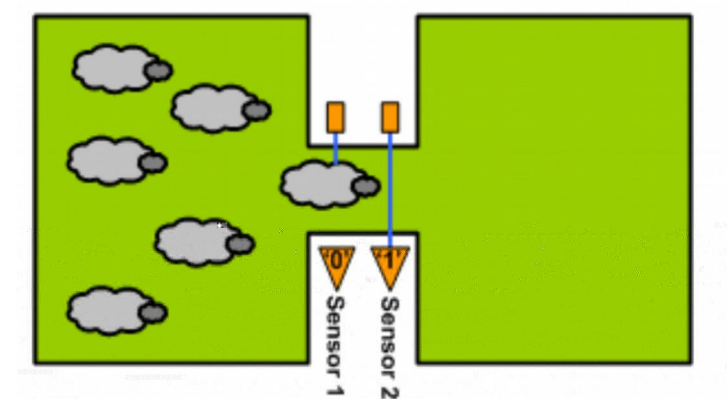


## MICROCONTROLLER PROBLEM and SAMPLE SOLUTION

From the book "Introduction to microcontroller programming" from [www.matrixtsl.com](http://www.matrixtsl.com)

### Exercise 7 Making Decisions

#### Program 8:



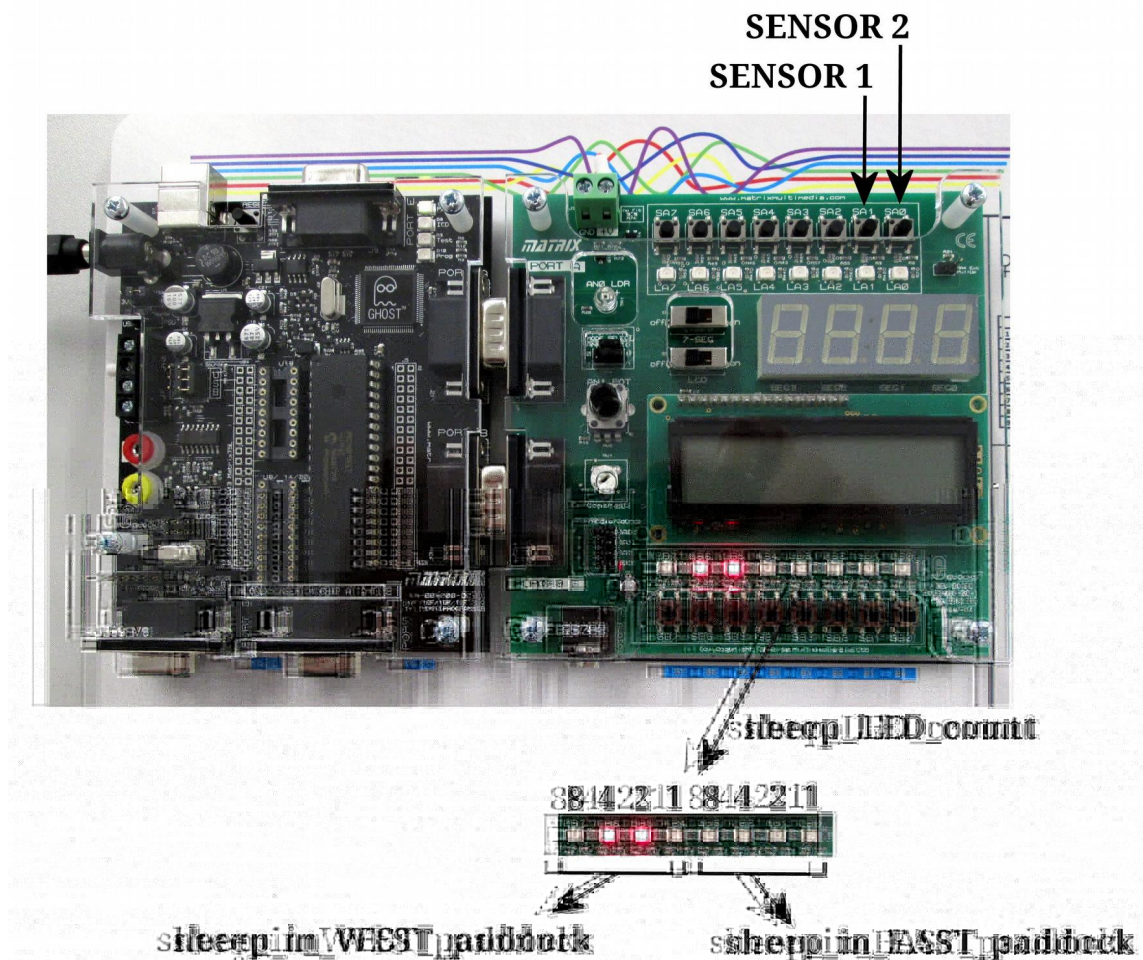
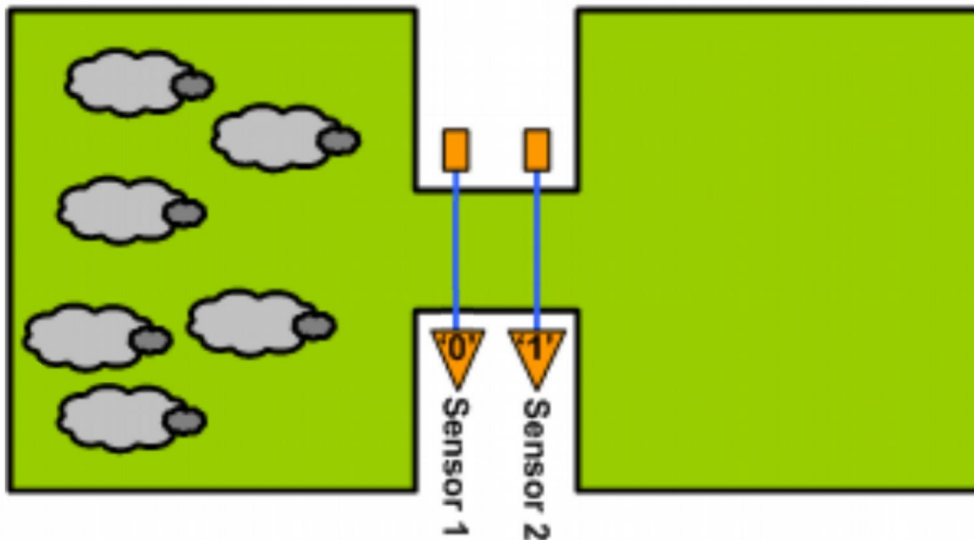
Six sheep are allowed to wander between two paddocks. Between the two paddocks, there are two sensors. Two are needed because sometimes a sheep triggers one sensor and then turns back.

write a program that counts and displays the number of sheep in each paddock:

