Programming Mobile Robots in ANSI C Language for PIC MCU's

János Simon, Tibor Szakáll, Zlatko Čović

Department of Informatics, Polytechnical Engineering College, Subotica
simon@vts.su.ac.yu
szakall.tibor@gmail.com
chole@vts.su.ac.yu

Abstract: This article presents small PIC-controlled robot vehicle using stepper motors. The basic idea is to use two steppers which are directly connected to the wheels and a few type of sensors for environment sensing. It can be programmed in C programming language to navigate on its own two driven wheel in appropriate surroundings. Developed algorithms can be tested in real environment. The goal of this work is to get students interested in the fields of engineering, robotics, and software development for a mobile robot.

1 Introduction

As technology is progressing, people tend to use more and more intelligent machines and devices which will lighten their daily chores and responsibilities. It is common knowledge that such devices are capable of performing a limited set of tasks and are only as 'intelligent' as their programmers. There are endless possibilities for the application of these intelligent automotive systems, with the demand constantly on the rise. The popularization of such technologies and the use of developing environments leads to the creation of higher quality software, and thus, to the appearance of ever more intelligent mobile robots.

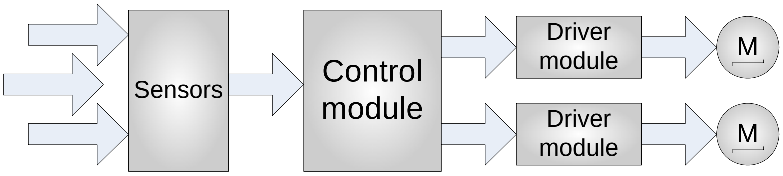


Figure 1
Block scheme of a mobile robot

Figure 1 shows a typical block scheme of a mobile robot with two wheels. Such systems are made up of a group of sensors, a control unit and an actuator. Data collection from the real world is done by sensors while actuators are used to carry out certain actions.

2 Control Module

The core of the system is made up of the PIC controller which is responsible for the navigation of the mobile robot and for communication over the USB port of the computer when needed.

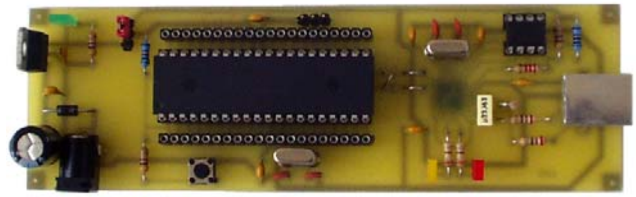


Figure 2
Control module

Figure 2 depicts the control module. In order to achieve an acceptable level of control over the mobile robot the microcontroller receives the signals from the sensors, and sends the right series of impulses to the step motors. The microcontrollers, the PIC16F877 at 20MHz and PIC18F452 at 40MHz are supported by the system. The USB interface is realized with the help of a FT232BM transceiver. The developing environment supports the assembler and the C programming language. These make it possible to write effective software and a readable code. Thanks to the carefully chosen components with this relatively low-priced module the result is a satisfactory platform for software development for the purpose of controlling the mobile robot on wheels.

Figure 3 shows a Protel schematic of the control module which is based on a PIC16F877 and FT232BM integrated circuits.

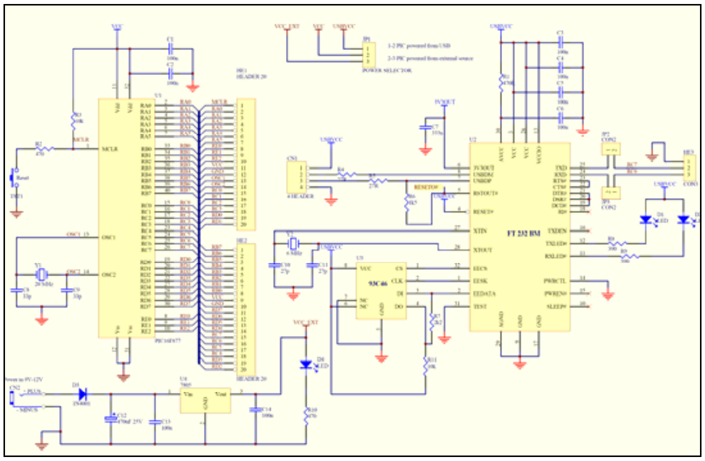


Figure 3
Control module schematic

3 Stepper Motors and Driver Circuits

A stepper motor is an electromechanical device which converts electrical pulses into discrete mechanical movements. The shaft of a stepper motor rotates in discrete step increments when electrical command pulses are applied to it in the proper sequence. The motors rotation has several direct relationships to these applied input pulses. The sequence of the applied pulses is directly related to the direction of motor shafts rotation. The speed of the motor shafts rotation is directly related to the frequency of the input pulses and the length of rotation is directly related to the number of input pulses applied. A stepper motor can be a good choice whenever controlled movement is required. They can be used to advantage in applications where you need to control rotation angle, speed and position.

