Peripheral and Interfacing Lab report

Topic: Line Follower Robot With 5 Sensors

Submitted by:
Md. Rahaduzzaman
Roll No: 12
Abu Mohammed Raisuddin
Roll No: 18
Sufi Khan
Roll No: 84
3rd Year, Session: 2008-09
Department of Computer Science and Engineering
University of Dhaka

Submitted to:
Ahsan Raja Chowdhury
Lecturer
&
Mahmudul Hasan
Lecturer
Department of Computer Science and Engineering
University of Dhaka

Content

Introduction:

What is a line follower robot: A line follower robot is a machine that can follow a line. Generally the line is black on white surface. But different color combination can be possible. Sensing a line and maneuvering the robot to stay on course, while constantly correcting wrong moves using feedback mechanism forms a simple yet effective closed loop system, as a programmer we get an opportunity to 'teach' the robot how to follow the line thus giving it a human-like property of responding to stimuli.

What is AVR microcontroller: Atmel's AVR® microcontrollers have a RISC core running single cycle instructions and a well-defined I/O structure that limits the need for external components. Internal oscillators, timers, UART, SPI, pull-up resistors, pulse width modulation, ADC, analog comparator and watch-dog timers are some of the features that can be found in AVR devices. AVR instructions are tuned to decrease the size of the program whether the code is written in C or Assembly or any other language. With on-chip in-system programmable Flash and EEPROM, the AVR is a perfect choice in order to optimize cost and get product quickly.

Apart from this almost all AVRs support **In System Programming** (ISP) i.e. we can reprogram it without removing it from the circuit. This comes very handy when prototyping a design or upgrading a built-up system. Also the programmer used for ISP is easier to build compared to the parallel programmer required for many old uCs. Most AVR chips also support **Boot Loaders** which take the idea of In System Programming to a new level.

Features like **I^2C** bus interface make adding external devices a cakewalk. While most popular uCs require at least a few external components like crystal, caps and pull-up resistors, with AVR the number can be as low as zero!

Equipments and prerequisite:

Equipments:

1. NextSAPIENS Development Boards with on board IC's
2. 5 light sensors (IR)
3. 16*2 LCD Display
4. Atmega16 microcontroller
5. USB burner for development board
6. AVR DUDE software.
7. BASCOM (BASIC compiler)

Prerequisites:

1. Knowledge of basic digital and analog electronics
2. Knowledge of BASIC language

Architectural overview of the line follower robot:

nextSAPIENS Board Features:

The NEXTSAPIENS Development Board Features are:-

1. 40 Pin Atmel ATmega16/32 microcontroller with internal system clock upto 8 MHz and externally upto 16 MHz
2. 16/32 KB FlashRAM memory for programs
3. 1/2 KB of SRAM

4. 512/1024 Bytes of EEPROM
5. One 6x1 Pin SPI Relimate Header
6. Eight 3x1 Pin Relimate header inputs for 8 analog sensors
7. One 16 Pin header to connect 16*2 alphanumeric LCD
8. Two onboard L293D drivers for motors (upto 600 mA per channel)
9. Dual 7805 Voltage regulator
10. Dual power input options (Through molex connector or through DC Jack)
11. Two programmable MicroSwitches
12. Two programmable LEDs
13. Two DPDT switches (one for power on/off and one for reset)
14. MAX 232 Level shifter for RS232 communication
15. One 3x1 Pin relimate header for RS2332 communication
16. Four 8 Pin bergistick headers (male) from each port of ATmega16/32
17. Wide input power range from 7 volts to 24 volts at 1.52 Amps
18. Board size of 6 x 3 inches, designed for educational and hobby purpose, on high quality PCB

Motor Connection:

MOTOR1
PWM Channel = PWM1B
Direction Bit = PortD.3
MOTOR 2
PWM Channel = PWM1A
Direction Bit = PortD.6

Sensor Connection:

J12 = PortA.7 (or) ADC(7)
J13 = PortA.5 (or) ADC(5)
J14 = PortA.3 (or) ADC(3)
J15 = PortA.1 (or) ADC(1)
J17 = PortA.6 (or) ADC(6)
J16 = PortA.4 (or) ADC(4)
J18 = PortA.2 (or) ADC(2)
J19 = PortA.0 (or) ADC(0)

ATmega16's Pin configuration:

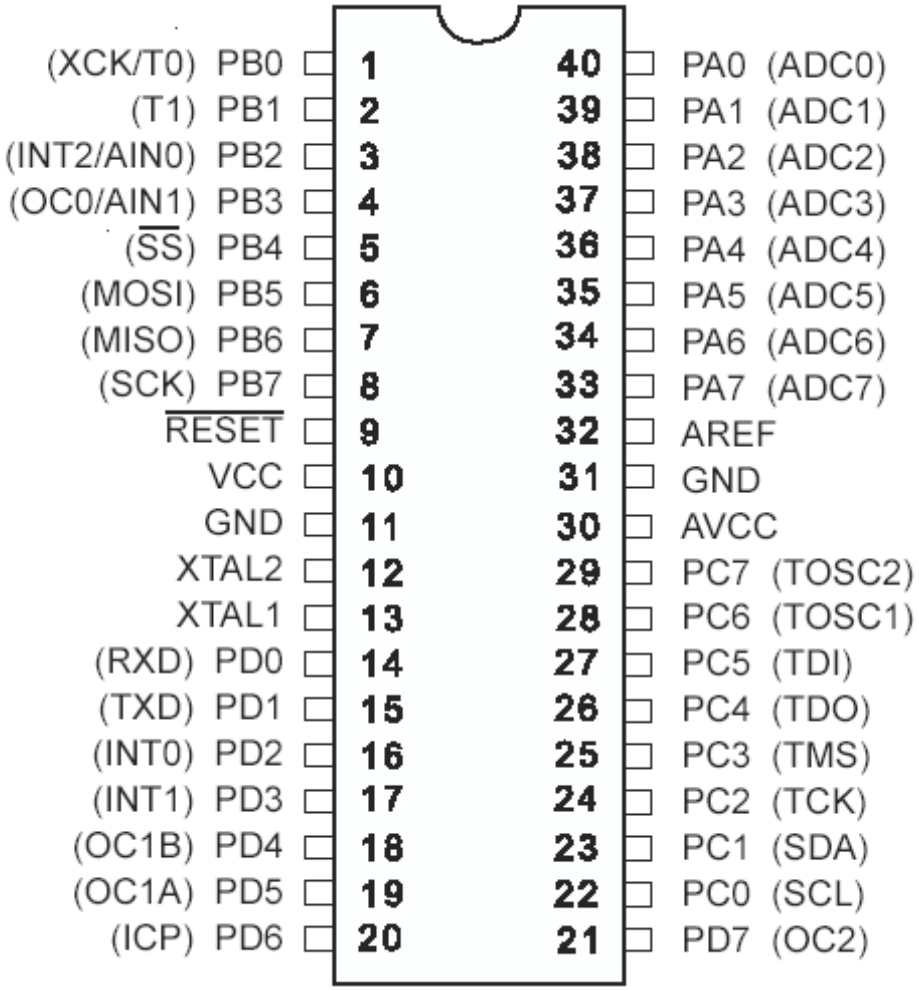

Figure: ATmega16 microcontroller's pin configuration

How it works:

The provided nextSAPIENS development board contains most of the required components we need. The motors and sensor are set. All we have to do is to get reading from the sensor and compare them with pre set data. For our assignment, the robot is to move on black line. So if the sensor detects black line it will move. If it detects that the black line is taking turn, the robot will take turn to move along with the line. If the robot senses no black line it will stop.

Implementation:

1. Connect the 5 sensors to sensor connector # J19, # J15, #J18, #J14, #J16
2. Connect the motor cable to the motor connectors.
3. Compile the code below and generate the hex file.
4. Connect the burner to the PC's USB port and ISP header on the development board.
5. Run the provided avrdude_gui program to burn the hex file to the Atmega16 microcontroller.
6. After burning the hex file to the microcontroller, unplug the burner connector.
7. Place the robot on white place to take reading form sensor for white place and then place on the black place and take reading of sensors.
8. When black and white configuration is complete, place the robot on the black line of the track so that it could start moving.

