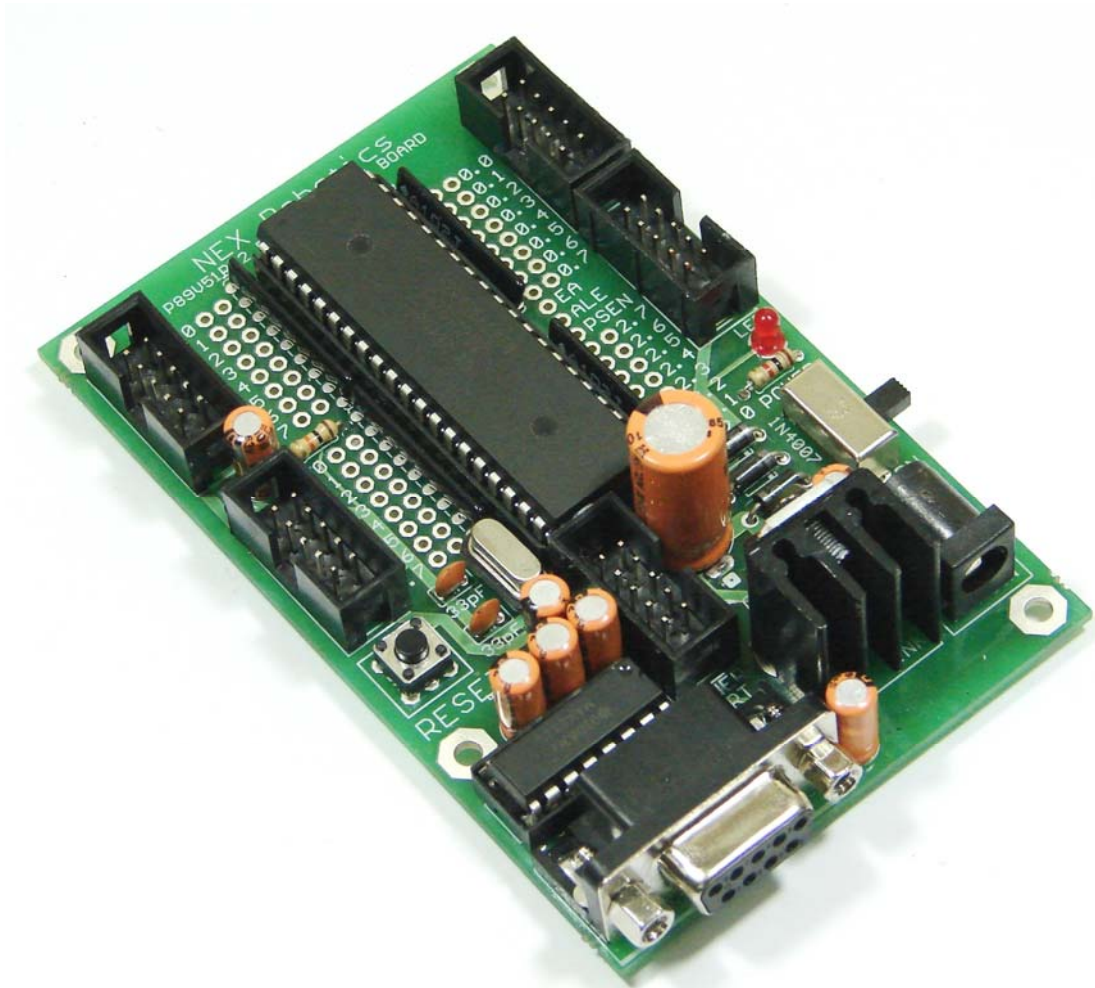
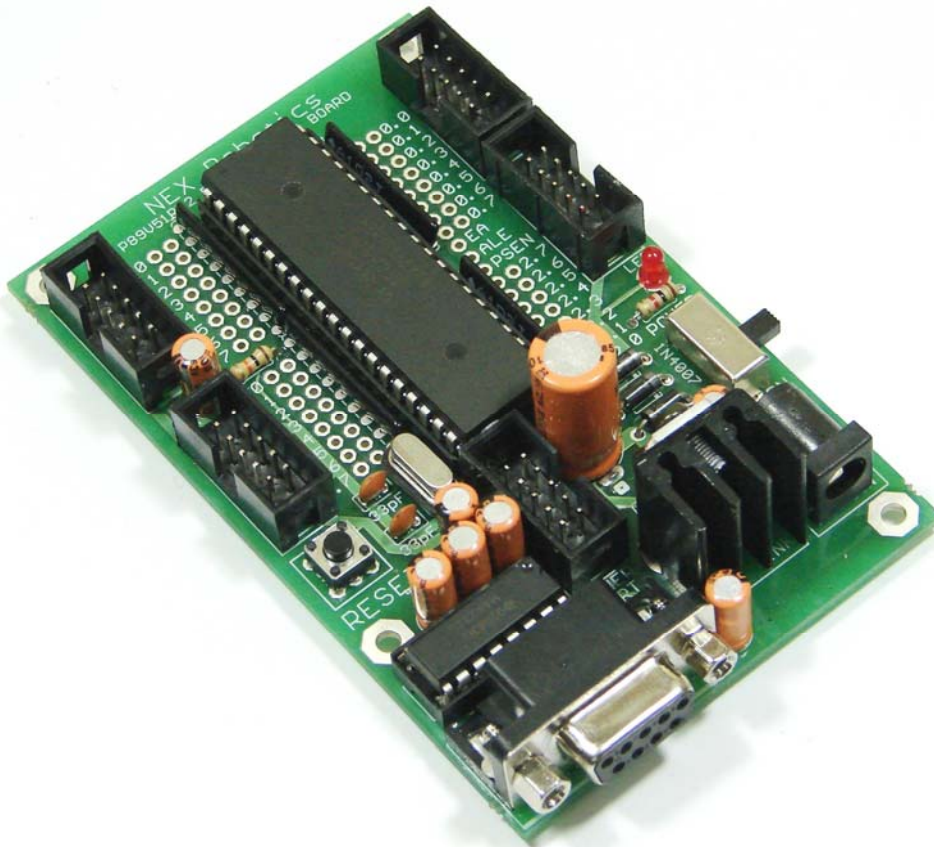


P89V51RD2 Development Board May 2010



P89V51RD2 Development Board



P89V51RD2 Development Board

Introduction:

P89V51RD2 Development Board is a low cost development board which have all the basic components needed for gaining a headstart on your first 8051 microcontroller projects. It is made from double sided PTH PCB board to provide extra strength to the connector joints for increased reliability. Board can work on 7 to 15V AC or DC supply. It has built-in reverse polarity protection. 7805 voltage regulator has heat sink for heat dissipation so that it can supply 1Amp current continuously without getting over heated. It has switches for reset and power. It also has RS232 interface with DB9 female connector based on MAX232. All the ports are connected to standard 10 pin FRC connectors. Open pads for connecting microcontroller's pins to external devices are also provided. It can be programmed using Flash Magic programming utility of NXP's own programming utility via serial port.

