Mechatronics Engineering

In-System Programming of AVR Microcontrollers

GUC German University in Cairo

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Contents

- AVR Microcontroller Programming Methods
  - Self-programming
  - In-System Programming (ISP)
    - ❖ Hardware Connection
    - ❖ Software Setup
    - ❖ Fuse Bits Setup
    - ❖ Create Project
    - ❖ Download Program

Contents

- **AVR Microcontroller Programming Methods**
  - Self-programming
  - In-System Programming (ISP)
    - Hardware Connection
    - Software Setup
    - Fuse Bits Setup
    - Create Project
    - Download Program

AVR Microcontroller Programming Methods

- Many AVR Microcontrollers support self-programming.
- For these devices, there are two main options for programming a microcontroller:

1. **Self-programming** through serial communication (as in Arduino boards).

1. **In-System Programming** (**ISP**) (requires external programmer board):

   - ❖ Through Serial Peripheral Interface (**SPI**) communication protocol.
   - ❖ Through parallel programming interface.

Contents

- **AVR Microcontroller Programming Methods**
  - **Self-programming**
  - In-System Programming (ISP)
    - Hardware Connection
    - Software Setup
    - Fuse Bits Setup
    - Create Project
    - Download Program

Self-programming

- Program memory is divided into two parts for MCU with self-programming capability:

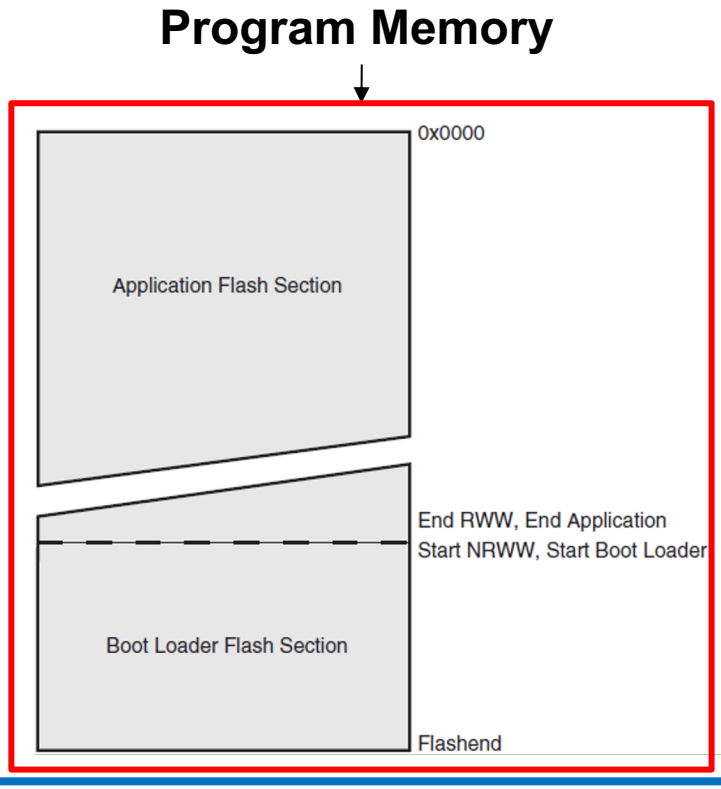
1. Application Flash Section:

   ❖ Stores application code.

