CONTENTS

Design, Memory and I/O maps are included in separate file for better presentation, details are included within this report.

ACKNOWLEDGEMENT

We want to express sincere gratitude to Dr. K. R. Anupama, for giving us this opportunity to work on the real-world application of microprocessors. During every stage of the project, our ability to apply concepts learned was significantly challenged, which eventually gave us a better insight into interfacing hardware with microprocessors.

We are also grateful to other instructors in this course for guiding us during the project. Also, we are grateful to Dr. K. R. Anupama for her documentation and learning resources and constant guidance, without which we could not have learned and applied concepts of microprocessors and interfacing in our project.

Finally, we extend our thanks to friends and other batchmates for continuous support and motivation.

PROBLEM STATEMENT

Smart garage system with given capacity, equipped with a system to automatically detect car entry or exit and provide necessary data for smooth functioning of garage.

SYSTEM REQUIREMENT

- The capacity of the garage is 2000 cars.
- System is used in underground parking lot of a hotel.
- Each user of the garage has a remote unit which he can use for opening and closing the garage door.
- Remote unit has only a single button.
- User is allowed to retrieve the car at any point of time
- A LCD Display is available indicating the number of cars in the garage.
- System runs from a standard power inlet available in the garage.
- When the number of cars reaches 2000, the LCD displays ”FULL”
- When there are no cars, the LCD displays “EMPTY”

System Specifications:

- Remote unit button toggles the condition of the garage door- i.e. if the door is opened it is closed and vice versa.
- The remote unit is used for short distances only.
- A DC motor is used for opening and closing the door – The motor is a 50V -3 A motor.
- Maximum frequency input to the motor system cannot exceed 100 KHz.
- The system should be able to distinguish between a person and a car.
- A switch is available that can be closed only by the weight of a car.
- System is used in the hotel- so you can assume that a valet parking system is followed – this indicates that only one person leaves the garage after the car is parked and only a single person enters the garage to retrieve the car
- The system also has to distinguish between entry and exit. You have to develop a scheme to distinguish between entry and exit of person/car. [Hint: Use any number of IR sensor pairs as required]
- Whether a car enters or a valet enters the door remains open for a period of five minutes.
- The door can close after 5 Minutes or when the valet uses the remote.
- The remote can be used inside as well as outside the garage.

SYSTEM DESIGN

The Smart Garage System comprises three central units – the garage door-remote, IR sensors, and switch(closed by the car's weight).
As depicted in Fig 19.1, the garage door unit will have one separate IR sensor. Apart from this, we have two IR sensors working independently of the garage door with a weight sensor between them.

We have a switch that can only be closed by car’s weight. To get an entry/exit scheme, we have placed two IR sensors. One additional IR sensor is used at the garage door. The detailed working of the system is explained below.

The remote would open the gate. When the gate is open, the car will move towards the sensor assembly outside the garage door. Sensor 1 will detect first, then the weight sensor, then Sensor 2. As soon as the weight switch is closed, we know it is a car and update the count. A similar set of events will happen when the car comes out, but now the inner IR will be triggered before the outer one. For a valet weight switch will remain open, and count will not be updated.

We have a motor for opening and closing the gate and a motor driver IC connected to the processor for controlling the motor. It is mentioned in user requirements that the garage door must remain open for 5 mins if a valet or a car crosses. So, we have one IR sensor at the door, which, when triggered, 8086 will reset 5 min timer. The timer will start running once the garage door is completely opened.