Specifications

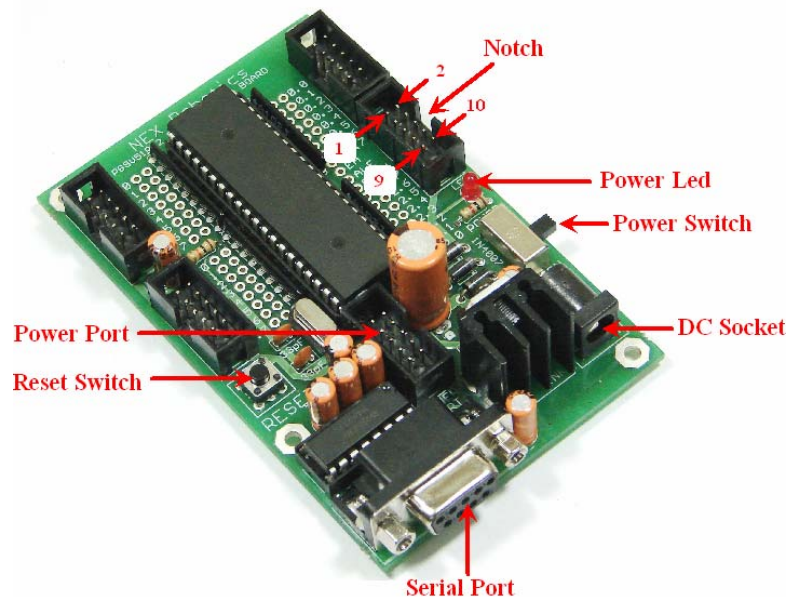
- Microcontroller: P89V51RD2 with 11.0592MHz crystal
- Double side high quality PTH PCB board to provide extra strength to the connector joints for increased reliability
- Power: 7 to 15V, AC or DC, Heat sink on 7805 for better current rating
- Reverse polarity protected
- Switches: Reset, Power
- RS232 serial interface
- 10 pin FRC connectors and soldering pads on all ports
- Compatible with General purpose prototyping board for development board for stackable design
- Application examples in Keil provided in the documentation CD

Package contains

- P89v51RD2 Development Board with P89v51RD2 microcontroller
- Documentation CD containing tutorial for P89v51RD2 microcontroller programming in Keil and application examples.

Documentation CD contain following Application examples

- I/O Port operation
- Interrupt control
- Timer as external Counter
- PWM (Pulse Width Modulation) for power control
- UART (Serial Communication)



P89V51RD2 Development Board

Warning:

Do not give Power supply more than 15 V AC/DC.

Note:

P89V51RD2 Development Board schematic is in the documentation CD

PORT FRC connector pin connections

Pin No.	Pin Function
1	Pin 0 of PORT X
2	Pin 1 of PORT X
3	Pin 2 of PORT X
4	Pin 3 of PORT X
5	Pin 4 of PORT X
6	Pin 5 of PORT X
7	Pin 6 of PORT X
8	Pin 7 of PORT X
9	5V 1A output of 7805 voltage regulator
10	Ground

Power Port pin connections

Power port is used to give power to the accessories boards compatible to the microcontroller development board

Pin No.	Pin Function
1	5V 1A output of 7805 voltage regulator
2	5V 1A output of 7805 voltage regulator
3	Ground
4	Ground
5	Vin DC (Input supply – 1.2V DC drop due to bridge rectifier)
6	Vin DC (Input supply – 1.2V DC drop due to bridge rectifier)
7	NC
8	NC
9	NC
10	NC

Loading Firmware on Development Board

P89V51RD2 Development Board programming involves two steps. First step is to write and compile the code and generate the “*.hex” file. Second step is to load this “*.hex” file on the microcontroller using Flash Magic software provided by NXP (formerly Phillips)

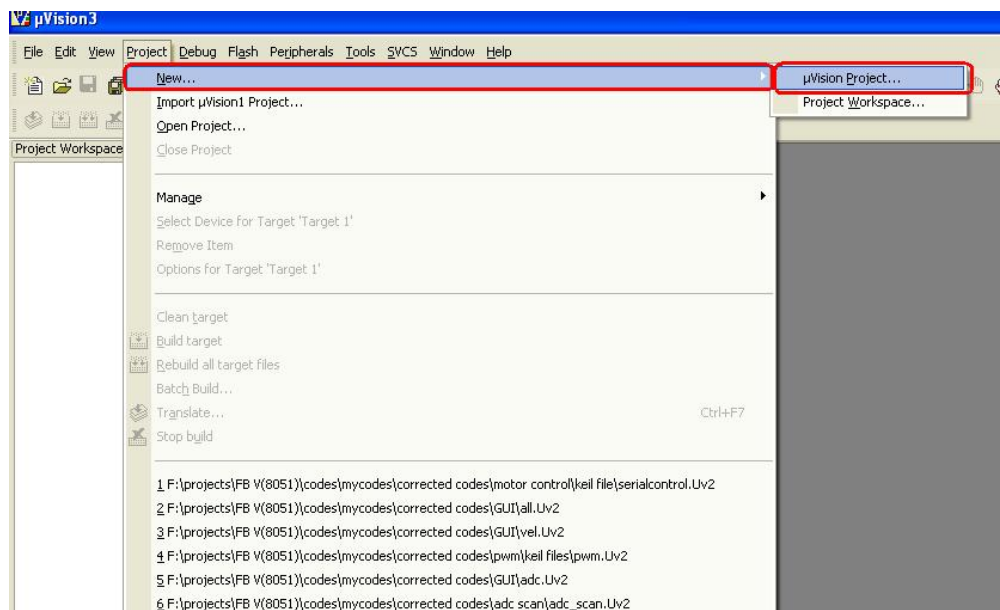
We are going to use Keil-U-Vision (Version 4) software for writing the code for the microcontroller. We can also use any other open source of proprietary software supporting P89V51RD2 microcontroller. P89V51RD2 Development Board CD contains free version of the uVision 4 software. You can also download latest version from <http://www.keil.com/dd/chip/3711.htm> and click [C51 Evaluation Software](#)

