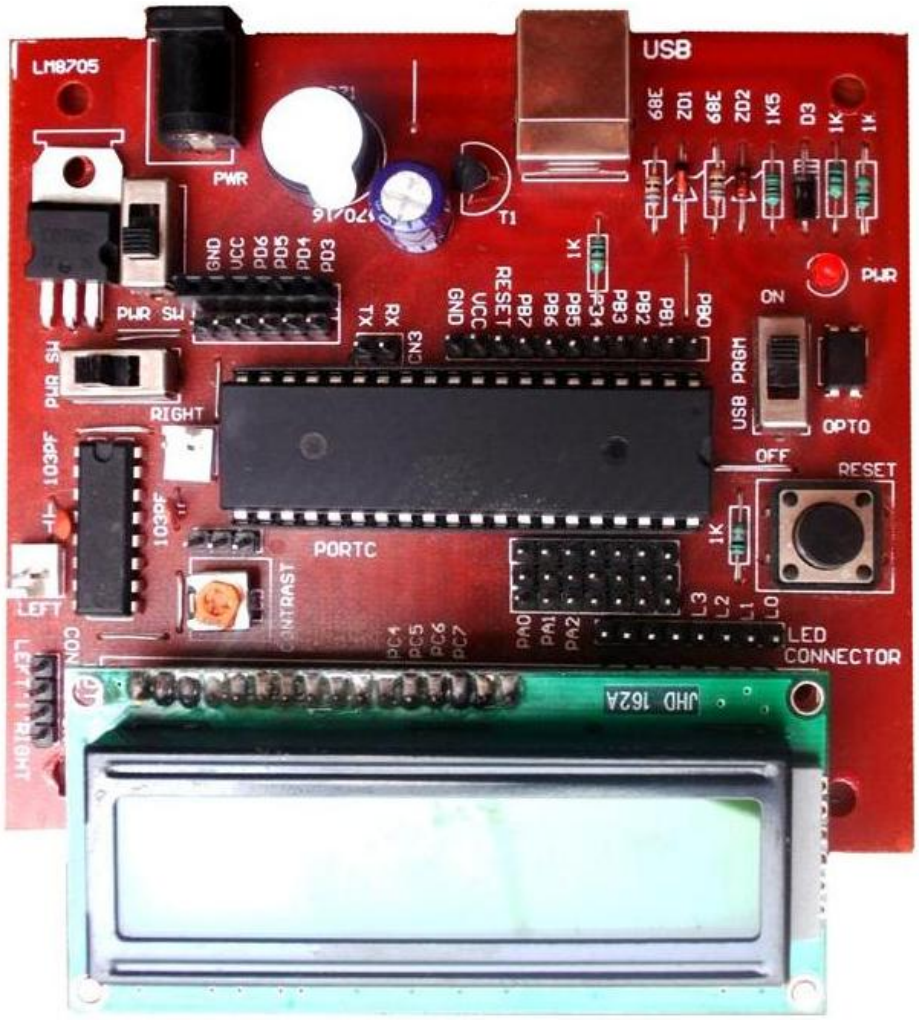
Email: admin@robomart.com
Contact: +91 - 8744000555

USER MANUAL FOR ATMEGA 16/32 ROBOTICS

MINI DEVELOPMENT BOARD V2.0

USING AVR STUDIO



Note: This tutorial is for the novice user. It shows how to set up AVR Studio 4.x and WinAVR so that C/C++ Source files will be compiled using GCC. For further information on AVR Studio, AVR Libc, GCC, AVR ISP, AVR JTAG ICE, and other tools please refer to the documentation which is also installed on the installation. Procedure described here is followed.

Email: admin@robomart.com
Contact: +91 - 8744000555

Table of Contents

1. Introduction 3
2. Features 4
3. Introductions to AVR Series (ATmega16) microcontroller 5
   3.1.Overview 5
   3.2.Pin Out of Atmega 16 5
   3.3.Pin Descriptions 6
   4.Introductions to Motor Driver 7
   4.1.H-Bridge 7
   4.2.L293D Dual H-Bridge Motor Driver 8
   4.3.Pin Diagram of L293D 8
   4.4.Interfacing 9
5. Block Diagram of Development Board Connected 10
6. Block Description of Development Board Connected 11
7. Precautions 12
8. Starting AVR Studio 13
   8.1.Writing your first C program 13
   8.2.Compilation of your first C program 14
9. Dumping Program in Development Board Connected 15
10. Connecting peripherals with Atmega 16/32 Robotics Mini Development Board V2.0 20
   10.1.DTMF Decoder module 20
   10.2.Sound Sensor 20
   10.3.Light Sensor 20
11. Study of some C program using Loops and variables 21
   11.1.Blinking LEDs 21
   11.2.Line follower Robot program 22

Email: admin@robomart.com
Contact: +91 - 8744000555

1. Introduction

Robots and robotic technologies have an intellectual and emotional appeal that transcends any other type of engineered product, and this appeal is felt no more so than with children and young adults. Robots and robotic technologies represent a practical application of physics, computer science, engineering, and mathematics, and provide a very powerful and flexible approach to demonstrate a variety of engineering concepts. In addition, robotics appeals to a broad range of interests and allows multiple points of access to science, mathematics, and engineering for many types of learners. As a result, robotic technology and robots are being used by an increasing number of educators at the college level to reinforce computer science and engineering theory, and to teach basic software and mechanical engineering at the grammar school, middle school and high school levels. The giant strides that we have made in the areas of Communications and Computers are possible only because of the great successes that we have achieved in the field of Electronics.

It is sometimes unbelievable, how many electronics gadgets that we carry these days in our person – Digital Wrist-watch, Calculator, Cell-phone, Digital Diary or a PDA, Digital Camera or a Video camera, etc. The different type of Electronic equipments that has invaded our offices and homes these days is also mind boggling. Many things we use at home and office are “remote controlled”, for example, Television (TV), Air-Conditioners, Audio equipment, Telephone, etc. It is almost close to “magic” how even a child, now-a-days, can switch channels, or increase decrease the volume of sound in a TV at home by just clicking on a few buttons sitting at the comfort of a sofa away from the Television apparently without any physical wiring or connection!

Again, we are astonished, how we are able to talk to our near and dear living several thousands of Kilometers away, from wherever we are, at home, office, on the road in a car, or in a classroom – by just clicking a few n umbers on our palm sized cellular phones! Electronics has made deep impact in several vital areas such as health care, medical diagnosis and Treatment, Air and space travels, Automobiles, etc. In short, the technological developments of several countries of the globe are directly related to their strengths in electronics design, manufacture, products and services. Just as we teach physics, chemistry, biology and mathematics in our schools, it is high time we start teaching our children at school, Electronics as a separate subject by itself. This brings us face to face to an important question: How to teach the basic concepts of such an important subject like Electronics most effectively? If one wants to gain a good understanding of electronics, he or she should build circuits and test them independently. For this one should acquire a practical knowledge of the characteristics of different devices and in constructing the various circuits. Let us try to learn such skills by the proven scheme of **“LEARNING BY DOING”.** There is only one way to learn to do anything: and I found Robotics is the best way to learn all this.

So let’s start doing Robotics!!!

Email: admin@robomart.com
Contact: +91 - 8744000555

2. Features

- Programming through USB or 6-Pin ISP (In System Programming) cable.
- Driver Software not required (Plug& Play device).
- Four ports available for user interface.
    - PORTA-8 Pin, PORTB-8 Pin, PORTC-8 Pin, PORTD-8 Pin.
- Port for USART communication.
- 7 Channel 10 Bit ADC Port.
- External Reset Switch.
- On board crystal oscillator of 12 MHz frequency.
- Two supply inputs-
    - One through Battery and Second through Adapter.
- On board 5 volt regulated power supply from DC socket for the board and external peripherals.
- Two H-Bridge implemented on board using L293D IC to drive two DC motor with 1A drive current at voltages 4.5 to 36 volt.
- On board Slider Switch for Enabling Motors.
- On board HXD ®Piezo Buzzer for audio interface with the user. On board 16 x 2 LCD interface with variable Contrast.
- On board 8 bit LEDs for quick monitoring of Port Input / Output.
- Dedicated DTMF decoder module and Light searching Sensor Add on Shield.
- On Board 7 Bit Sensor interface for Robosapiens Sensor Module.
- On board Slider Switch for Boot Mode selection.

