BATCH - 27

2014A3PS291G - ABHINAV SINGH
2014A7PS356G - PULKIT AGARWAL
2014A7PS014G - ROHAN GOEL
2014A8PS496G - SHIKHAR SALUJA

Design Specifications

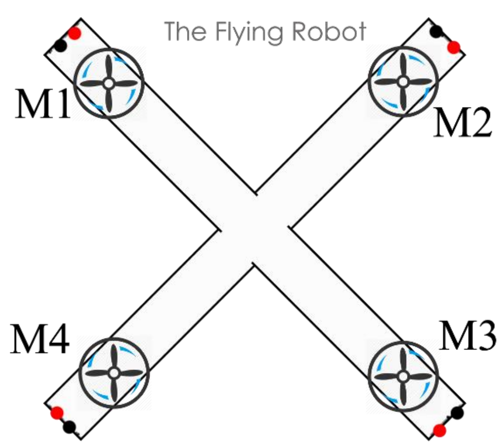
Problem Statement 33: Flying Robot

- The Flying Robot is a hovering robot that possesses the capability to remain flying directly over an area of interest until a designated mission is complete.
- The airframe is made up of 4 rods that span out from the centre and form an X shaped pattern. Attached to each rod is a DC motor and gear assembly for a propeller.
- The pitch of adjacent propellers is reversed to compensate for counter rotation, making two propeller rotate clockwise and the other two counter clockwise.
- For sensors piezo gyros are placed on the three axis of rotation provide the yaw, pitch and roll angles of the hovering robot for control.
- Gyro sensors outputs are used to adjust the speed of 4 motors to control the flying robot. In addition to the gyro sensors, four IR sensors are placed on the robot to detect obstructions if the robot approaches walls or obstacles.

Clarifications to the Problem Statement

- The robot is assumed to be already flying and we just have to maintain its stability.
- If an obstacle is detected by any of the 4 IR Sensors, a corresponding LED is switched on, indicating the detection of the obstacle.

Illustration



Assumptions made when implementing the design

1. The weight of the robot is assumed to be around ~900g (similar to the weight of many do-it-yourself quadcopters available online. So, keeping the ratio of the total thrust required by the motors at full power to the weight of the robot as 2:1, we get the total thrust as 1800g or 450g per motor. That is approximately the same as that of the motor that we chose.
2. The flying robot has three degrees of freedom: Yaw, Pitch and Roll, all of which are measured by a piezo gyroscope. Due to the limitation of the ADC, only one gyro input can be converted to digital code and read by the microprocessor at once. But this is overcome due to the speed of the microprocessor, which can process the input at a faster rate than the sampling rate of the piezo gyro chip, and thus can process the inputs of the gyroscopes sequentially.
3. We are designing the IR Sensor with the help of an IR transmitting diode, receiver and comparator. On detecting an obstacle, a LED is lighted to denote which IR sensor detected the obstacle.

Components used

Item	**Chip Number / Specification**	**Quantity**	**Addresses**	**Additional Information**
Microprocessor	8086	1		
Programmable Interval Timer	8254	2	8254(1): Counter0: A0h Counter1: A2h Counter2: A4h CREG: A6H 8254(2): Counter0: B0h Counter1: B2h Counter2: B4h CREG: B6H	
Programmable Peripheral Interface	8255A	2	8255A(1): Port A: 80h Port B: 82h Port C: 84h CREG: 86h 8255A(2): Port A: 90h Port B: 92h Port C: 94h	

			CREG: 96h	
Analog to Digital Convertor	ADC0808	1		Used with 8255A(2)
Piezo Gyroscope	Murata ENV-05D	3		Used with ADC0808
Motor Driver	L293D	2		1 chip controls 2 motors.
Motor	EMAX MT2204	4		Voltage: 12V Thrust: 440 g Speed: 16300 RPM
Propeller	Carbon Fiber Propeller 6”*3”	4		To be used with motor
Quad Differential Comparator	LM339	1		1 chip has 4 comparators, used to make IR Sensor
IR LED (Transmitter)	Everlight 5mm Infrared LED IR533C	4		
IR Receiver	Everlight 5mm Silicon PIN Photodiode , T-1 3/4 PD333-3B/H0/L2	4		
Aluminium Rods	9” each	4		Used in making the body of the robot
Batteries	-Turnigy 3S 2200mAh Lipo Battery Pack 11.1v-12v -20-30C 3S - 5V Battery	1 1		For Motor For V_{CC}
Clock Generator	8284	1		Feed Clock to the Microprocessor
Crystal	15Hz HC 49US	1		Used with 8284
3:8 Decoder	74LS138	1		Use A_{12}, A_{13} and A_{14} as selection lines
EPROM	MM2716E 8-bit data bus (Size 2K)	4	ROM1: 00000h-00FFFh ROM2: FF000h-FFFFFh	Memory Addressing covers both even & odd banks
RAM	IDT6116SA CMOS Static RAM (Size 2K)	2	RAM1: 01000h-01FFFh	Memory Addressing covers both even & odd banks
Octal Buffers	74LS244	1		Strengthen the read,write and M/IO’ signals
Octal Bus Transceiver	74LS245	2		Demultiplex data lines

Octal D-Type Trans	74LS373	3		Demultiplex Address lines

For specific information on a particular component please consider its datasheet which resides in the folder ‘datasheets’.

Explanation

- Gyro Angle and PWM (motor speed handling)

For Yaw and Pitch:

Angle Deviation	**Corresponding duty cycle on motors to balance** (on opposite motors)
-60° to -30°	70% PWM and 30% PWM
-30° to 0°	60% PWM and 40% PWM
0°	50% PWM and 50% PWM
0° to 30°	40% PWM and 60% PWM
30° to 60°	30% PWM and 70% PWM