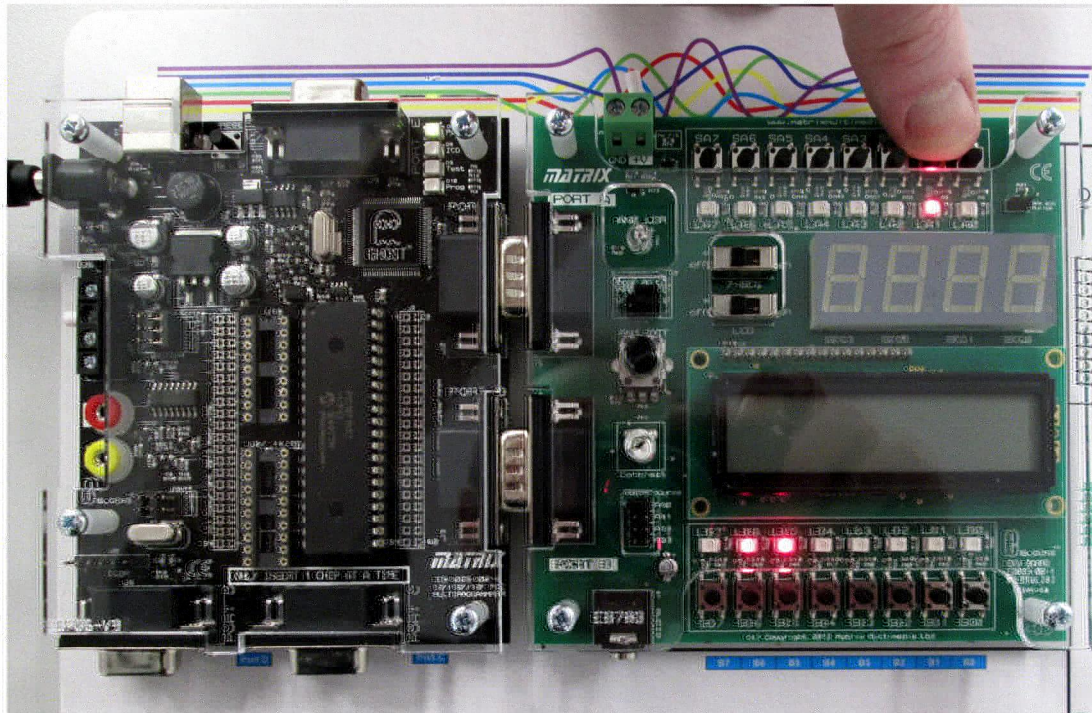
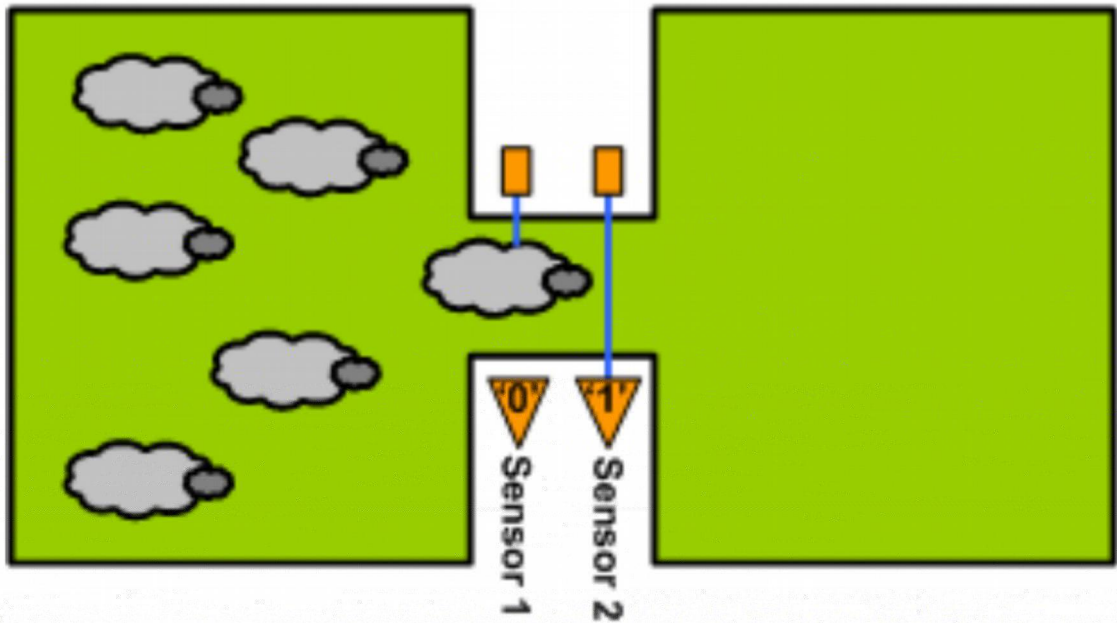
- the results are shown in binary form on the LEDs of port B;
- the four LSB LEDs show the number in the left-hand paddock;
- the four MSB LEDs show the number in the right-hand paddock;
- when switch '0' is pressed, it represents a sheep triggering Sensor 1;
- when switch '1' is pressed, it represents a sheep triggering Sensor 2; (A sheep is longer than the gap between the sensors!)

save this program, download it to the microcontroller and test it.

