Wall Follower Robot

Introduction

An *Wall Follower Robot* may be defined as a robot which can follow any kind of wall and is capable of changing its path according to the shape of the wall.

The basic tasks of an *Wall follower Robot* can be divided into 3 sections, namely

- Navigation
- Processing
- Execution

The robot must be equipped with some means by which it can navigate its surrounding to check the existence of a wall or the shape of the wall. In this project we have used Ultrasonic Sensors for the purpose of navigation .

After navigating the surrounding the robot must be capable of processing the input data from the navigation section. In this project the processing section is done with an EAB, which comes along with the microcontroller PIC18F26K22.

While processing the robot takes decision according to the algorithm designed for it.

After processing the navigated data the robot must do some work (e.g. Movement etc). In this project we have used motor driver IC L293D to drive the motors/wheels according to the processed output from the EAB.

Ultrasonic Sensor

Ultrasonic sensors work on a principle similar to radar or sonar which evaluate attributes of a target by interpreting the echoes from radio or sound waves respectively. Ultrasonic sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor. An analog Ultrasonic Sensor produces analog signal according to the distance from the object.

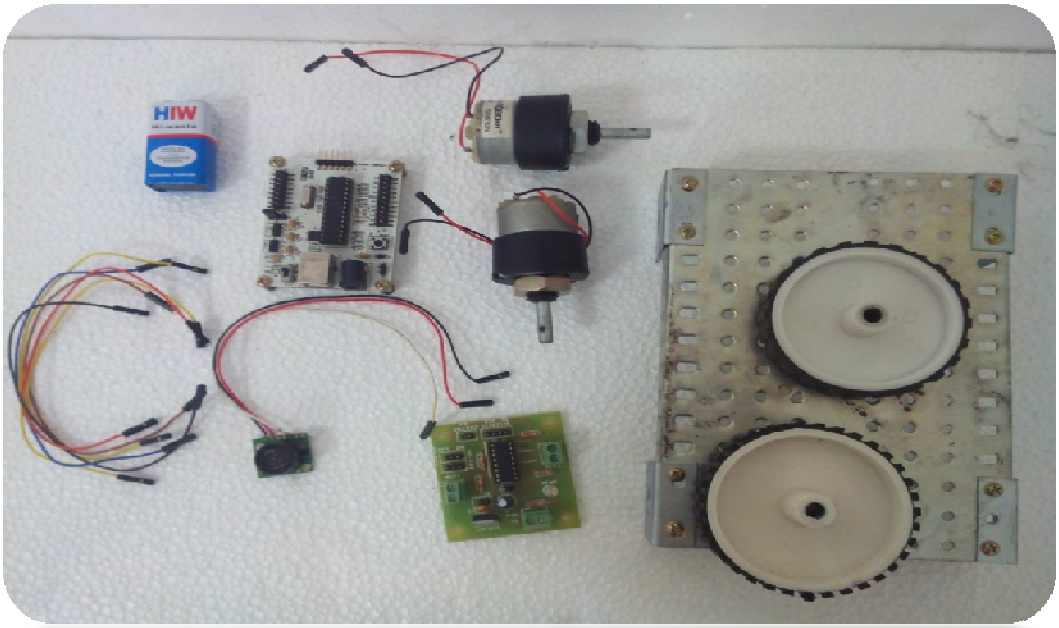
Motor Driver Board

The motor driver board consists of IC L293D. It has 4 inputs and 4 outputs. Since, the current from the EAB is not sufficient enough to drive the motors, so we have to use this Driver to increase the current by which we can drive the DC Motors.

Components

The Components required for building the Wall follower Robot are:

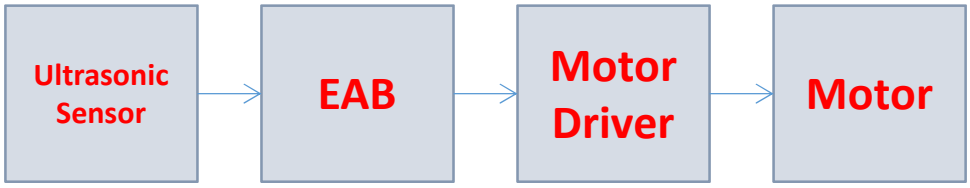
- Embedded Application Board
- Motor Driver Board (containing L293D)
- Analog Ultrasonic Sensor (1 nos)
- DC Motors (2 nos)
- Chassis
- Connectors(Jumpers)
- 9V battery
- Wheels
- Walls