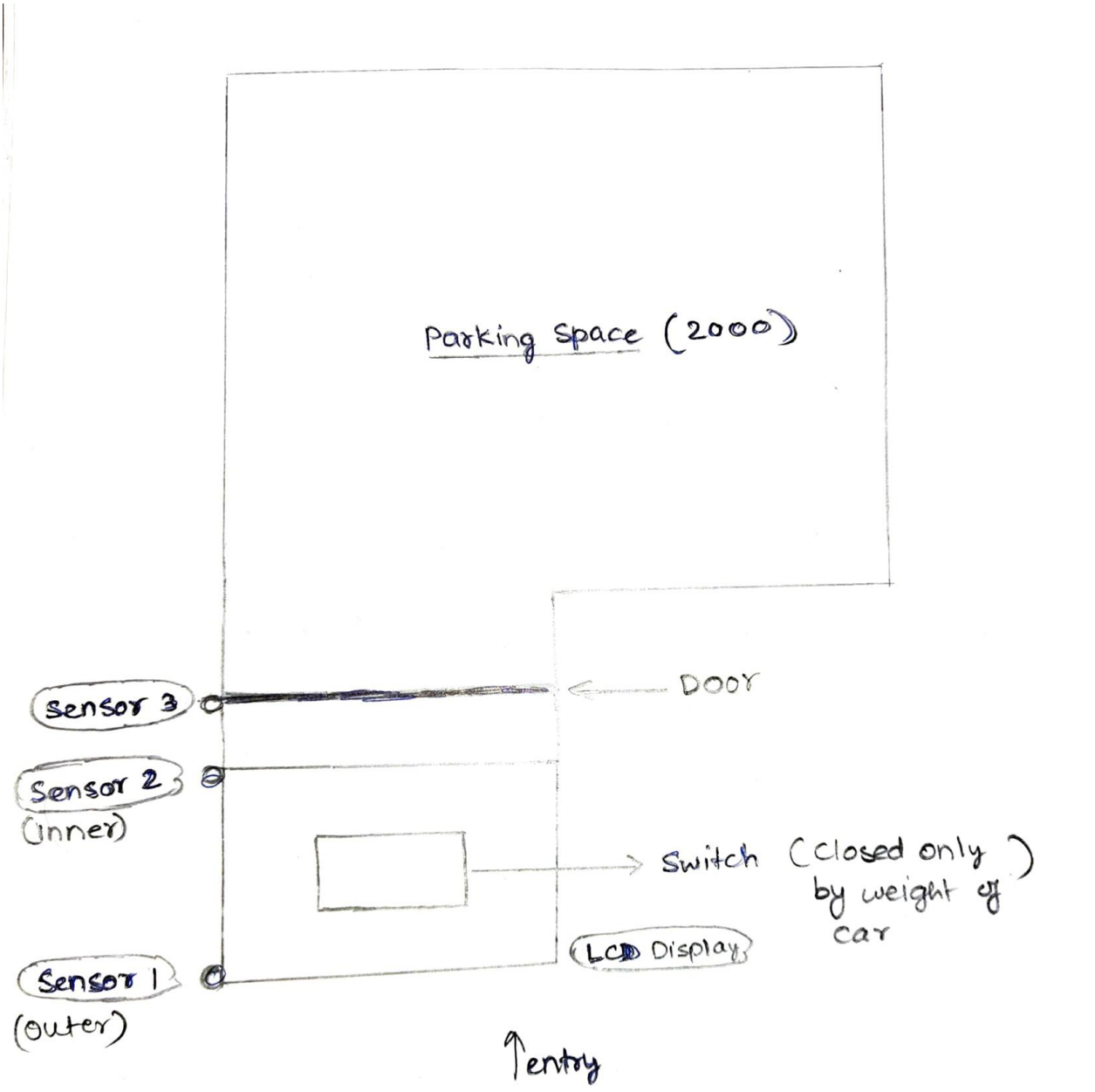
Parking Space (2000)
Sensor 3
DOOR
Sensor 2
(Inner)
Switch (Closed only by weight of car)
LCD Display
Sensor 1
(Outer)
entry

Fig 19.1

ASSUMPTIONS

1. Gate is assumed to be closed at the beginning.
2. Only one car can either enter or exit the gate at a particular time.
3. The car does not enter sensor 1 unless the gate is completely opened.
4. The car continues to move in one direction at every point.
5. Clockwise and anti-clockwise rotation of motor closes and opens the gate respectively. The time taken by motor to operate the gate doesn't effects working of system in any way.
6. Valet do not try to enter with a car if the parking is full, for this purpose LCD Display is placed outside the garage.
7. Clockwise rotation will open the door, and anticlockwise will close it. The Motor will take 10 sec to completely open a closed gate and vice versa.
8. When weight switch is closed we have 0 as input to circuit and vice versa.
9. User will use remote outside the sensor assembly, ie; sensor 1 and 2 are placed close to garage door.
10. For explanation purpose sensors are kept, outside the gate but we can keep them inside too, as long as none of the assumptions are violated.
11. No 2 remotes are pressed simultaneously.