1. Boot Loader Section:

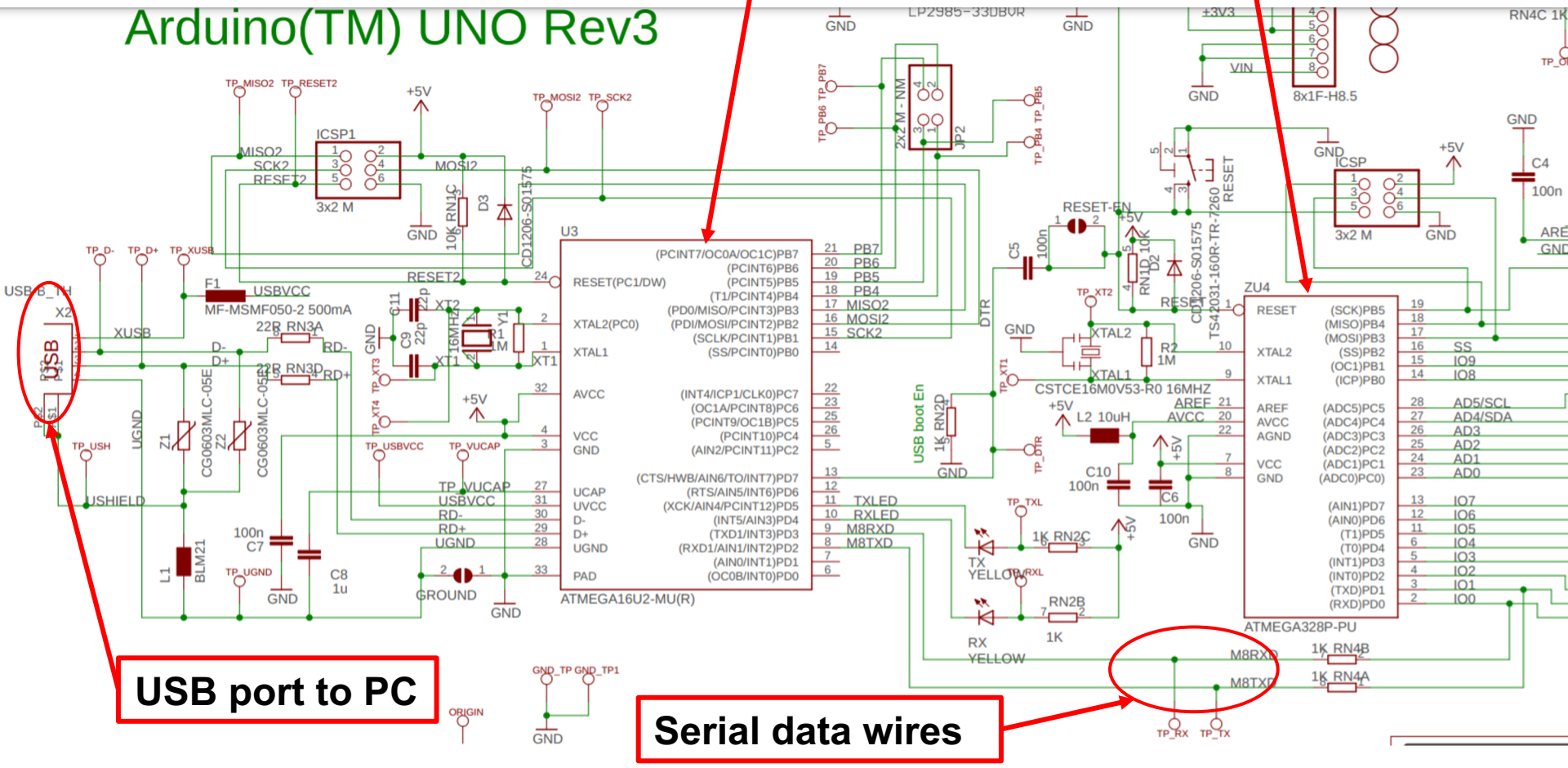
   ❖ Contains code (firmware) that allows reprogramming the program memory.

   ❖ **Example**: Arduino boot loader code reads serial data and writes it to program memory.