## HOW TO REPRESENT THE PROBLEM/SOLUTION WITH eBLOCKS:

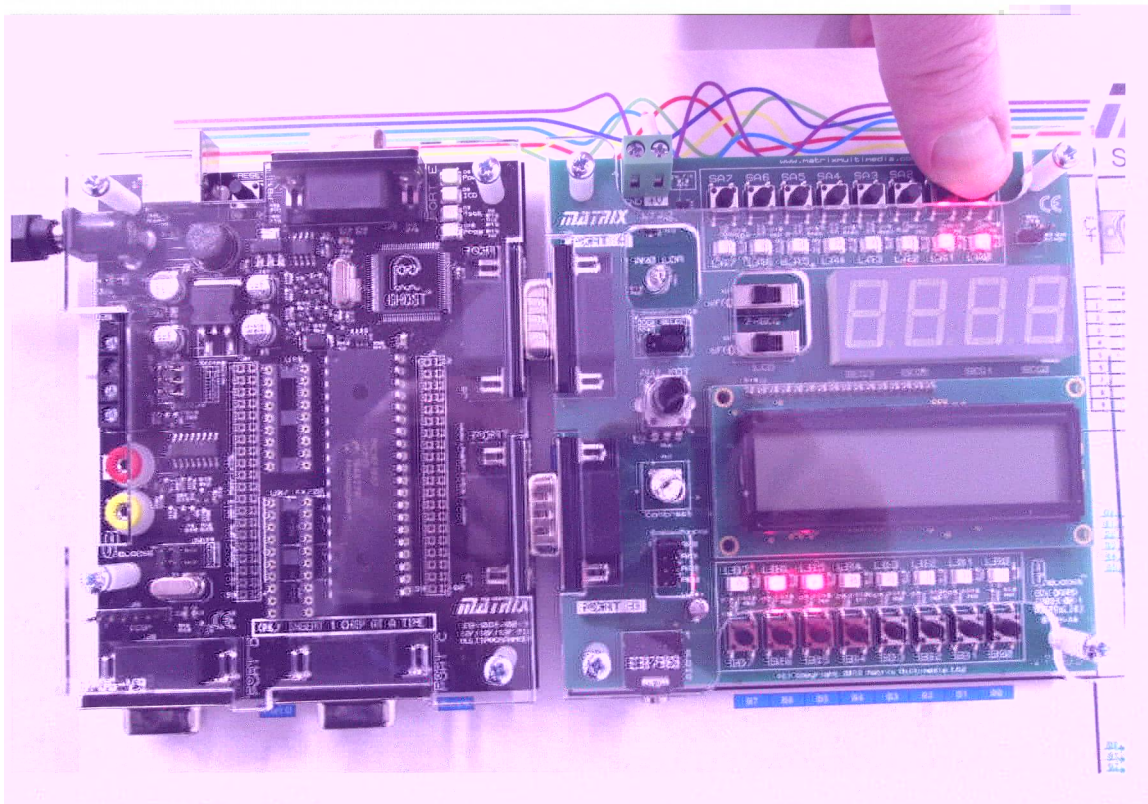
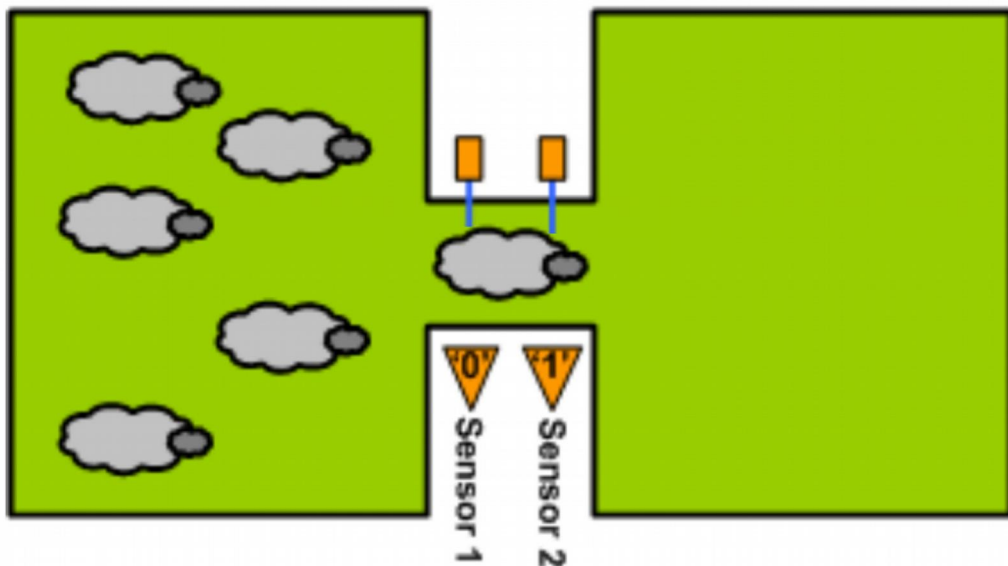


You start off with 6 sheep in the West paddock and none in the East paddock.

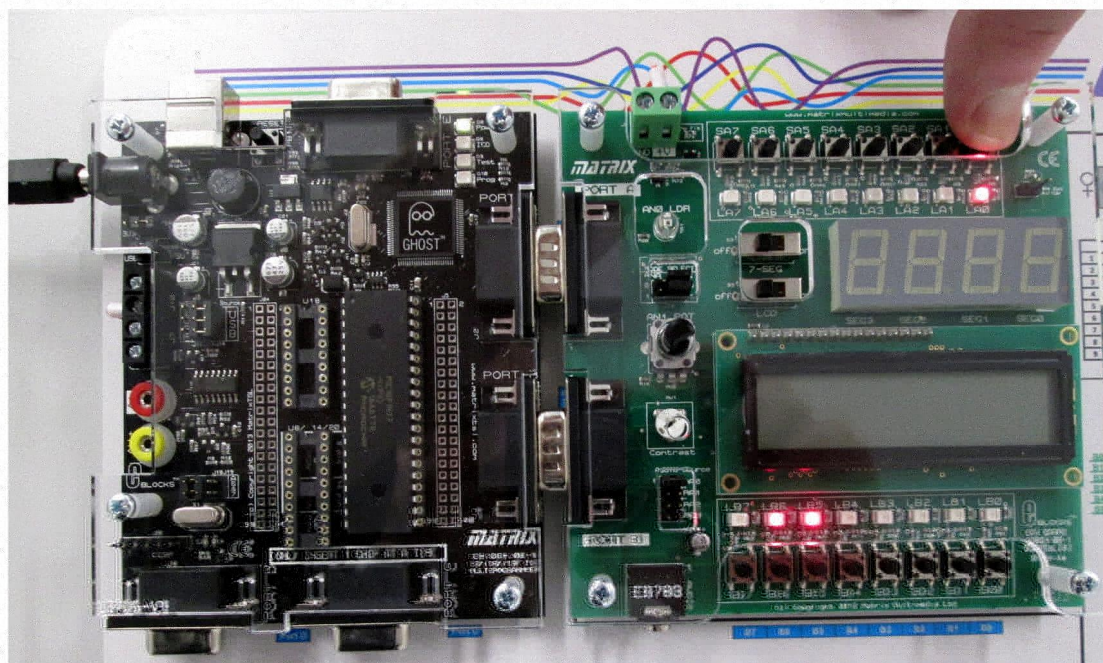
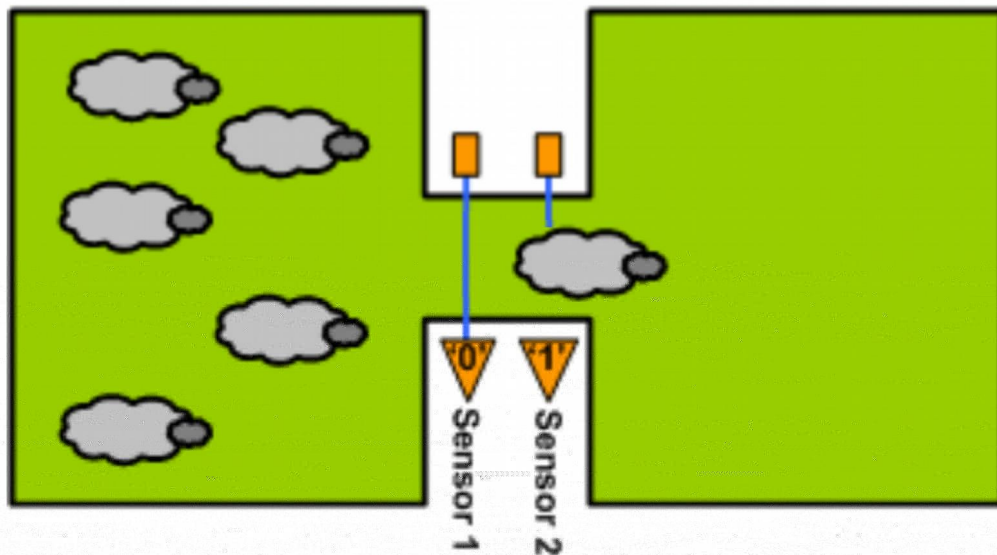


When a sheep blocks sensor 1, you still have 6 sheep in the West paddock and none in the East paddock. The sheep could change it's mind and back out of the alley.