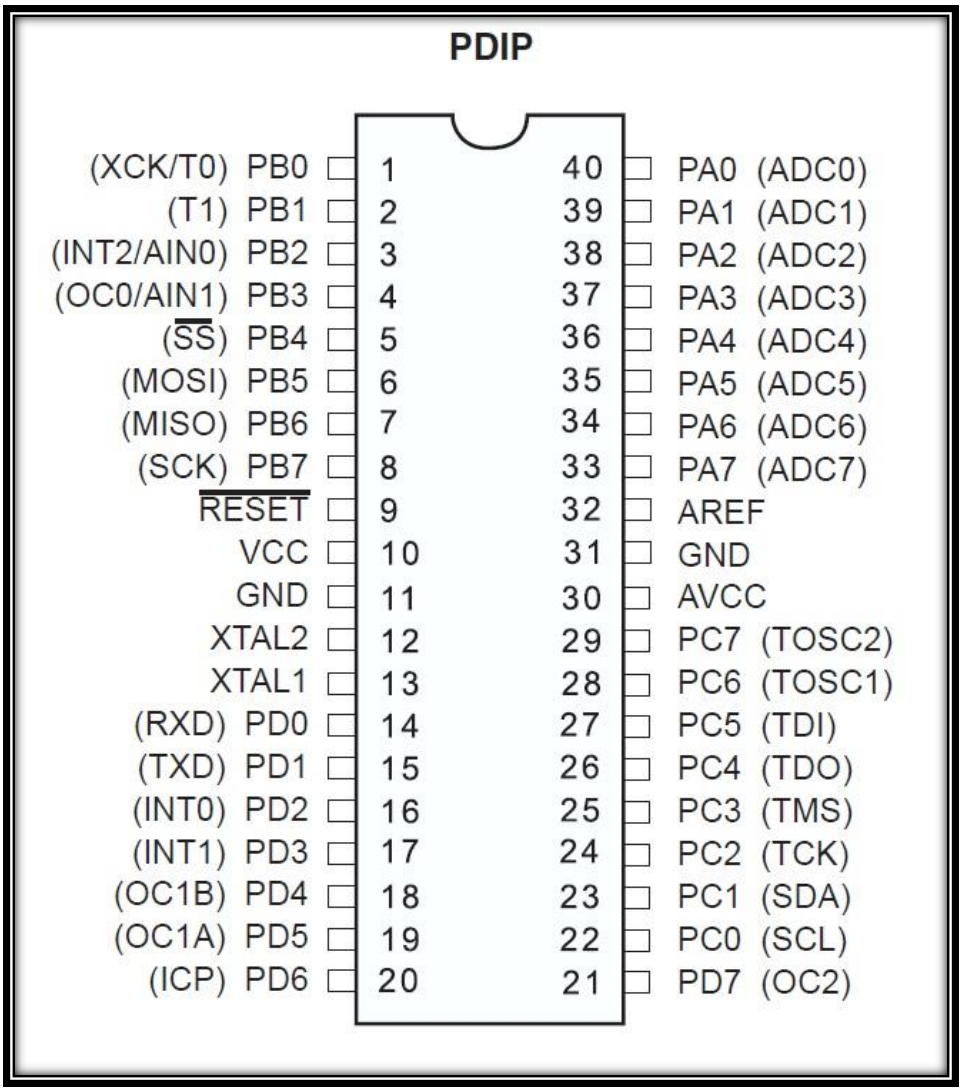
Email: admin@robomart.com
Contact: +91 - 8744000555

3. Introductions to AVR Series (ATmega16) microcontroller

3.1. Overview

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced **RISC** architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 16 MIPS per 16 MHz allowing the designer to optimize power consumption versus processing speed.

3.2. Pin Out of Atmega 16



Email: admin@robomart.com
Contact: +91 - 8744000555

3.3. Pin Descriptions

VCC: Digital supply voltage.

GND: Ground.

Port A (PA7-PA0): Port A is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port A also serves the functions of various special features of the ATmega16.

Port B (PB7-PB0): Port B is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special features of the ATmega16.

Port C (PC7-PC0): Port C is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5 (TDI), PC3 (TMS) and PC2 (TCK) will be activated even if a reset occurs.

Port D (PD7-PD0): Port D is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the ATmega16.

RESET: Reset Input. A low level on this pin for longer than the minimum pulse
Length will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

XTAL1: Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2: Output from the inverting Oscillator amplifier.

Email: admin@robomart.com
Contact: +91 - 8744000555

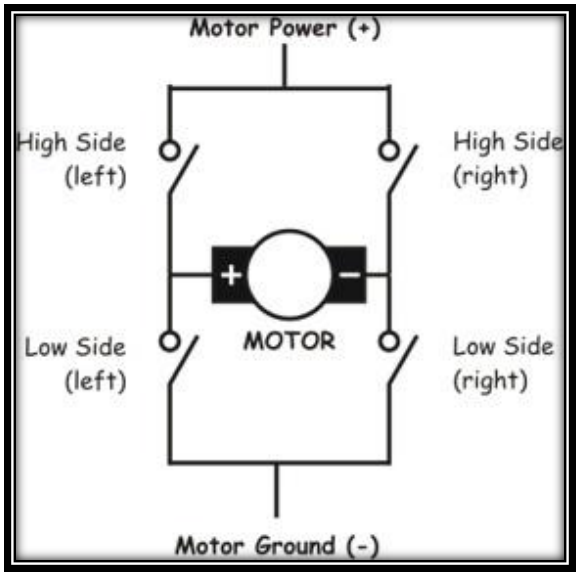
AVCC: AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.

AREF: AREF is the analog reference pin for the A/D Converter.

4. Introductions to Motor Driver

Whenever a robotics hobbyist talk about making a robot, the first thing comes to his mind is making the robot move on the ground. And there are always two options in front of the designer whether to use a DC Motor or a stepper motor. When it comes to speed, weight, size, cost... DC motors are always preferred over stepper motors. There are many things which you can do with your DC motor when interfaced with a microcontroller. For example you can control the speed of motor, you can control the direction of rotation, you can also do encoding of the rotation made by DC motor i.e. keeping track of how many turns are made by your motors etc. So you can see DC motors are no less than a stepper motor. In this part of tutorial we will learn to interface a DC motor with a microcontroller. Usually H-bridge is preferred way of interfacing a DC motor. These days many IC manufacturers have H-bridge motor drivers available in the market like L293D is most used H-Bridge driver IC. H-bridge can also be made with the help of transistors and MOSFETs etc. rather of being cheap, they only increase the size of the design board, which is sometimes not required so using a small 16 pin IC is preferred for this purpose.

