Smart Garage System

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE,
PILANI
PILANI CAMPUS

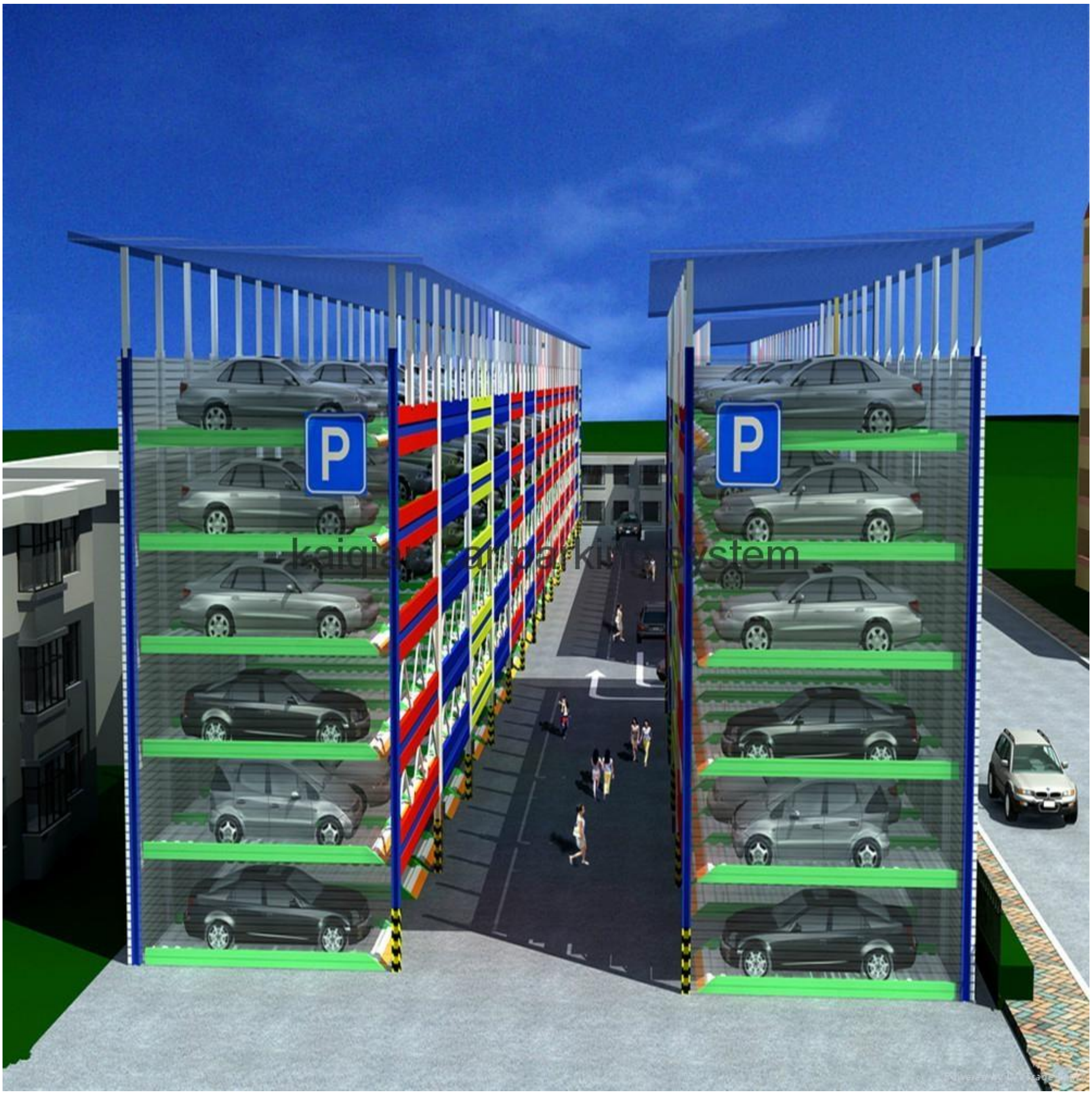
PROJECT NUMBER: 21

GROUP NUMBER: 10

Yash Bansal : 2014B2A7238P
Aman Gupta : 2014B2A7358P
Abhishek Jain : 2014B2A7363P
Subham Swastik Dora : 2014B2A7644P

