

MARKING CRITERIA AND STUDENT FEEDBACK

This section details the assessment criteria. The *extent* to which these are demonstrated by you determines your mark. The marks available for each criterion are shown. Lecturers will use the space provided to comment on the achievement of the task(s), including those areas in which you have performed well and areas that would benefit from development/improvement.

Common Assessment Criteria Applied	Marks available	Marks awarded
1. Research-informed Literature Extent of research and/or own reading; selection of credible sources; application of appropriate referencing conventions.	10	5
GOOD SEE GRADING CRITERIA		
2. Knowledge and Understanding of Subject Extent of knowledge and understanding of concepts and underlying principles associated with the discipline.	25	18
EXCELLENT		
3. Analysis Analysis, evaluation and synthesis; logic, argument and judgement; analytical reflection; organisation of ideas and evidence.	25	14
GOOD.		
4. Practical Application and Deployment Deployment of methods, materials, tools and techniques; application of concepts; formulation of innovative and creative solutions to solve problems.	30	26
EXCEPTIONAL.		
5. Skills for Professional Practice Attributes in professional practice: individual and collaborative working; deployment of appropriate media; presentation and organisation.	10	6
VERY GOOD		

Assignment Mark (Assessment marks are subject to ratification at the Exam Board. These comments and marks are to give feedback on module work and are for guidance only until they are confirmed.)

Late Submission Penalties (tick if appropriate)

Up to four weeks late

First Attempt?

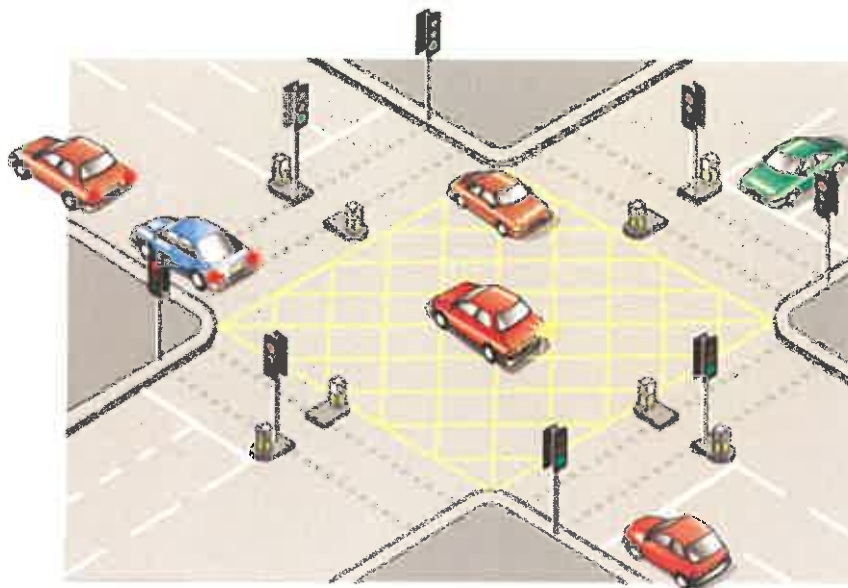


69 %

PROGRAMMABLE LOGIC CONTROLLERS

ASSIGNMENT 1:

PLC PROGRAM DESIGN USING FBD



SEAN BEYNON

CONTENTS

• OVERVIEW	PAGE 2
• PART A	
○ TASK 1 (A)	PAGE 3, 4
○ TASK 1 (B)	PAGE 5, 6
○ TASK 2 (A)	PAGE 7
○ TASK 2 (B)	PAGE 8
○ TASK 3 (A)	PAGE 9,10
○ TASK 3 (B)	PAGE 10
• PART B	
○ TASK 4 (A)	PAGE 11
○ TASK 4 (B)	PAGE 11
○ TASK 4 (C)	PAGE 12
○ TASK 5	PAGE 12, 13
• CONCLUSION	PAGE 14
• BIBLIOGRAPHY	
• APPENDIX 1	
○ FLOW DIAGRAMS FOR THE PROGRAMMING TASKS	

OVERVIEW

The programmable Logic Controllers (PLC) first development stage was recorded as the 1st January 1968 by a young engineer called Dick Morley and a team of engineers from Bedford. From that day the PLC has revolutionised the automation industry with its very compact, rugged and durable design. Before the introduction of PLC's, the only way machinery could be controlled was the use of hundreds or maybe thousands of relays. This process was not only very time consuming but very expensive.

In this assignment research has been carried out on the various types of Programmable Logic Controllers (PLC), explaining their applications, characteristics and method of protecting them from outside influences.

Research was also gathered on Logic gate functions within the PLC program, showing their functionality through the use of truth tables. Other functions researched included timers and counters, which are widely used when designing PLC programs.

As part of the practical aspect of this assignment four programs were designed to demonstrate the understanding of PLC programming and the functions within the Mitsubishi Alpha software, Solving and adhering to all aspects of the questions.

/SDP

As part of **TASK 1(a)**, research was carried out on the 3 types of PLC's currently available to determine the characteristics that define each type and their applications.

The Unitary PLC's are small and compact that includes all basic components in one compact unit. These basic components consist of a power supply, central processing unit (CPU), input module and the output module shown in the example image Fig.1. These Unitary PLC's are fitted directly to the component that it is controlling.

Although they do tend to be the cheapest option of PLC's on the market they do have their limitation. They cannot be extended to (i.e. fixed Input /Output capacities (I/O)), and if a feature of the PLC fails then a complete replacement unit is required.

Unitary PLC's are used for applications that do not require many I/O'S and a limited memory and program size. Due to its very simple and basic development they are ideal for small applications, such as car park barrier or a pedestrian crossing.