3.1 Bipolar Stepper Motors

Bipolar motors are known for their excellent size/torque ratio, and provide more torque for their size than unipolar motors. Unlike unipolar stepper motors, bipolar units require more complex driver circuitry. Bipolar motors are designed with separate coils that need to be driven in either direction (the polarity needs to be reversed during operation) for proper stepping to occur. This presents a driver challenge. Bipolar stepper motors use the same binary drive pattern as a unipolar motor, only the '0' and '1' signals correspond to the polarity of the voltage applied to the coils, not simply 'on-off' signals. Figure 4 shows a basic 4-phase bipolar motor's coil setup.

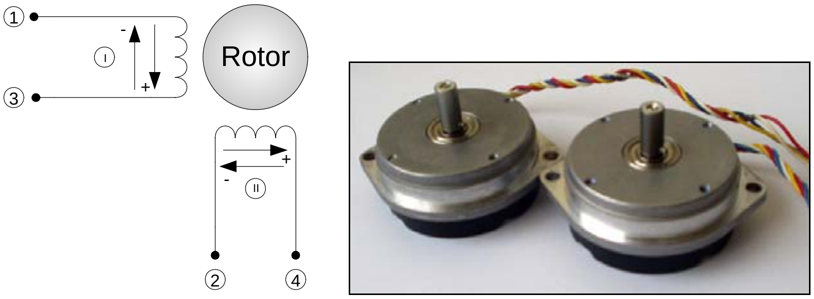


Figure 4

Bipolar stepper motors

A circuit known as an 'H-bridge' is used to drive bipolar stepper motors. Each coil of the stepper motor needs its own H-bridge driver circuit. Typical bipolar steppers have 4 leads, connected to two isolated coils in the motor. ICs specifically designed to drive bipolar steppers are available. The most popular are the L297/298 series from ST Microelectronics.

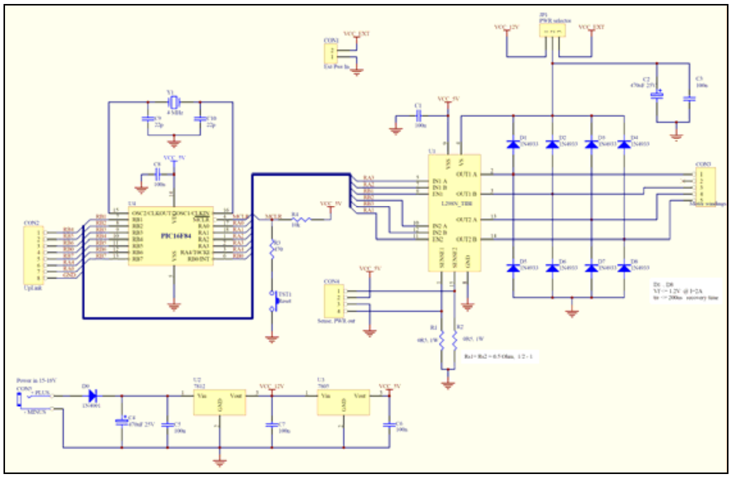


Figure 5

Driver module schematic

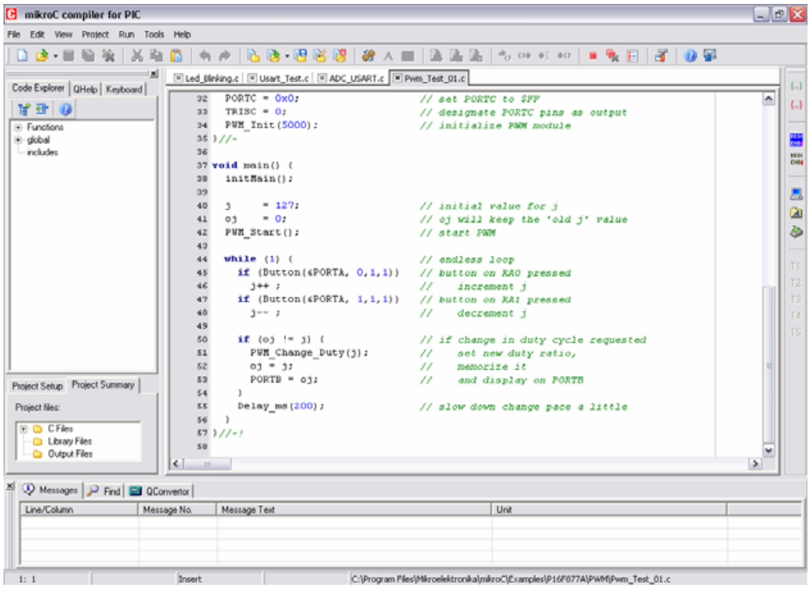
Figure 5 shows a Protel schematic of the H-bridge driver module using the L298 IC from ST Microelectronics in combination with PIC 16F84 instead of L297 logic driver.