Problem Statement:

To design a Smart Garage System to be used in an underground parking of a hotel with a maximum capacity of 2000 cars.A remote unit for opening and closing of hte garage door would be with every user of the garage.The remote unit has only one button.A LCD display would available indicating the number of cars in the garage. System runs from a standard power inlet available in the garage.Full and empty status will be shown through an LED.



Specifications:

-Remote unit button toggles the condition of the garage door- i.e. if the door is opened it isclosed 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.
-The system distinguishes between a person and a car using a switch 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 by using two IR sensors on either side of the gate.Depending on the sequence in which the sensors are triggered (by man\car) , an entry or exit is recorded and the count of vehicles is changed accordingly in case of cars.
-Whether a car enters or a valet enters the door remains open for a period of five minutes.During these 5 minutes the system keeps checking if the remote is triggered.
-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.
-An LCD screen (LM016L) is used to display the count of cars in the garage while an LED screen displays “FULL” when there are 2000 cars and “EMPTY” when there are no cars.

Components Used

Microprocessor	INTEL 8086	1
4 KB ROM chips	2732	4
2 KB RAM chips	6116	2
PPI	8255	2
TIMER	8253	2
LCD display	LCDLM016L	1
LED Display		2
3to 8 line Decoder	74LS138	2
AND gates	7408	1
OR gate		6
NOT gate	7404(six)	2
Motor Driver	LS239D	1
Motor	50 V , 3A	1
Bidirectional buffer	74LS245	2
Unidirectional buffers	74LS244	2
Unidirectional latches	74LS373	3

Clock Generator	8284	1
Comparator IC	LM139	1
IR receiver	TSOP 1738	3
Weight Sensor		1

ALGORITHM:

-To determine if a car/person is entering or leaving IR sensors are placed on both the inside and outside of the garage.

-The heavier car is differentiated from a person by using pressure transducers.

-A remote is also used for opening and closing of the garage. The sensors and remote signals are continuously checked for any valid input. A user can press the remote to open or close the garage.

-A count is kept for the number of cars in the garage which is updated whenever a car enters or leaves the garage. This count is displayed on the LCD screen and if the count is 2000, a red LED glows whereas in the case of 0, a green LED glows.

-Whenever a door is opened using a remote, it can be closed by pressing the remote again. Otherwise, it will automatically close after 5 minutes.

-When a car wants to enter the garage, then the user presses the remote. This would open the gate using the motor and the car can enter the garage. The door will remain open till the remote is again pressed else it will automatically close in 5 minutes. The count of cars will be updated.

-Whenever the valet wants to enter or leave the garage, the remote has to be pressed.

-When the car wants to leave the garage, the remote is pressed by the user. This would close the gate using the motor and the car can leave the garage. The door will remain open till the remote is again pressed else it will automatically close in 5 minutes. The count of cars is updated.

Assumptions:

-The garage is initially empty and the garage door is closed.
-Voltage reference for weight(force) sensor is 0.5V
-Voltage reference for IR sensors is 0.2V.
-The LED shows GREEN when the garage is empty, and shows RED when it is full, otherwise both wont light up.

-The remote is enabled manually by us on Proteus using buttons.

