14 Combining multiple Buttons and LEDs

Activity: Work the following exercises as directed by your tutor:

Purpose of this Activity

To learn about input (sensing) and output (indicating) interfaces of microprocessor systems.

Learning Outcome

- To be able to combine input and output signals of a microprocessor.
- To competently recall and use functions to use for I/O on the RPi.
- To be able to write, compile and test C code from given specifications.

Task 1

Following on from last session's code that read and displayed the state of one or several digital input lines (simulated with buttons), finish the following functionality:

- Use I/O ports B1, B2, B3 to read the three pushbutton states.

- Use I/O ports B9, B10, B11 as the corresponding output display on the LEDs.

Task 2

Extend the code such that if buttons B1 and B2 are pressed simultaneously, the while loop exits and the Gertboard interface clear up routines are executed.

Functions to use:

```
while ( condition )
{
    statements to execute while condition is true (<>0)
}
```

and

```
if ( condition )
{
    statements to execute if condition is true (<>0)
}
else
{
    statements to execute if condition is false (=not true =0)
}
```

Task 3

Find and write down all new C-functions that we have encountered over the last few sessions with a brief explanation and example of how to use them.

Task 4 - Starting the Digital Lock

A) Write a program that counts how often a button (1,2,3) has been pressed and displays the outcome after each button press on the screen.

   You can use individual integer variables for these counters or an integer array variable for all three.

   a. When implemented, consider how you can ensure that only button press changes get counted, and not a running count while the button is pressed.

B) Can you write a program that acts as digital lock, i.e. buttons have to be pressed a pre-determined number of times, e.g. B1,B2,B3 == [4,7,3], for an output line to go high to actuate the release of a lock?

   a. Consider counting presses only to single digits (0…9) and then wrap around. How can you implement this?