HARDWARE USED

S. No.	Component	Quantity	Purpose
1.	8086 Microprocessor	1	Central Processor
2.	8255A Programmable Peripheral Interface	1	Interface to Programmable parallel I/O
3.	8253 Programmable Interval Timer	1	Clock Generator
4.	8259A	1	Programmable Interrupt Controller
5.	6116 (2KB RAM)	2	RAM
6.	2716 (2KB ROM)	4	EPROM
7.	74LS138(3:8 Line Decoder)	2	Address Decoder
8.	74LS373 (Latch)	3	Latching the bus
9.	74LS245 (Buffer)	2	BI-Directional Buffer
10.	74LS244 (Buffer)	1	Buffer(Control signals)
11.	IE-0530HP	3	IR Emitters
12.	PIC-1018SCL	3	IR Receiver (Sensing distance upto 15 meters)
13.	LM016L	1	LCD Display
14.	TB67H451FNG	1	Motor Driver
15.	Weight Switch	1	Differentiate between car and valet
16.	SN74LVC1G08-Q1 (Single 2-ip AND gates)	3	Interfacing

Table 19.1

WORKING OF THE SYSTEM

Generation of Clock

1. To generate interrupt after 10 sec

2.5MHZ is input to 8253. We need two counters in Mode 2, with count values 27500 and 1000 respectively. For 5 min timer we will count 30 interrupts of this signal. For closing and opening of door, 1 interrupt would suffice, as it was assumed, door toggles its states completely in 10 sec.

Calculations:

At every 10 sec, we need an interrupt. So, when output goes low we use it as interrupt, let output generated in mode 2 has 90% duty cycle,

*So time period = 10 + (0.10 * 10) sec*
= 11 sec

Required frequency = 1/11 Hz

*So count = 2.5 * (10 ^ 6) * 11 = 275 * (10^5)*

Max count to a counter = 65553d = FFFFh

From this appropriately selecting count value, we choose 27500,1000 as our count values to feed in counters respectively.

2. To generate square wave of frequency less than 100KHz

Using the third counter in 8253 in mode 3 with count value of 30.

Calculations:

1 counter in Mode 3,

Max frequency is 100KHz, so

*Min count = (2.5 * (10 ^ 6)) / (100 * (10^3)) = 25*

Arbitrarily selecting count = 30

*Frequency generated = (2.5 * (10^6))/30 ≈ 83.33 KHz*

Working of motor and motor driver

A 50 V, 3A DC motor driven by a motor driver (Toshiba TB67H451FNG rated at 50V and max 3.5A, PWM DC motor driver) is used for opening or closing the gate. The motor gets switched on either on pressing the button on remote or automatically when 5 minutes are elapsed after a car or valet has passed after opening it from remote.
In motor driver we have, have Vref = 4V, logical input 1(IN1), logical input 2(IN2). IN1 and IN2 will be controlled to control the direction of rotation of motor, depending on following table. For controlling input frequency to be less than or equal to 100KHz, we will combine inputs IN1 and IN2 with a signal(frequency < 100 KHz) generated as shown above.

In selected driver L = 0 V (min) to 0.8 V (max)
H = 2.0 V (min) to 5.5 V (max)

If IN1 = IN2 = L for a 0.7ms or less then we are in stop mode.
If IN1 = IN2 = L for a 1.5ms or more then we are in standby mode.