Memory Mapping:

Memory Interfacing Devices

ROM1_Even -> 00000h-01FFEh
ROM1_Odd -> 00001h-01FFFh
RAM1_Even -> 02000h-02FFEh
RAM2_Odd -> 02001h-02FFFh
ROM2_Even -> FE000h-FFFFEh
ROM2_Odd -> FE000h-FFFFFh

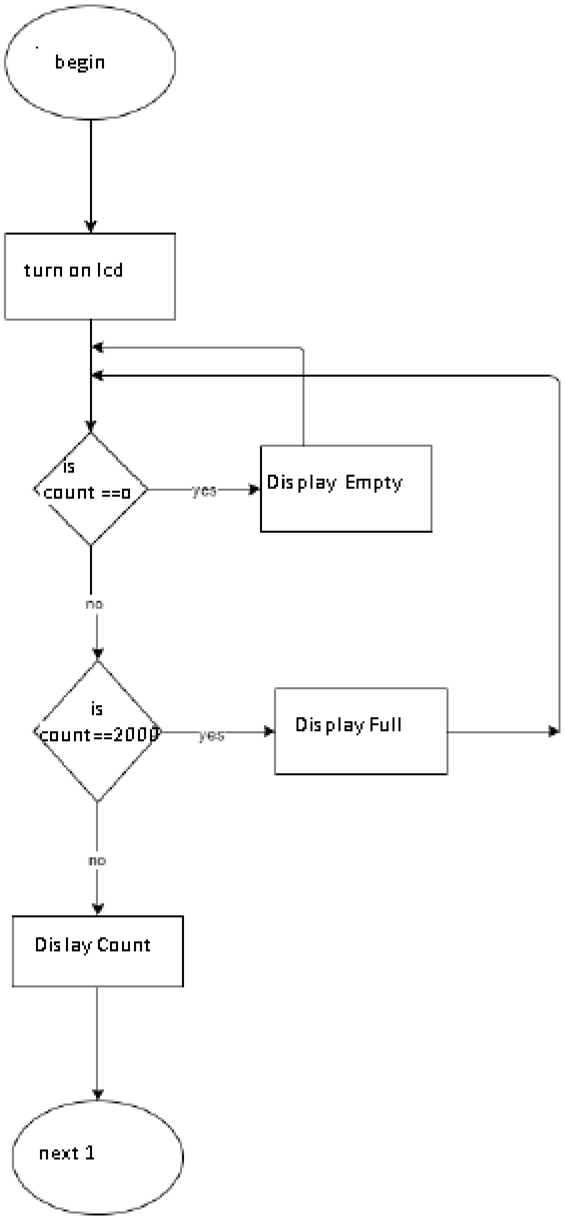
8255
PORT A -> 00h
PORT B -> 02h
PORT C -> 04h
CONTROL REGISTER -> 06h

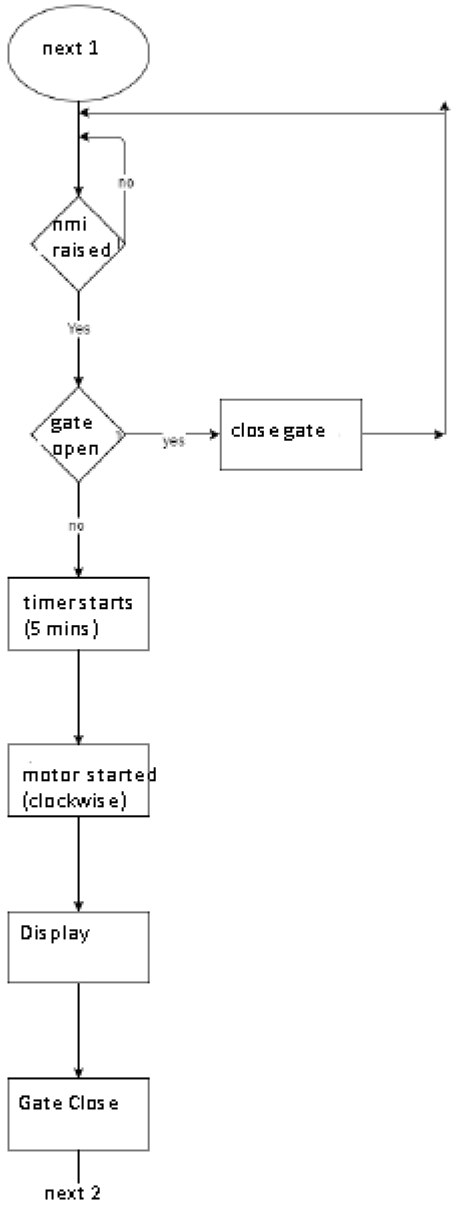
Timer
COUNTER1 -> 08h
COUNTER2 -> 0Ah
COUNTER3 -> 0Ch
CONTROL REGISTER -> Oeh

