Experiment-1

Aim: Installation of AVR STUDIO and familiarization with ATMega32 AVR Development Board.

Objectives: After successfully completion of this experiment students will be able to,

- Learn Setup procedure of programming environment of AVR microcontrollers on Windows host machine.
- Get familiarization with ATMega32 AVR Development Board which will be main hardware tool for this laboratory curriculum.

Equipment required:

- Windows7 or later based host computer
- ATMega32 Development board
- USBasp Programmer
- Jumper Wires

Software required:

- WinAVR installation setup
- AVR Studio4 installation setup
- USBasp driver installation setup

Theory:

Introduction of WinAVR:

- WinAVR is necessary for AVR programming.
- This software utility contains all the AVR family mcu specific libraries and tools.
- Latest version of WinAVR can be downloaded from the following website, and it is free-of-charge. Please don't install the latest version as the reviews are not good. It is deleting your PATH environment entirely.
- **WinAVR-20100110** version is used for this setup.

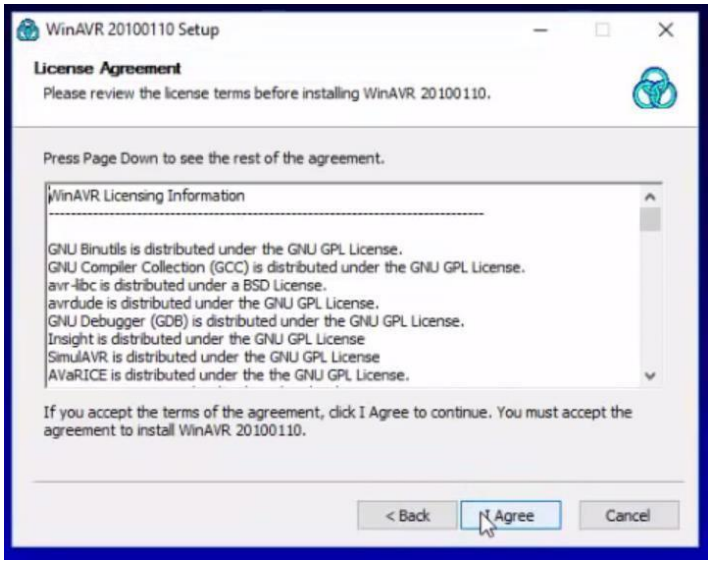
WinAVR Installation Steps:

1. Browse downloaded WinAVR-20100110-install.exe and double click on it to launch installation.

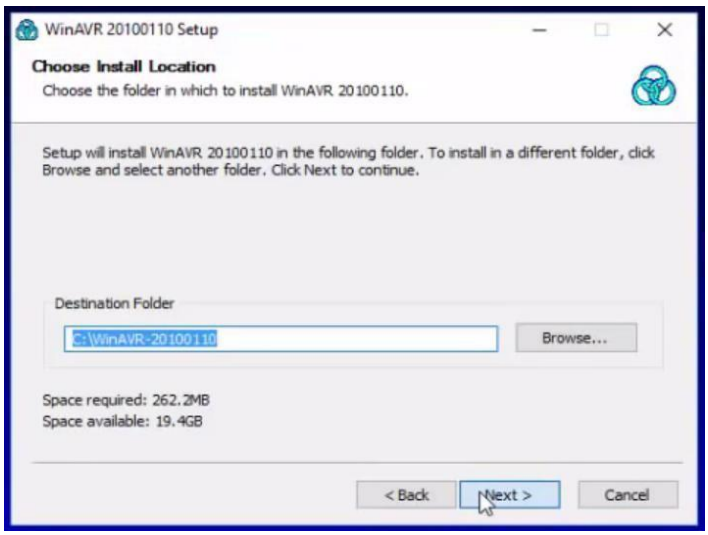
2. Select a language and click OK.

3. The initial installation window is as below.

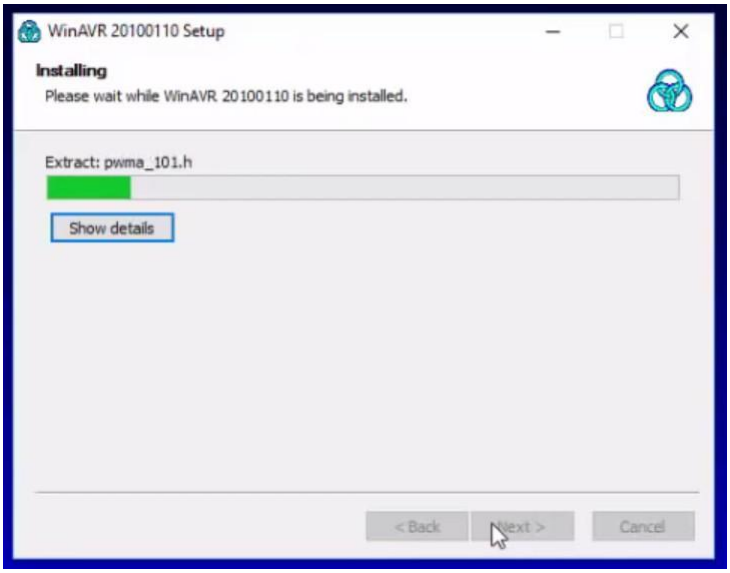
3. The window below is about the license agreement. You must agree to continue the installation, Click **I Agree** button.



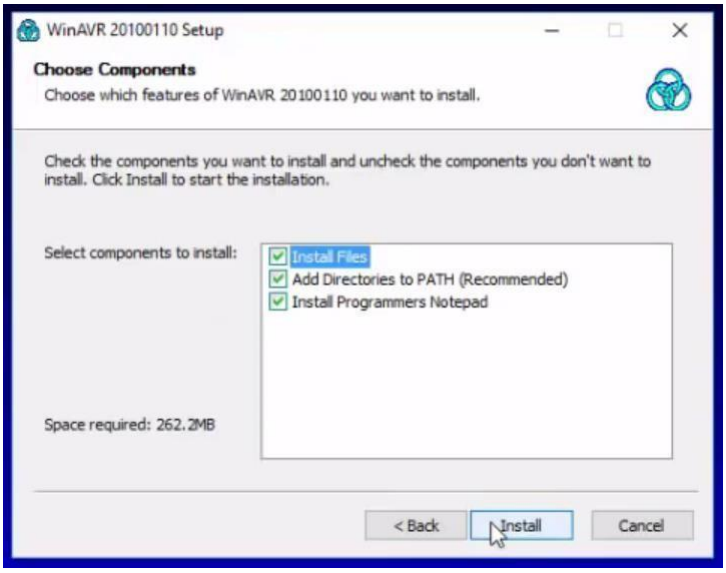
4. Select a folder for the installation or use default installation location, and then click **Next** button.



5. Select programs and environment options. If there are no specific reasons, it is recommended to install all in the list. Click **Install** Button.