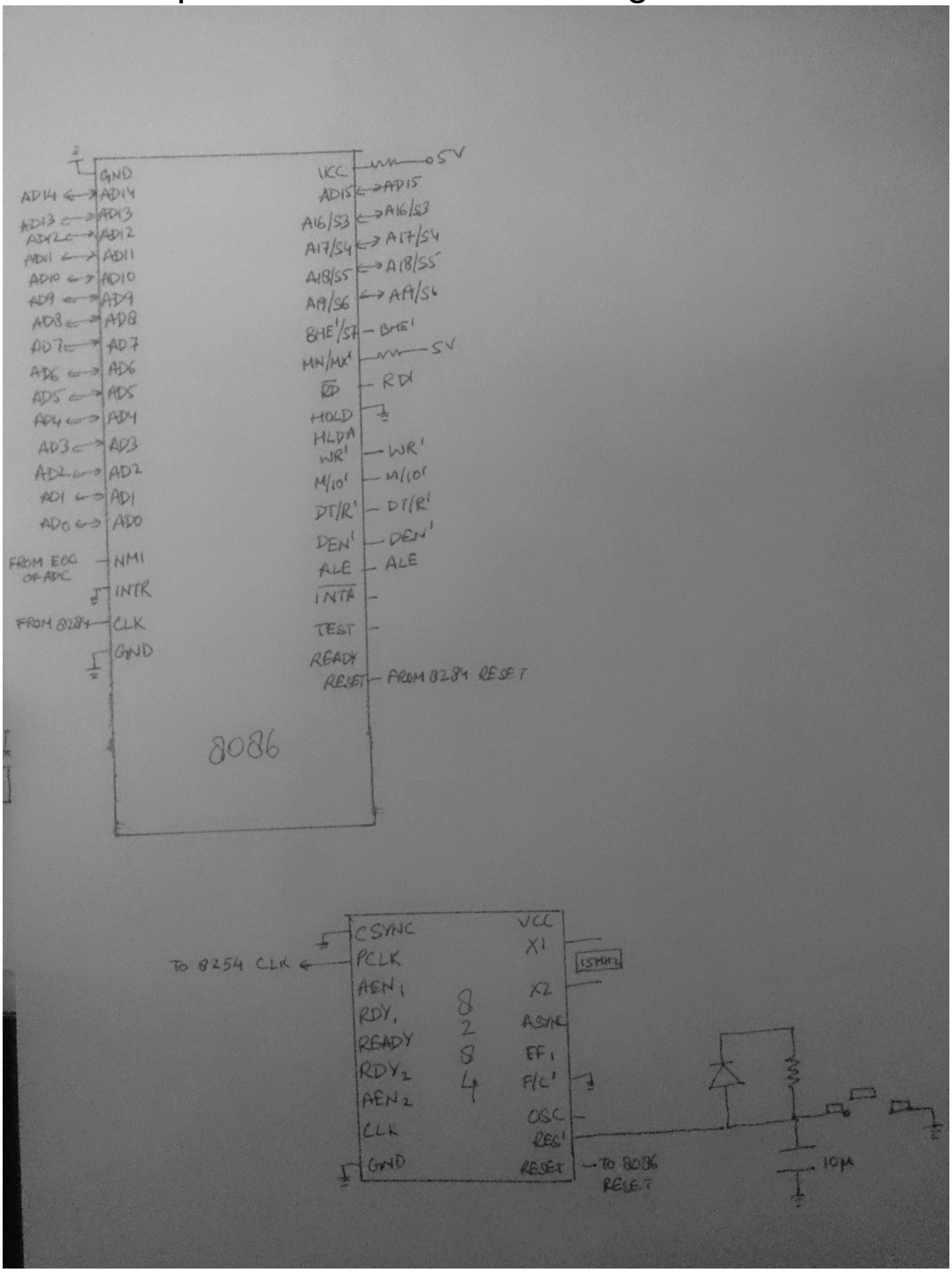
For roll, the motors are made to spin at the same speed, which cancels out the net torque and thus stabilizes the robot.

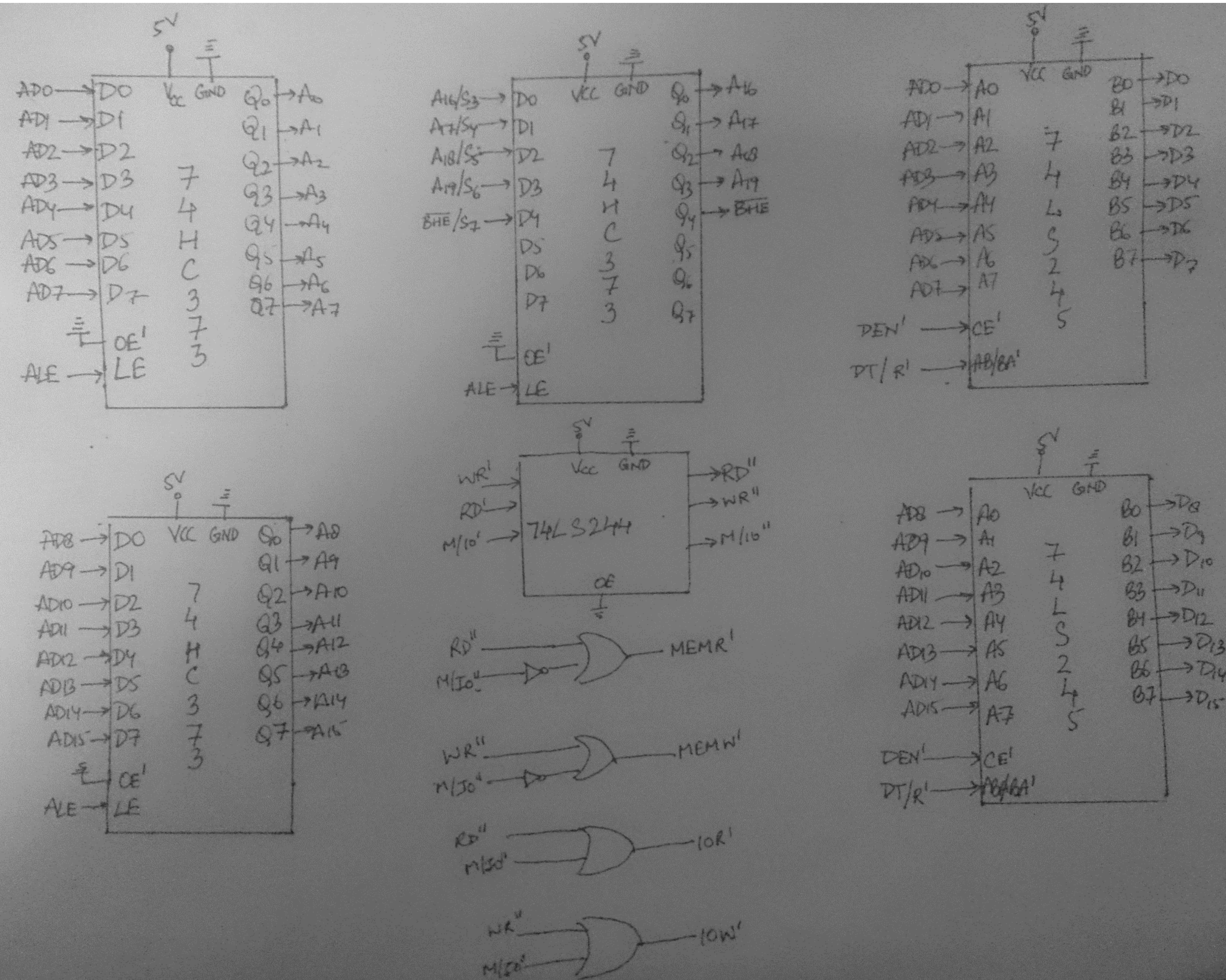
Angle deviation is calculated by integrating the angular velocity and hence an appropriate response is given by changing the PWM of the motors which in turn changes the RPM of the motors.

