Wireless Robot

Introduction

An *Wireless Robot* may be defined as a robot which can be controlled wirelessly.

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

- Data Communication
- Processing
- Execution

The robot must be equipped with some means by which it can communicate with its commander. In this project we have used X-Bee modules for the purpose of communication.

This project consists of two Xbee modules. One is at the transmitter's end and other is at the receiver's end. The data sent by the user can be received by the robot's Xbee module. Through UART communication the microcontroller of the EAB at the receiver end will collect the data.

After communicating with the commander, the robot must be capable of processing the input data. 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 received 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.

Copyright: Brigosha reserves all rights in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. Brigosha is the registered trademark of Brigosha Technologies Pvt. Ltd. Specifications within this document are preliminary and subject to change without notification. Document is not intended for publication.

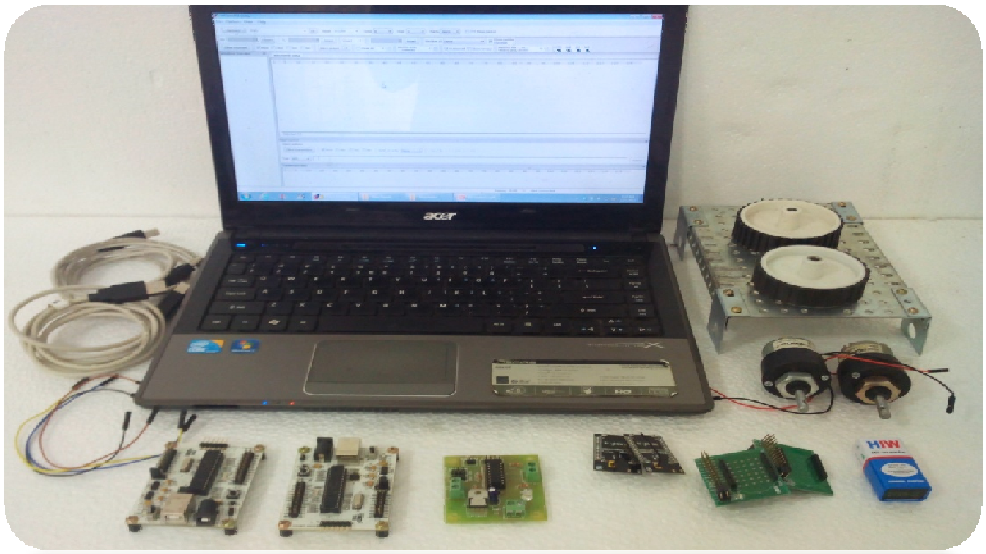
Motor Driver Board

The motor driver board consists of ICL293D. 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 Wireless Robot are:

- Embedded Application Board (2 pc)
- Motor Driver Board (containing L293D)
- XBee(1 pair)
- DC Motors (2 nos)
- Chassis
- USB Cable
- Connectors(Jumpers)
- 9V battery
- Wheels
- Expansion Card