Writing program in KEIL uVision3

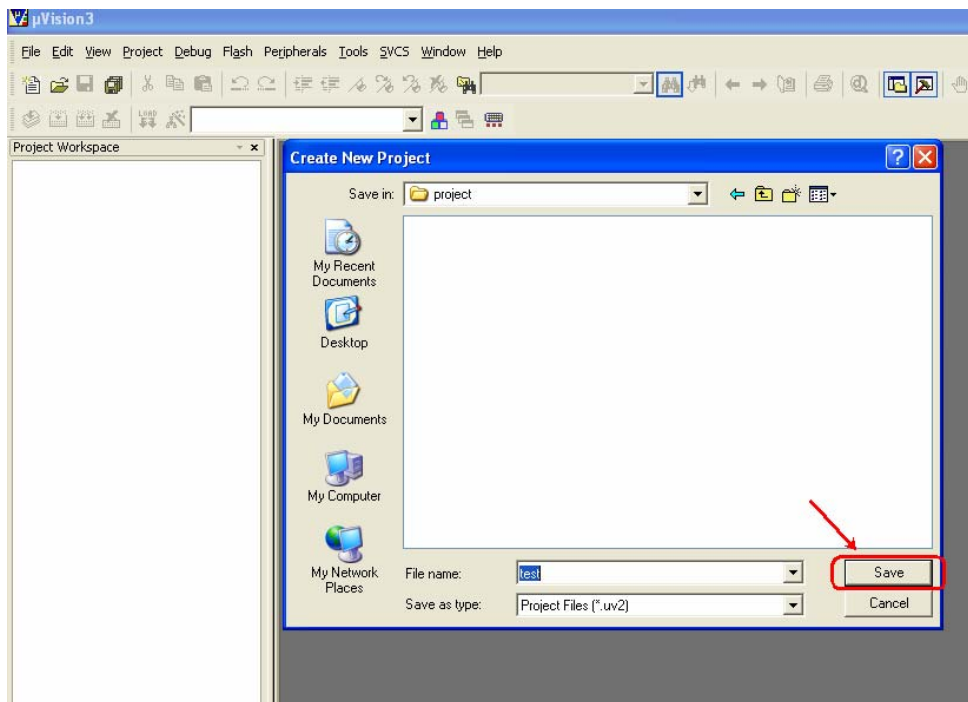
Start with Keil-U-Vision



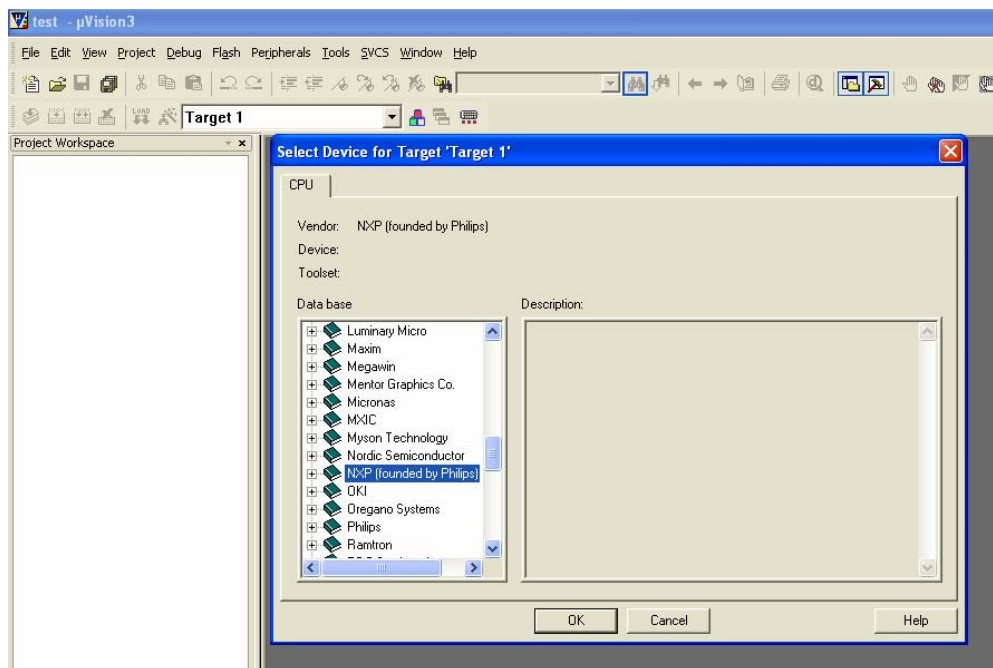
Go to project and start new project.

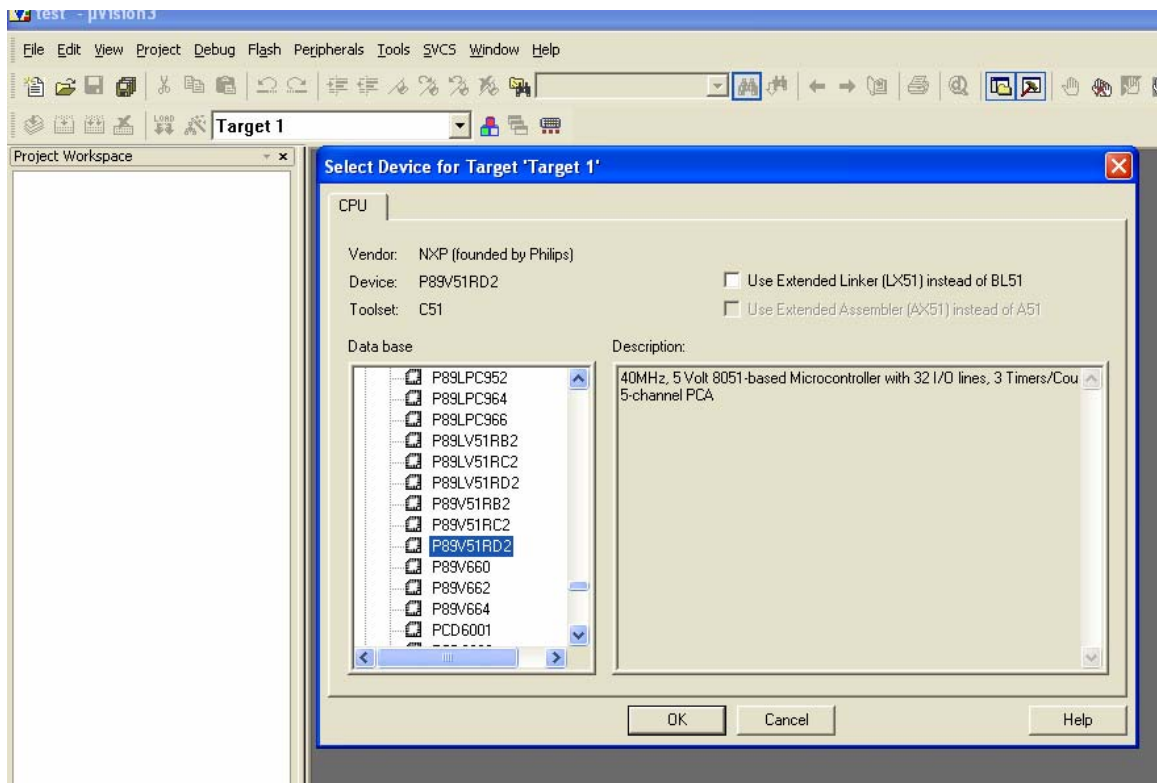


Name your project and save it in your project folder. (always create new folder for new project)



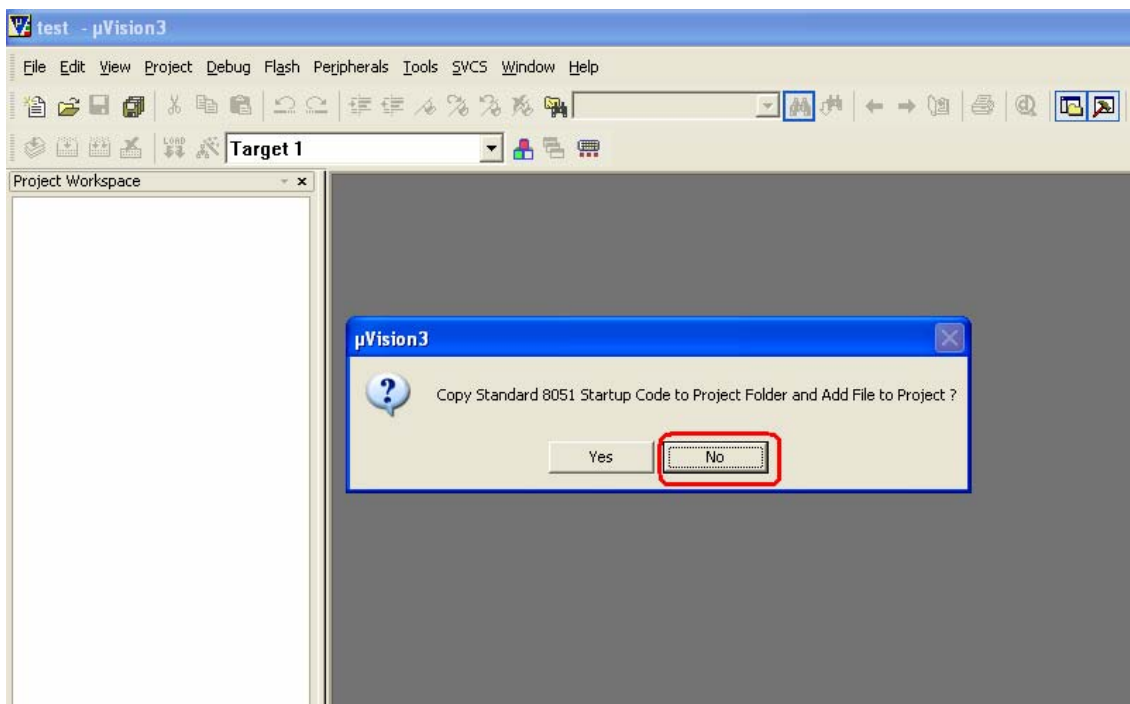
A dialogue box will open asking you to select your device used. Select the appropriate device for e.g. P89V51RD2. Then click OK. P89V51RD2 can be found in the NXP (founded by Philips) directory.