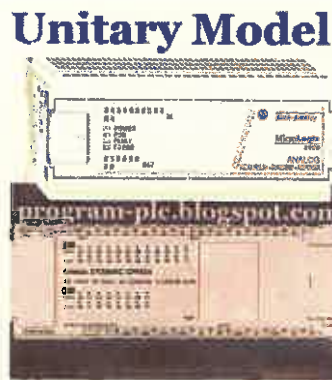


Fig.1 An example of a Unitary PLC model- <http://plcprogram-info.blogspot.co.uk>

The second Programmable logic controller type is a 'Modular PLC (Fig.2). This type PLC can contain several different modules that are coupled together to totally customise a controller. The Modular PLC has a base module that has the same basic components as the unitary PLC, but has the capability to add additional I/O modules or other modules to customise the PLC for the specified design required.

These module configurations can be adapted or changed to cope with changes to the controlled system. Also any feature failures can be exchanged one for one, rather than a new complete PLC set up. Although these are more versatile than the Unitary PLC they tend to be more expensive.

The Modular PLC is widely used in the manufacturing and food processing industries to control the process line. These times of applications require a lot of I/O's to ensure the process runs correctly.



Fig .2 Example of a modular PLC- <http://plcprogram-info.blogspot.co.uk>

The final Programmable Logic Controller is the Rack-Mounted PLC Design that provides very similar characteristics and designs as the modular PLC system, but is implemented slightly differently. Each module within this PLC is totally individual and held in an organised racking system and communicates through the network rack (Fig.3).



Fig.3 Example of Rack-Mounted PLC-<http://www.oocities.org/~buttonbill/m110b.htm>

The concept behind the PLC system is that larger systems can be built without becoming too complicated and cluttered. The modules are generally well organised on the racking system to aid any modifications to the system or replacing features that have failed. Due to the organisation and capability of communicating through networks, this type of PLC potentially has more I/O than any other type of PLC system.

Although the Rack-Mounted PLC system is more versatile, it is by far the most expensive of the three PLC system types.

SDP

TASK 1 (b) requires information about the basic components that form a Programmable Logic Controller, and a method of protecting the system from potential overload or fault conditions.

Typically a PLC system is made up of five basic components, these are illustrated in Fig.4.

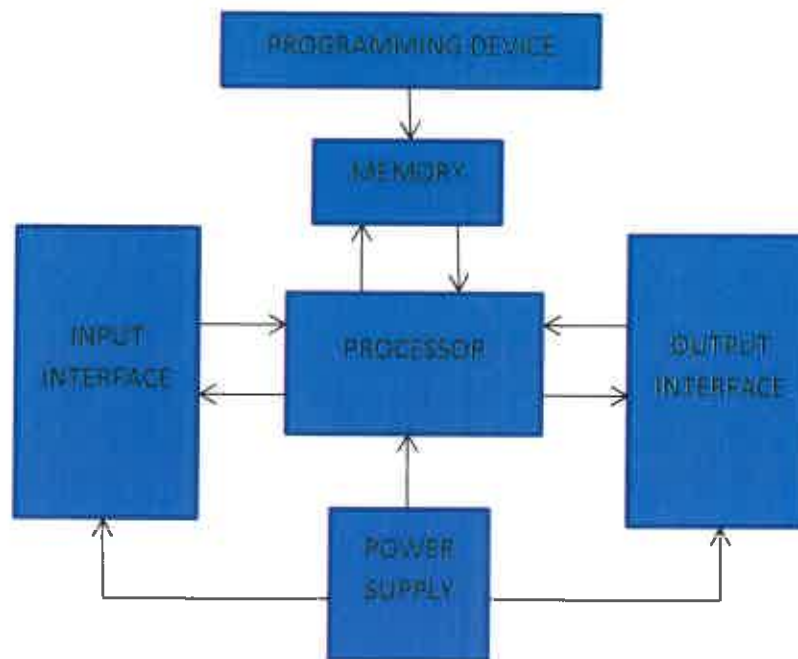


Fig.4 Diagram illustrating a basic PLC system- (W. Bolton)

The programming device is used to develop a program for a specified application. Later in this assignment software called Mitsubishi Alpha will be used to develop programs that will then be transferred to the memory unit of the PLC.

The memory unit stores the program developed and is used for the control function for the microprocessor.

The processor (CPU) unit contains the microprocessor. This interprets the input signals and carries out the control actions that are specified in the developed program and stored in the memory. The processor then provides signals to the actioned outputs.

The input and output interfaces are sections of the PLC that receive and then communicate information to and from the processor. A typical input can be a switch or sensor and an output can be a motor for a specific item of machinery.

Finally a PLC requires a power supply, which is provided by a power supply unit. The unit converts an A.C. mains supply to a D.C. supply necessary for the processor the I/O interface circuits to function.

Those internal units could easily get damaged by surges or under fault conditions, therefore a protection device called an Opto-coupler (also known as opto-isolator or photo-coupler) is used to protect under these conditions.

This is an electronic component that interconnects two separate electrical circuits by means of a light sensitive optical interface. As seen in Fig.5, the design consists of an LED and a semi-conductive phot sensitive device that emits the infra-red beam that is produced for the LED. Due to the two separate electrical circuits the I/O interfaces cannot be directly affected by any potential overload situation or under fault conditions.