Here PWM (with desired duty cycle) is generated through the means of programmable timer 8254

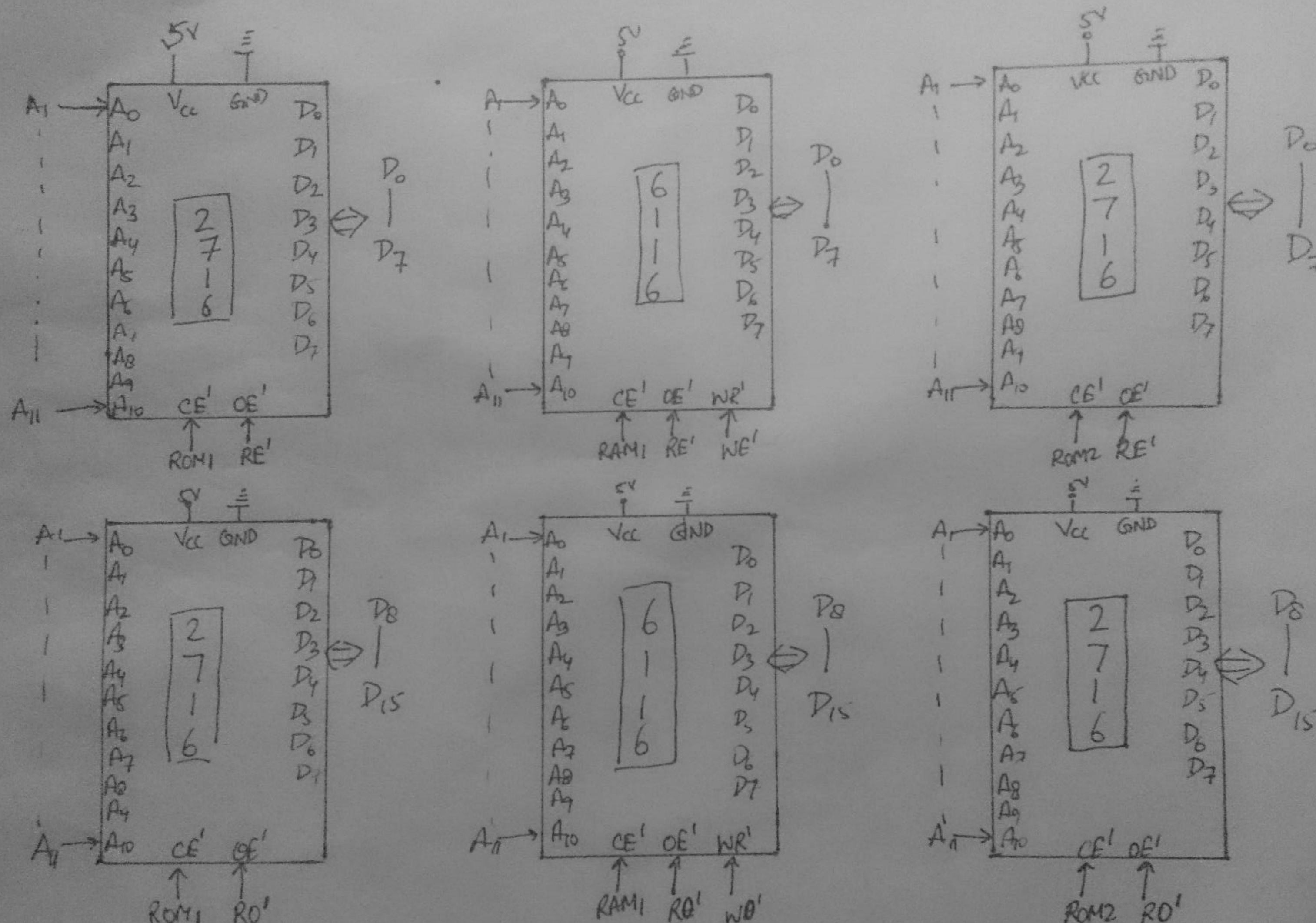
Also M1 and M3 rotate anticlockwise whereas M2 and M4 rotate clockwise (which is controlled by the motor driver). This is done to compensate for counter rotation.

