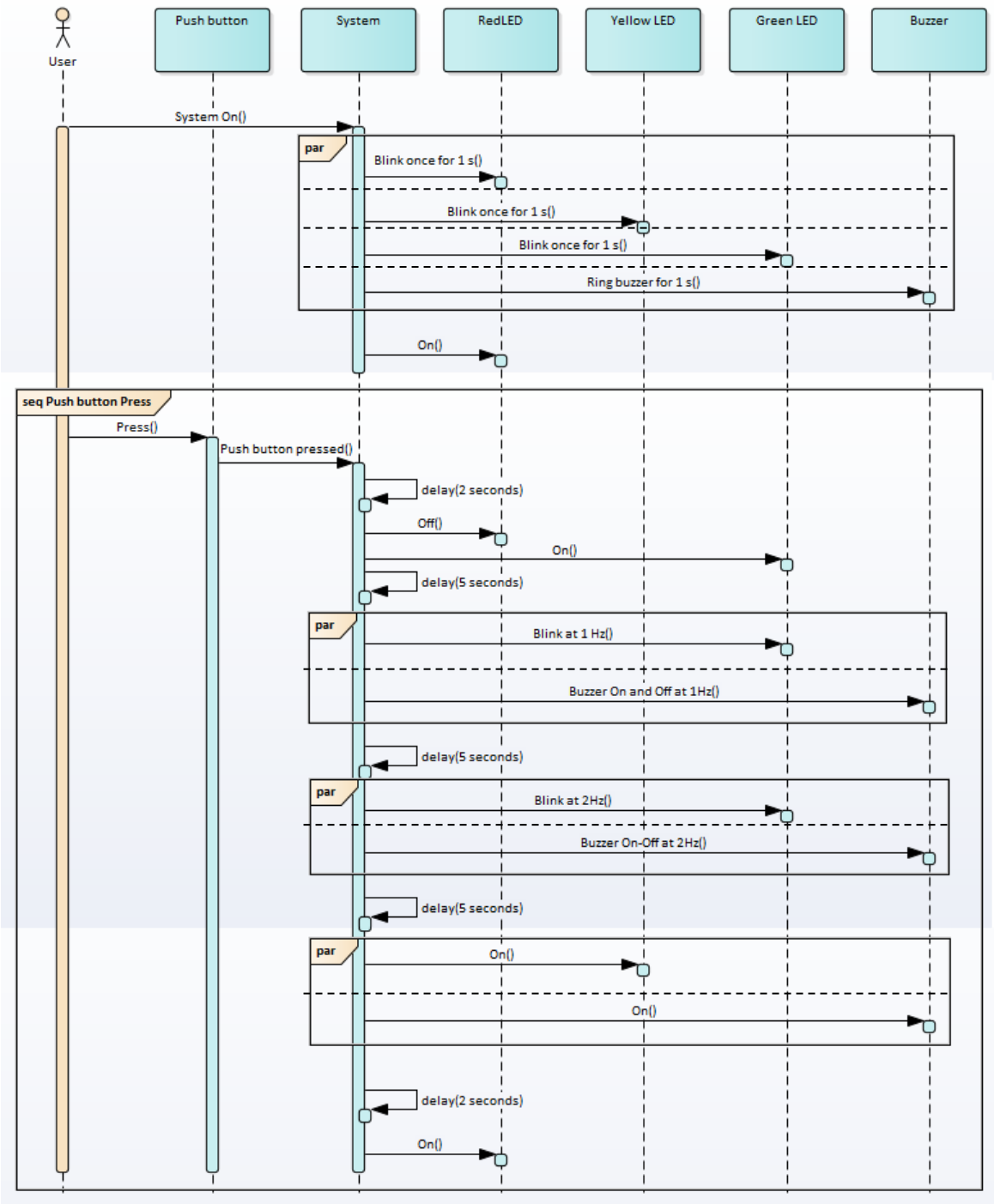
The code has been made generic for

- Led blinking, taking the arguments as PIN number, duration and frequency. Here, PIN number, duration and frequency are integral values.
- Ringing the buzzer at a particular frequency.
- Turning on the buzzer
- Turning on the LED

Thus, the utilization of these function for other PIN numbers in the future is made easy.