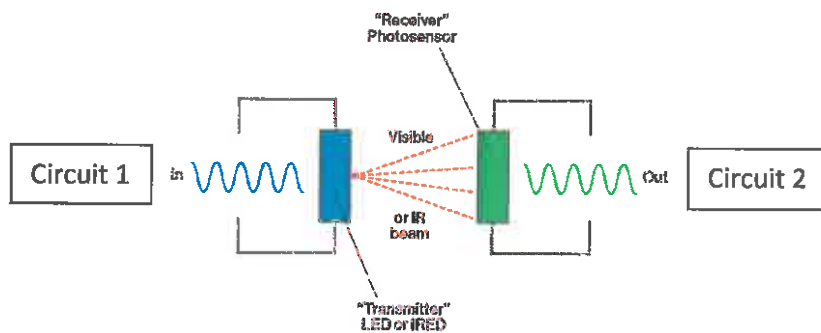


Fig.5- basic construction of an opto-coupler

TASK 2 (a) requires truth tables to be produced for logic gate functions assuming there are two inputs (A&B) and one output (Q).

- AND

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

As seen in the truth table, output Q can only be achieved when both A and B inputs are on.

- OR

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

The OR logic gate produces an output (Q) when A, B or BOTH input are on.

- NAND

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

The only time a NAND gate doesn't produce an output is when both A and B inputs are on.

- NOR

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

The NOR gate is opposite to the NAND and only produces an output when both inputs are off.

- XOR

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

The XOR gate only provides an output when either A or B inputs are on.

In TASK 4 (b) a truth diagram was produced for logic gate circuit in Fig.6.

Should start from the right hand side.

A	B	C	Q
0	0	0	0
1	0	0	0
0	1	0	0
0	0	1	0
1	1	0	1
1	0	1	0
0	1	1	1
1	1	1	1

The Logic gate circuit provided in Fig.6 is over complicated, and does not required the OR and AND gates highlighted within the red box. There are only three possible combinations of inputs to achieve the output Q.

These combinations are:

1. A & B
2. B & C
3. A, B & C

This concept is illustrated in the truth table above.

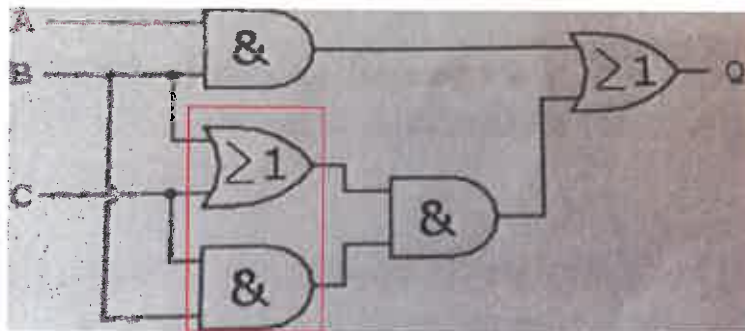


Fig.6- Logic gate circuit

There could be issues when using the current configuration of the logic gate circuit in Fig.6, any one of the inputs could be lost to the PLC and the output Q would still be achieved. This could cause problems with process sequence in which the program is designed for.

When designing control system, many of the tasks require a way of controlling time. Therefore PLC'S have timers built into the system that provide an instruction to wait a set amount of time before doing something (control time). This time duration is pre-set in multiples of the specified time base, either seconds or fractions of a second, using the internal CPU clock.

For **TASK 3 (a)**, explanations have been given for the operation of an ON delay timer, OFF delay timer and a pulse timer. A timing diagram for each timer will aid with the explanation.

The Off-Delay timer is the opposite of the on-delay timer listed above. The Timer Off Delay instruction begins to count a set time when the input on the timer goes off. If the input remains off and the timer has reached its set time interval then the timer output goes off. The accumulated value is reset when the input goes on regardless of whether the timer has timed out. Fig.7 provides a visual illustration to aid this explanation.

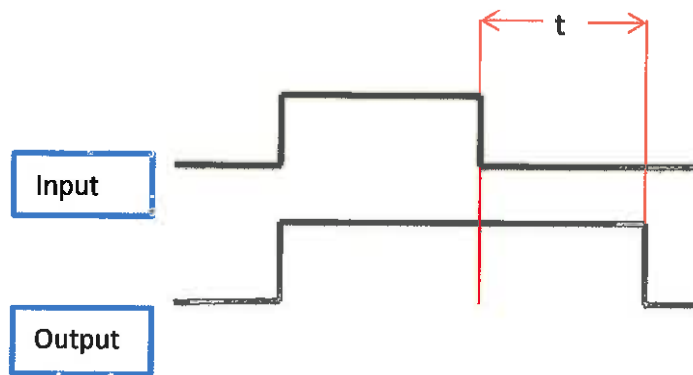


Fig. 7 timing diagram for an OFF delay timer

An On-Delay timer basically delays turning a specific application on. The Timer On Delay instruction begins to count a set time when the input on the timer is on. If the input remains on and the timer has reached its set time interval then the timer produces an output. The accumulated value is reset when the input goes off regardless of whether the timer has timed out. Fig.8 provides a visual illustration to aid this explanation. These types of timers are in most PLC's, and therefore most commonly used timer function.

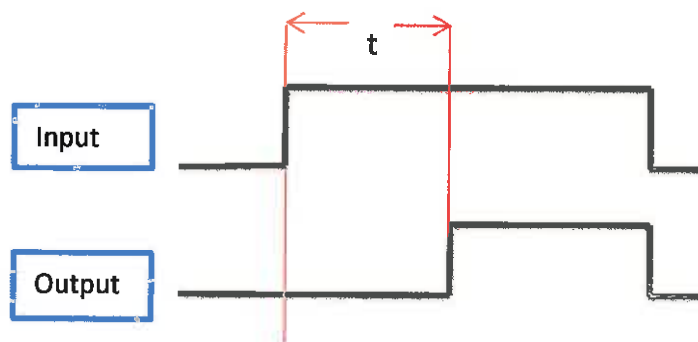


Fig. 8 timing diagram for an ON delay timer

The pulse timer in Fig.9 switches the output off for a fixed period of time. Once that set time interval has been reached the output returns to the on position. This sequence is continued as long as there is an input maintained onto the pulse timer. Fig.9 shows an illustration of a timing diagram for the timer switching off for a fixed period of time (t). A pulse timer can also be used in the opposite sequence. Where the normally open contact, when there is an input on the timer, closes for a timed interval and then returns to the open position when time has lapsed. See Fig. 10 timing diagram for a pulse timer