The complete hardware circuit diagram

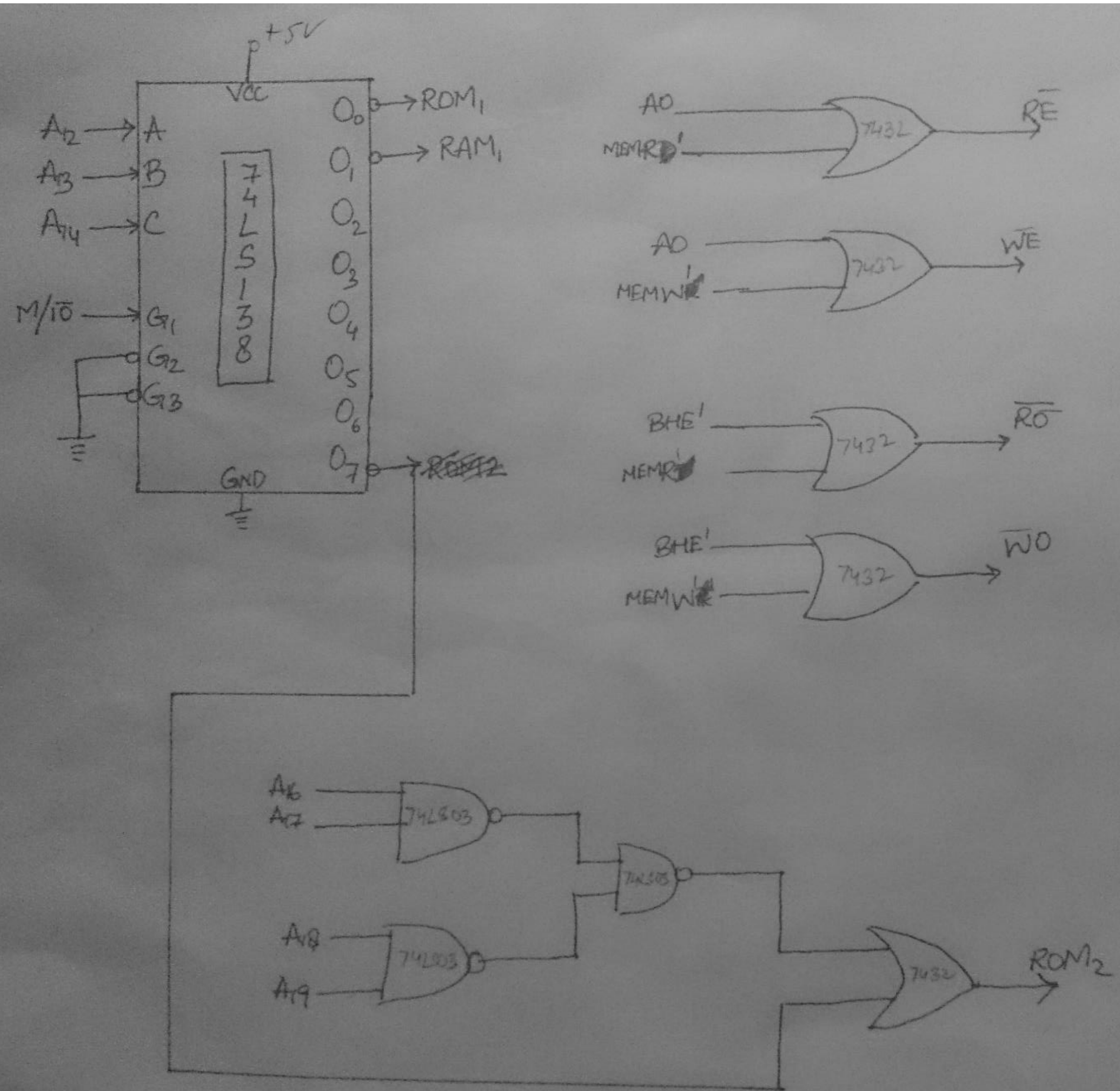




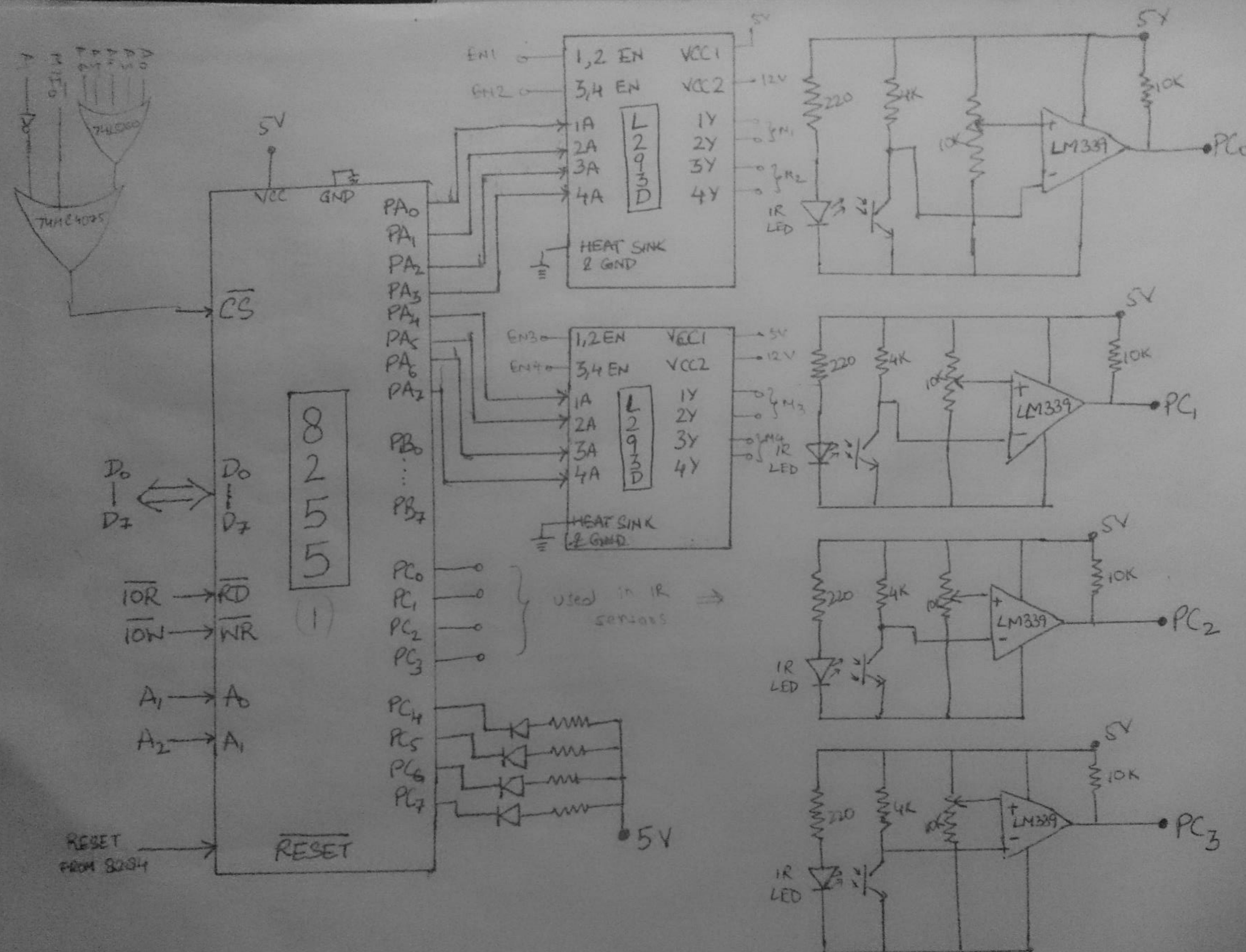
74HC373
74HC373
74LS245
74LS245
74LS244
5V
VCC
GND
OE'
LE
ALE
CE'
DEN'
DT/R'
AB/BA'
A16/S3
A17/S4
A18/S5
A19/S6
BHE/S7
BHE
WR'
RD'
M/IO'
RD''
WR''
M/IO''
MEMR'
MEMW'
IOR'
IOW'

