PART 2

HARDWARE PROJECT DESCRIPTION

TOPIC: Electronic Voting Machine Using AVR Microcontroller AtMega32

DESCRIPTION - Create a digital voting system where users can cast votes for four candidates (A, B, C, and D) using push buttons and displaying the vote counts on a 16x2 LCD.

HARDWARE REQUIRED:

- AVR Microcontroller AtMega32
- 16x2 LCD
- Power Supply
- Resistors of 10K ohm
- Capacitor of 100nF
- Buttons
- ISP Programmer

HARDWARE DESCRIPTION:

- **AtMega32**: Acts as the brain of the system. It processes button inputs, updates vote counts, and controls the LCD.
- **16x2 LCD**: Displays vote counts for the candidates in real-time.
- **Resistors**: Pull Up resistors, Ensure the input pins of the microcontroller are not left in a floating state when buttons are not pressed.
- **Buttons:** One button for each candidate to cast votes. One additional RESET button used to reset the candidates votes to zero.

SOFTWARE USED:

- Microchip Studio
- Proteus Professional

CODE:

```
#include <avr/io.h>
```

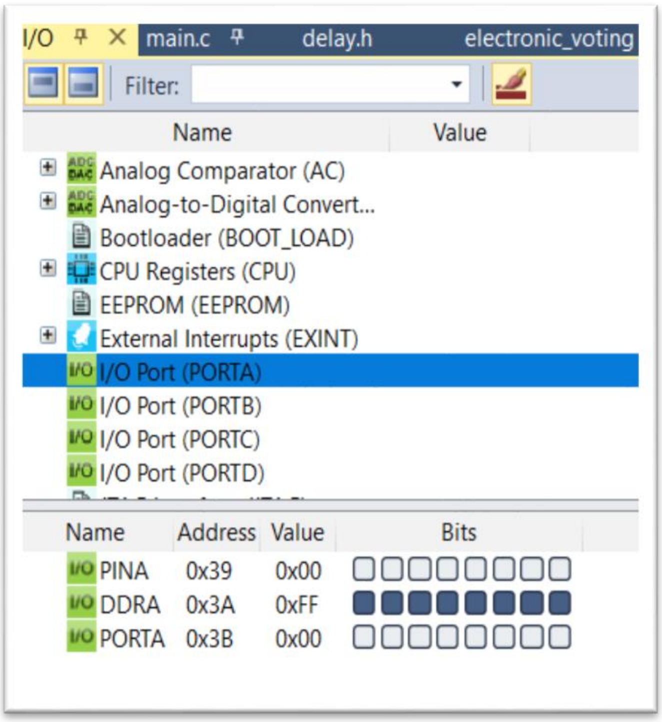
```
#include <util/delay.h>
#include <stdlib.h>

#define F_CPU 1000000UL
#define E 5
#define RS 6

void send_a_command(unsigned char command);
void send_a_character(unsigned char character);
void send_a_string(const char *str);
void display_count(int count, char *buffer, char  lcd_position);

int main(void)
{
        DDRA = 0xFF;
        DDRD = 0xFF;
        DDRB = 0x00;
        PORTB = 0xFF;

        _delay_ms(50);

        unsigned char COUNTA = 0;
        unsigned char COUNTB = 0;
        unsigned char COUNTC = 0;
        unsigned char COUNTD = 0;

        char SHOWA [3];
        char SHOWB [3];
        char SHOWC [3];
        char SHOWD [3];

        send_a_command(0x01);
        _delay_ms(50);
        send_a_command(0x38);
        _delay_ms(50);
        send_a_command(0x0F);

        while (1)
        {
                lcd_set_cursor(0, 0);
                send_a_string("A=");
                display_count(COUNTA, SHOWA, 0x80);

                lcd_set_cursor(0, 8);
                send_a_string("B=");
                display_count(COUNTB, SHOWB, 0x80 + 8);

                lcd_set_cursor(1, 0);
                send_a_string("C=");
                display_count(COUNTC, SHOWC, 0xC0);

                lcd_set_cursor(1, 8);
                send_a_string("D=");
                display_count(COUNTD, SHOWD, 0xC0 + 8);
```

```
if (bit_is_clear(PINB, 0))
{
        COUNTA++;
        _delay_ms(300);
        while (bit_is_clear(PINB, 0));
}
if (bit_is_clear(PINB, 1))
{
        COUNTB++;
        _delay_ms(300);
        while (bit_is_clear(PINB, 1));
}
if (bit_is_clear(PINB, 2))
{
        COUNTC++;
        _delay_ms(300);
        while (bit_is_clear(PINB, 2));
}
if (bit_is_clear(PINB, 3))
{
        COUNTD++;
        _delay_ms(300);
        while (bit_is_clear(PINB, 3));
}
if (bit_is_clear(PINB, 4))
{
```

```
                COUNTA = COUNTB = COUNTC = COUNTD = 0;
                _delay_ms(300);
                while (bit_is_clear(PINB, 4));
            }
        }
}
void send_a_command(unsigned char command)
{
        PORTA = command;
        PORTD &= ~(1 << RS);
        PORTD |= (1 << E);
        _delay_ms(1);
        PORTD &= ~(1 << E);
        PORTA = 0;
}
void send_a_character(unsigned char character)
{
        PORTA = character;
        PORTD |= (1 << RS);
        PORTD |= (1 << E);
        _delay_ms(1);
        PORTD &= ~(1 << E);
        PORTA = 0;
}
void send_a_string(const char *str)
{
        while (*str != '\0')
```

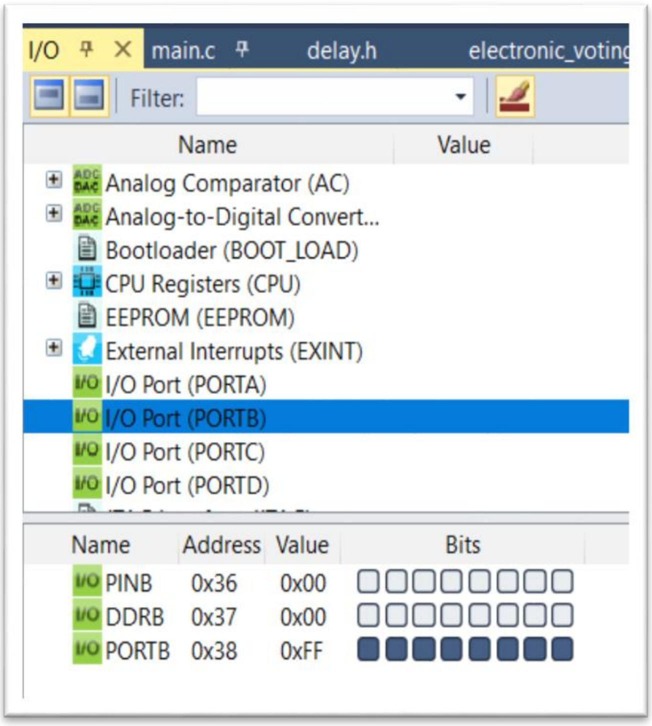
```
{
                send_a_character(*str);
                str++;
        }
}
void lcd_set_cursor(char row, char col)
{
        char address;
        if (row == 0)
        address = 0x80 + col;
        else
        address = 0xC0 + col;
        send_a_command(address);
}
void display_count(int count, char *buffer, char  lcd_position)
{
        if (count > 99) count = 99;
        itoa(count, buffer, 10);
        if (count < 10) {
                send_a_string("0");
        }
        send_a_string(buffer);
}
```

AVR OUTPUTS:



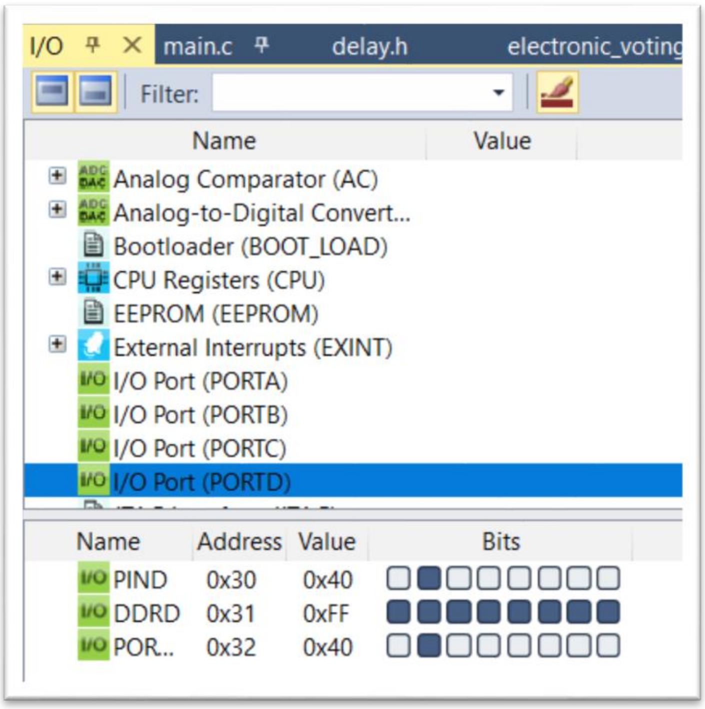
Name	Value
Analog Comparator (AC)	
Analog-to-Digital Convert...	
Bootloader (BOOT_LOAD)	
CPU Registers (CPU)	
EEPROM (EEPROM)	
External Interrupts (EXINT)	
I/O Port (PORTA)	
I/O Port (PORTB)	
I/O Port (PORTC)	
I/O Port (PORTD)	

Name	Address	Value	Bits
PINA	0x39	0x00	□□□□□□□□
DDRA	0x3A	0xFF	■■■■■■■■
PORTA	0x3B	0x00	□□□□□□□□



Name	Value
Analog Comparator (AC)	
Analog-to-Digital Convert...	
Bootloader (BOOT_LOAD)	
CPU Registers (CPU)	
EEPROM (EEPROM)	
External Interrupts (EXINT)	
I/O Port (PORTA)	
I/O Port (PORTB)	
I/O Port (PORTC)	
I/O Port (PORTD)	

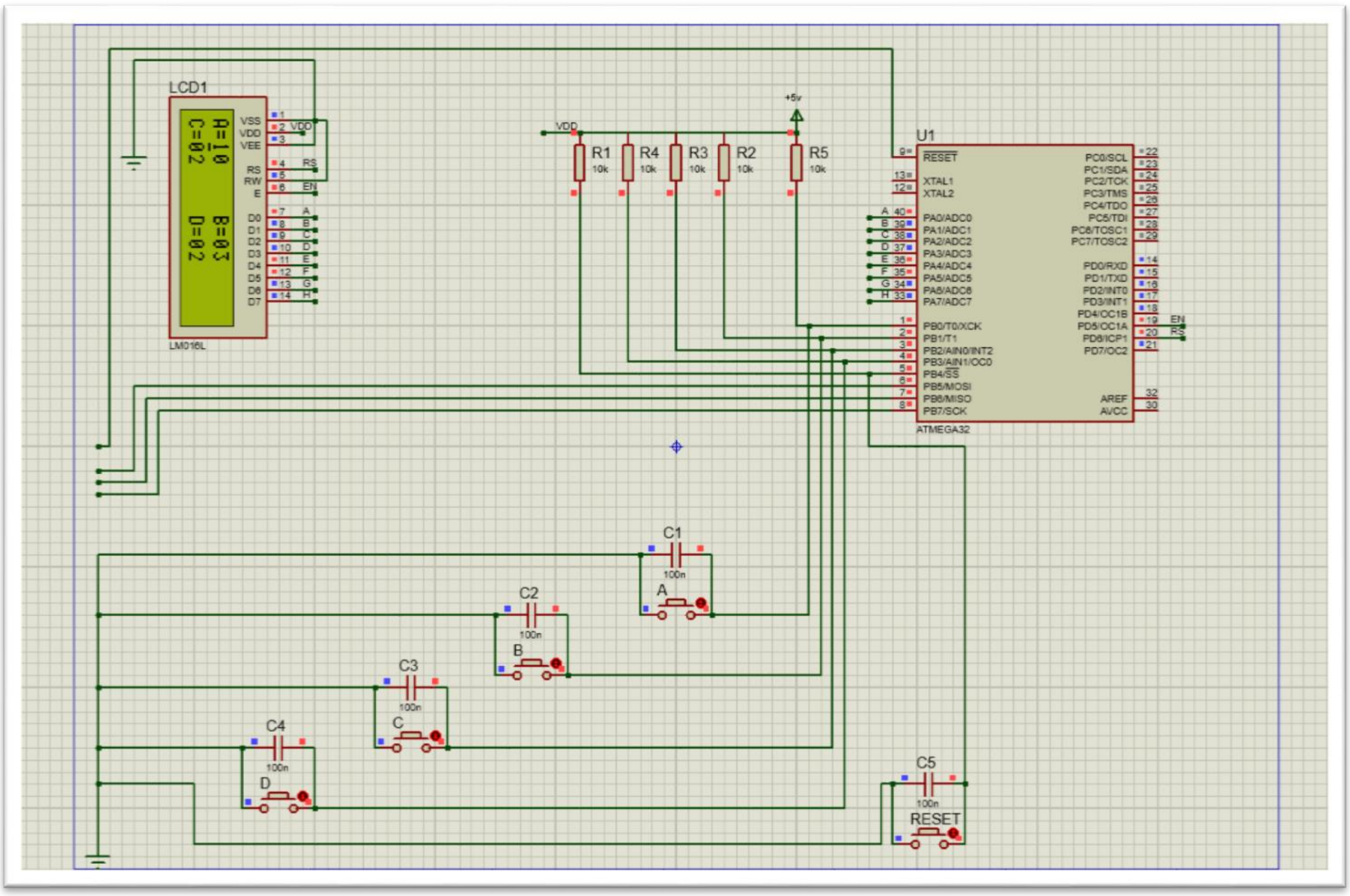
Name	Address	Value	Bits
PINB	0x36	0x00	□□□□□□□□
DDRB	0x37	0x00	□□□□□□□□
PORTB	0x38	0xFF	■■■■■■■■



Name	Value
Analog Comparator (AC)	
Analog-to-Digital Convert...	
Bootloader (BOOT_LOAD)	
CPU Registers (CPU)	
EEPROM (EEPROM)	
External Interrupts (EXINT)	
I/O Port (PORTA)	
I/O Port (PORTB)	
I/O Port (PORTC)	
I/O Port (PORTD)	

Name	Address	Value	Bits
PIND	0x30	0x40	□■□□□□□□
DDRD	0x31	0xFF	■■■■■■■■
POR...	0x32	0x40	□■□□□□□□

CIRCUIT AND PROTEUS SIMULATION



Description on different types of hardware that can be used in your project other than what you have used.

Microcontroller: ATmega328P (Arduino Uno): Offers an easier-to-use platform with built-in bootloader support. Can simplify programming via the Arduino IDE.