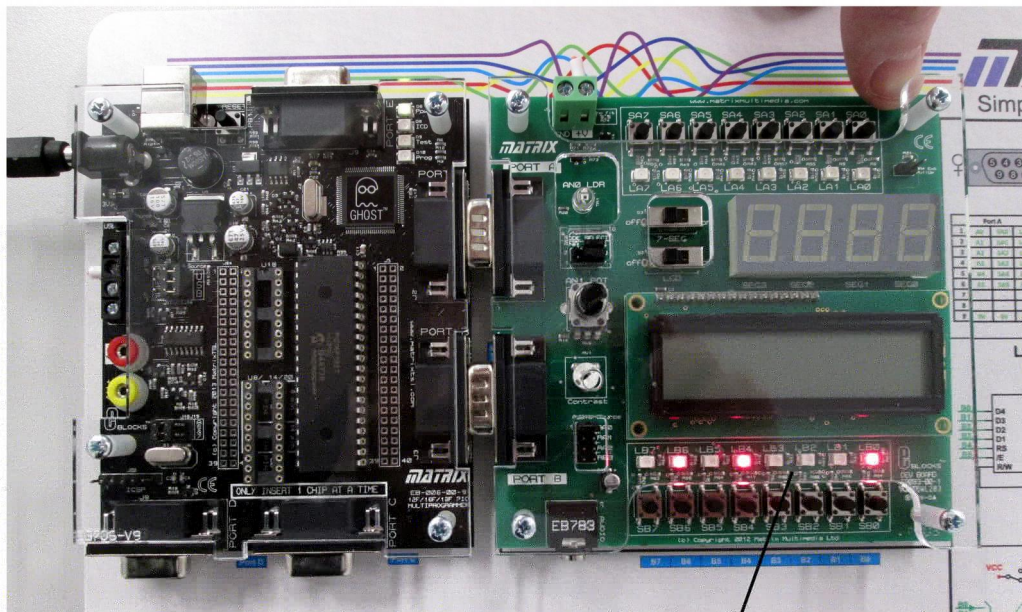
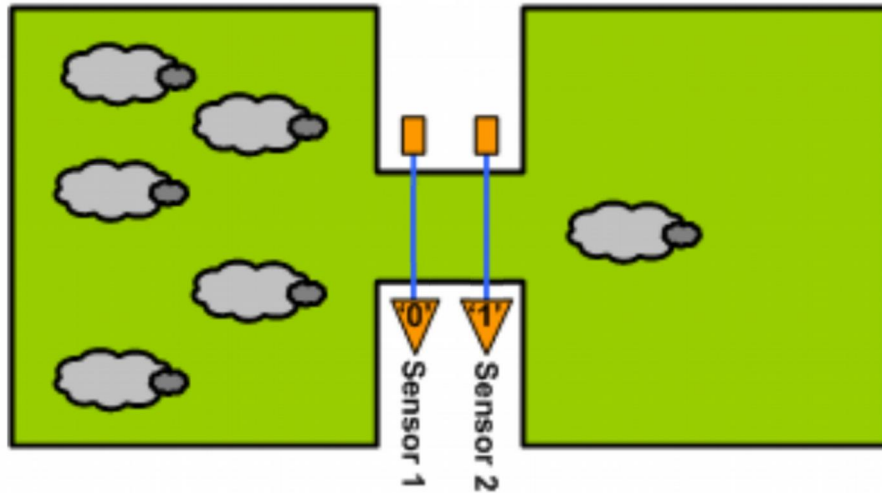


When a sheep blocks sensor 1 and sensor 2, you still have 6 sheep in the West paddock and none in the East paddock. The sheep could still change it's mind and back out of the alley.



When a sheep moves further into the alley, blocking sensor 2 and unblocking sensor 1, you still have 6 sheep in the West paddock and none in the East paddock. The sheep could still change it's mind and back out of the alley.