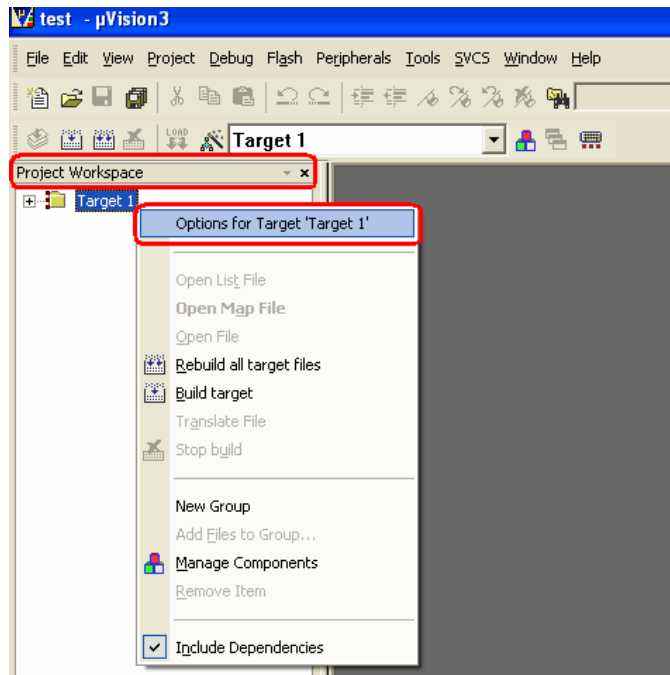


Next it will ask you if you want to add the A51STARTUP code.

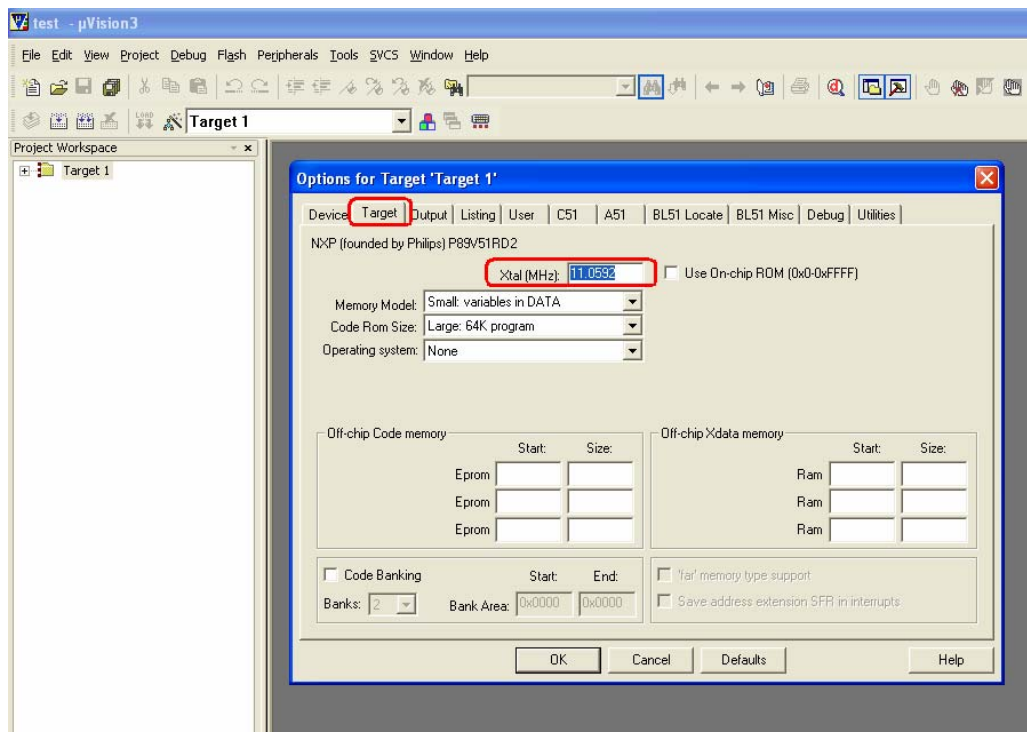
Say "NO"



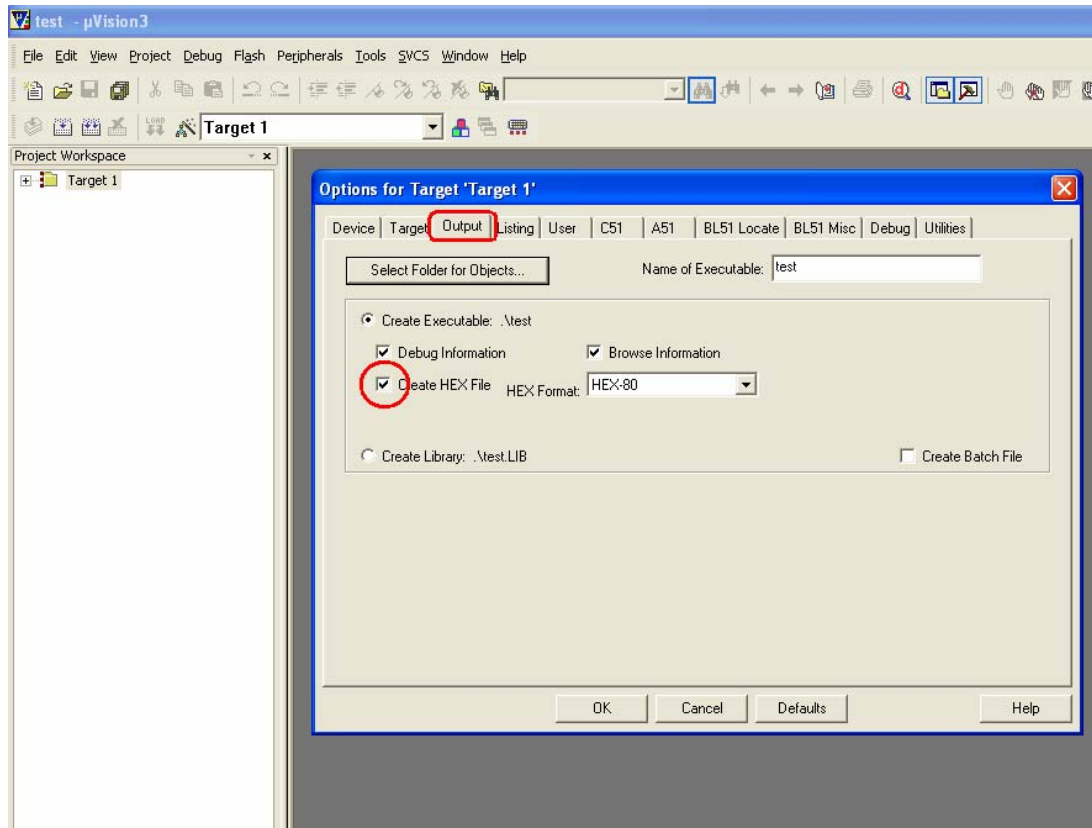
In the Project Workspace right click on ‘Target 1’ and select options for target as shown. A dialogue box to choose different options will open.



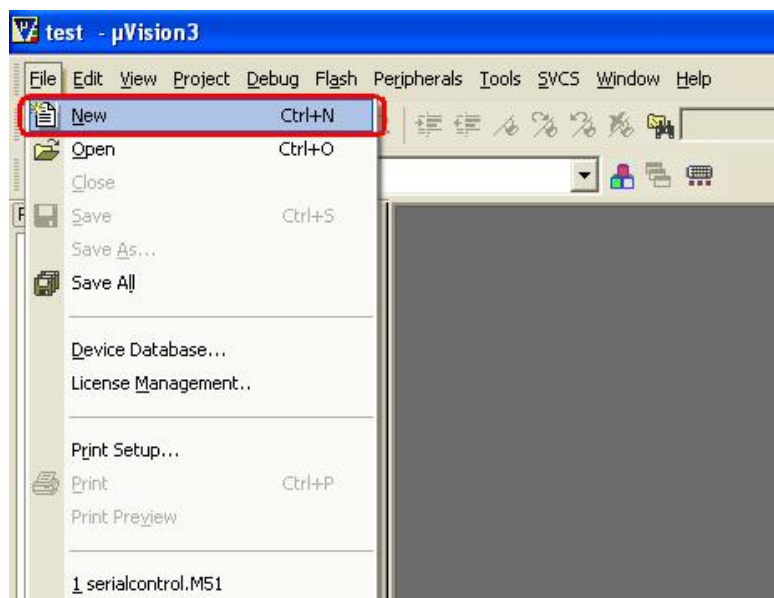
Click on the ‘Target’ tab. Enter the frequency of the crystal. For Development Board its 11.0592 MHz.