Display Alternative:

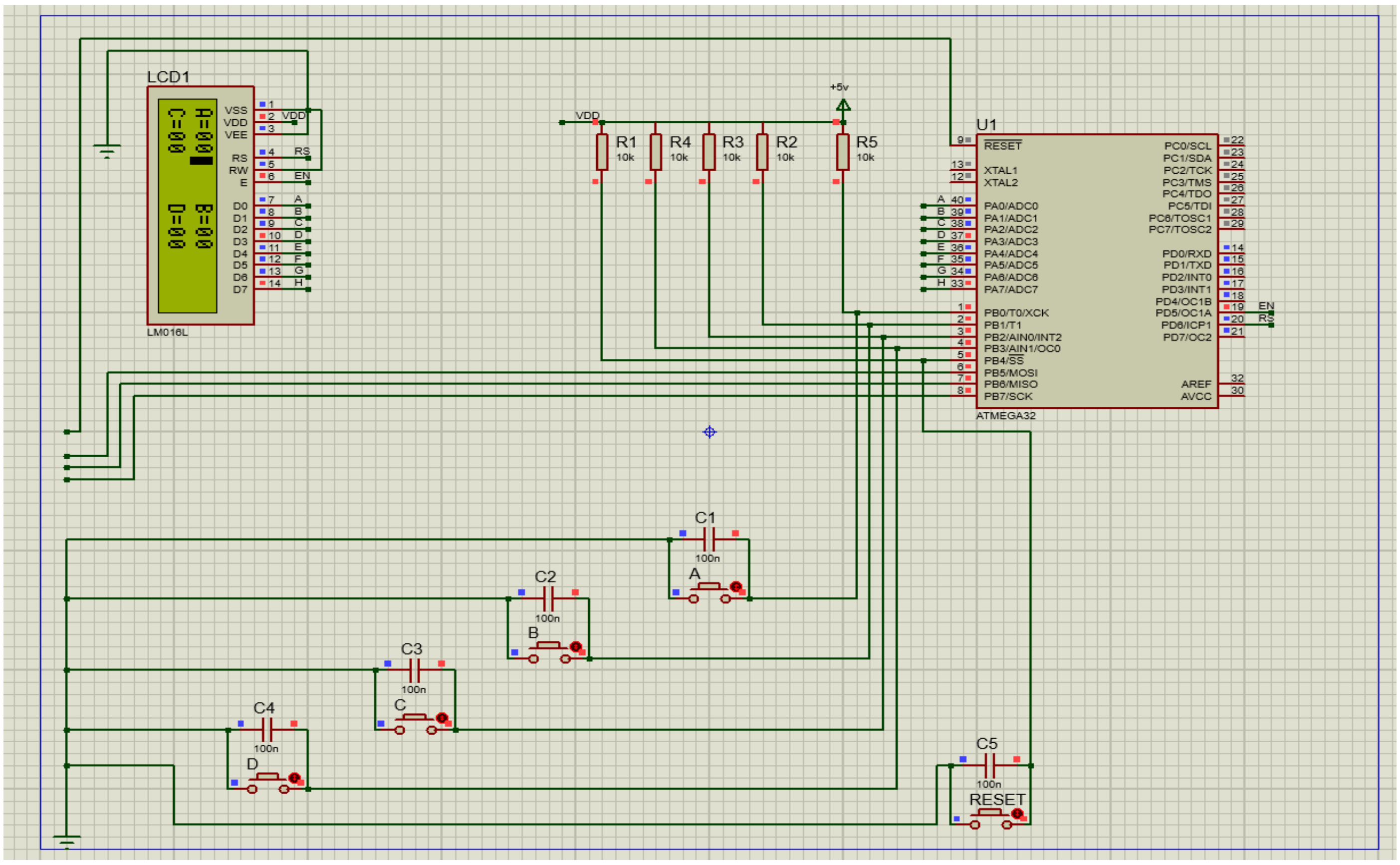
- 7 Segment Displays
- Oled Display

Input Device Alternatives:

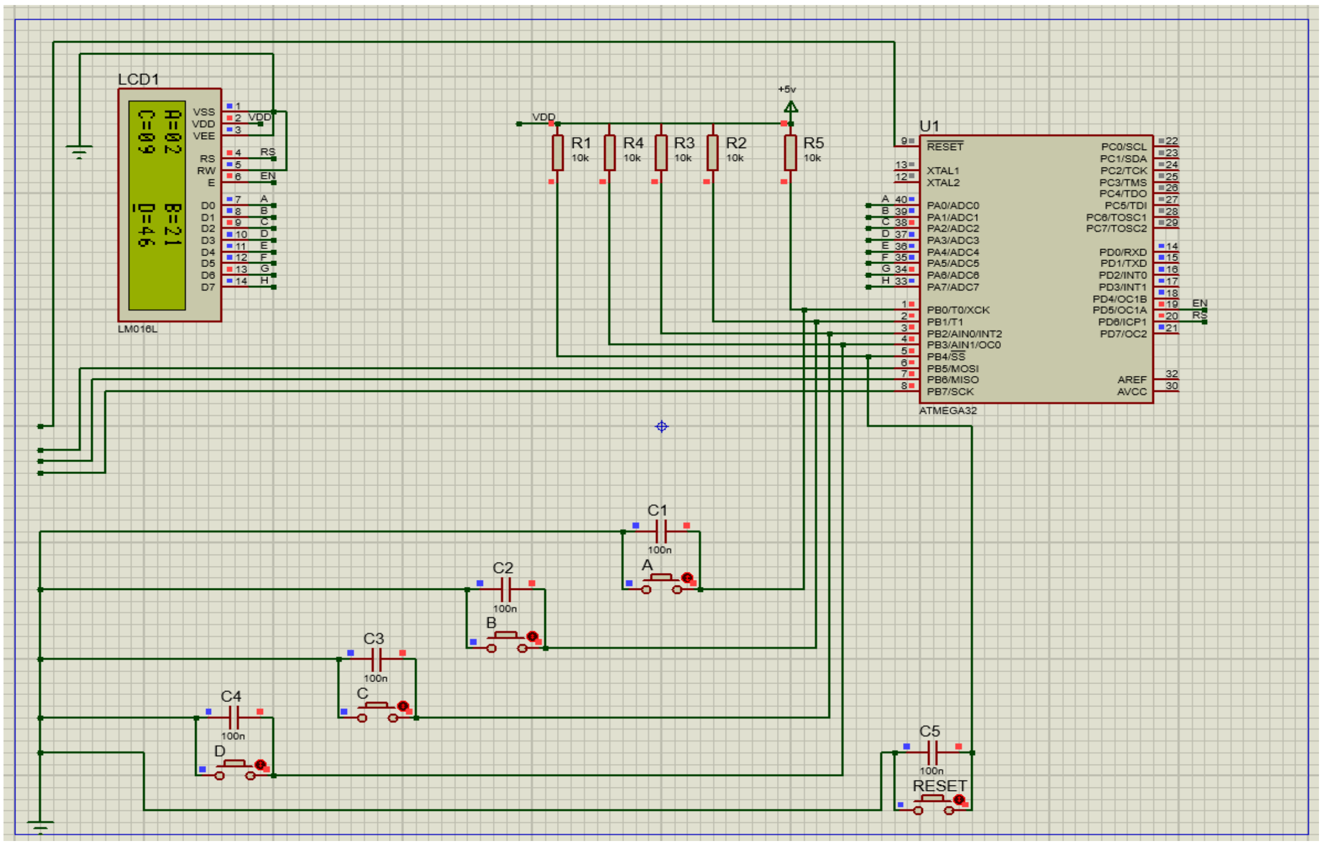
- Touch Buttons
- Keyboard for a greater number of Candidates

WORKING OF VOTING MACHINE:

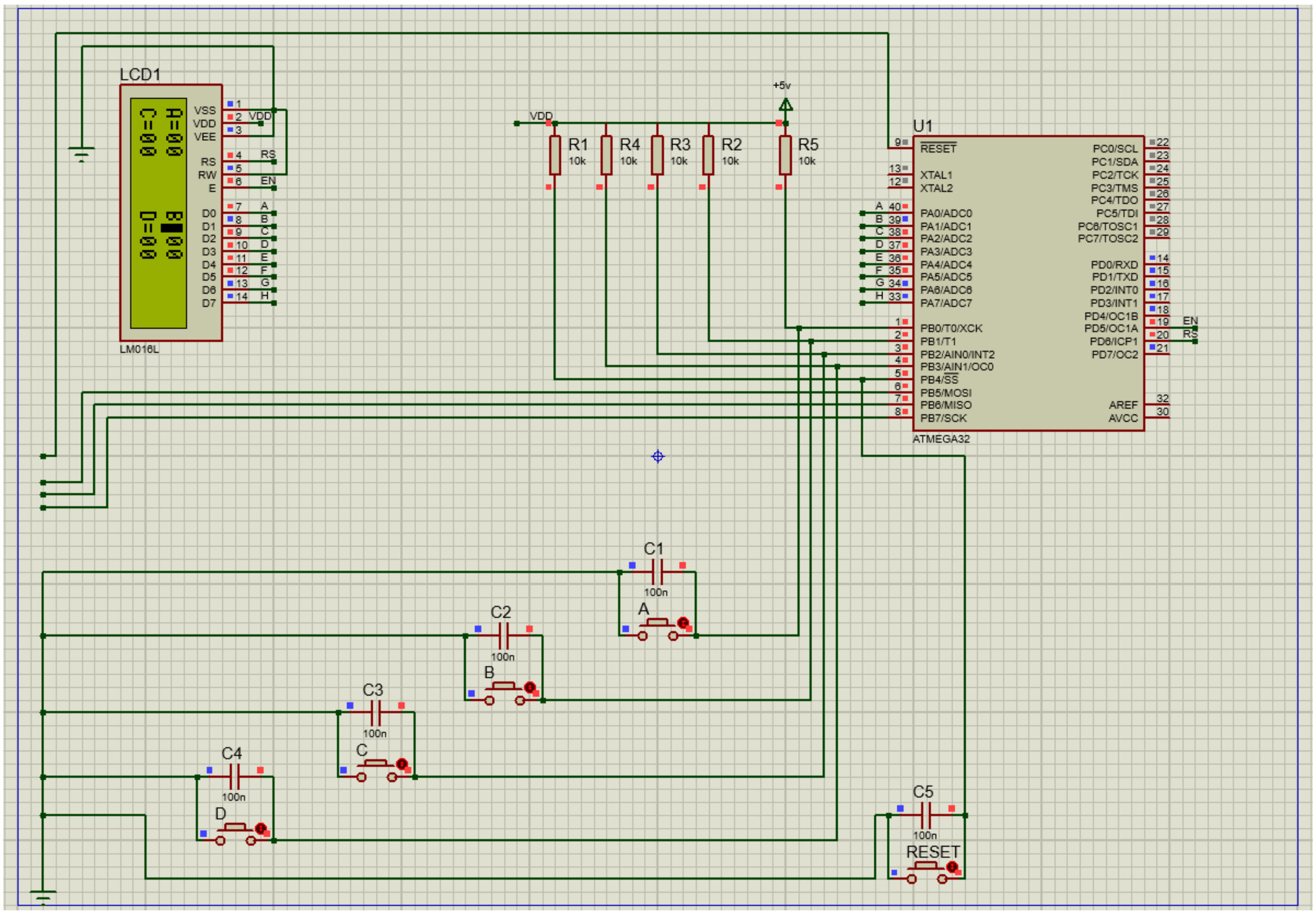
1. Running of Simulation



2. Voting for different Candidates



3. Resetting the votes to ZERO. Using the RESET button



SOME VIVA BASED QUESTIONS

1. What are the functions of PORTA, PORTB, and PORTD in this project?

- **PORTA**: Used to send data and commands to the LCD. It serves as the data bus for communication between the microcontroller and the LCD.
- **PORTB**: Configured as input for reading the state of the buttons. Internal pull-up resistors are enabled to prevent floating inputs.
- **PORTD**: Used for control signals (RS and E) to the LCD.

2. What happens when a pin of PORTB is set low?

When a pin of PORTB is set low, it means that the corresponding button connected to that pin has been pressed. Since internal pull-up resistors are enabled, the pin is normally high. Pressing the button creates a low signal by connecting the pin to ground. This state is detected using the bit_is_clear() macro.

3. What are the commands 0x01, 0x38, and 0x0F used for?

- **0x01**: Clears the LCD display and moves the cursor to the home position (top-left corner).
- **0x38**: Configures the LCD for 8-bit mode and two-line display with a 5x8 dot matrix character font.
- **0x0F**: Turns on the LCD display, enables the cursor, and sets the cursor to blinking mode.

4. What is the purpose of the itoa function in this project?
The ITOA function is used to convert an integer value (the vote count) into a string. This is necessary because the LCD can only display characters, not numerical data directly. By converting the vote count to a string, it can be sent to the LCD for display.

5.Why is a debounce delay used in the project?

- When a button is pressed or released, mechanical contacts can cause multiple rapid transitions (bouncing) before settling into a stable state. This can lead to erroneous multiple detections of a single button press. The debounce delay ensures that the button is stable before the program processes the input, preventing such false triggers.

6.How do you position the cursor on a specific row and column of the LCD?

To position the cursor on a specific row and column of the LCD, the lcd_set_cursor function calculates the correct address based on the row and column values.

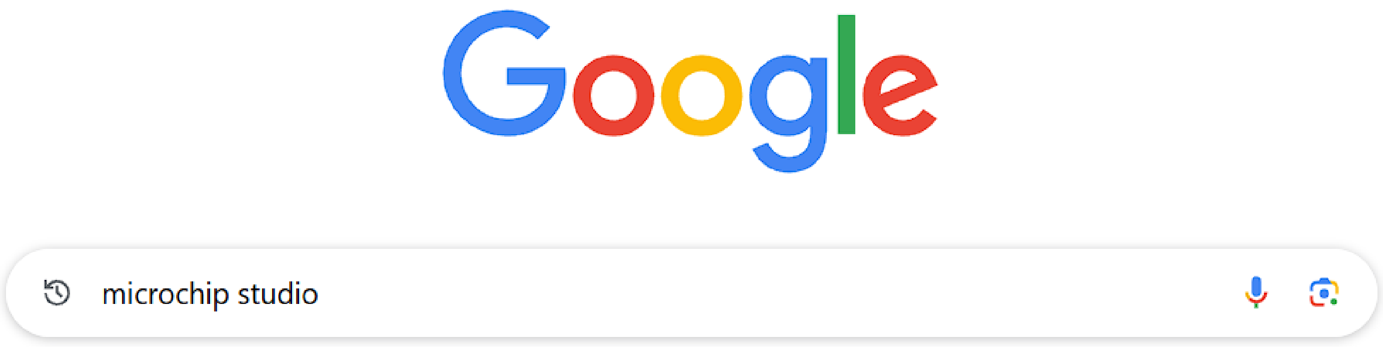
- For the first row, the address is 0x80 + col.
- For the second row, the address is 0xC0 + col.
- The calculated address is sent to the LCD as a command using the send_a_command function. For example:
- lcd_set_cursor(0, 4) moves the cursor to the first row, 5th column.
- lcd_set_cursor(1, 9) moves it to the second row, 10th column.