sheep\_LED\_count

8 4 2 1 8 4 2 1

sheep\_in\_WEST\_paddock

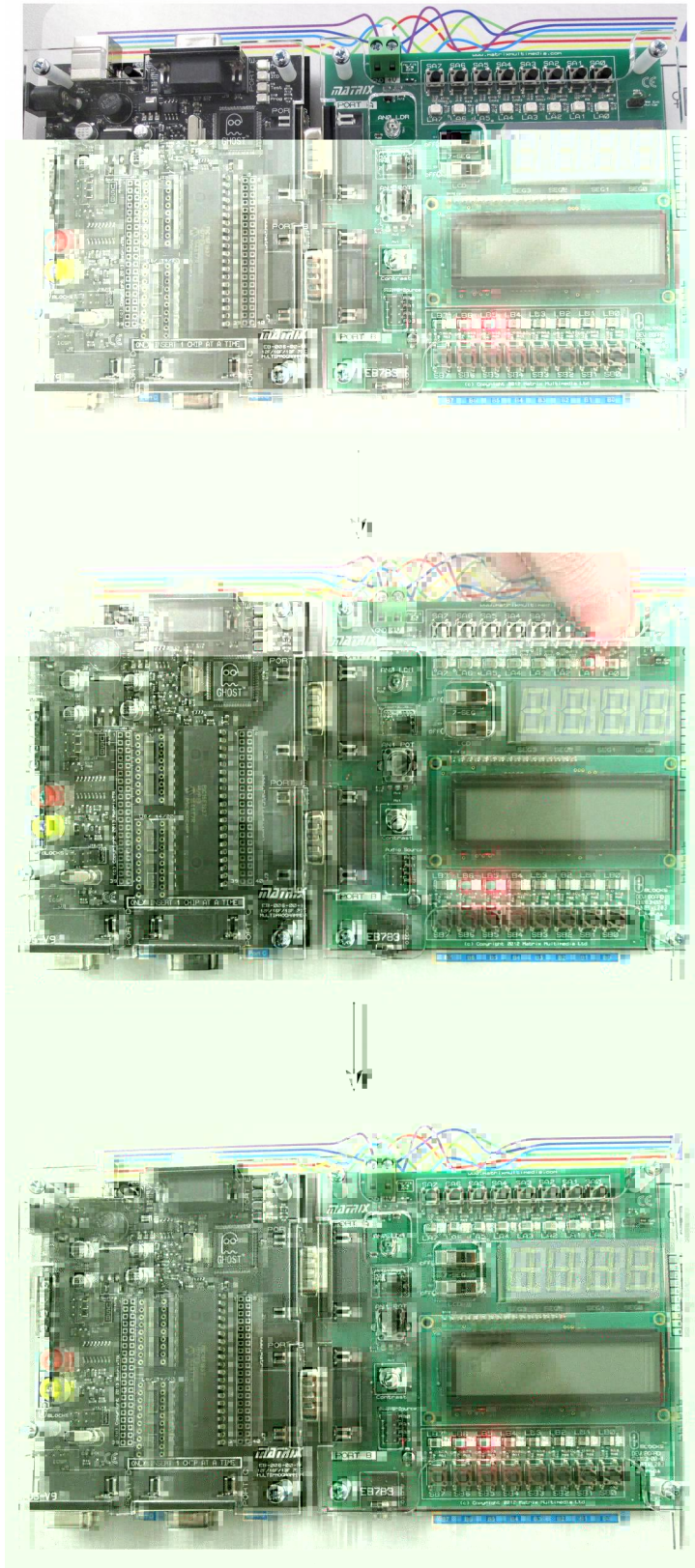
sheep\_in\_EAST\_paddock

When a sheep moves completely out of the alley, unblocking both sensors, you now have 5 sheep in the West paddock and 1 in the East paddock.

## INDECISIVE SHEEP !

When testing your solution, make sure it can cope with sheep that change their minds in the alley.

**Scenario 1 - sheep blocks sensor 1, then backs out of alley:**



Scenario 2 - sheep blocks both sensors, then backs out of alley.

Scenario 3 - sheep almost completes journey, then backs out of alley.

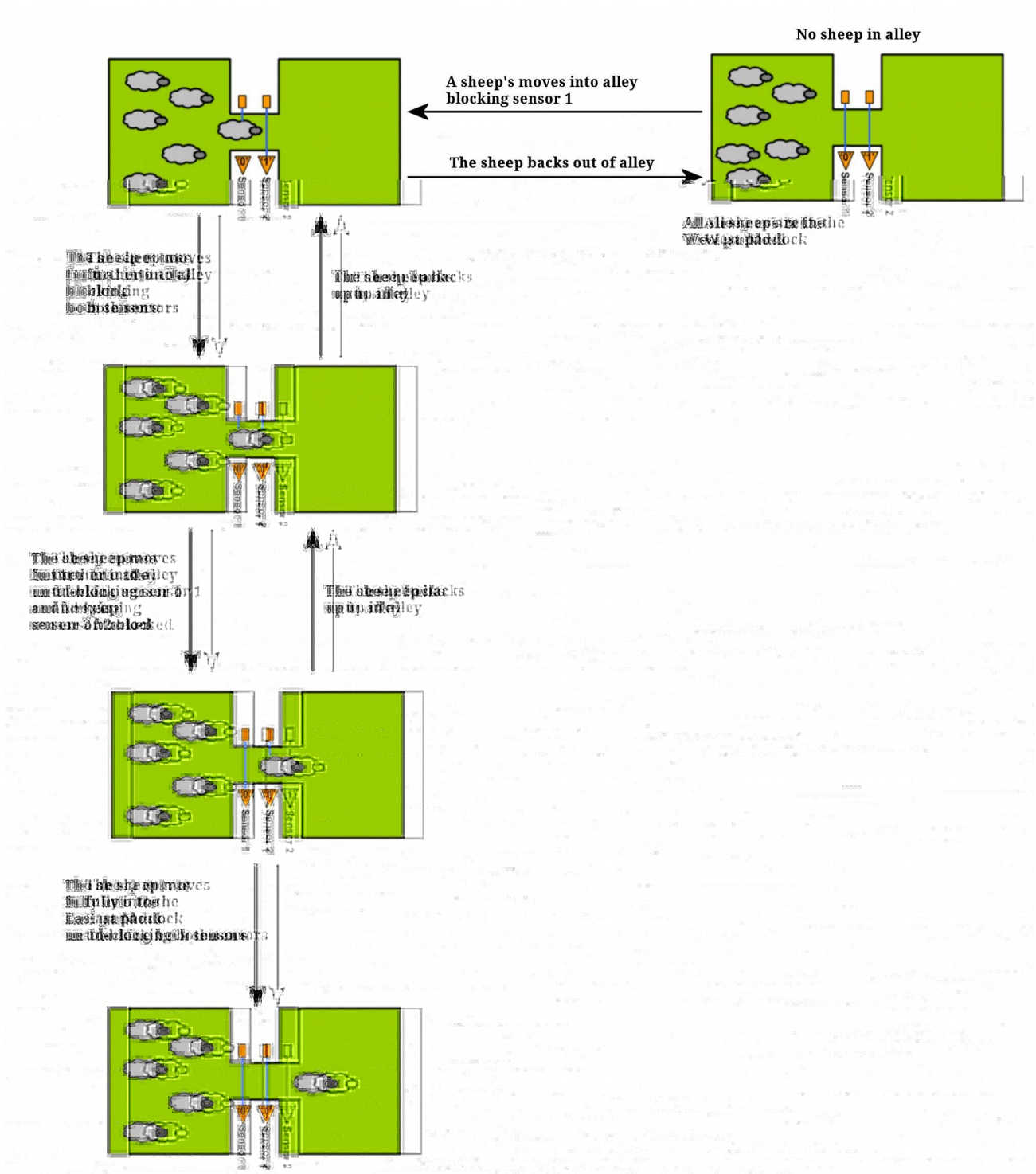
Scenario 4 - sheep almost completes journey, then backs up in alley,  
then changes it's mind again and continues to other paddock.

Note: If you are using the lecturer's sample solution for this problem,  
Scenario 4 fails. Can you find the problem and fix it?



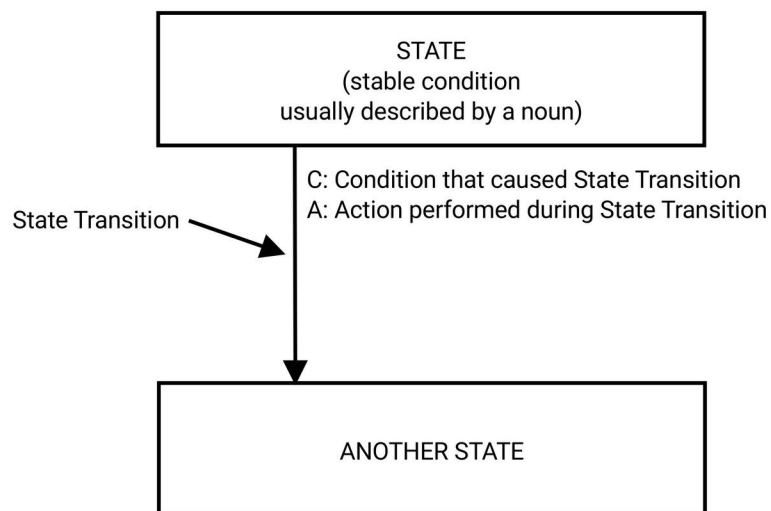
## HINT 1:

Translating the problem into a diagram can help you see the problem (and a possible solution) better than a wordy description.