6. WinAVR installation is now started.



7. WinAVR installation is completed. Click **Finish** button.


WinAVR 20100110 Setup
Completing the WinAVR 20100110 Setup Wizard
WinAVR 20100110 has been installed on your computer.
Click Finish to close this wizard.
< Back
Finish
Cancel

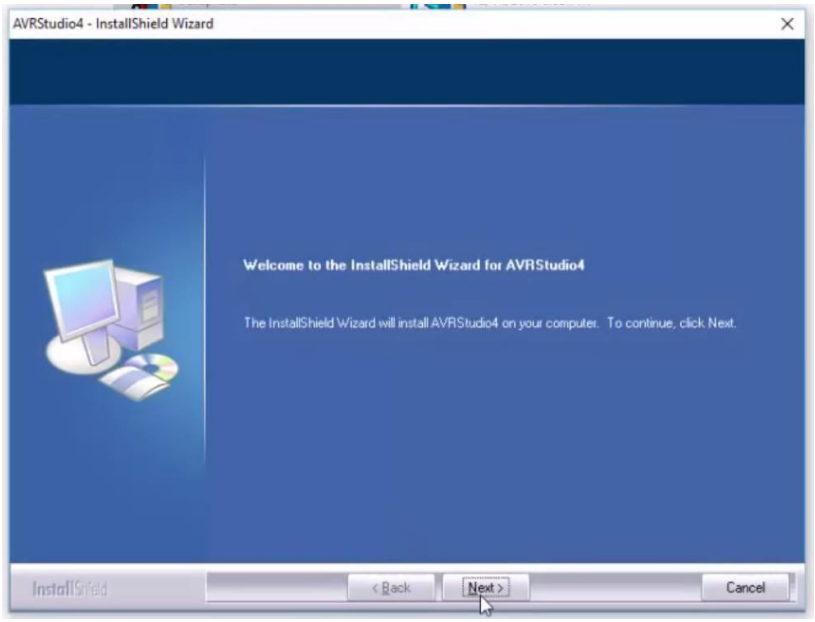
Introduction of AVR Studio4:

- AVR studio4 is an Integrated Development Environment (IDE) by ATMEL for developing applications based on 8-bit AVR microcontroller.
- AVR Studio4 Supports C and Assembly language programming of AVR Microcontrollers.
- AVR Studio4 can be downloaded from ATMEL archive by browsing url mentioned here.
  https://www.microchip.com/en-us/development-tools-tools-and-software/avr-and-sam-downloads-archive
- AVR Studio 4.17 (build 666) is used in this setup

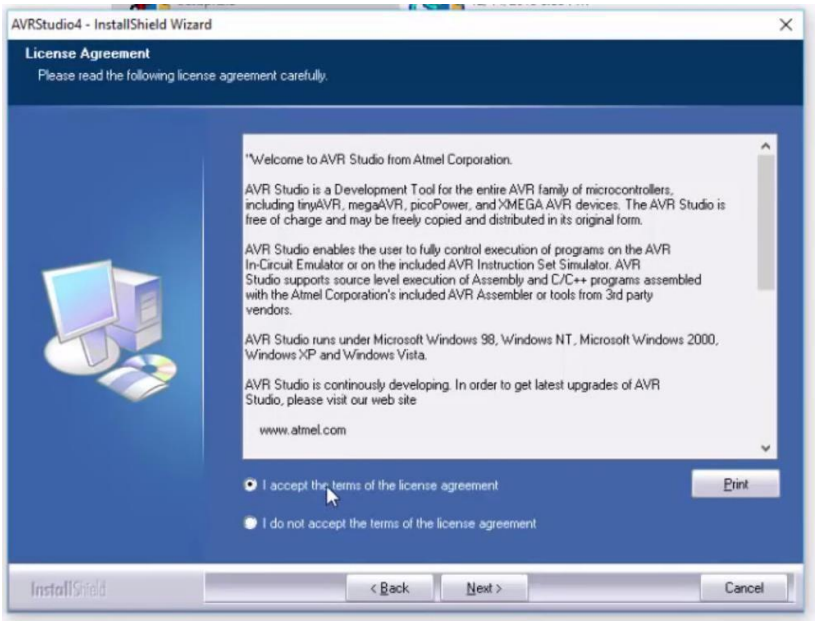
AVR Studio Installation steps:

1. Browse downloaded AvrStudio4Setup.exe and double click on it to launch installation.

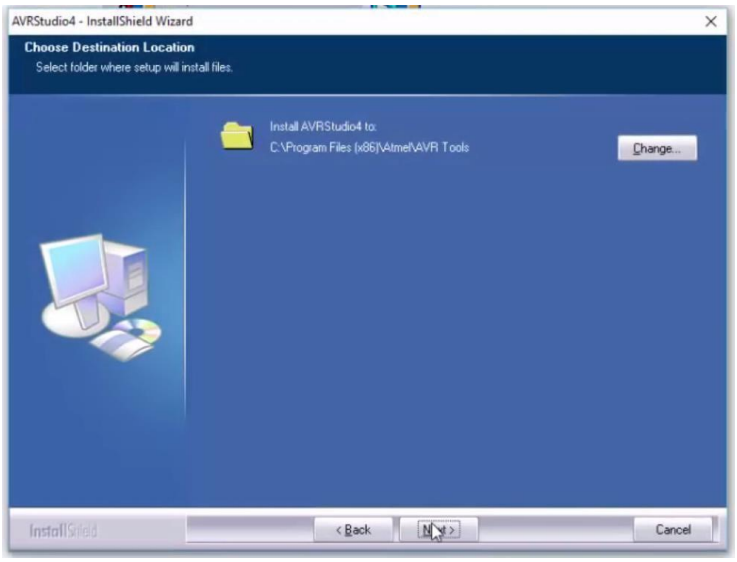
2. Once AVR studio prepared for installation click **Next** button.



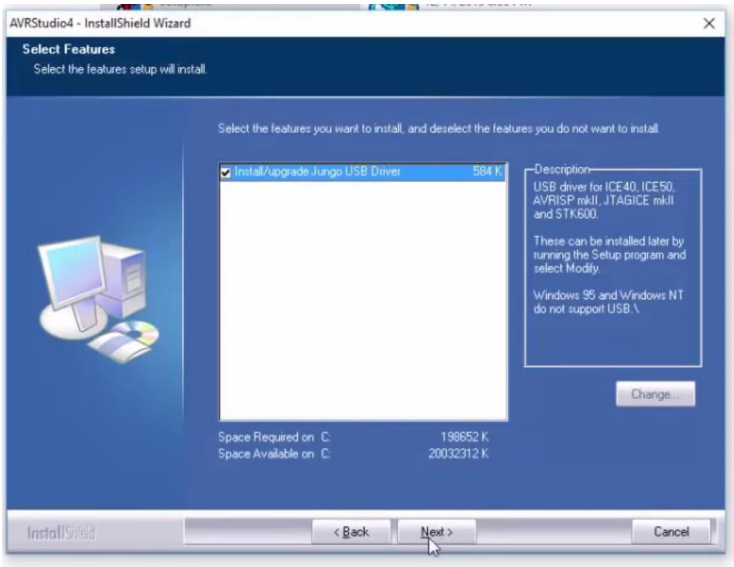
3. Accept license agreement by clicking radio button and click **Next** button.