Self-programming

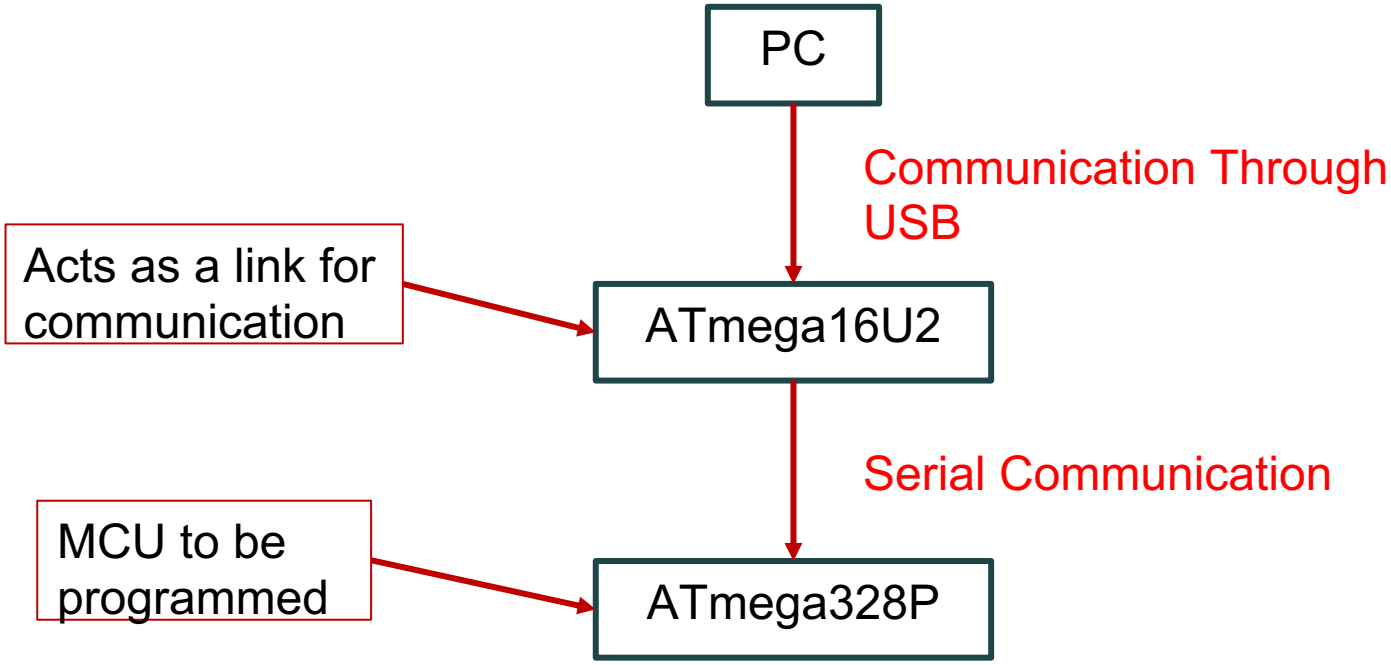
- Arduino Uno Board Circuit:



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science
German University in Cairo

Self-programming

- Programming ATmega328P from PC on Arduino Uno Board:



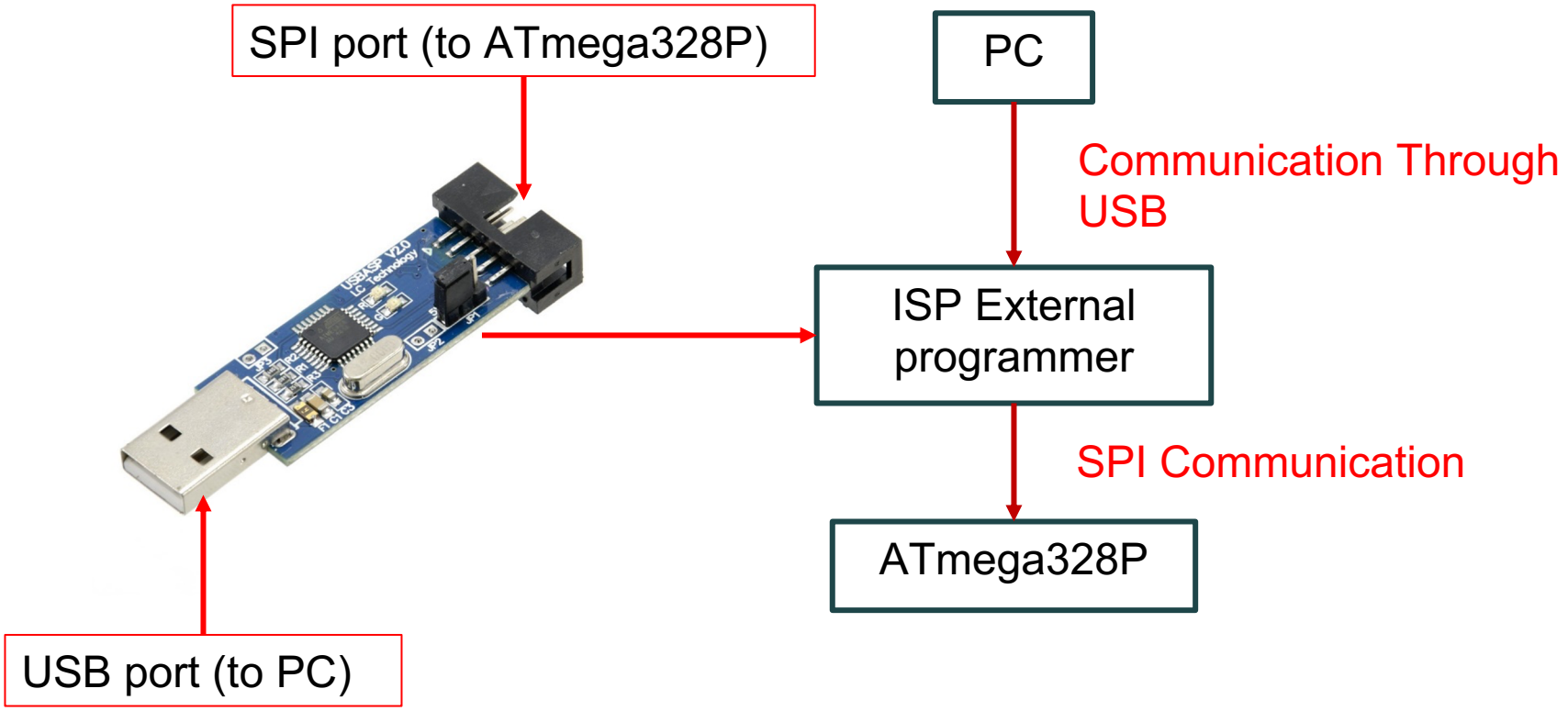
- Code responsible of reprogramming the Atmega328P is written in the bootloader section.

Contents

- **AVR Microcontroller Programming Methods**
  - Self-programming
  - **In-System Programming (ISP)**
    - Hardware Connection
    - Software Setup
    - Fuse Bits Setup
    - Create Project
    - Download Program

In-System Programming (ISP)

- Program Memory can be programmed using SPI communication protocol.
- Connection between ISP programmer (USBasp) and microcontroller:



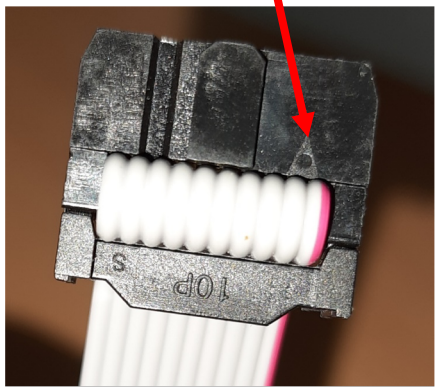
Contents

- **AVR Microcontroller Programming Methods**
  - Self-programming
  - **In-System Programming (ISP)**
    - ❖ **Hardware Connection**
    - ❖ Software Setup
    - ❖ Fuse Bits Setup
    - ❖ Create Project
    - ❖ Download Program

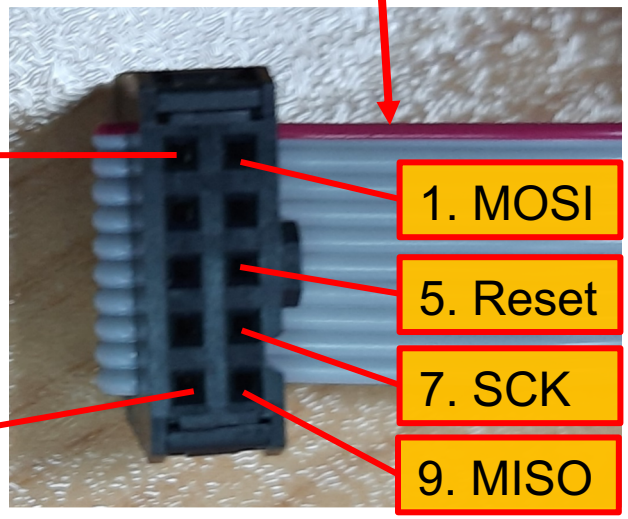
In-System Programming (ISP) Connector

- 10-pin Programmer Connector Label:

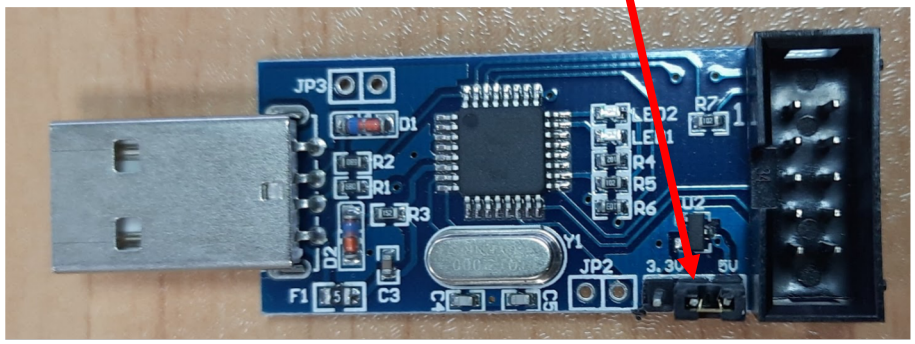
Usually, a triangle indicates pin 1 location



Red Wire Refers to Pin 1

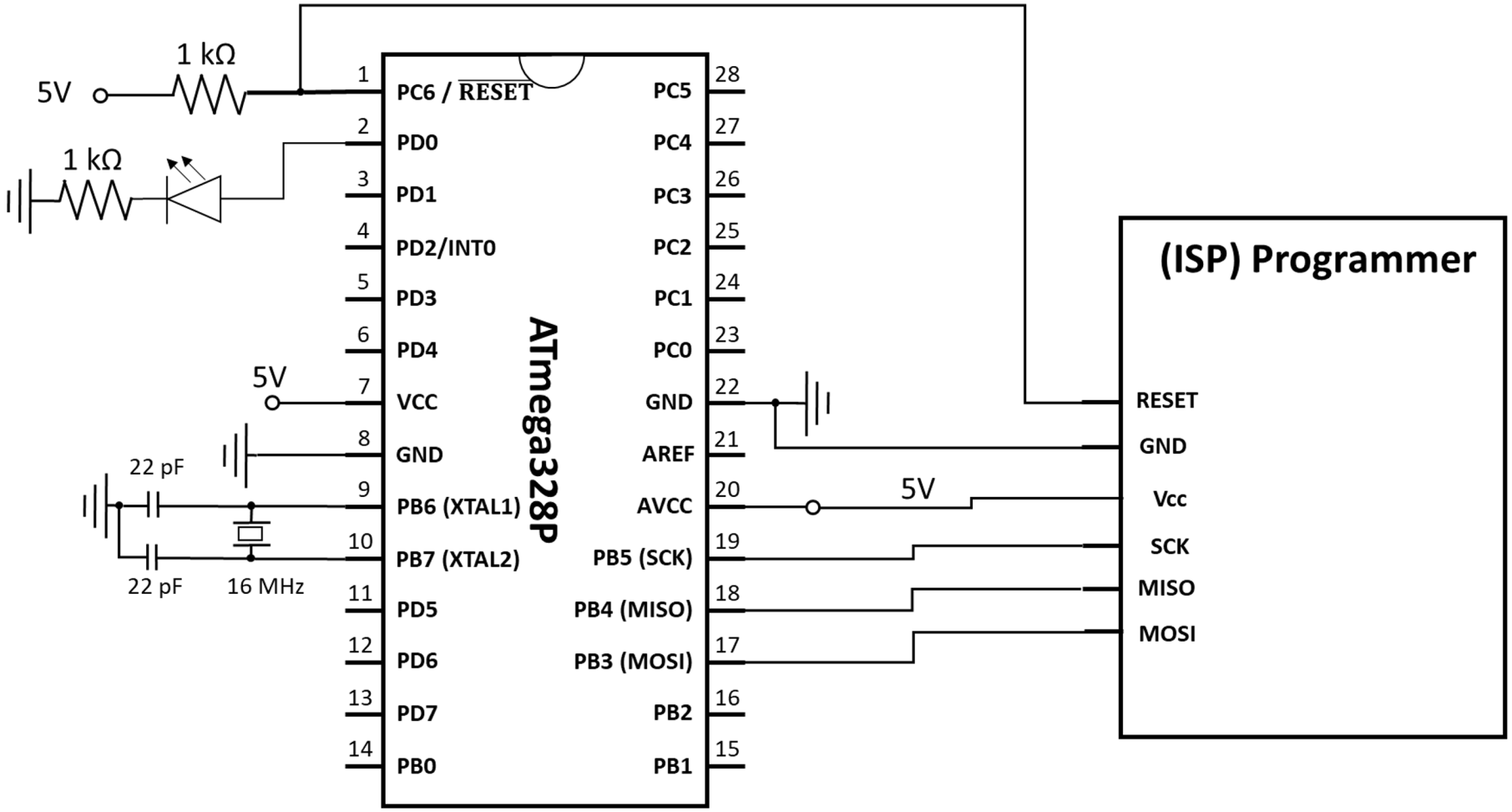


Set Jumper on programmer to 5v

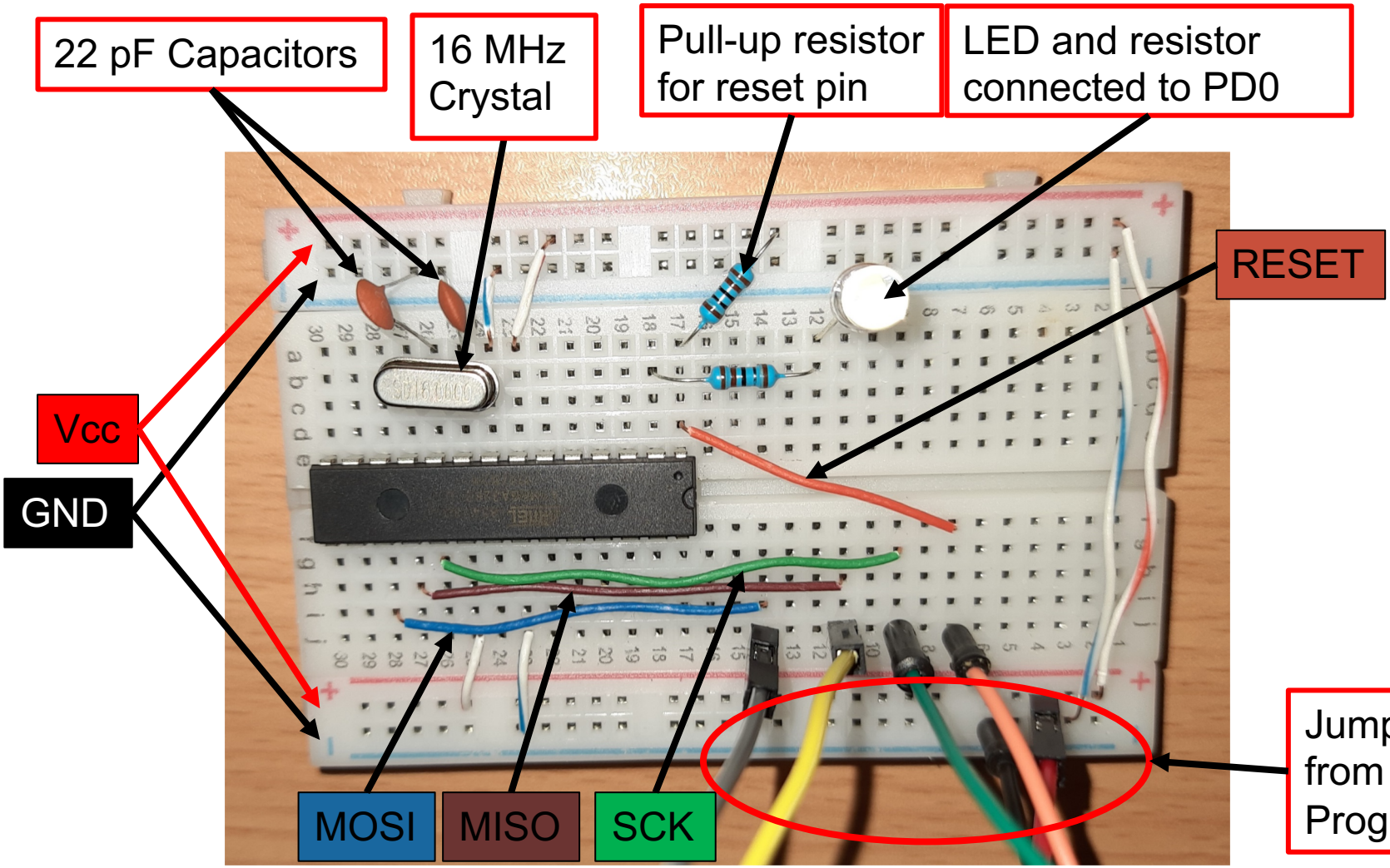