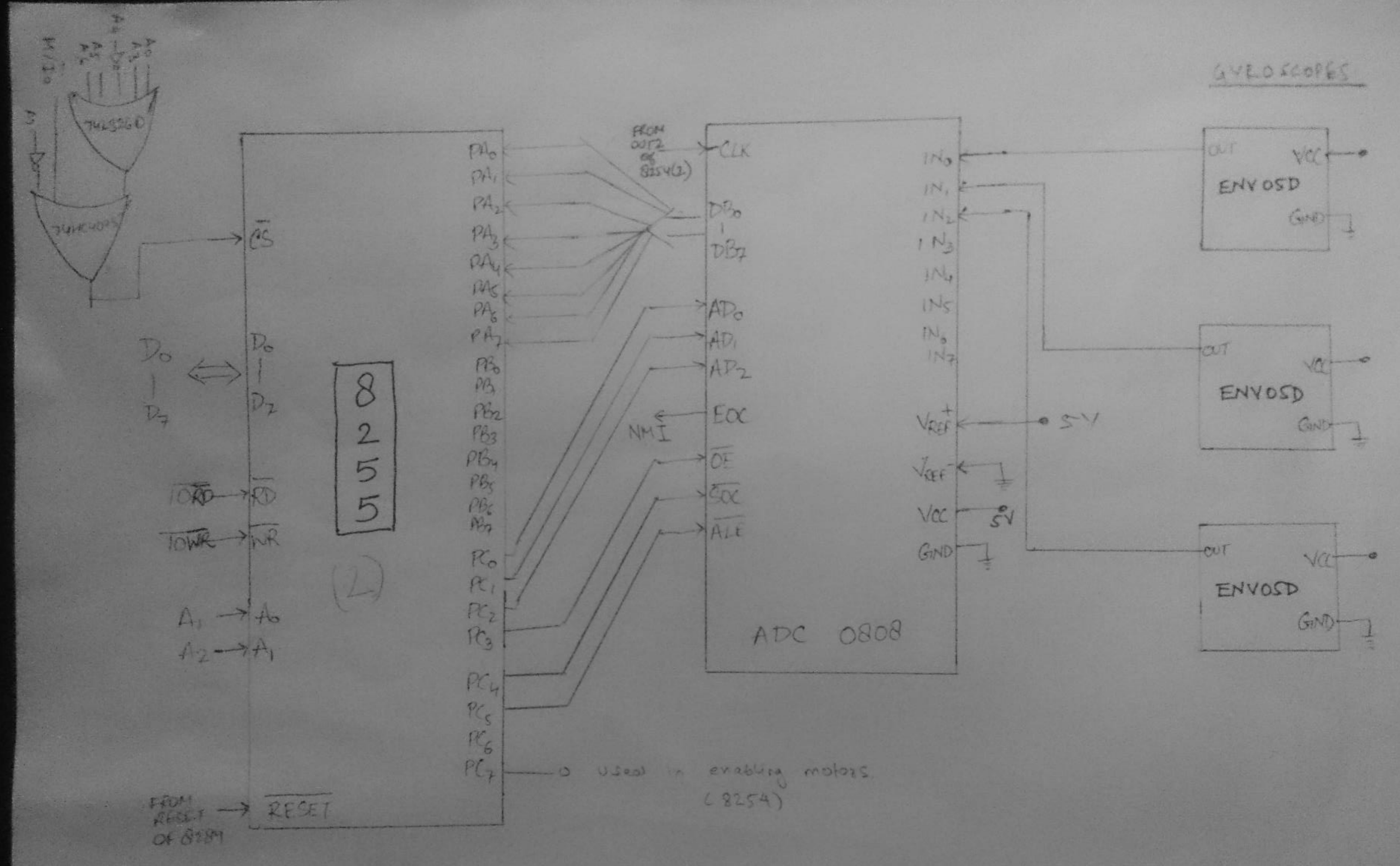


2716
ROM1
RE'
CE'
OE'
6116
RAM1
WE'
WR'
ROM2
RO'
WO'
5V
Vcc
GND
A1
A11
A0
A10
D0
D7
D8
D15

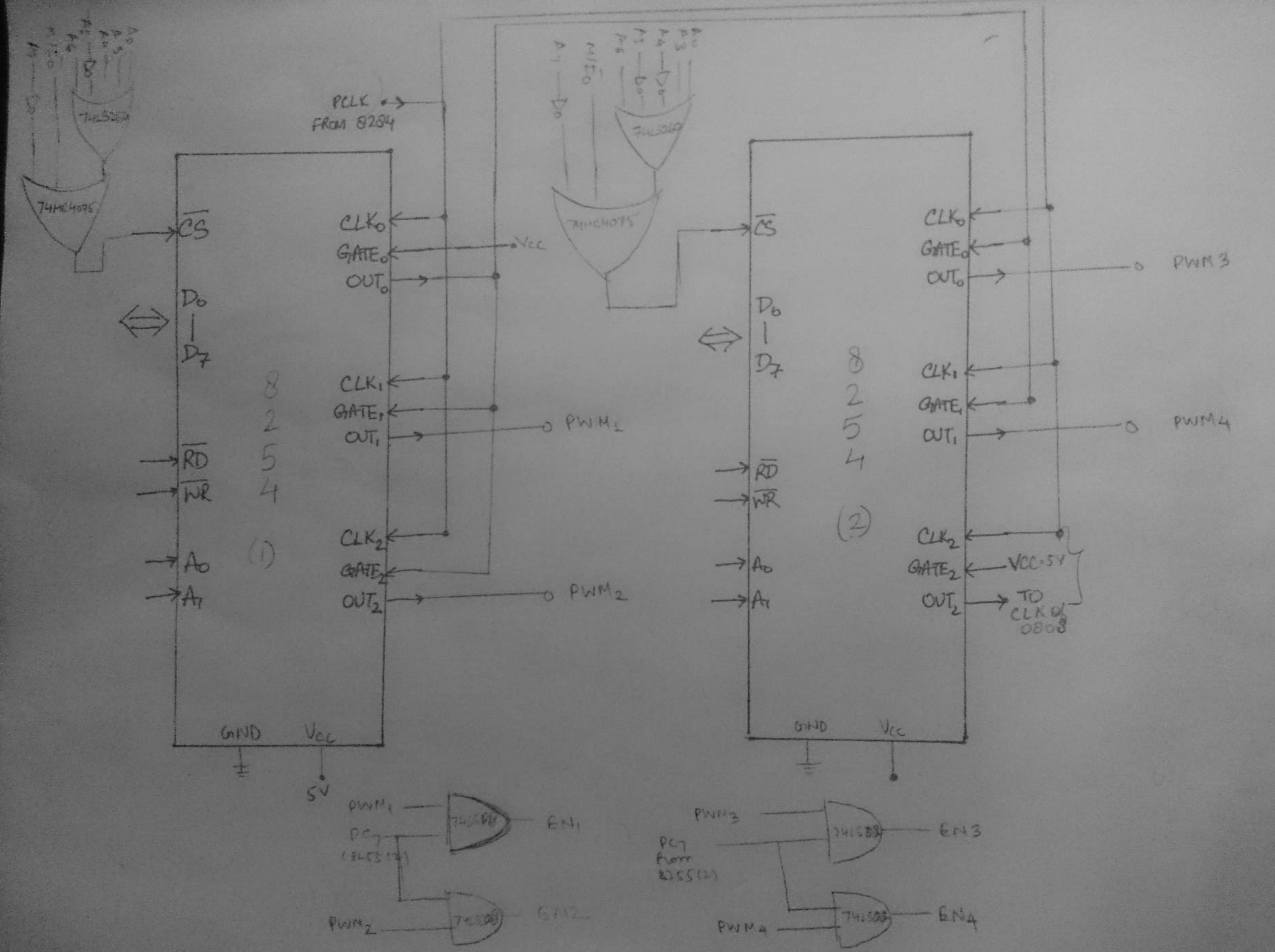


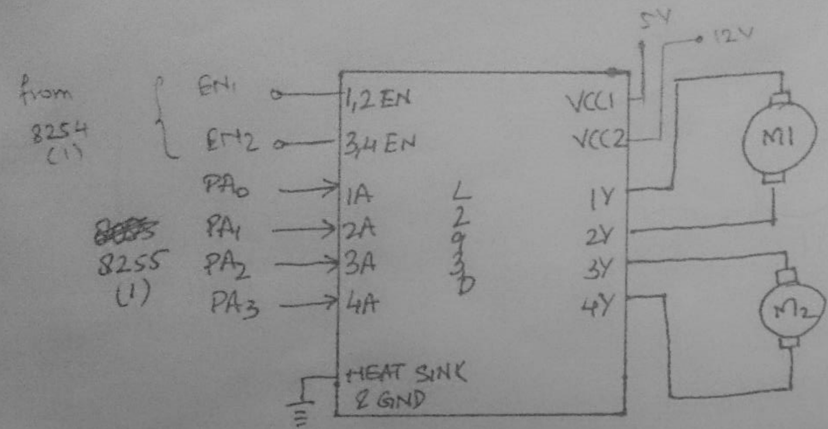
+5V
VCC
74LS138
A2
A3
A14
M/IO
A
B
C
G1
G2
G3
GND
O0
O1
O2
O3
O4
O5
O6
O7
ROM1
RAM1
A0
MEMRD'
7432
RE
MEMW'
WE
BHE'
MEMR'
RO
WO
A16
A17
A18
A19
74LS03
ROM2

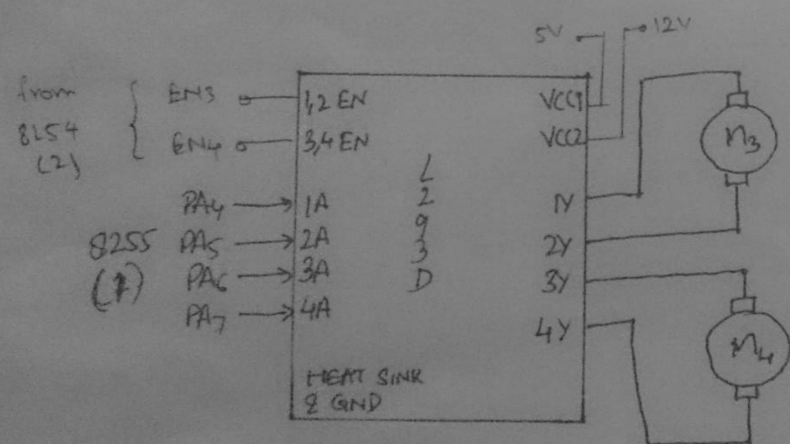

74LS00
74HC4075
5V
VCC
GND
$\overline{CS}$
8255
D_0
D_7
$\overline{IOR}$
$\overline{RD}$
$\overline{IOW}$
$\overline{WR}$
A_1
A_0
A_2
A_1
RESET FROM 8284
$\overline{RESET}$
PA_0
PA_1
PA_2
PA_3
PA_4
PA_5
PA_6
PA_7
PB_0
PB_7
PC_0
PC_1
PC_2
PC_3
PC_4
PC_5
PC_6
PC_7
used in IR sensors
EN1
EN2
EN3
EN4
1,2 EN
3,4 EN
VCC1
VCC2
1A
2A
3A
4A
1Y
2Y
3Y
4Y
L293D
HEAT SINK 2 GND
12V
M_1
M_2
M_3
M_4
IR LED
220
4K
10K
LM339