Installation of AVR Studio and Proteus Simulation

STEP TO SETUP MICROCHIP STUDIO

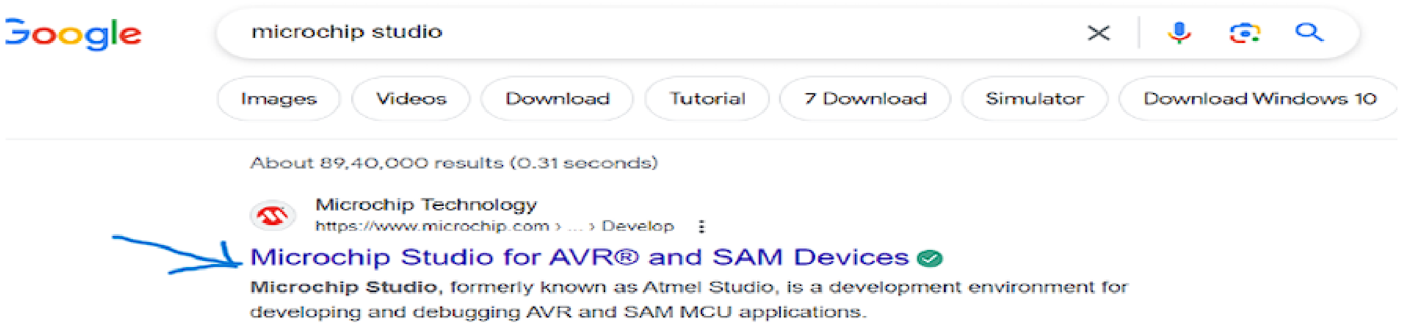
STEP 1.

Search Microchip studio on Google



STEP 2.

Click on the First Link as shown



About 89,40,000 results (0.31 seconds)

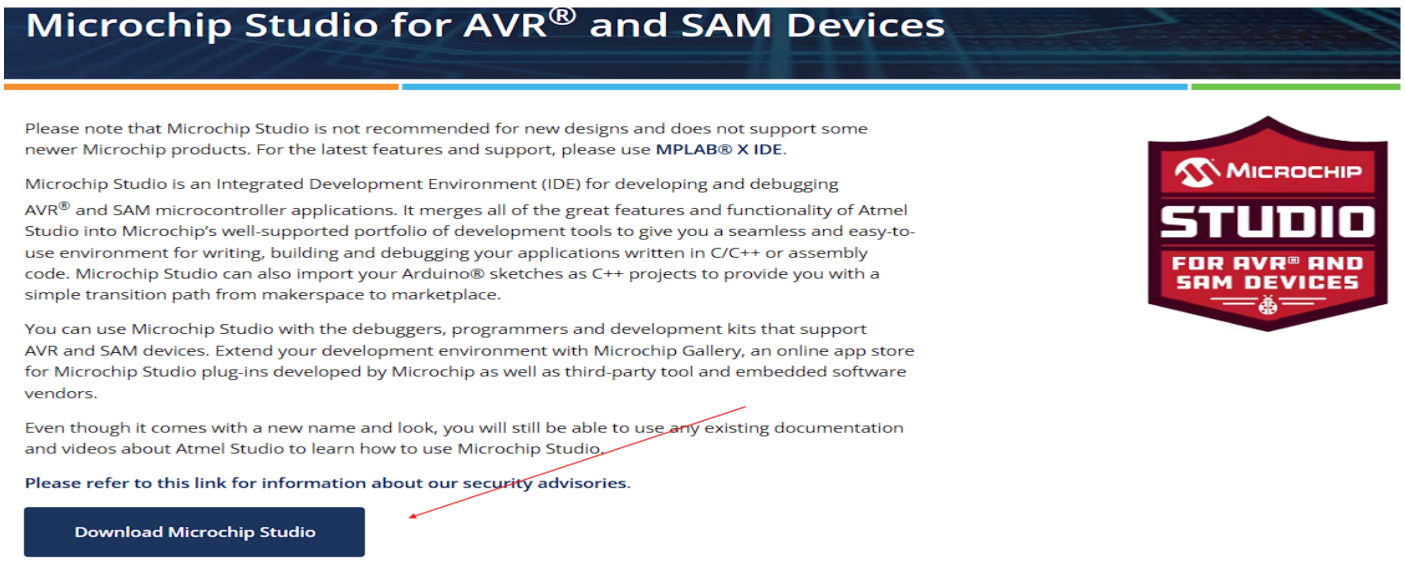
Microchip Technology
https://www.microchip.com › ... › Develop

Microchip Studio for AVR® and SAM Devices

Microchip Studio, formerly known as Atmel Studio, is a development environment for developing and debugging AVR and SAM MCU applications.

STEP 3.

Click on the download tab.


Microchip Studio for AVR® and SAM Devices

Please note that Microchip Studio is not recommended for new designs and does not support some newer Microchip products. For the latest features and support, please use MPLAB® X IDE.

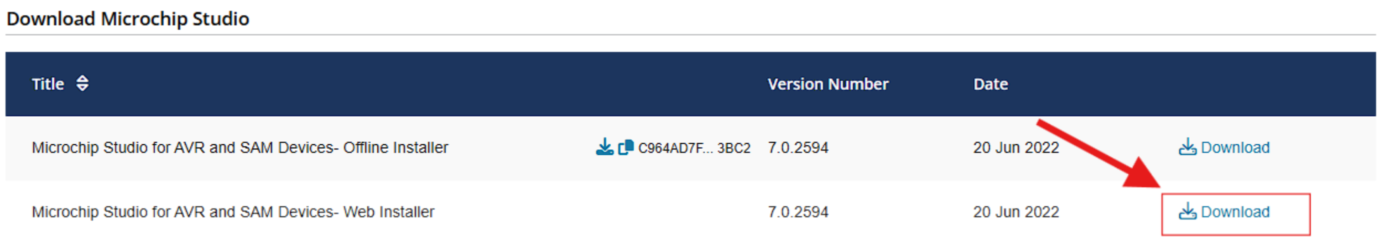
Microchip Studio is an Integrated Development Environment (IDE) for developing and debugging AVR® and SAM microcontroller applications. It merges all of the great features and functionality of Atmel Studio into Microchip's well-supported portfolio of development tools to give you a seamless and easy-to-use environment for writing, building and debugging your applications written in C/C++ or assembly code. Microchip Studio can also import your Arduino® sketches as C++ projects to provide you with a simple transition path from makerspace to marketplace.

You can use Microchip Studio with the debuggers, programmers and development kits that support AVR and SAM devices. Extend your development environment with Microchip Gallery, an online app store for Microchip Studio plug-ins developed by Microchip as well as third-party tool and embedded software vendors.

Even though it comes with a new name and look, you will still be able to use any existing documentation and videos about Atmel Studio to learn how to use Microchip Studio.

Please refer to this link for information about our security advisories.

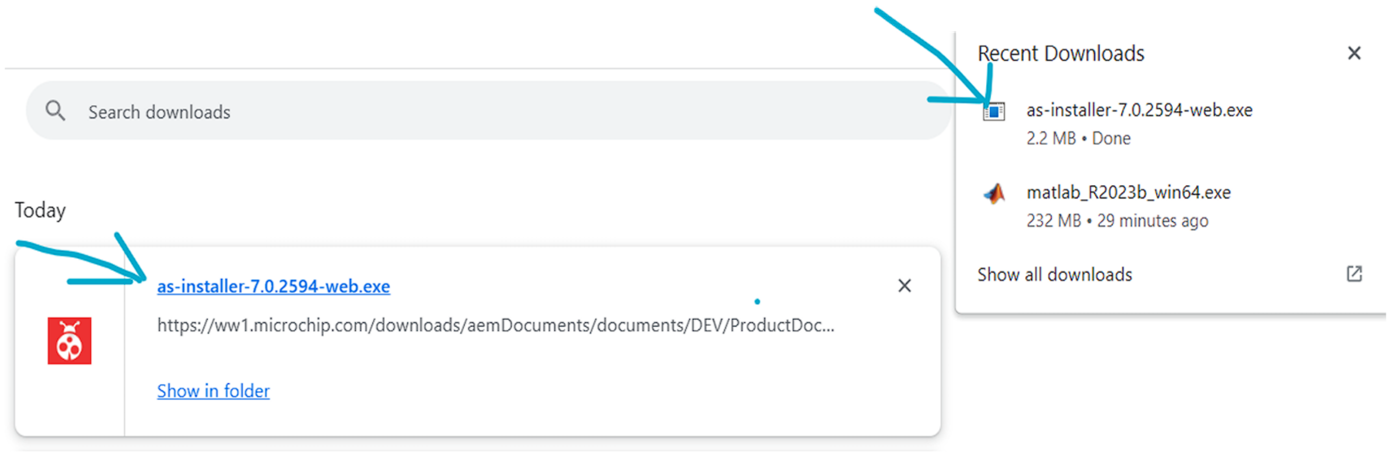
And now again click on the download tab.


Download Microchip Studio

Title		Version Number	Date	
Microchip Studio for AVR and SAM Devices- Offline Installer	C964AD7F... 3BC2	7.0.2594	20 Jun 2022	Download
Microchip Studio for AVR and SAM Devices- Web Installer		7.0.2594	20 Jun 2022	Download

STEP 4.

Software will be downloaded



as-installer-7.0.2594-web.exe
https://ww1.microchip.com/downloads/aemDocuments/documents/DEV/ProductDoc...
Show in folder

Recent Downloads
as-installer-7.0.2594-web.exe
2.2 MB • Done
matlab_R2023b_win64.exe
232 MB • 29 minutes ago
Show all downloads

STEP 5. NOW INSTALL THE SOFTWARE INTO YOUR PC.

• Verify the hardware and software requirements from the "System Requirements" section

• Make sure your user have local administrator privileges

• Save all your work before starting. The installation might prompt you to restart, if required.

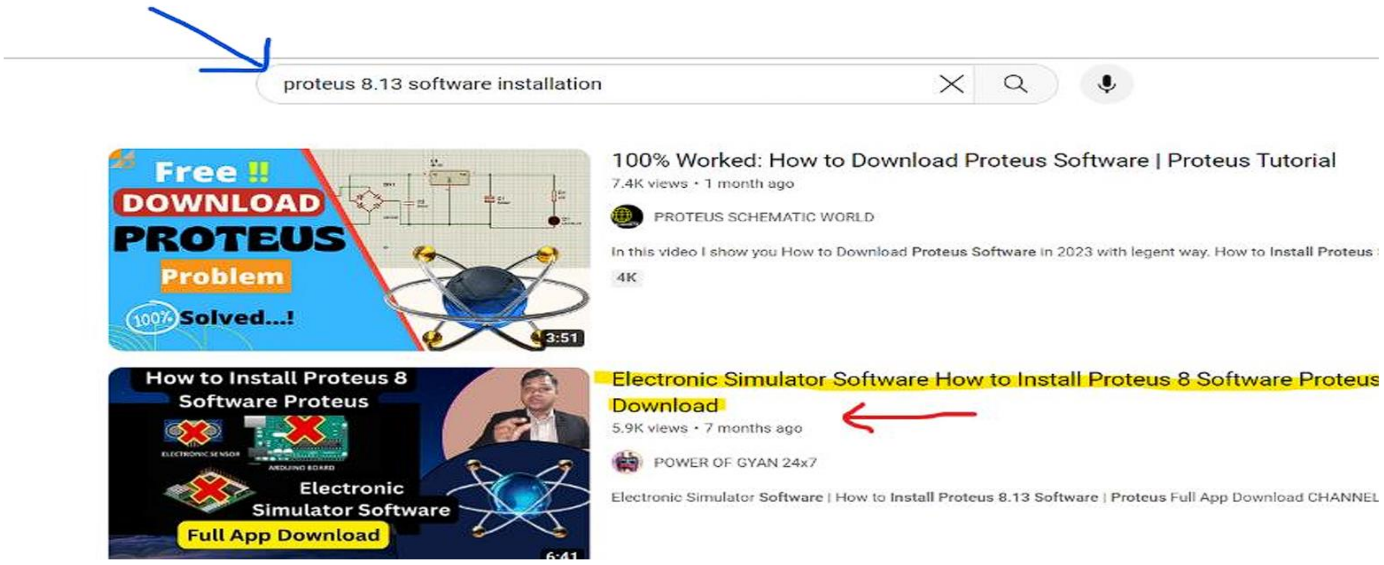
• Disconnect all Atmel USB/Serial hardware devices

• Double click the installer executable file and follow the installation wizard

• Once finished, the installer displays an option to Start Atmel Studio after completion. If you choose to open, then note that Atmel Studio will launch with administrative privileges, since the installer was either launched as administrator or with elevated privileges.

INSTALLING PROTEUS 8.13 PROFESSIONAL

The link for the above software is given below. just open the drive through the link, download and then install proteus on your pc. the screenshot for the same are attached below:
1.) Search "proteus 8.13 software installation on youtube."
2.) Open the shown video and open the drive link in its description, and follow the steps

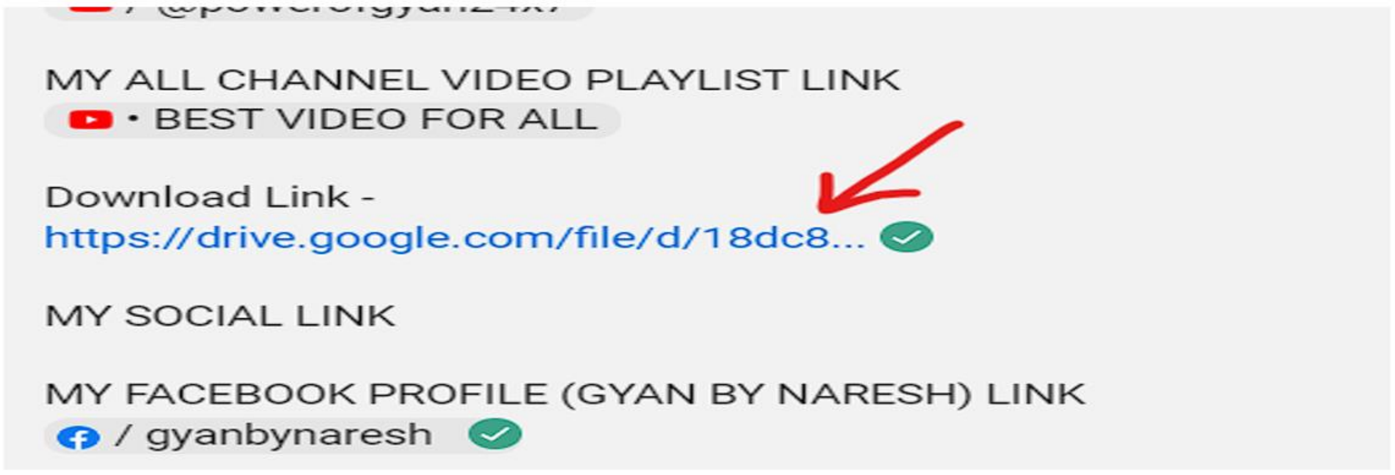

proteus 8.13 software installation

100% Worked: How to Download Proteus Software | Proteus Tutorial
7.4K views • 1 month ago
PROTEUS SCHEMATIC WORLD
In this video I show you How to Download Proteus Software in 2023 with legent way. How to Install Proteus
4K