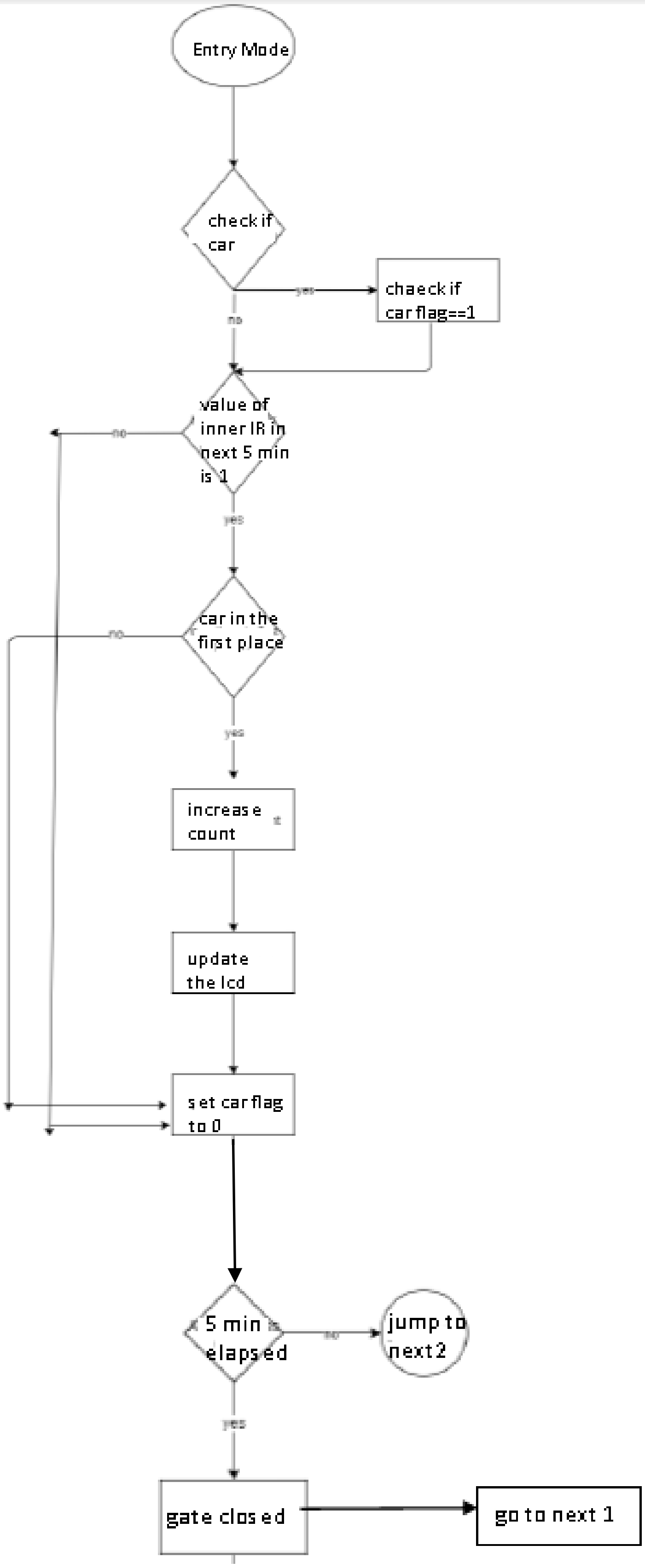
Timer 2
COUNTER 1 -> 10h
COUNTER 2 -> 12h
COUNTER 3 ->14h
CONTROL REGISTER->16h