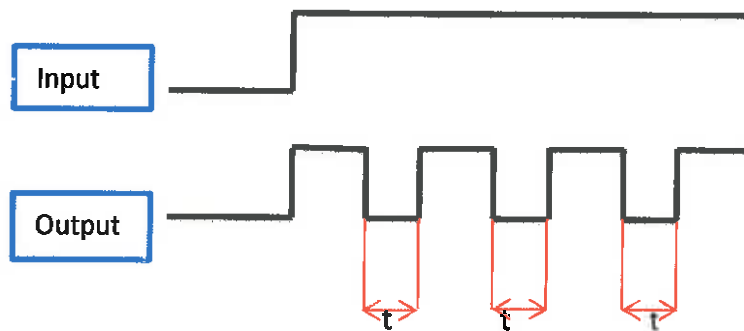


Fig.9 Timing diagram for a pulse timer

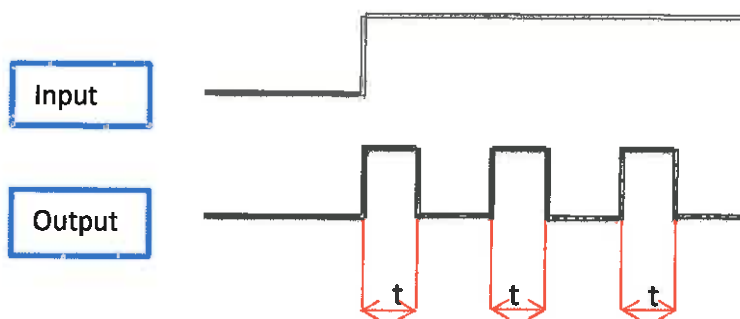


Fig.10 Timing diagram for a pulse timer

TASK 3 (b)

A counter is pre-set to a specified number and when an input pulse is received it will operate its contact. Every time the counter receives an input pulse, which could be from a type of sensor, the number increases or decreases until its combined value meets the pre-set value or zero.

There are two types of counter, down counters and up counters. Down counters count from a pre-set value to zero, and up counters count from zero to a pre-set value. These counters can be used individually or instantaneously.

PART B is the practical side of the assignment, where Mitsubishi Alpha Software was used to develop PLC programs to perform specified tasks. When completed these programs needed to be demonstrate to the assessor.

TASK 4 (a) was to develop a program for a washing machine cycle. In appendix 1 a flow chart was designed to demonstrating the thought processes and capturing those processes ready to develop the program. As you can see from Fig. 11 the input was from a momentary on switch, which then started the washing cycle with the water supply pump, heater, then drain pump, each part of the cycle having an 'ON' indication light. This program was developed using 'on delay timers' and 'set re-set' functions to set and re-set the outputs in the specified time needed to complete this task.

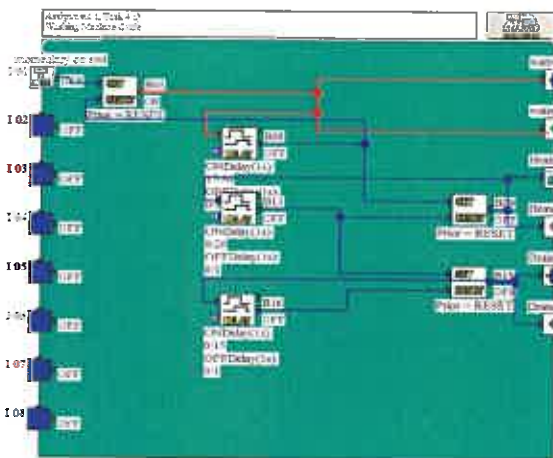


FIG. 11 washing machine program

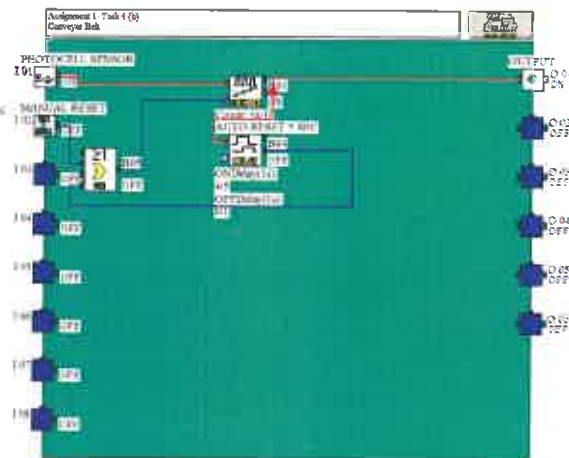


Fig. 12 conveyor belt program

TASK4 (b) was to develop a program for a conveyor belt system that will produce an output after receiving a 10 pulse input from a photocell sensor. The system also needed to have a manual and automatic counter reset. Using a flow diagram, appendix 1 task 4(b), the design process was considered and then the PLC program could be developed. The output was achieved by using an 'up counter' with a pre-set value of 10 pulses. Once the 10 pulses had been reached the counter produced an output. An 'ON delay timer' was used to achieve the automatic reset but delayed it for 5 seconds. Fig.12 shows the design of the conveyor belt program.