Electronic Simulator Software How to Install Proteus 8 Software Proteus Download
5.9K views • 7 months ago
POWER OF GYAN 24x7
Electronic Simulator Software | How to Install Proteus 8.13 Software | Proteus Full App Download CHANNEL

Free !!
DOWNLOAD
PROTEUS
Problem
100% Solved...!
3:51
How to Install Proteus 8 Software Proteus
Electronic Simulator Software
Full App Download

Open the description

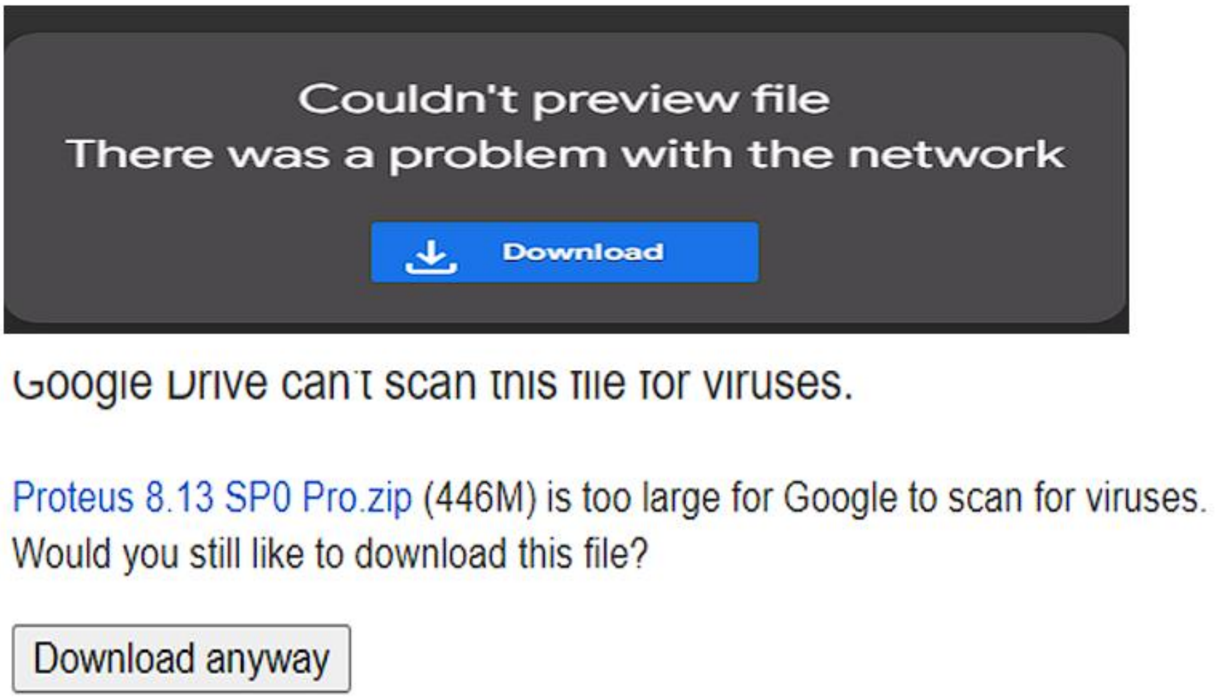

MY ALL CHANNEL VIDEO PLAYLIST LINK
• BEST VIDEO FOR ALL

Download Link -
https://drive.google.com/file/d/18dc8...

MY SOCIAL LINK

MY FACEBOOK PROFILE (GYAN BY NARESH) LINK
/ gyanbynaresh

Click on the drive link


Couldn't preview file
There was a problem with the network

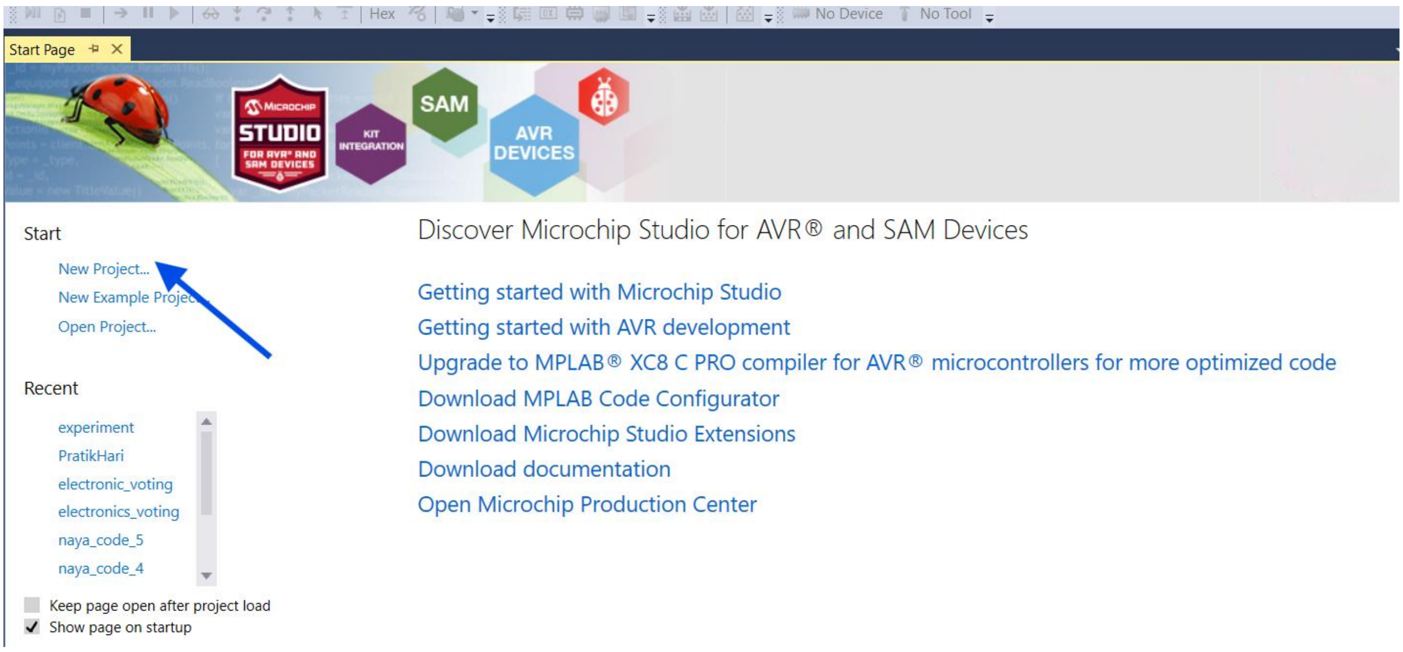
Google Drive can't scan this file for viruses.

Proteus 8.13 SP0 Pro.zip (446M) is too large for Google to scan for viruses. Would you still like to download this file?

And download it…

STARTING OF MICROCHIP STUDIO

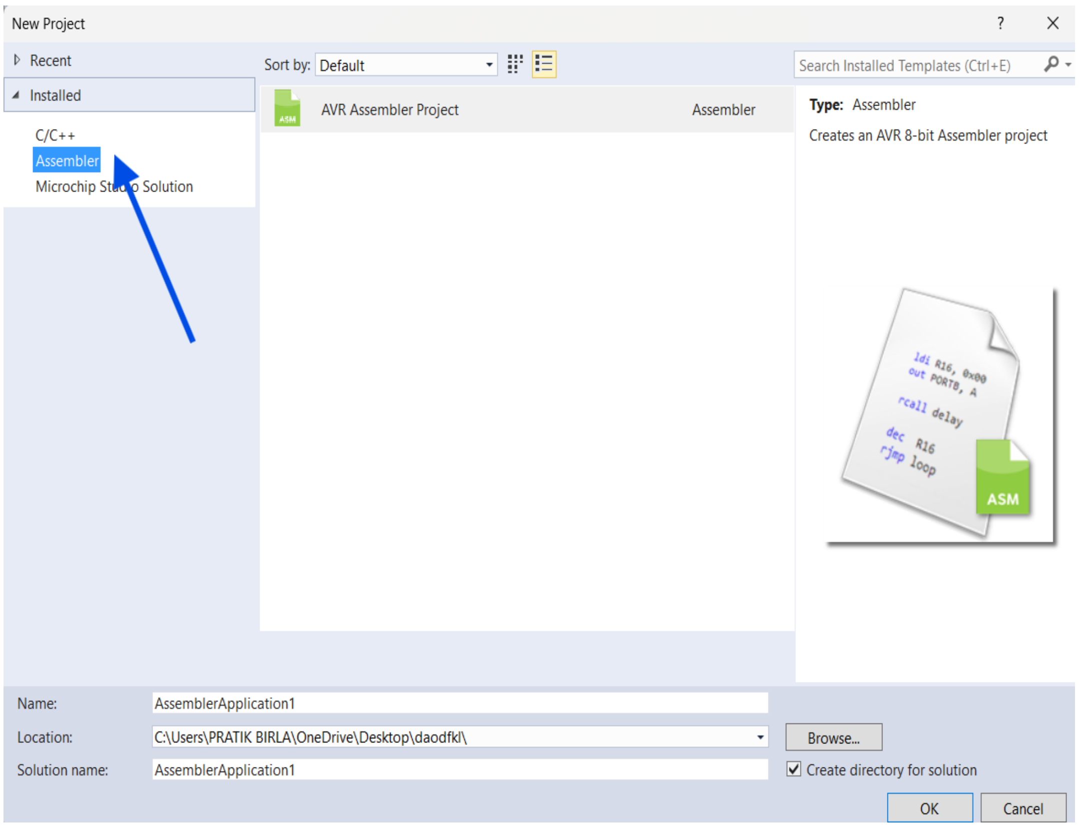
- Click on the new Project



Discover Microchip Studio for AVR® and SAM Devices

Getting started with Microchip Studio
Getting started with AVR development
Upgrade to MPLAB® XC8 C PRO compiler for AVR® microcontrollers for more optimized code
Download MPLAB Code Configurator
Download Microchip Studio Extensions
Download documentation
Open Microchip Production Center

- Select the
  "ASSEMBLER" for the code in Assembly language
  if the code in C/C++ select the above option
- And Enter the project name


New Project

- Recent
- Installed
  - C/C++
  - Assembler
  - Microchip Studio Solution

Sort by: Default

Search Installed Templates (Ctrl+E)

AVR Assembler Project	Assembler

Type: Assembler

Creates an AVR 8-bit Assembler project

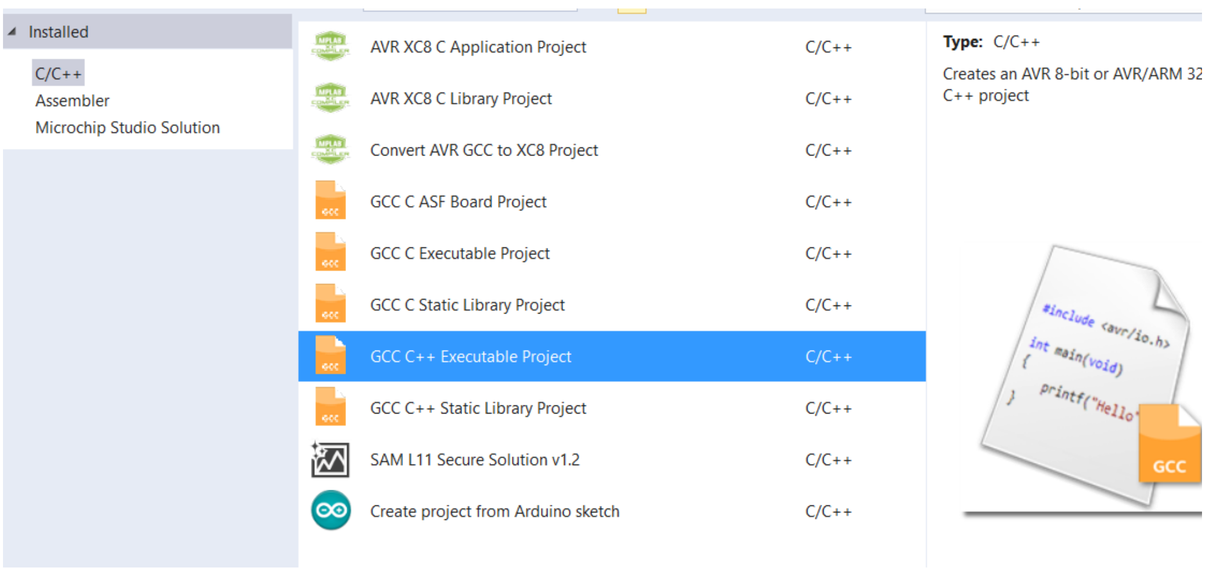
```
ldi R16, 0x00
out PORTB, A
rcall delay
dec R16
rjmp loop
```

Name:	AssemblerApplication1
Location:	C:\Users\PRATIK BIRLA\OneDrive\Desktop\daodfkl\
Solution name:	AssemblerApplication1

Browse...