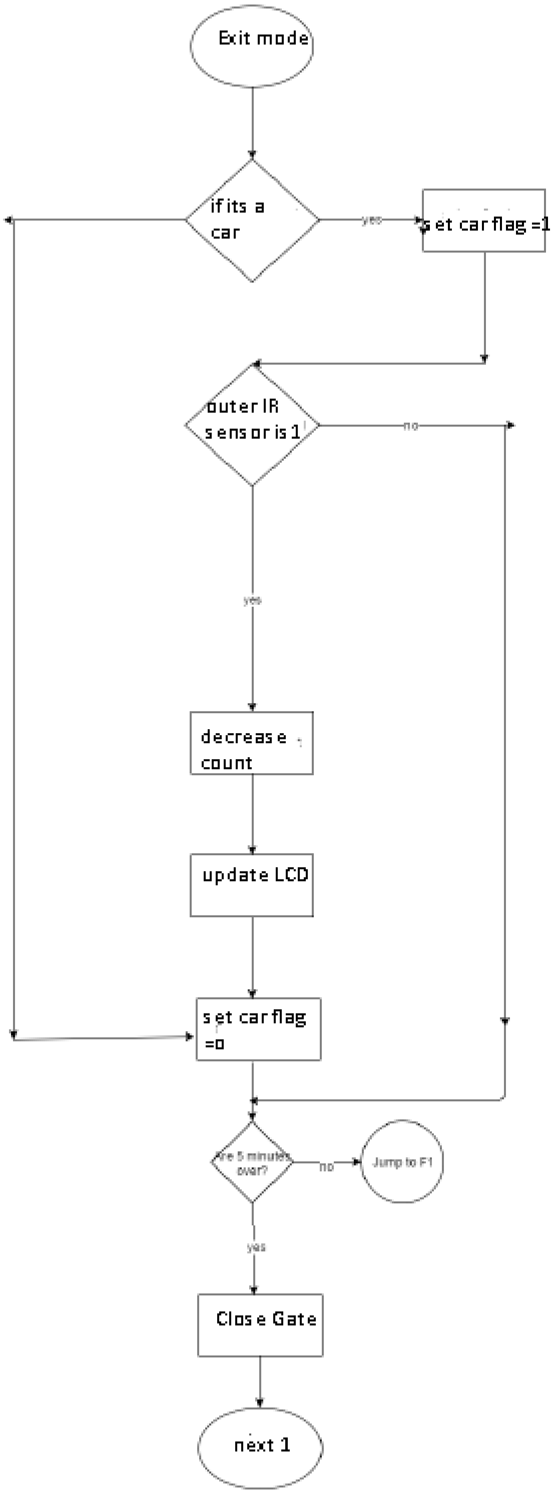
Assignment of Input Ports
PA0->Remote Input
PA1->Outer_IR Input
PA2->Transducer(Weight Sensor)
PA3->Inner_IR Input
PA4->Remote Timer

Flow Chart:


begin
turn on lcd
is
count ==o
yes
Dis play Empty
no
is
count==2000
yes
Dis play Full
no
Dis lay Count
next 1


next 1
no
nmi raised
Yes
gate open
yes
close gate
no
timer starts (5 mins)
motor started (clockwise)
Display
Gate Close
next 2


Entry Mode
check if car
yes
chaeck if car flag==1
no
value of inner IR in next 5 min is 1
no
yes
car in the first place
no
yes
increase count
update the lcd
set car flag to 0
5 min elapsed
no
jump to next2
yes
gate closed
go to next 1