## State diagram - key to symbols:



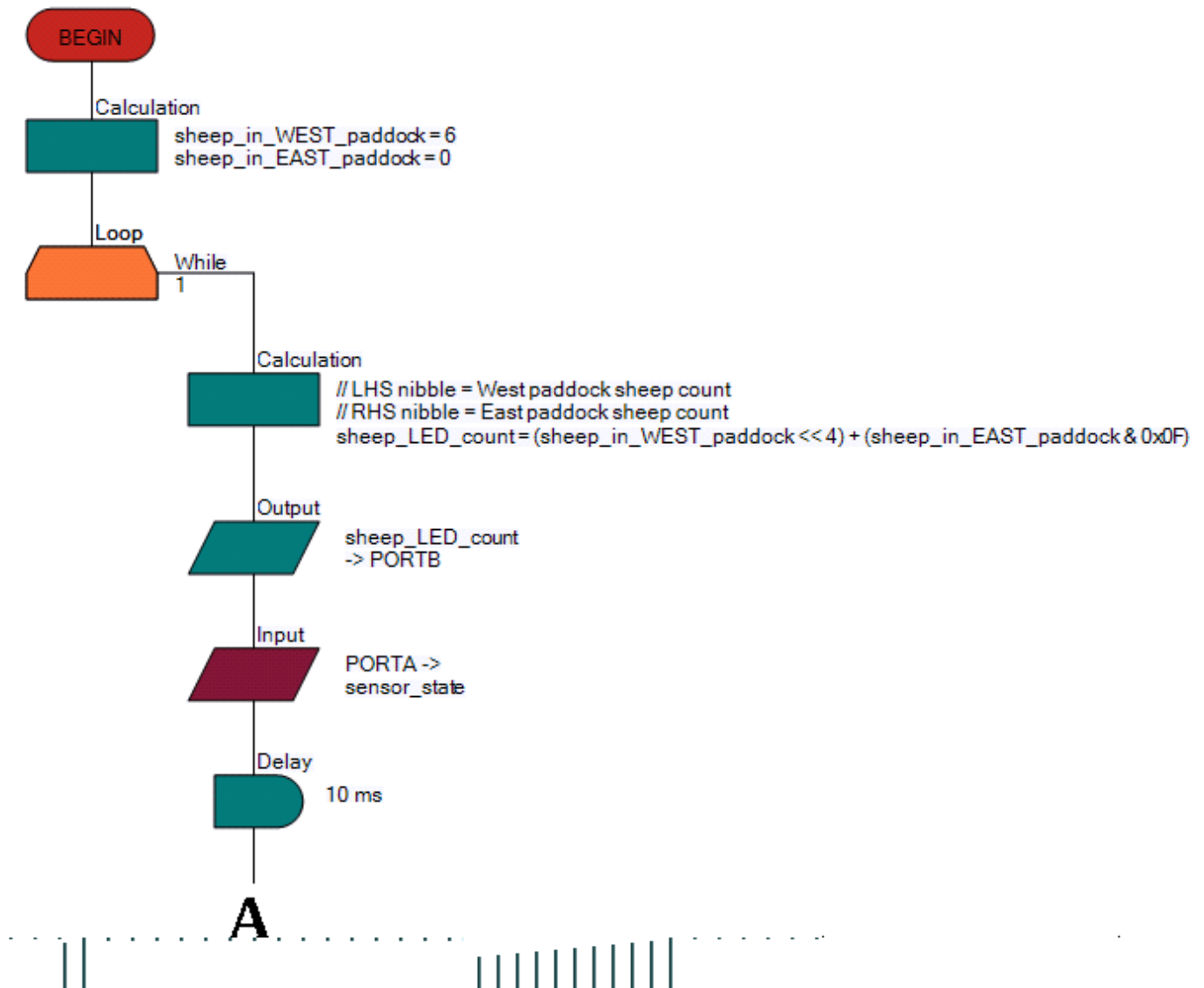
## Rules of state diagram design:

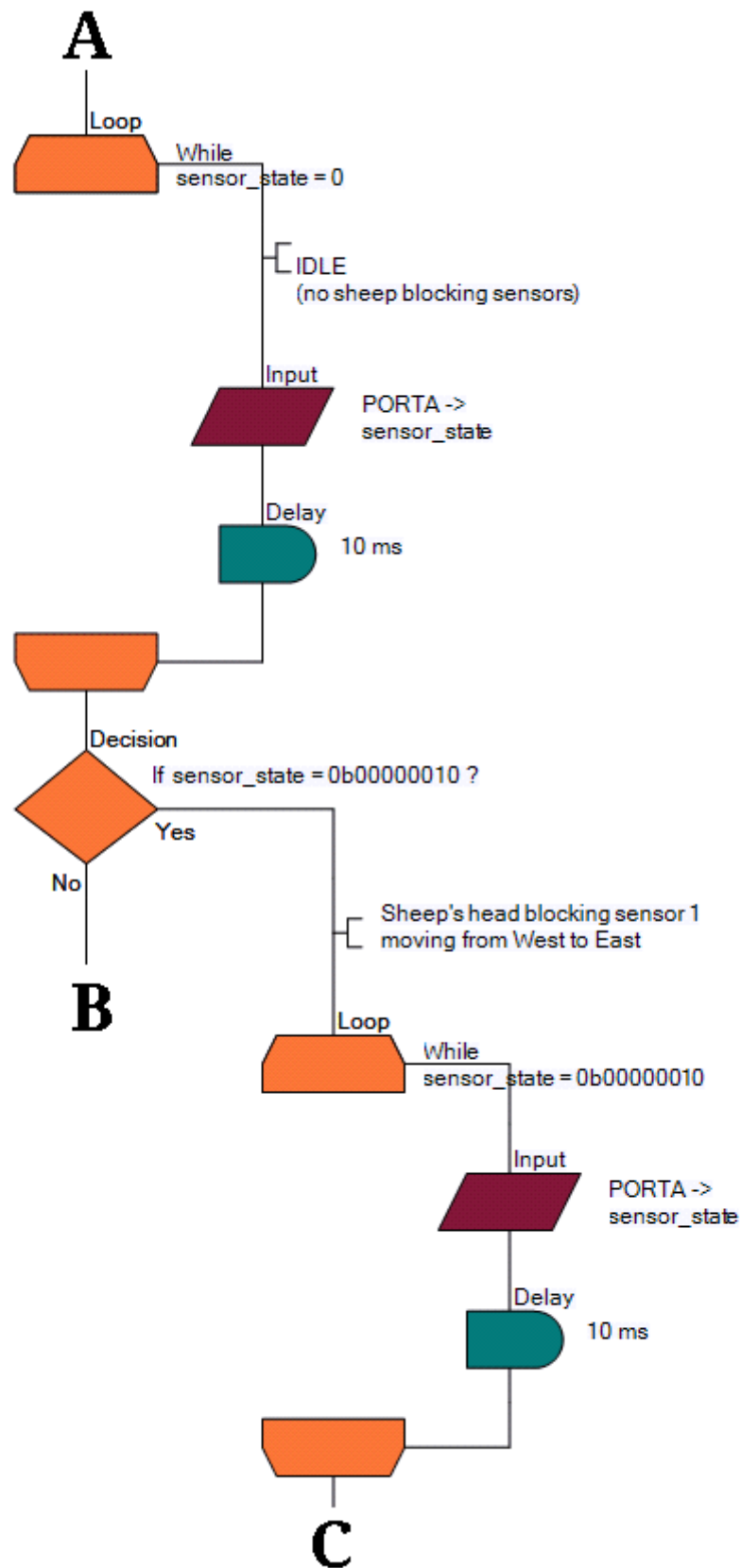
- A state represents a stable condition of the system (usually described by a noun - e.g. "IDLE").
- A condition is needed to move between states (a state transition) and must contain a verb - e.g. "switch pressed".
- During a state transition, an optional action (another verb) can occur - e.g. "update speed display".