TASK 4 (c) was to develop a detection system to count people entering and leaving a store, displaying a warning if the number of people in the store reached 50. This program also needed a manual reset to clear stored value. The task was achieved by using an 'up/ down counter', which could account for people entering and leaving the store simultaneously. A pre-set value of 50 was entered into the counter to indicate a warning when the value had been reached. A flow diagram was produced when first developing the program (appendix 1 task 4 (c)).

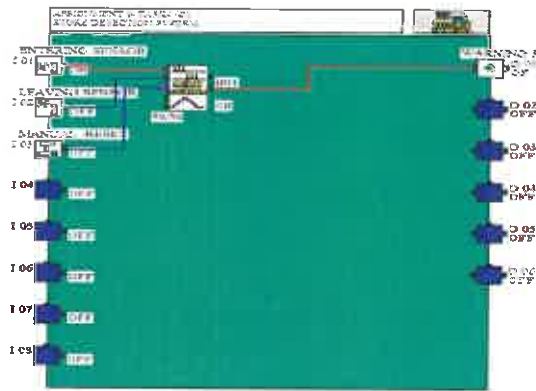


Fig. 13 store detection system

TASK 5

This task required the development of a program that will operate a set of 2-way traffic lights using the timed sequence in table 1.

Table 1

Timing (s)	5	10	15	20	25	30	35	40	45	50	55	60
North / South Traffic Light Set Sequence												
Set 1	Red											
	Amber											
	Green											
East / West Traffic Light Set Sequence												
Set 2	Red											
	Amber											
	Green											

The program is initiated by a momentary push button switch with a one shot function. The program will then continually carry on through the sequence automatically. The sequence and timing of the traffic lights are controlled by the use of 'ON delay timers' and 'set/ reset function'. 'OR' gates were used to allow the traffic lights to continually cycle automatically and also to allow individual lights to be used twice at different stages of the sequence.

Fig.14 shows the program developed to achieve specified criteria. The program was witnessed by the assessor. The flow diagram for the initial thought process and design is shown in Appendix 1. Task 5.

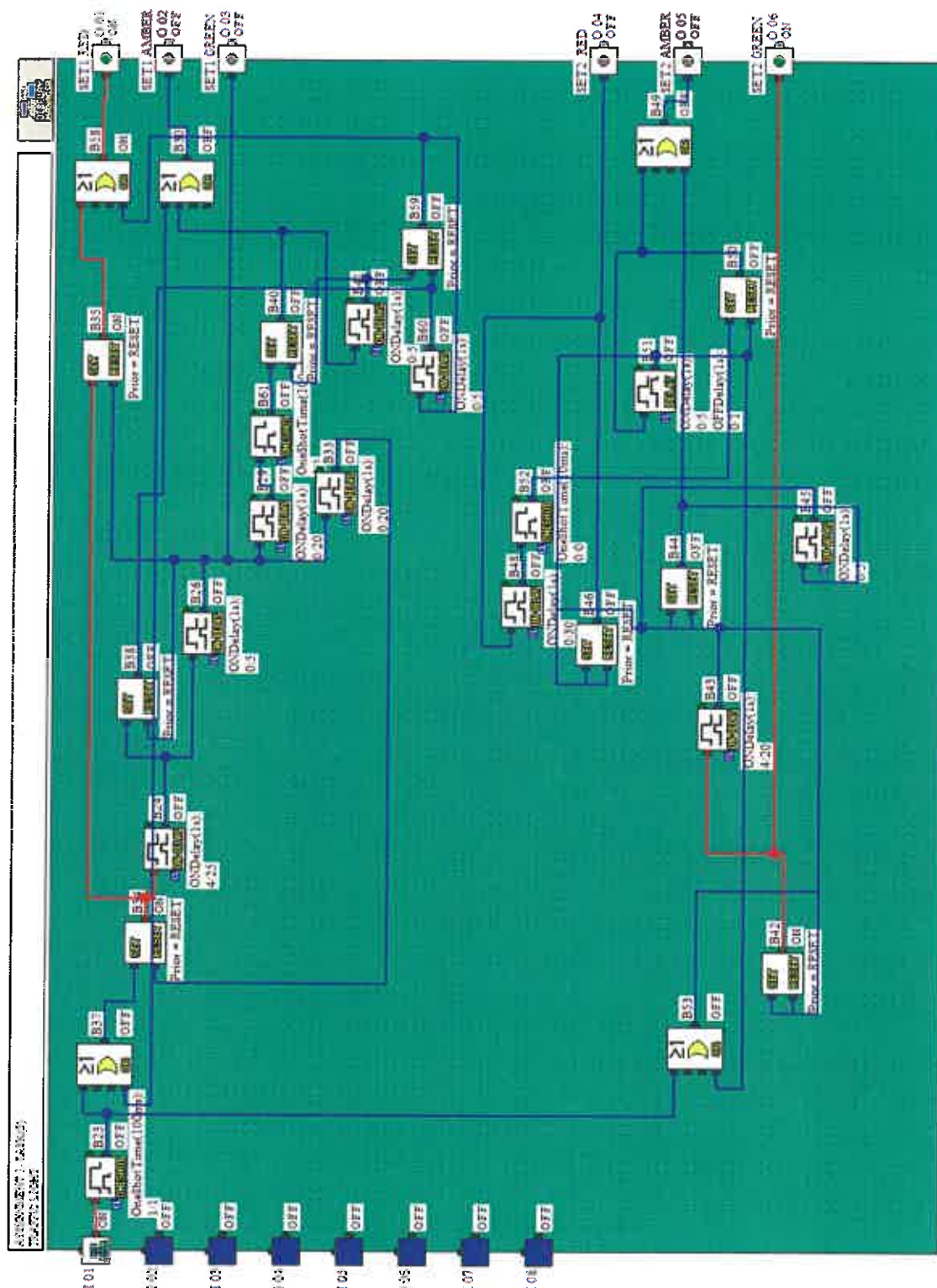


Fig. 14 2-way traffic light system

CONCLUSION

This assignment has increased knowledge of the different types of Programmable Logic Controllers, understanding where each is applied, their specification and functionality.

