K. J. Somaiya College of Engineering, Mumbai-77
(An Autonomous College Affiliated to University of Mumbai)

INTERNSHIP PROJECT REPORT

2018-19

Different line following path with object sensing

Name of Students

Shrey Shah (1613042)

Surya Maheswar (1613048)

Sahil Rajpurkar (1613130)

Internship In-charge

1) Mahesh Warang Sir
2) Parul Ma'am

Topic: Line following using object (colour) detection.

01. Introduction

The system consists of Fire Bird V ATMEGA2560 robot which is a microcontroller-based robot used for any kind of robotic application. This was designed by IIT-Bombay along with MHRD. It is a "Open source doctrine" in its software and hardware design therefore it would be compatible with any software like Keil, Matlab, Scilab and LabVIEW. It has the name Fire Bird V because it the 5th in its series. It has 2 Atmega controller where it acts as a master and a slave controller. The basic components in this robot is that it has IR sensors, proximity sensors, white line sensor, SHARP IR sensor, buzzers, LCD and LED along with Ni-MH battery. The two Atmega controllers are the master control is the Atmega 2560 and the slave control is the Atmega 8 controller.

Fig1:Basic structure of Fire Bird V Robot

Components

- **Atmega 2560**
- **3 IR sensors (2,3,4)**
- **LCD display**
- **White line sensors**
- **L2938**

INTERFACING:

In this project we have used Atmel studio software to program the Atmega 2560 which is the microcontroller used for the control of the bot. The registers that are used are described below. In this Project there are 3 Line following sensors interfaced on the F port and also 3 other IR sensors are interfaced on the same port for sensing the object. The motors are interfaced on the PortA and PWM is of Timer 5 which is on the L port.

Registers PINS:

PORT CONFIG:
DDRA
DDRF

Timer Initialization:
TCCR5A
TCCR5B

Pulse with Modulation:
OCR5A
OCR5B

ADC for converting sensor values:

ADCSRB: ENABLE MUX5
ADMUX

ADCSRA: CONVERSION BIT

Pin Connection:

Motor speed and direction:
PortA 0, 1, 2, 3 direction pins
PortL 3, 4 PWM pins

Sensors:
PortF 5, 6, 7 IR sensors
PortF 1, 2, 3 Line following sensors.

1. Program:

```
/*
 * ADC.c
 *
 * Created: 21-06-2018 12:17:24
 * Author : Shrey shah
 */

#define F_CPU 14745600UL
#include <avr/io.h>
#include <util/delay.h>
#include "lcd.h"
int s;
void full_left();
void full_right();
int cnt;
void config()
{
DDRF =0x00;
PORTF=0x00;
DDRA |=0x0F;
PORTA &=0xF0;
DDRL |=0x18;
}

void fwd()
{
PORTA &=0xF0;
PORTA |=0x06;
OCR5A =255;
OCR5B =230;
}
void fwd_left()
{
PORTA &=0xF0;
```

```
PORTA |=0x06;
OCR5A =150;
OCR5B =220;
}

void fwd_right()
{
PORTA &=0xF0;
PORTA |=0x06;
OCR5A =235;
OCR5B=150;
}
void full_right()
{
PORTA=0b00000110;  //0x02;
OCR5A=235;
OCR5B=40;
}
void full_left()
{
PORTA=0b00001001;  //0x02;
OCR5A=50;
OCR5B=235;
}
void stop()
{
PORTA &=0xF0;
PORTA |= 0x00;
}

void timer_init()
{
TCCR5A |=(1<<WGM50)|(1<<COM5A1)|(1<<COM5B1);
TCCR5B |=(1<<WGM52)|(1<<CS50)|(1<<CS52);
}

void adc_init()
{
ADMUX|=(1<<REFS0);//ref is equal to vcc
ADCSRA|=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);//prescaler 128
}

unsigned char ADC_Conversion(unsigned char Ch)
{
unsigned char a;
if(Ch>7)
{
ADCSRB = 0x08;
}
Ch = Ch & 0x07;
```

```
ADMUX= 0x20 | Ch;
ADCSRA = ADCSRA | 0x40; //Set start conversion bit
while((ADCSRA&0x10)==0); //Wait for ADC conversion to complete
a=ADCH;
ADCSRA = ADCSRA|0x10; //clear ADIF (ADC Interrupt Flag) by writing 1 to it
ADCSRB = 0x00;
return a;
}
int main(void)
{
config();
adc_init();
init_ports();
lcd_init();
timer_init();
    while (1)
    {
int a=ADC_Conversion(1);
        int b=ADC_Conversion(2);
int c=ADC_Conversion(3);
int d=ADC_Conversion(5);
lcd_print(1,1,c,3);
lcd_print(1,5,b,3);
lcd_print(1,9,a,3);
lcd_print(2,1,d,3);


if(d<70)
{
 if(b<15 && a>=25 && c>=25)
 {
 fwd();
 cnt=0;
     }
 else if (c<15)
 {
 fwd_left();
 }
 else if (a<15 )
 {
 fwd_right();
 }
  else if(a<15 && b<15)
  {
   cnt = 1;
  }
  else if(c<15 && b<15)
  {
      cnt =2;
      }
```

```
  else if (a>25 && b>25 && c>25)
  {
  if(cnt == 1)
  {
  full_left();
  }
  else
  {
  stop();
  }
   if (cnt =2)
  {
  full_right();
  }
  else
  {
 stop();
  }
}
}
else
{
  stop();
}
}
}
```