4.1. H-Bridge

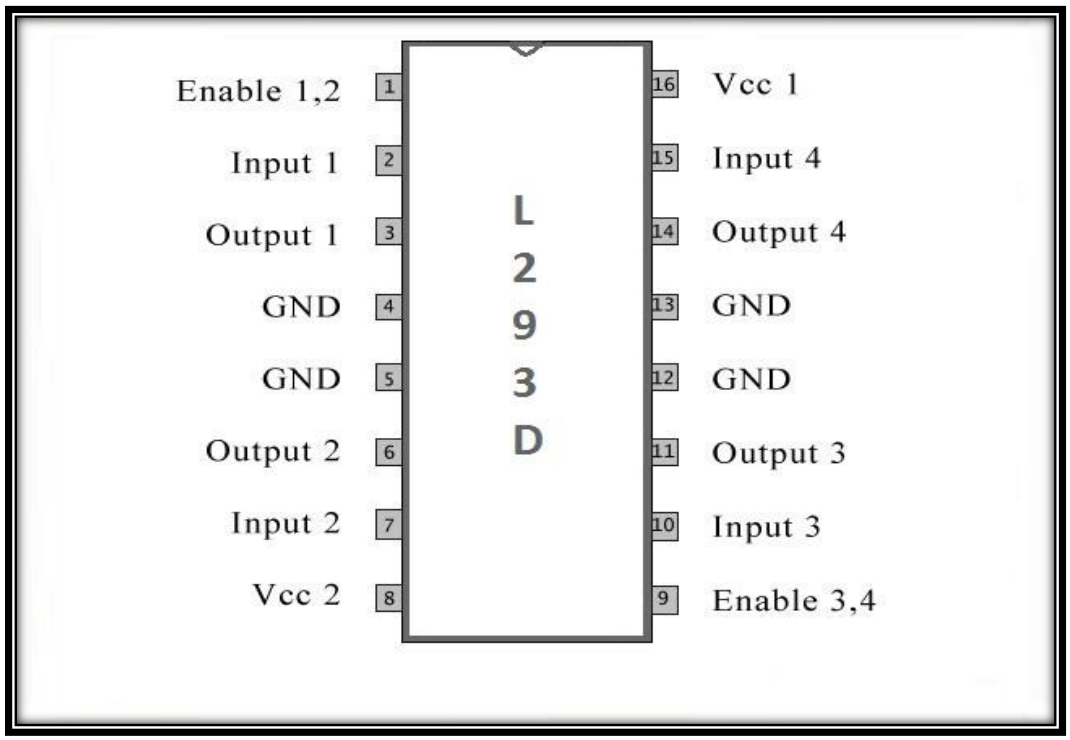


Email: admin@robomart.com
Contact: +91 - 8744000555

4.2. L293D Dual H-Bridge Motor Driver

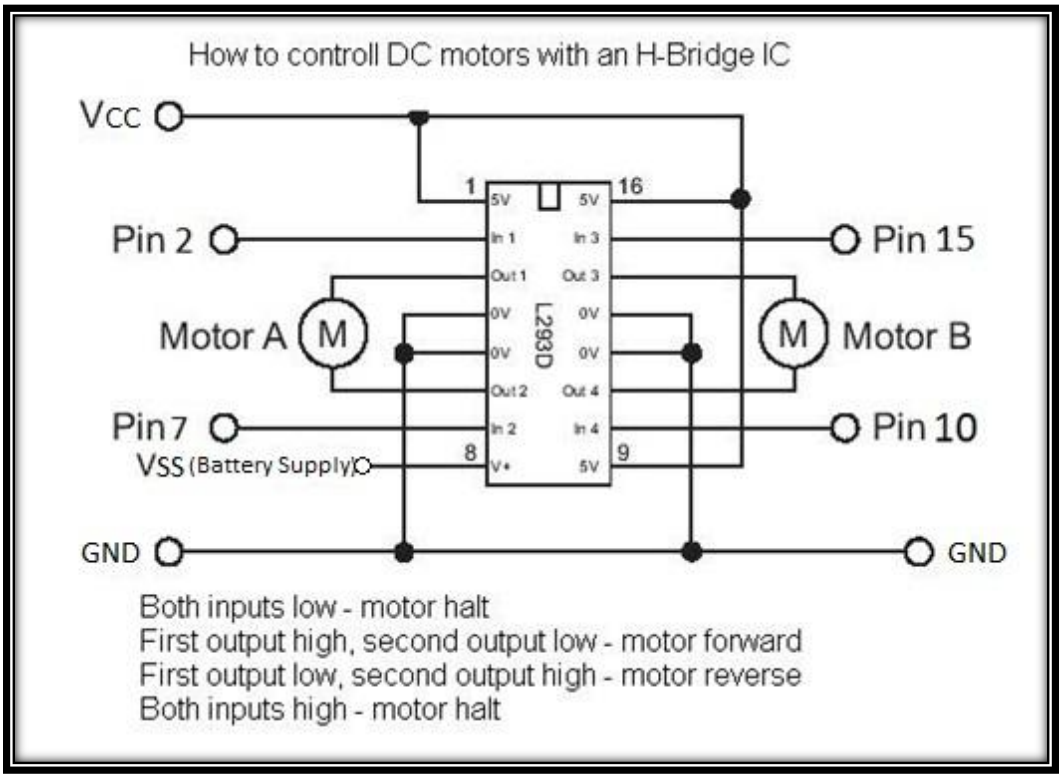
L293 series of chips are power H-bridge motor drivers. The L293D chip is in 16-pins dip packages, and has two h-bridge drivers. An H bridge is typically capable of running one DC motor bidirectional (forward, backwards, off), or two separate motors unidirectional (forward, off). Thus a L293 chip can run two motors bidirectional, or 4 unidirectional.

4.3. Pin Diagram of L293D



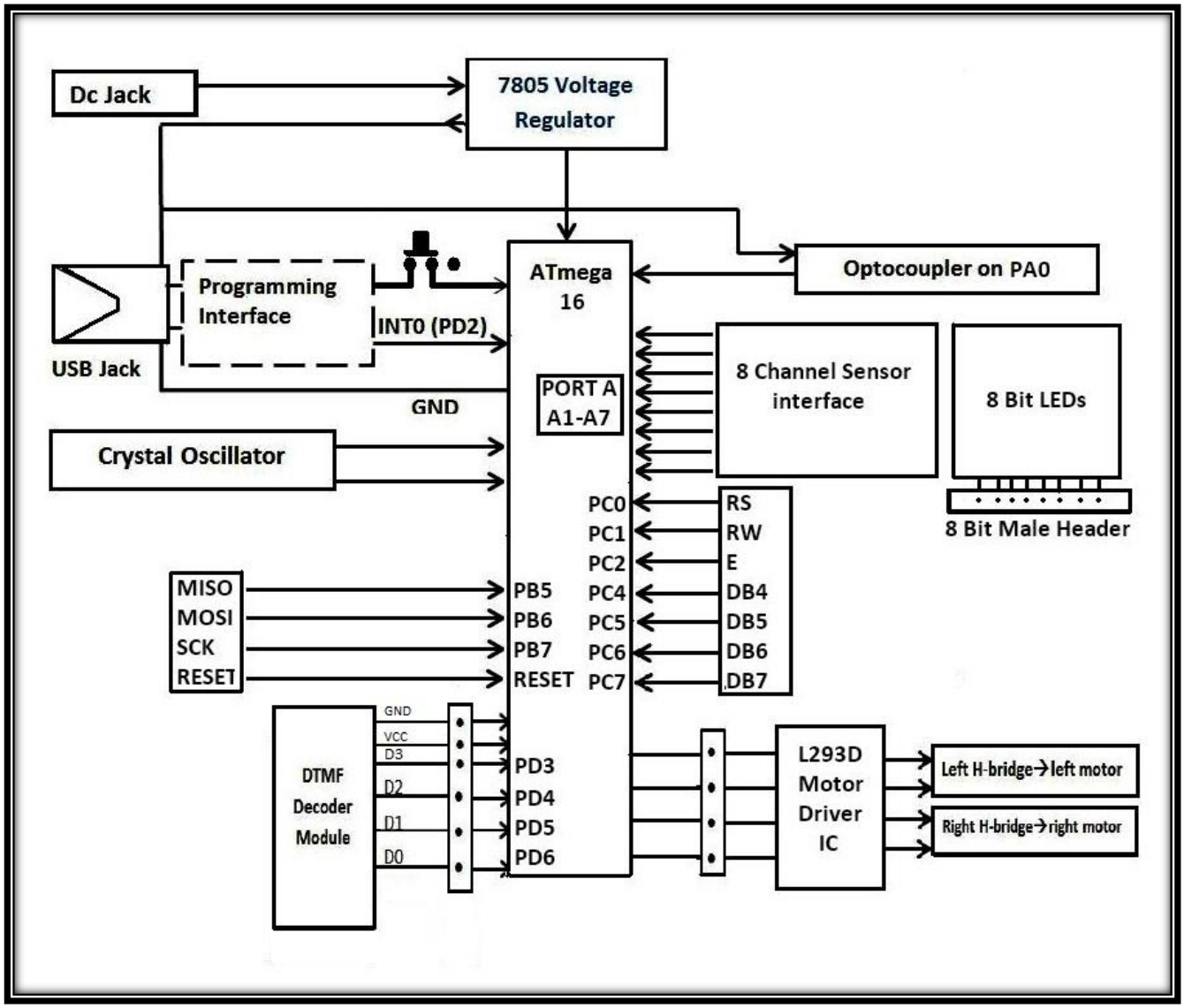
Email: admin@robomart.com
Contact: +91 - 8744000555