The production of truth tables helped get the understanding of each Logic gate. They were then utilized in the programming problem tasks.

The research of timers and counters enabled me to understand their application and include them in the programming problems.

Between carrying out research and producing flow diagrams, the practical programs could be developed using the Mitsubishi Alpha software. All aspects of the tasks were carried out and successfully completed, although I would consider changes to the traffic light task. The program designed has SET 1 AND SET 2 working as individual systems after the initial push start has been activated. The design would be better for safety purposes if the SETS were interlinked to ensure that green could not be achieved on each set.

BIBLIOGRAPHY

Books

W. Bolton. (2000). Programmable Logic Controllers, second edition

Khaled Kamel, Eman Kamel. (2013). Programmable Logic Controllers: Industrial Control,

Electronic

<http://plcprogram-info.blogspot.co.uk>

<http://www.oocities.org>

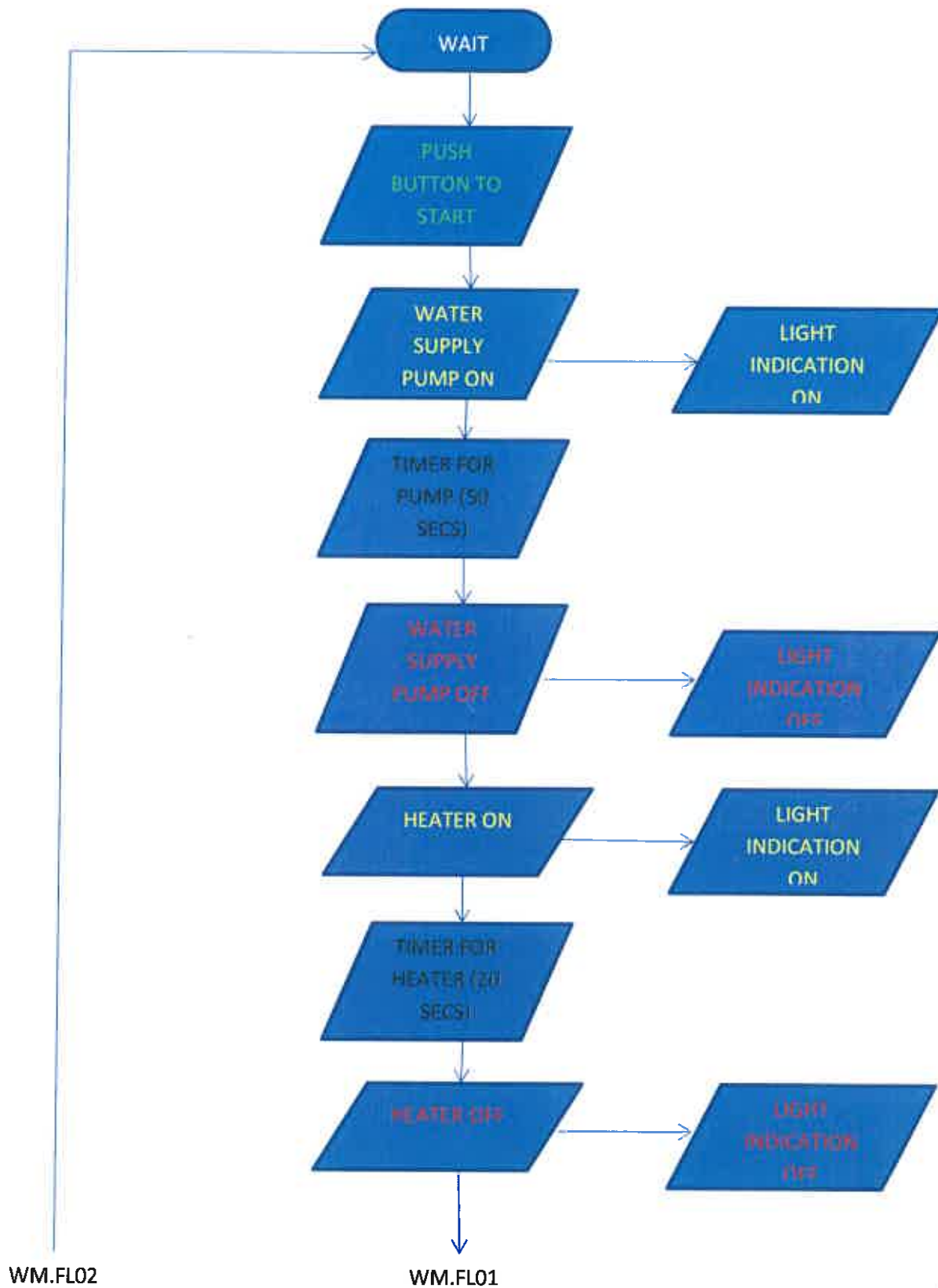
<http://www.plcmanual.com/plc-instructions-ii-timers>