IN1	**IN2**	**OUT1**	**OUT2**	**MODE**
L	L	OFF (HI-Z)	OFF (HI-Z)	Stop / (standby after 1ms)
H	L	H	L	Forward
L	H	L	H	Reverse
H	H	L	L	Brake

Let,
Forward => Clockwise rotation of motor, ie; open a close gate
Reverse => Anti-Clockwise rotation of motor, ie; close an open gate

When Motor is not used we will send LOW input and motor driver will be in standby mode, with no current flowing in the DC motor from OUT1 and OUT2.
[According to the datasheet, there is 30 μs delay for motor to come out of standby in operation mode.]

Working of LCD

LCD Display(Hitachi's LM016L, upto 80 characters) is used to display the number of cars present in the garage at every instant. The count is provided by 8255 PIC chip whenever a car either enters or exits. It will display number of cars in the garage at any time. When 0 cars then it will show, 'EMPTY' and when 2000 cars it will show 'FULL'

HANDLING OF THE SENSORS OUTPUT

There are 3 sensors used in the system as show in Fig 19.1.
IR emitter emits infrared ray detected by the receiver. If an object is crossing it then it blocks path of light to mark the receiver output changes.

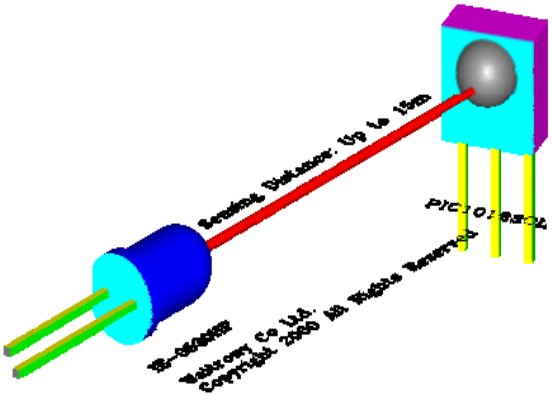

Fig 19.2: *Waitrony Co Ltd.*
Copyright 2000 All Rights Reserved

OUT = { (≥4.5V) ON, no object crossed receiver is getting emitter's IR rays
(≤0.4V) OFF, path of rays blocked by and object.

Where OUT is named as sensor output.
As mentioned in figure, maximum range for proper detection is 15m.

[Note: Selected sensors output is Active Low]

GENERATION OF INTERRUPTS THROUGH IR SENSORS

Corresponding outputs of Sensor 1, 2 and 3 are used as OUT1, OUT2, OUT3 respectively. And output from weight switch be W in active low, ie; when it is closed we have (W = 0).

	OUT1	W	OUT2
➔ State 1:	1	1	1
➔ State e2:	0	1	1

State e2 is next of state 1 on car entry.

➔ State ex2:	1	1	0

State ex2 is next of state 1 on car exit.

Entry scheme

- ➔ When car enter from State 1 we move to State e2, we provide this as interrupt to system and system branches in ISR_entry.
- ➔ Then we use polling for waiting till W = 0 and then we update the count.
- ➔ Then we wait for **car to reach sensor 2** (OUT2 == 0) and after that we, wait for **car to completely cross sensor 2**(OUT2 == 1) and, we exit the ISR_entry.

In case of valet is going through the sensor unit then things will be as

- ➔ After branching to ISR_entry in State e2, we branch to ISR_entry where W will remain 1, so along with that we keep on checking for (OUT2 == 0), as depicted in Fig 19.3. After that we wait again for (OUT2 == 1) and, we exit ISR.

Exit Scheme

For car exit

- ➔ From State 1 after interrupt if we reach State ex2 then we branch to ISR_exit.
- ➔ We wait for W == 0, and update the count.
- ➔ Then we wait for **car to reach sensor 1** (OUT1 == 0) and after that we, wait for **car to completely cross sensor 1** (OUT1 == 1) and, we exit the ISR_exit.

For valet exit

- ➔ After branching to ISR_entry in State ex2, we branch to ISR_exit where W will remain 1, so along with that we keep on checking for (OUT1 == 0), as depicted in Fig 19.3. After that we wait again for (OUT1 == 1) and, we exit ISR.