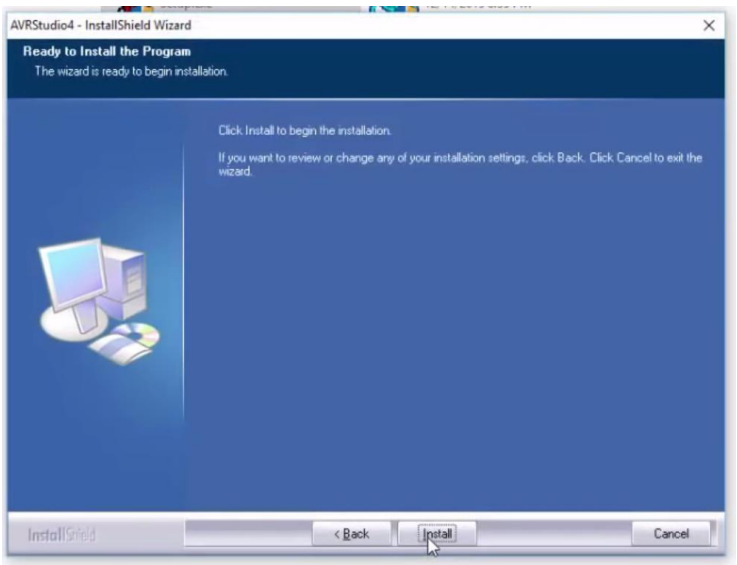
4. Change installation folder or keep default location **Next** button.



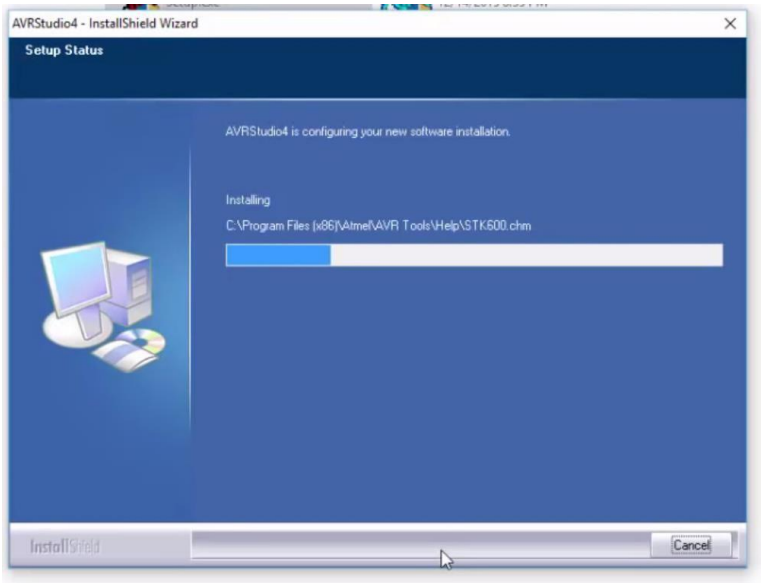
5. Check all driver installation related radio button and click **Next** button.

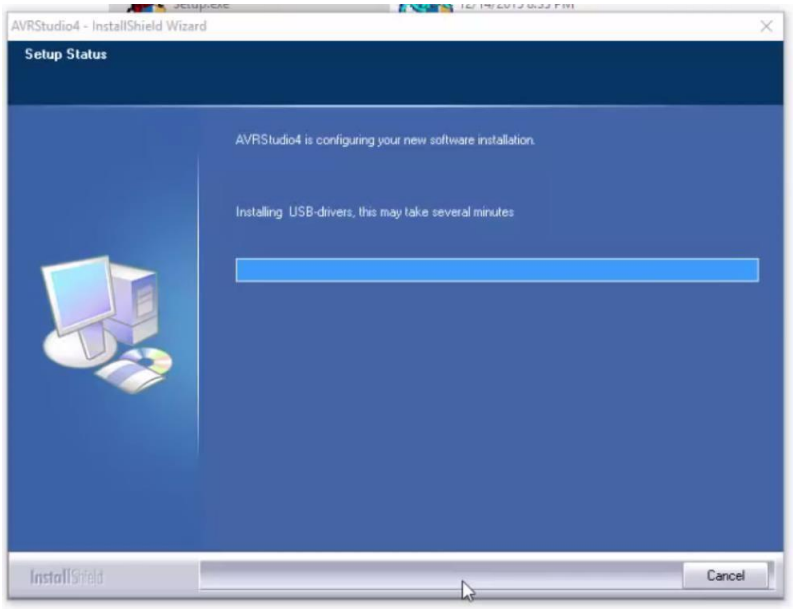


6. Click **Next** button to start installation process.

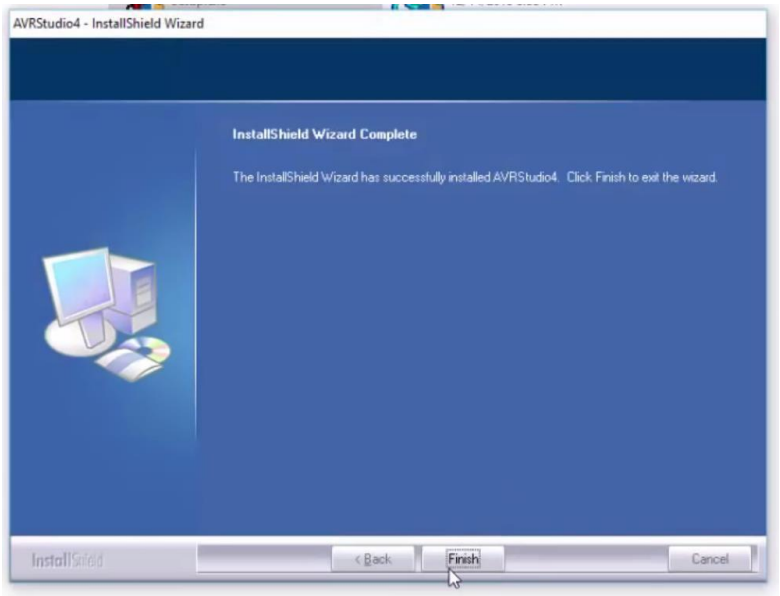


7. AVR Studio installation and driver installation will take few minutes.





8. Once installation of AVR Studio is completed following screen appeared, click **Finish** button to complete installation.



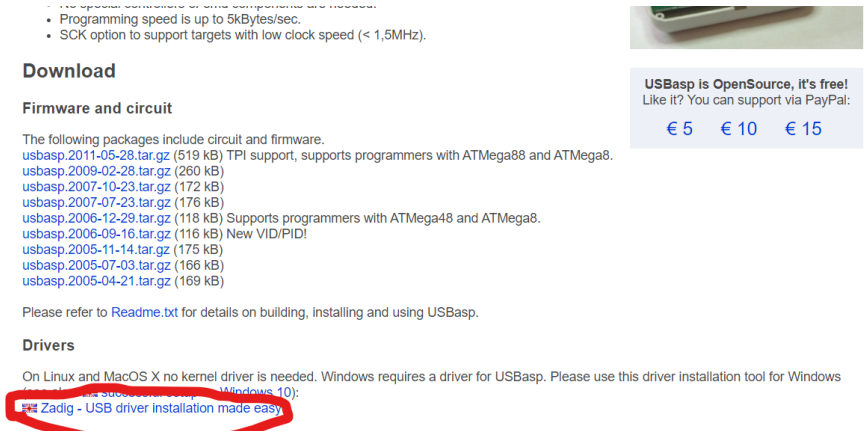
USBasp Driver Installation:

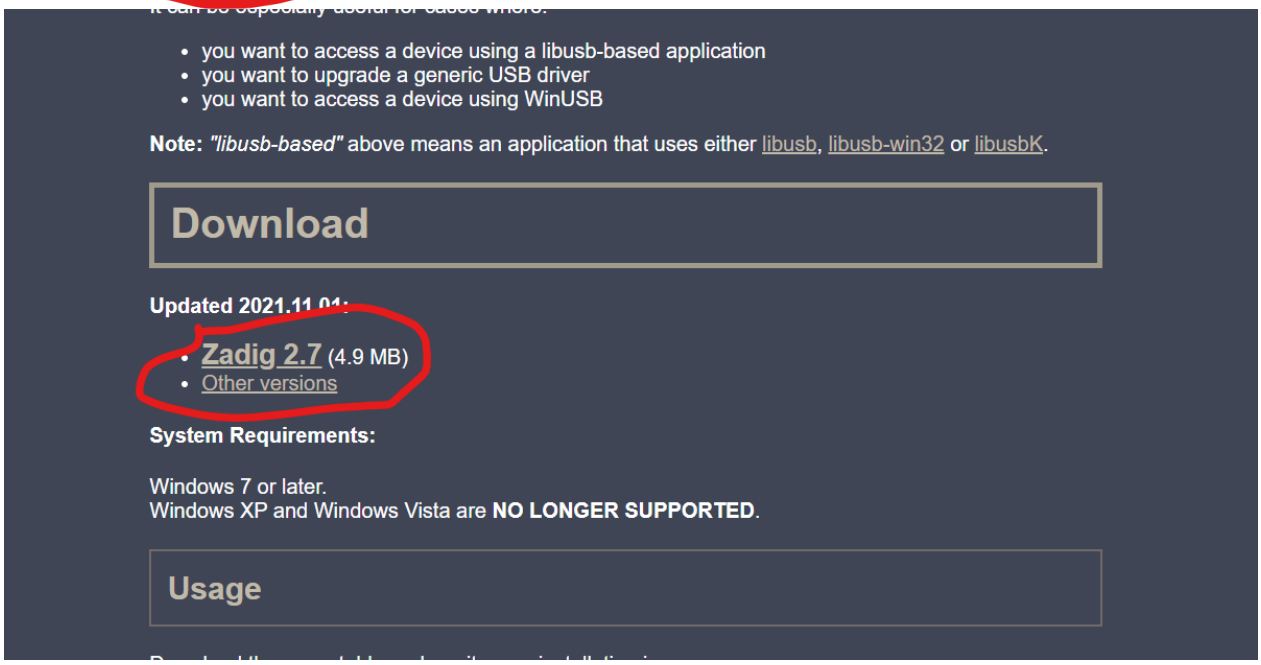
for upload, executable file to microcontroller using USBasp programmer, device driver of the same need to be installed separately.

Here are the steps for USBasp device driver installation,

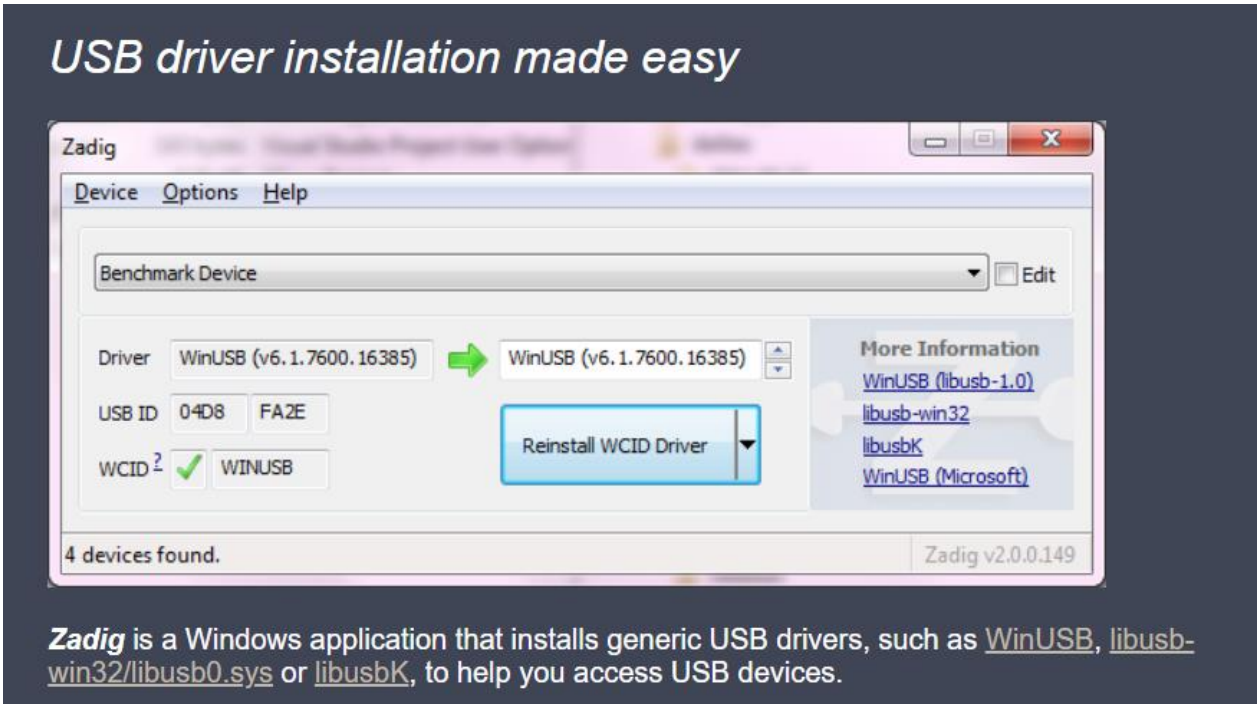
- Download Zadig software for the USBasp's driver installation from the website

https://www.fischl.de/usbasp/.