4.4. Interfacing



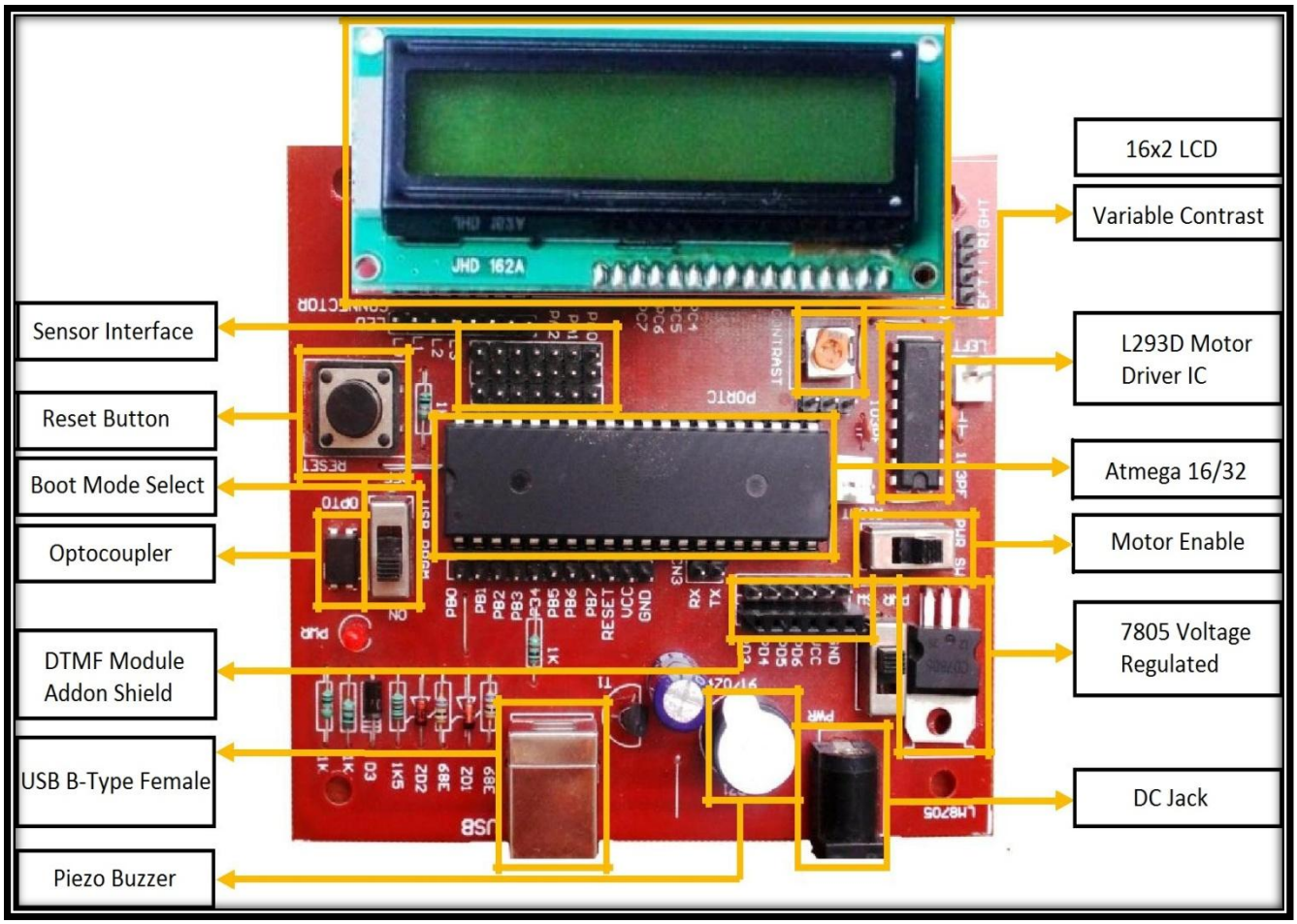
Email: admin@robomart.com
Contact: +91 - 8744000555

5. Block Diagram of Development Board Connected



Email: admin@robomart.com
Contact: +91 - 8744000555

6. Block Description of Development Board Connected



Email: admin@robomart.com
Contact: +91 - 8744000555

7. Precautions

- USB Boot loader must be used always. If kit is once programmed using ISP, USB Boot loader will Boot loading firmware will get erased and cannot be used for programming.
- Booting Mode must be selected using the Slider Switch before programming.
- Pin PD2 should not be used for interfacing as it is used for Boot loader.
- Novice users must not use ISP Programmer on this kit.