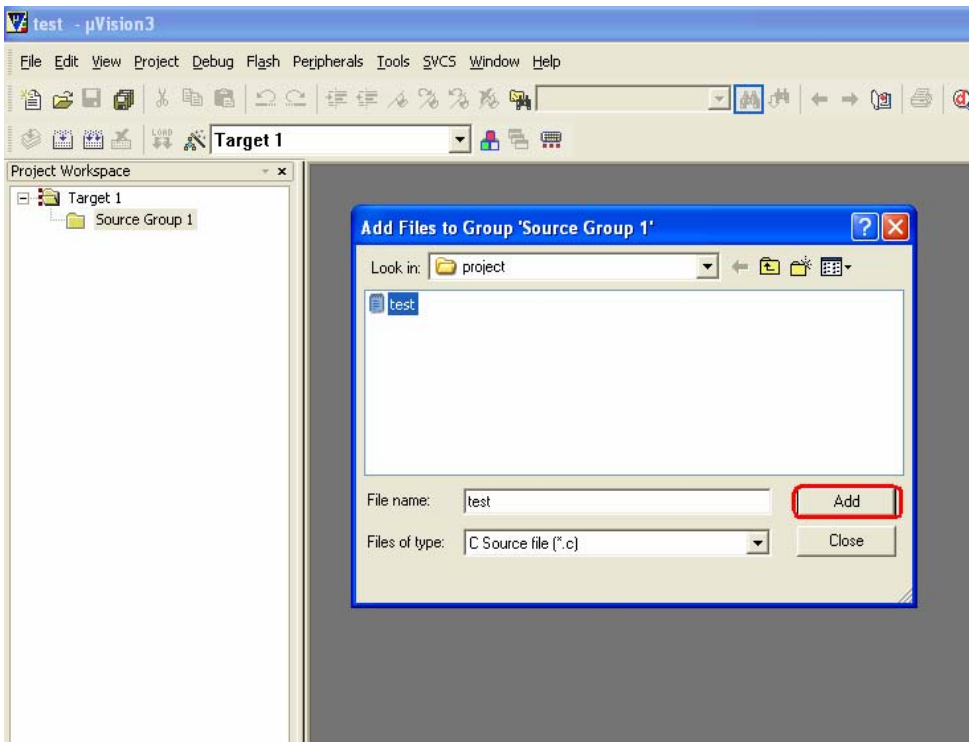
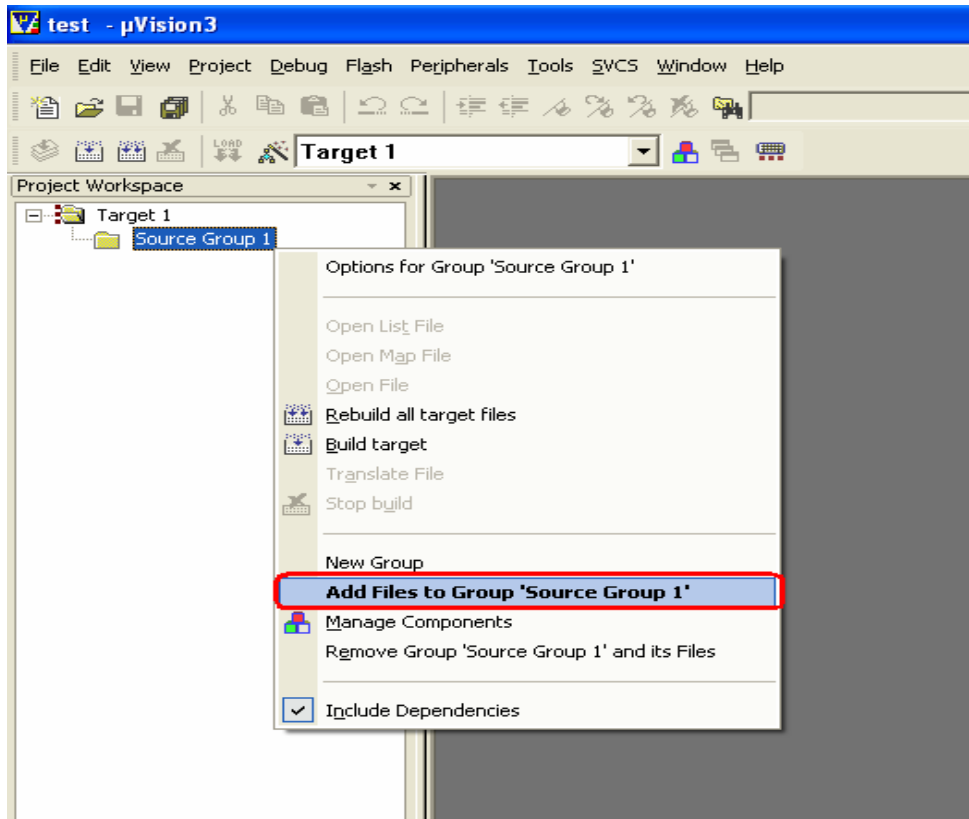
Go to the output tab and tick on the file to create HEX-file. Then click OK to save your options. Other tabs can be left with the default options.



After this is done, open a new file and save it with the project file as a C file i.e. with the extension “.c”.



Add this file to the project by Right-clicking on “Source Group” and choosing to add files to group. Select the appropriate ‘.C’ file to be added.



You can write macros and save them with an “.h” extension and add them to your project files. Write your ‘C’ code and save the file.

Add (write): `#include <intrins.h>`

Add (write): `#include "p89v51rx2.h"`

Copy file: ‘p89v51rx2.h’ from the folder ‘Experiments’ from the documentation CD and paste it in your existing project folder before building target.

Note: A declaration with the following syntax `#include <intrins.h>` would direct the compiler to look for the .h file in its own parent directory whereas the syntax `#include “intrins.h”` would direct the compiler to search for the .h file in the parent directory where the current project is stored. Therefore the double quotes “ ” are used to declare the files specific to the Microcontroller chip and its peripherals where as `< >` are used to declare more generic files which are compatible with multiple chips.

```
//code for buzzer beep
```

```
//connect Buzzer to Port pin 2.7.
```

```
#include <intrins.h>
```

```
#include "p89v51rx2.h"
```

```
sbit buzzer=P2^7; //buzzer = 1; buzzer off, buzzer = 0; buzzer on,
```

```
// function for giving a delay of ms milliseconds
```

```
void delay_ms(unsigned int ms)
```

```
{
  unsigned int i,j;
```

```
  for(i=0;i<ms;i++)
```

```
  for(j=0;j<53;j++);
```

```
}
```

```
void main (void)
```

```
{
  while(1)
```

```
{
  buzzer=0;           //switch ON the buzzer
```