In-System Programming (ISP)

- Schematic diagram of connection between ISP programmer and MCU:



ISP Connection to ATmega328P



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Contents

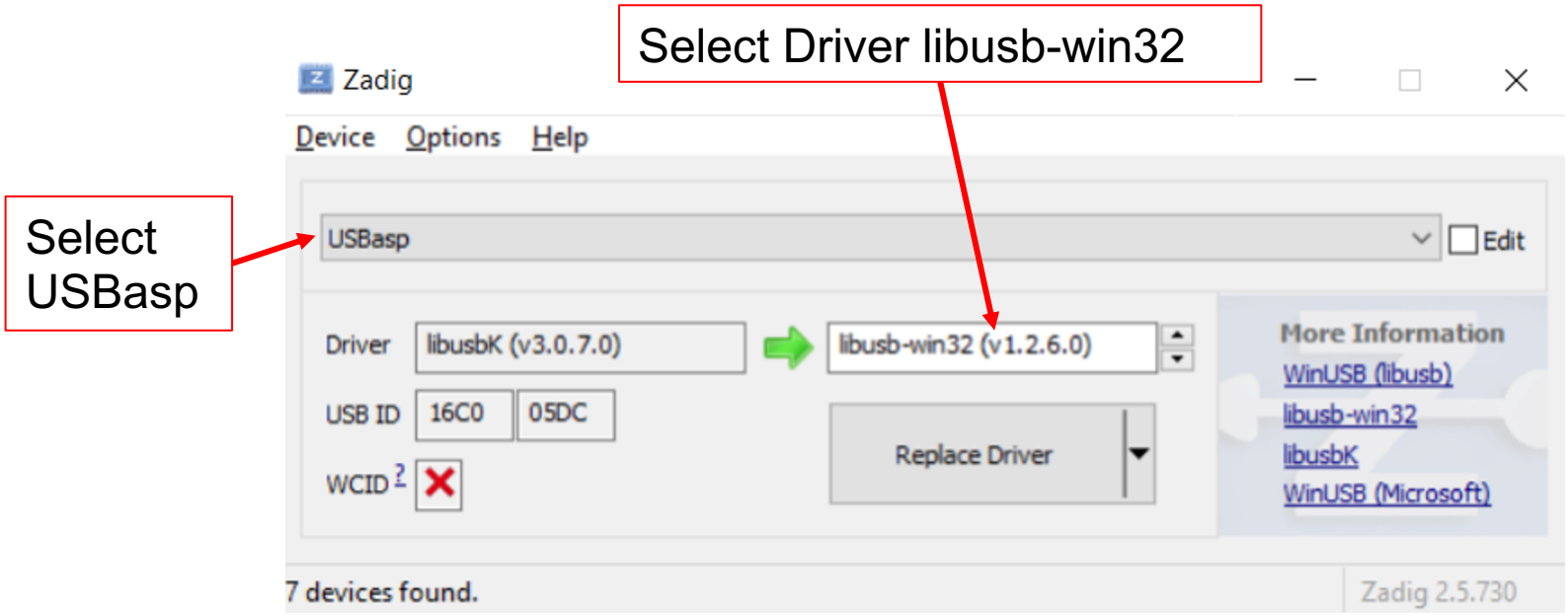
- **AVR Microcontroller Programming Methods**
  - Self-programming
  - **In-System Programming (ISP)**
    - Hardware Connection
    - **Software Setup**
    - Fuse Bits Setup
    - Create Project
    - Download Program