[✓] Create directory for solution

OK Cancel


- Installed
  - C/C++
  - Assembler
  - Microchip Studio Solution

AVR XC8 C Application Project	C/C++
AVR XC8 C Library Project	C/C++
Convert AVR GCC to XC8 Project	C/C++
GCC C ASF Board Project	C/C++
GCC C Executable Project	C/C++
GCC C Static Library Project	C/C++
GCC C++ Executable Project	C/C++
GCC C++ Static Library Project	C/C++
SAM L11 Secure Solution v1.2	C/C++
Create project from Arduino sketch	C/C++

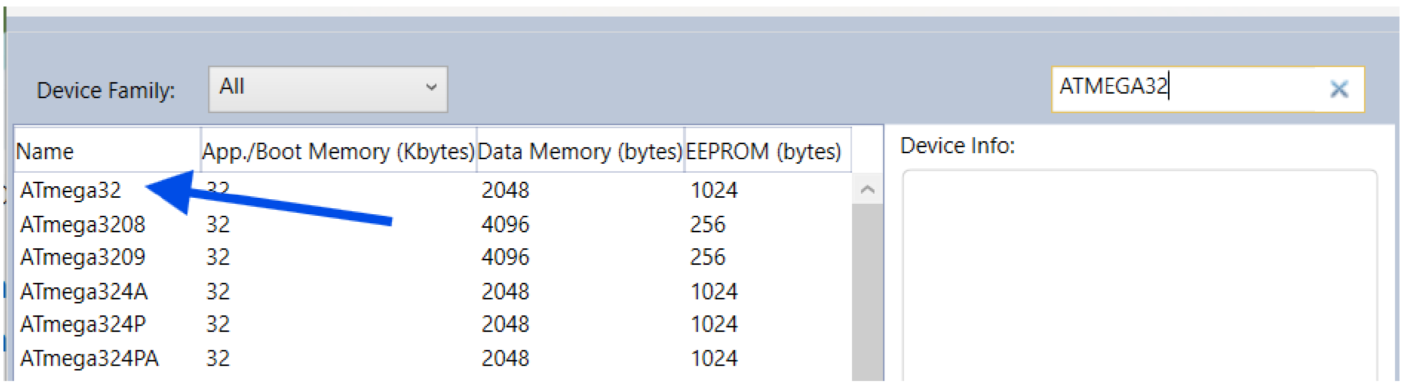
Type: C/C++

Creates an AVR 8-bit or AVR/ARM 32

C++ project

```
#include <avr/io.h>
int main(void)
{
printf("Hello
}
```

- Now Choose the Microcontroller

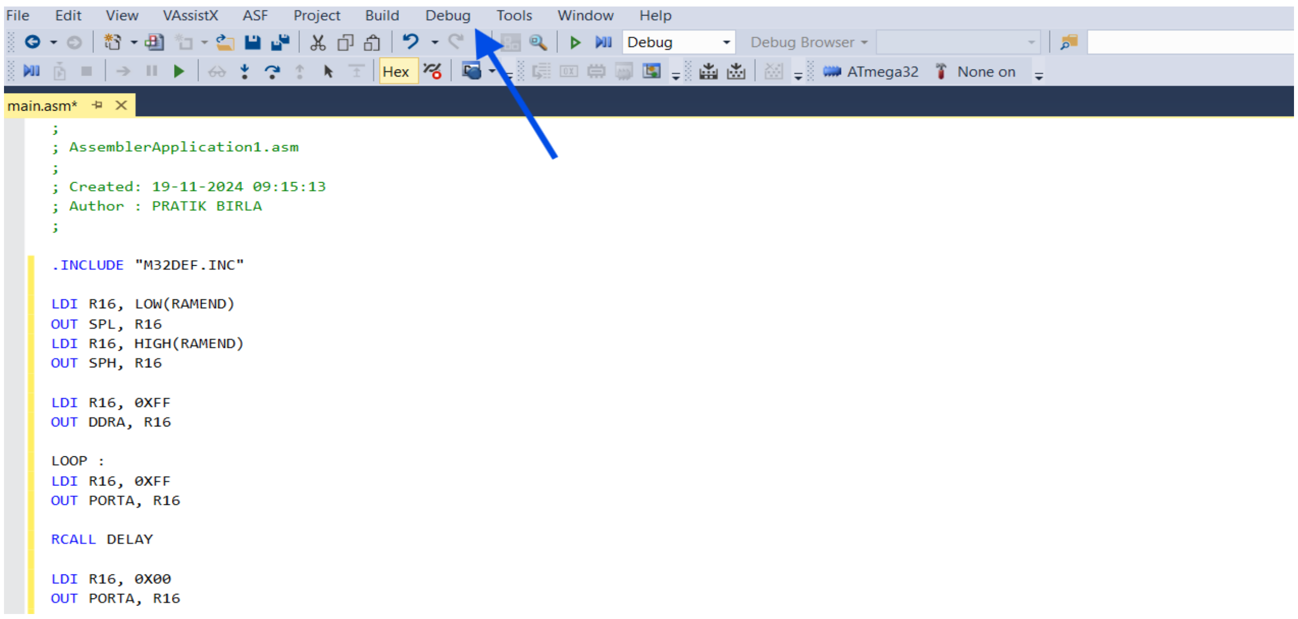

Device Family: All

ATMEGA32

Name	App./Boot Memory (Kbytes)	Data Memory (bytes)	EEPROM (bytes)
ATmega32	32	2048	1024
ATmega3208	32	4096	256
ATmega3209	32	4096	256
ATmega324A	32	2048	1024
ATmega324P	32	2048	1024
ATmega324PA	32	2048	1024

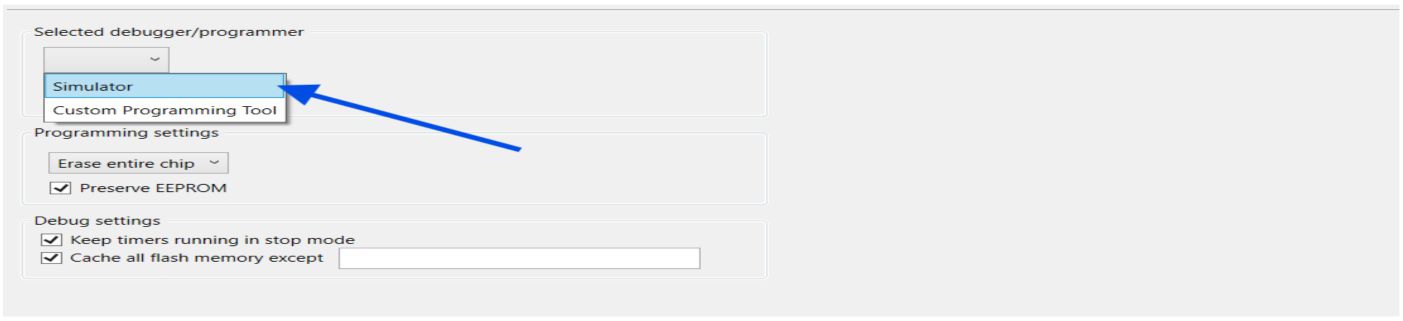
Device Info:

- Delete the given program and starting coding your program(fig.6 a&b)



```
;
; AssemblerApplication1.asm
;
; Created: 19-11-2024 09:15:13
; Author : PRATIK BIRLA
;

.INCLUDE "M32DEF.INC"

LDI R16, LOW(RAMEND)
OUT SPL, R16
LDI R16, HIGH(RAMEND)
OUT SPH, R16

LDI R16, 0XFF
OUT DDRA, R16

LOOP :
LDI R16, 0XFF
OUT PORTA, R16

RCALL DELAY

LDI R16, 0X00
OUT PORTA, R16
```

- After coding the entire program,click on debug on the menu barand select continue or press f5 key window 2 will appear for the first time.now click ok and select assembler,and open the main.asm file.and again press f5 key.(if there are no errors the code will be deugged and following window will appear else errors window will be shown).
- Click on continue when the pop-up box appear and select the simulator



Selected debugger/programmer

Simulator
Custom Programming Tool

Programming settings

Erase entire chip

☑ Preserve EEPROM

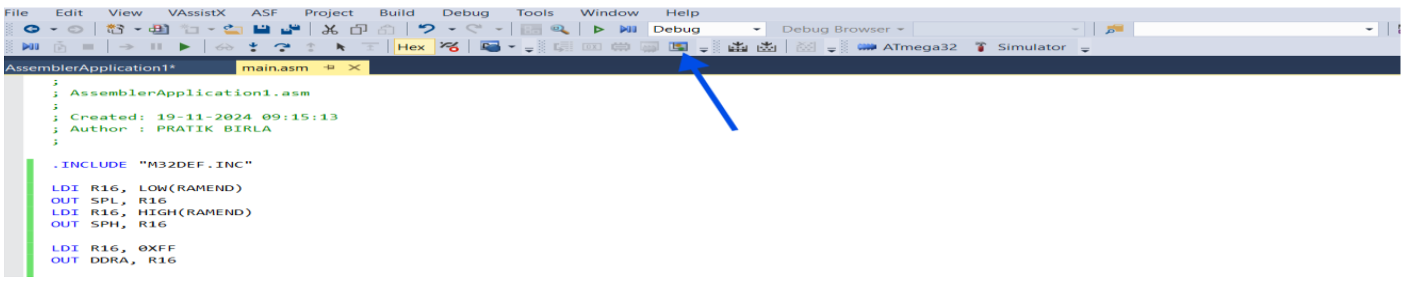
Debug settings

☑ Keep timers running in stop mode

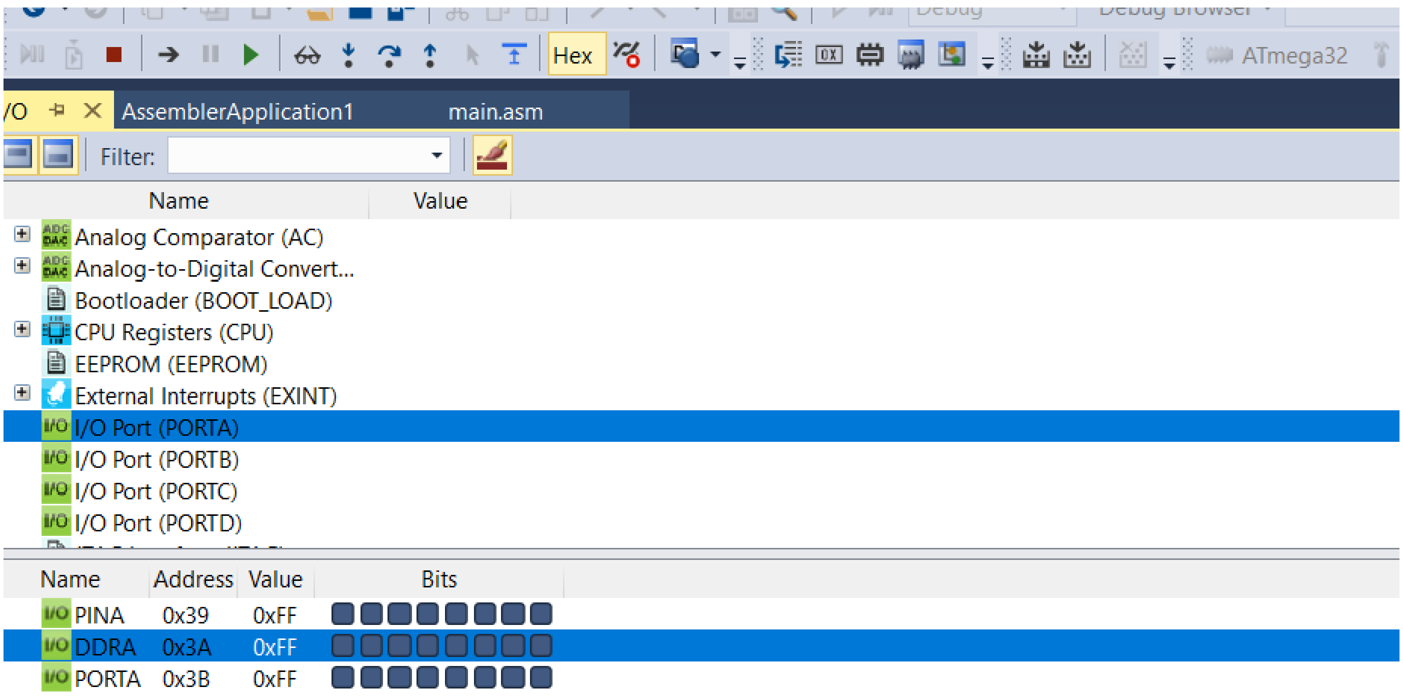
☑ Cache all flash memory except

- Eliminate the error and debugg again.

- Now the program is sucessfully debugged.search for i/o port (4th point) on tool bar menu to read the status of different ports.(shown below).



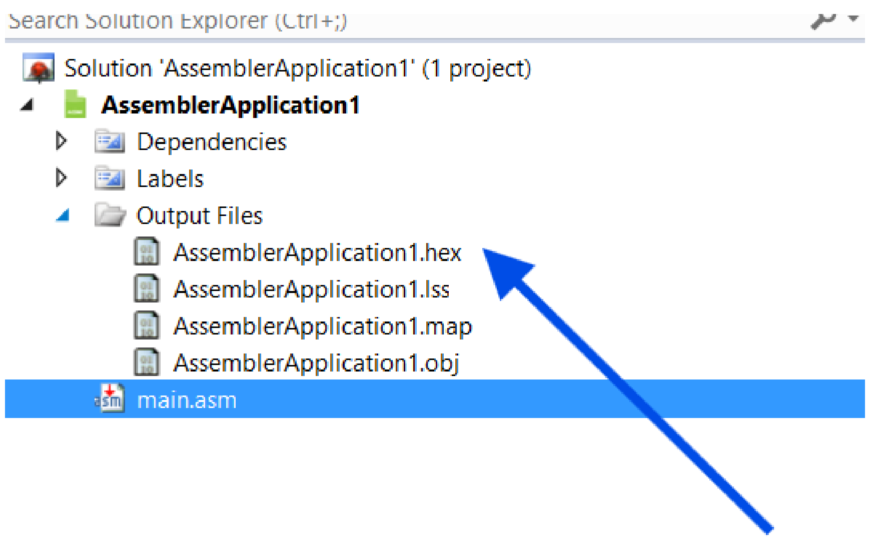
```
;
; AssemblerApplication1.asm
;
; Created: 19-11-2024 09:15:13
; Author : PRATIK BIRLA
;

.INCLUDE "M32DEF.INC"

LDI R16, LOW(RAMEND)
OUT SPL, R16
LDI R16, HIGH(RAMEND)
OUT SPH, R16

LDI R16, 0XFF
OUT DDRA, R16
```



Name	Value
Analog Comparator (AC)	
Analog-to-Digital Convert...	
Bootloader (BOOT_LOAD)	
CPU Registers (CPU)	
EEPROM (EEPROM)	
External Interrupts (EXINT)	
I/O Port (PORTA)	
I/O Port (PORTB)	
I/O Port (PORTC)	
I/O Port (PORTD)	

Name	Address	Value	Bits
PINA	0x39	0xFF	■■■■■■■■
DDRA	0x3A	0xFF	■■■■■■■■
PORTA	0x3B	0xFF	■■■■■■■■

- Accessing hex file of the program.
- a) first stop debugging(ctrl+shift+f5) and click main.asm file(fig.7)
- b) click on the output files in the solution explorer and select.
- Hex file copy the path of the file. (the path will be used to load/burn the hex file on the avr chip)



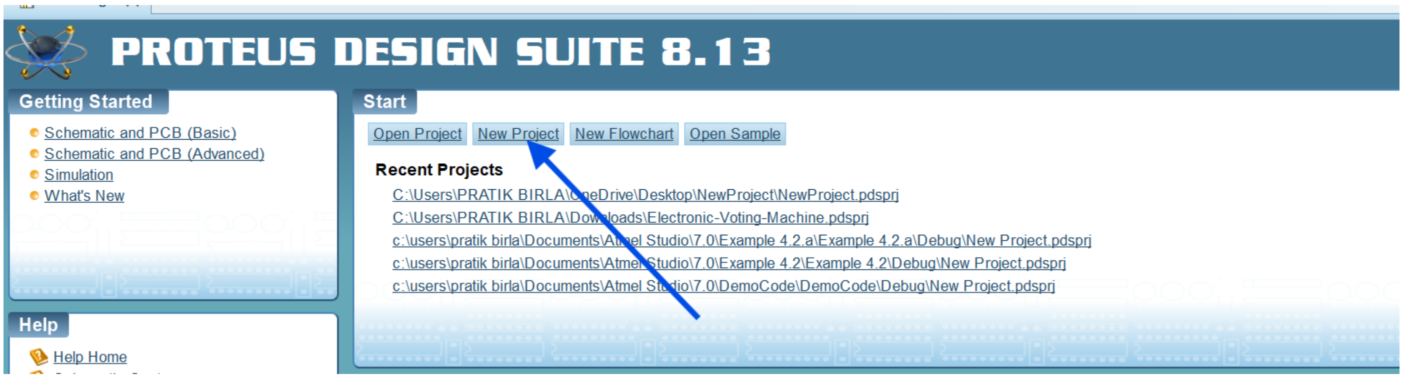
```
Solution 'AssemblerApplication1' (1 project)
  AssemblerApplication1
    Dependencies
    Labels
    Output Files
      AssemblerApplication1.hex
      AssemblerApplication1.lss
      AssemblerApplication1.map
      AssemblerApplication1.obj
    main.asm
```