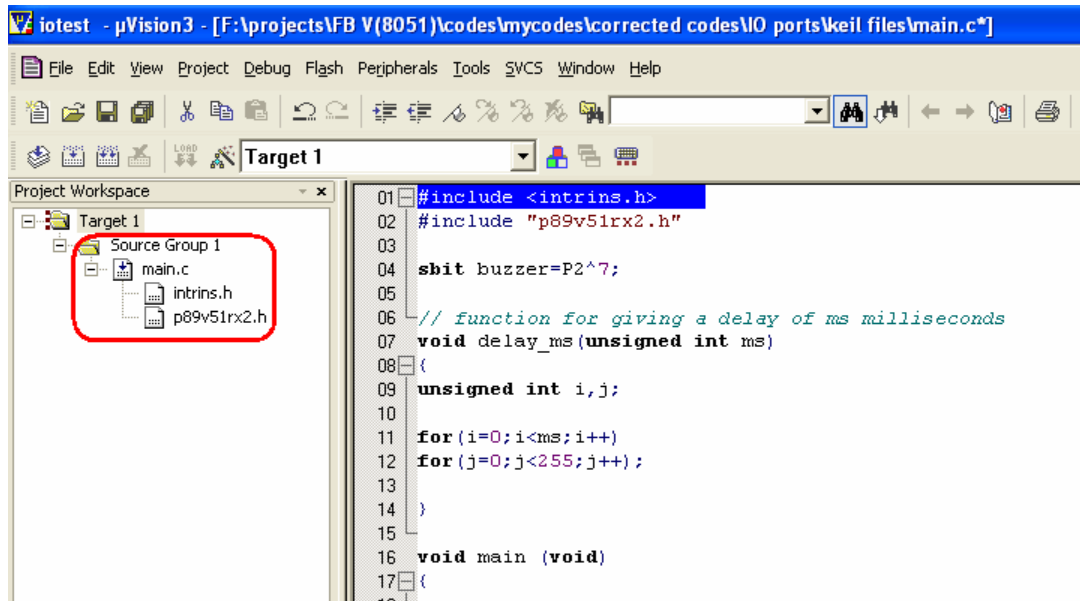
```
  delay_ms(100);     //give delay of 100 milliseconds
```

```
  buzzer=1;         //switch off the buzzer
```

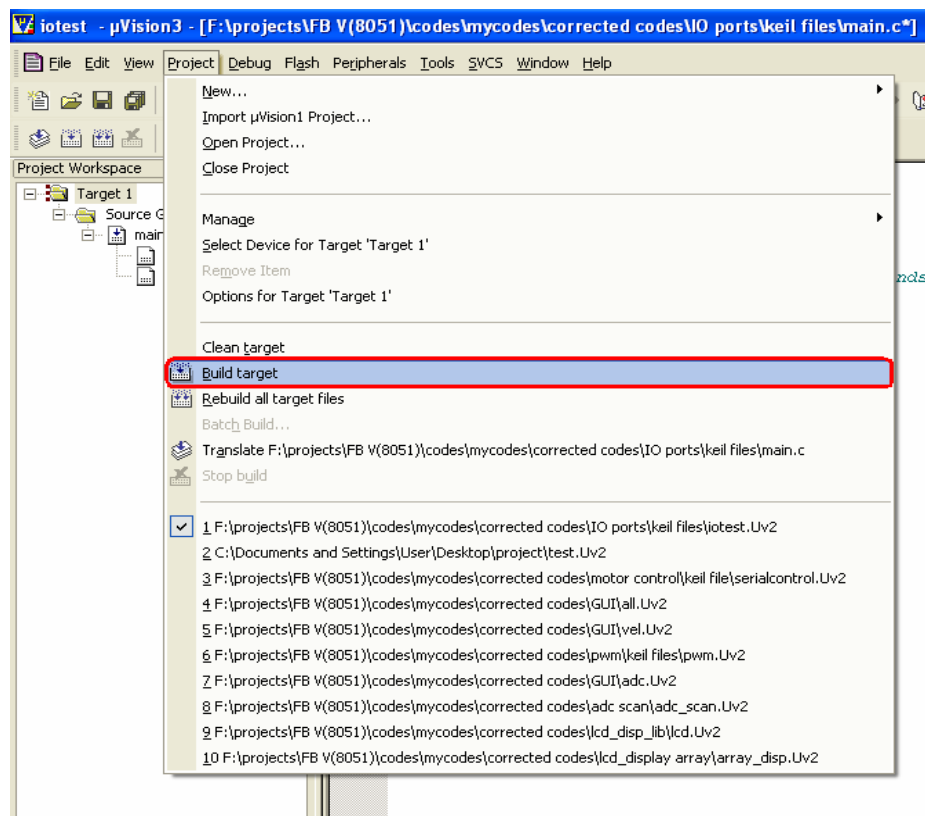
```
  delay_ms(100);
```

```
}
```

```
}
```



Now click on the “Project” tab and choose to “Build Target” as shown.



Check for any errors in the ‘Output Window’.

If there are no errors then ‘Hex’ file will be created and stored in the Project folder. You can then download this file onto your microcontroller using the In System Programming (ISP) software i.e. using flash tools like Flash Magic or FLIP or parallel programming as supported by your microcontroller.

Note: Use ‘rebuild all target files’ option if you are including multiple header file

Loading the generated Hex file on the microcontroller using Serial Port

Flash Magic

Version 4.18.1224

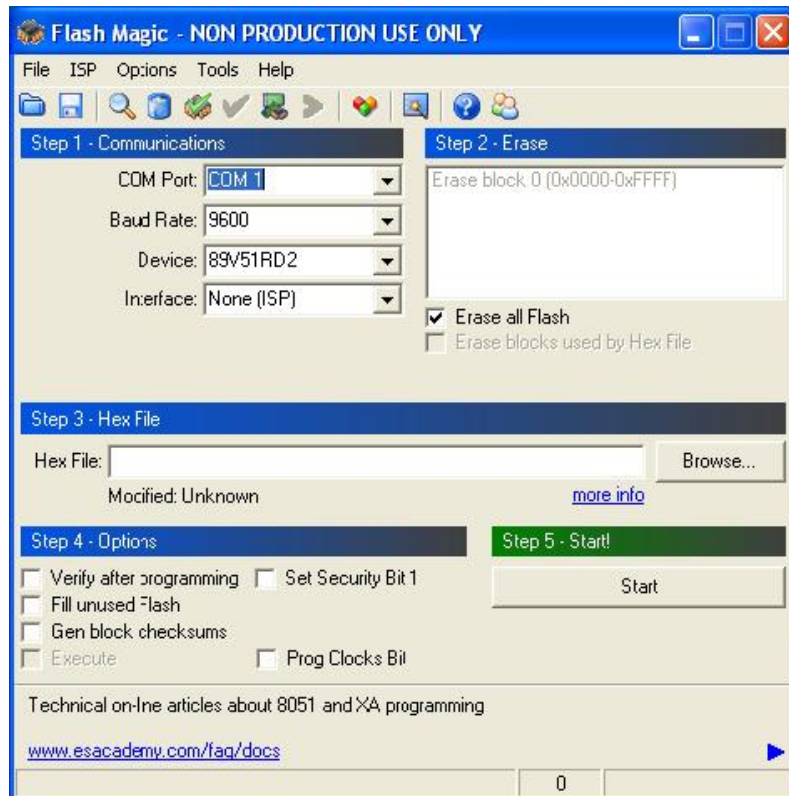
Flash Magic is Windows software which allows easy access to all the In System Programming (ISP) features provided by the devices.

These features include:

- Erasing the Flash memory (individual blocks or the whole device)
- Programming the Flash memory
- Reading Flash memory
- Reading the signature bytes
- Reading and writing the security bits
- Direct load of a new baud rate (high speed communications)
- Sending commands to place device in Boot loader mode

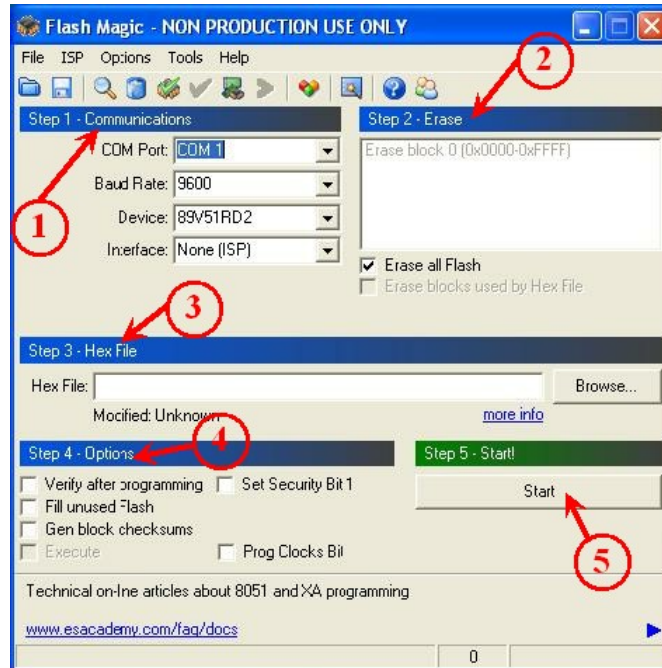
Flash Magic provides a clear and simple user interface to these features

Go to 'Flash Magic' Icon, it will open the main window as show below.

