Basic System Requirements

- You are to design a controller for a traffic light system.
- The traffic lights are to operate according to the UK standard.

o The traffic light phases are Red, Red-Amber, Green, Amber, Red

- The lights are to control a crossroads which includes a pedestrian crossing.
- There is one major road, one minor road and a pedestrian crossing on all roads.
- The implementation of the lights is a reasonably simple one.

o Both directions of the main road are green together
o Both directions of the side road are green together
o The crossing is on all four roads, and all traffic must be on red when the crossing is green

- Sensors will be used to detect traffic on both directions of the side road and both directions of the main road. Each of these is in the form of an opto-resistor.
- There are four push button switches for the crossing, each with associated LED to represent the ‘wait’ light on a real crossing.

Design Constraints

These are restrictions which have been placed on the project by for practical or operational reasons.

- The system should be implemented in C, using the PIC18F452 development board.
- All development and testing will need to be completed in simulation on MPLAB X, and on your local PIC board. When you come to demonstrate the system, you will need to load your code onto one of the demonstrator boards and run it. You will need to think carefully about how you will simulate your code to ensure that all aspects of the code run correctly.

System Block Diagram

There are a number of different ways in which this system could be implemented but one possible implementation is shown in figure 1 overleaf. At the heart of the system is a finite state machine. This will control the operation of the traffic lights and the progression from one ‘state’ to another. At its most basic, you can think of the states as being the different outputs from the system.

Because the inputs from the road sensors are analogue, you will need to do some input processing on these to convert these analogue signals to digital. Because the state machine will only process based on the inputs *at that time* you may also need to think about whether you need some internal way of remembering the activation of any input sensors until they are responded to. There are four different sensors, one for each road and each of these will need to be processed independently

The pedestrian crossing buttons are wired in parallel on the board, so pressing any of the buttons will trigger the same crossing input. This is an active low input and is connected to RB0 so can be used to trigger an interrupt on the board. Again, you will need to consider if you need to internally ‘remember’ that this has been pressed in order to respond later.

The output of the state machine is the number of that state- although you can also consider the output decoding as part of the design of the state machine- so you will need to determine how you go about decoding the various state numbers into the required outputs in that state.

The internal co-processing block has been included because you may wish to include time delays or other processes which aren’t actually part of the state machine itself, but which will interact with the state machine to affect its progress.