NOTE:

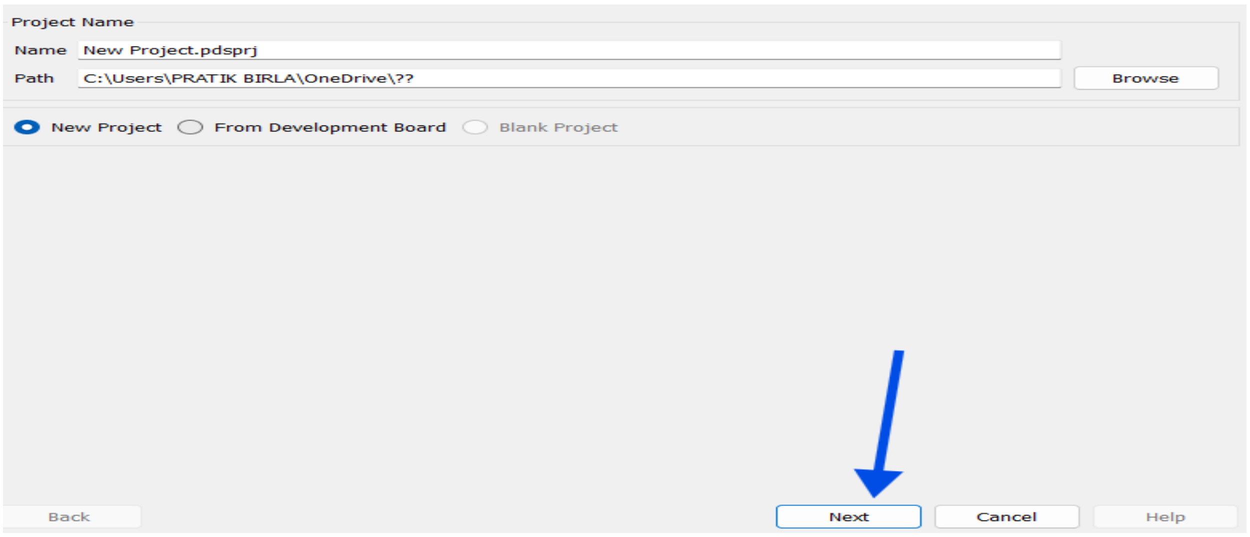
THE ABOVE CODE AND DEBUGGING IS SHOWN FOR ASSEMBLY LANGUAGE PRORAM. FOR C/C++ THE PROCESS IS NEARLY SAME.

STARTING PROTEUS SIMULATION

- **Open the software**
- **Click on new Project**

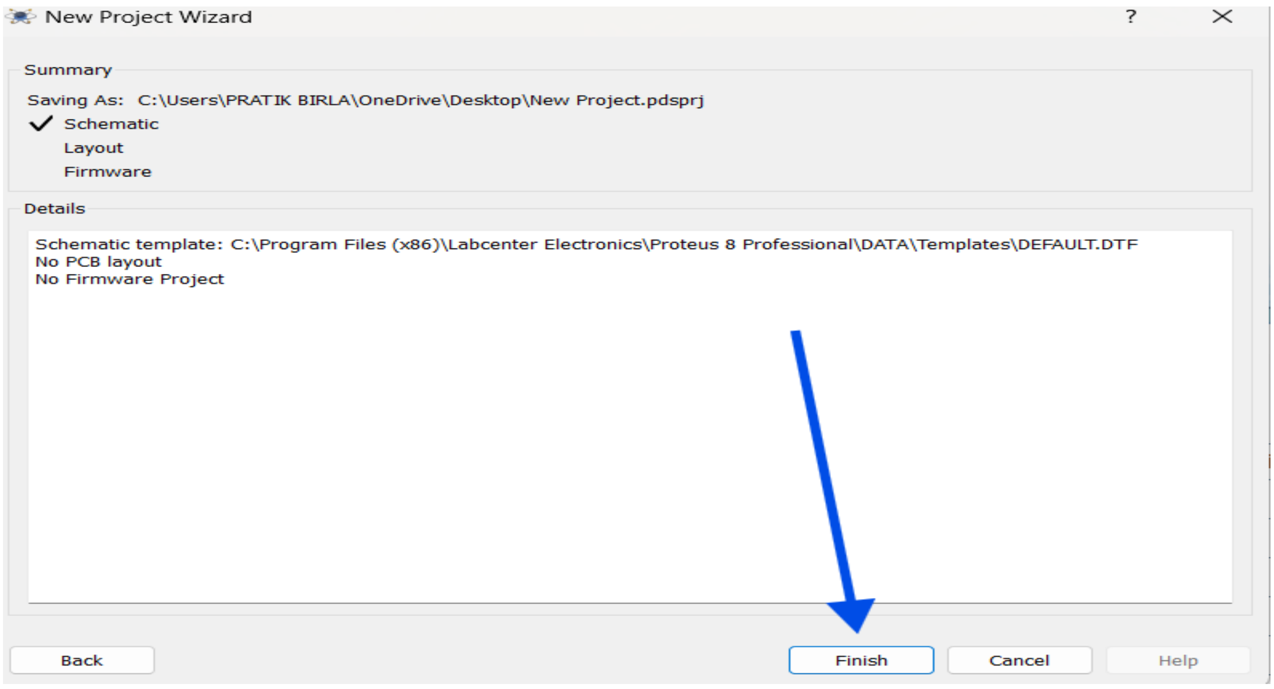


C:\Users\PRATIK BIRLA\OneDrive\Desktop\NewProject\NewProject.pdsprj
C:\Users\PRATIK BIRLA\Downloads\Electronic-Voting-Machine.pdsprj
c:\users\pratik birla\Documents\Atmel Studio\7.0\Example 4.2.a\Example 4.2.a\Debug\New Project.pdsprj
c:\users\pratik birla\Documents\Atmel Studio\7.0\Example 4.2\Example 4.2\Debug\New Project.pdsprj
c:\users\pratik birla\Documents\Atmel Studio\7.0\DemoCode\DemoCode\Debug\New Project.pdsprj



Name New Project.pdsprj
Path C:\Users\PRATIK BIRLA\OneDrive\??

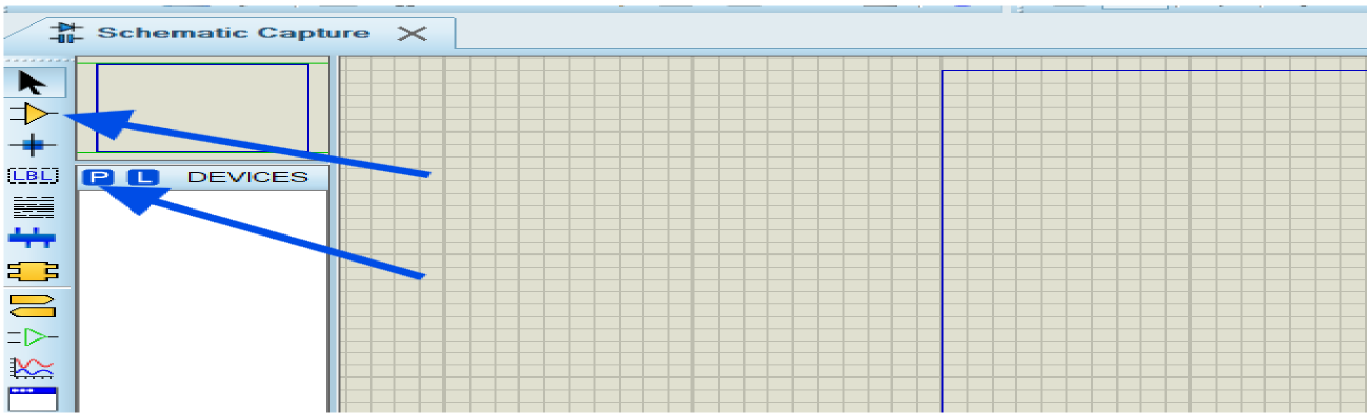
[✓ New Project] From Development Board Blank Project

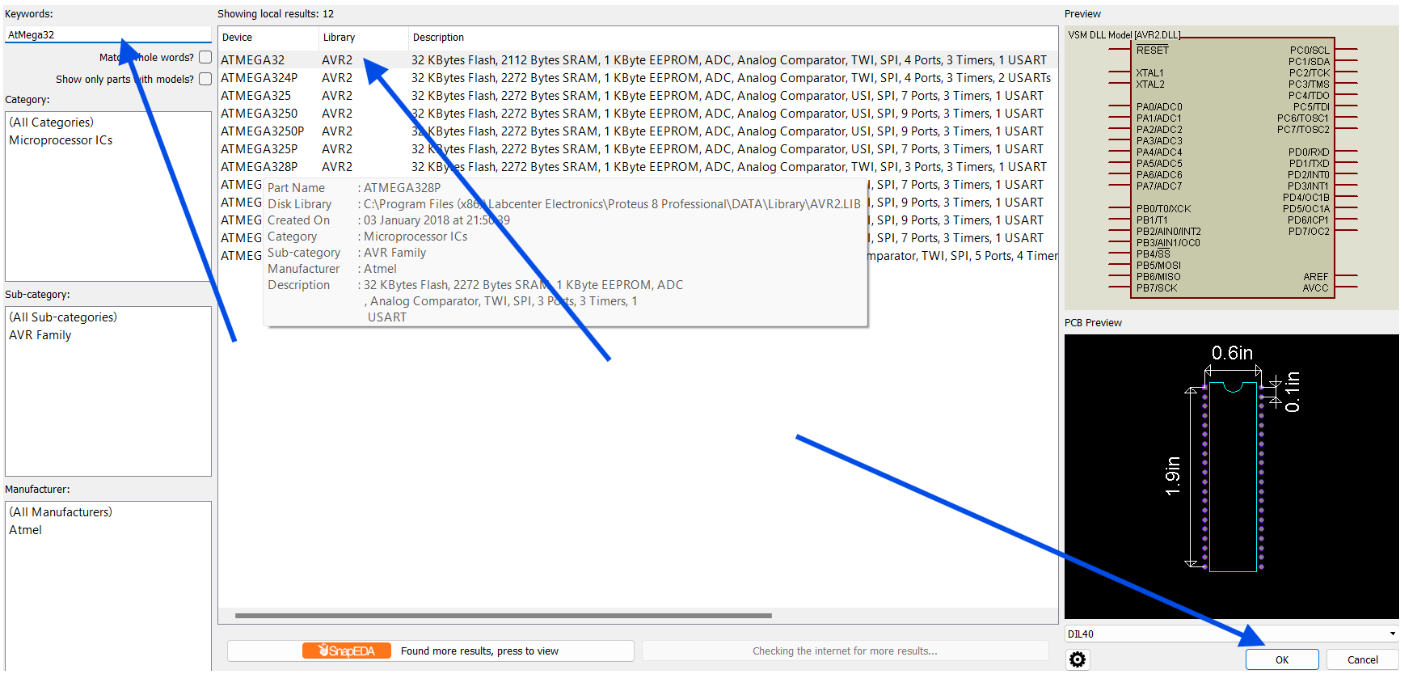


Summary
Saving As: C:\Users\PRATIK BIRLA\OneDrive\Desktop\New Project.pdsprj
✓ Schematic
Layout
Firmware

Details
Schematic template: C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\DATA\Templates\DEFAULT.DTF
No PCB layout
No Firmware Project

- Click on the "p" button shown in the figure and search "atmega32" on the keyboard. Select the device by double clicking or selecting "ok" as shown. (you can use more than one component just search by name)





Keywords:
AtMega32
Match whole words? ☐
Show only parts with models? ☐
Category:
(All Categories)
Microprocessor ICs
Sub-category:
(All Sub-categories)
AVR Family
Manufacturer:
(All Manufacturers)
Atmel

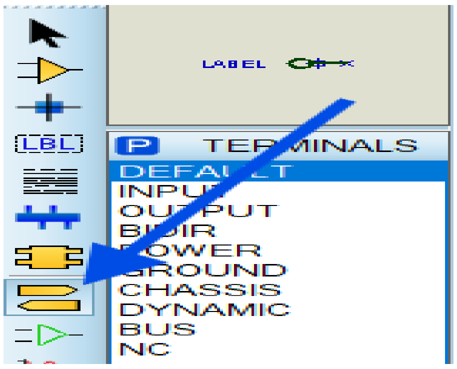
Showing local results: 12

Device	Library	Description
ATMEGA32	AVR2	32 KBytes Flash, 2112 Bytes SRAM, 1 KByte EEPROM, ADC, Analog Comparator, TWI, SPI, 4 Ports, 3 Timers, 1 USART
ATMEGA324P	AVR2	32 KBytes Flash, 2272 Bytes SRAM, 1 KByte EEPROM, ADC, Analog Comparator, TWI, SPI, 4 Ports, 3 Timers, 2 USARTs
ATMEGA325	AVR2	32 KBytes Flash, 2272 Bytes SRAM, 1 KByte EEPROM, ADC, Analog Comparator, USI, SPI, 7 Ports, 3 Timers, 1 USART
ATMEGA3250	AVR2	32 KBytes Flash, 2272 Bytes SRAM, 1 KByte EEPROM, ADC, Analog Comparator, USI, SPI, 9 Ports, 3 Timers, 1 USART
ATMEGA3250P	AVR2	32 KBytes Flash, 2272 Bytes SRAM, 1 KByte EEPROM, ADC, Analog Comparator, USI, SPI, 9 Ports, 3 Timers, 1 USART
ATMEGA325P	AVR2	32 KBytes Flash, 2272 Bytes SRAM, 1 KByte EEPROM, ADC, Analog Comparator, USI, SPI, 7 Ports, 3 Timers, 1 USART
ATMEGA328P	AVR2	32 KBytes Flash, 2272 Bytes SRAM, 1 KByte EEPROM, ADC, Analog Comparator, TWI, SPI, 3 Ports, 3 Timers, 1 USART
ATMEG		, SPI, 7 Ports, 3 Timers, 1 USART
ATMEG		, SPI, 9 Ports, 3 Timers, 1 USART
ATMEG		, SPI, 9 Ports, 3 Timers, 1 USART
ATMEG		, SPI, 7 Ports, 3 Timers, 1 USART
ATMEG		mparator, TWI, SPI, 5 Ports, 4 Timer