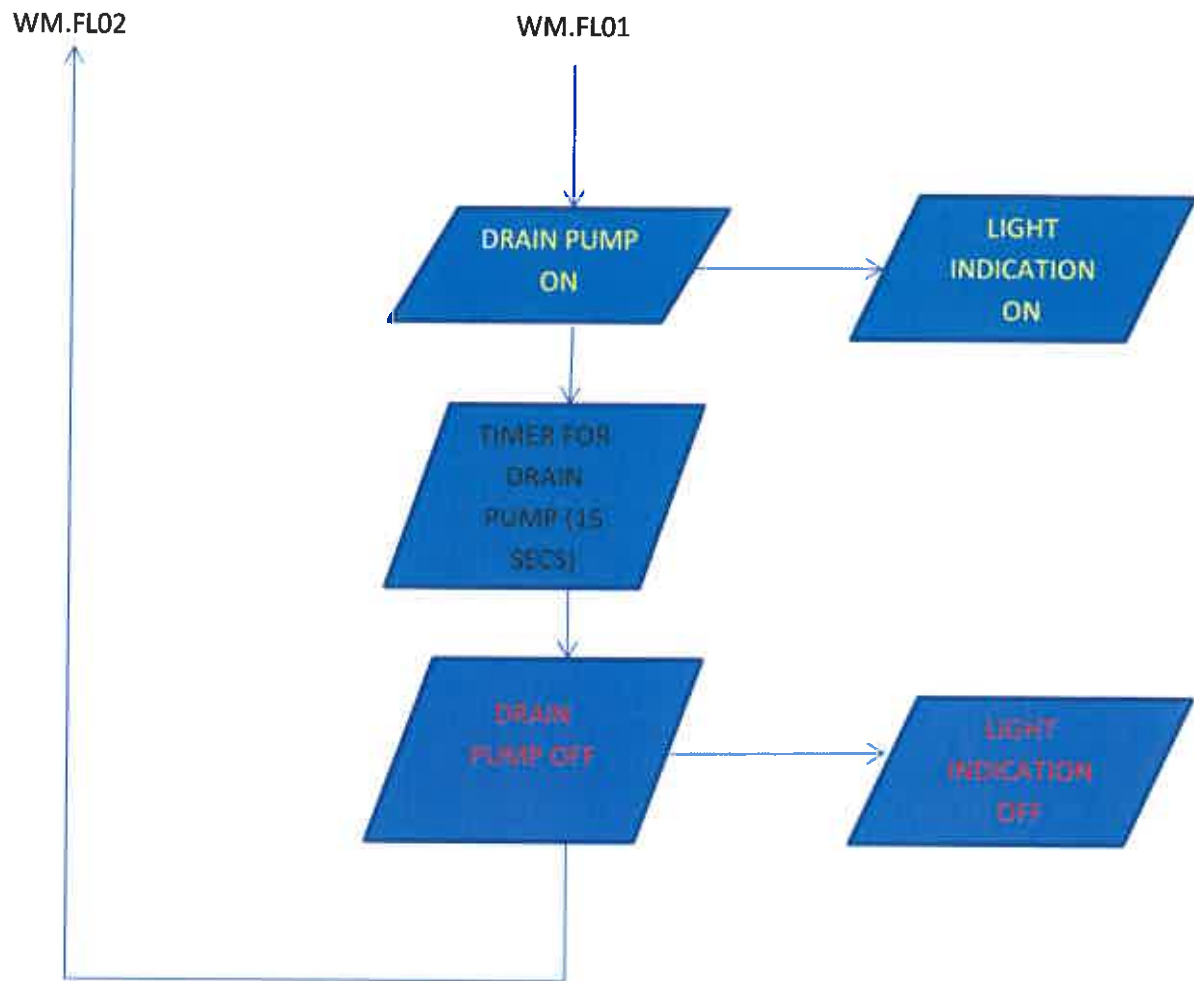
/SDP

Appendix 1

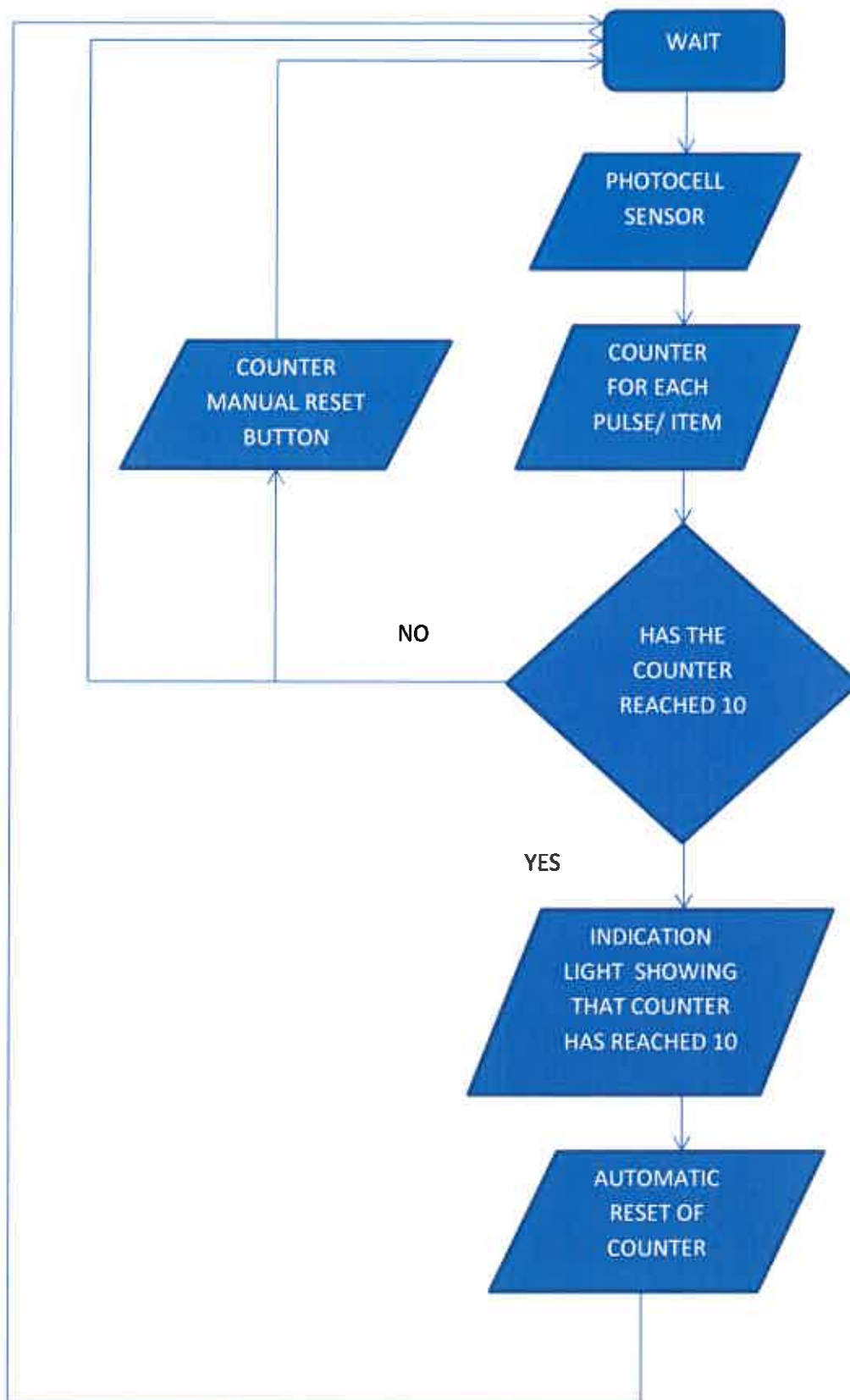
Appendix 1

Task 4.(a) Flow Diagram for the washing machine cycle.