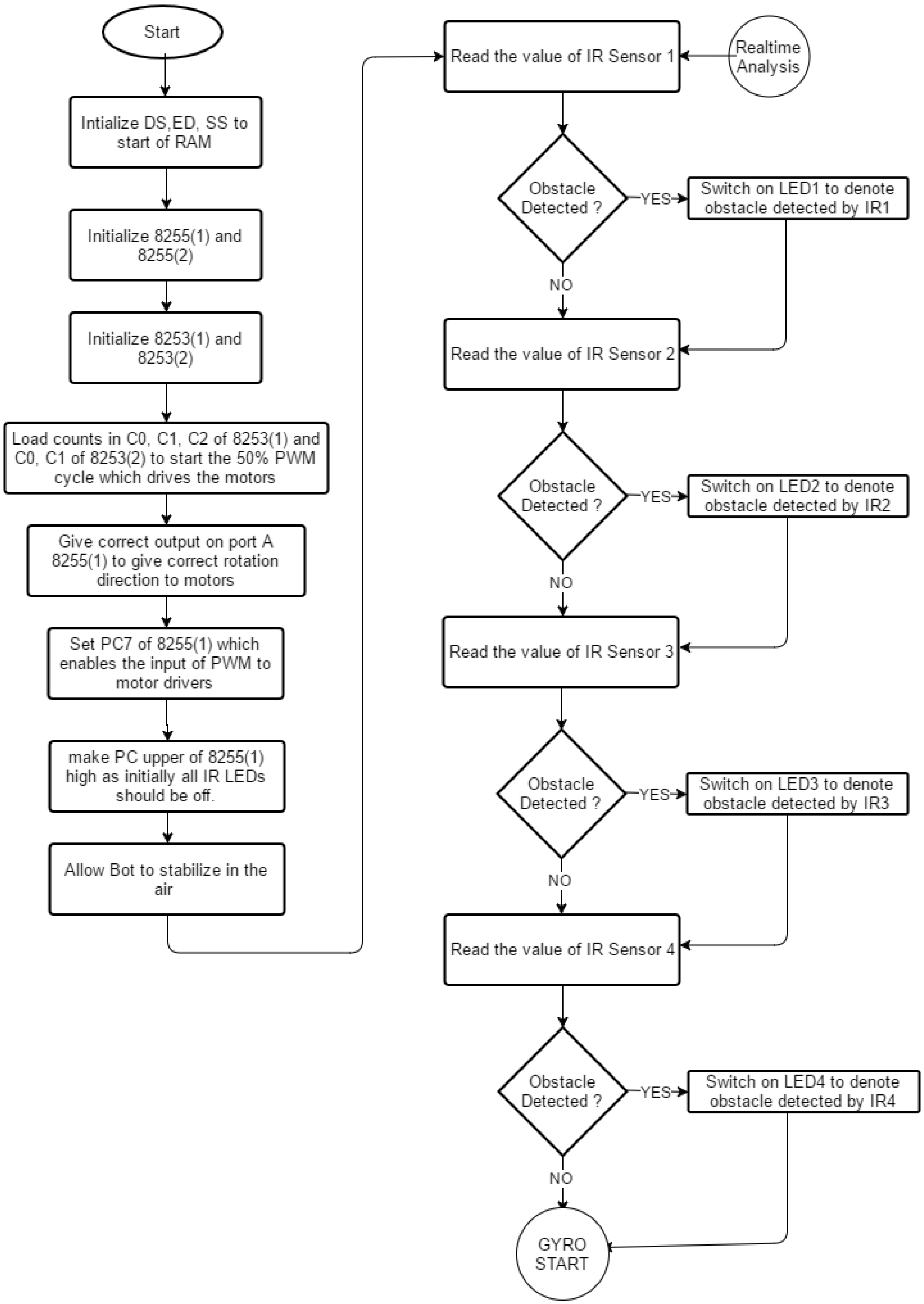
GYROSCOPES
74LS260
74HC4075
M/Io
CS
D0
D7
8255
(2)
IORD
RD
IOWR
WR
A1
A0
A2
RESET
FROM RESET OF 8284
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
Used in enabling motors (8254)
FROM OUT2 OF 8254(2)
CLK
DB0
DB7
AD0
AD1
AD2
EOC
NMI
OE
SOC
ALE
ADC 0808
IN0
IN1
IN2
IN3
IN4
IN5
IN6
IN7
VREF+
VREF-
VCC
GND
5V
ENVOSD
OUT
VCC
GND

