LINE FOLLOWING ROBOT

BY

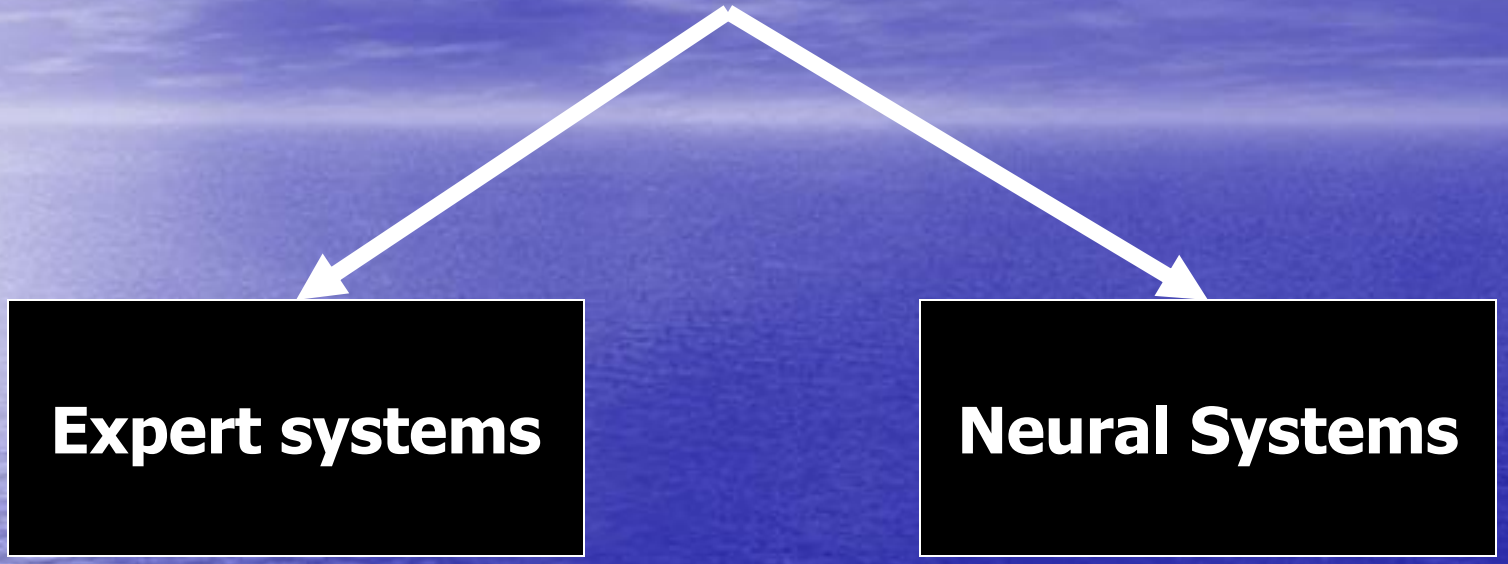
AMITHASH E. PRASAD

(1GA01EC002)

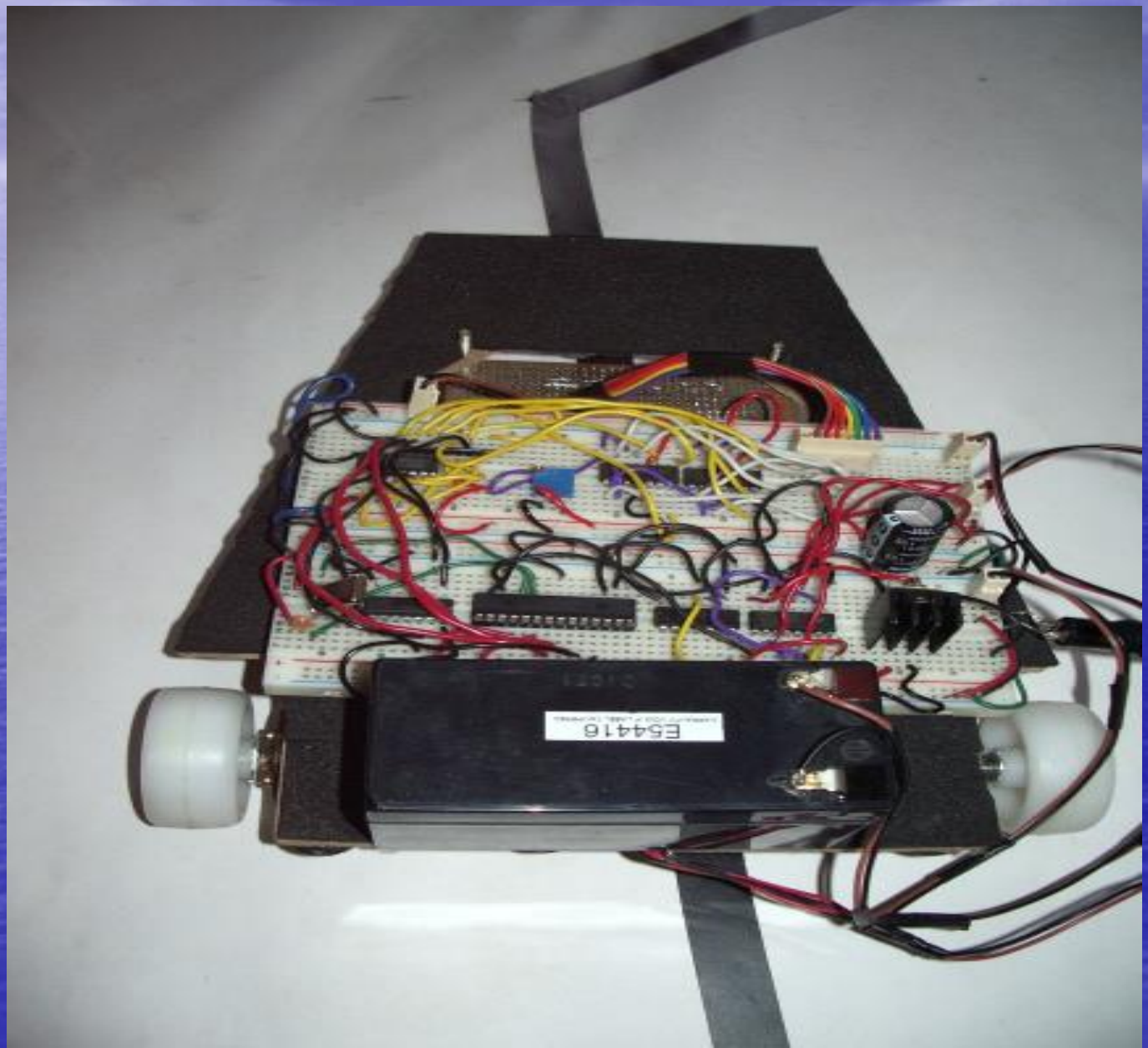
What is a robot?

- Robots are machines which do a task which would otherwise be done by human labor.
- Robots may or may not possesses intelligence.

TYPES OF ROBOTIC INTELLIGENCE



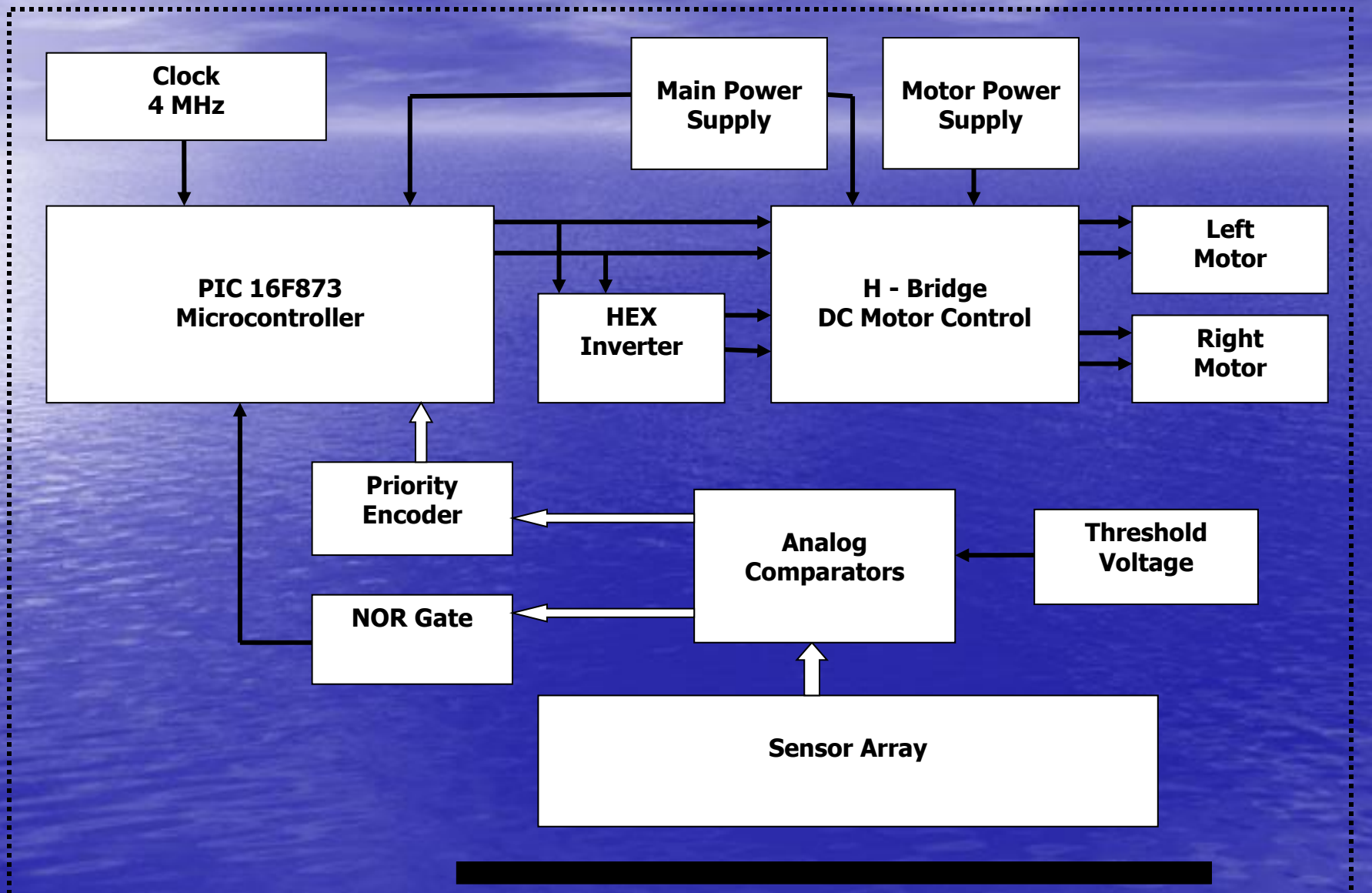
THE LINE FOLLOWING ROBOT (LFR)



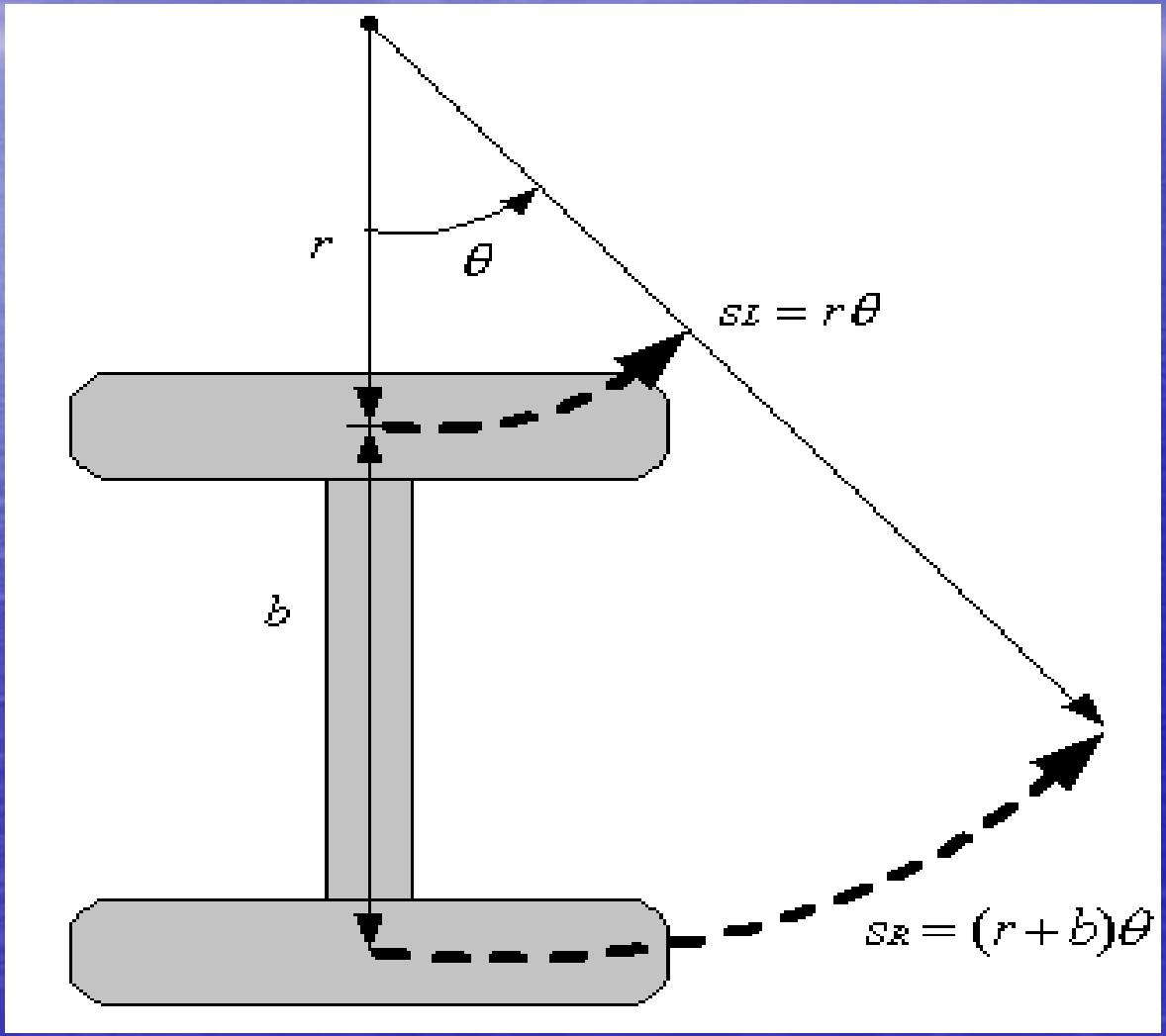
REQUIREMENT OF THE LFR

- The robot must be capable of following a line.
- It should be capable of taking various degrees of turns
- It must be prepared of a situation that it runs into a territory which has no line to follow. (Barren land syndrome)
- The robot must also be capable of following a line even if it has breaks.
- The robot must be insensitive to environmental factors such as lighting and noise.
- It must allow calibration of the line's darkness threshold.
- Scalability must be a primary concern in the design.
- The color of the line must not be a factor as long as it is darker than the surroundings.

