A
PRESENTATION
ON
EMBEDDED SYSTEMS

CONTENTS

- ABOUT THE COMPANY
- ABOUT THE EMBEDDED SYSTEMS
- EXAMPLES OF EMBEDDED SYSTEMS
- DESCRIPTION OF SOFTWARES
- INTRODUCTION OF PROJECT
- WORKING OF PROJECT
- CODING OF PROJECT

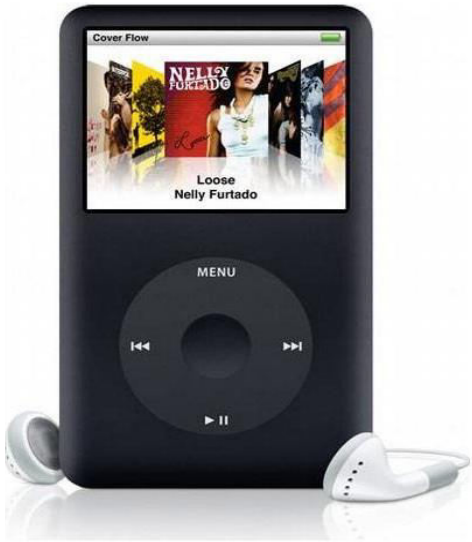
ABOUT THE COMPANY

- It is an autonomous body of the Government of Rajasthan under the Department of Technical Education.
- Foundation stone of CEG was laid down on 8th December 2006 at Jaipur.
- Rajasthan is the second state that is running this program after the highly acclaimed and successful program "Jawahar Knowledge Centre" in Andhra Pradesh.
- Provides a conducive environment for creating industry employable IT professionals by the way of arranging seminars lecturers, vocational trainings and industry relevant software trainings.
- Provides training in embedded system, C++ , java ,CCNA networking, CCNP, VLSI Technology, WAB Development etc.

ABOUT EMBEDDED SYSTEMS

- Embedded basically reflects the facts that they are an integral part of the system.
- It is a computer system that is built to control one or a few dedicated functions, and is not designed to be programmed by the end user in the same way that a desktop computer is.
- Contains processing cores that are either Micro-Controllers or Digital Signal Processors.
- An embedded system is designed to run on its own without human intervention, and may be required to respond to events in real time.

EXAMPLES OF EMBEDDED SYSTEMS



KEY COMPONENTS OF EMBEDDED SYSTEMS

PROCESSORS:

- It is the central processing unit known as the heart of the embedded systems.
- It is the hardware that executes the software and brings life to the embedded system.
- Controls the activities of all the other circuits.

MEMORY:

- Used to store the software that the processor will run.
- Also provides storage for data such as program variables, intermediate results, status information and any other data generated throughout the operation.

DESCRIPTION OF SOFTWARE

- In this system use KEIL software. The Keil software 8051 development tools can be used by any level of programmer to get the 8051 microcontroller architecture.

SOFTWARE DEVELOPMENT CYCLE: The Keil µVision project development cycle procedure is as follows:

- Create a project, select the target chip.
- Create source files in C or assembly.
- Build your application with the project manager.
- Correct errors in source files & also test the linked application.

INTRODUCTION OF PROJECT

COLLISION AVOIDANCE ROBOT

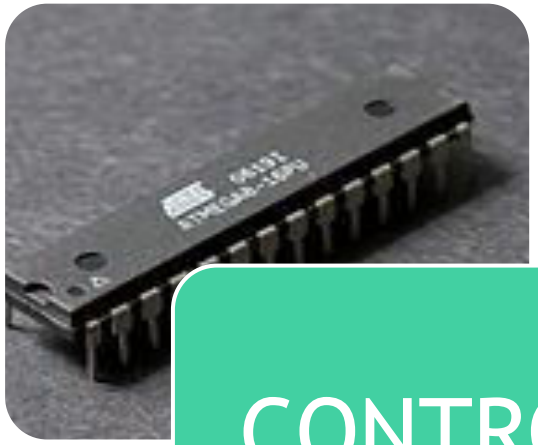
- A Robot is a mechanical device which performs automated physical tasks, either according to direct human supervision, a pre-defined program, or a set of general guidelines using artificial intelligence technique.