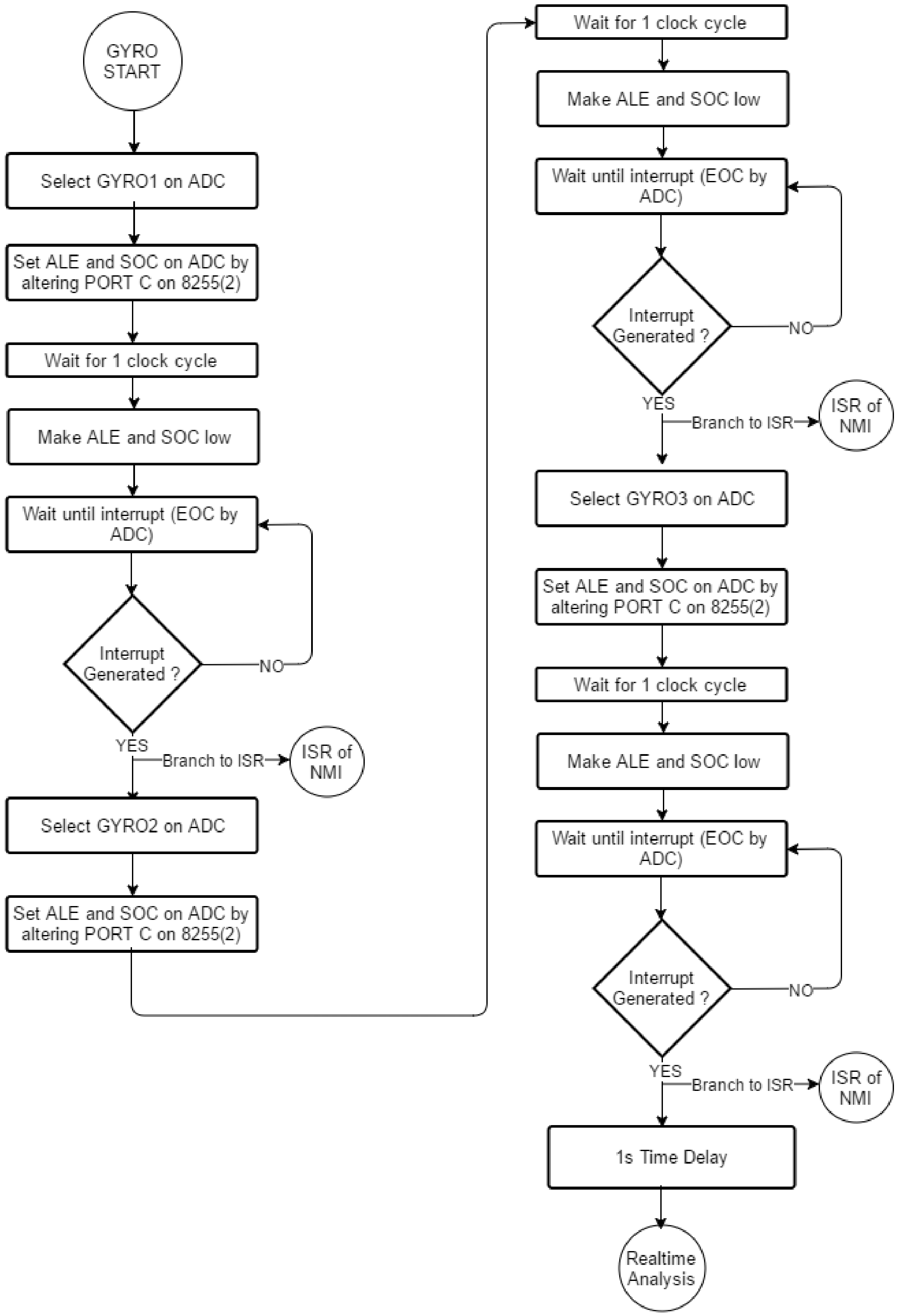

PCLK
FROM 8284
74LS260
74HC4075
CS
D0
D7
RD
WR
A0
A1
8254
(1)
(2)
CLK0
GATE0
OUT0
CLK1
GATE1
OUT1
CLK2
GATE2
OUT2
Vcc
GND
5V
PWM1
PWM2
PWM3
PWM4
VCC=5V
TO CLK of 0808
EN1
EN3
EN4
PC7 from 8255(2)

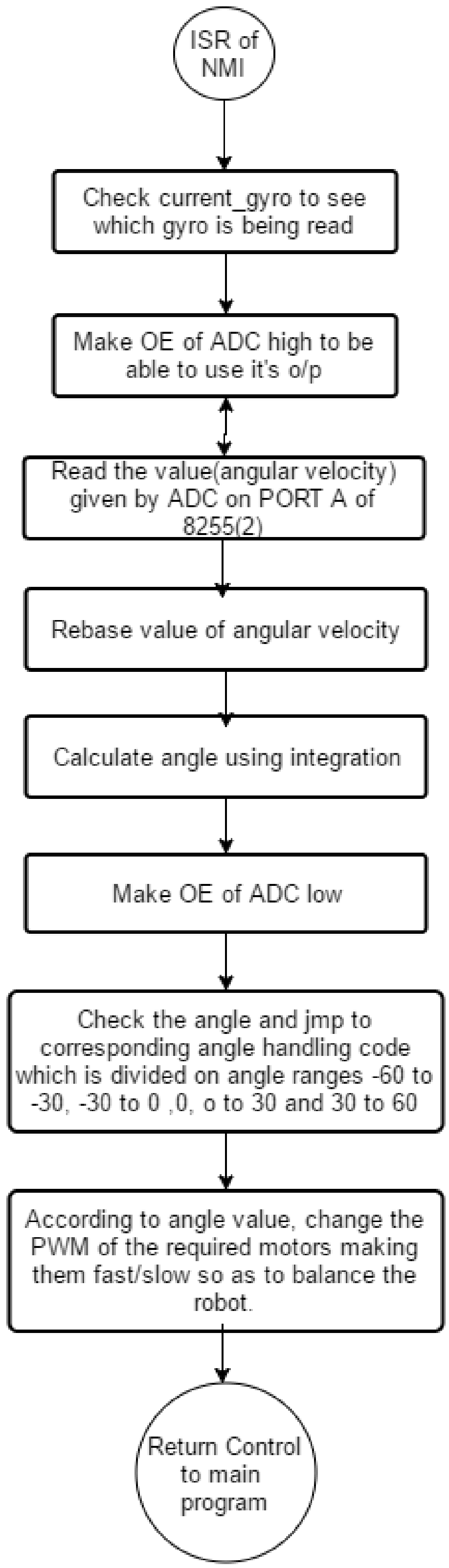

from 8254 (1)
EN1
EN2
8255 (1)
PA0
PA1
PA2
PA3
1,2 EN
3,4 EN
1A
2A
3A
4A
L293D
HEAT SINK & GND
VCC1
VCC2
1Y
2Y
3Y
4Y
5V
12V
M1
M2


from 8254 (2)
EN3
EN4
8255 (1)
PA4
PA5
PA6
PA7
1,2 EN
3,4 EN
1A
2A
3A
4A
L293D
HEAT SINK & GND
VCC1
VCC2
1Y
2Y
3Y
4Y
5V
12V
M3
M4

Flow chart of the software




GYRO START
Select GYRO1 on ADC
Set ALE and SOC on ADC by altering PORT C on 8255(2)
Wait for 1 clock cycle
Make ALE and SOC low
Wait until interrupt (EOC by ADC)
Interrupt Generated ?
NO
YES
Branch to ISR
ISR of NMI
Select GYRO2 on ADC
Set ALE and SOC on ADC by altering PORT C on 8255(2)
Wait for 1 clock cycle
Make ALE and SOC low
Wait until interrupt (EOC by ADC)
Interrupt Generated ?
NO
YES
Branch to ISR
ISR of NMI
Select GYRO3 on ADC
Set ALE and SOC on ADC by altering PORT C on 8255(2)
Wait for 1 clock cycle
Make ALE and SOC low
Wait until interrupt (EOC by ADC)
Interrupt Generated ?
NO
YES
Branch to ISR
ISR of NMI
1s Time Delay
Realtime Analysis


ISR of NMI
Check current_gyro to see which gyro is being read
Make OE of ADC high to be able to use it's o/p
Read the value(angular velocity) given by ADC on PORT A of 8255(2)
Rebase value of angular velocity
Calculate angle using integration
Make OE of ADC low
Check the angle and jmp to corresponding angle handling code which is divided on angle ranges -60 to -30, -30 to 0 ,0, o to 30 and 30 to 60
According to angle value, change the PWM of the required motors making them fast/slow so as to balance the robot.
Return Control to main program

References

- The Hardware Design (the pen and paper design) is based on the slides uploaded as part of the Microprocessor Programming & Interfacing CS/ECE/EEE/INSTR F241 course.
- The Proteus Design and the Assembly Code has been completed based on the design done in the Proteus Tutorial Class and the sample design and the template code uploaded on the edx website[1].

[1] https://edge.edx.org/courses/course-v1:BITSX+F241+2015-16_Semester_II/b68a3d8ab70946febf00893d970c8c5d/