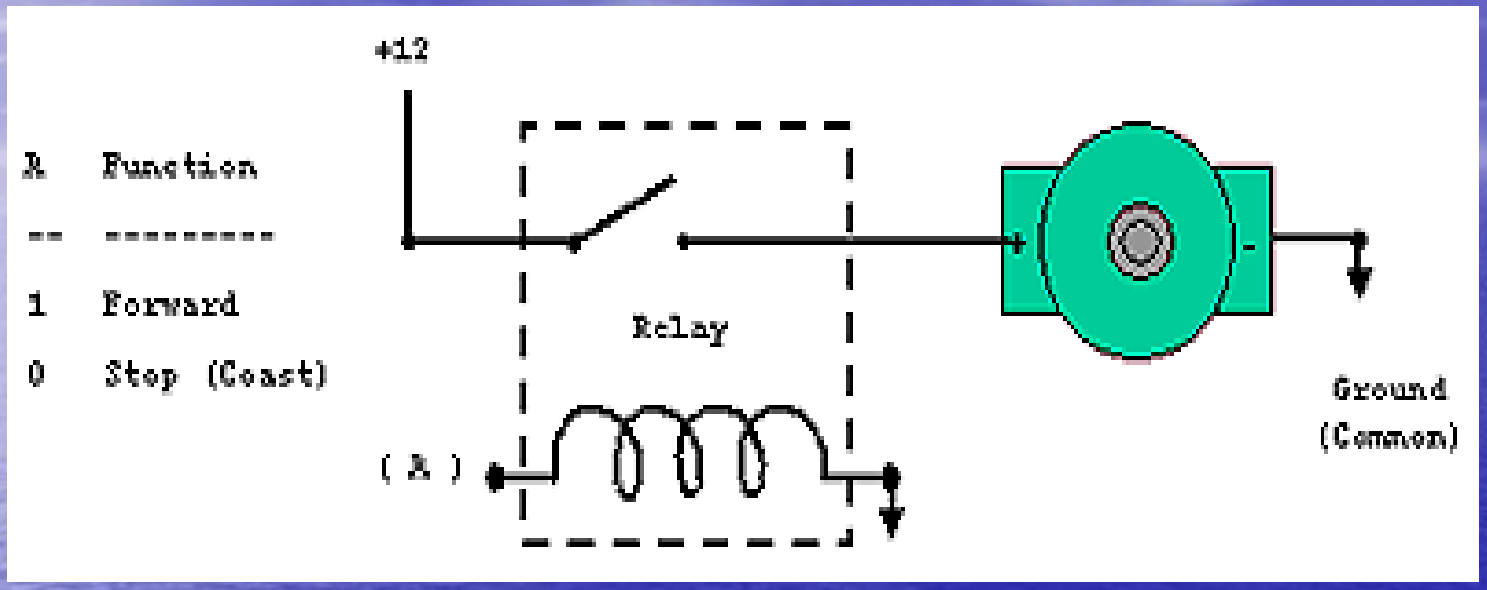
THE BLOCK DIAGRAM

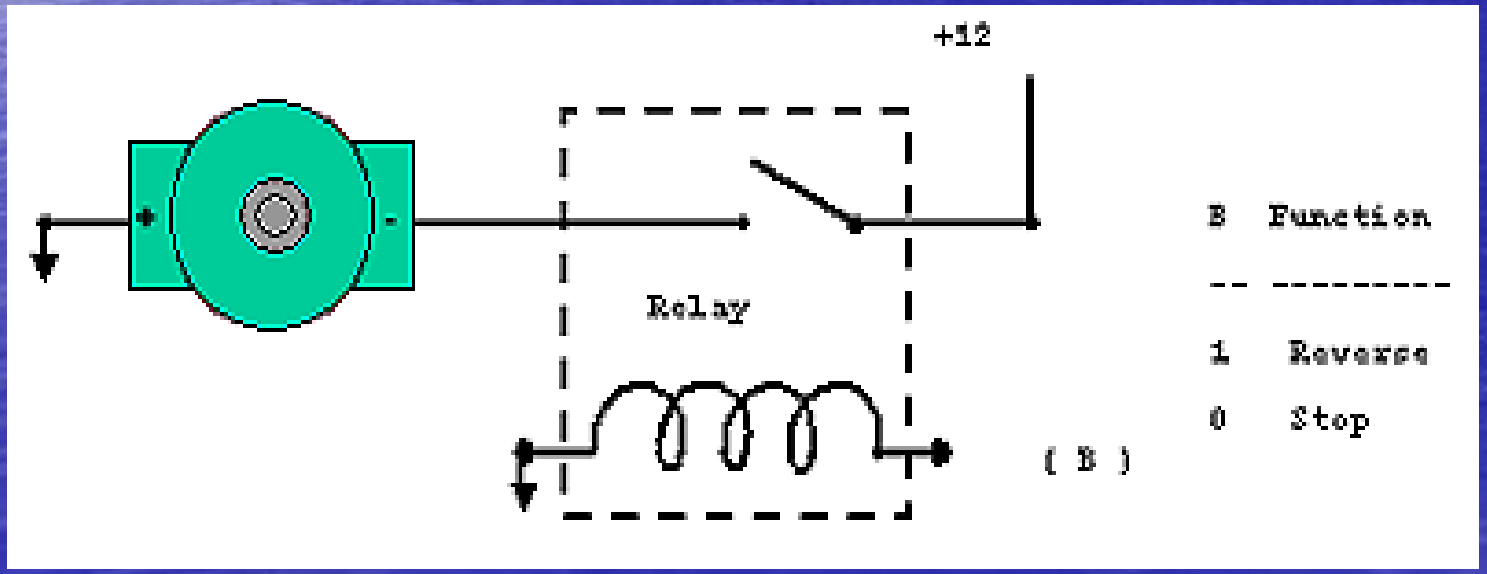


THE DIFFERENTIAL STEERING SYSTEM

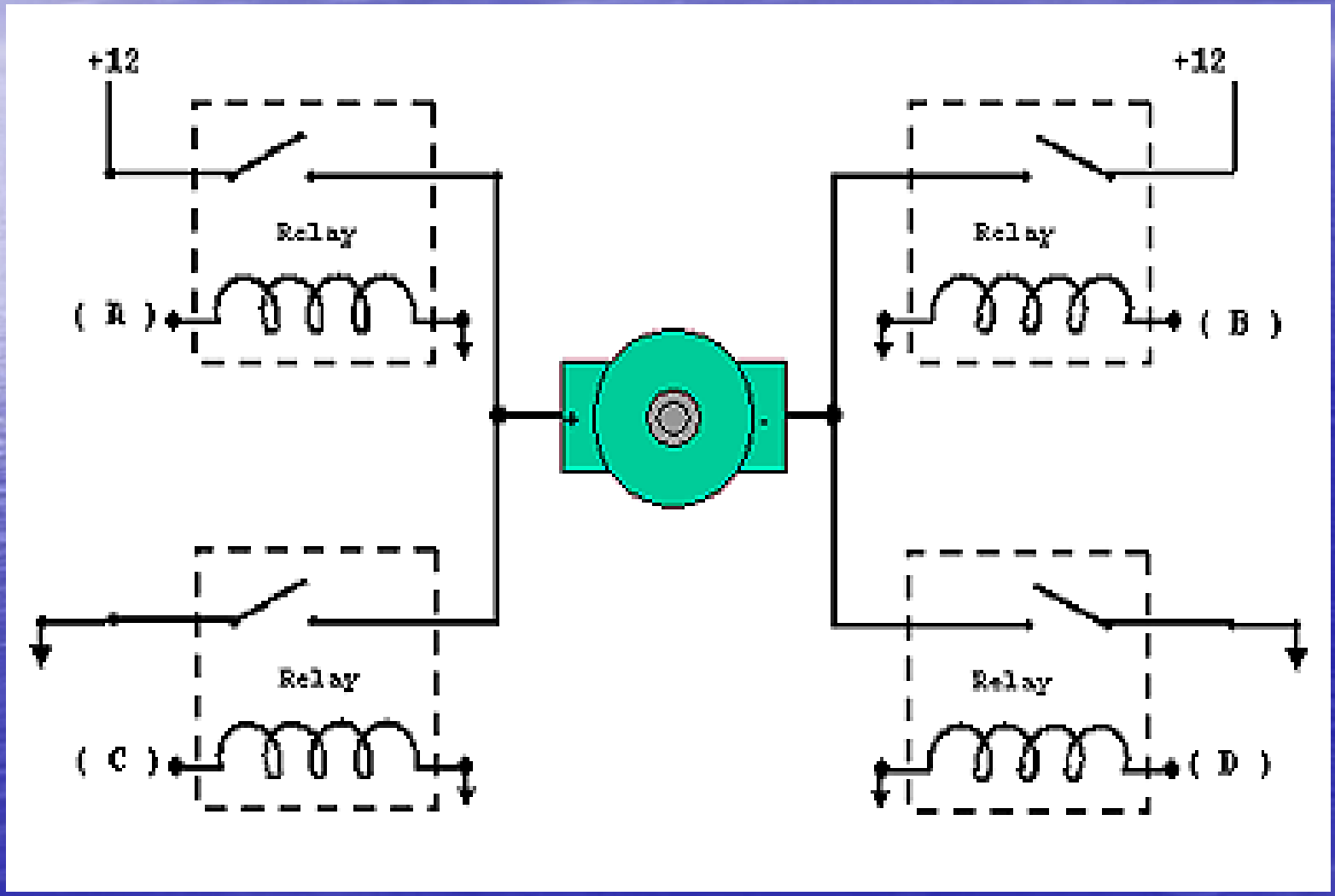


MOTOR CONTROL

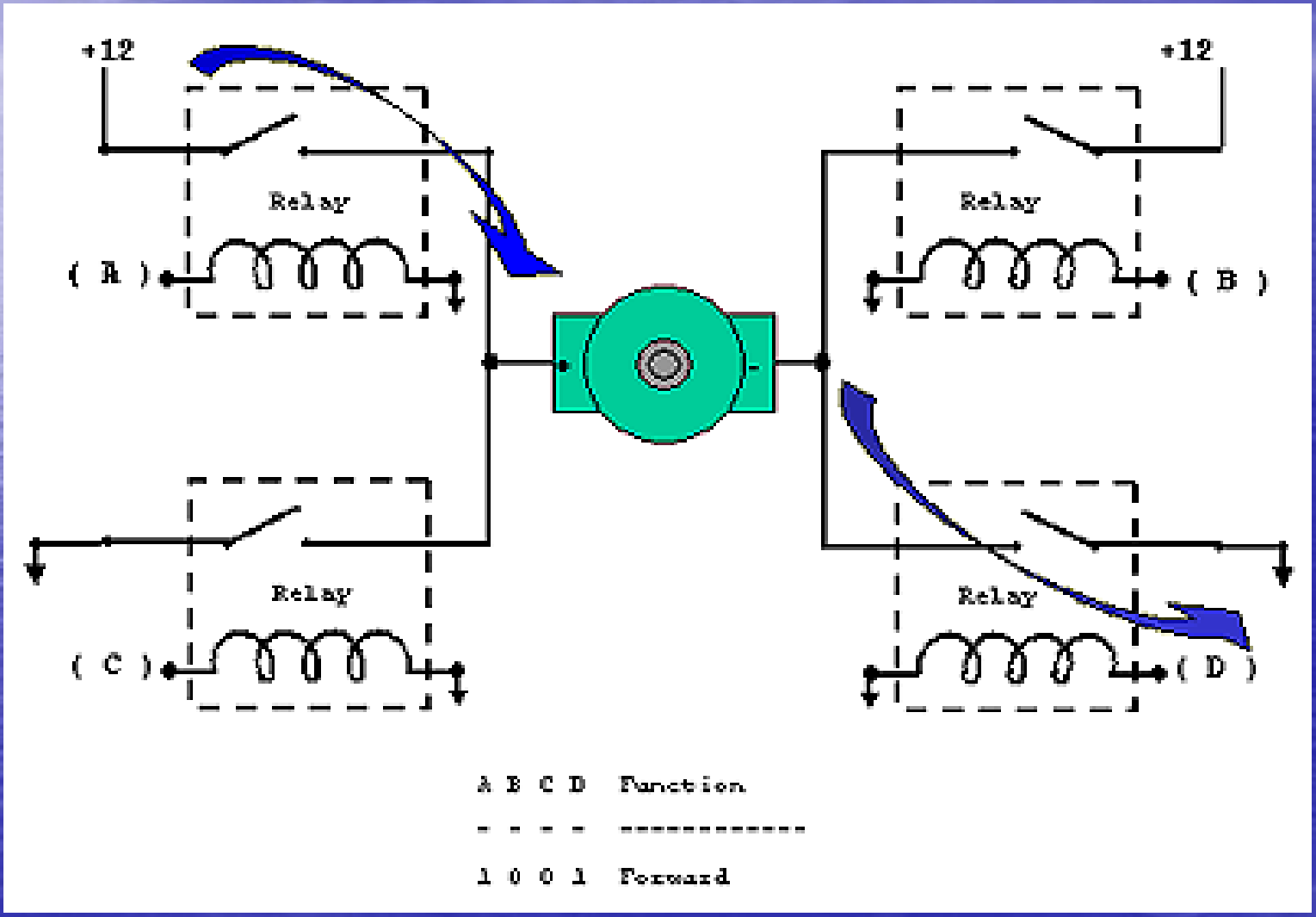




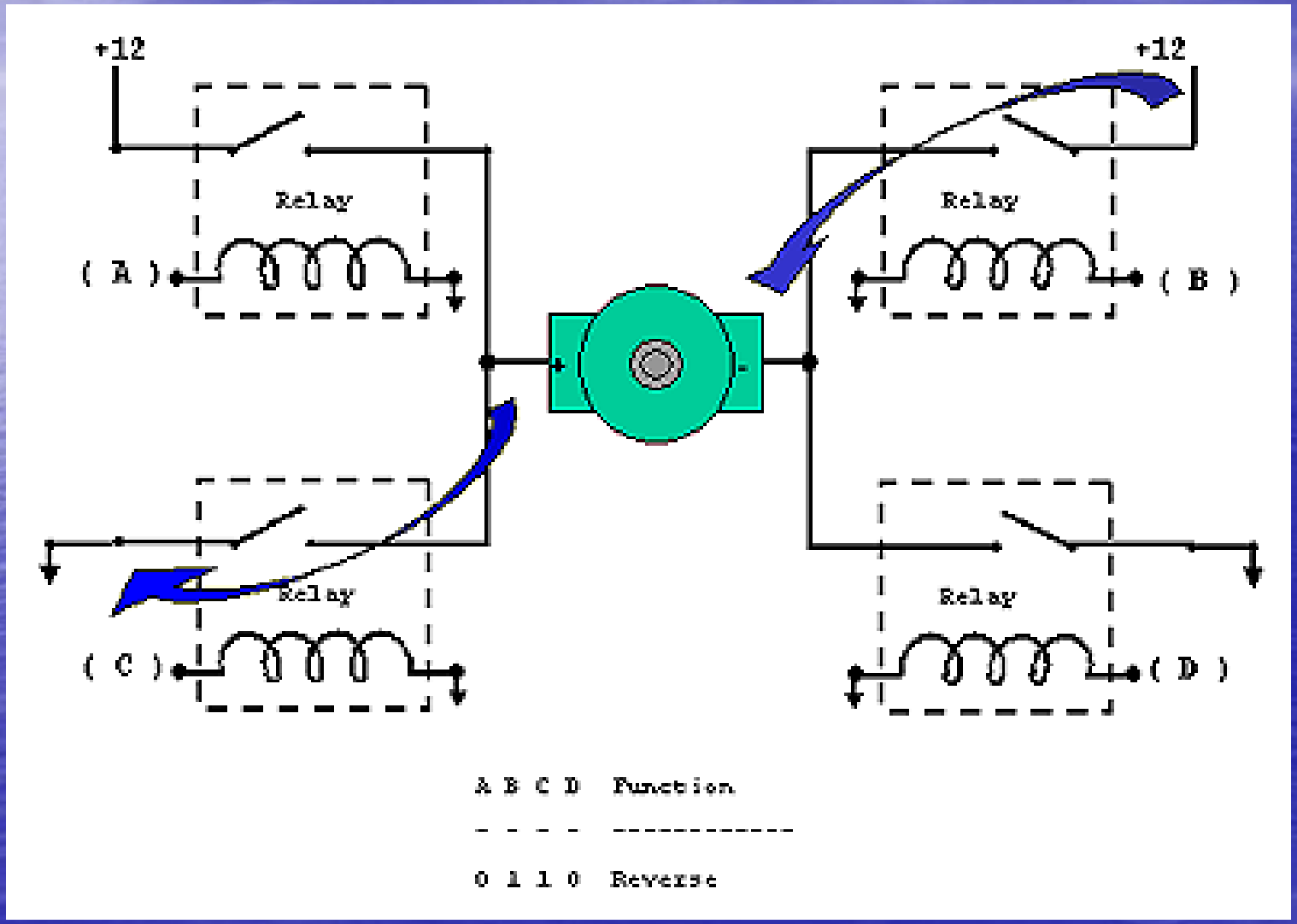
H-BRIDGE MOTOR CONTROL



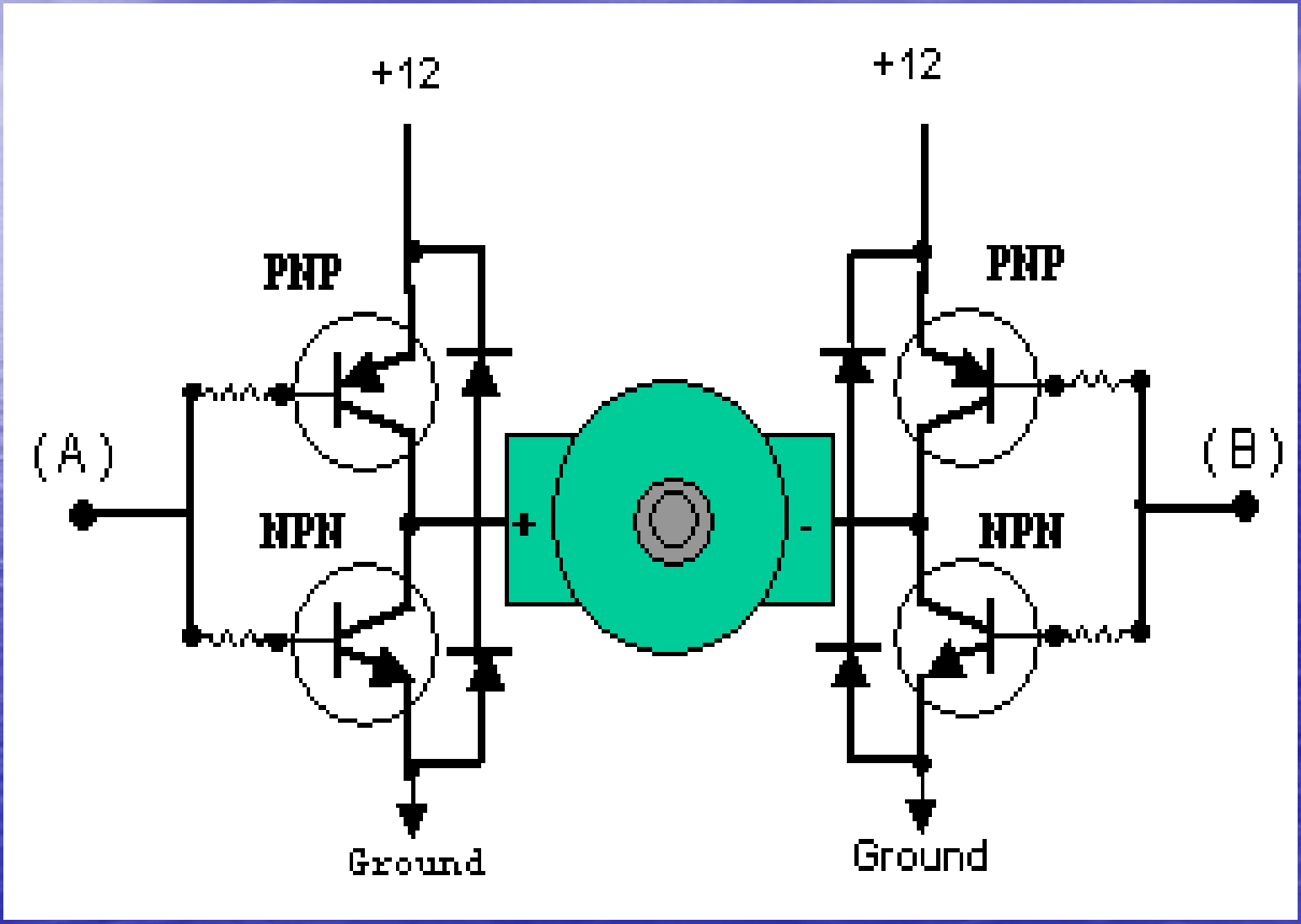
H-BRIDGE MOTOR CONTROL



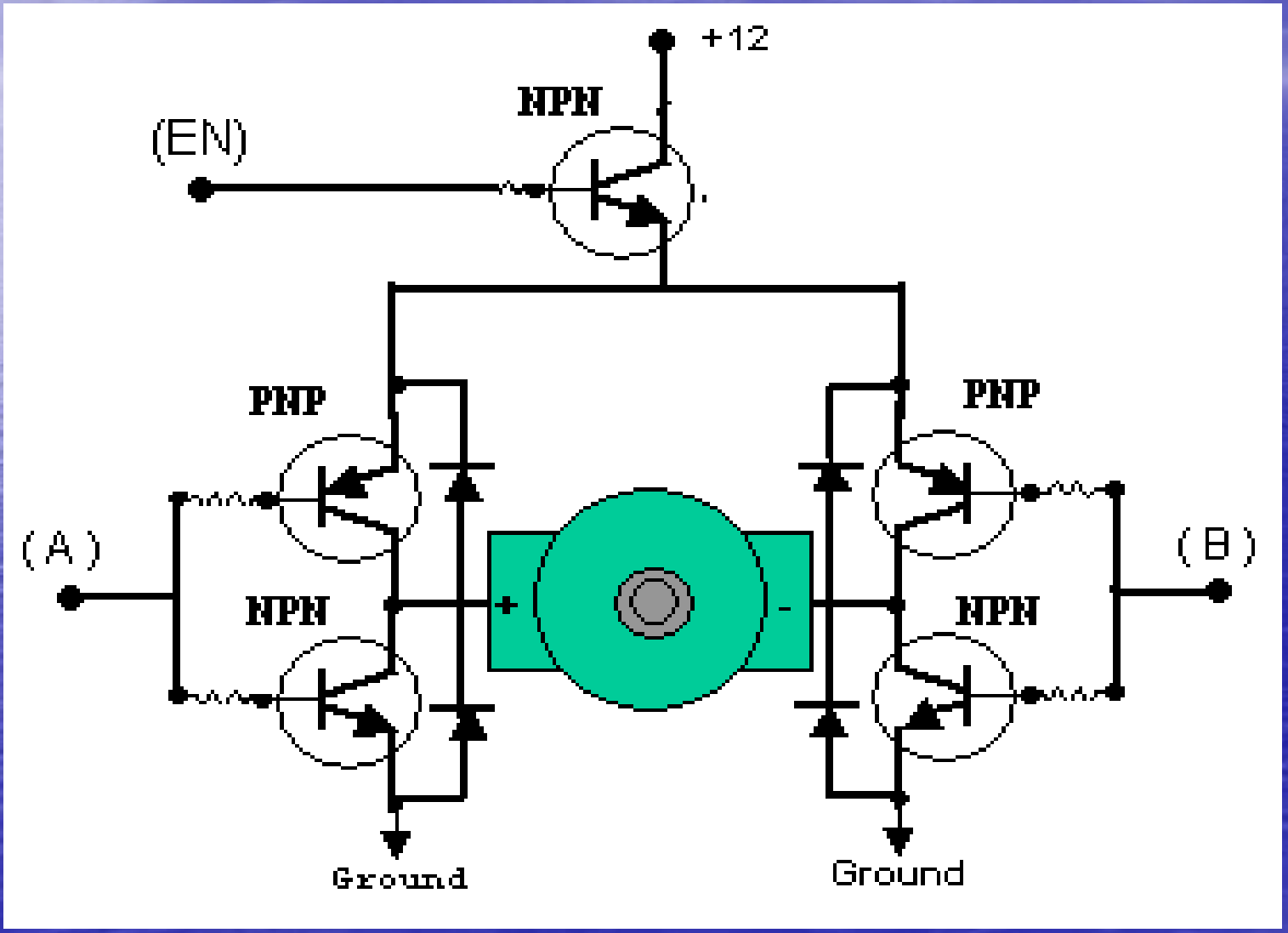
H-BRIDGE MOTOR CONTROL



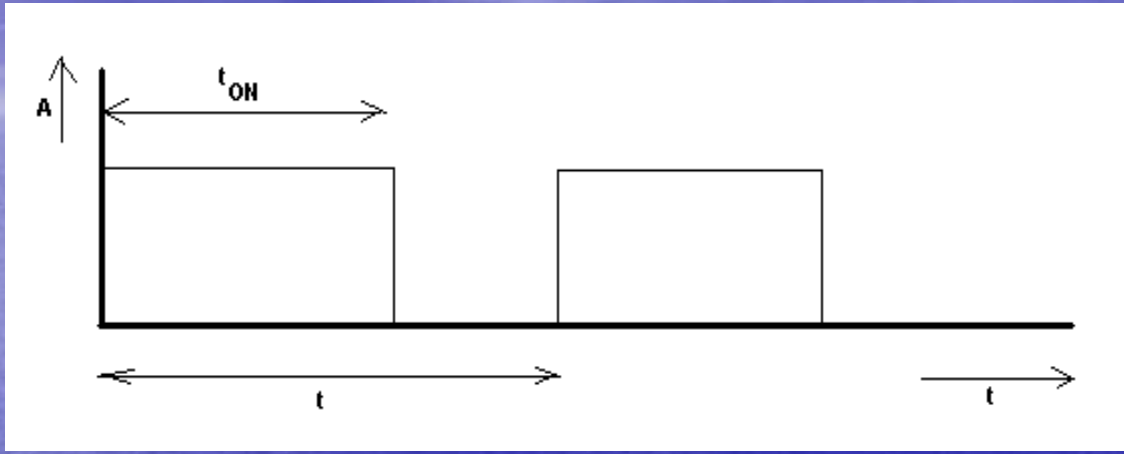
H-BRIDGE USING TRANSISTORS



H-BRIDGE WITH SPEED CONTROL



PWM SPEED CONTROL



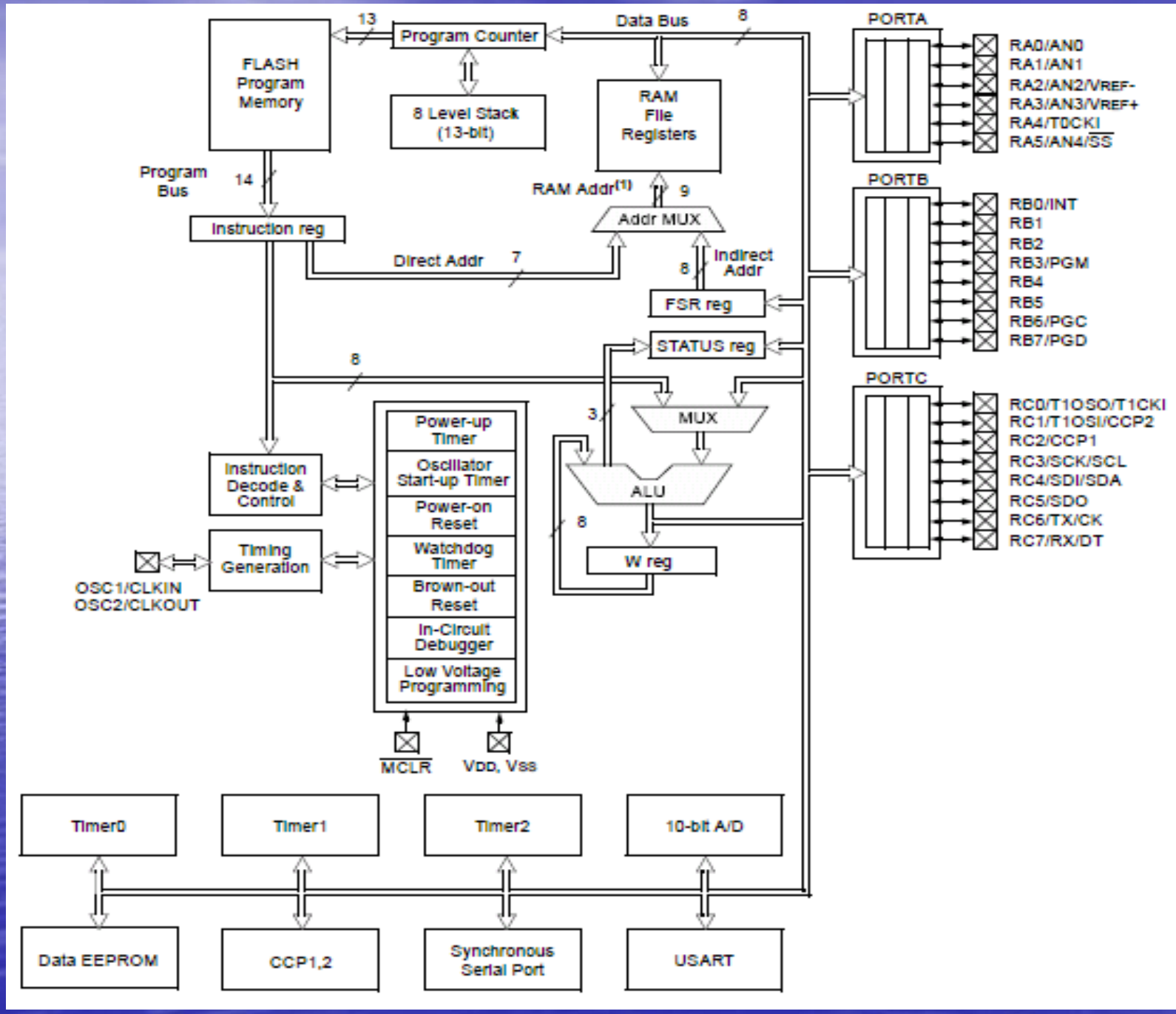