Product Design

1. Sequence Diagram

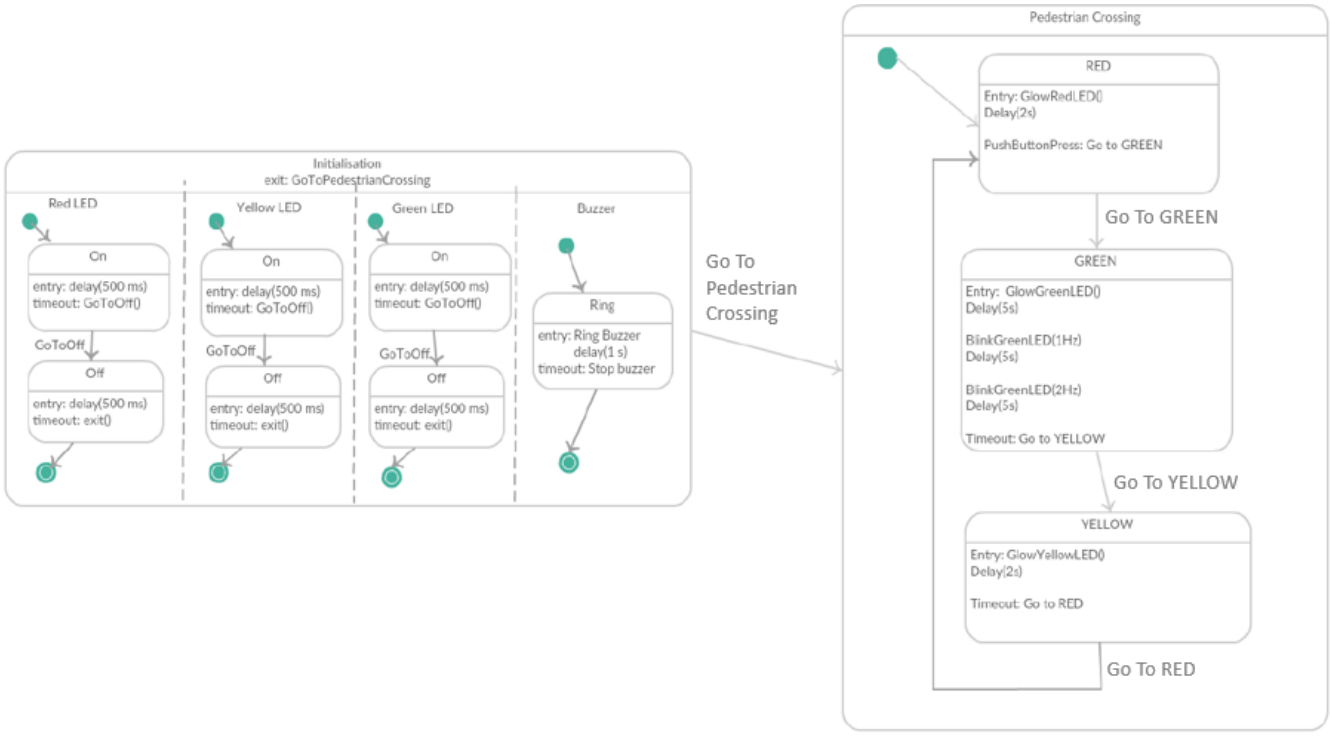


Here, the operations performed inside “par” block are concurrent operations, separated by a dashed line.

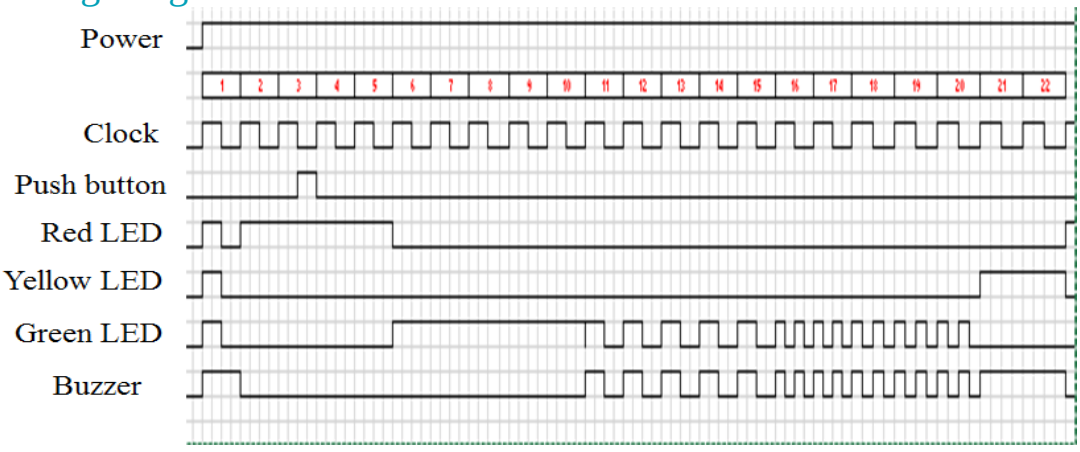
As we have used a delay() function instead of timer, we are not waiting for any acknowledgement from the timer module on timeout. All operations are performed synchronously. Hence, all operations have been denoted using synchronous transitions.

After spawning the threads, they are synchronized using pthread_join() function. So the parallel threads are also denoted as synchronous calls.

2. State Chart



3. Timing Diagram



4. Activity Diagram