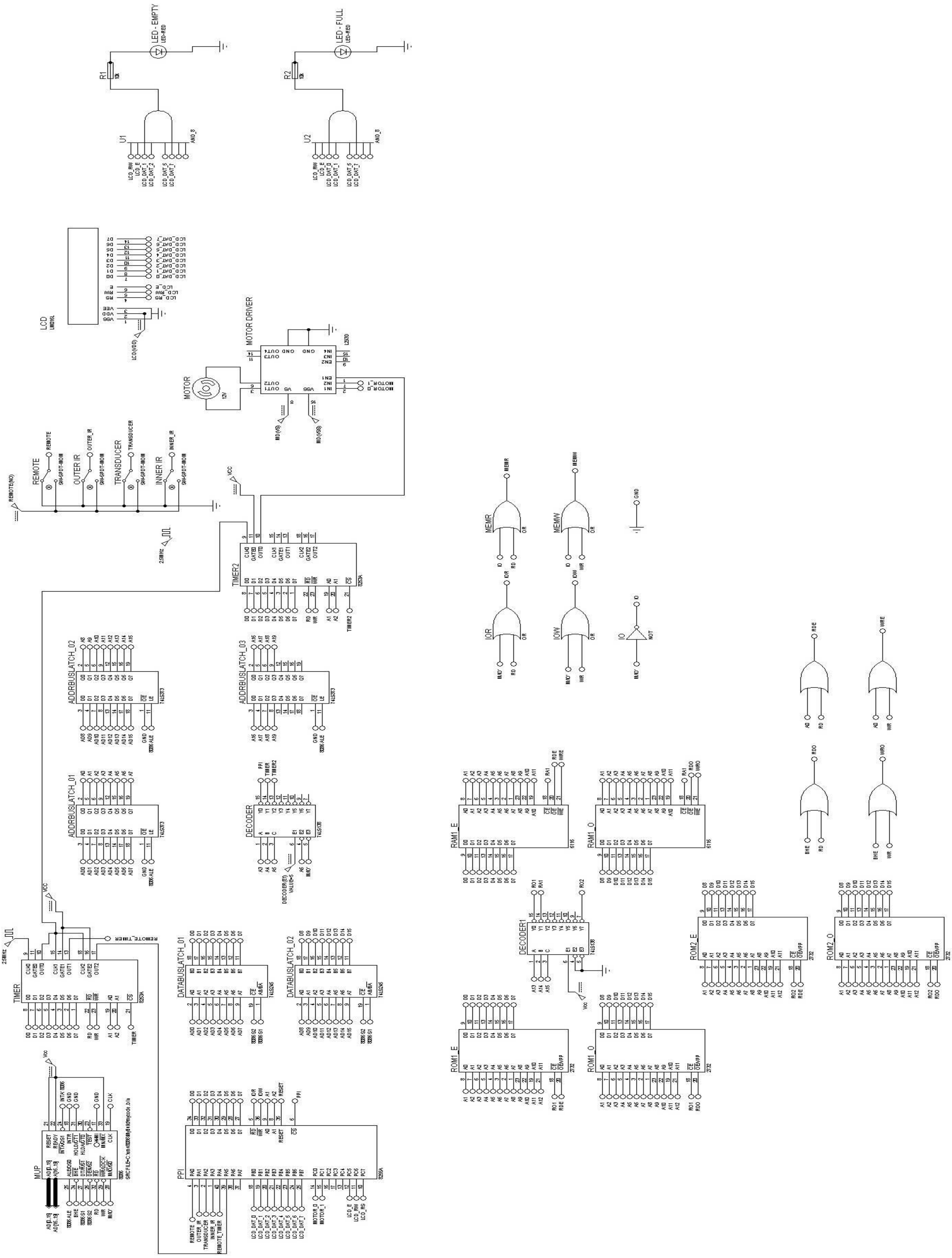
ASM CODE :

```
.model tiny
.8086
```

```
.data

        strlen db 0
        empty db 'EMPTY'
        full db 'FULL'
        count dw 0

;assigning port addresses
    inadd_word equ 00h
    lcd_data equ 02h
    lcd_motor_control equ 04h
    creg_io equ 06h

    timer_clock equ 08h
    timer_remote equ 0ah
    timer_door equ 0ch
    creg_timer equ 0eh

    timer_clock2 equ 10h
    timer_remote2 equ 12h
    timer_door2 equ 14h
    creg_timer2 equ 16h

    jmp st1
    db 1024 dup(0)

.code
.startup

st1:

    mov cx, 0000h
    mov bx, 0000h
    mov dx, 0000h


inits:

    mov ax,0200h
    mov ds,ax
    mov es,ax
    mov ss,ax
    mov sp,0FFFEH

    mov al,10000000b
    out creg_io,al

    mov al, 00110110b
    out creg_timer, al
    mov al, 0A8h  ;10101000
    out timer_clock, al  ;Timer 1 intialize
    mov al, 61h   ;01100001
    out timer_clock, al

   mov al,00110011b       ;Timer 2 intialize
   out creg_timer2,al

   call lcd_init
   call lcd_update

garageclosed:
    in al, inadd_word
    and al, 00000001b
    cmp al, 1
    je opendoor
    jmp garageclosed

garageopen:
    mov cl,0
    mov ah, 0                  ; reset car flag to 0
    in  al, inadd_word
          mov bl, al
    and bl, 00000001b
    cmp bl, 00000001b          ; check for remote press
    je closedoor
    mov bl, al
    and bl, 00010000b
    cmp bl, 00010000b          ; check for timeout (5 minutes)
    je closedoor
    mov bl, al
    and bl, 00000010b
    cmp bl, 00000010b          ; check for outer IR
    je entering
    mov bl, al
    and bl, 00001000b
    cmp bl, 00001000b          ; check for inner IR
    je exiting
    jmp garageopen

closedoor:
    call motor_clockwise
    call motor_start
    call start_door_timer
    stillclosing:
        in al, inadd_word
        and al, 00100000b
        cmp al, 00100000b       ; wait for door to close completely
        jne stillclosing

    call motor_stop
    jmp garageclosed

opendoor:
    call start_remote_timer
    call motor_anticlockwise
    call motor_start
    call start_door_timer

    stillopening:
        in al, inadd_word
        and al, 00100000b
        cmp al, 00100000b       ; wait for door to open completely
        jne stillopening

    call motor_stop
    jmp garageopen

entering:
    mov cl,0
    in al, inadd_word
    mov bl, al
    and bl, 00000001b
    cmp bl, 00000001b