$$A = \frac{1}{t}\int v\,dt$$

$$A = \frac{1}{t}\int_{0}^{t_{ON}} +5\,dt$$

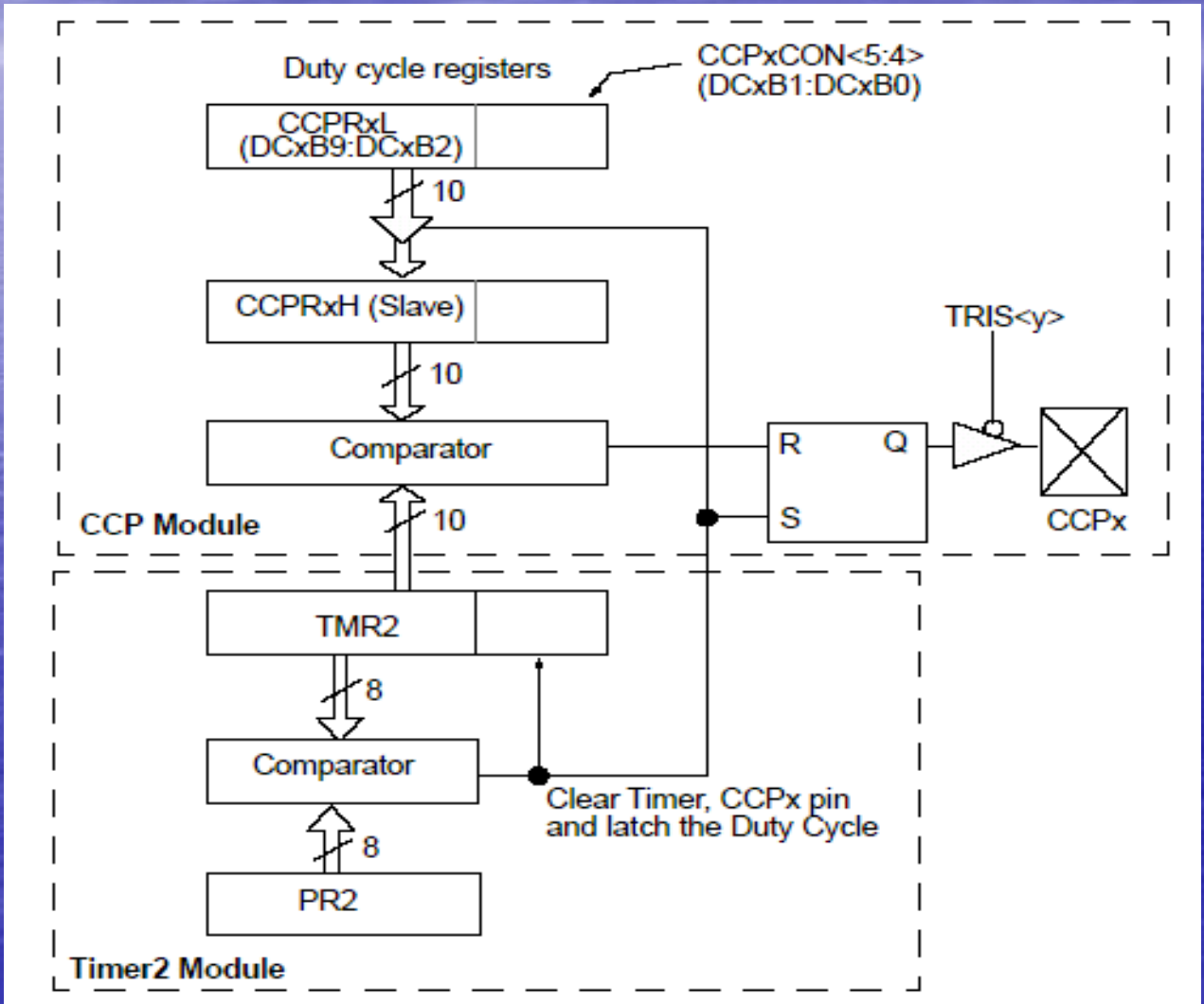
$$A = 12\frac{t_{ON}}{t} = 12V \times \rho$$

THE PIC MICRO CONTROLLER (PIC 16F873)

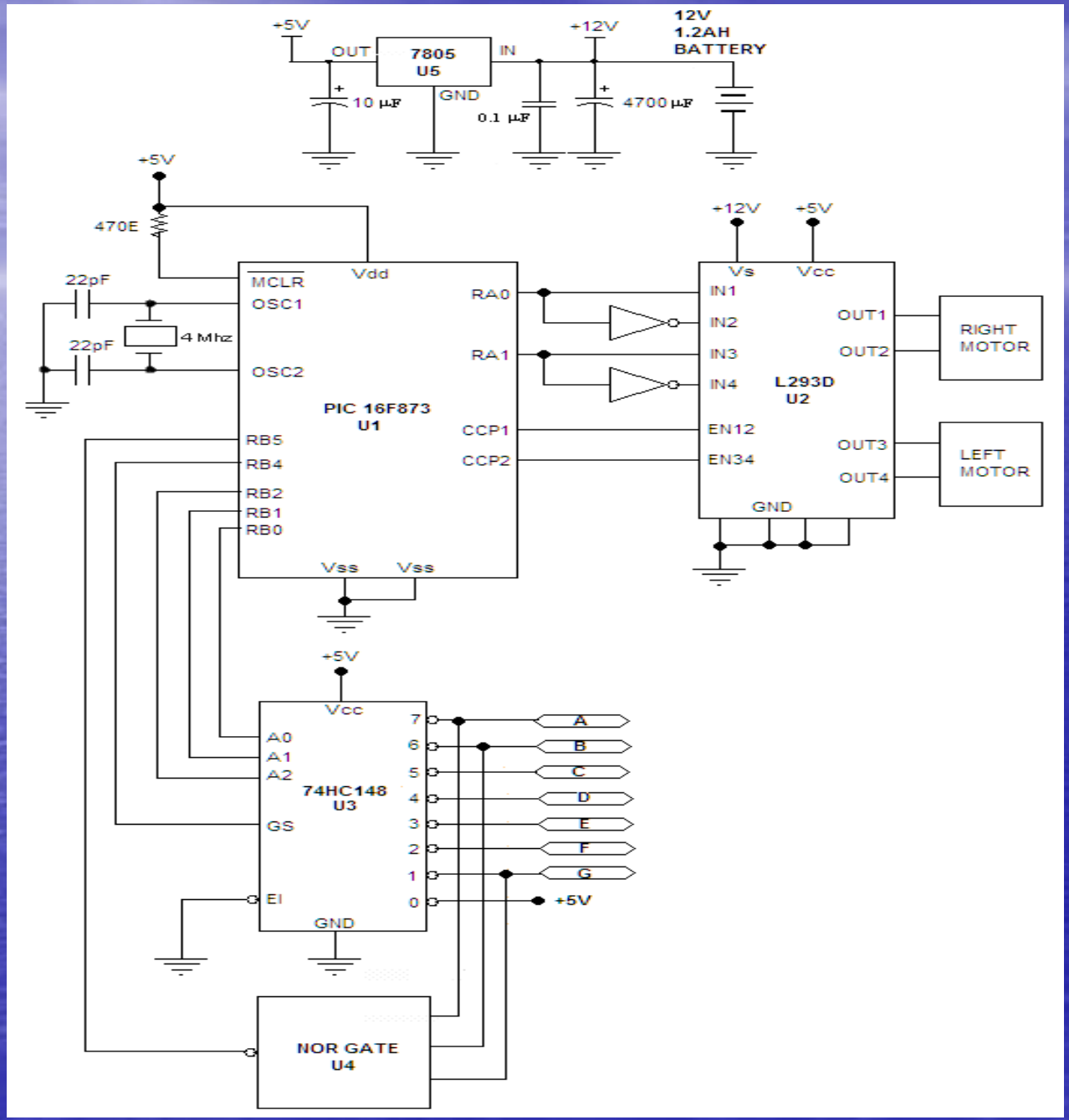
- RISC ARCHETECTURE
- 1 WORD INSTRUCTION LENGTH
- FIXED INSTRUCTION EXICUTION TIME
- 3 PORTS (A, B & C)
- 2 CCP (CAPTURE, COMPARE & PWM) MODULES
- 192 BYTES DATA RAM & 128 BYTES OF EEPROM DATA MEMORY
- UPTO 13 INTERRUPT SOURCES
- 3 TIMERS
- Power saving SLEEP mode
- A/D CONVERTOR

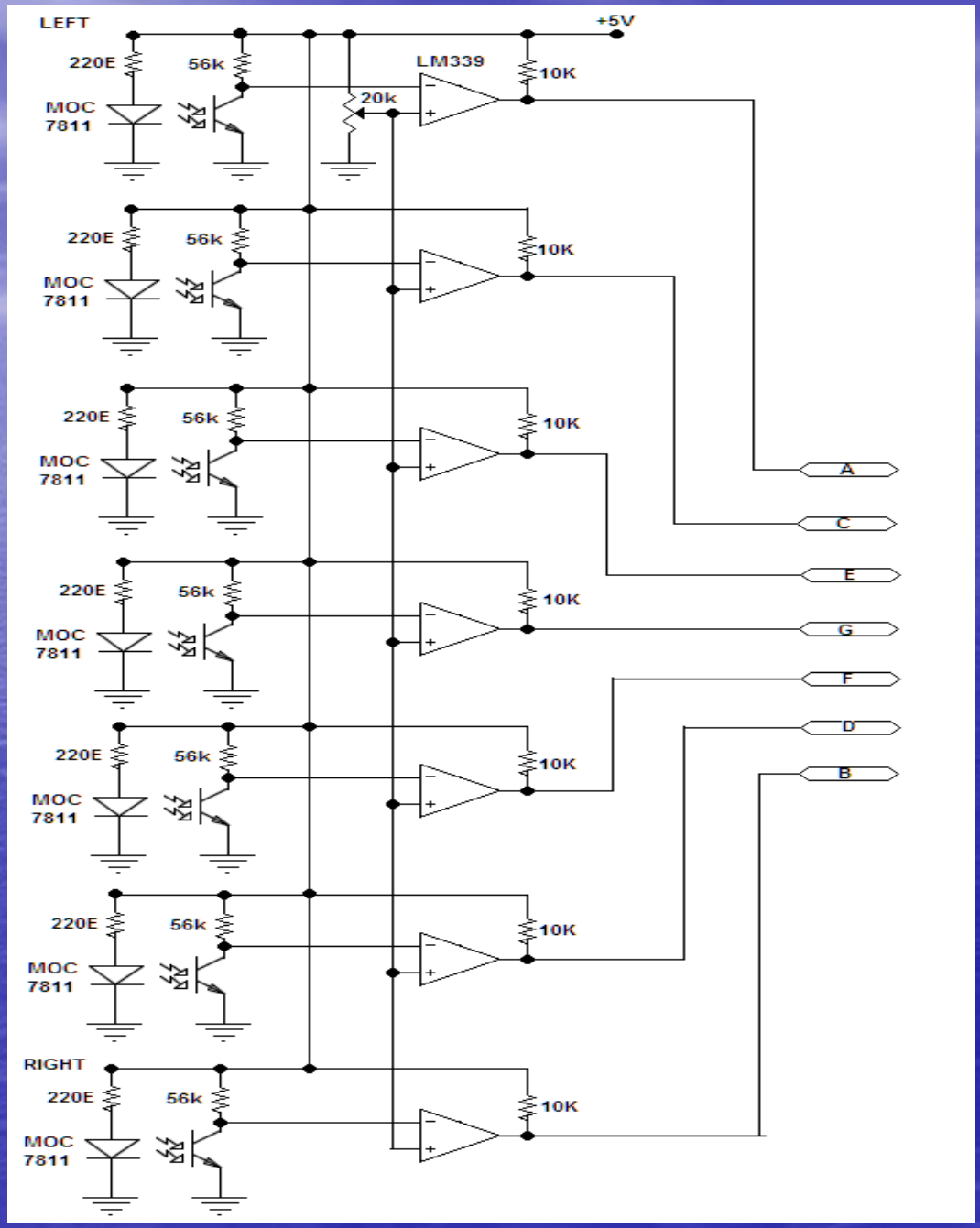

FLASH Program Memory
13
Program Counter
Data Bus
8
8 Level Stack (13-bit)
RAM File Registers
Program Bus
14
RAM Addr(1)
9
Addr MUX
Instruction reg
Direct Addr
7
8
Indirect Addr
FSR reg
STATUS reg
8
3
MUX
Instruction Decode & Control
Power-up Timer
Oscillator Start-up Timer
Power-on Reset
Watchdog Timer
Brown-out Reset
In-Circuit Debugger
Low Voltage Programming
ALU
8
W reg
Timing Generation
OSC1/CLKIN
OSC2/CLKOUT
MCLR
VDD, VSS
PORTA
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS
PORTB
RB0/INT
RB1
RB2
RB3/PGM
RB4
RB5
RB6/PGC
RB7/PGD
PORTC
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
Timer0
Timer1
Timer2
10-bit A/D
Data EEPROM
CCP1,2
Synchronous Serial Port
USART