Email: admin@robomart.com
Contact: +91 - 8744000555

8. Starting AVR Studio

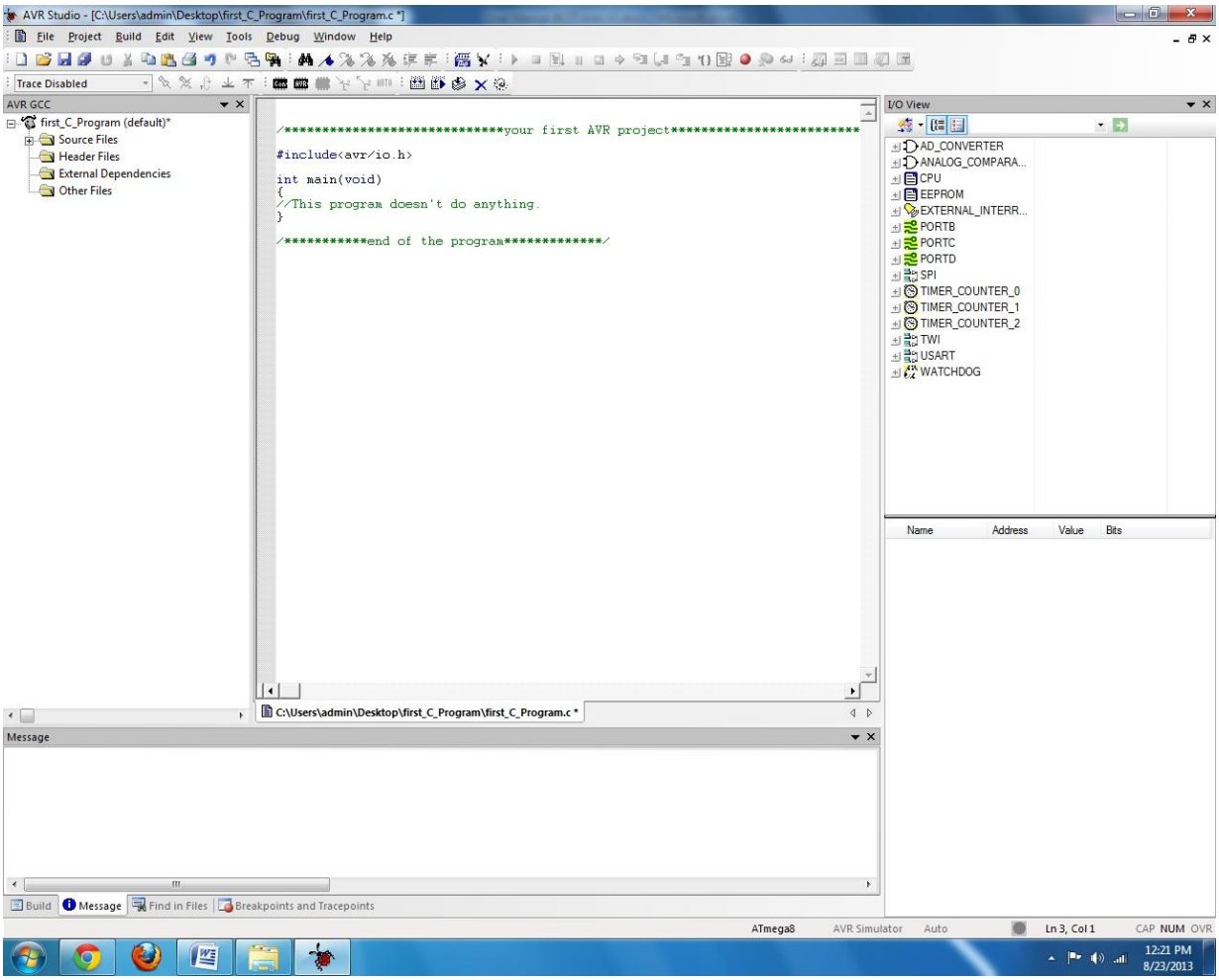
8.1. Writing your first C program

```
/*****************************your first AVR project*************************/
#include<avr/io.h>

int main(void)
{
//This program doesn't do anything.
}

/************end of the program*************/
```

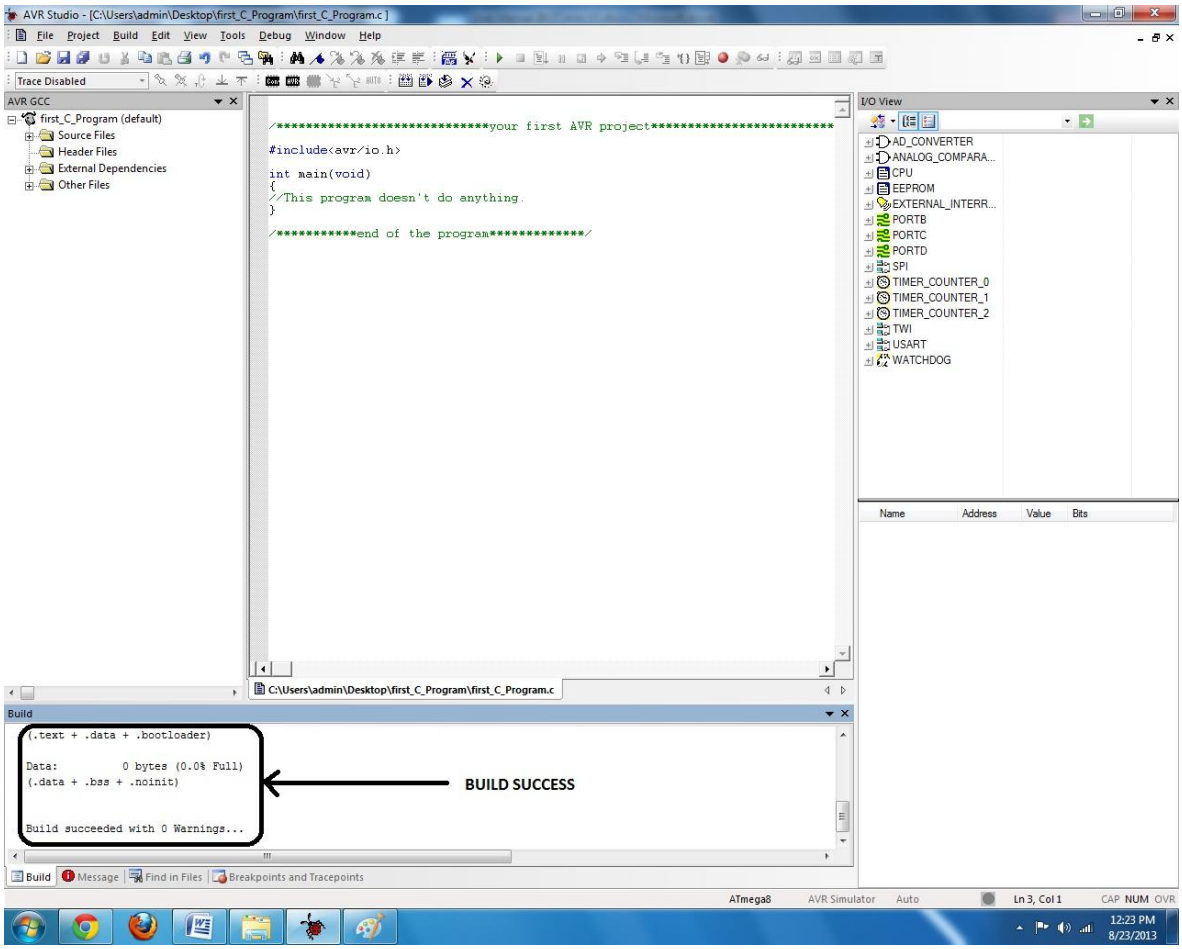
Write down the above code to your AVR studio Text Editor. See Screen shot of the same below.



Email: admin@robomart.com
Contact: +91 - 8744000555

8.2. Compilation of your first C program

Press F7 key to compile your project or See screen shot to compile using GUI interface.