[Note: Polling is not very efficient as we need to check at regular intervals, so instead of directly using it, we use interrupts to tell us that system needs to interact with OUT1, OUT2, and W and then we use polling. This can be justified by the assumptions that car or valet will not stop between the sensor assembly, so we will cycle back to State 1 in matter of seconds and exit ISR where we use polling.]

Use of OUT3 is just to detect a valet or car and send an interrupt to system to reset the 5 mins timer to meet the user requirements of door remaining open for 5 mins after a valet or car passes. Reason for including this is stated in reasons for design choices section at the end. We keep a separate variable named G which we toggle every time gate is opened or closed. G = 0, indicate that gate is closed and G = 1 indicate that gate Is open. For reset, we will provide low to high pulse on gate of all counters of 8253 so that the initial count value get loaded in them to reset 5 min period.

DESIGN AND MEMORY MAP

Design Diagrams and, Memory and I/O Map is included in another file to maintain readability. File name **designDiagram.pdf**.

FLOWCHART

We will continue referring states as described in previous section. In first diagram a 3 digit binary number is included with format:
LSB (**Bit 0**)= OUT2, **Bit 1** = W , MSB (**Bit 2**)= OUT1.
OUT1 = output from sensor 1
OUT2 = output from sensor 2
OUT3 = output from sensor 3
(For sensor 1,2, and 3 if object detected then output is 0)
W = output from weight switch (if 0, car otherwise valet)
G = 0 indicating door is open and 1 indicating door is closed
Count = variable storing number of occupied parking spaces
S1 = remote input (is 0 the button pressed, otherwise not)
ISR_entry = ISR for interrupt on sensor 1(for entry)
ISR_exit = ISR for interrupt on sensor 2(for exit)
ISR_reset = ISR for interrupt on sensor 3(to reset 5 min timer)
ISR_1 = ISR for interrupt on remote press
Proc_open = for opening the gate
Proc_close = for closing the gate