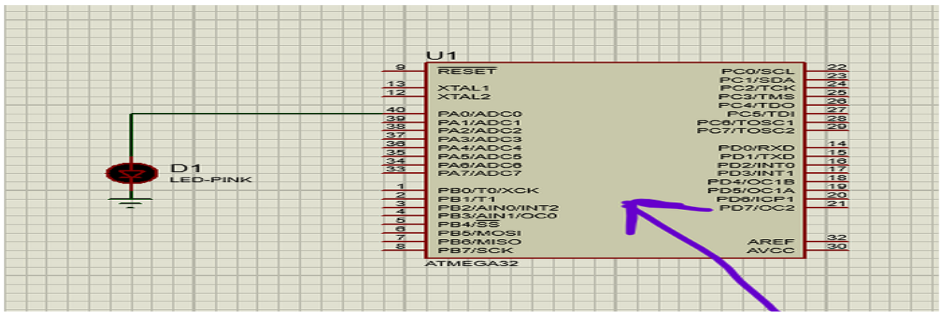
Part Name : ATMEGA328P
Disk Library : C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\DATA\Library\AVR2.LIB
Created On : 03 January 2018 at 21:50:39
Category : Microprocessor ICs
Sub-category : AVR Family
Manufacturer : Atmel
Description : 32 KBytes Flash, 2272 Bytes SRAM, 1 KByte EEPROM, ADC, Analog Comparator, TWI, SPI, 3 Ports, 3 Timers, 1 USART

Preview
VSM DLL Model [AVR2.DLL]
PCB Preview
0.6in
0.1in
1.9in
DIL40
SnapEDA Found more results, press to view
Checking the internet for more results...
OK
Cancel

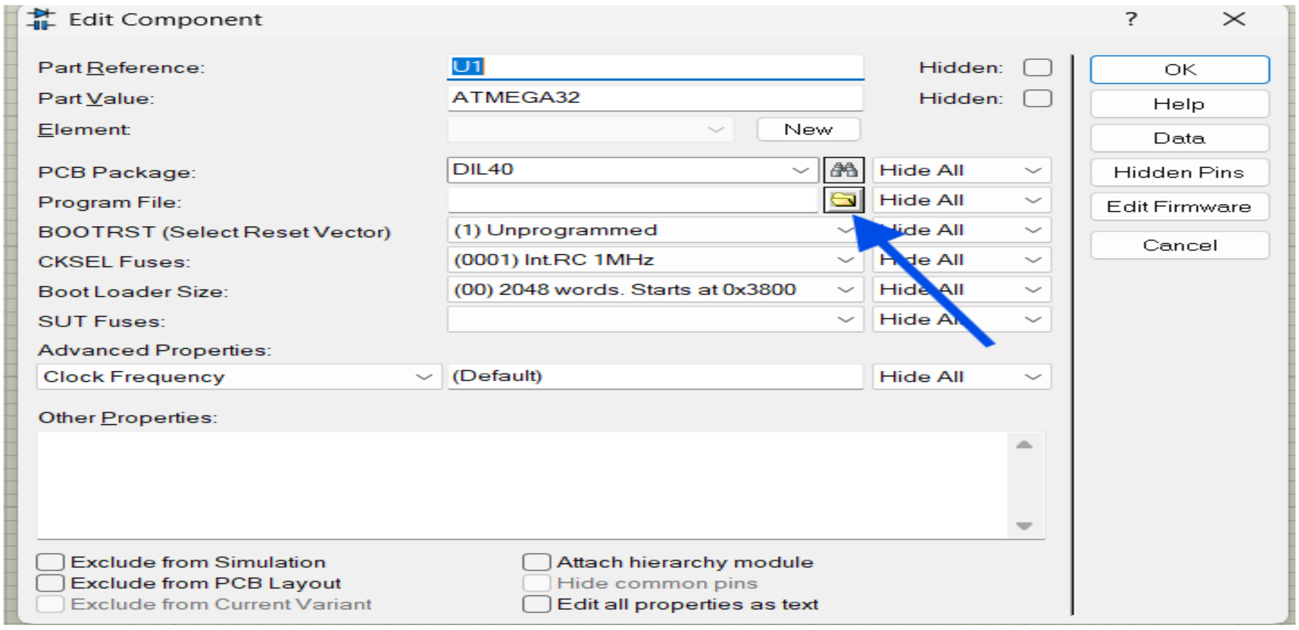
You can get the ground and other imp.components here.



- Complete the circuit as shown and then double click the controller unit to load the hex file created from microchip studio



- Click on the "program file" option also you can change the crystal freq. Using "cksel fuses" option. (do not change other functions).



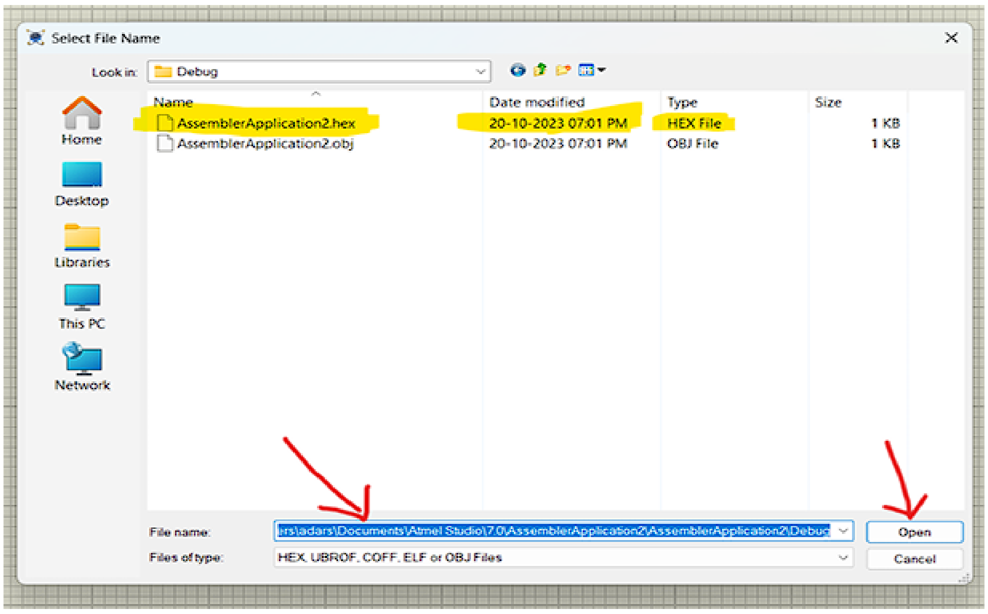
Edit Component

Part Reference:	U1	Hidden:	
Part Value:	ATMEGA32	Hidden:	
Element:		New	
PCB Package:	DIL40	Hide All	
Program File:		Hide All	
BOOTRST (Select Reset Vector)	(1) Unprogrammed	Hide All	
CKSEL Fuses:	(0001) Int.RC 1MHz	Hide All	
Boot Loader Size:	(00) 2048 words. Starts at 0x3800	Hide All	
SUT Fuses:		Hide All	
Advanced Properties:			
Clock Frequency	(Default)	Hide All	

Other Properties:

Exclude from Simulation
Exclude from PCB Layout
Exclude from Current Variant
Attach hierarchy module
Hide common pins
Edit all properties as text

- Paste the path of HEX file or select manual



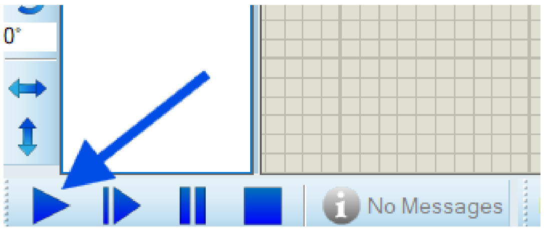
Select File Name

Look in: Debug

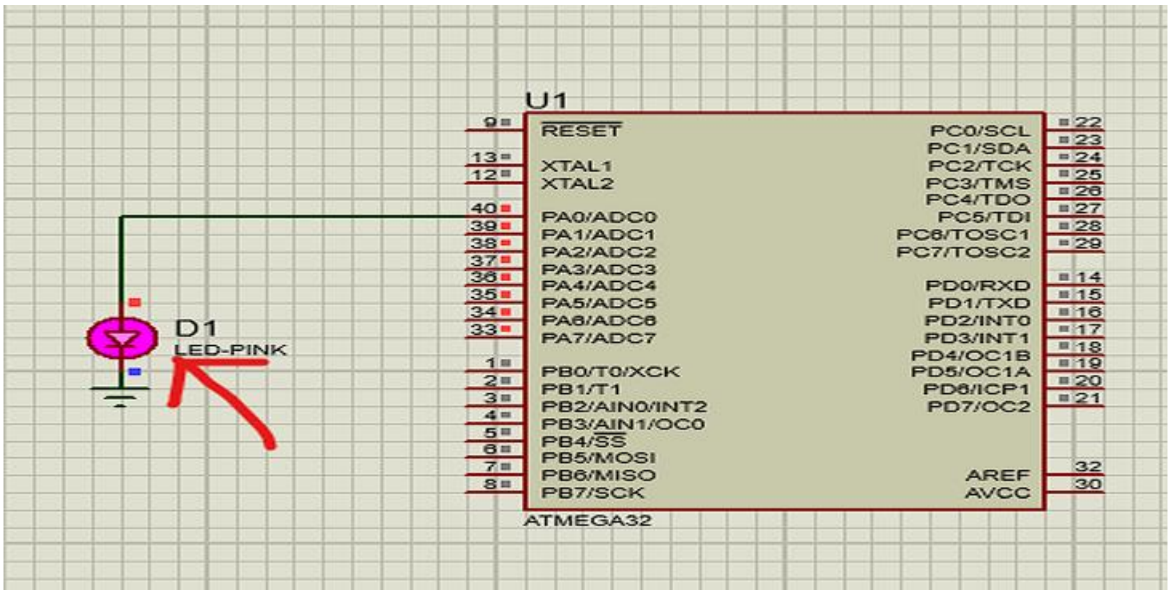
Name	Date modified	Type	Size
AssemblerApplication2.hex	20-10-2023 07:01 PM	HEX File	1 KB
AssemblerApplication2.obj	20-10-2023 07:01 PM	OBJ File	1 KB

File name: ers\adars\Documents\Atmel Studio\7.0\AssemblerApplication2\AssemblerApplication2\Debug
Files of type: HEX, UBROF, COFF, ELF or OBJ Files

- You can Run the simulation from left bottom corner of the window

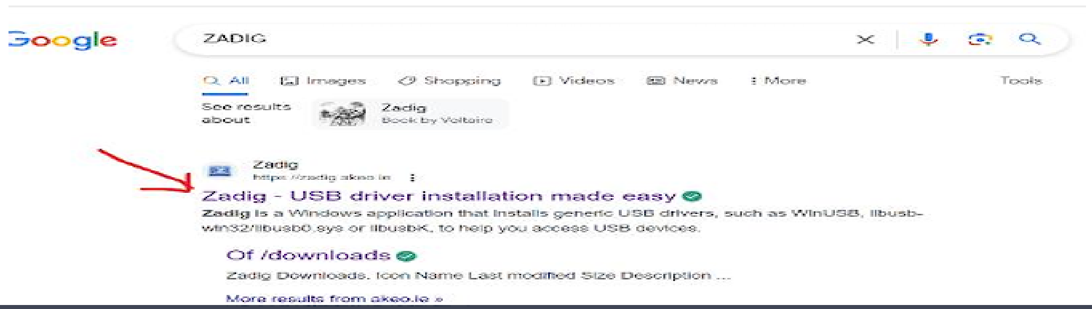


- The program will start burning into you proteus software as shown below.



HOW TO BURN HEX FILE ON HARDWARE ATMEGA32

1.) DOWNLOAD DRIVER INSTALLER NAMED “ZADIG” AND FILE BURNER “PROGISP”.



ZADIG

Zadig
https://zadig.akeo.ie

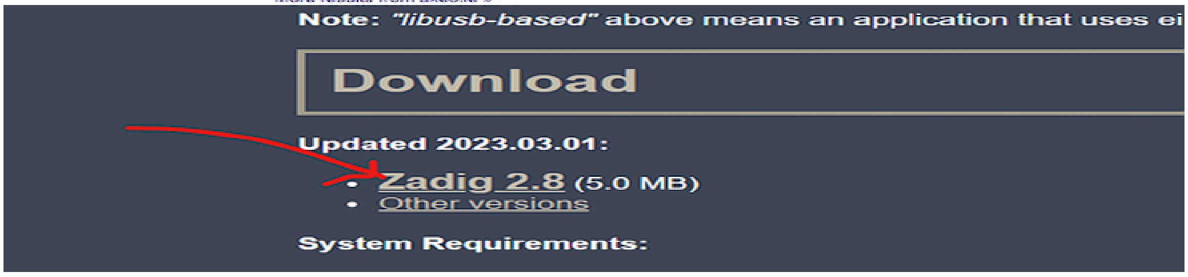
Zadig - USB driver installation made easy

Zadig is a Windows application that installs generic USB drivers, such as WinUSB, libusb-win32/libusb0.sys or libusbK, to help you access USB devices.

Of /downloads

Zadig Downloads. Icon Name Last modified Size Description ...

More results from akeo.ie »



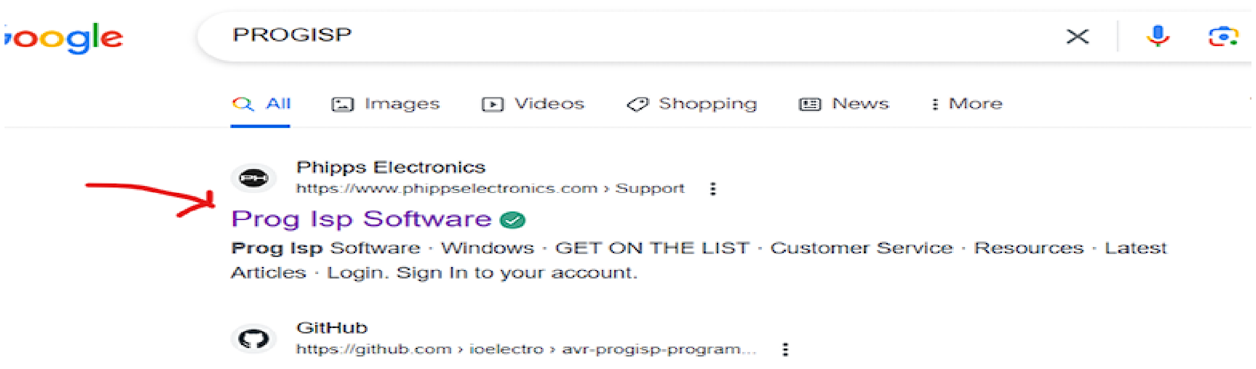
Note: *"libusb-based"* above means an application that uses ei

Download

Updated 2023.03.01:

- Zadig 2.8 (5.0 MB)
- Other versions

System Requirements:



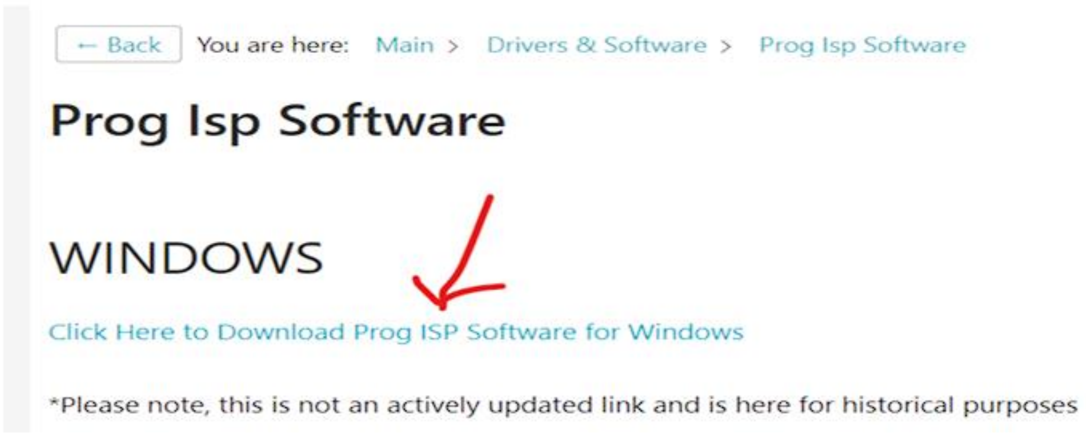
PROGISP

Phipps Electronics
https://www.phippselectronics.com › Support

Prog Isp Software

Prog Isp Software · Windows · GET ON THE LIST · Customer Service · Resources · Latest Articles · Login. Sign In to your account.