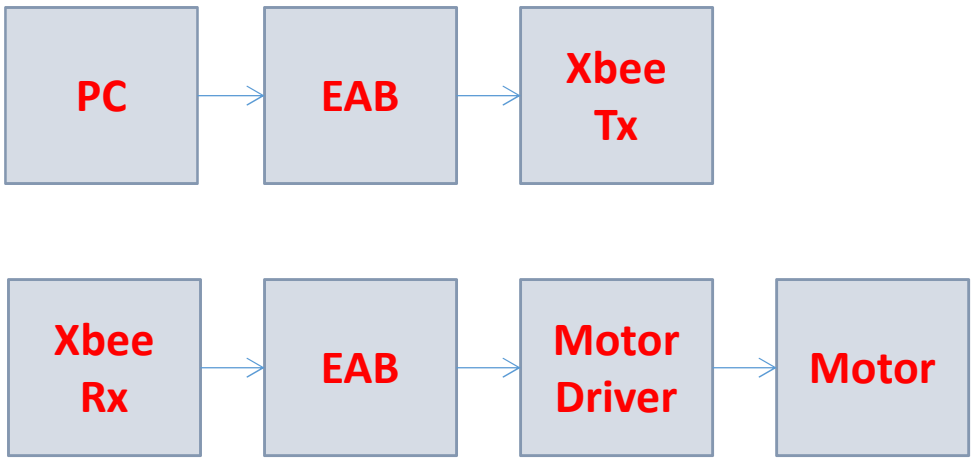


Copyright: Brigosha reserves all rights in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. Brigosha is the registered trademark of Brigosha Technologies Pvt. Ltd. Specifications within this document are preliminary and subject to change without notification. Document is not intended for publication.

Application Notes

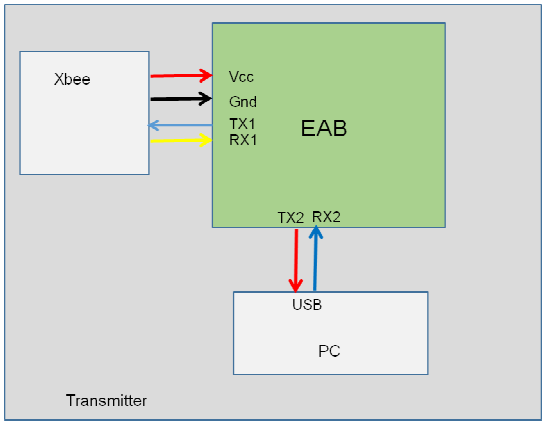
Block Diagram

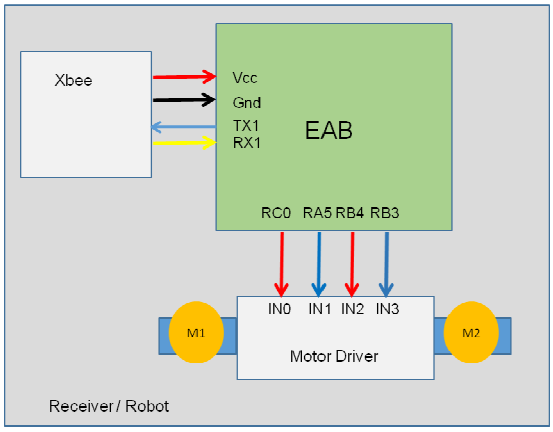
Block level representation of the different blocks of the Wireless Robot.