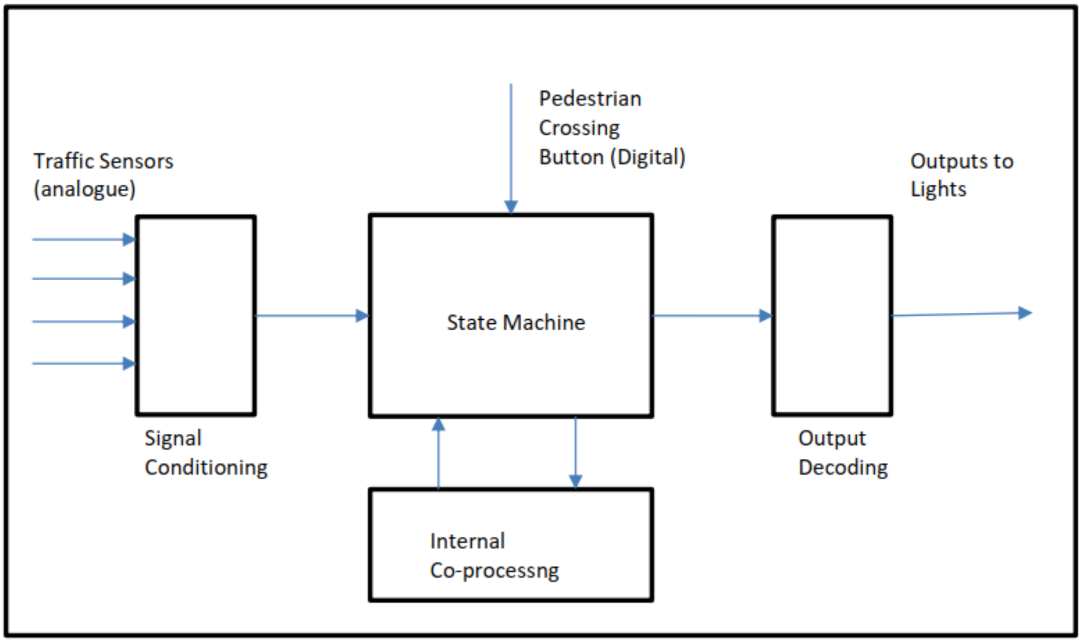


Figure 1: Possible Block Diagram of Traffic Light System

Project Stages

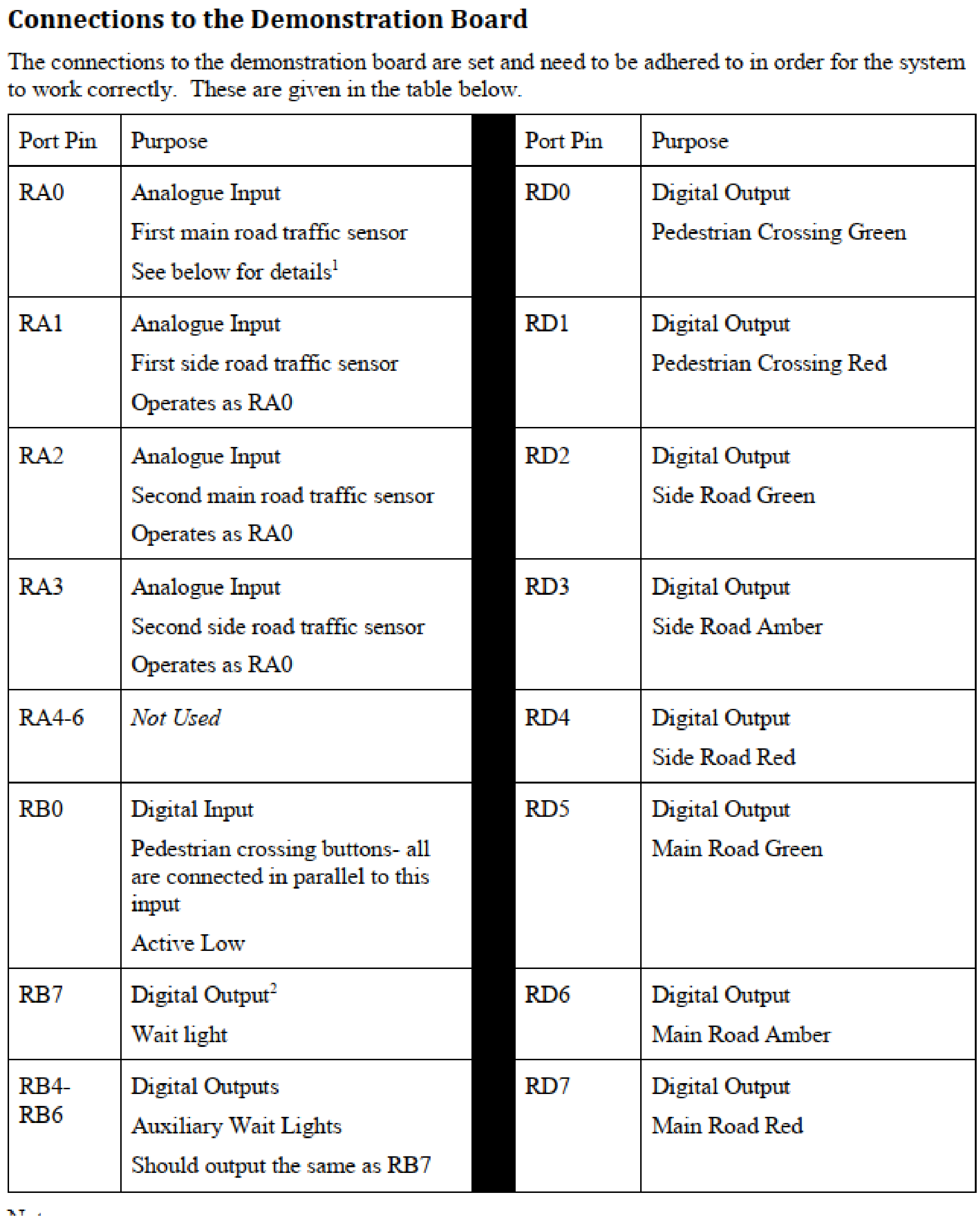
Part 1 ; A functional design of a *finite state machine* to satisfy the requirements above will be produced.

Part 2 ; The state machine should be implemented in C. There should be a clear link between the state machine design and the code developed to implement it.

Part 3 ; The code should then be implemented on the PIC18F452 development boards

Part 4 ; Tests will be undertaken to demonstrate that the system meets the requirements.

Part 5 ; A review will be undertaken to identify any shortcomings or improvements which could be made.



Connections to the Demonstration Board

The connections to the demonstration board are set and need to be adhered to in order for the system to work correctly. These are given in the table below.

Port Pin	Purpose	Port Pin	Purpose
RA0	Analogue Input First main road traffic sensor See below for details[1]	RD0	Digital Output Pedestrian Crossing Green
RA1	Analogue Input First side road traffic sensor Operates as RA0	RD1	Digital Output Pedestrian Crossing Red
RA2	Analogue Input Second main road traffic sensor Operates as RA0	RD2	Digital Output Side Road Green
RA3	Analogue Input Second side road traffic sensor Operates as RA0	RD3	Digital Output Side Road Amber
RA4-6	*Not Used*	RD4	Digital Output Side Road Red
RB0	Digital Input Pedestrian crossing buttons- all are connected in parallel to this input Active Low	RD5	Digital Output Main Road Green
RB7	Digital Output[2] Wait light	RD6	Digital Output Main Road Amber
RB4-RB6	Digital Outputs Auxiliary Wait Lights Should output the same as RB7	RD7	Digital Output Main Road Red

Note:

[1] The traffic sensors are analogue voltage signals which are generated by potential dividers consisting of an opto resistor and a fixed resistor in series. A voltage greater than about 1.4V indicates the presence of a vehicle. A voltage less than about 0.5V indicates the absence of a vehicle.

[2] All Outputs are Active High

Safety

You should not be in a position where traffic and pedestrians have a green light at the same time, nor should both the major and minor roads have a green light at the same time.
To this end, the exact sequence of the lights should be considered, and any potential issues resulting from any changes you might make.

Aspects to be considered

State Machine Design: This aspect will look at the development of the state transition diagram, and the state transition table, is it valid, is it complete etc.
Initial Code Design: This section will look at what evidence is presented, of the design of the code. You should note that design is expected to be completed in advance of any coding being undertaken.
Final Code:

1. Does the code match the design?
2. Is there a clear link to the start machine design?
3. Is the code well structured?
4. Is the code well commented?

Operation: will look to see the system in operation

Results and Conclusions: results to show how the system has been tested and valid conclusions drawn from those results