GitHub
https://github.com › ioelectro › avr-progisp-program...



← Back   You are here: Main > Drivers & Software > Prog Isp Software

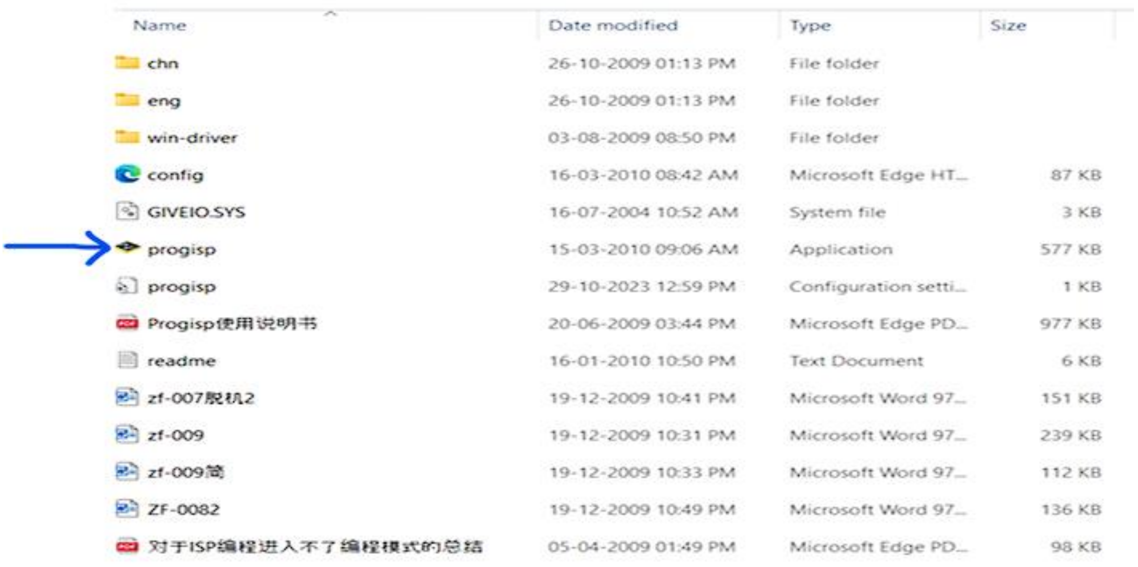
Prog Isp Software

WINDOWS

Click Here to Download Prog ISP Software for Windows

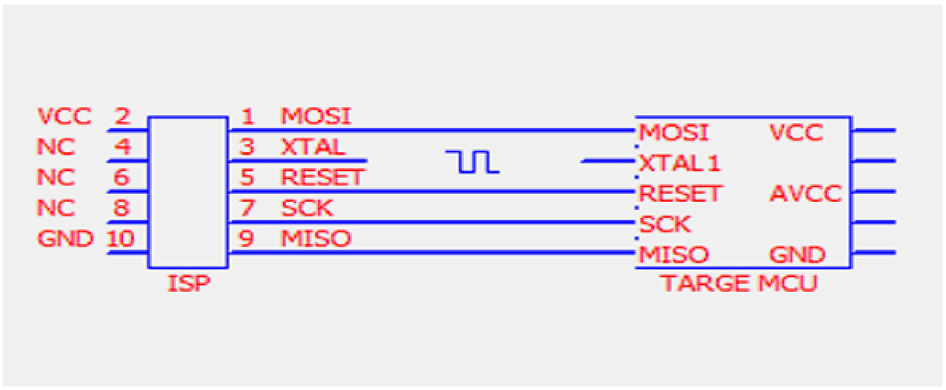
*Please note, this is not an actively updated link and is here for historical purposes

2.) THE PROGISP WILL BE IN ZIPFILE SO EXTRACT THR FILES AND YOU WILL FIND PROGISP NAMED APPLICATION.

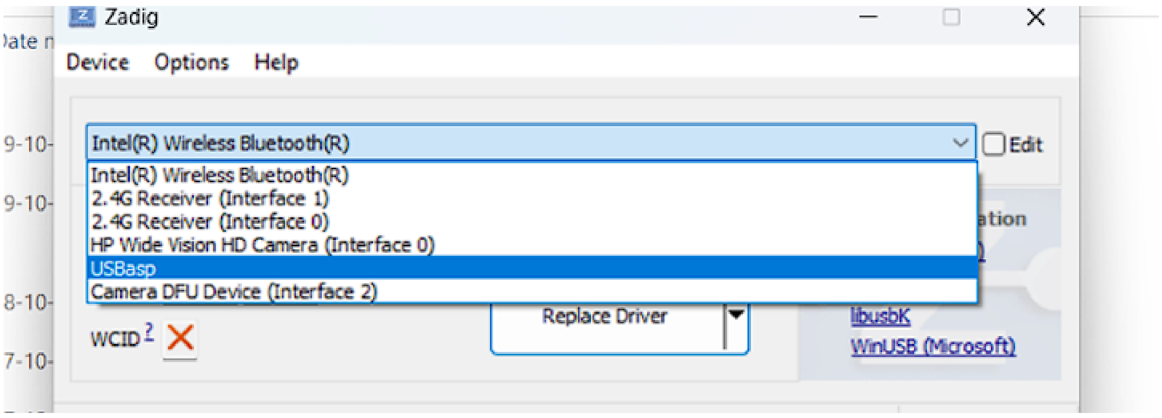


Name	Date modified	Type	Size
chn	26-10-2009 01:13 PM	File folder	
eng	26-10-2009 01:13 PM	File folder	
win-driver	03-08-2009 08:50 PM	File folder	
config	16-03-2010 08:42 AM	Microsoft Edge HT...	87 KB
GIVEIO.SYS	16-07-2004 10:52 AM	System file	3 KB
progisp	15-03-2010 09:06 AM	Application	577 KB
progisp	29-10-2023 12:59 PM	Configuration setti...	1 KB
Progisp使用说明书	20-06-2009 03:44 PM	Microsoft Edge PD...	977 KB
readme	16-01-2010 10:50 PM	Text Document	6 KB
zf-007脱机2	19-12-2009 10:41 PM	Microsoft Word 97...	151 KB
zf-009	19-12-2009 10:31 PM	Microsoft Word 97...	239 KB
zf-009简	19-12-2009 10:33 PM	Microsoft Word 97...	112 KB
ZF-0082	19-12-2009 10:49 PM	Microsoft Word 97...	136 KB
对于ISP编程进入不了编程模式的总结	05-04-2009 01:49 PM	Microsoft Edge PD...	98 KB

3.) NOW CONNECT THE USBASP(DRIVER/BURNER) TO THE ATMEGA32 WITH PROPER PIN CONNECTIONS ,VCC AND GND.(ONLY CONNECT MOSI((PB5),MISO(PB6),SCK(PB7),RST,VCC &GND)

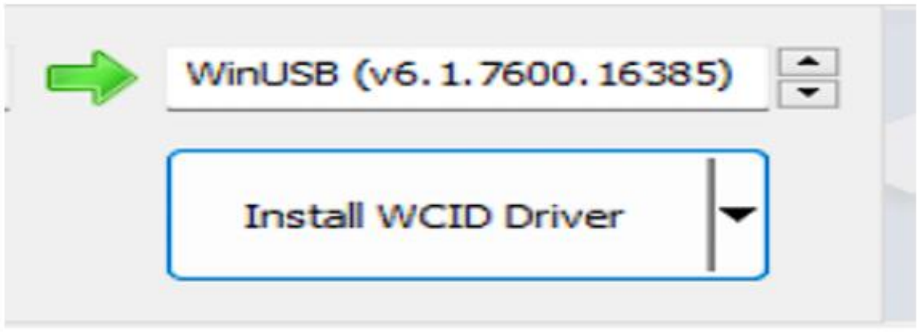


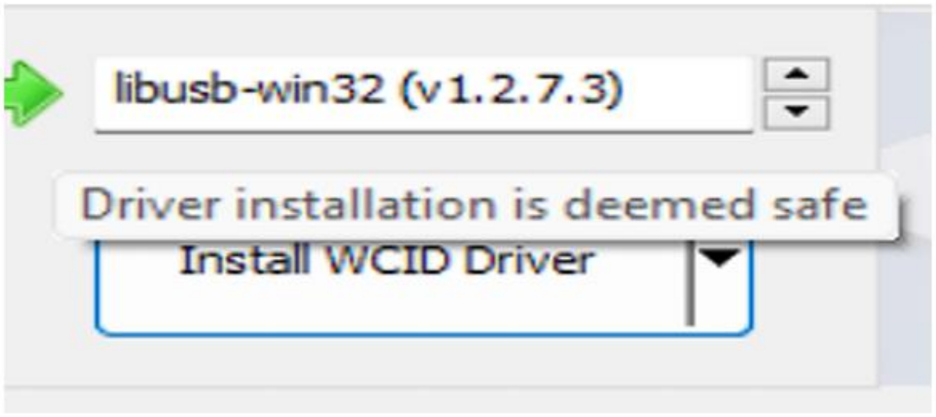
4.) NOW INSERT THE USBASP TO THE PC AND OPEN ZADIG,


Zadig
Device Options Help
Intel(R) Wireless Bluetooth(R)
Edit
Intel(R) Wireless Bluetooth(R)
2.4G Receiver (Interface 1)
2.4G Receiver (Interface 0)
HP Wide Vision HD Camera (Interface 0)
USBasp
Camera DFU Device (Interface 2)
WCID
Replace Driver
libusbK
WinUSB (Microsoft)

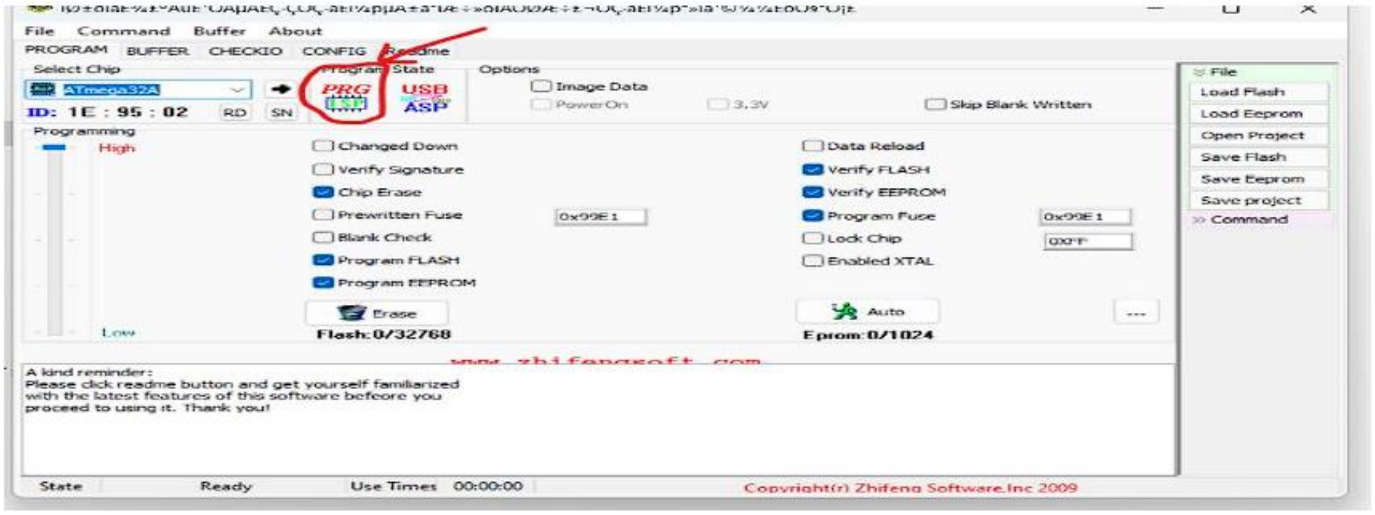
5.) SELECT OPTIONS ->LIST ALL DEVICES->SELECT USBASP FROM THOSE DEVICES.

6.) NOW FIRST INSTALL “WinUSB (v6.1.7600.16385)” DRIVER TO THE PC THEN INSTALL” libusb-win32 (v1.2.7.3)” DRIVER TO THE PC.





7.) NOW OPEN PROGISP AND SELECT THE REQUIRED CHIP.(NOTE THAT PRG-ISP MUST GLOW WHICH INDICATES THAT THE CHIP IS CONNECTED TO PC to BURN hex files).



Select Chip: ATmega32A
ID: 1E : 95 : 02

Options
☐ Image Data
☐ PowerOn
☐ 3.3V
☐ Skip Blank Written

Programming
☐ Changed Down
☐ Verify Signature
☑ Chip Erase
☐ Prewritten Fuse 0x99E1
☐ Blank Check
☑ Program FLASH
☑ Program EEPROM
☐ Data Reload
☑ Verify FLASH
☑ Verify EEPROM
☑ Program Fuse 0x99E1
☐ Lock Chip 0XFF
☐ Enabled XTAL

Flash:0/32768
Eprom:0/1024

www.zhifengsoft.com

A kind reminder:
Please click readme button and get yourself familiarized with the latest features of this software befeore you proceed to using it. Thank you!

State Ready Use Times 00:00:00 Copyright(r) Zhifeng Software.Inc 2009

8.) WHEN PRG-ISP GLOWS YOU CAN CLICK ON “ERASE” TO ERASE THE CHIP,AND CLICK LOAD FLASH TO ADD THE HEX FILE ,THEN CLICK “AUTO” TO FINALLY LOAD THE HEX FILE TO THE CHIP(ATMEGA32).

• SOME EXTRA ERRORS: ”CHIP ERROR”-USE JUMPER(JP3) ON USBASP.

9.) YOU WILL SE “CHIP ERASED SUCCESSFULLY” ON THE MESSAGE TABLE BELOW THE INTERFACE.

NOTE: DO NOT CHANGE or WRITE ANY FUSE BITS ON THE CHIP AND PROGISP SOFTWARE. 10.)MAKE THE REQUIRED HARDWARE CIRCUIT ON THE PCB OR BREADBOARD AND LOAD/BURN THE HEX FILE AND RUN THE PROGRAM.