WHAT IS A COLLISION AVOIDANCE SYSTEM?

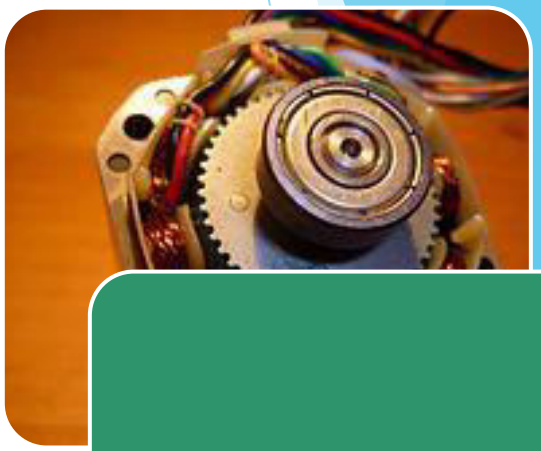
- A Collision Avoidance System can be defined as a device that detects possible obstructions in the way of a host vehicle (i.e. the vehicle that has the system installed in it), and helps in evading a collision.

BLOCK DESCRIPTION OF THE SYSTEM

SENSORS



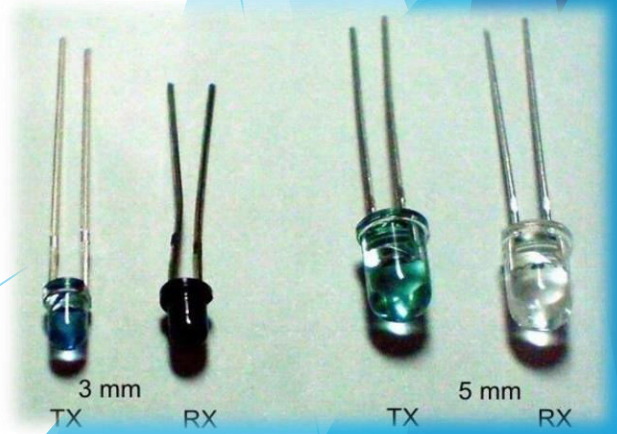
CONTROL UNIT



ACTUATOR

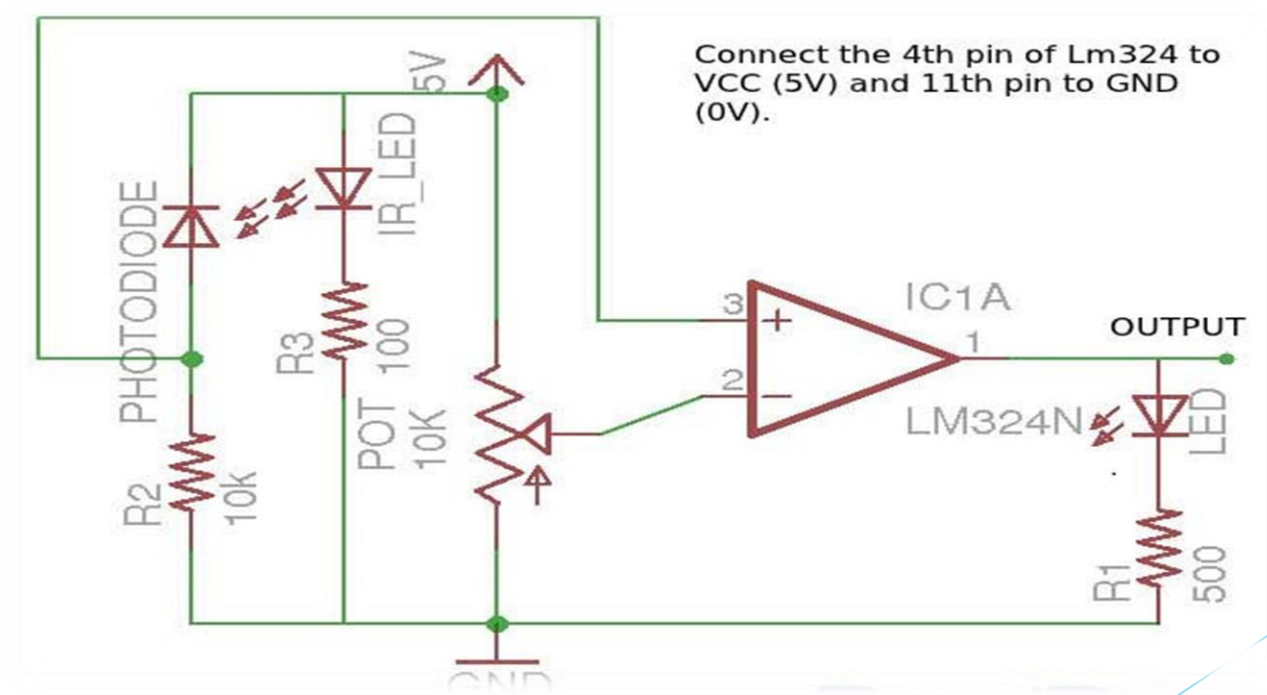
SENSORS

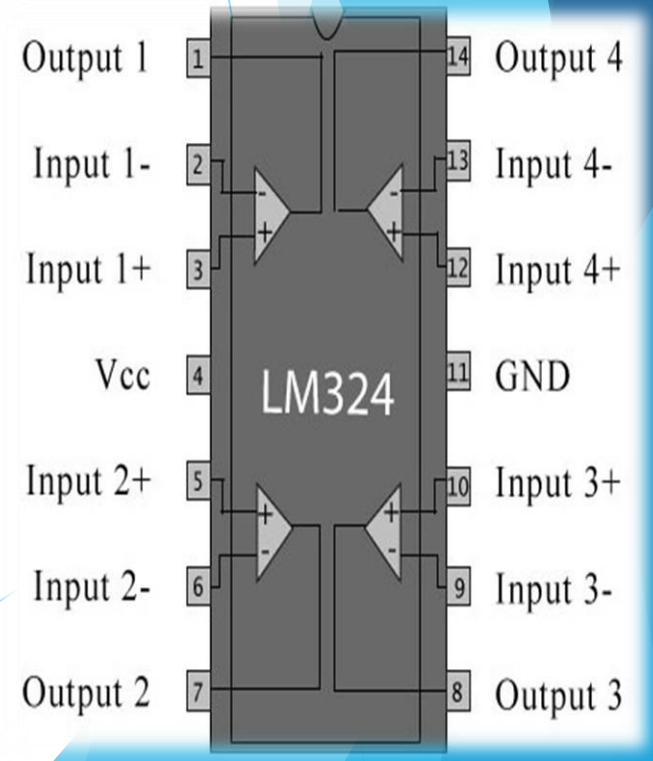
- This part of the system contains the sensors which perceive the signals reflected by the objects falling in the path of host vehicle. This is an important part of the system as based on the signals received the sensors the system is made to perform some scheduled tasks.
- There are different types of sensors that can be used for the purpose viz., IR sensors, Radars and even digital cameras.
- The system here, being just a prototype uses two IR sensors with a range of 1-3m. One sensor is placed in front of the vehicle and one at the rear end. Based on the signals from these sensors the vehicle will move.
- The sensors are interfaced with the microcontroller using the sensors driving IC LM324.



DESCRIPTION OF LM324 IC

- The LM324 series consists of four independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages.