Schematic Diagram

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





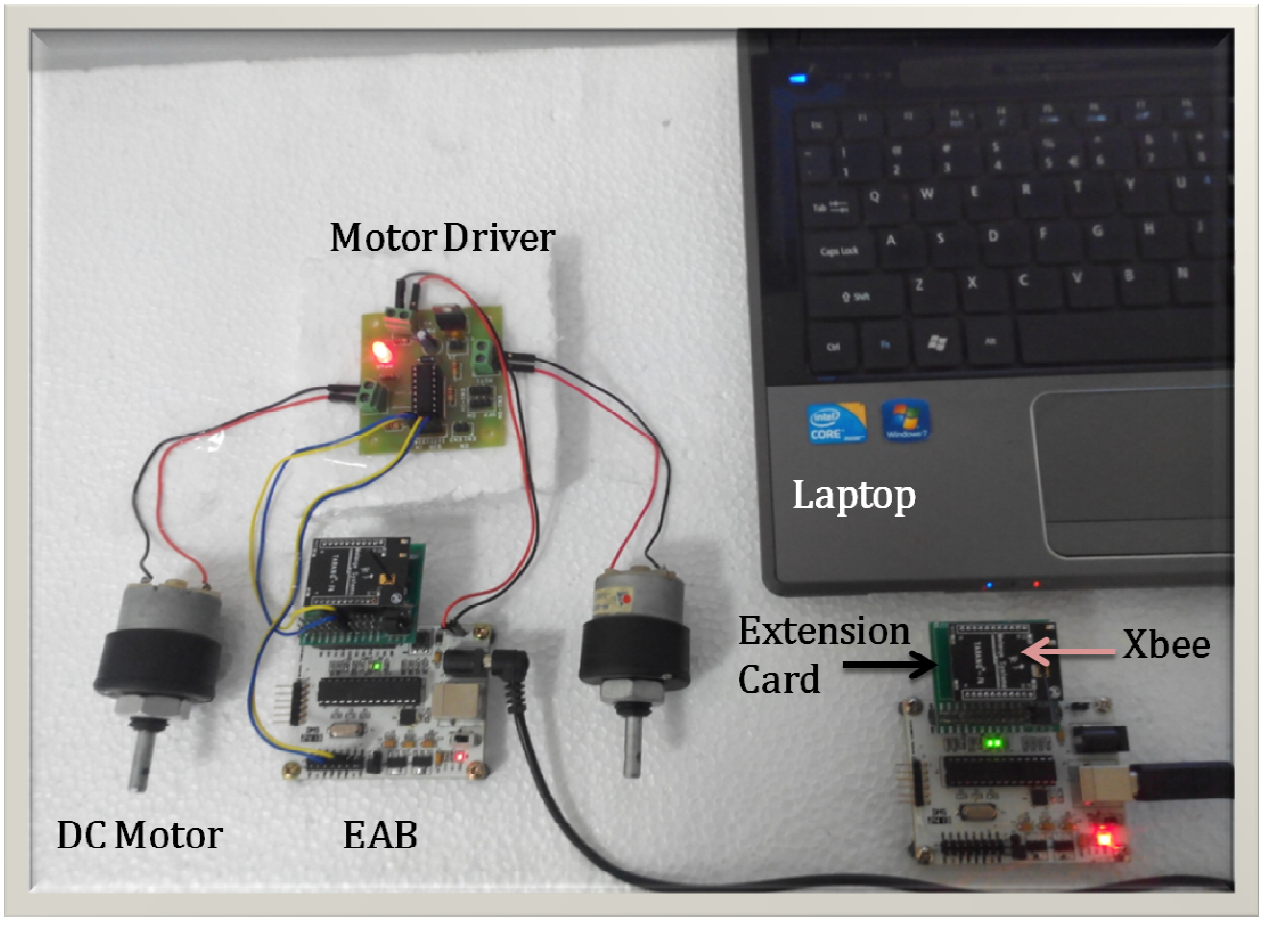
Note: *Any GPIO pin can be used for this project. Set the particular GPIO pins to output and provide a high /low signal to the pin.*

Copyright: Brigosha reserves all rights in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. Brigosha is the registered trademark of Brigosha Technologies Pvt. Ltd. Specifications within this document are preliminary and subject to change without notification. Document is not intended for publication.