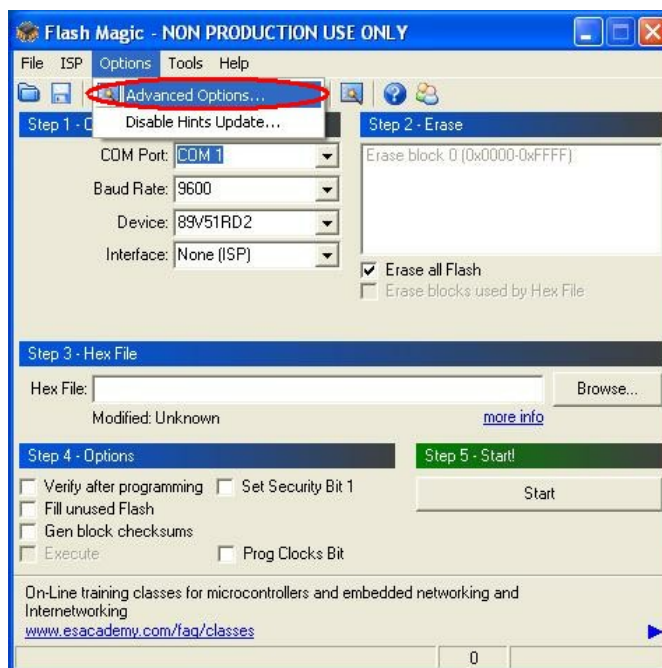


In main window you can see five types of sections.

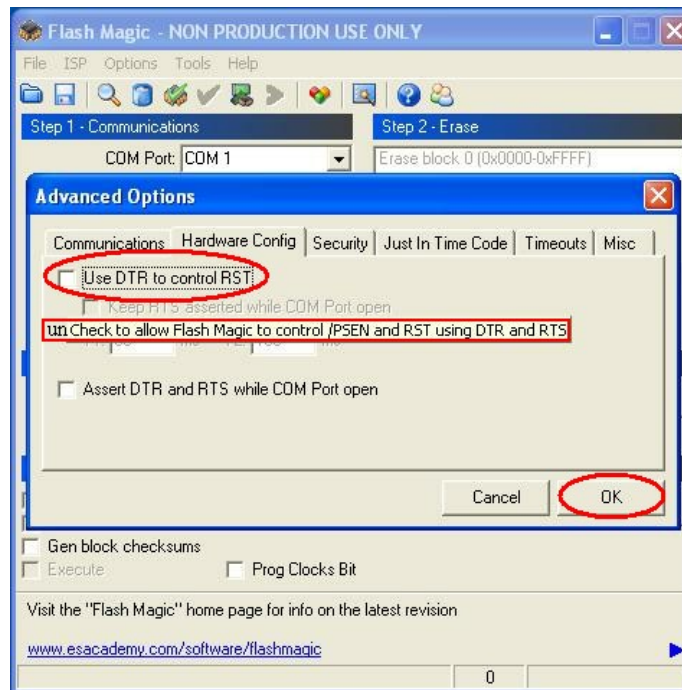
1. Communications
2. Erase
3. Hex File
4. Options
5. Start



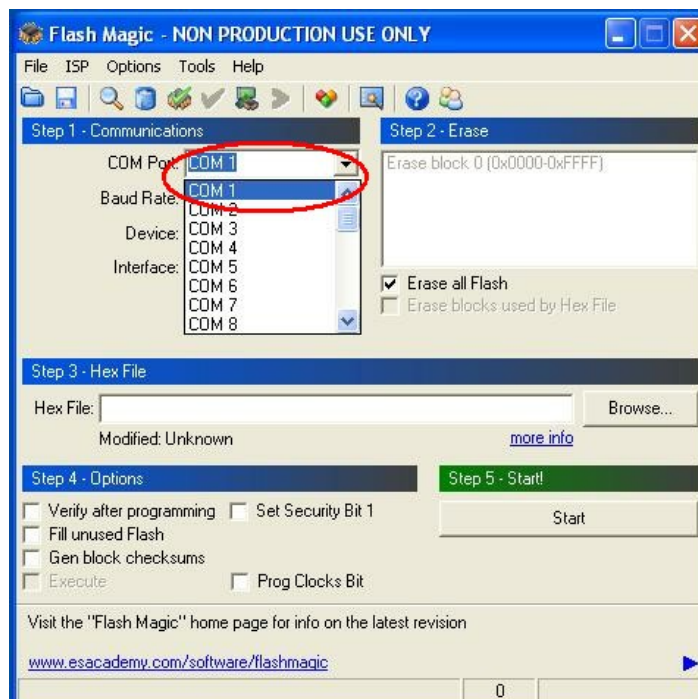
Go to “Option” in the toolbar, and select first menu ‘Advance options.’



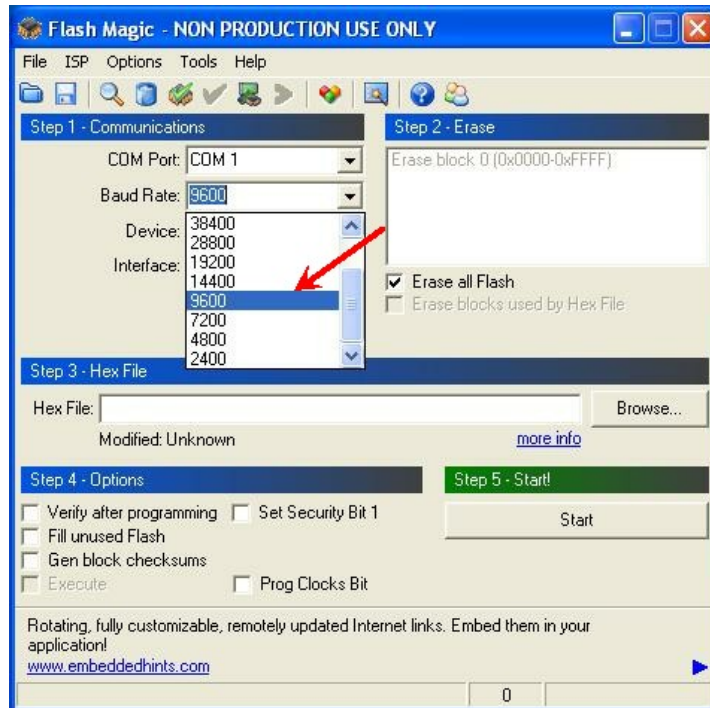
It will show you extracted window with many options.