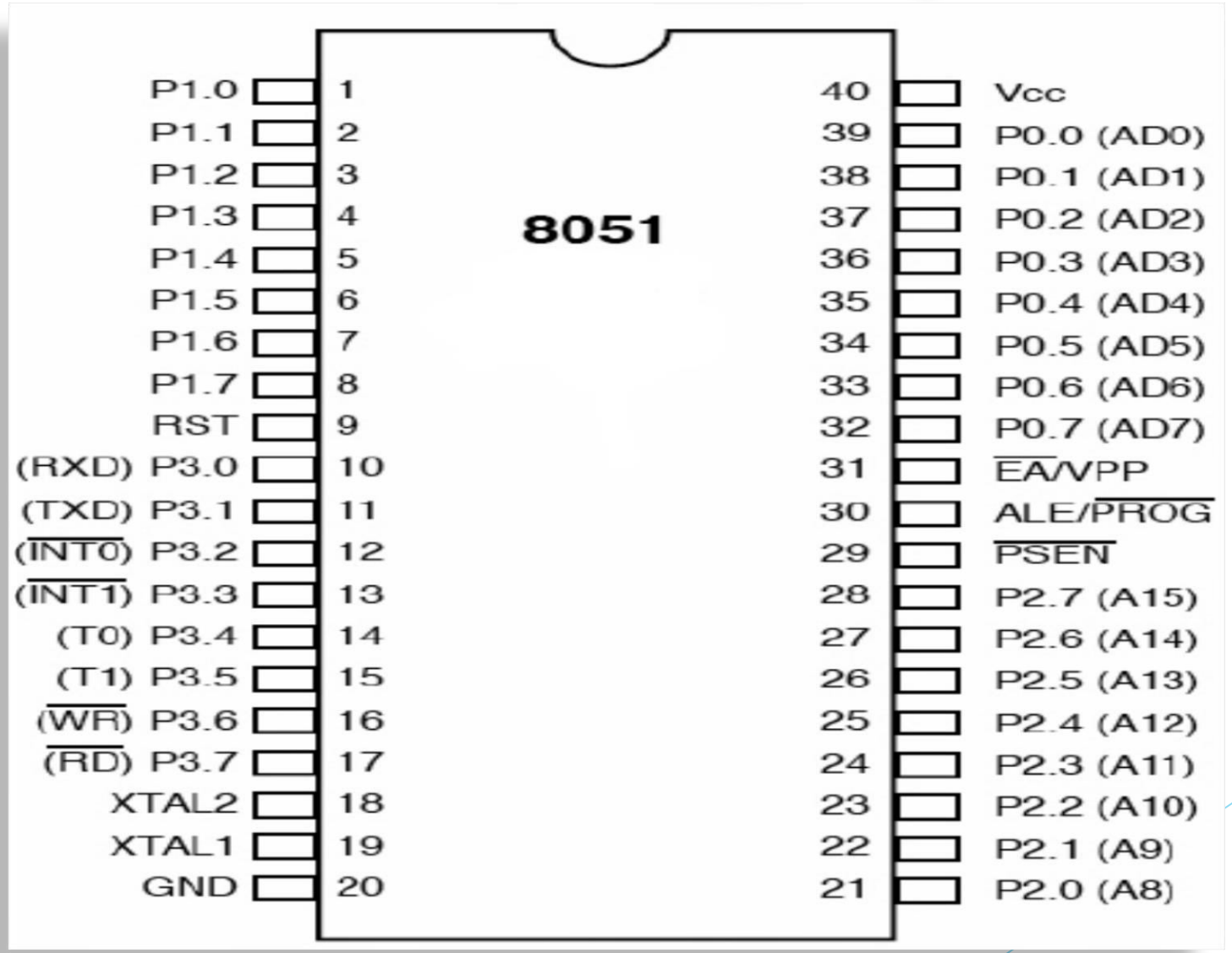




CONTROL UNIT (8051 µP)

- The control unit consists of a microcontroller which receives signals from sensors, and decides what operation to be performed by the system. The microcontroller used here is 8051 core.
- The microcontroller here is interfaced with a DC motor whose motion is controlled on the basis of the signals received from the sensors at the front and rear of the vehicle.
- It is a low-power, high-performance CMOS 8-bit microcontroller with 4K bytes of In-System Programmable Flash memory, 128 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry.