### HINT 3:

(taken from the lecturer's sample solution)





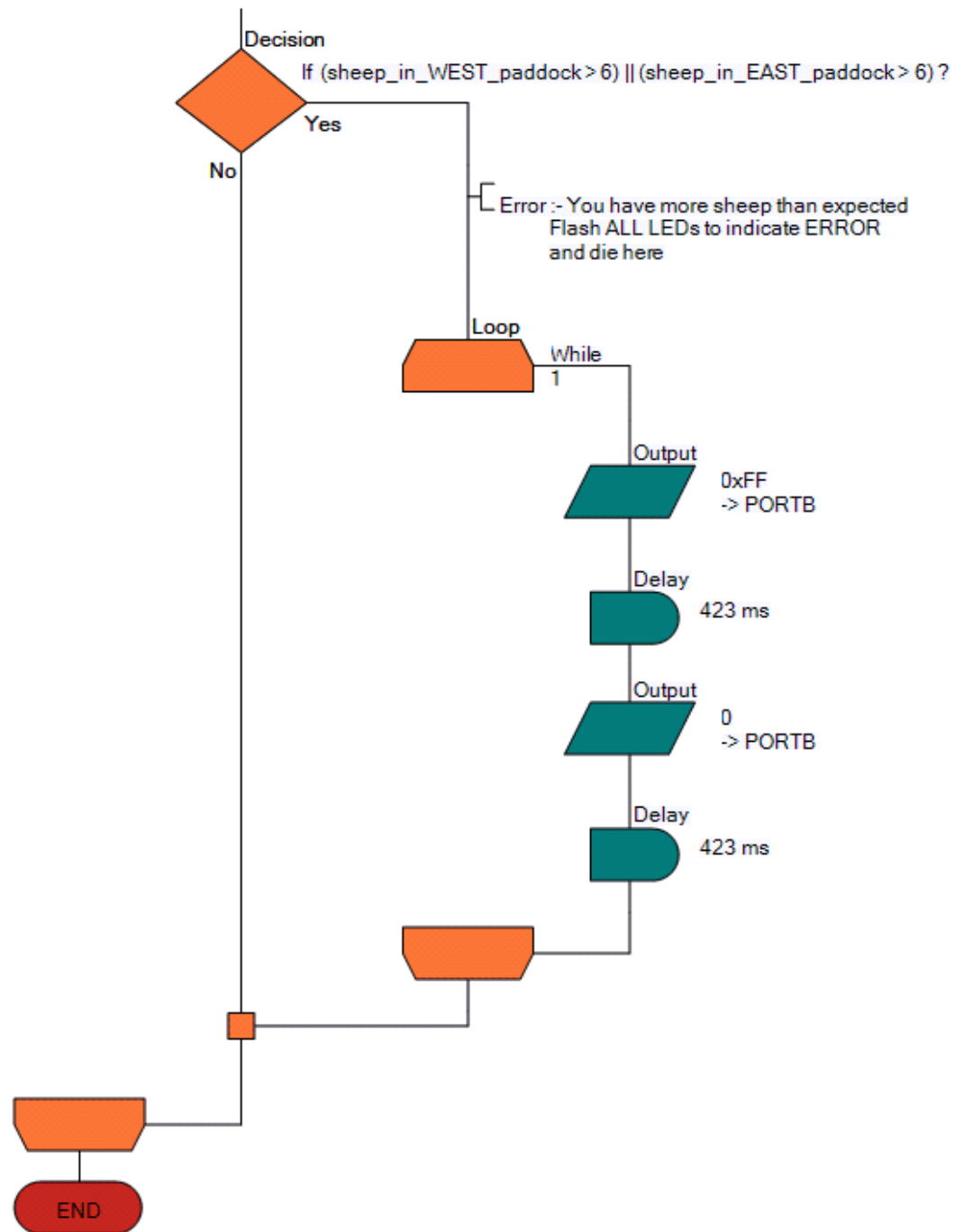
When you can move the sheep correctly from west to east, add the code to handle east to west moves.

What happens when you try to move a sheep from west to east or east to west and there are no sheep to move (i.e. `sheep_in_WEST_paddock = 0` or `sheep_in_EAST_paddock = 0` before you start blocking sensors)?

Add some code to handle the condition described above.