PICMICRO CCP MODULES IN PWM MODE



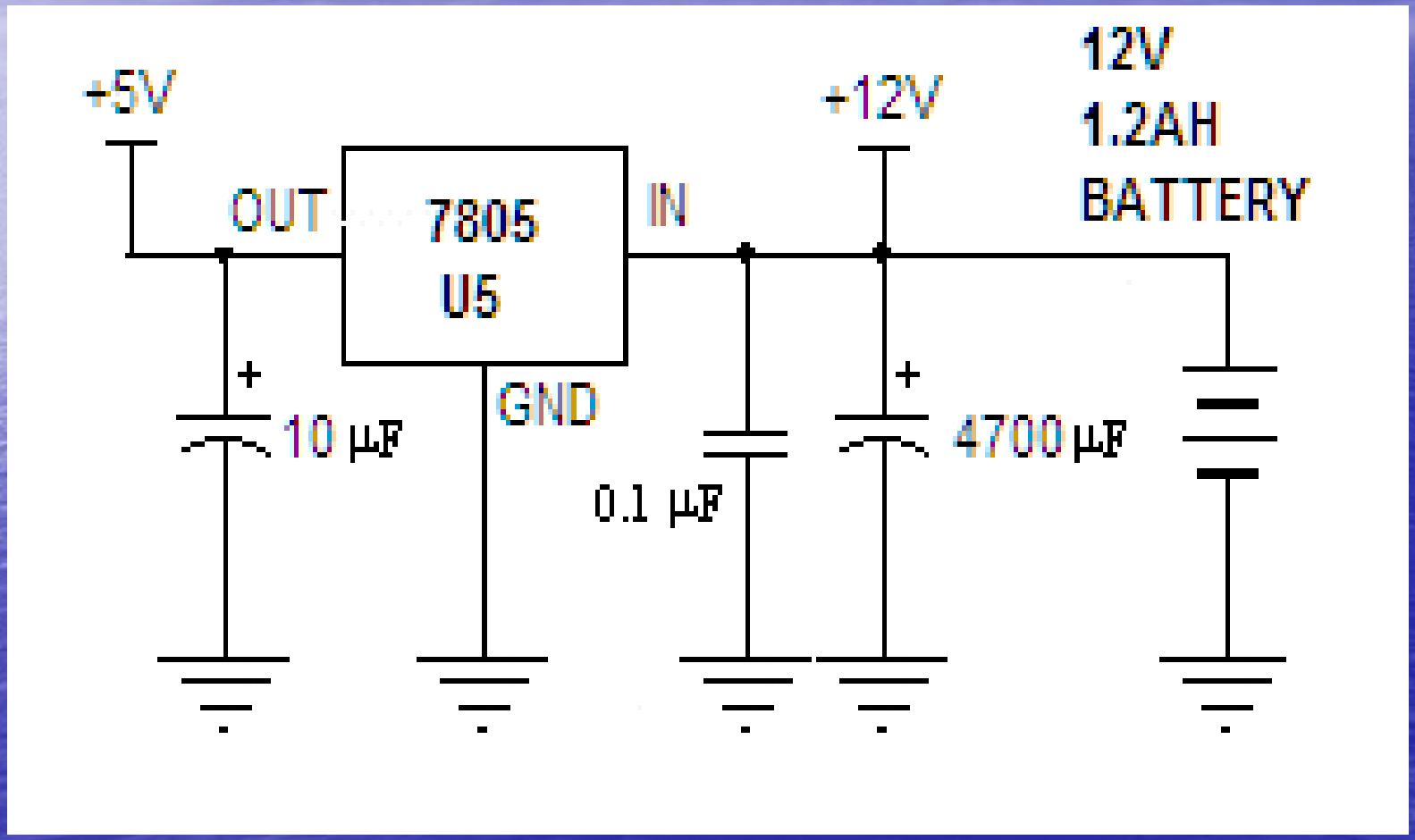
THE SCHEMATIC


+5V
OUT
7805
U5
IN
+12V
12V
1.2AH
BATTERY
GND
10 µF
0.1 µF
4700 µF
+5V
470E
+12V
+5V
22pF
MCLR
Vdd
OSC1
RA0
Vs
Vcc
IN1
IN2
OUT1
RIGHT
MOTOR
22pF
4 Mhz
IN3
OUT2
RA1
OSC2
IN4
L293D
U2
PIC 16F873
U1
CCP1
EN12
OUT3
RB5
CCP2
EN34
LEFT
MOTOR
RB4
OUT4
RB2
GND
RB1
RB0
Vss
Vss
+5V
Vcc
7
A
A0
6
B
A1
5
C
A2
74HC148
U3
4
D
GS
3
E
2
F
1
G
EI
0
+5V
GND
NOR GATE
U4


LEFT
+5V
220E
56k
MOC
7811
LM339
20k
10K
A
C
E
G
F
D
B
RIGHT

HARDWARE DESCRIPTION

THE POWER SUPPLY



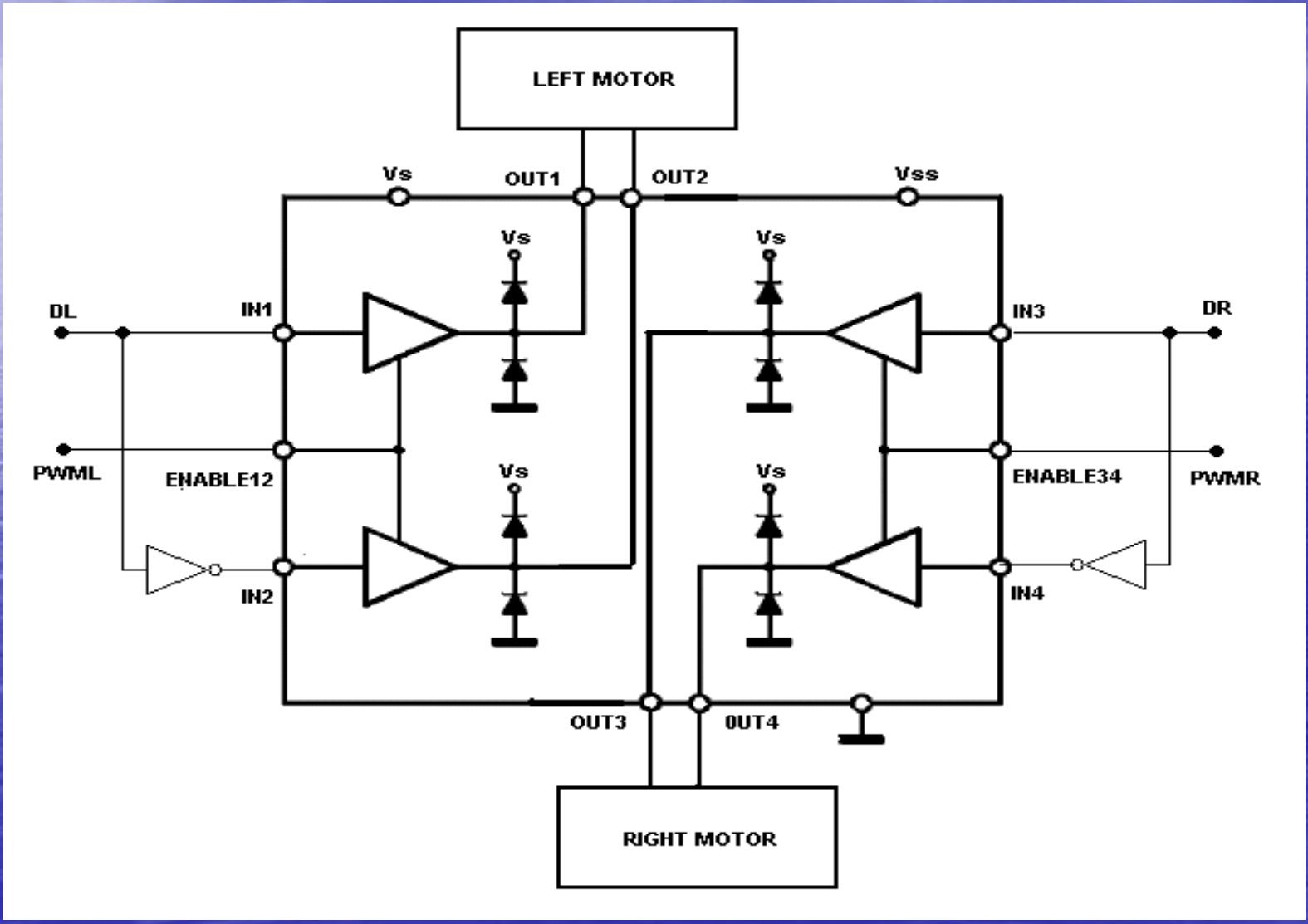
MOTORS

Motor speed = 2400 rpm @ 6v

Using gears the speed is reduced to 30 rpm @ 6v.

The motors are run at 12v, so an effective speed of 60 rpm is achieved, with a considerable increase in torque.