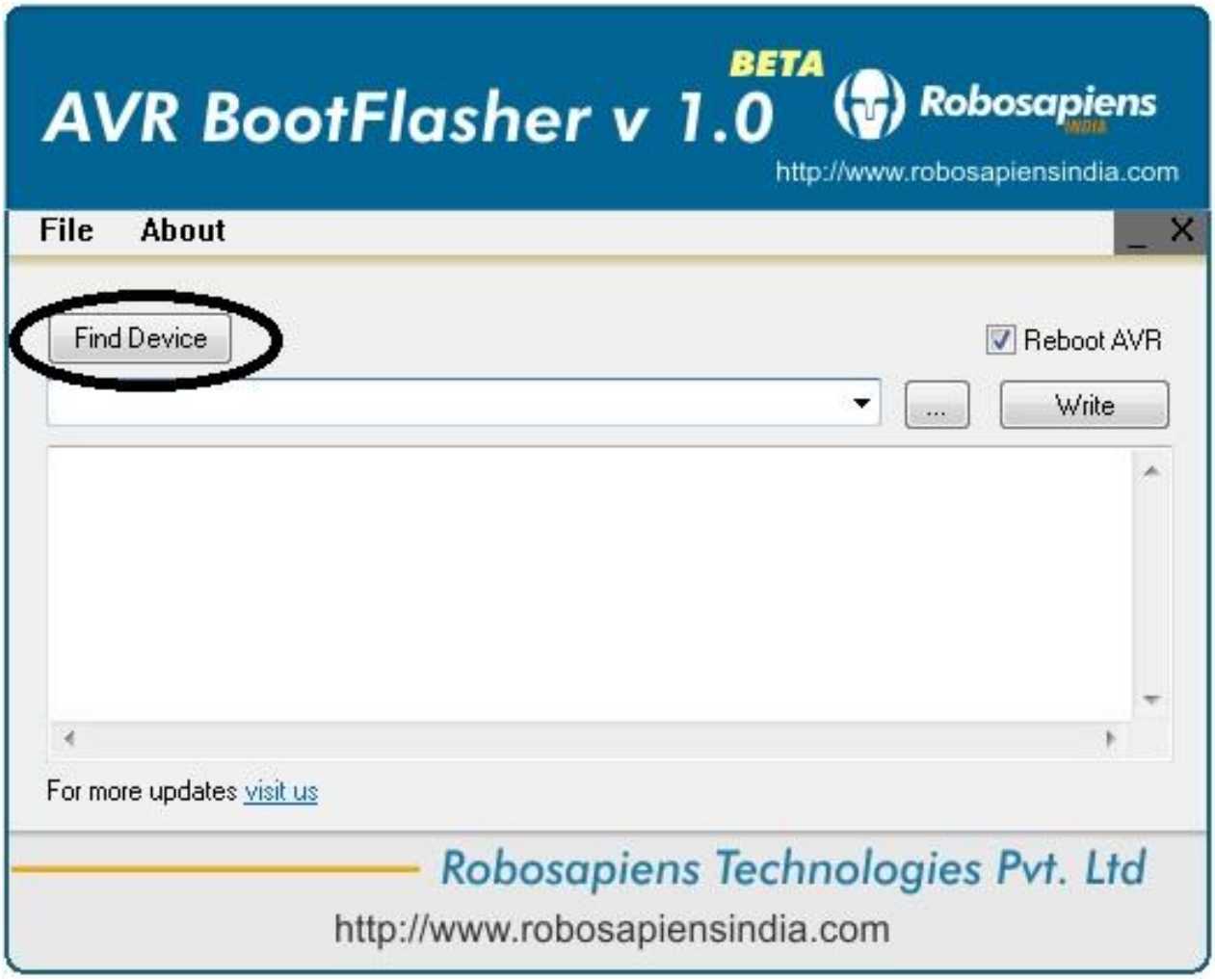
Email: admin@robomart.com
Contact: +91 - 8744000555

9. Dumping Program in Development Board Connected

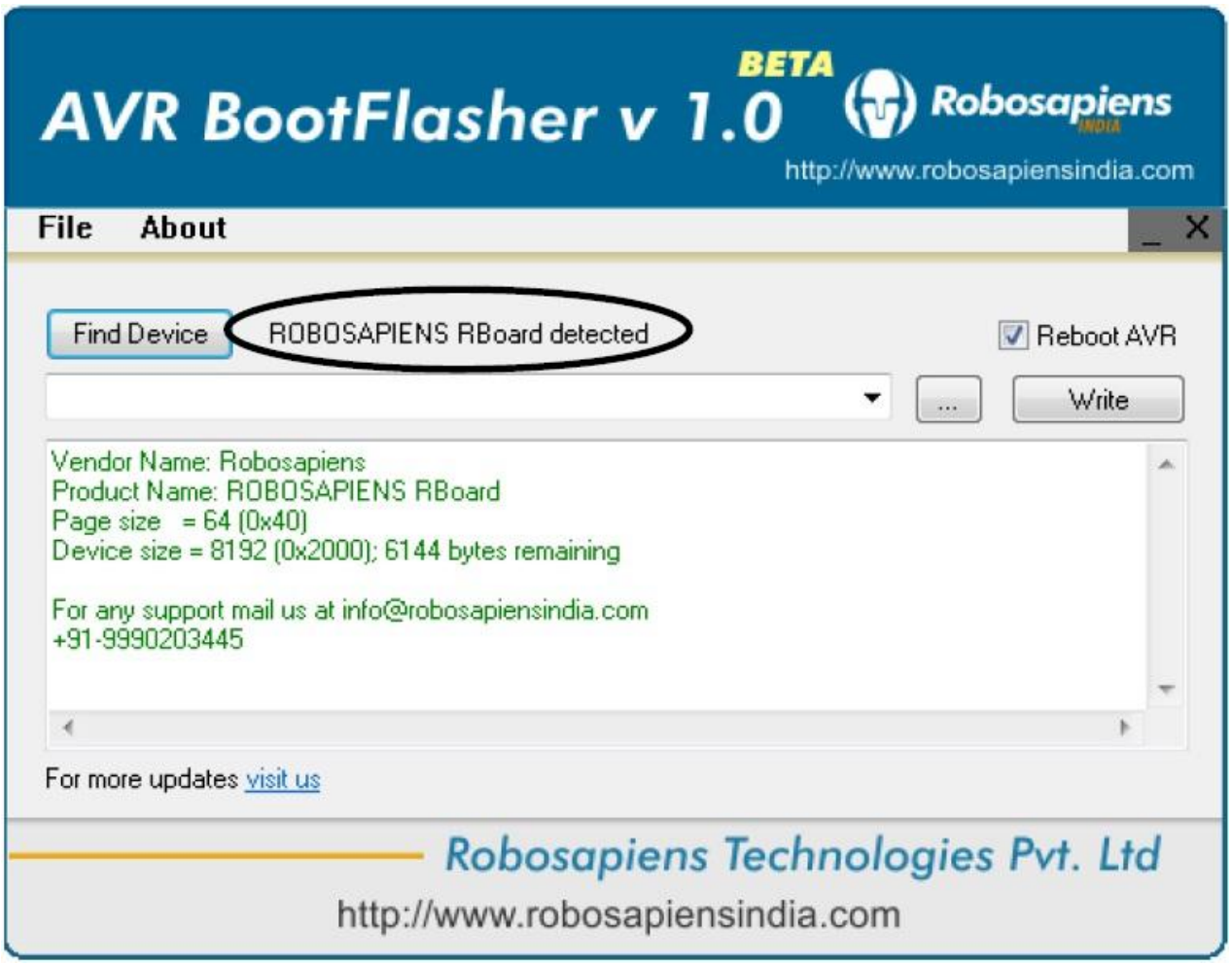
Step 1. Connect one end of your USB Cable with Computer's USB Port and connect other end with the development board.

Email: admin@robomart.com
Contact: +91 - 8744000555

Step 2. Click on to the button 'Find Device'.