Connection Description

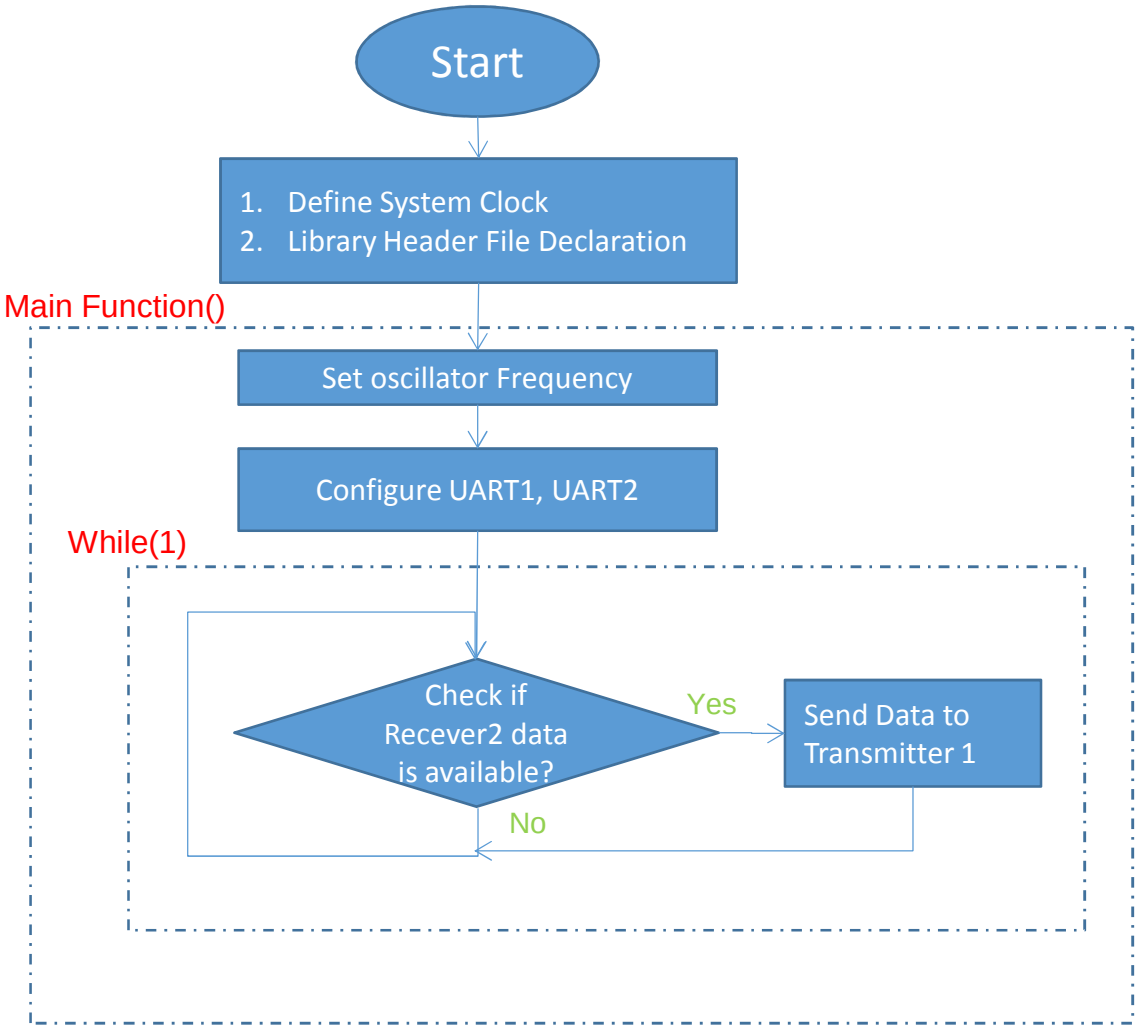
In this project we have used one Xbee module as wireless communication device. Each one of them is connected with one EAB. This module is placed on the Extension Card and then connected to EAB's 22 pin header. Xbee and EAB shares UART1 for communication. One of these EAB-Xbee pair is available with PC, to send command while the other is with the robot to receive the command.

The output of the robot side's EAB are from the pins RC0, RA5, RB4 and RB3. These pins are responsible to control the robot and they are connected to the IN0, IN1, IN2 and IN3 pins of the Motor Driver board, respectively. Both the Motor Driver Board and the EABs are powered with 12V rechargeable battery.



