Robot Navigation Using Xbee

Mustafa Lokhandwala - 13D070043
Jashan Singhal - 130070026
Group 5 - Monday
TAs: Harshal & Prasad

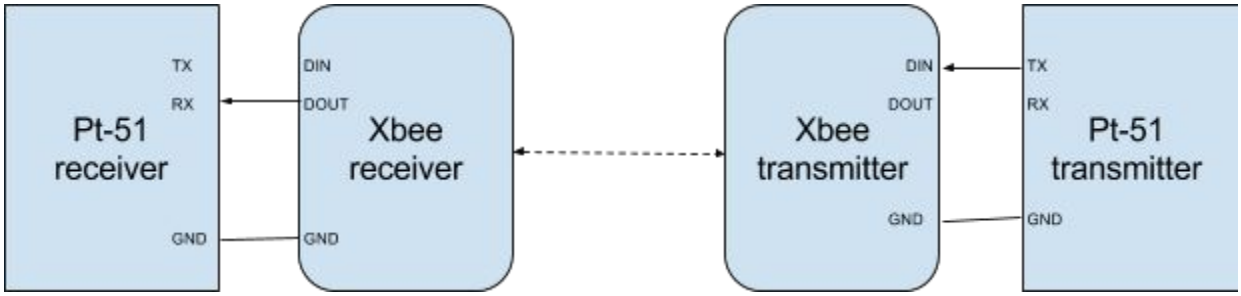
Objective:-

1. Understand the interface of UART and Xbee.
2. Understand the system design.

Deliverables:-

In this project, we have to navigate a robot using a wireless remote. For wireless communication, we have to use two Xbee modules.We need two Pt-51 board, one as receiver and another as transmitter.

Block Diagram



Description of the design

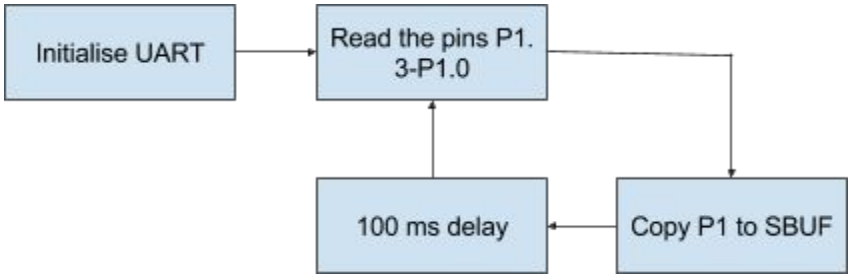
At the transmitter end, Pt-51 board reads DIP switch input (which is connected to P1.0-P1.3) for robot direction. It sends switch value to Xbee module over UART. At the receiver end, Xbee module receives data from the transmitter Xbee. It sends the data to receiver side Pt-51 board. Pt-51 decides motor control signals based on received data.

Direction command	Switch position(P1.3 to P1.0)
Forward	1111
Reverse	0110
Right turn	X011
Left turn	110X
Stop	X00X

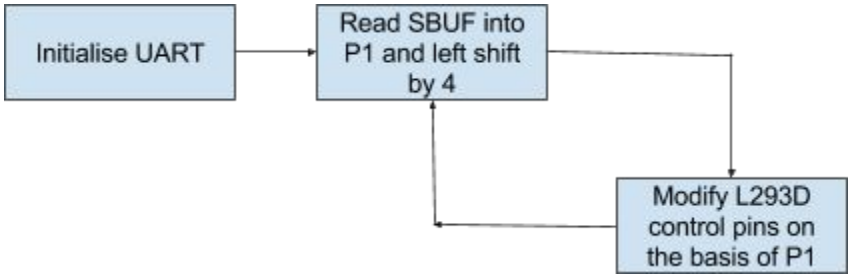
X means don't care, 1 means ON and 0 means OFF.

Flow-chart of the program

Receiver Side:



Transmitter Side:



Shortcomings/ Future improvements-

a. In this project we implemented hard forward, hard backward , hard right turn and hard left turn (problem statement demanded so) whereas given the resources, we could have implemented PWM to get the bot to move at our desired speed and turn gently.
b. Since Zigbee offers two-way communication, with the help of necessary components we can calculate the speed of the bot at the zigbee on-board and transmit it to the zigbee at source, giving us a better control of the bot.
c. To add to the aesthetics, we can also add an accelerometer at the source, and depending upon its orientation we can transmit the desired values and perform the required function.

Transmitter side code:-

```
/* @section  I N C L U D E S */
#include "at89c5131.h"

void init_uart();
void delay_ms(int);

void main(void)
{
       init_uart();
       while(1)
       {
```

```
                SBUF = P1;
                while(!TI);
                TI  = 0;
                delay_ms(100);
        }
}

void init_uart()
{
        TMOD = 0x20;
        TH1     = 0xF3;
        SCON = 0x50;
        TR1  = 1;
}

/**
 * FUNCTION_PURPOSE: A delay of around 1000us for a 24MHz crystel
 * FUNCTION_INPUTS: void
 * FUNCTION_OUTPUTS: none
 */
void delay_ms(int delay)
{
        int d=0;
        while(delay>0)
        {
                for(d=0;d<382;d++);
                delay--;
        }
}
```

Receiver side code:-

```
#include "at89c5131.h"
#include "stdio.h"

sbit C1 = P0^0;
sbit C2 = P0^1;
sbit C3 = P0^2;
sbit C4 = P0^3;
sbit O1 = P1^5;
sbit O2 = P1^6;
sbit O3 = P1^4;
sbit O4 = P1^7;
void init_uart()
{
        TMOD= 0x20;
        TH1= 0xF3;
        SCON=0x50;
        TR1=1;
```

```
        }

void main(void){
        init_uart();
        while(1)
        {       RI=0;
                while(!RI);
                P1=SBUF<<4;

                if(P1==0xF0)
                {C1=1;
                 C2=0;
                 C3=1;
                 C4=0;
                }

                else if(P1==0x60)
                {C1=0;
                 C2=1;
                 C3=0;
                 C4=1;}

                else if(O1==0&&O2==0){
                        C1=0;
                        C2=0;
                        C3=0;
                        C4=0;
                }

                else if(O3==1&&O1==1){
                  C1=1;
                        C2=0;
                        C3=0;
                        C4=1;

                }

                else if(O2==1&&O4==1){
                  C1=0;
                        C2=1;
                        C3=1;
                        C4=0;
                }
                else{}


        }
}
```