THE H-BRIDGE CONTROL HARDWARE

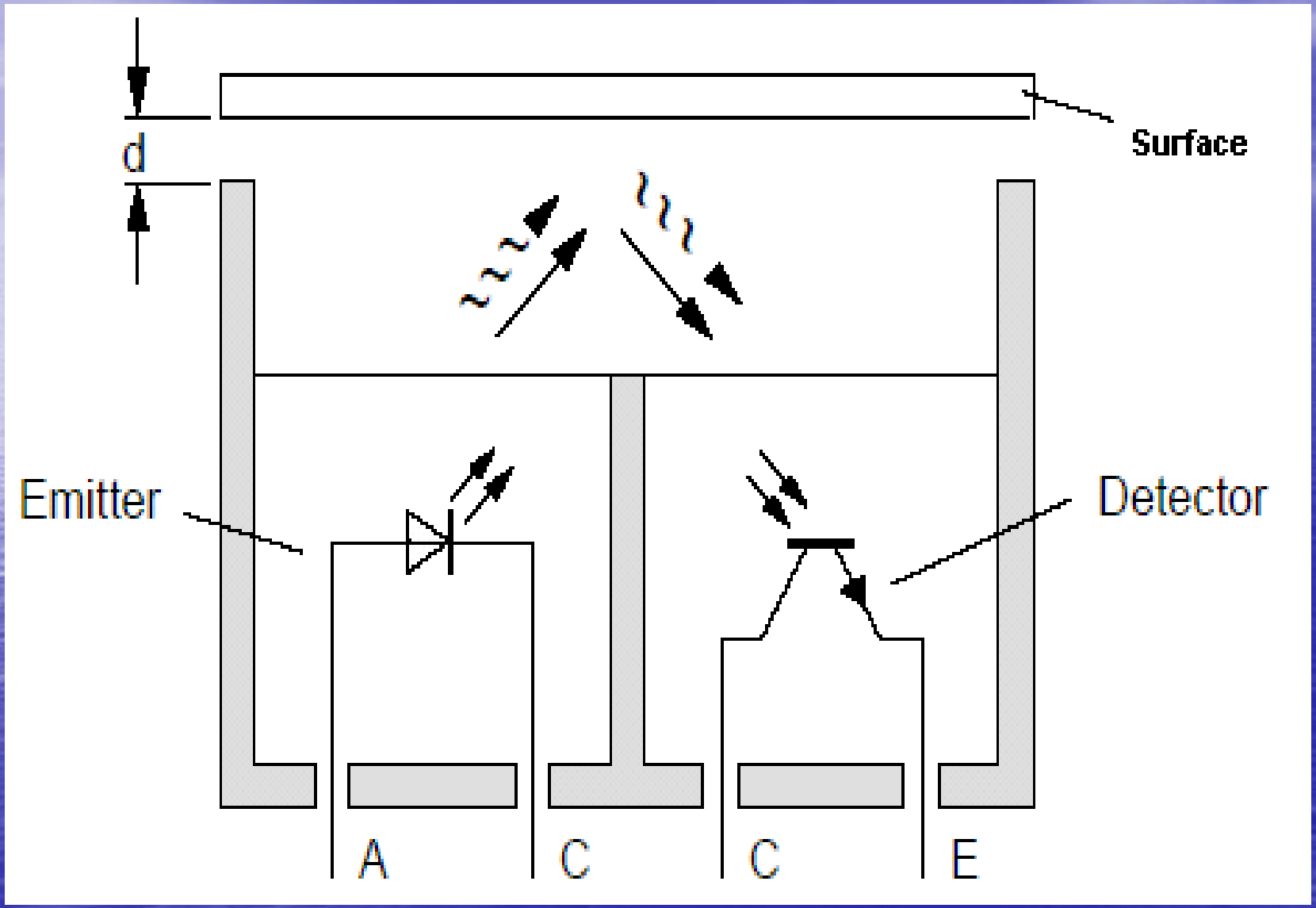


STATE TABLE

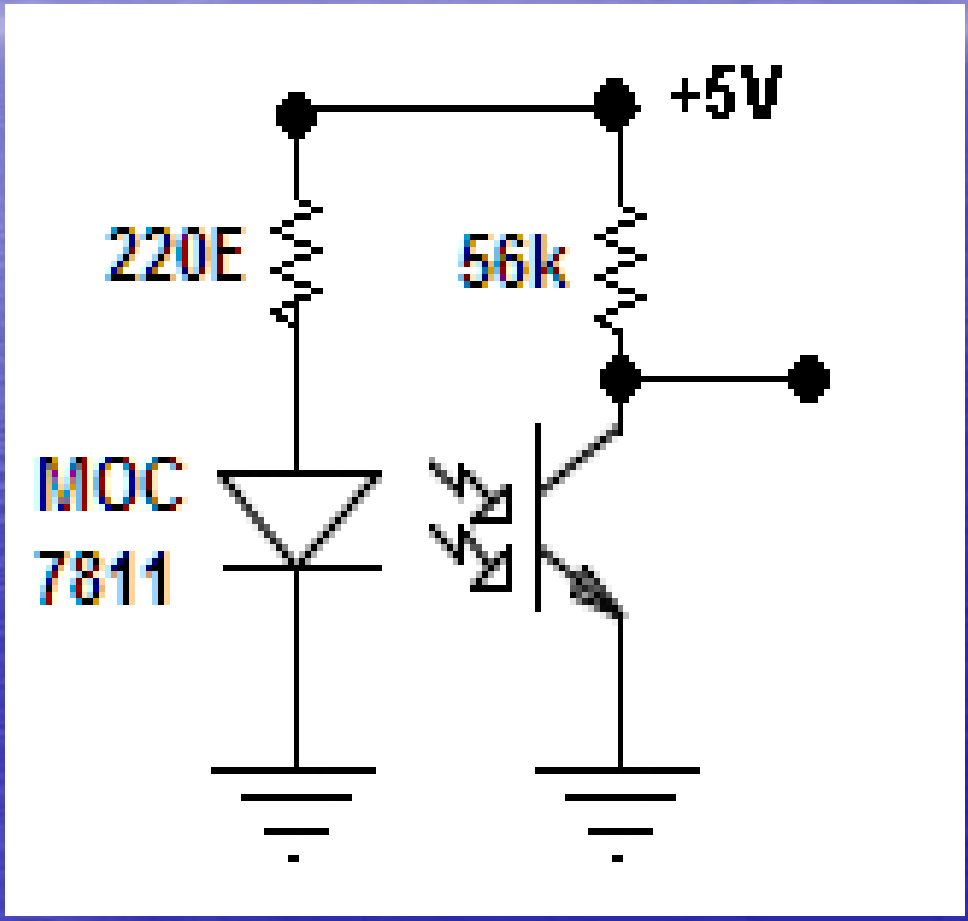
IN1	IN2	IN3	IN4	OPERATION
1	0	1	0	BOTH MOTORS FORWARD (MOVE FORWARD)
0	1	0	1	BOTH MOTORS BACKWARD (MOVE BACKWARD)
1	0	0	1	RIGHT MOTOR BACKWARD LEFT MOTOR FORWARD (TURN RIGHT)
0	1	1	0	RIGHT MOTOR FORWARD LEFT MOTOR BACKWARD (TURN LEFT)

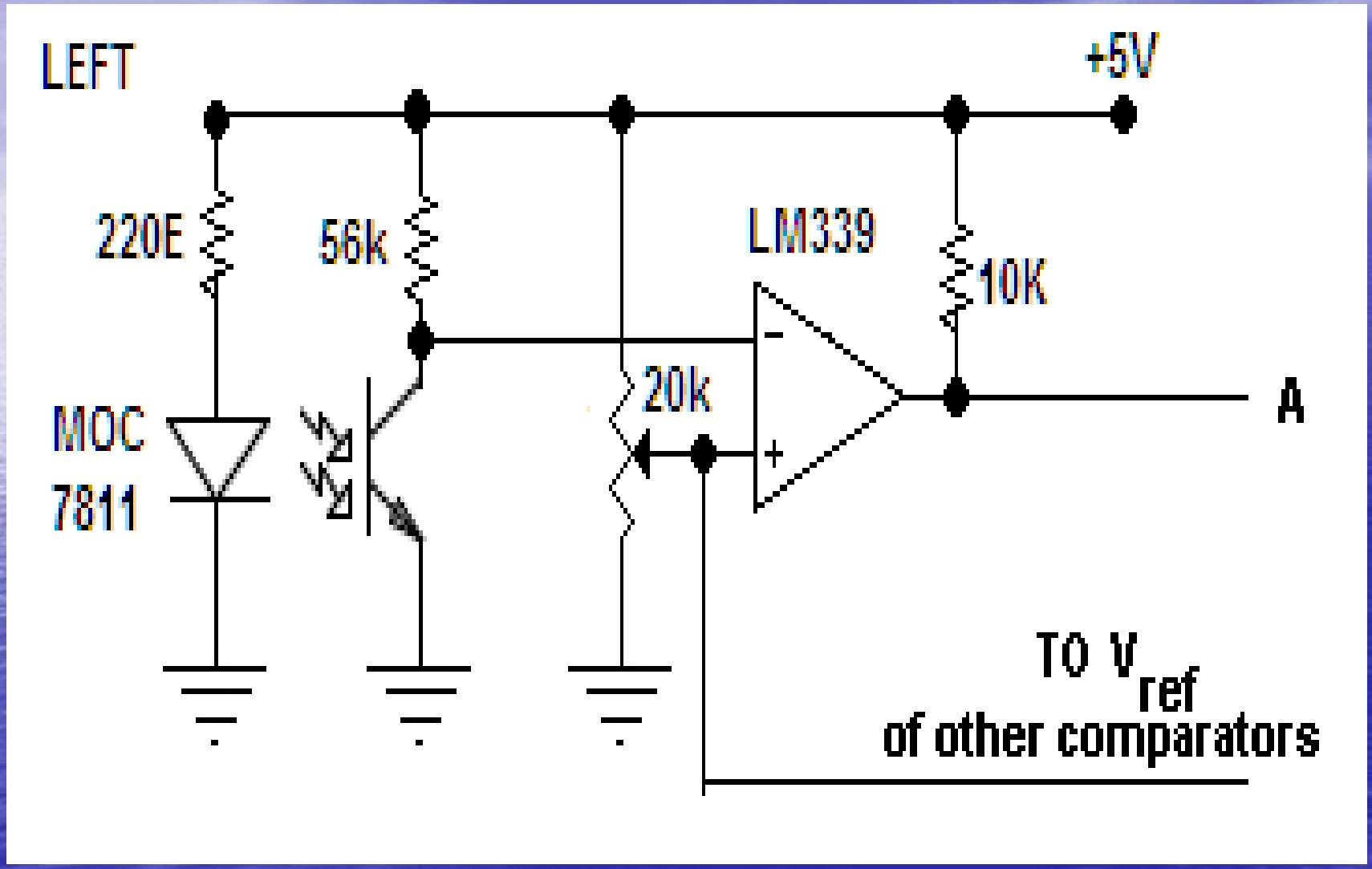
THE IR SENSORS

- Interrupter sensor modified to be a reflective sensor
- ~950nm wavelength
- Lens fitted to emitter and detector of focal length of 4mm