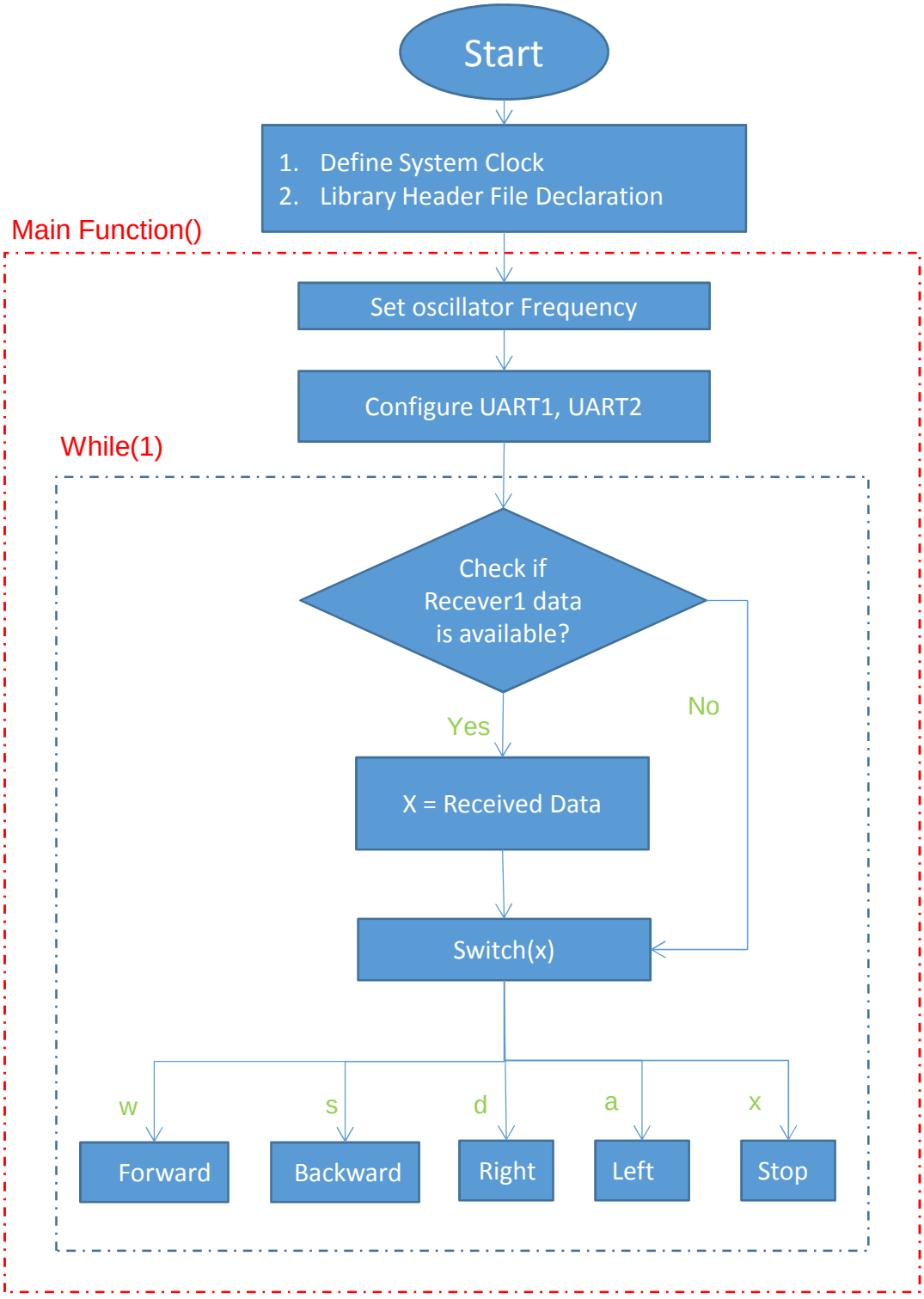
Copyright: Brigosha reserves all rights in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. Brigosha is the registered trademark of Brigosha Technologies Pvt. Ltd. Specifications within this document are preliminary and subject to change without notification. Document is not intended for publication.

Code Flow Chart For User Side's EAB



Copyright: Brigosha reserves all rights in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. Brigosha is the registered trademark of Brigosha Technologies Pvt. Ltd. Specifications within this document are preliminary and subject to change without notification. Document is not intended for publication.

Code Flow Chart For Robot Side's EAB



Copyright: Brigosha reserves all rights in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. Brigosha is the registered trademark of Brigosha Technologies Pvt. Ltd. Specifications within this document are preliminary and subject to change without notification. Document is not intended for publication.