Source Code:

```
$regfile = "m16def.dat"
$crystal = 4000000

Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = Portb.4 , Db5 = Portb.5 , Db6 = Portb.6 , Db7 = Portb.7 , E = Portb.3 , Rs = Portb.2
Config Adc = Single , Prescaler = Auto , Reference = Avcc
Config Timer1 = Pwm , Pwm = 8 , Prescale = 1 , Compare A Pwm = Clear Down , Compare B Pwm = Clear Down

Config Porta = Input
Dim Aw As Integer
Dim Bw As Integer
Dim Cw As Integer
Dim Dw As Integer
Dim Ew As Integer
Dim Ab As Integer
Dim Bb As Integer
Dim Cb As Integer
Dim Db As Integer
Dim Eb As Integer

Dim Aavg As Integer
Dim Bavg As Integer
Dim Cavg As Integer
Dim Davg As Integer
Dim Eavg As Integer

Dim A As Integer
Dim B As Integer
Dim C As Integer
Dim D As Integer
```

```
Dim E As Integer
Cls
Start Adc
Lcd "Place on white"
Waitms 2000
Cls
Lcd "configuring..."
Aw = Getadc(0)
Bw = Getadc(1)
Cw = Getadc(2)
Dw = Getadc(3)
Ew = Getadc(4)
Waitms 1000
Cls
Cls
Lcd Aw ; " " ; Bw ; " " ; Cw
Lowerline
Lcd Dw ; " " ; Ew
Waitms 2000
Cls
Lcd "Place on black"
Waitms 2000
Cls
Lcd "configuring..."
Ab = Getadc(0)
Bb = Getadc(1)
Cb = Getadc(2)
Db = Getadc(3)
Eb = Getadc(4)
Waitms 1000
Cls
Lcd Ab ; " " ; Bb ; " " ; Cb
Lowerline
Lcd Db ; " " ; Eb
Waitms 2000
Aavg = Aw + Ab
Aavg = Aavg / 2
Bavg = Bw + Bb
Bavg = Bavg / 2
Cavg = Cw + Cb
Cavg = Cavg / 2
Davg = Dw + Db
Davg = Davg / 2
Eavg = Ew + Eb
Eavg = Eavg / 2

Do
Cls
```

```
A = Getadc(0)
B = Getadc(1)
C = Getadc(2)
D = Getadc(3)
E = Getadc(4)


If A < Aavg And B < Bavg And C < Cavg And D < Davg And E < Eavg Then
Pwm1a = 0
Portd.3 = 0
Pwm1b = 0
Portd.6 = 0

Elseif A > Aavg And B > Bavg And C > Cavg And D > Davg And E > Eavg Then
Pwm1a = 100
Portd.3 = 0
Pwm1b = 100
Portd.6 = 0

Elseif A < Aavg And B < Bavg And C > Cavg And D > Davg And E > Eavg Then
Pwm1a = 200
Portd.3 = 0
Pwm1b = 200
Portd.6 = 1

Elseif A > Aavg And B > Bavg And C > Cavg And D < Davg And E < Eavg Then
Pwm1a = 200
Portd.3 = 1
Pwm1b = 200
Portd.6 = 0

Elseif A < Aavg And B < Bavg And C > Cavg And D < Davg And E < Eavg Then
Pwm1a = 100
Portd.3 = 0
Pwm1b = 100
Portd.6 = 0

Elseif A < Aavg And B < Bavg And C > Cavg And D > Davg And E < Eavg Then
Pwm1a = 0
Portd.3 = 0
Pwm1b = 100
Portd.6 = 0

Elseif A < Aavg And B > Bavg And C > Cavg And D < Davg And E < Eavg Then
Pwm1a = 100
Portd.3 = 0
Pwm1b = 0
Portd.6 = 0
```

```
Elseif D > Davg Or E > Eavg Then
Pwm1a = 200
Portd.3 = 0
Pwm1b = 100
Portd.6 = 1

Elseif A > Aavg Or B > Eavg Then
Pwm1a = 100
Portd.3 = 1
Pwm1b = 200
Portd.6 = 0
End If

Loop

End
```

Possible Improvements:

1. Use of differential steering with gradual change in wheel speeds.
2. Use of Hysteresis in sensor circuit using LM339
3. Use of ADC so that the exact position of the line can be interpolated
4. Use of Wheel Chair or three wheel drive to reduce traction.
5. General improvements like using a low dropout voltage regulator, lighter chassis etc

References:

The soft copy of nextSAPIENS board manual from the twelve batch's guy. A special thanks to twelve batch for their help in this assignment.

Conclusion:

The core of the line follower is the ATmega16 microcontroller. So it operates the motor in the way we program it. A clean correct program and faultless sensor and motor can run a line follower robot correctly.