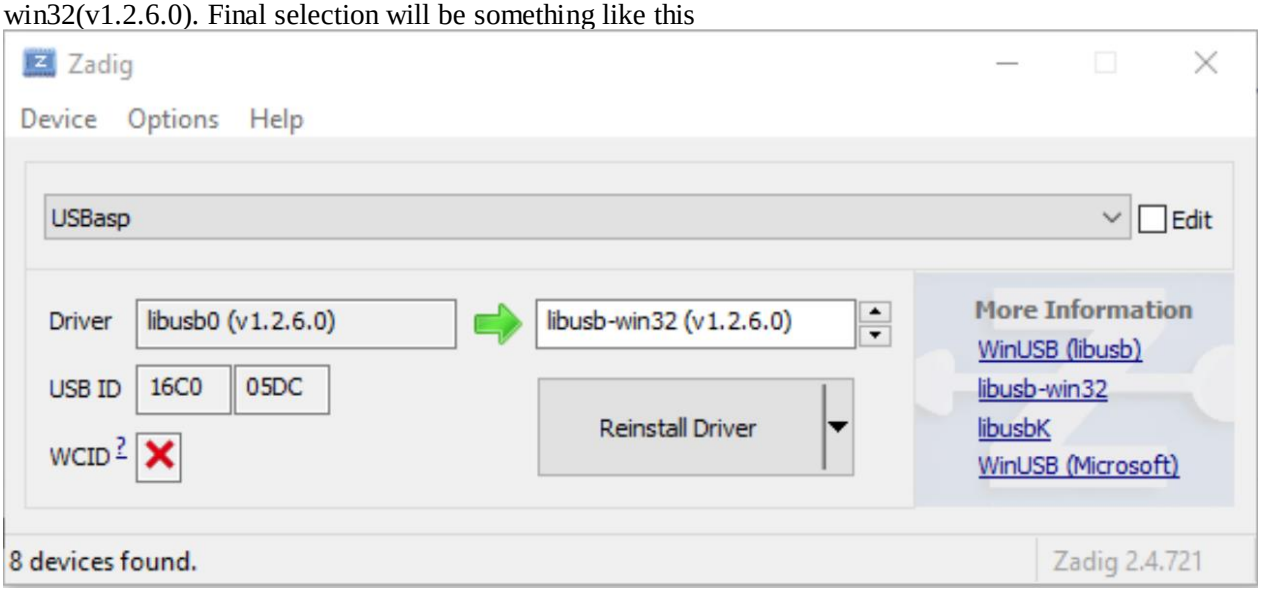


- 

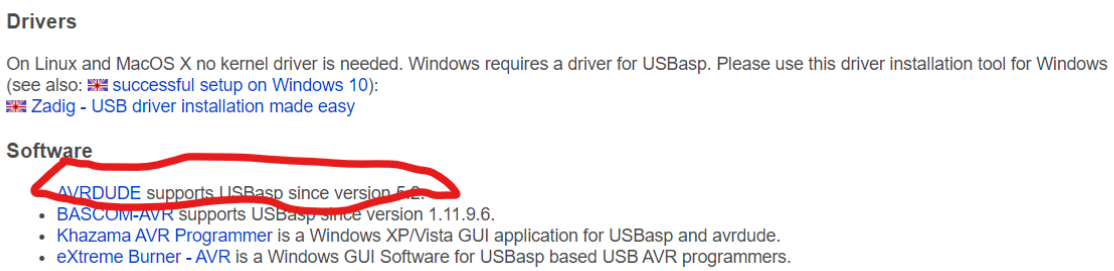
- After installation of Zadig, the opening window will be similar to this



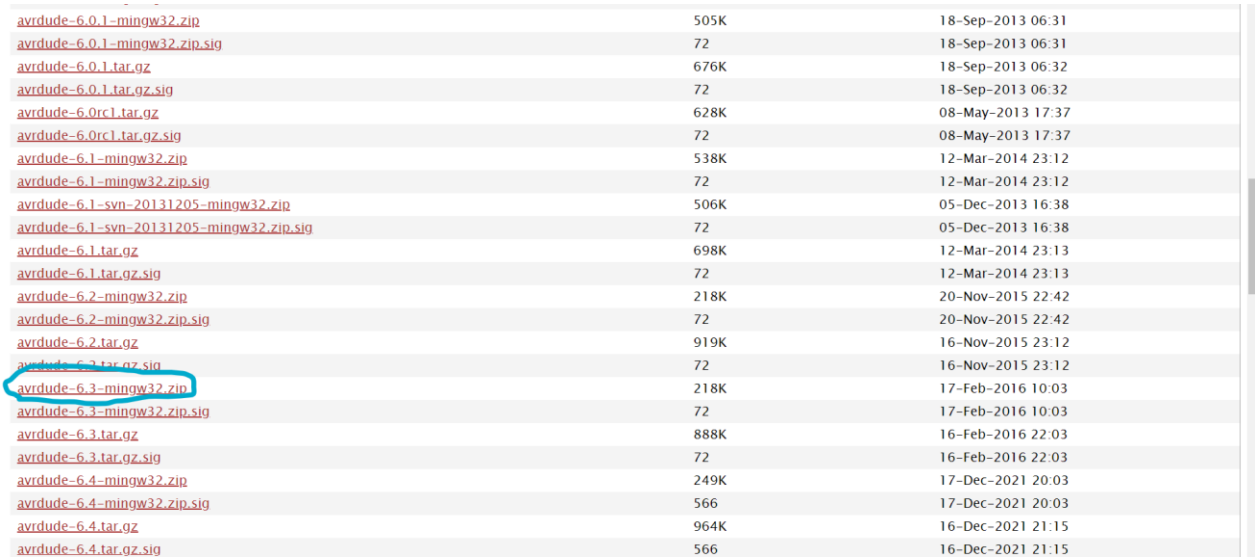
- Now, select USBasp as a device from the drop down menu and select the driver as libusb-win32(v1.2.6.0). Final selection will be something like this



- Now, you have to install AVRDUDE software, from the website https://www.fischl.de/usbasp/.



- From that please select the version “avrdude-6.3-mingw32”

