TEST PLAN

The test plan combines of debugging the different subsystems of the robot. My robot had four subsystems, two line sensors in front, one object sensor in front and the motors attached on the side.

In order to generate a test plan, I installed three LED’S. Two of which depicts the working of the line sensors and the other was for the object sensing. A push button was implemented for the motors.

LED 1 = PD2 – for object sensing

LED 2= PD4 – for line sensor on the right

LED 3= PD7- for line sensor on the left

Push button=PB2- for the motors

**TROUBLESHOOT:**

If the sensors in the front don’t work, the first thing to do is check the connections. Make sure that all the wires are plugged in and are not tangled up. Also the distance of the line sensor should not be more than 2 inches from the ground. The connections for the sensors are PC0 and PC1.  
  
If the object sensor doesn’t work or the LED on PD2 doesn’t glow then, check the resistance value of the IR transmitter and also that the IR transmitter is connected to the PB1 pin. And the IR receiver is connected to PB0 pin. Also the connection for IR receiver is left most pin goes to PB0, the middle one goes to ground and the right most goes to +5V.

When the button is pushed, the wheels should go backward and if it doesn’t happen then make sure that the motor driver is in the working condition and the direction pins are connected to PB4 and PB5. Also the PWM pins are connected to PD5 and PD6.

**CODE:**

**#include** <avr/io.h>

**#include** <avr/interrupt.h>

**#include** <MSOE/delay.c>

**#include** <MSOE/bit.c>

**#include** <MSOE/lcd.c>

**void** **initPin** (**void**); // pin initialization, Timer configuration

**int** data,a,b;

**void** **sensor**();

**void** **motor\_configuration**(**void**);

**void** **backward**();

**#define** sw1 2

**#define** off 0

**#define** on 1

**int** **push1**();

**int** **main** (**void**)

{

initPin();

motor\_configuration();

sei();

**while**(1)

{

sensor();

**if**(push1()==on)

{

backward();

}

**else**

{

PORTD&=(~(1<<PORTD5))&(~(1<<PORTD6));

}

**if** (!data)

{

PORTD|=(1<<PD2);

}

**else**

{

PORTD &=~(1<<PD2);

}

**if**(b==0)

{

PORTD |=(1<<PD4);

}

**else**

{

PORTD &=~(1<<PD4);

**if**(a==0)

PORTD |=(1<<PD7);

**else**

{

PORTD &=~(1<<PD7);

}}

}

}

**void** **initPin** (**void**)

{

DDRB|=(1<<PB1);

DDRB &= ~(1<<PB0);

DDRB &= ~(1<<sw1);

DDRD|=((1<<PD2)|(1<<PD7)|(1<<PD4)|(1<<PD5)|(1<<PD6)); // set PD3 as output

PORTB |=(1<<sw1);

TCCR1A = 0x00; // avoid mess during the program loading

TCCR1B = 0x00;

TCCR0A = 0x00; // avoid mess during the program loading

TCCR0B = 0x00;

TCCR1B|=(1<<WGM12);// timer operate in CTC mode

TCCR1B|=(1<<CS10); // set prescaler to 1, desired frequency 37kHz

OCR1A = 210; // set TOP of Timer1 as 210

TCCR1A|=(1<<COM1A0); // set toggle on PB1

TCCR2A|=(1<<WGM21)|(1<<WGM20);// set fast PWM to timer2

TCCR2B|=(1<<WGM22);

TCCR2A|=(1<<COM2B1);// noninverting mode

OCR2A=255; // set frequency

TCCR2B|=(1<<CS22)|(1<<CS21)|(1<<CS20); //set prescaler to 256

OCR2B=6; // set duty cycle

TIMSK2|=(1<<OCIE2B)|(1<<OCIE2A); // enable interrupt when compare match with OCR2A/B occurs

}

ISR(**TIMER2\_COMPA\_vect**)

{

TCCR1B|=(1<<CS10); // start time counter

TCCR1A|=(1<<COM1A0); // connect timer to pin

}

ISR(**TIMER2\_COMPB\_vect**)

{

PORTB &= (~(1<<PB1)); // set PB1 LOW

TCCR1B &=(~(1<<CS10)); // stop time counter

TCCR1A &=~(1<<COM1A0); // disconnect timer from pin

data = (PINB&(1<<PB0)) ;

}

**void** **sensor**()

{

//Port initialization as output C register

DDRC |= (1<<PC1) | (1<<PC0);

PORTC |= (1<<PC0)|(1<<PC1);

//delay 10 micro seconds

**delay\_us**(10);

//Port initialization as input C register and pull-up resistor is off

DDRC = 0;

PORTC =0;

//delay 7 milli seconds

**delay\_ms**(7);

//read PINC value attached to 1st sensor

a= PINC & (1<<PC0);

b= PINC & (1<<PC1);

}

**void** **motor\_configuration**()

{

DDRD|=(1<<PORTD5)|(1<<PORTD6);//outputs pin5 and 6

PORTD&=(~(1<<PORTD5))&(~(1<<PORTD6));//disable robot

DDRB|=(1<<PORTB4)|(1<<PORTB5);

}

**void** **backward**()

{

PORTD|=(1<<PORTD5)|(1<<PORTD6);//enable robot

PORTB|=(1<<PORTB4);

PORTB &=(~(1<<PORTB5));

}

**int** **push1**(**void**)

{

**if**((PINB&(1<<sw1))==(1<<sw1))

{

**return** off;

}

**else**

{

**return** on;

}

}