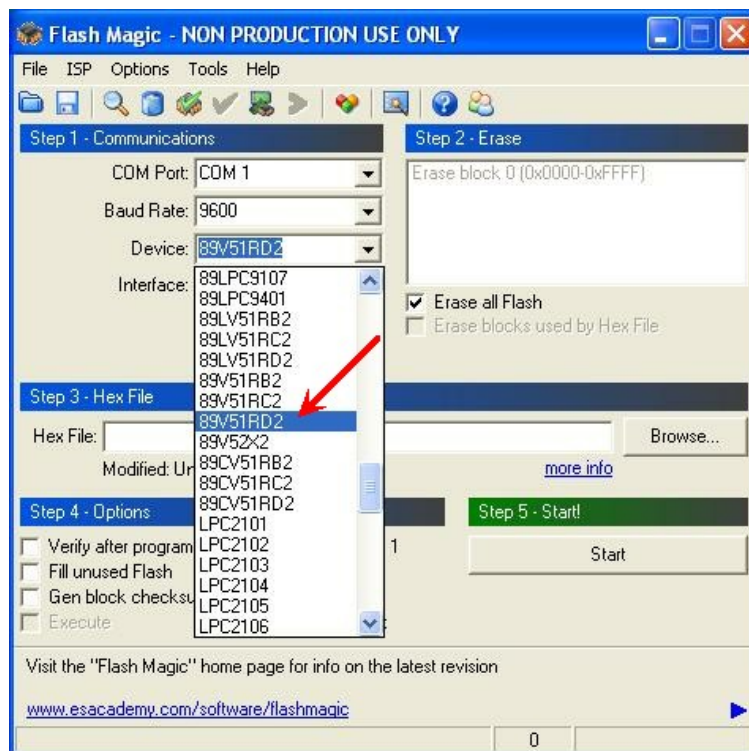
Uncheck the box which is highlighted to don't allow Flash Magic to control PSEN & RST using DTR & RTS. Now go for Communication selection, select 'COM 1' from 'COM PORT' Option if you are using serial port. If you are using USB to serial converter from NEX Robotics then find out COM port number with the help of documentation provided with the USB to serial converter.



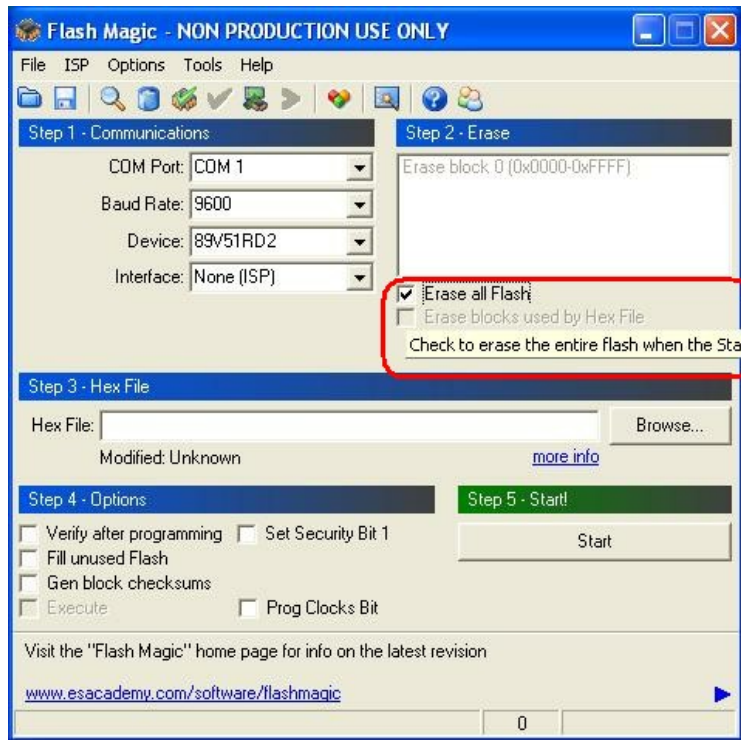
Select 'Baud Rate' 9600 from Baud Rate option.



Select proper Device '89V51RD2' from Device option.

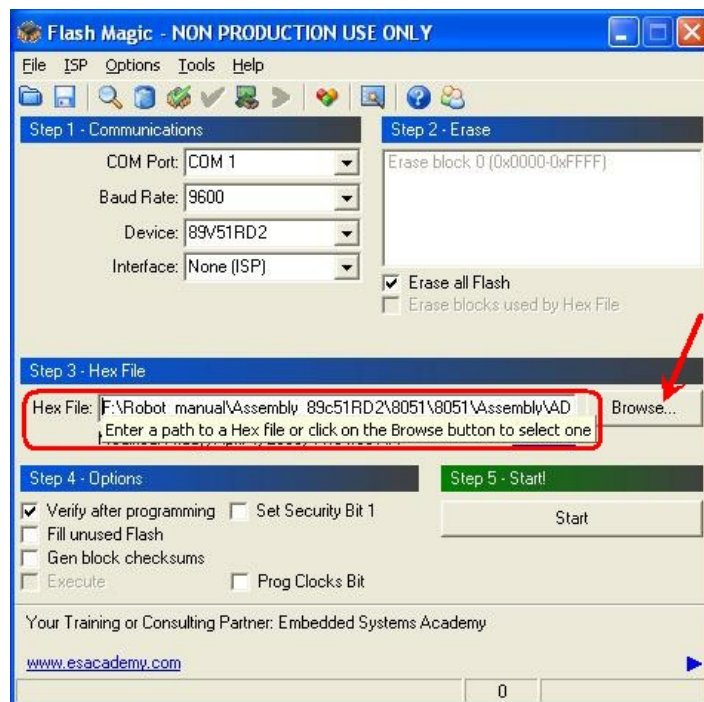


Go to 'Erase' section for erasing the all 'Flash' or Blocks used by 'HEX File'

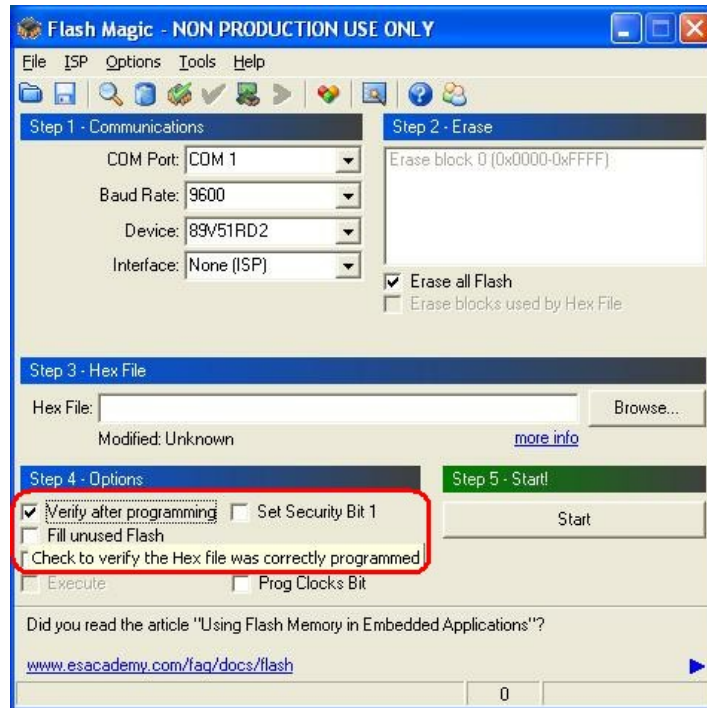


Check on the check box to Erase all Flash.

Go to 'HEX File' section click 'Browse' to select proper Hex file.



Go to ‘Option’ section to select ‘Verify after programming’ option. This will verify hex file after loading.



After doing all the required settings, connect serial cable between Development Board and PC. Turn on the Board and click on the ‘Start’, it will ask to ‘RESET TO DEVICE IN TO ISP MODE’, now press RESET Switch on the Development Board, Flash magic will load hex file on the robot and verify it for correctness. If you are using USB to serial converter from NEX Robotics then find out COM port number with the help of documentation provided with the USB to serial converter.

Notice

The contents of this manual are subject to change without notice. All efforts have been made to ensure the accuracy of contents in this manual. However, should any errors be detected, NEX Robotics welcomes your corrections. You can send us your queries / suggestions at

info@nex-robotics.com



Content of this manual is released under the Creative Commons cc by-nc-sa license.

For legal information refer to:

<http://creativecommons.org/licenses/by-nc-sa/3.0/legalcode>



- ⚠ **Product's electronics is static sensitive. Use the product in static free environment.**
- ⚠ **Read the user manuals completely before start using this product**



Recycling:

Almost all the part of this product are recyclable. Please send this product to the recycling plant after its operational life. By recycling we can contribute to cleaner and healthier environment for the future generations.