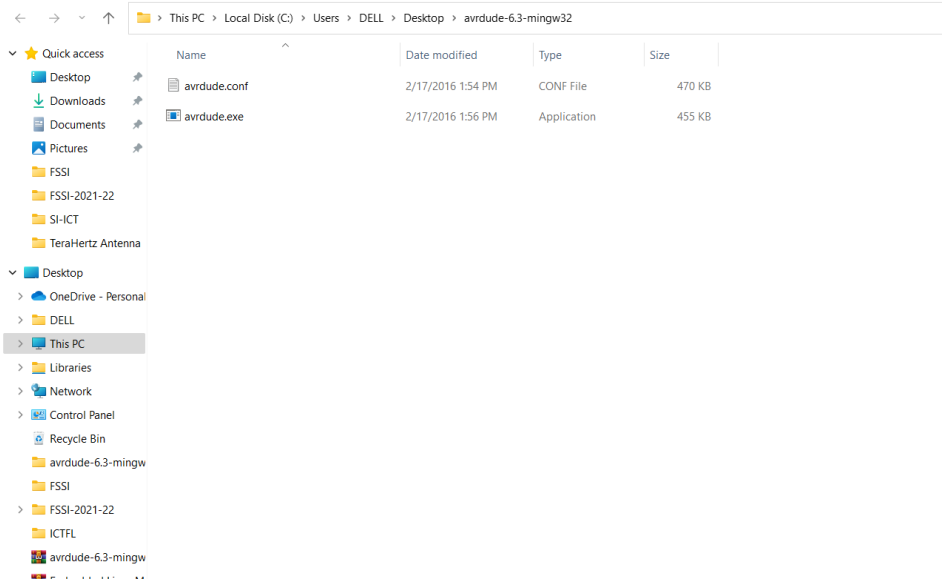


avrdude-6.0.1-mingw32.zip	505K	18-Sep-2013 06:31
avrdude-6.0.1-mingw32.zip.sig	72	18-Sep-2013 06:31
avrdude-6.0.1.tar.gz	676K	18-Sep-2013 06:32
avrdude-6.0.1.tar.gz.sig	72	18-Sep-2013 06:32
avrdude-6.0rc1.tar.gz	628K	08-May-2013 17:37
avrdude-6.0rc1.tar.gz.sig	72	08-May-2013 17:37
avrdude-6.1-mingw32.zip	538K	12-Mar-2014 23:12
avrdude-6.1-mingw32.zip.sig	72	12-Mar-2014 23:12
avrdude-6.1-svn-20131205-mingw32.zip	506K	05-Dec-2013 16:38
avrdude-6.1-svn-20131205-mingw32.zip.sig	72	05-Dec-2013 16:38
avrdude-6.1.tar.gz	698K	12-Mar-2014 23:13
avrdude-6.1.tar.gz.sig	72	12-Mar-2014 23:13
avrdude-6.2-mingw32.zip	218K	20-Nov-2015 22:42
avrdude-6.2-mingw32.zip.sig	72	20-Nov-2015 22:42
avrdude-6.2.tar.gz	919K	16-Nov-2015 23:12
avrdude-6.2.tar.gz.sig	72	16-Nov-2015 23:12
avrdude-6.3-mingw32.zip	218K	17-Feb-2016 10:03
avrdude-6.3-mingw32.zip.sig	72	17-Feb-2016 10:03
avrdude-6.3.tar.gz	888K	16-Feb-2016 22:03
avrdude-6.3.tar.gz.sig	72	16-Feb-2016 22:03
avrdude-6.4-mingw32.zip	249K	17-Dec-2021 20:03
avrdude-6.4-mingw32.zip.sig	566	17-Dec-2021 20:03
avrdude-6.4.tar.gz	964K	16-Dec-2021 21:15
avrdude-6.4.tar.gz.sig	566	16-Dec-2021 21:15

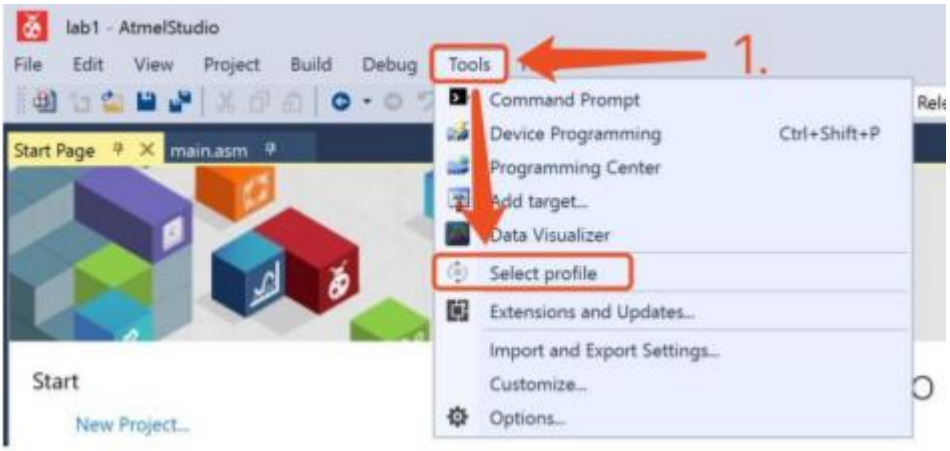
Now, the last step is to setup the external tool in microchip studio to flash the code in the development board.

Setup External Programmer

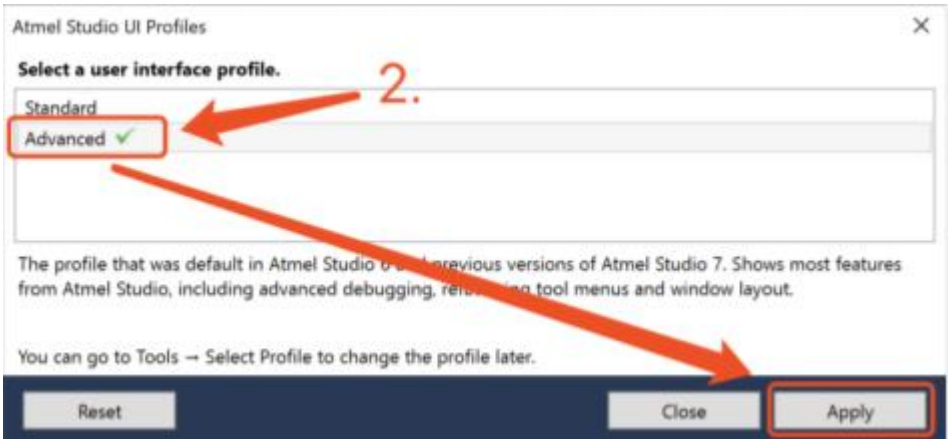
- Unzip the avrdude and save it in your preferred directory. Please make sure that both the files are present.



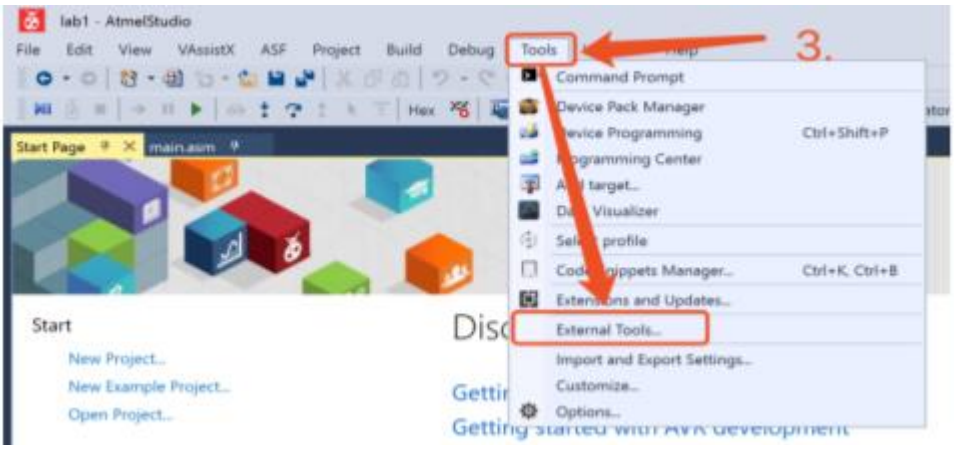
- Enable “Advanced Mode” In order to setup an External programmer, Atmel Studio needs to be in 'Advanced' profile. This can be found under the Tools menu. Go to > Tools > Select Profile.



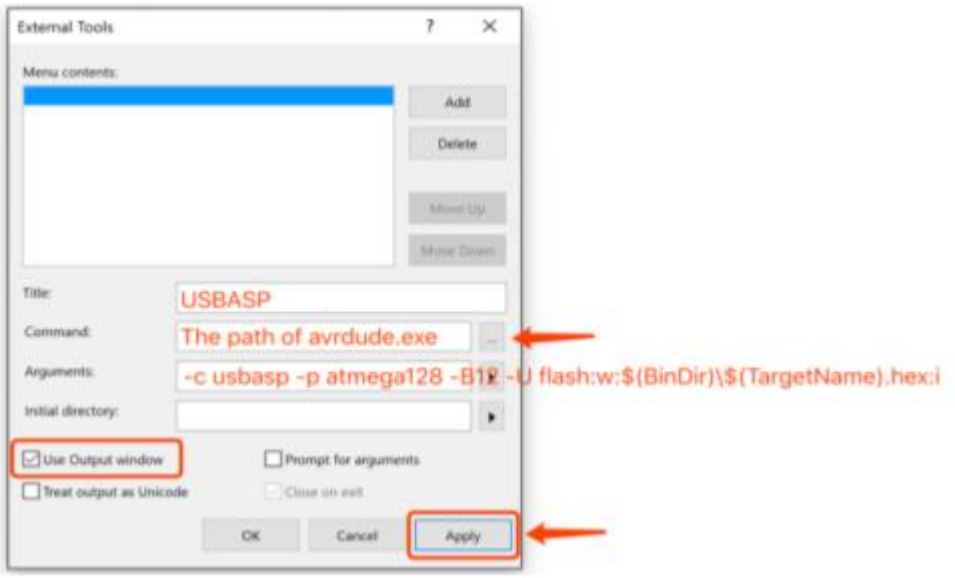
Select “Advanced” and click “Apply”