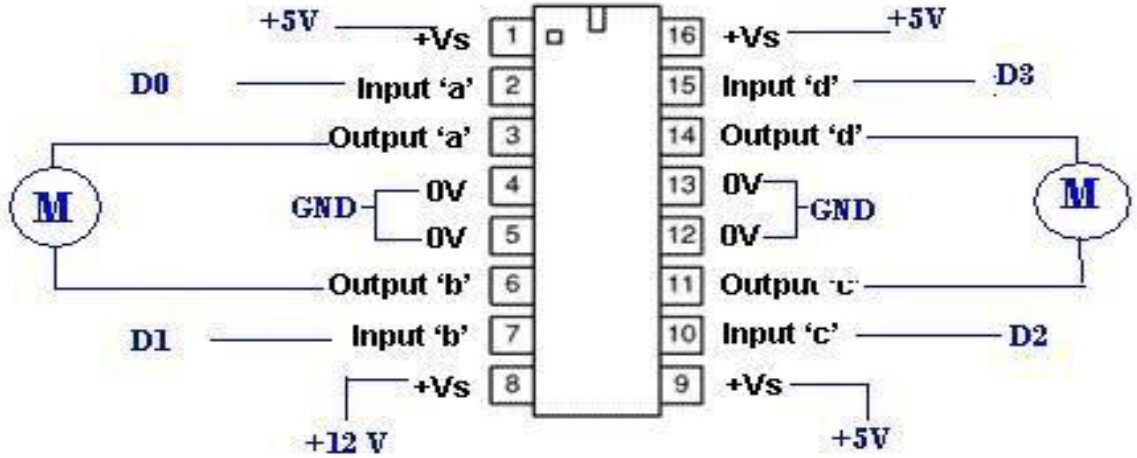
PIN DESCRIPTION OF 8051 µP

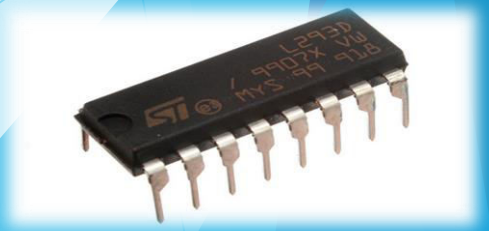


ACTUATORS

- The actuator used in this system is a DC Motor.
- The motor is interfaced with the microcontroller using the motor driving IC L293D.
- This IC has an H-bridge built into it, which allows the motor to be run in both clockwise and anti-clockwise by changing the polarity.
- This motor is responsible for the movement of the vehicle.
- The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V.