    je closedoor
    mov bl,al
    and bl, 00010000b
    cmp bl, 00010000b           ; check for timeout (5 minutes)
    je closedoor
    mov bl, al
    and bl, 00000100b
    cmp bl, 00000100b           ; check for car
    jne nc00
    mov ah, 1
    nc00:
        mov bl, al
```

```
      and bl, 00001000b
      cmp bl, 00001000b       ; check for inner IR
      jne entering
   cmp ah, 1
   jne nc01
   inc count
   call lcd_update
   nc01:
      in al, inadd_word
      mov bl, al
      and bl, 00001000b
      cmp bl, 00001000b       ; debounce
      je nc01
   jmp garageopen

exiting:
   mov cl,0
   in  al, inadd_word
   mov bl, al
   mov bl, al
   and bl, 00000001b
   cmp bl, 00000001b
   je closedoor

   and bl, 00010000b
   cmp bl, 00010000b          ; check for timeout (5 minutes)
   je closedoor

   mov bl, al
   and bl, 00000100b
   cmp bl, 00000100b          ; check for car
   jne nc10

   mov ah, 1
   nc10:
   mov bl, al
      and bl, 00000010b
      cmp bl, 00000010b       ; check for outer IR
      jne exiting
   cmp ah, 1
   jne nc11
   dec count

   call lcd_update
   nc11:
        in al, inadd_word
        mov bl, al
        and bl, 00000010b
        cmp bl, 00000010b       ; debounce
        je nc11
    jmp garageopen


lcd_init proc near
    mov al, 00001111b
    out lcd_data, al
    mov bl, 00100000b
    call setlcdmode
    mov bl, 00000000b
    call setlcdmode
    ret
lcd_init endp

lcd_update proc near
    call lcd_clear
    mov al, ' '
    call lcd_add_lcd
    cmp count, 0
    jnz notempty
    lea di, empty
    mov strlen, 5
    jmp loaded
    notempty:
        cmp count, 2000
        jnz notfull
        lea di, full
        mov strlen, 4
        jmp loaded
        notfull:
                    call lcd_bcd
            ret
          loaded:
          call lcd_add_word
    ret
lcd_update endp

          lcd_bcd proc near
        mov ax, count
        mov cx, 0
        converting:
            mov bl, 10
            div bl
            add ah, '0'
            mov bl, ah
            mov bh, 0
            push bx
            inc cx
            mov ah, 0
            cmp ax, 0
            jne converting

        printing:
            pop ax
            call lcd_add_lcd
            loop printing
        ret
    lcd_bcd endp

    lcd_add_word proc near
          mov cl, strlen

          putting:
          mov al, [di]