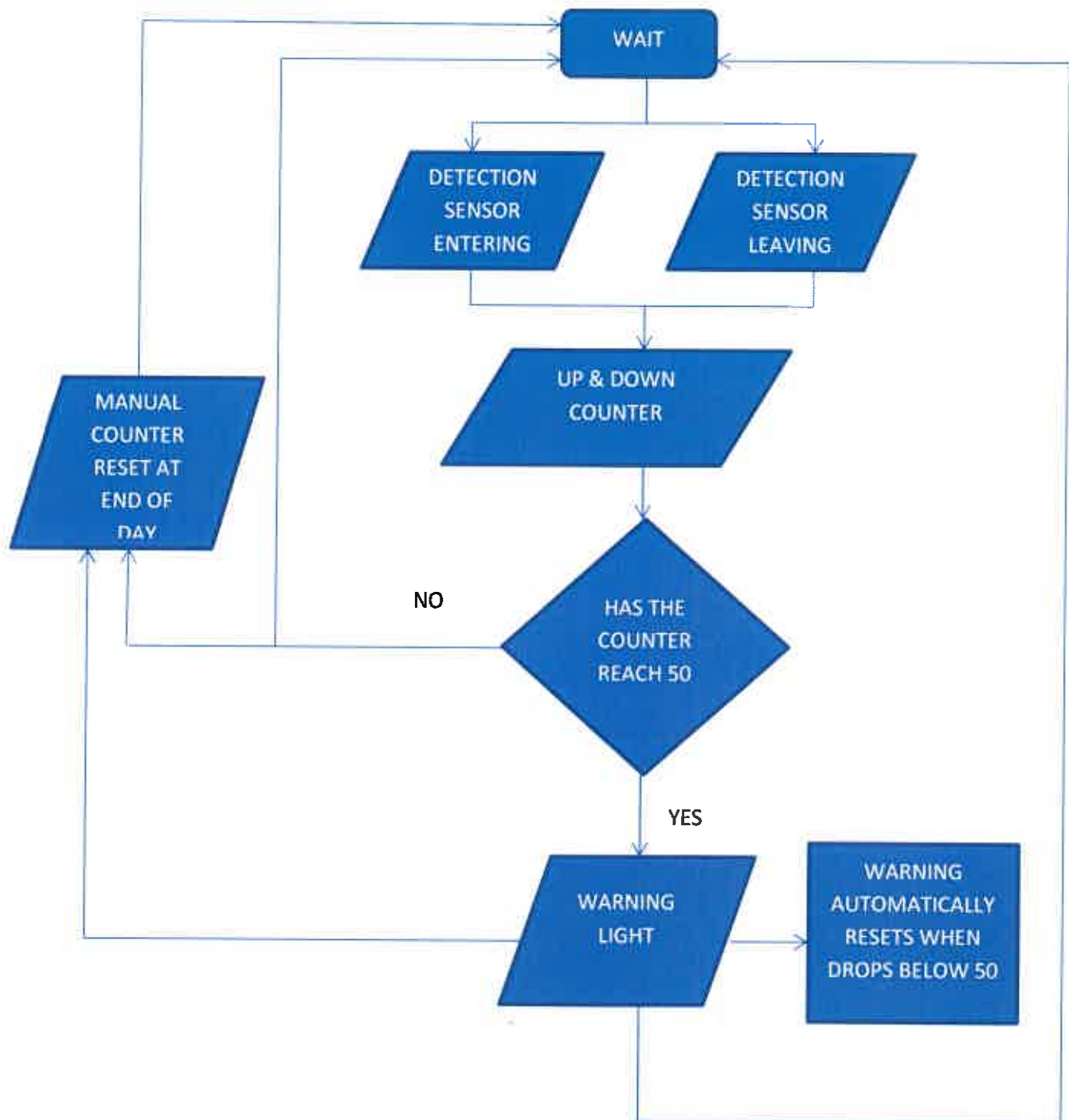




Task 4. (b) Flow diagram for the conveyor belt.



Task 4. (c) Flow diagram for the store detection system.



Task 5.Flow Diagram for
Traffic Light System