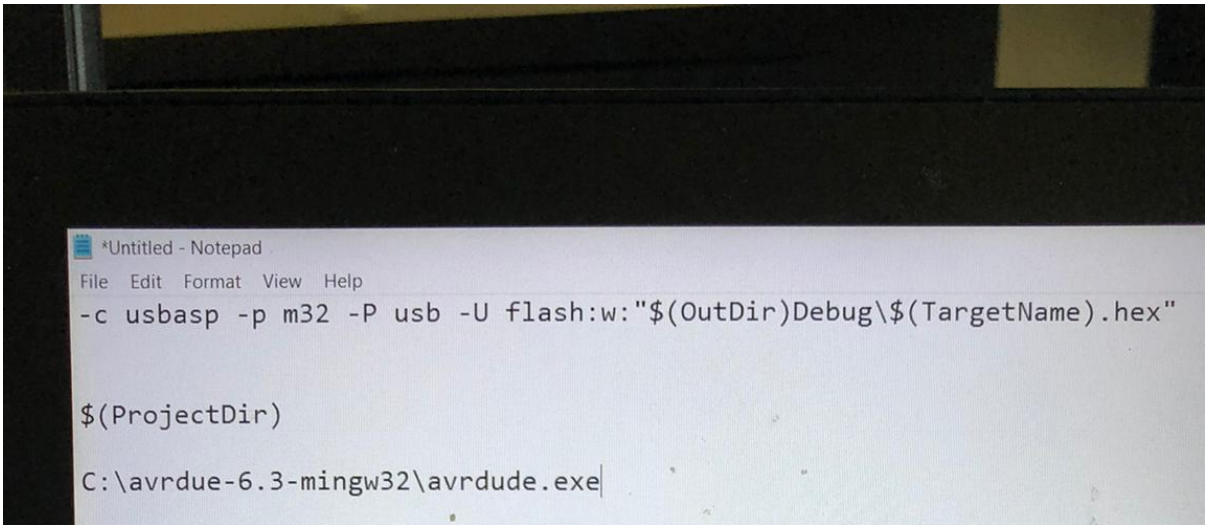
- Add “External Tools” Go to > Tools > External Tools.



Add new external tool



Please give Arguments, Initial directory, and command as follows

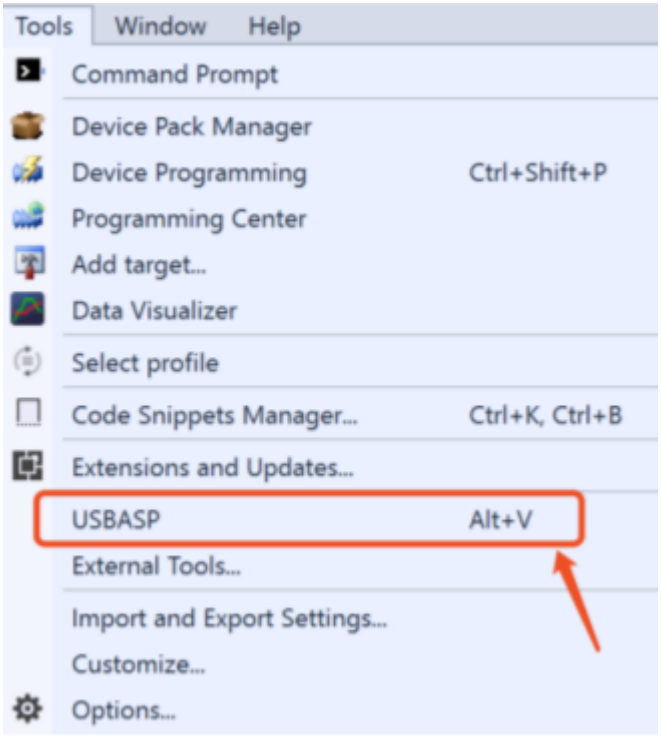


Here in the command section, please give the command where you have actually unzipped your AVRDUDE software.
Check ☐ "Use Output window", Uncheck ☐ "Treat output as Unicode" and ☐ "Prompt for arguments". Then, click "Apply" to save the setting.

- Flash hex file with external programmer

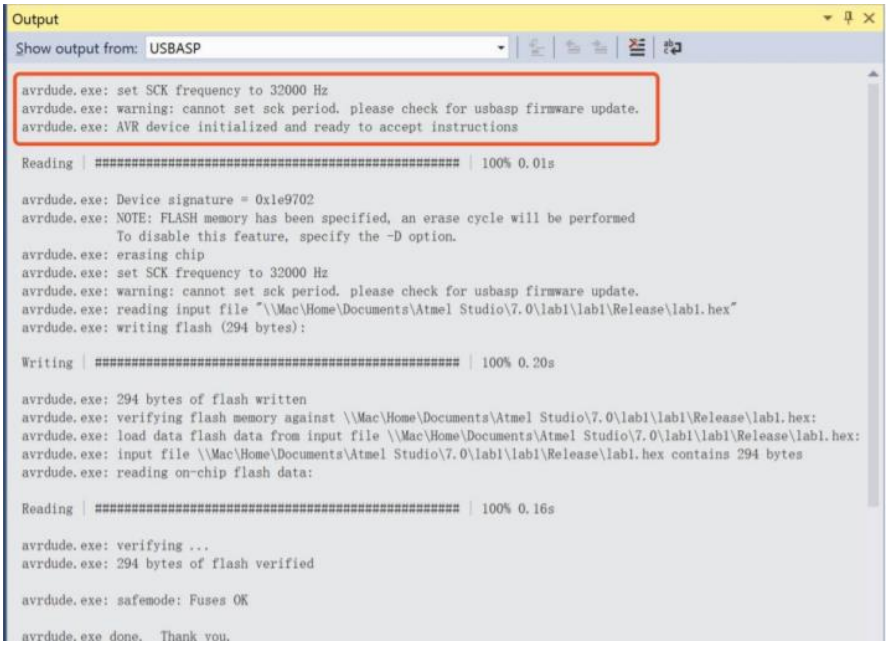
First, make sure to build your project.

Go to > Tool > USBASP, or use the shortcut of your choice.