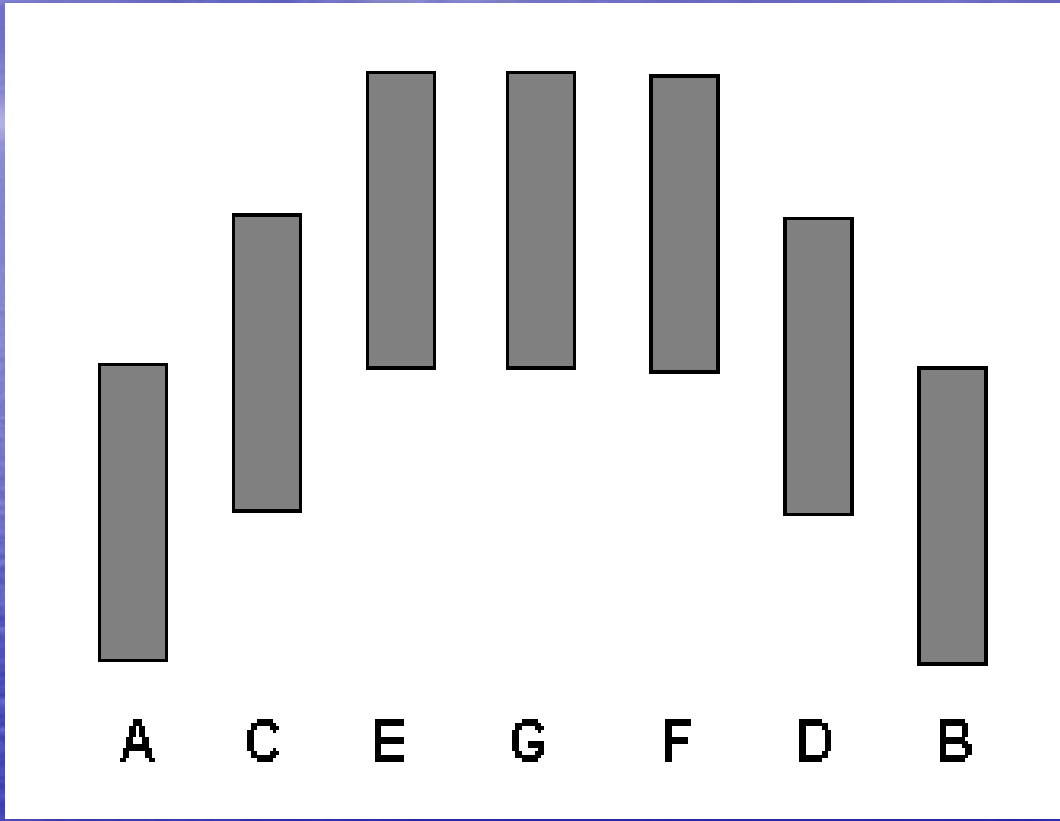

d
Surface
Emitter
Detector
A
C
C
E

SENSOR CIRCUIT



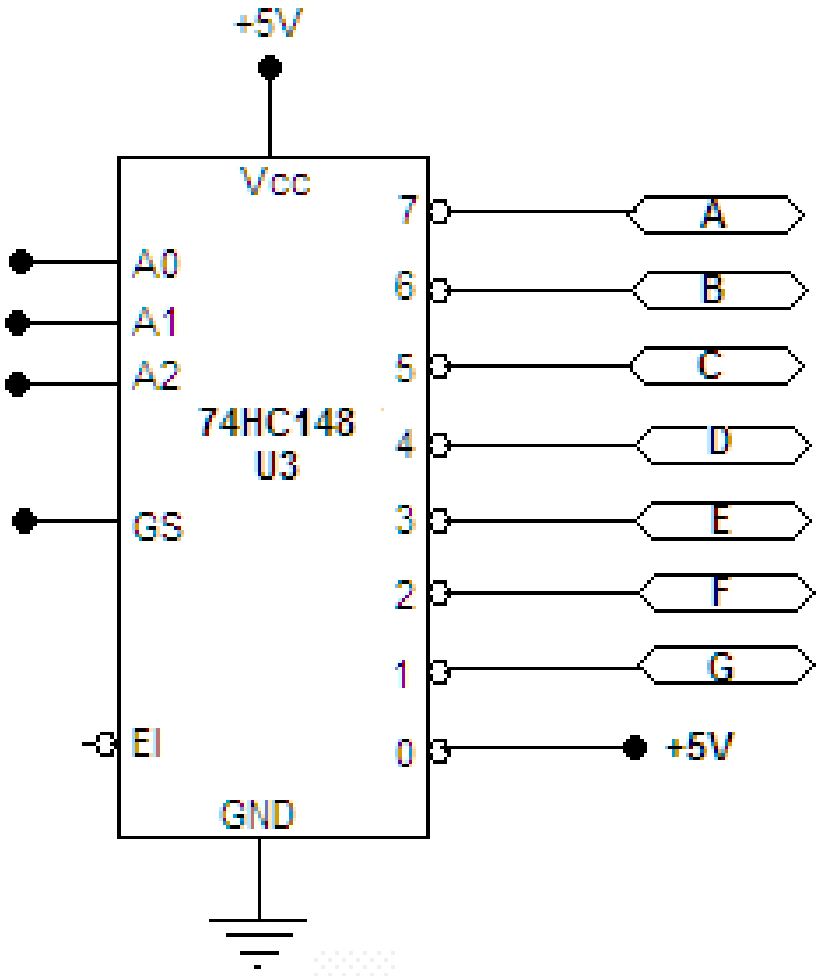

LEFT
+5V
220E
56k
LM339
10K
20k
MOC
7811
A
TO V ref
of other comparators

SENSOR ARRAY



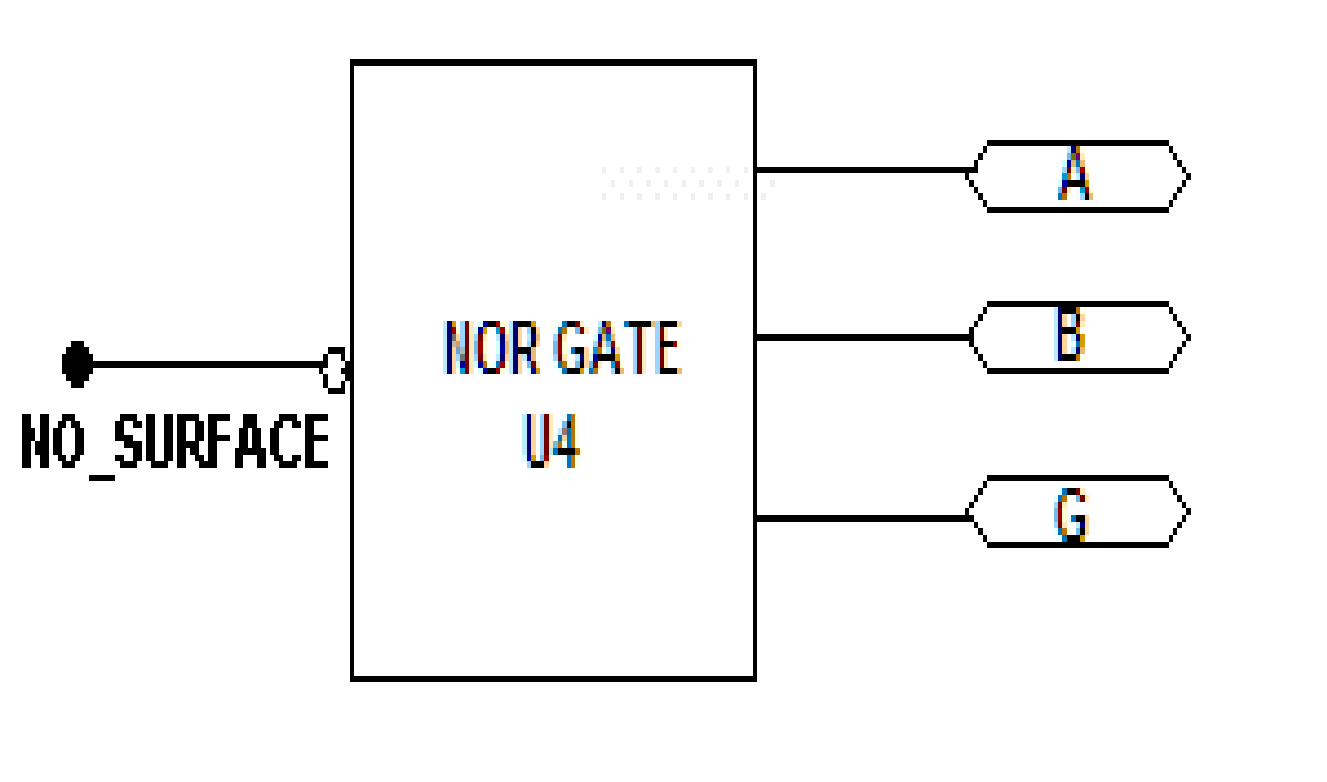
MINIMUM DISTANCE BETWEEN SENSORS IS 1cm

THE PRIORITY ENCODER



Inputs									Outputs				
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

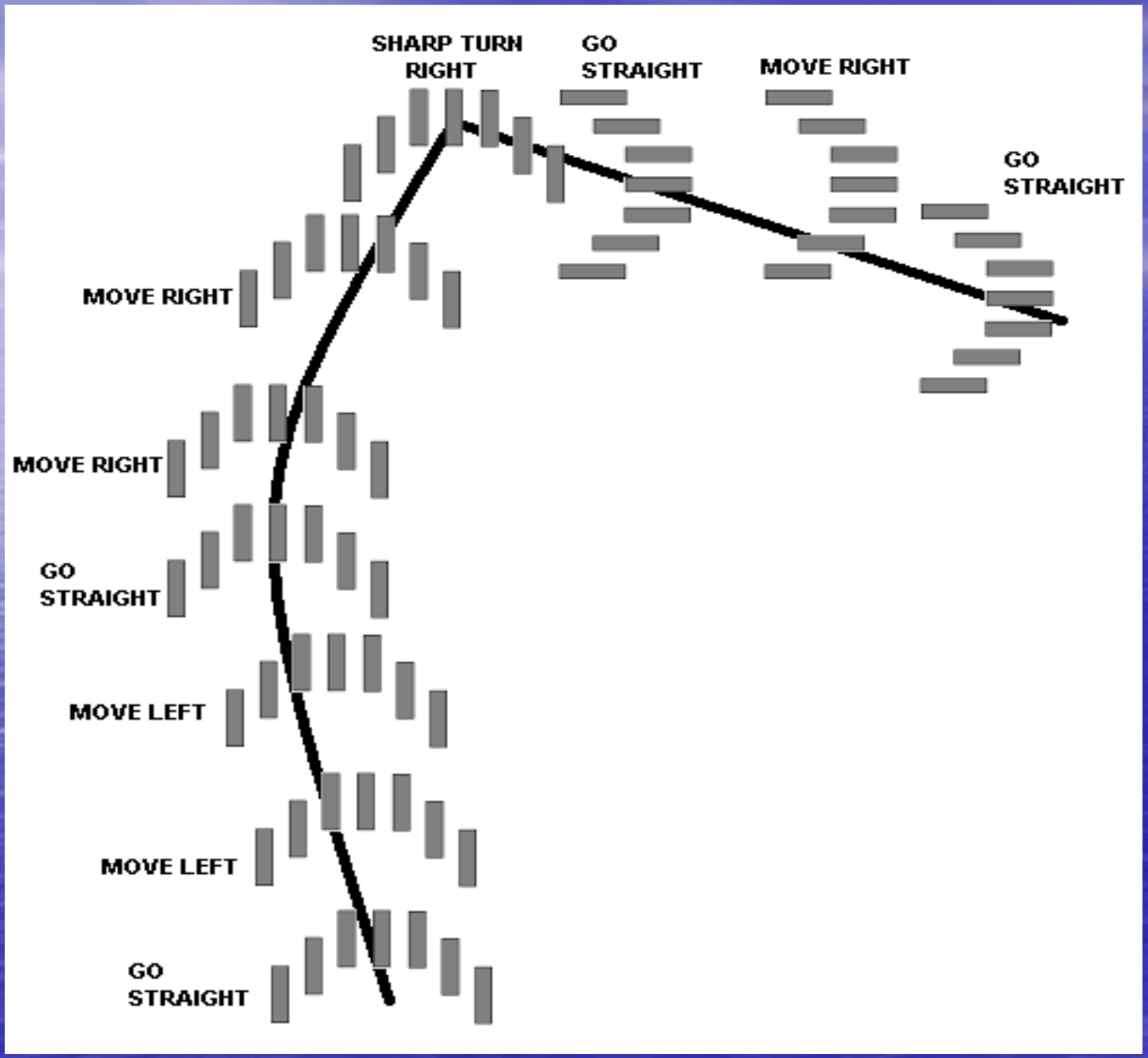
THE NO SURFACE LOGIC



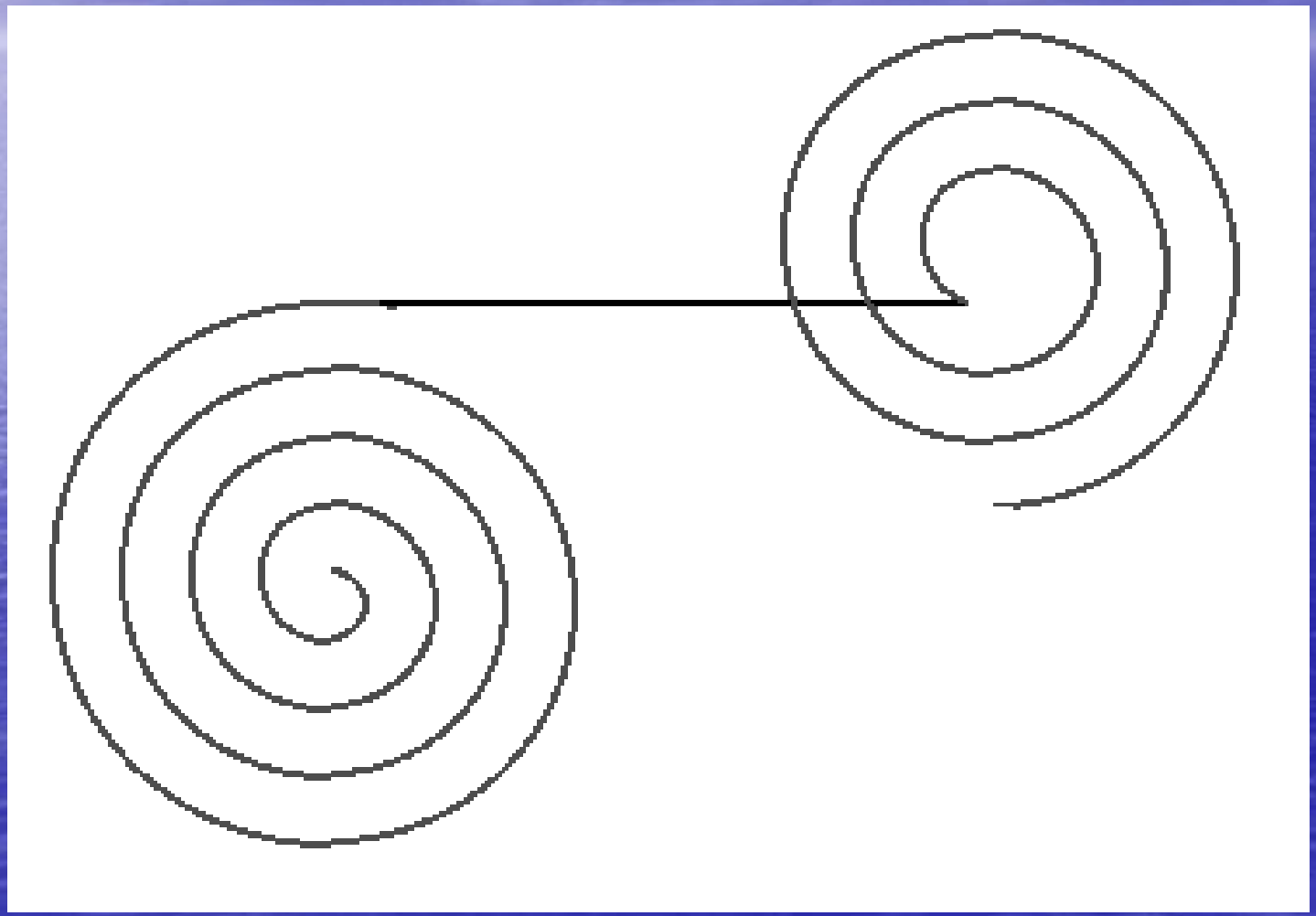
INPUTS TO THE MICROCONTROLLER

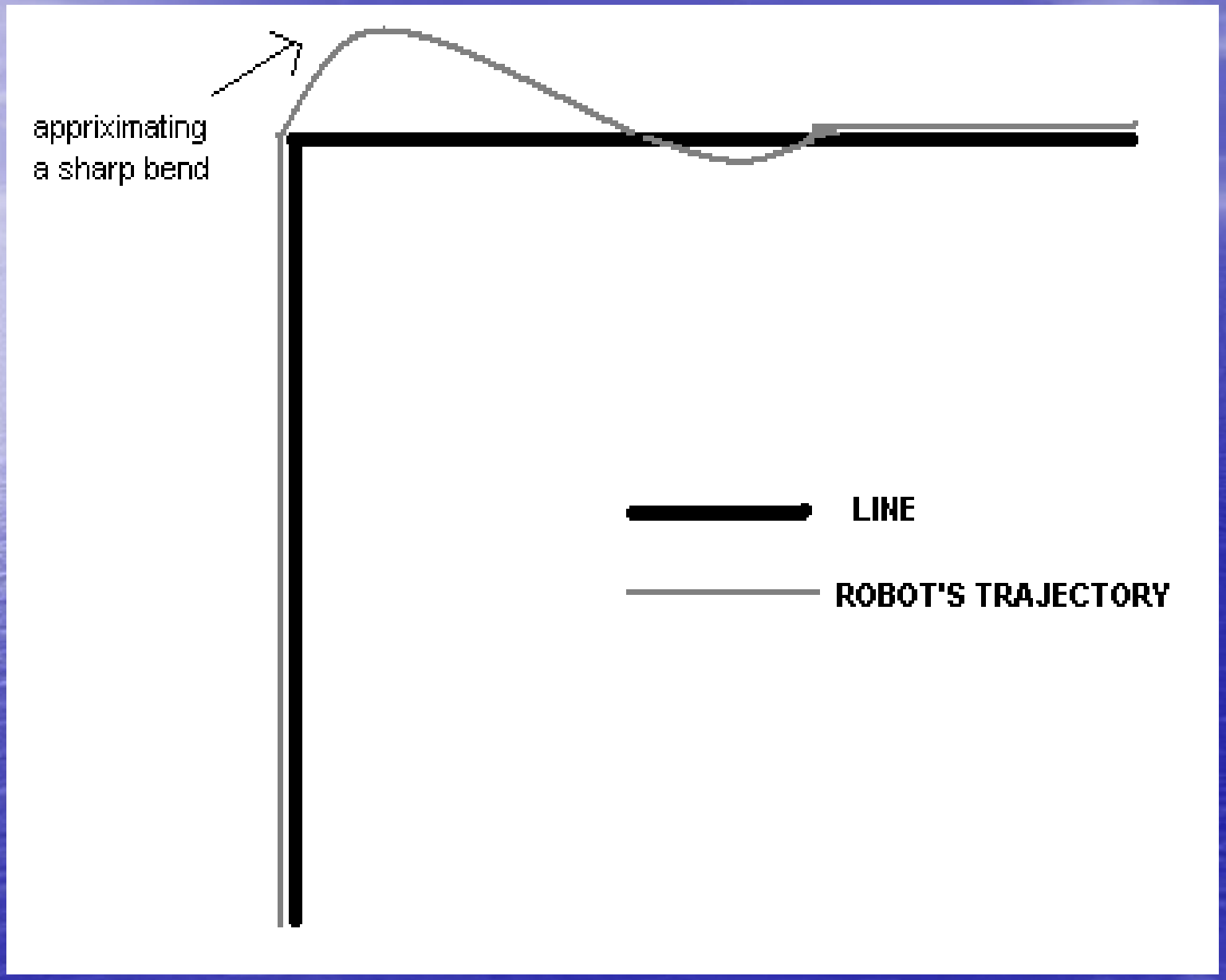
NS	GS	A2	A1	A0	STATE IN	ACTION
1	X	X	X	X	No surface is detected	Stop the motors
0	1	X	X	X	No line is detected	Execute the no line code (specially designed algorithm)
0	0	0	0	0	A detects the line	Sharp turn left
0	0	0	0	1	B detects the line	Sharp turn right
0	0	0	1	0	C detects the line	Turn left
0	0	0	1	1	D detects the line	Turn right
0	0	1	0	0	E detects the line	Move left
0	0	1	0	1	F detects the line	Move right
0	0	1	1	0	G detects the line	Go straight
0	0	1	1	1	Forbidden state	Software reset the processor