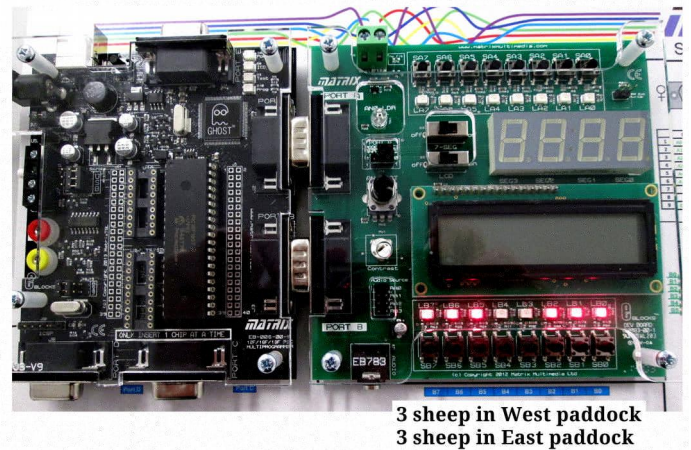
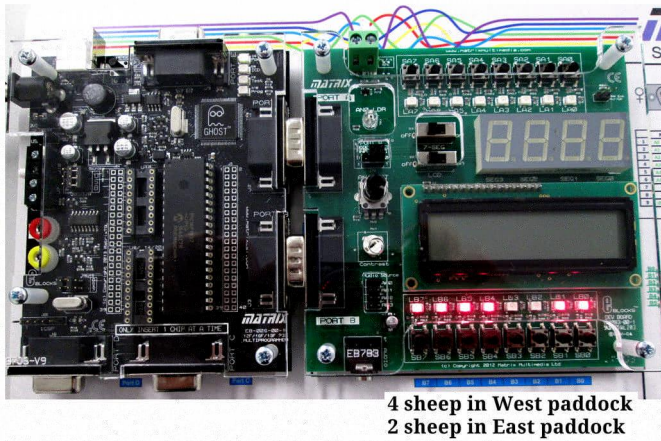
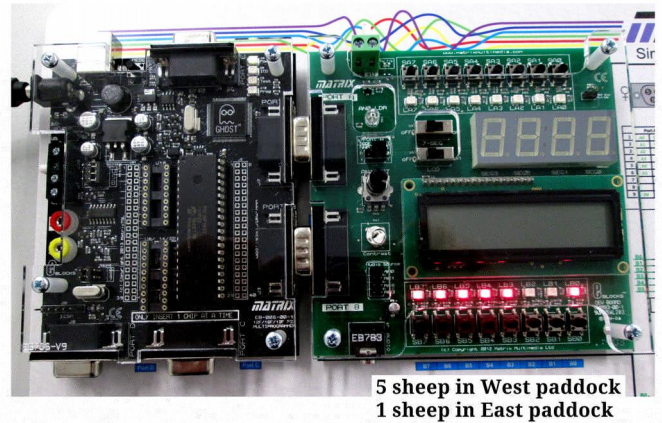
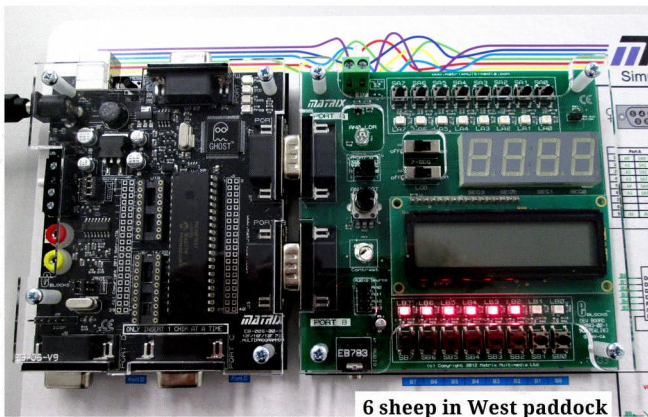


One solution for handling rouge sheep:



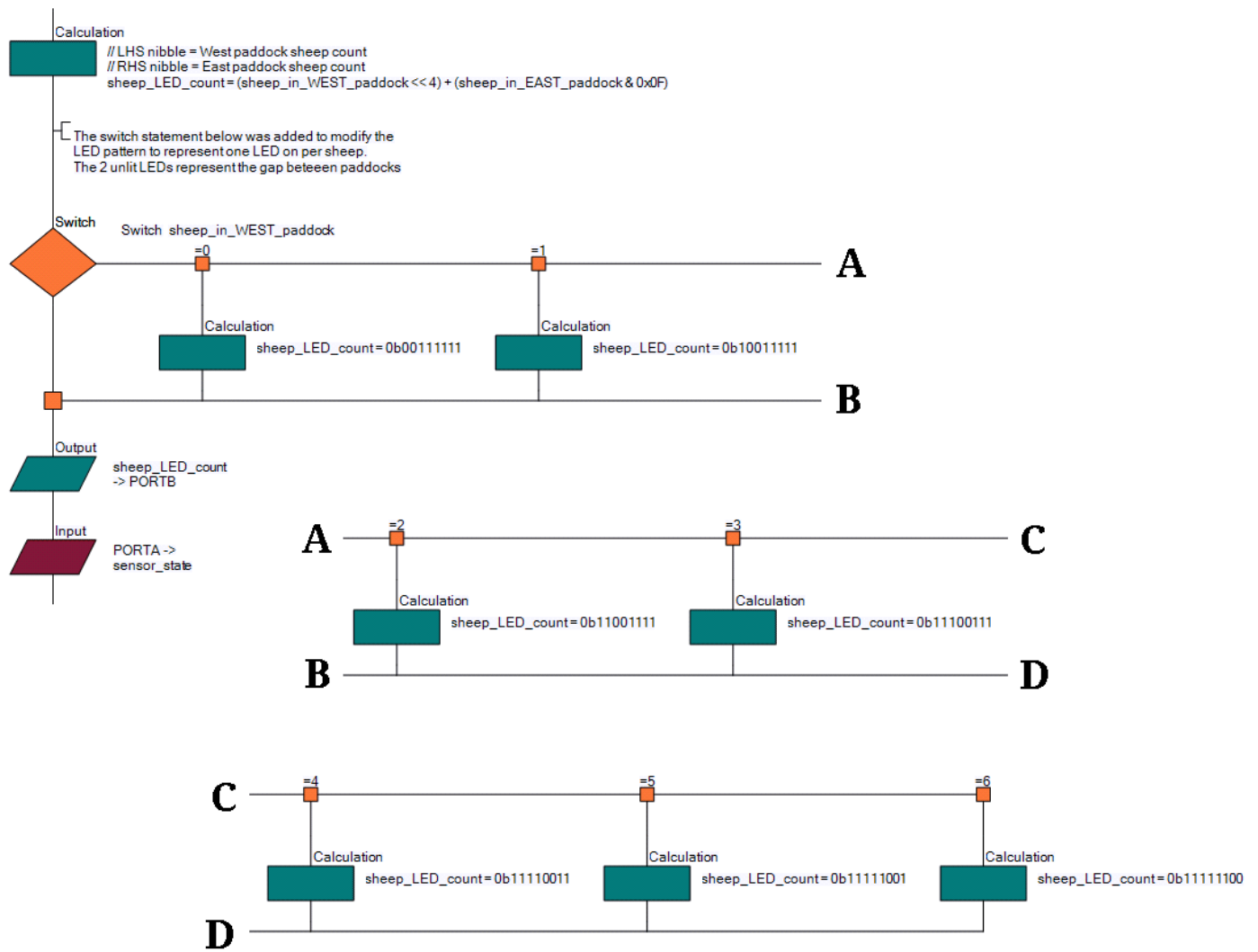
## Advanced Task (once you solve the original problem)

Modify the program to use one LED to represent a sheep and two unlit LEDs to represent the gap between paddocks.



etc

Mapping sheep\_LED\_count from High/Low nibble to one LED per sheep.





## Further Advanced Task (Re-mapping LEDs using 'C' rather than Flowcode)

Alternative mapping for sheep\_LED\_count High/Low nibble to one LED per sheep using 'C' code.

**Note:** Flowcode 7 does not simulate 'C' code, so you will not see the new LED patterns on the simulation, however it does work when downloaded to the microcontroller board.