Application Notes

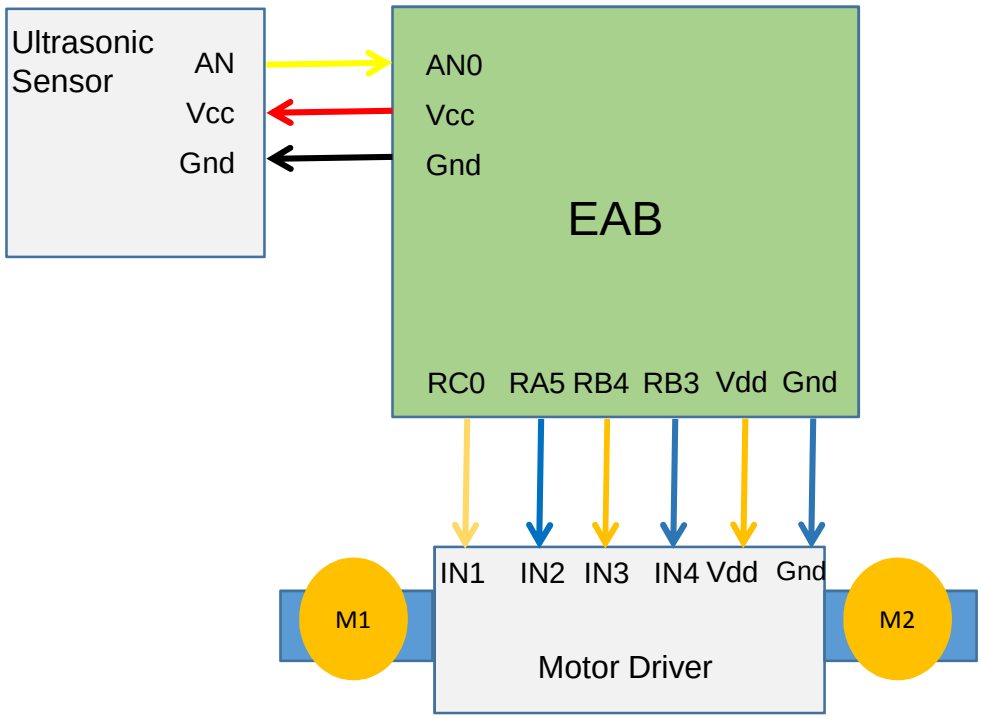
Block Diagram

Block level representation of the different blocks of the Wall follower Robot.