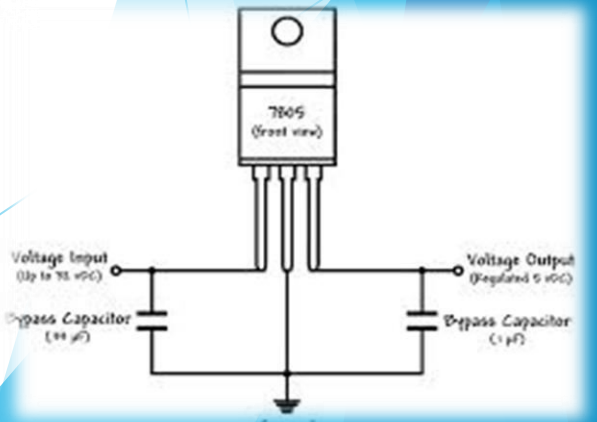
Circuit Diagram



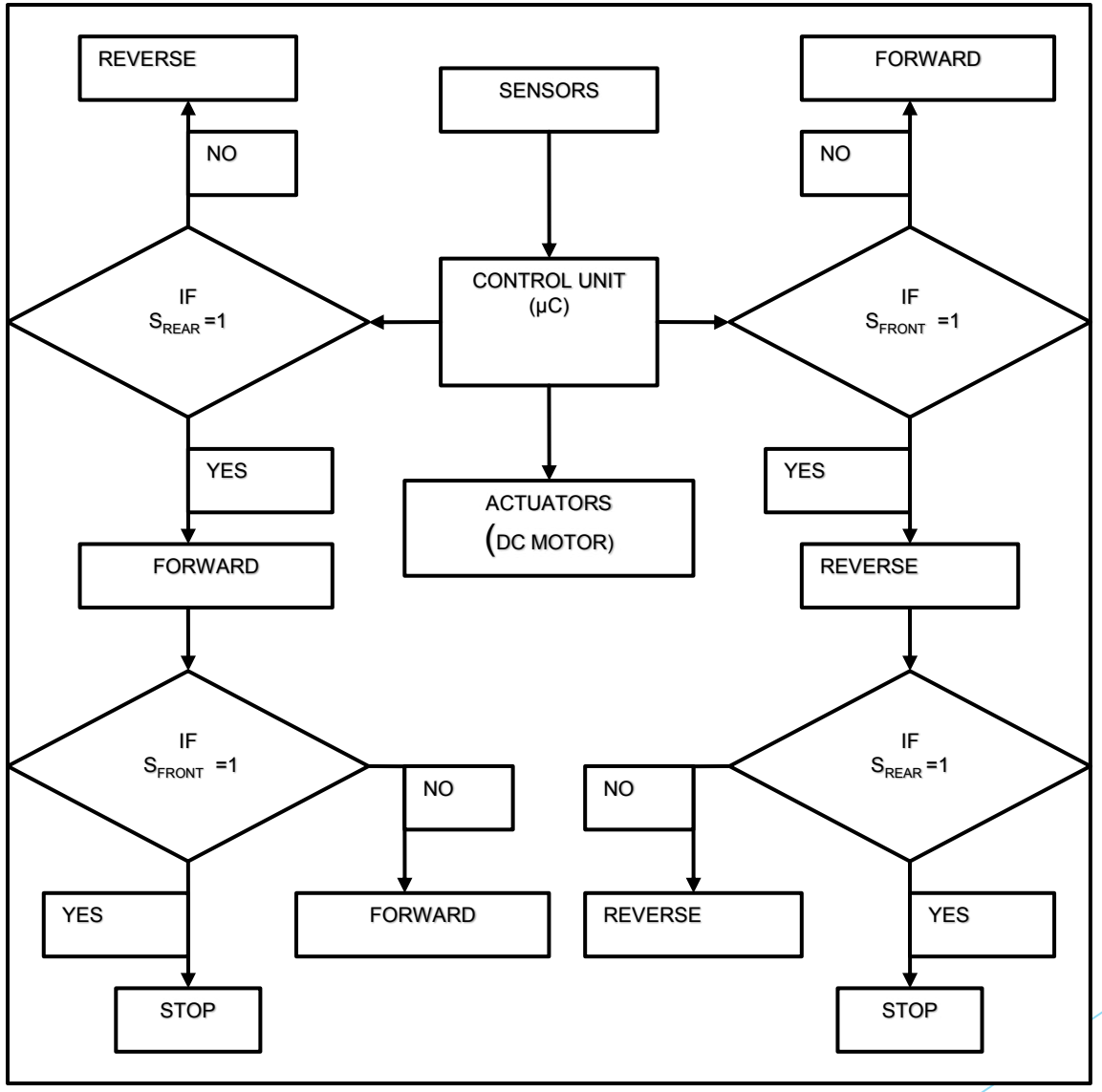


VOLTAGE REGULATORS

- It is a member of 78xx series of fixed linear voltage regulator ICs. The voltage source in a circuit may have fluctuations and would not give the fixed voltage output.
- The xx in 78xx indicates the fixed output voltage it is designed to provide. 7805 provides +5V regulated power supply.
- Capacitors of suitable values can be connected at input and output pins depending upon the respective voltage levels.
- These devices can be used with external components to obtain adjustable voltages and currents.
- If adequate heat sinking is provided, they can deliver over 1A output current.