          call lcd_add_lcd
          inc di
          loop putting
        ret
    lcd_add_word endp

    lcd_add_lcd proc near
        push ax
        out lcd_data,al
        mov bl,10100000b
        call setlcdmode
        mov bl,10000000b
        call setlcdmode
        pop ax
        ret
    lcd_add_lcd endp

    lcd_clear proc near
        mov al, 00000001b
        out lcd_data, al
        mov bl,00100000b
        call setlcdmode
        mov bl,00000000b
        call setlcdmode
        ret
    lcd_clear endp


    setlcdmode proc near
        in al, lcd_motor_control
        and al, 00011111b
        or al, bl
        out lcd_motor_control, al
        ret
    setlcdmode endp

    start_door_timer proc near
        mov al, 10110000b
        out creg_timer, al
        mov al, 90h
        out timer_door, al
        mov al, 01h
        out timer_door, al
        ret
    start_door_timer endp

    start_remote_timer proc near
        mov al, 01110000b
        out creg_timer, al
        mov al, 30h
        out timer_remote, al
        mov al, 75h
        out timer_remote, al
        ret
    start_remote_timer endp

    motor_stop proc near
        in al, lcd_motor_control
        and al, 11111100b
        or al, 00000000b
        out lcd_motor_control, al
        ret
    motor_stop endp

    motor_anticlockwise proc near
        in al, lcd_motor_control
```

```
        and al, 11111100b
        or al, 00000001b
        out lcd_motor_control, al
        ret
    motor_anticlockwise endp

    motor_clockwise proc near
        in al, lcd_motor_control
        and al, 11111100b
        or al, 00000010b
        out lcd_motor_control, al
        ret
    motor_clockwise endp

    motor_start proc near
          in al,0cah
          out timer_clock2,al
          in al,08h
          out timer_clock2,al
          in al,0d4h
          out timer_clock2,al
          in al,30h
          out timer_clock2,al
        ret
    motor_start endp

.exit
end
```

LED - EMPTY
LED - FULL
LED-RED
R1
R2
10k
U1
U2
AND_8
LCD
LM016L
MOTOR DRIVER
L293D
MOTOR
12V
REMOTE
OUTER IR
TRANSDUCER
INNER IR
SW-SPDT-MOM
REMOTE(NO)
VCC
25MHz
TIMER2
8253A
TIMER
ADDRBUSLATCH_01
ADDRBUSLATCH_02
ADDRBUSLATCH_03
74LS373
DECODER
DECODER1
74LS138
DATABUSLATCH_01
DATABUSLATCH_02
74LS245
MUP
8086
PPI
8255A
MEMR
MEMW
IOR
IOW
IO
NOT
OR
RAM1_E
RAM1_O
6116
ROM1_E
ROM1_O
ROM2_E
ROM2_O
2732
REMOTE_TIMER
GND
Vcc