PROCESSES INVOLVED

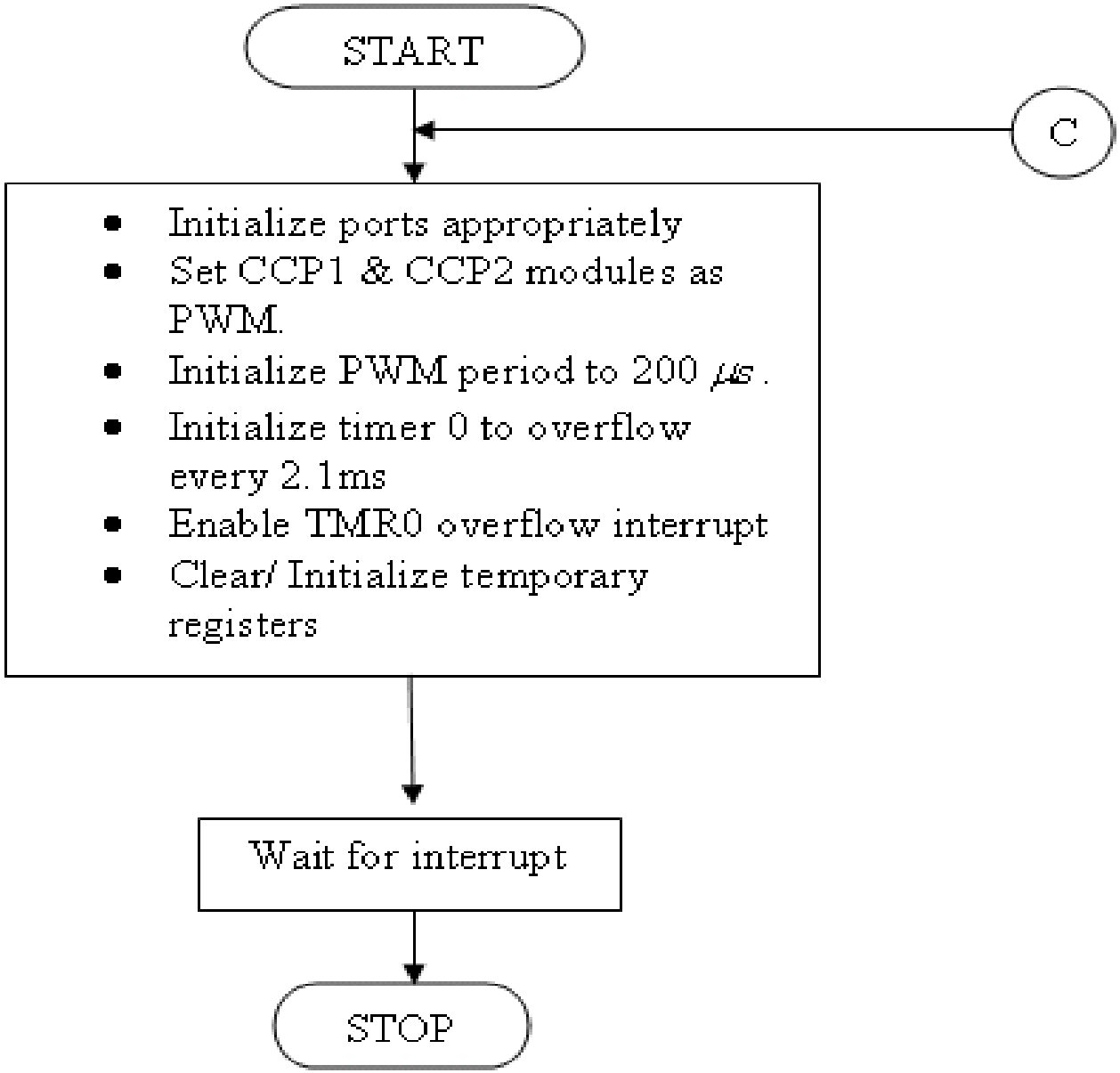

SHARP TURN
RIGHT
GO
STRAIGHT
MOVE RIGHT
GO
STRAIGHT
MOVE RIGHT
MOVE RIGHT
GO
STRAIGHT
MOVE LEFT
MOVE LEFT
GO
STRAIGHT

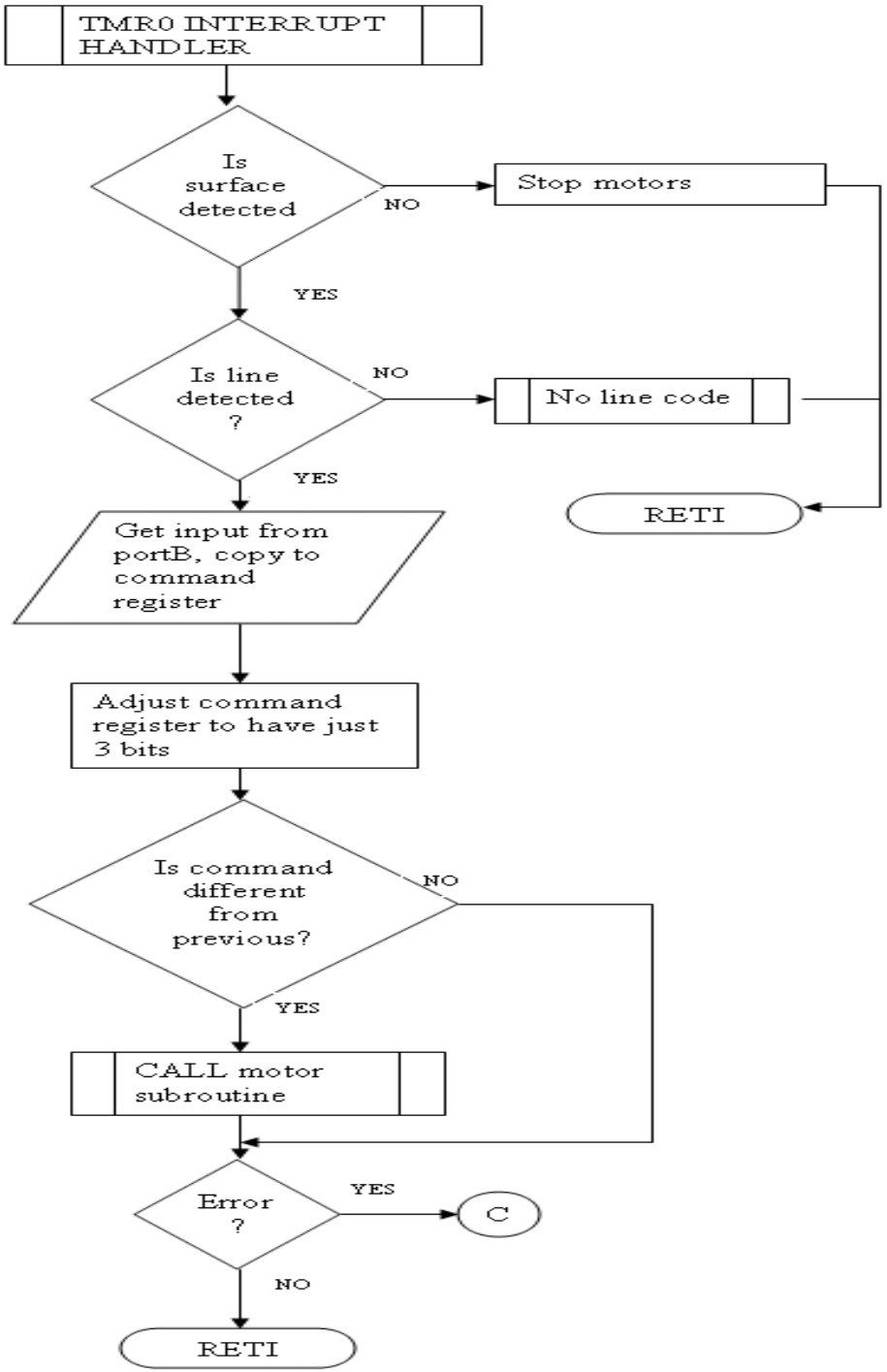
LINE FIND MODE

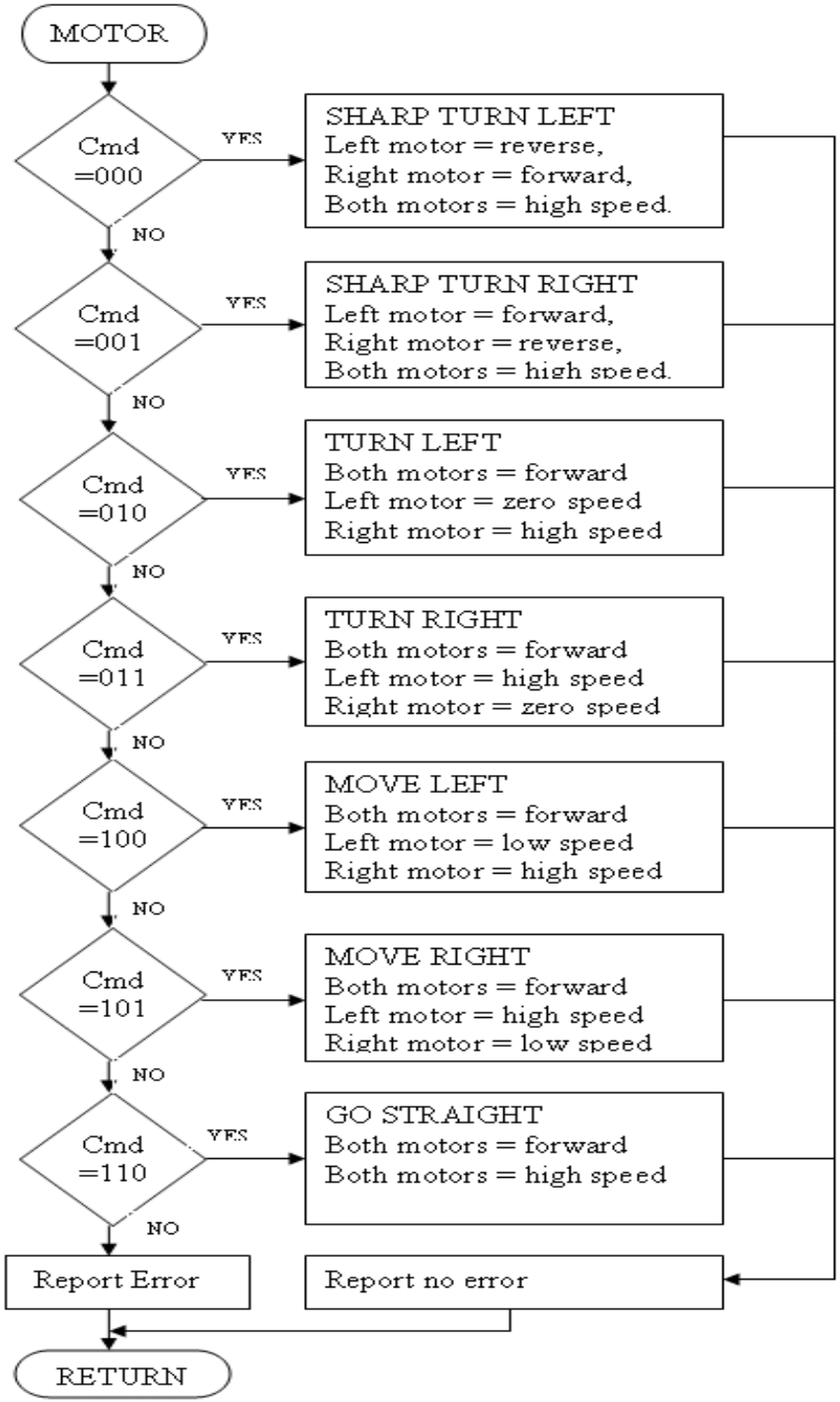



appriximating
a sharp bend
LINE
ROBOT'S TRAJECTORY

FLOW CHART


START
C
Initialize ports appropriately
Set CCP1 & CCP2 modules as PWM.
Initialize PWM period to 200 μs.
Initialize timer 0 to overflow every 2.1ms
Enable TMR0 overflow interrupt
Clear/ Initialize temporary registers
Wait for interrupt
STOP


TMR0 INTERRUPT HANDLER
Is surface detected
NO
Stop motors
YES
Is line detected ?
NO
No line code
YES
RETI
Get input from portB, copy to command register
Adjust command register to have just 3 bits
Is command different from previous?
NO
YES
CALL motor subroutine
Error ?
YES
C
NO
RETI


MOTOR
Cmd =000
YES
SHARP TURN LEFT
Left motor = reverse,
Right motor = forward,
Both motors = high speed.
NO
Cmd =001
YES
SHARP TURN RIGHT
Left motor = forward,
Right motor = reverse,
Both motors = high speed.
NO
Cmd =010
YES
TURN LEFT
Both motors = forward
Left motor = zero speed
Right motor = high speed
NO
Cmd =011
YES
TURN RIGHT
Both motors = forward
Left motor = high speed
Right motor = zero speed
NO
Cmd =100
YES
MOVE LEFT
Both motors = forward
Left motor = low speed
Right motor = high speed
NO
Cmd =101
YES
MOVE RIGHT
Both motors = forward
Left motor = high speed
Right motor = low speed
NO
Cmd =110
YES
GO STRAIGHT
Both motors = forward
Both motors = high speed
NO
Report Error
Report no error
RETURN

APPLICATIONS

- Industrial automated equipment carriers
- Automated cars.
- Tour guides in museums and other similar applications.
- Second wave robotic reconnaissance operations.

LIMITATIONS

- Choice of line is made in the hardware abstraction and cannot be changed by software.
- Calibration is difficult, and it is not easy to set a perfect value.
- The steering mechanism is not easily implemented in huge vehicles and impossible for non-electric vehicles (petrol powered).
- Few curves are not made efficiently, and must be avoided.

…LIMITATIONS

- Lack of a four wheel drive, makes it not suitable for a rough terrain.
- Use of IR even though solves a lot of problems pertaining to interference, makes it hard to debug a faulty sensor.
- Lack of speed control makes the robot unstable at times.

FUTURE SCOPE

- Software control of the line type (dark or light) to make automatic detection possible.
- “Obstacle detecting sensors” to avoid physical obstacles and continue on the line.
- Distance sensing and position logging & transmission.

RESULT AND CONCLUSION

- The robot follows a line as demonstrated.
- It effectively overcomes problems such as “barren land syndrome” and line breaks.
- The hardware and software works as designed.

BIBLIOGRAPHY

- Programming and Customizing the PIC microcontroller by *Myke Predko*
- PICmicro Mid-Range MCU Family Reference Manual by *MICROCHIP*
- PIC Robotics, A beginner's guide to robotics projects using the PICmicro by *John Iovine*

…BIBLIOGRAPHY

Websites referred…

- The Seattle Robotics Society Encoder library of robotics articles
- Dallas Personal Robotics Group. Most of these tutorials and articles were referred.
- Go Robotics.NET, this page has many useful links to robotics articles.

…BIBLIOGRAPHY

- Carnegie Mellon Robotics Club. This is the links page with lots of useful resources
- This page is called the "Micro-mouse Handbook" and an excellent tutorial for small scale robotics.
- This is the main website of microchip. Thousands of application notes, tutorials & manuals can be found here.

Thank you