WORKING OF THE SYSTEM



WORKING OF PROJECT

- As per the flow diagram, the microcontroller receives signals from the sensors placed in front (S_{Front}) and at the rear end (S_{Rear}) of the vehicle. Based on the signals the vehicle is moved forward and backward.
- If the signal from S_{Front} is high (i.e. this sensor detects the object) then the microcontroller turns the direction of the motor to make the vehicle move in reverse direction, else the vehicle will continue moving in the forward direction.
- When signal from S_{Front} is high and the vehicle starts moving back then, the signal from S_{Rear} is checked by the microcontroller. Now if S_{Rear} is high then the vehicle stops, otherwise the vehicle continues moving in reverse direction. This process takes place when initially the vehicle was moving in the forward direction.
- If initially the vehicle was moving in reverse direction then S_{Rear} is checked first and then S_{Front} is analyzed for similar conditions as explained above.

CODING

```
$MOD52
ORG 0000H

MAIN:       MOV P0,#0FFH;
            MOV P1,#0FFH;
            MOV P2,#00H;

CHECK:            MOV C,P1.0;
            JC LINE;
            JNC CHECK;
LINE:             MOV C,P0.1;
                        JNC LINE1:
                              MOV C,P0.0;
                        JC L1;
```

CONTINUED….

```
$MOD52
ORG 0000H

MAIN:        MOV P0,#0FFH;
        MOV P1,#0FFH;
        MOV P2,#00H;

CHECK:       MOV C,P1.0;
        JC LINE;
        JNC CHECK;
LINE:       MOV C,P0.1;
              JNC LINE1:
                  MOV C,P0.0;
              JC L1;
```

