15 Implementing an electronic lock

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

Purpose of this Activity

To learn about controlling button input count and implementing an application to control an electronic three digit lock.

Learning Outcome

- To be able to combine input and output signals of a microprocessor.
- To be able to determine and use signal edge transitions.
- To be able to implement an application from specifications.

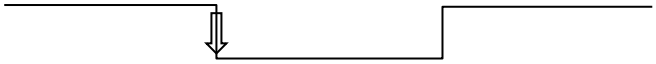
Task 1

Following on from last session's code that counted up while a button was pressed, implement the following functionality:

- At present, the counter counts up every time the while loop is run. It should only count when a button is actually pressed (and not while being held down).

  Implement code to detect the negative edge on the input signal line. Only when a transition from a previous register value [HIGH] to the current register value [LOW] occurs should a button press be registered.

  Roll out this change to all three buttons.



Detect button
press here.

- Also implement a 'roll-over' function that ensures that the three counters only count up in digits from 0 to 9:

```
if (B1count >= 10)
   { B1count = 0; }
```

Task 2

We finish implementing the code for the electronic lock.

- Remove the code that detects the simultaneous button press of B1 and B3 to exit the loop.

- Implement code that compares the stored button presses with a pre-defined secret 3-digit code.

  Remember: if the secret code is 247, this can be written as 2x100 + 4x10 + 7x1.

- When the secret code is entered correctly, output line B9 should light up (e.g. to actuate an electromagnetic lock opening mechanism).

Note 3

Revise the function of the following operators:

&

&&

|

||

==

!=

Task 4

After implementation, test the code lock thoroughly.
Does it work as expected?