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

User Side's Code

```
#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)
{

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

    Serial2.Open(9600);                     // Select UART2 at 9600 Baud Rate
    Serial1.Open(9600);                     // Select UART1 at 9600 Baud Rate

    /*** PLACE THE REPETITIVE TASKS IN THIS LOOP ***/
    while(1)
    {
        /*** SEND THE RECEIVED DATA ON SERIAL PORT ***/
        if(Serial2_RxFlag)                  // Check receiver1 flag
        {
            Serial1_SendByte(Serial2_ReadByte());   //Transmit via UART1
            Serial2_RxFlag=0;               // Clear the Flag
        }
    }
}
/*-------------------------------------------------------------------------*/
```

Copyright: Brigosha reserves all rights in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. Brigosha is the registered trademark of Brigosha Technologies Pvt. Ltd. Specifications within this document are preliminary and subject to change without notification. Document is not intended for publication.

Robot Side's Code

```
#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 variable***/
    uchar Rcv_Data=0;

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

    /**Set Timer0 at 0.2 Sec**/
    Timer0.SetPeriod(Timer0.config.PRESCALER_8,Timer0.config.COUNTER_16BIT);

    Serial2.Open(9600);                     // Select UART2 at 9600 Baud Rate
    Serial1.Open(9600);                     // Select UART1 at 9600 Baud Rate

    PinDigitalOut(RC0);                     // RC0 as Digital Output
    PinDigitalOut(RA5);                     // RA5 as Digital Output
    PinDigitalOut(RB4);                     // RB4 as Digital Output
    PinDigitalOut(SDO2);                    // SDO2 as Digital Output

    /*** PLACE THE REPETITIVE TASKS IN THIS LOOP ***/
    while(1)
    {
        /*** SEND THE RECEIVED DATA ON SERIAL PORT ***/
        if(Serial1_RxFlag)                  // Check receiver1 flag
        {
            Serial2_SendByte(Serial1_ReadByte());   //Transmit via UART2
            Rcv_Data=Serial1_ReadByte();            //Read data from UART1
            Serial1_RxFlag=0;               // Clear the flag
        }
```

Copyright: Brigosha reserves all rights in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. Brigosha is the registered trademark of Brigosha Technologies Pvt. Ltd. Specifications within this document are preliminary and subject to change without notification. Document is not intended for publication.

```
switch(Rcv_Data)                        //check condition
{
    case 'W':
    case 'w':
    {
        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

        break;
    }
    case 'A':
    case 'a':
    {
        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

        Timer0_Flag = 0;
        while(Timer0_Flag==0);
        Rcv_Data = '0';

        PinWrite.RC0 = 0;               // Set RC0 Output Low
        PinWrite.RA5 = 0;               // Set RA5 Output Low
        PinWrite.RB4 = 0;               // Set RB4 Output Low
        PinWrite.SDO2= 0;               // Set SDO2 Output Low

        break;
    }
    case 'D':
    case 'd':
    {
        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

        Timer0_Flag = 0;
        while(Timer0_Flag==0);
        Rcv_Data = '0';

        PinWrite.RC0 = 0;               // Set RC0 Output Low
        PinWrite.RA5 = 0;               // Set RA5 Output Low
        PinWrite.RB4 = 0;               // Set RB4 Output Low
        PinWrite.SDO2= 0;               // Set SDO2 Output Low
        break;
    }
```