2. If we attach TCS colour sensor at adc pin 9(externally)

```
/*
 *Fire_bird_project.c
 *
 * Created: 21-06-2018 12:17:24
 * Author : Sahil
 */

#define F_CPU 14745600UL
#include <avr/io.h>
#include <util/delay.h>
#include "lcd.h"

int s_mode=0,a,b,c,d;

void port_init();
void fwd();
void fwd_left();
void fwd_right();
void stop();
void timer_init();
unsigned char ADC_Conversion(unsigned char Ch);
void adc_init();
```

```
void linefollow();
void readprint();
void process();

int main(void)
{
	port_init();
	adc_init();
	init_ports();
	lcd_init();
	timer_init();

   while (1)
   {
		readprint();
		process();
   }
}

void port_init()
{
	DDRF =0x00;
	PORTF=0x00;
	DDRA |=0x0F;
	PORTA &=0xF0;
	DDRL |=0x18;
}

void process()
{
	switch (s_mode)
		{
			case 0:
			if (d<250)
			{
				s_mode=1;
			}
			else if(d>250 && d<512)
			{
				s_mode=2;
			}
			break;
			case 1:
			linefollow();
			break;
			case 2:
			fwd_left();
			_delay_ms(1000);
			if(a)
			{
			linefollow();
			}
```

```
                        break;
                }
}

void fwd()
{
      PORTA &=0xF0;
      PORTA |=0x06;
      OCR5A =255;
      OCR5B =230;
}
void fwd_left()
{
      PORTA &=0xF0;
      PORTA |=0x06;
      OCR5A =150;
      OCR5B =220;
}

void fwd_right()
{
      PORTA &=0xF0;
      PORTA |=0x06;
      OCR5A =235;
      OCR5B=150;
}
void stop()
{
      PORTA &=0xF0;
      PORTA |= 0x00;
}

void timer_init()
{
      TCCR5A |=(1<<WGM50)|(1<<COM5A1)|(1<<COM5B1);
      TCCR5B |=(1<<WGM52)|(1<<CS50)|(1<<CS52);
}

void adc_init()
{
      ADMUX|=(1<<REFS0);//ref is equal to vcc
      ADCSRA|=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);//prescaler 128
}

unsigned char ADC_Conversion(unsigned char Ch)
{
      unsigned char a;
      if(Ch>7)
      {
            ADCSRB = 0x08;
      }
      Ch = Ch & 0x07;
```

```
	ADMUX= 0x20 | Ch;

	ADCSRA = ADCSRA | 0x40; //Set start conversion bit
	while((ADCSRA&0x10)==0); //Wait for ADC conversion to complete
	a=ADCH;
	ADCSRA = ADCSRA|0x10; //clear ADIF (ADC Interrupt Flag) by writing 1 to it
	ADCSRB = 0x00;
	return a;
}

void linefollow()
{
	if(b<15 && a>=25 && c>=25)
	{
		fwd();
	}
	else if (c<15)
	{
		fwd_left();
	}
	else if (a<15 )
	{
		fwd_right();
	}
	else if (a>25 && b>25 && c>25)
	{
		stop();
	}
}

void readprint()
{
		a=ADC_Conversion(1);
		b=ADC_Conversion(2);
		c=ADC_Conversion(3);
		d=ADC_Conversion(9);
		lcd_print(1,1,c,3);
		lcd_print(1,5,b,3);
		lcd_print(1,9,a,3);
		lcd_print(2,1,d,3);
}
```

Algorithm:

1) Initialize function includes all port initialization of the chip (Atmel 2560).
2) Start the timer function which is used for controlling the speed of bot.
3) Write the ADC function for getting sensor reading.
4) Write stop ADC function for getting the final data from the 4 sensor used for project.
5) Store each reading in different variables.
6) Now compare the line following sensor reading with threshold readings.
7) With different condition check for left right straight and stop condition.
8) For all the different conditions we need to have Different PWM values accordingly to which the bot follows the line.
9) In the main function the whole line following code will be inside the 4^{th} sensors reading (IR sensor), which helps to detect the object placed.

REFERENCES

Examples of writing references is given below. <11 point>

[1]. V. P. Gountis and A. G. Bakirtzis, "Bidding strategies for electricity producers in a competitive electricity marketplace," IEEE Trans. Power System, vol. 19, no. 1, pp. 356–365, Feb. 2004. [2]. J. Clerk Maxwell, "A Treatise on Electricity and Magnetism", 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.