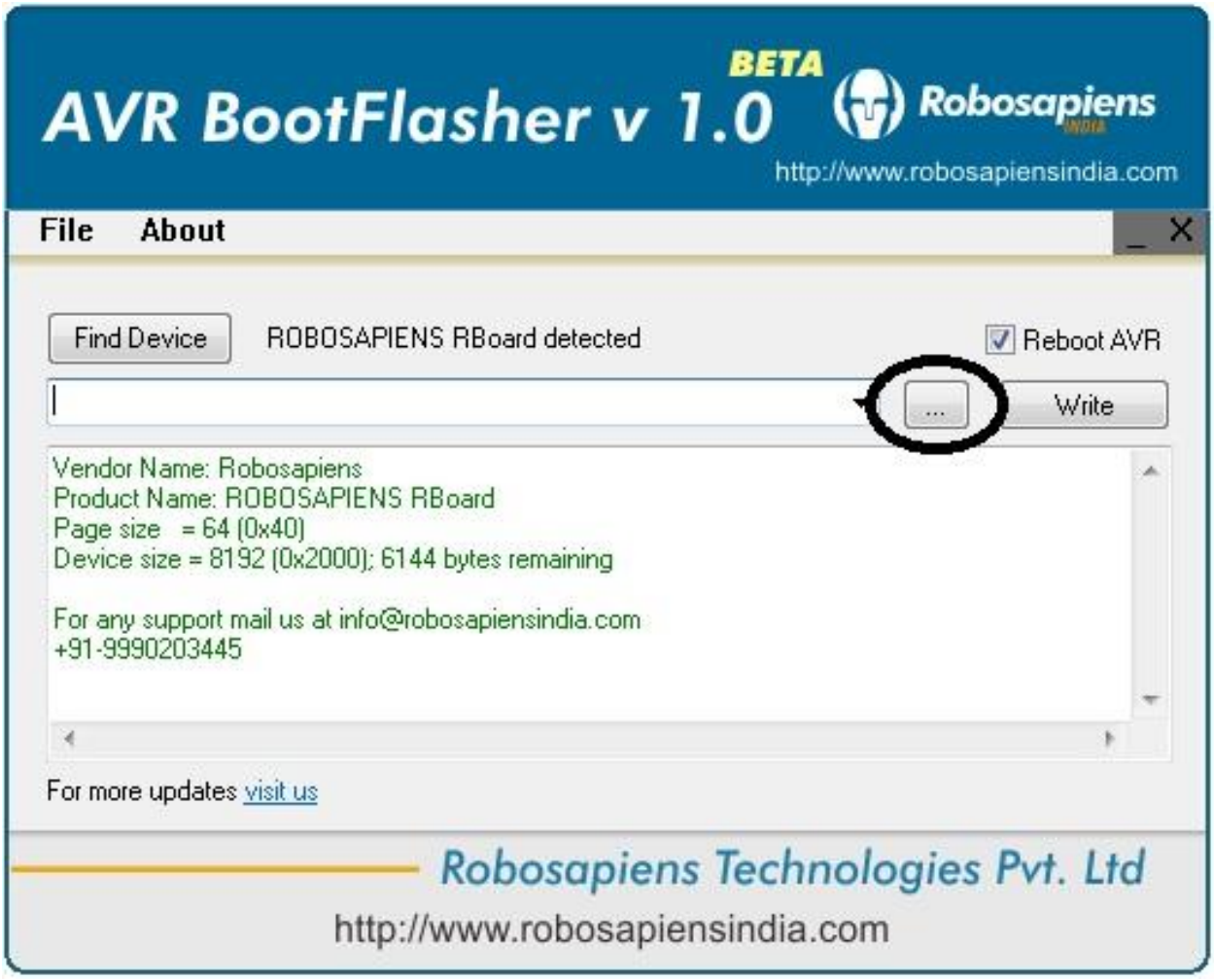


Email: admin@robomart.com
Contact: +91 - 8744000555



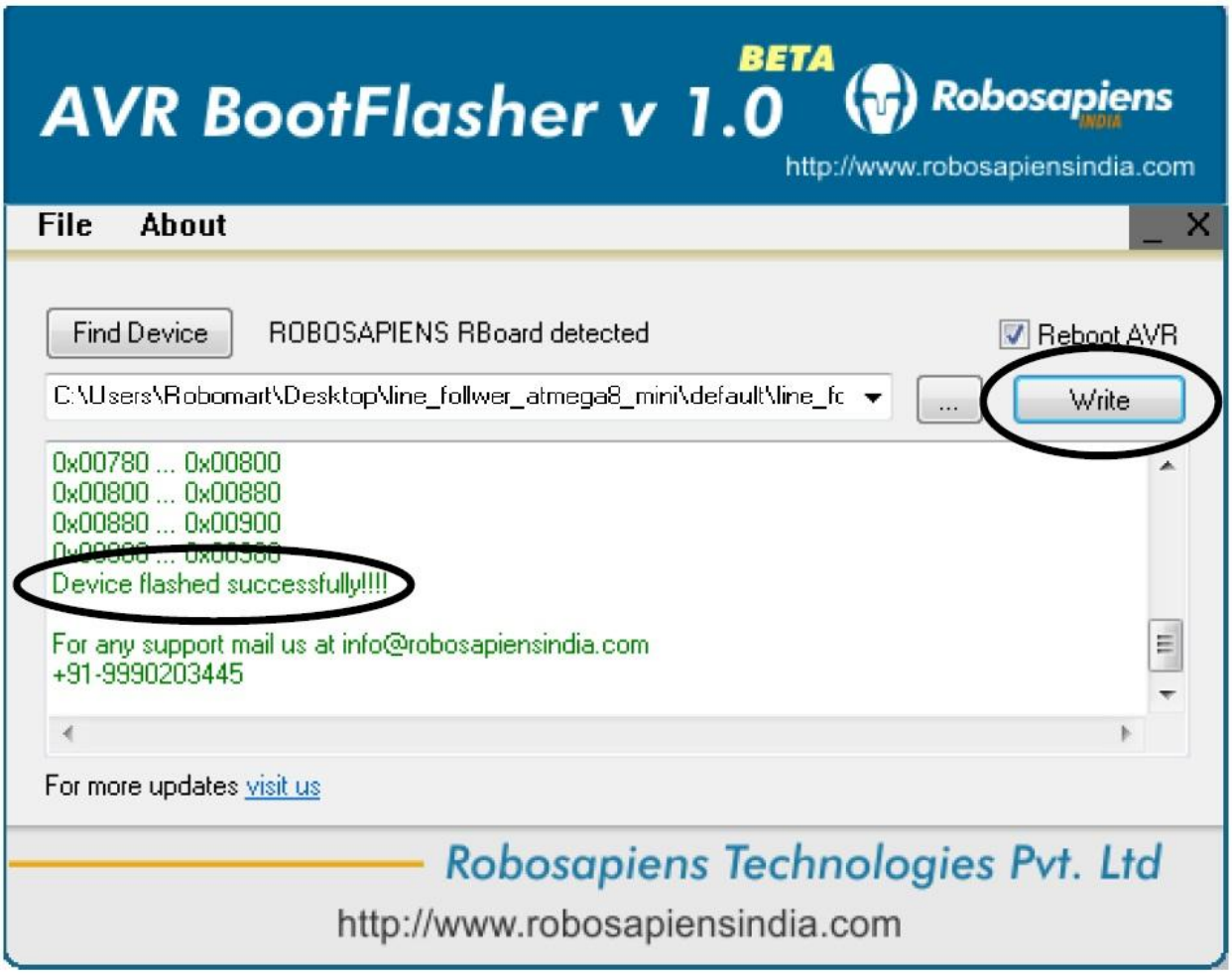
Email: admin@robomart.com
Contact: +91 - 8744000555

Step 3. Browse your '.hex' file to be flashed in the device.



Email: admin@robomart.com
Contact: +91 - 8744000555

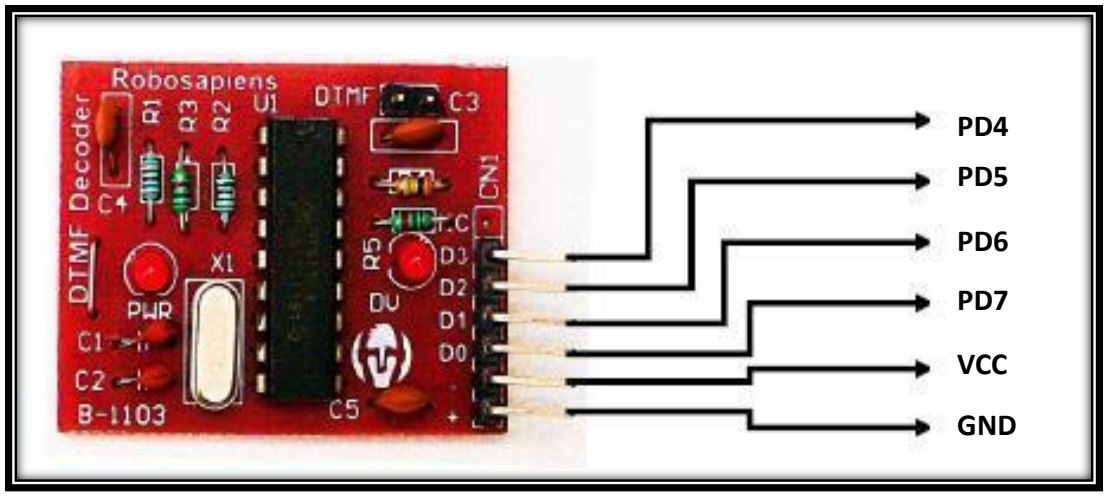
Step 4. Finally click on to the write button to burn the .hex file in the MCU.