Copyright: Brigosha reserves all rights in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. Brigosha is the registered trademark of Brigosha Technologies Pvt. Ltd. Specifications within this document are preliminary and subject to change without notification. Document is not intended for publication.

```
            case 'S':
            case 's':
            {
                PinWrite.RC0 = 0;                   // Set RC0 Output Low
                PinWrite.RA5 = 1;                   // Set RA5 Output High
                PinWrite.RB4 = 0;                   // Set RB4 Output Low
                PinWrite.SDO2= 1;                   // Set SDO2 Output High
                Timer0_Flag = 0;
                while(Timer0_Flag==0);
                Rcv_Data= '0';
                PinWrite.RC0 = 0;                   // Set RC0 Output Low
                PinWrite.RA5 = 0;                   // Set RA5 Output Low
                PinWrite.RB4 = 0;                   // Set RB4 Output Low
                PinWrite.SDO2= 0;                   // Set SDO2 Output Low
                break;
            }
            case 'X':
            case 'x':
            {
                PinWrite.RC0 = 0;                   // Set RC0 Output Low
                PinWrite.RA5 = 0;                   // Set RA5 Output Low
                PinWrite.RB4 = 0;                   // Set RB4 Output Low
                PinWrite.SDO2= 0;                   // Set SDO2 Output Low
                break;
            }
        }
    }
}
/*-------------------------------------------------------------------------------*/
```

Copyright: Brigosha reserves all rights in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. Brigosha is the registered trademark of Brigosha Technologies Pvt. Ltd. Specifications within this document are preliminary and subject to change without notification. Document is not intended for publication.

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 both 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 send command through laptop, using Hterm. The Robot moving accordingly, in the arena.

Commands:

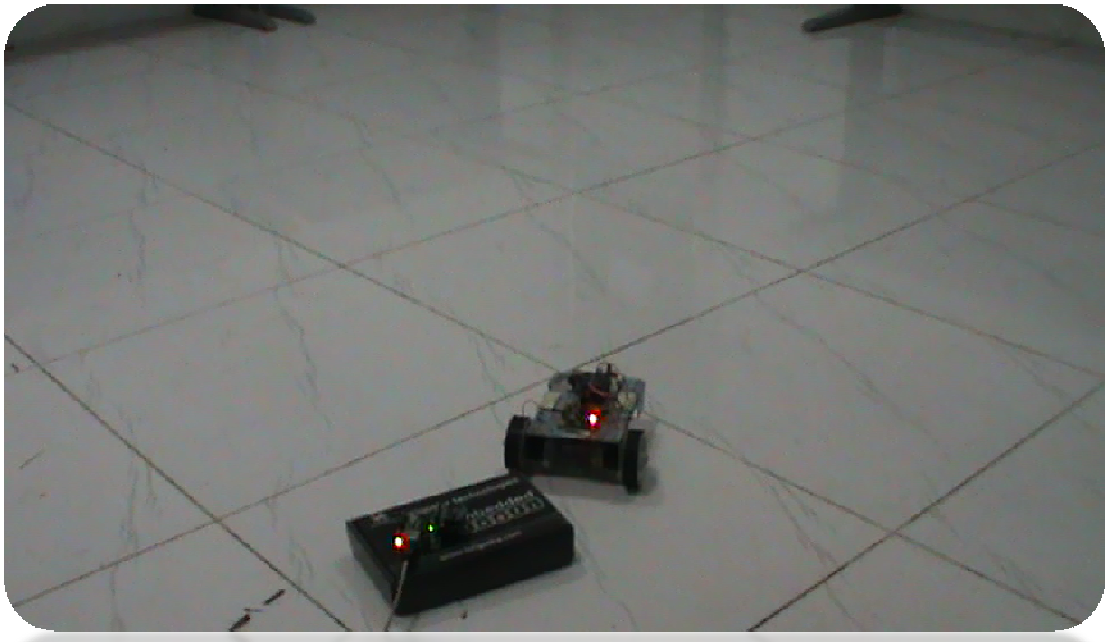
w = forward

s = backward

a = left

d = right

x = stop



More Projects

Various other applications can be built using Wireless Robot.

Some of such applications are given below:

- Industrial Automation
- Spy Robot

Copyright: Brigosha reserves all rights in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. Brigosha is the registered trademark of Brigosha Technologies Pvt. Ltd. Specifications within this document are preliminary and subject to change without notification. Document is not intended for publication.