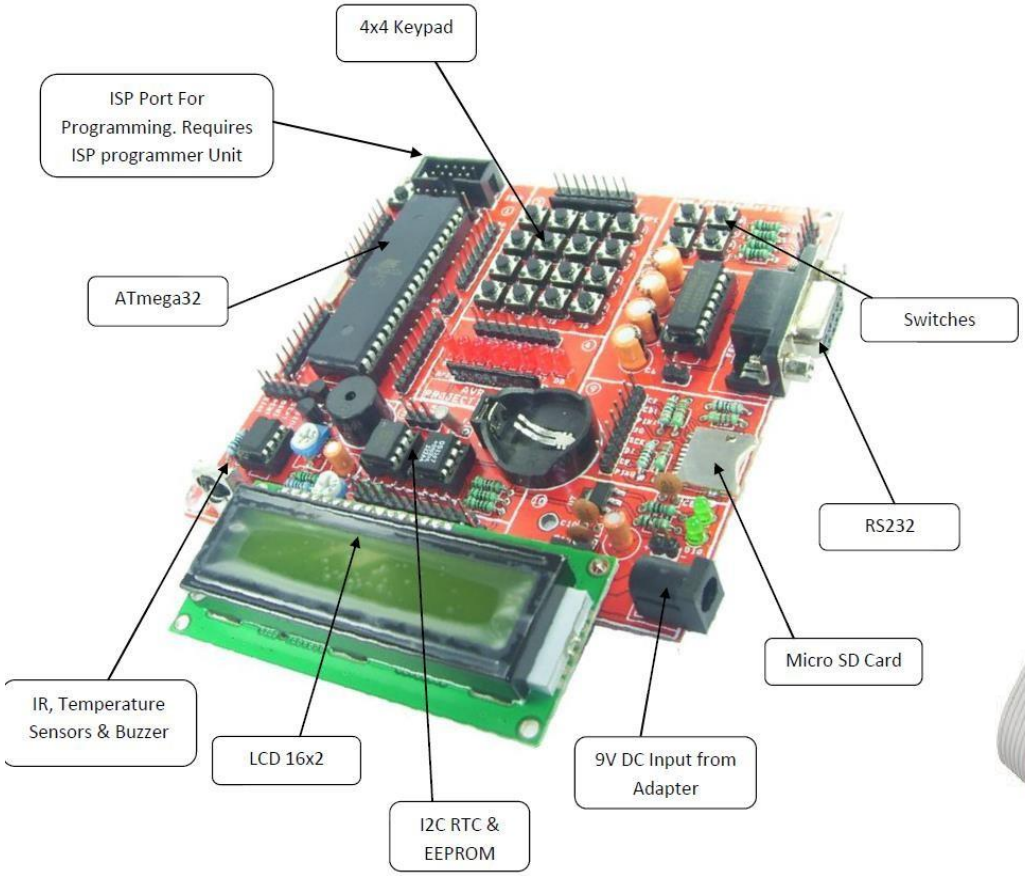
DONE!

If you see the following message, you have the hex file flashed on the AVR board. You may see a warning in the output window, but it can be ignored.



Introduction of ATmega32 development board:

ATMega32 development board having broad range of on board peripherals. This board is either powered by 9V DC adapter or 5V DC supply available from USBasp programmer. All interfacing peripherals are sharing power supply connection only, while connecting them to ATMega32 F-F wires are required. For know more about schematic and specifications of this board, please refer Appendix-A of this Lab manual. (already shared in Canvas)



Background Information

Overview of the Processor

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
- High Endurance Non-volatile Memory segments
  - 32K Bytes of In-System Self-programmable Flash program memory
  - 512 Bytes EEPROM
  - 1K Byte Internal SRAM
- JTAG (IEEE std. 1149.1 Compliant) Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
- I/O and Packages
  - 32 Programmable I/O Lines, 40 pin DIP package
- Operating Voltages
  - 4.5 - 5.5V for ATmega16
- Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L
  - Active: 1.1 mA, Idle Mode: 0.35 mA, Power-down Mode: < 1 μA

Experiment-1 Post Lab Exercise

Student Name: Aryan Langhanoja
Enrollment No: 92200133030

Answer the following Questions:

1) **Which pins of ATMega32 are used by USBasp programmers to upload executable files to ATMega32 flash?**
   - USBasp programmers typically use the SPI (Serial Peripheral Interface) pins of the AVR microcontroller to program the device. For the ATmega32, the SPI pins are as follows:
     - MOSI (Master Out Slave In) - This pin is used for data output from the programmer to the microcontroller.
     - MISO (Master In Slave Out) - This pin is used for data input from the microcontroller to the programmer.
     - SCK (Serial Clock) - This pin provides the clock signal for synchronizing data transfer.
     - RESET - This pin is used to reset the microcontroller during programming.
   - The pins on the ATmega32 corresponding to these SPI pins are:
     - MOSI: PB5 (Pin 19)
     - MISO: PB6 (Pin 20)
     - SCK: PB7 (Pin 21)
     - RESET: PC6 (Pin 1)

2) **List onboard peripherals available on the ATMega32 development board.**
   - Peripherals available on the ATMega32 Development board :-

1) LEDs
2) Push Buttons
3) LCD (Liquid Crystal Display)
4) Potentiometers (Variable Resistors)
5) Serial Interface (UART)
6) External EEPROM
7) Headers (for I/O access)
8) Crystal Oscillator
9) Voltage Regulator
10) ISP (In-System Programming) Header

3) **What are the possible clock frequencies for the Internal RC oscillator of ATMega32? What is the frequency of the external crystal oscillator on the ATMega32 development board?**
   - The ATmega32 microcontroller offers several options for its internal RC oscillator frequency. The possible clock frequencies for the internal RC oscillator of the ATmega32 are typically:

1) 1 MHz
2) 2 MHz
3) 4 MHz
4) 8 MHz

   - A commonly used frequency for the external crystal oscillator on ATmega32 development boards is 16 MHz

4) **What is the use of a coin cell on the ATMega32 development board?**
   - The coin cell battery on an ATmega32 development board is typically used to provide backup power for maintaining the real-time clock (RTC) or other low-power features when the main power source is disconnected.

5) **Identify the connection type of LEDs (Active HIGH or Active LOW) and Switches (Normally ON or Normally OFF).**
   - LEDs are typically connected in an active HIGH configuration, meaning they illuminate when the corresponding output pin is set to HIGH. Switches are often connected in a normally OFF configuration, meaning they are in the open state (not pressed) by default.

6) **What is the importance/function of USBASP programmer?**
   - The USBASP programmer is used to program AVR microcontrollers, including the ATMega32, by connecting it to a computer via USB. It allows for uploading firmware and code to the microcontroller, facilitating the development and testing of embedded systems.

7) **When you build the solution of your project, which kind of file will be created?**
   - .HEX file will be created.

8) **List all the serial communication protocols supported by the ATMEGA 32 development board.**
   - ATmega32 development boards typically support multiple serial communication protocols, including:
     1) UART (Universal Asynchronous Receiver-Transmitter)
     2) SPI (Serial Peripheral Interface)
     3) I2C (Inter-Integrated Circuit, also known as TWI - Two-Wire Interface)

9) **What is the name of the IDE supported by the AVR development board?**
   - Microchip Studio

10) **Give some examples of input and output devices that can be interfaced with the AVR development board.**
   - Input Devices:

     1) Push Buttons
     2) Switches
     3) Potentiometers (Variable Resistors)
     4) Keypads
     5) Rotary Encoders
     6) Sensors (e.g., Temperature, Light, Distance)
     7) RFID Readers
     8) GPS Modules

- Output Devices:

1) LEDs (Light Emitting Diodes)
2) LCD (Liquid Crystal Display)
3) Seven-Segment Displays
4) Servo Motors
5) Stepper Motors
6) DC Motors
7) Buzzers
8) Relays
9) Solenoids
10) LED Matrix Displays