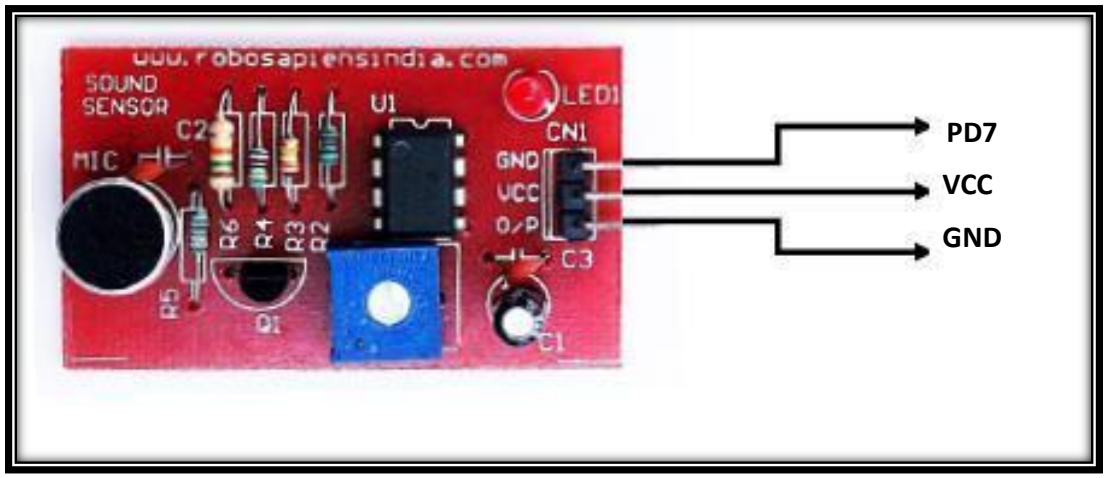
Email: admin@robomart.com
Contact: +91 - 8744000555

10. Connecting peripherals with Atmega 16/32 Robotics Mini Development Board V2.0

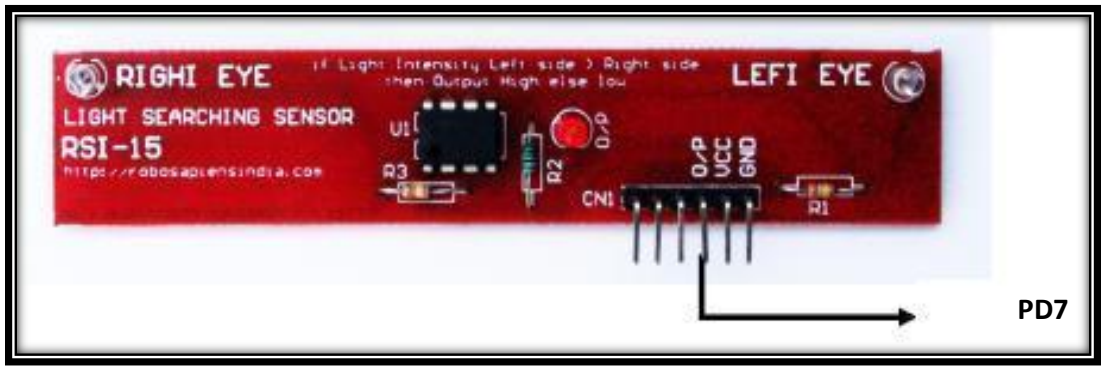
10.1. DTMF Decoder module



10.2. Sound Sensor



10.3. Light Sensor



Email: admin@robomart.com
Contact: +91 - 8744000555

11. Study of some C program using Loops and variables

11.1. Blinking LEDs

```
//Project name          : Blinking LEDs
// Designed By          : ROBOMART
//                        http://www.robomart.com

/* Program for "Blinking LEDs"
_____________________________________________________________
Connection settings of Kit
LEDs---@---------------------->PB0-PB7
Buzzer------------------------->PB4
BOOT Loader Condition Check--->PA0 (If 0 boot loader section else program execution section of Flash memory)
RESET------------------------->PIN 9
Crystal Oscillator (12MHz) ----->PIN 12 & 13
VB=Battery Supply
VCC=regulated 5V+
Gnd= Ground (0V)
*/

#define F_CPU 120000000UL
#include<avr/io.h>
#include<util/delay.h>
#include "robosapiens.c"

int main(void)
{
DDRB=0b11111111;   // PB0 - PB7 of PORTB are set as output.
While (1)                // infinite while loop
{
   PORTB=~PORTB;
   Wait (0.5);          //wait function defined in robosapiens.c file function argument:
                         // time in second
}

}
```

Email: admin@robomart.com
Contact: +91 - 8744000555

11.2. Line follower Robot program

```
//Project name        : Black Line Follower Robot
// Designed By        : ROBOMART
//                      http://www.robomart.com

/* Program for "Robot Following Black Line"
_____________________________________________________

Connection settings of Kit

LEDs---@------------------>PB0 – PB7
Left Sensor---------------->PA4
Right Sensor--------------->PA5
Buzzer--------------------->PB4
SOUND SENSOR----------->PD6
RIGHT MOTOR (+)-------->PB3
RIGHT MOTOR (-)--------->PB2
LEFT MOTOR (-)----------->PB1
LEFT MOTOR (+)---------->PB0
BOOT Loader Condition Check----->PA0 (If 0 boot loader section else program execution
section of Flash memory)
RESET------------------------->PIN 9
Crystal Oscillator-------------->PIN 12 & 13
VB=Battery Supply
VCC=regulated 5V+
Gnd = Ground (0V)
*/

#include<avr/io.h>

void main()
{
DDRA=0b00000000;    //set PORTA as input port
DDRB=0b00011111;    //PB0, PB1, PB2, PB3 as output port
int ls=0, rs=0;          // define & initialize ls, rs integer as 0 to
                         // acquire the left sensor status in ls and right sensor
                         // status in rs

While (1)                // create infinite loop
{
ls=(PINA&0b00100000);   //acquire only left sensor status connected at PA5
rs=(PINA&0b00010000);   // acquire only right sensor status connected at PA4
```

……continued

Email: admin@robomart.com
Contact: +91 - 8744000555

```
if((ls==0b00000000)&(rs==0b00000000))   //check sensor status for both sensor OFF
{
PORTB=0b00001111;  //stop
ls=0;                    //set sensor status off
rs=0;                    //set sensor status off
}

if((ls==0b00100000)&(rs==0b00000000))  //check sensor status for left sensor=ON and
                                                  //right sensor=OFF
{
PORTB=0b00011000; //turn right and buzzer ON
ls=0;                   //set sensor status off
rs=0;                    //set sensor status off
}

if((ls==0b00000000)&(rs==0b00010000)) //check sensor status for left sensor=OFF and
                                                  //right sensor=ON
{
PORTB=0b00010001;  //turn left and buzzer ON
ls=0;                    //set sensor status off
rs=0;                     //set sensor status off
}

if((ls==0b00100000)&(rs==0b00010000)) //check sensor status for both sensor ON
{
PORTB=0b00001001;  //move forward
ls=0;                    //set sensor status off
rs=0;                    //set sensor status off
}

}

}
```

Email: admin@robomart.com
Contact: +91 - 8744000555

Thanks for purchasing Atmega16/32 Robotics Mini Development Board V2.0 from