4 The MikroC Development Tool

The mikroC is a powerful, feature rich development tool for PIC microcontrollers. It is designed to provide the developer with the easiest possible solution for developing applications for embedded systems, without compromising performance or control. Currently available libraries are:

- ADC Library
- CAN Library
- CANSPI Library
- Compact Flash Library
- Compact Flash FAT Library v2.xx
- Conversions Library
- EEPROM Library
- Ethernet Library
- Flash Memory Library
- Graphic LCD Library
- I2C Library
- Keypad Library
- LCD Library
- LCD8 Library
- Manchester Code Library
- Multi Media Card Library
- OneWire Library
- PS/2 Library
- PWM Library
- RS-485 Library
- Secure Digital Library
- Software I2C Library
- Software SPI Library
- Software UART Library
- Sound Library
- USART Library
- USB HID Library
- Util Library
- ANSI C Standard Libraries
- Conversions Library
- Trigonometry Library
- Sprint Library

PIC is one of the most popular 8-bit microcontroller in the world, used in a wide variety of applications, prized for its efficiency, is the natural choice for developing embedded systems. mikroC provides a successful match featuring highly advanced IDE, ANSI compliant compiler, broad set of hardware libraries, comprehensive documentation, and plenty of ready-to-run examples. mikroC provides a set of libraries which simplifies the initialization and use of PIC MCU and its modules. Library functions do not require any header files to be included, they can be used anywhere in the project. Figure 6 shows the mikroC developing environment.


mikroC compiler for PIC
File Edit View Project Run Tools Help
Code Explorer QHelp Keyboard
Functions
global
includes
Led_Blinking.c Usart_Test.c ADC_USART.c Pwm_Test_01.c
Project Setup Project Summary
Project files:
C Files
Library Files
Output Files
Messages Find QConvertor
Line/Column Message No. Message Text Unit
1: 1 Insert C:\Program Files\Mikroelektronika\mikroC\Examples\P16F877A\PWM\Pwm_Test_01.c

```
  PORTC = 0x0;                        // set PORTC to $FF
  TRISC = 0;                          // designate PORTC pins as output
  PWM_Init(5000);                     // initialize PWM module
}//~

void main() {
  initMain();

  j     = 127;                        // initial value for j
  oj    = 0;                          // oj will keep the 'old j' value
  PWM_Start();                        // start PWM

  while (1) {                         // endless loop
    if (Button(&PORTA, 0,1,1))        // button on RA0 pressed
      j++ ;                           //    increment j
    if (Button(&PORTA, 1,1,1))        // button on RA1 pressed
      j-- ;                           //    decrement j

    if (oj != j) {                    // if change in duty cycle requested
      PWM_Change_Duty(j);             //    set new duty ratio,
      oj = j;                         //    memorize it
      PORTB = oj;                     //    and display on PORTB
    }
    Delay_ms(200);                    // slow down change pace a little
  }
}//~!
```

Figure 6
MikroC environment

MikroC has plenty of examples included for development and to use as building bricks for any project. A few code snippets are given for an example.

```
unsigned short i;

void main()
{
  USART_init(2400);

  while (1)
  {
   if (USART_Data_Ready())
       {
       i = USART_Read();
       USART_Write(i);
       }
  }
}
```

A) USART example

```
void main()
{
 PORTB = 0;
 TRISB = 0;

 do
   {
   switch (PORTB)
   {
    case 0x00: PORTB = 0xFF; break;
    case 0xFF: PORTB = 0x00;
   }
   Delay_ms(1000);
   }
 while (1);
}
```

B) SWITCH structure example

```
char *text = "SISY 2006";

void main()
{
  LCD_Init(&PORTB);

  LCD_Cmd(LCD_CLEAR);

  LCD_Cmd(LCD_CURSOR_OFF);

  LCD_Out(1,1, text);

  Delay_ms(1000);

  LCD_Out(2,1,"Subotica");
}
```

C) LCD example

```
void main()
{
  char oldstate = 0;
  TRISB = 0xff;
  TRISD = 0;

do
{
  if (Button(&PORTB, 1, 1, 1))
    oldstate = 1;
  if (oldstate && Button(&PORTB, 1, 1, 0))
   {
    PORTD = ~PORTD;
    oldstate = 0;
   }
}
while(1);
}
```

D) BUTTON example

Conclusions

The process of designing, assembling and programming of mobile robots is the combination of mechanical, electronic and programming skills. Such developing environment enables a large degree of freedom in the choice of algorithms for the implementation and gives the user the possibility to compare the results of implementation in order to make the best possible choice of algorithm for a given task. The mechanical principle, implemented algorithm and electronics are very similar to large serious systems, the only difference being the size and sophistication of this system. Such developing environment may be extremely useful when acquiring the approach to a certain problem and the basic rules of programming mobile robots.

References

[1] MikroElektronika: MikroC - C Compiler for Microchip PIC microcontrollers, 2005

[2] Parallax Inc: Robotics with the Boe-Bot, 2004, ISBN 1-928982-03-4

[3] Myke Predko: Programming & Customizing PICmicro Microcontrollers, 2002 ISBN 0-07-136172-3

[4] Gyula Mester: Introduction to Control of Mobile Robots, 2006 ISBN 86-85525-01-2

[5] Simon János, Szakáll Tibor, Zlatko Čović: Environment for implementation and development of algorithms for movement of mobile robots on wheel, 2006, JISA - Belgrade