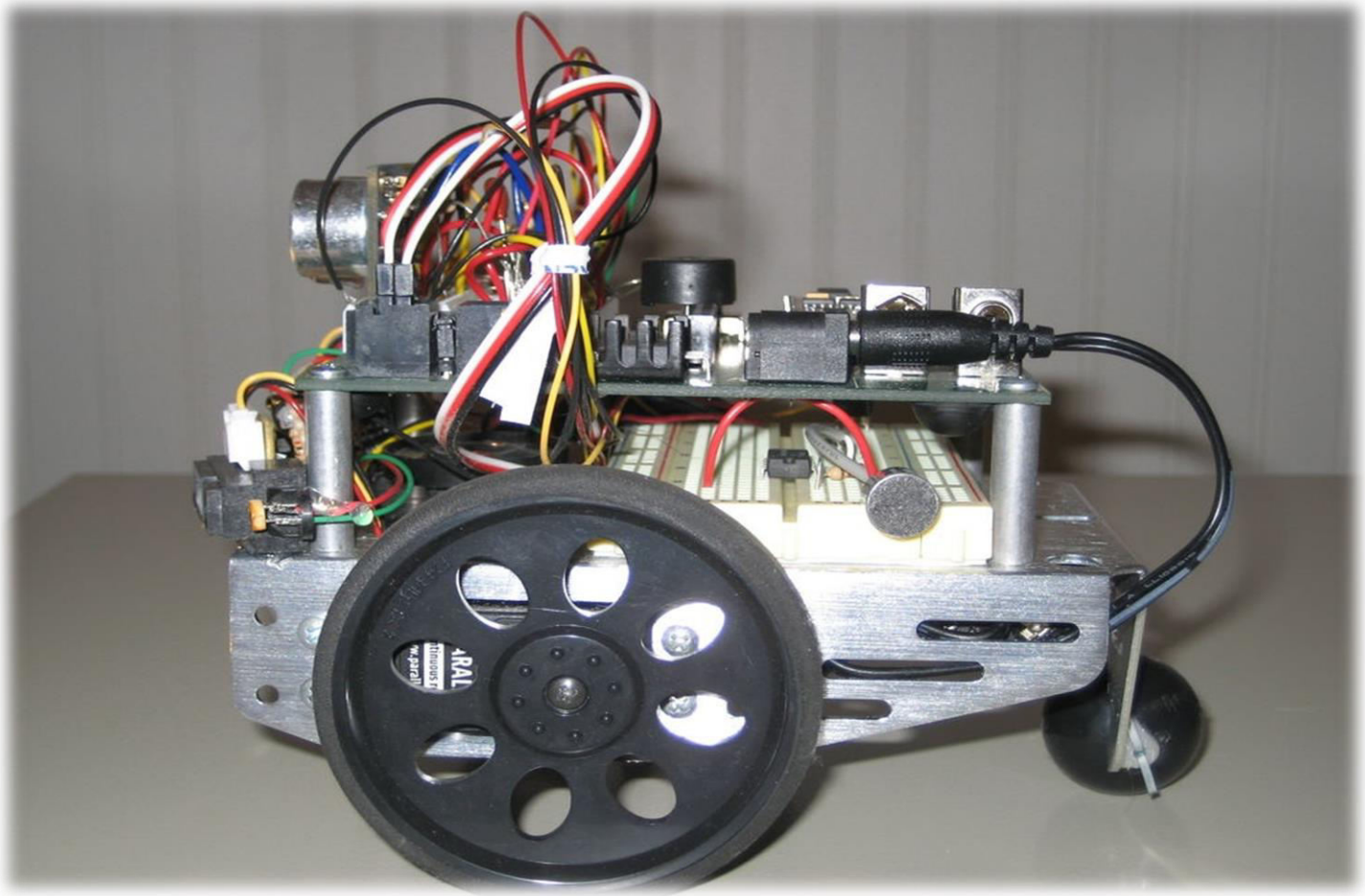
CONTINUED….

```
LINE2:      MOV C,P0.2;
            JNC LINE3;
            ACALL LEFT;
MOV C,P1.0;
            JC LINE;
            SJMP CHECK;

LINE3:      ACALL BACK;
            MOV C,P1.0;
            JC LINE;
            SJMP CHECK;
```

```
FORWARD:   MOV C,P0.3;
           JNC TURN;
           MOV P2,#1BH;
           ACALL DELAY;
           MOV P2,#12H;
           ACALL DELAY1;
           RET;

RIGHT:     MOV P2,#18H;
           ACALL DELAY;
           MOV P2,#10H;
           ACALL DELAY1;
           RET;

LEFT:      MOV P2,#03H;
           ACALL DELAY;
           MOV P2,#02H;
ACALL DELAY1;
           RET;

BACK:      MOV P2,#2DH;
           ACALL DELAY;
           MOV P2,#24H;
           ACALL DELAY1;
           RET;
```

```
TURN:       MOV P2,#2BH;
            ACALL DELAY2;
            MOV P2,#22H;
            ACALL DELAY1;
            SJMP FORWARD;

DELAY:      MOV R0,#77H;
DELY1:      DJNZ R0,DELY1;
            RET;


DELAY1:           MOV R0,#07H;
DALY1:      DJNZ R0,DALY1;
            RET;

DELAY2:           MOV R0,#02H;
DLY1:       MOV R1,#0FFH;
DLY2:       MOV R2,#0FFH;
DLY3:       DJNZ R2,DLY3;
            DJNZ R1,DLY2;
            DJNZ R0,DLY1;
            RET;

END;
```

PROJECT



REFRENCES

- www.ceg.rajasthan.gov.in
- www.wikkipedia.com
- www.atmelmicrocontrollers.com
- www.engineersgarage.com
- www.8051projects.info


Thank You!

QUERIES ???