Schematic Diagram

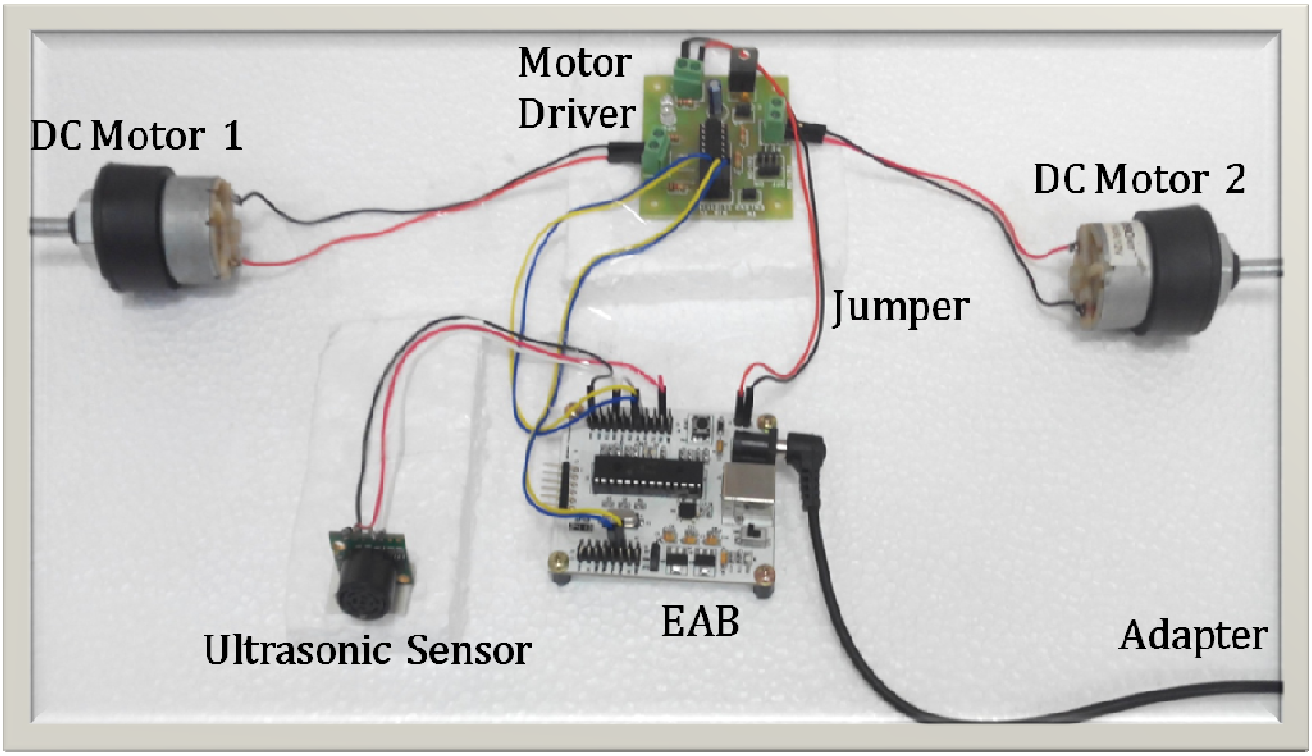
The Schematic diagram illustrates the circuit connections for designing the application.