Main Program

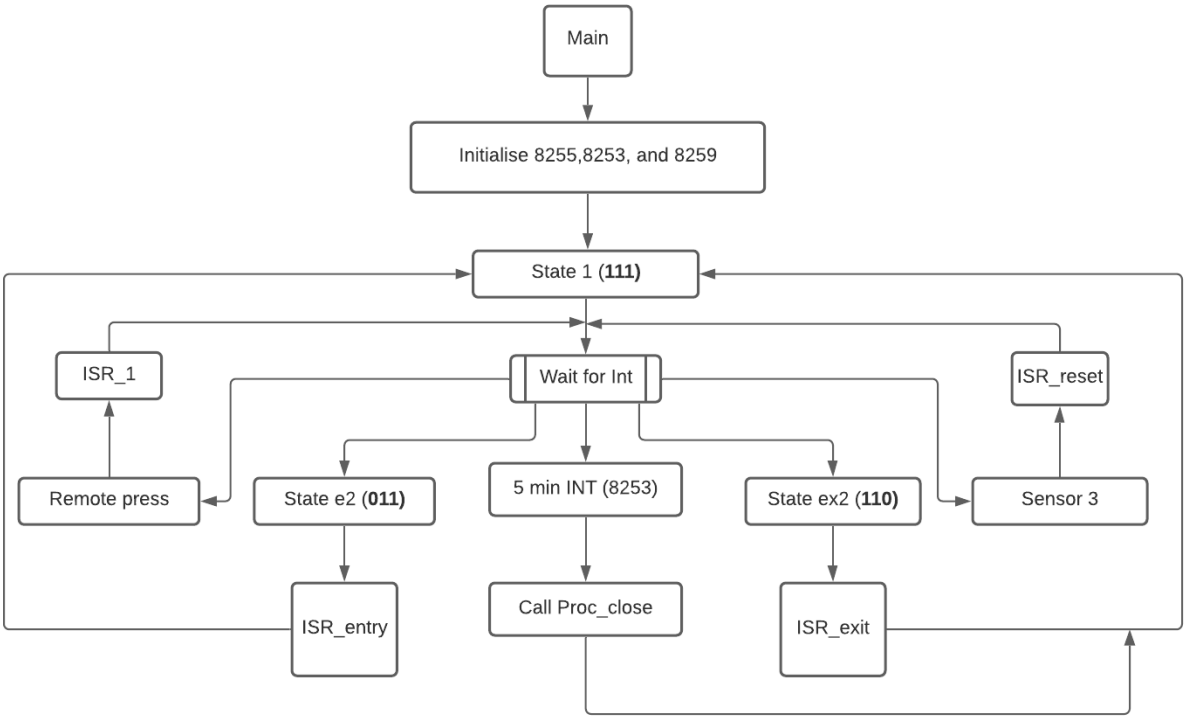

Fig 19.3

ISR_Entry

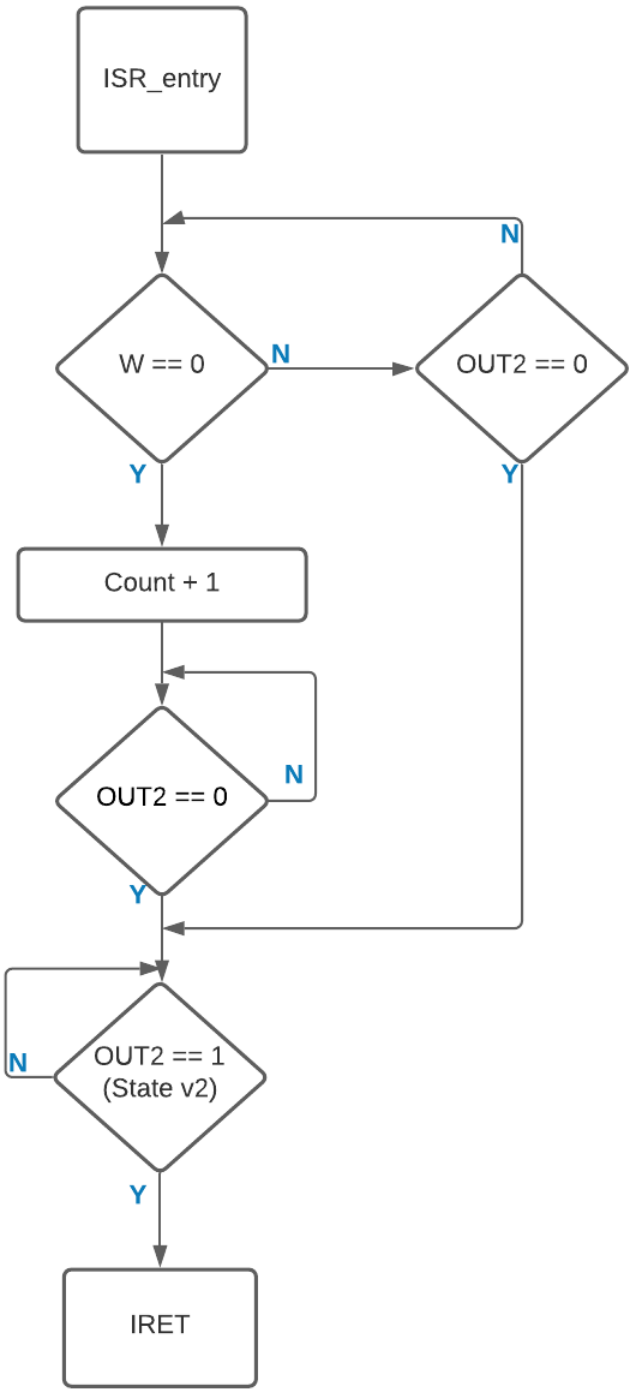

Fig 19.4

ISR_exit

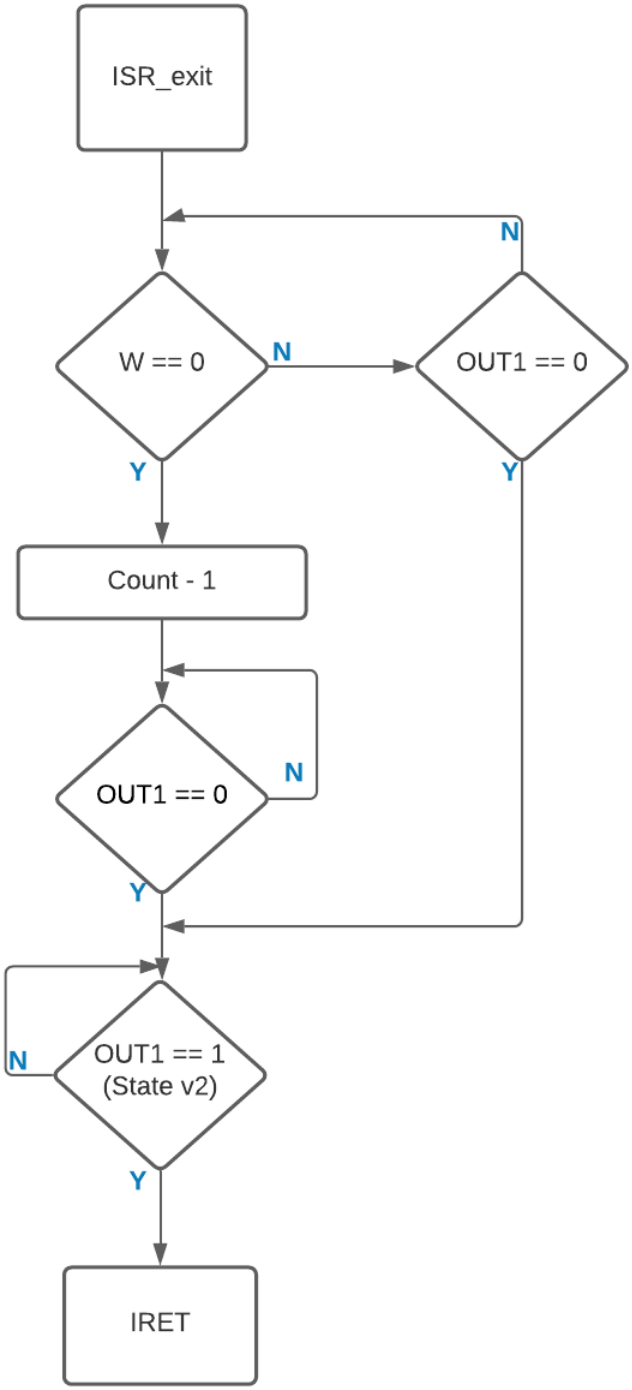

Fig 19.5

ISR_reset

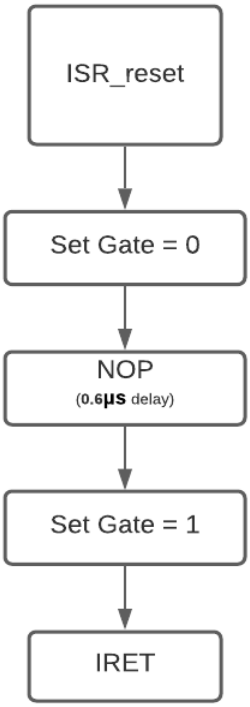

Fig 19.6

ISR_1

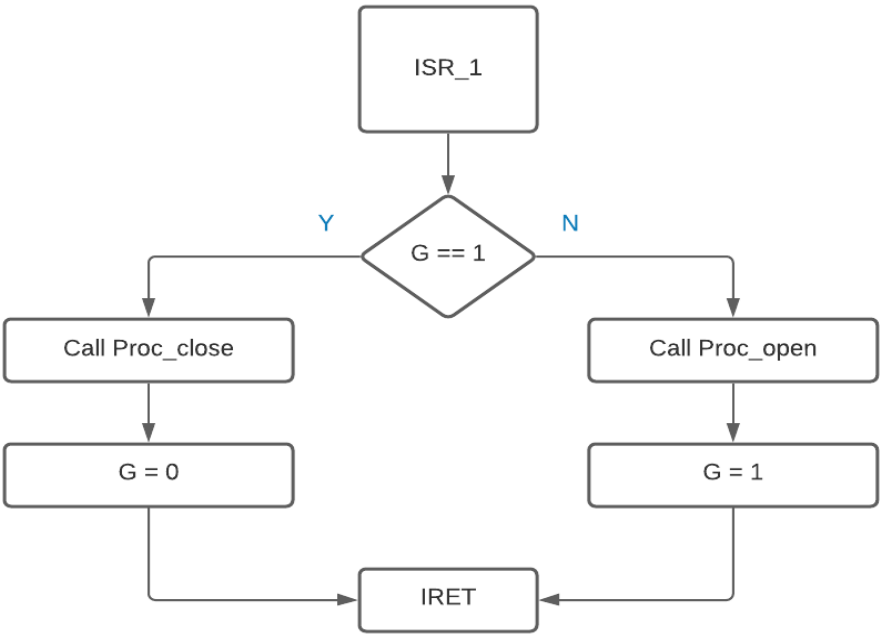

Fig 19.7

PROC_OPEN

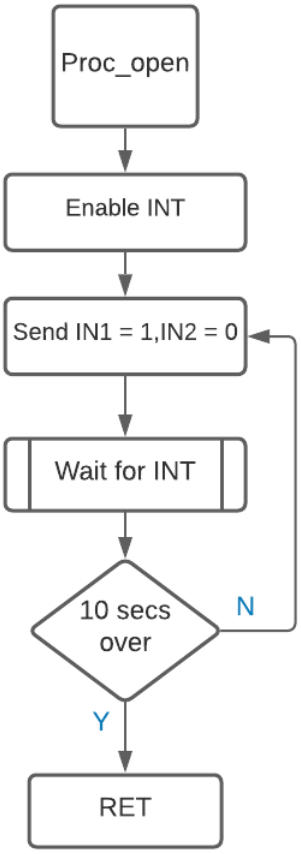

Fig 19.8

PROC_CLOSE

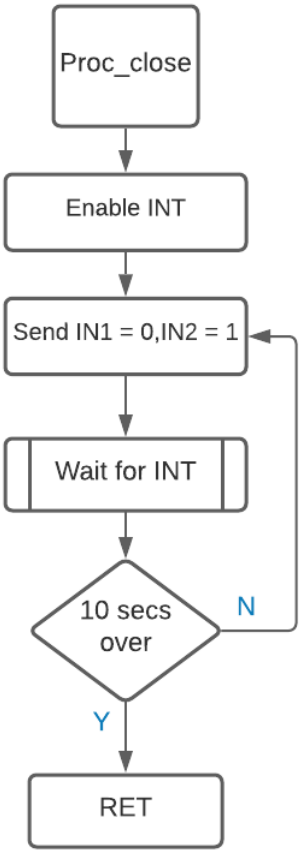

Fig 19.9

REASONS FOR DESIGN CHOICES

- Use of 3 sensors: We can use two sensors by incorporating the reset logic in entry and exit ISRs, but then our sensor assembly and garage door relative position will be a constraint for the user. We have just taken an instance of placement of working units, but there can be many configurations. So, to give the user more flexibility on placing entry/exit sensors w.r.t garage door, the additional sensor is added for resetting the 5 min period.

- Use of 4 ISRs: Every system component is reliable on user input, i.e., we do not need to check for sensors until there is a vehicle or person, so constant checking (Polling) would be an inefficient solution, so we used interrupts for every user input.

- Use of System delay for resetting the timer: For resetting the timer, we just need to reset the gate for some clock cycles and again set it. This period does not need to be very accurate, so instead of using 8253 and causing hardware overhead, a simple NOP is done, which will give sufficient delay in the gate for our purpose to be fulfilled.

Variations in Proteus Implementation with Justification

1. ROM used in memory map is 2716 but in proteus design we will simulate using available 2732.
2. IR sensors and weight switch are replaced by - Switches that toggle between 0 – 5 V – as all sensors are not there in Proteus.
3. Motor driver used in Proteus do not meet desired specifications – as Proteus doesn't have any driver matching requirements.
4. Available dc motor in proteus is not of desired capacity.