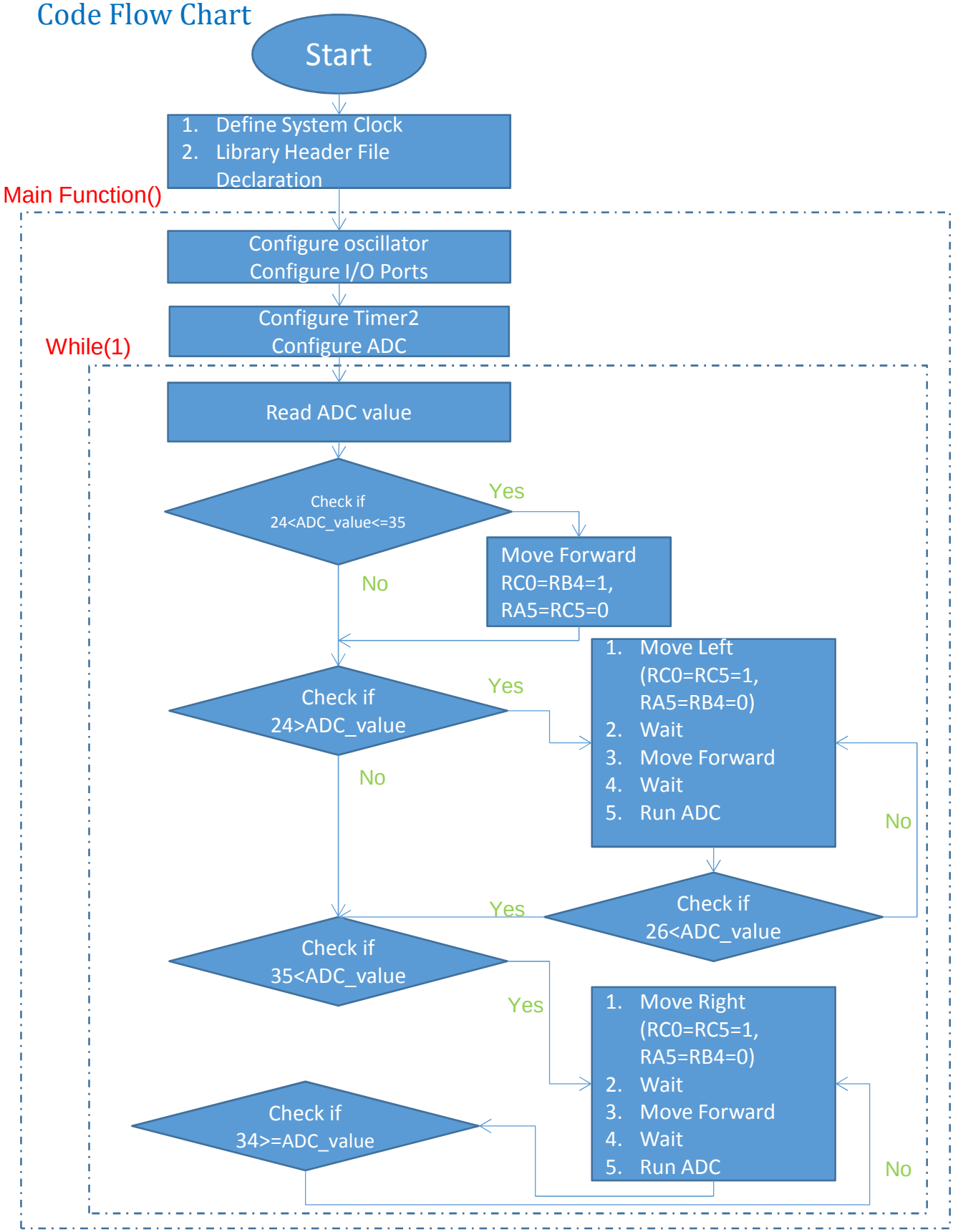
Connection Description

In this project we have used one Ultrasonic Sensor module as wall detector. It is placed in the Right-Front corner of the robot. This module has 3-pins, as GND, VCC and DATA . The GND and VCC pins are connected to the GND and VCC pins of an EAB. The Analog DATA pin of the sensor is connected to the AN0 pin of the EAB.

The output is from the RC0, RA5, RB4 and RB3 pins of the EAB. These pins are responsible to control the robot and they are connected to the IN1, IN2, IN3 and IN4 pins of the Motor Driver board, respectively. Both the Motor Driver Board and the EAB are powered with 12V rechargeable battery.



Code Flow Chart



Application Notes

Source Code

The Source code shown below is the firmware to be flashed in the microcontroller of the Embedded Application Board. The Source code is commented for better understanding of the user.

Refer to the EAB User Guide and the EAB Programming Guide for more details on how to Flash(burn) program(Source Code) in the microcontroller of Embedded Application Board.

```
#define SYS_CLK 8000000         // Required for delay macro functions
                                // Default 1MHZ, else change as per configuration
/*** INCLUDE STANDARD HEADERS & LIBRARY ***/
#include <stdio.h>
#include <stdlib.h>

#include "EAB_Library.h"

/*** GLOBAL VARIABLES ****/

/*-------------------------------------------------------------------------*/
void main(void)
{
    /*** LOCAL VARIABLES ***/
    ushort LSB=0;
    short ADCvalue;

    /*** INTITALIZE OSCILLATOR, PERIPHERAL & HARDWARE ***/
    Oscillator.SetFreq_8MHZ();                  // Select system clock at 8 MHz

    Timer2.SetPeriod(Timer2.config.PRESCALER_16,Timer2.config.POSTSCALER_16,255);
                                                // Set Timer2 at maximum period

    PinDigitalOut(RC0);                 //RC0 as digital output
    PinDigitalOut(RA5);                 //RA5 as digital output
    PinDigitalOut(RB4);                 //RB4 as digital output
    PinDigitalOut(SDO2);                //SDO2 as digital output
    PinDigitalOut(RC5);                 //RC5 as digital output

    /*** OPEN ADC  ***/
    PinAnalogIn(AN0);                   //AN0 as analog input
    ADC.SetChannel(CHANNEL_AN0);        //Select ADC channel0
    ADC.Open();                         //Open ADC
```

```
/*** PLACE THE REPETITIVE TASKS IN THIS LOOP ***/
while(1)
{
    ADCvalue = ADC.ReadData();          //Read ADC value
    LSB = ADCvalue;

    if(LSB>24 && LSB<=35)               //check for condition
    {
        PinWrite.RC0=1;                 //Set RC0 output High
        PinWrite.RA5=0;                 //Set RA5 output Low
        PinWrite.RB4=1;                 //Set RB4 output High
        PinWrite.SDO2=0;                //Set SDO2 output Low
        PinWrite.RC5=0;                 //Set RC5 output Low
    }
    if(LSB<=24)
    {
        while(LSB <=26)
        {
            PinWrite.RC0=1;             //Set RC0 output High
            PinWrite.RA5=0;             //Set RA5 output Low
            PinWrite.RB4=0;             //Set RB4 output Low
            PinWrite.SDO2=1;            //Set SDO2 output High
            PinWrite.RC5=1;             //Set RC5 output High

            Timer2_Flag=0;
            while(Timer2_Flag==0);

            PinWrite.RC0=1;             //Set RC0 output High
            PinWrite.RA5=0;             //Set RA5 output Low
            PinWrite.RB4=1;             //Set RB4 output High
            PinWrite.SDO2=0;            //Set SDO2 output Low
            PinWrite.RC5=0;             //Set RC5 output Low

            Timer2_Flag=0;
            while(Timer2_Flag==0);

            ADCvalue = ADC.ReadData();
            LSB = ADCvalue;
        }
    }
```

```
        if(LSB >35)
        {
            while(LSB>=34)
            {
                PinWrite.RC0=0;             //Set RC0 output Low
                PinWrite.RA5=1;             //Set RA5 output High
                PinWrite.RB4=1;             //Set RB4 output High
                PinWrite.SDO2=0;            //Set SDO2 output Low
                PinWrite.RC5=0;             //Set RC5 output Low

                Timer2_Flag=0;
                while(Timer2_Flag==0);

                PinWrite.RC0=1;             //Set RC0 output High
                PinWrite.RA5=0;             //Set RA5 output Low
                PinWrite.RB4=1;             //Set RB4 output High
                PinWrite.SDO2=0;            //Set SDO2 output Low
                PinWrite.RC5=0;             //Set RC5 output Low
                Timer2_Flag=0;
                while(Timer2_Flag==0);

                ADCvalue = ADC.ReadData();
                LSB = ADCvalue;
            }
        }
    }
}
/*-------------------------------------------------------------------------*/
```

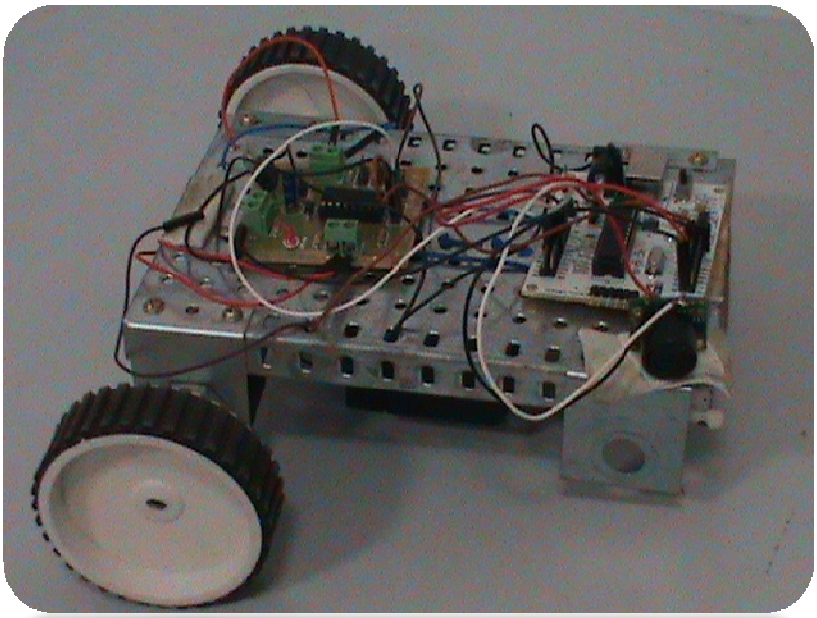
Application Notes

How to Operate

Follow the steps mentioned below in order to operate the project…

- Prepare an arena as you wish.
- Flash the code into the microcontroller.
- Connect each and every part properly.
- Place the robot in the arena.
- Power the EAB, Sensor Board and Motor Driver Circuit with 9V/12V DC. Carefully check the polarities and then connect .
- Switch ON the EAB.

Now you can see the Robot moving automatically, in the arena, keeping a particular range of distance from the wall.



More Projects

Various other applications can be built using Wall follower Robot.

Some of such applications are given below:

- Industrial Automation
- Security System