Setup USBasp Driver

1. Connect ISP programmer (USBasp) to the PC through a USB port.

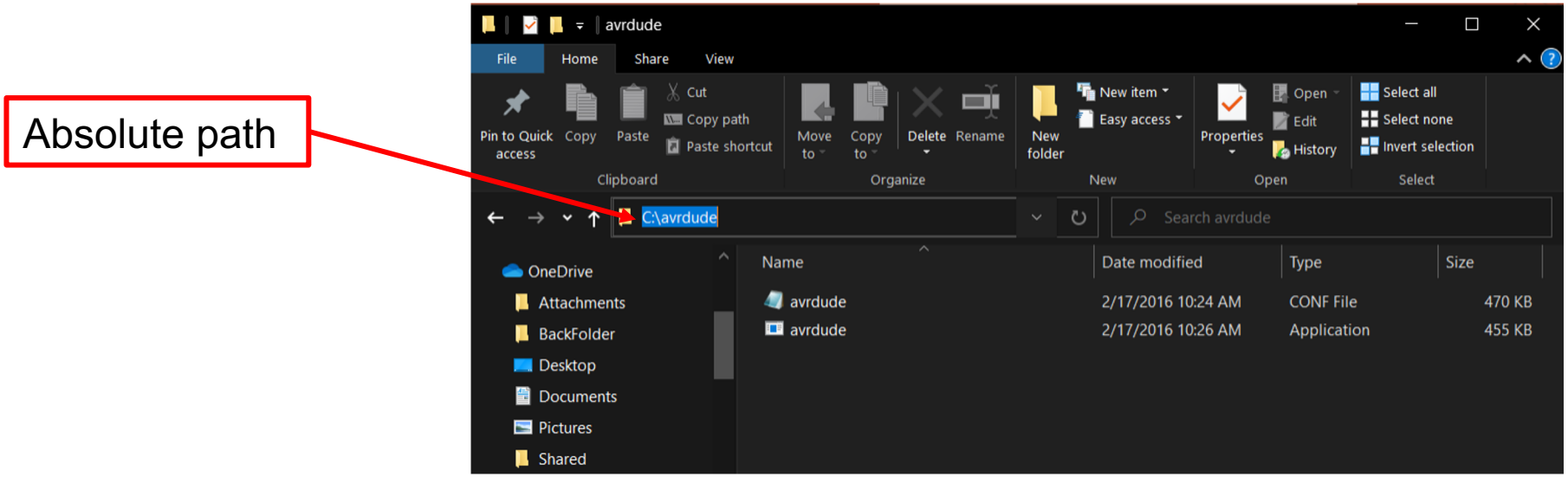
1. To install the driver, download Zadig from https://zadig.akeo.ie/

1. Open Zadig and select USBasp from the dropdown menu. Then set the driver to “libusb-win32” as shown in Figure.



Download AVRdude

- Download AVRdude tool from: http://download.savannah.gnu.org/releases/avrdude/
- If using windows, download the latest mingw32.zip version.
- Extract the archive and obtain the absolute path to “avrdude.exe” (the path will be required later) as shown.



Contents

- **AVR Microcontroller Programming Methods**
  - Self-programming
  - **In-System Programming (ISP)**
    - Hardware Connection
    - Software Setup
    - **Fuse Bits Setup**
    - Create Project
    - Download Program

Set Fuse bits for ATmega328P

- The fuse bits are a group of bits responsible for the configuration of the microcontroller.

- They can be either selected manually from the datasheet or using an AVR fuse calculator such as: https://www.engbedded.com/fusecalc/

- Notes:
  - By default, fuse bits for unprogrammed MCUs are set to use an internal oscillator **"Int. RC Osc 8MHz".** To use an external crystal oscillator the fuse bits must be modified.

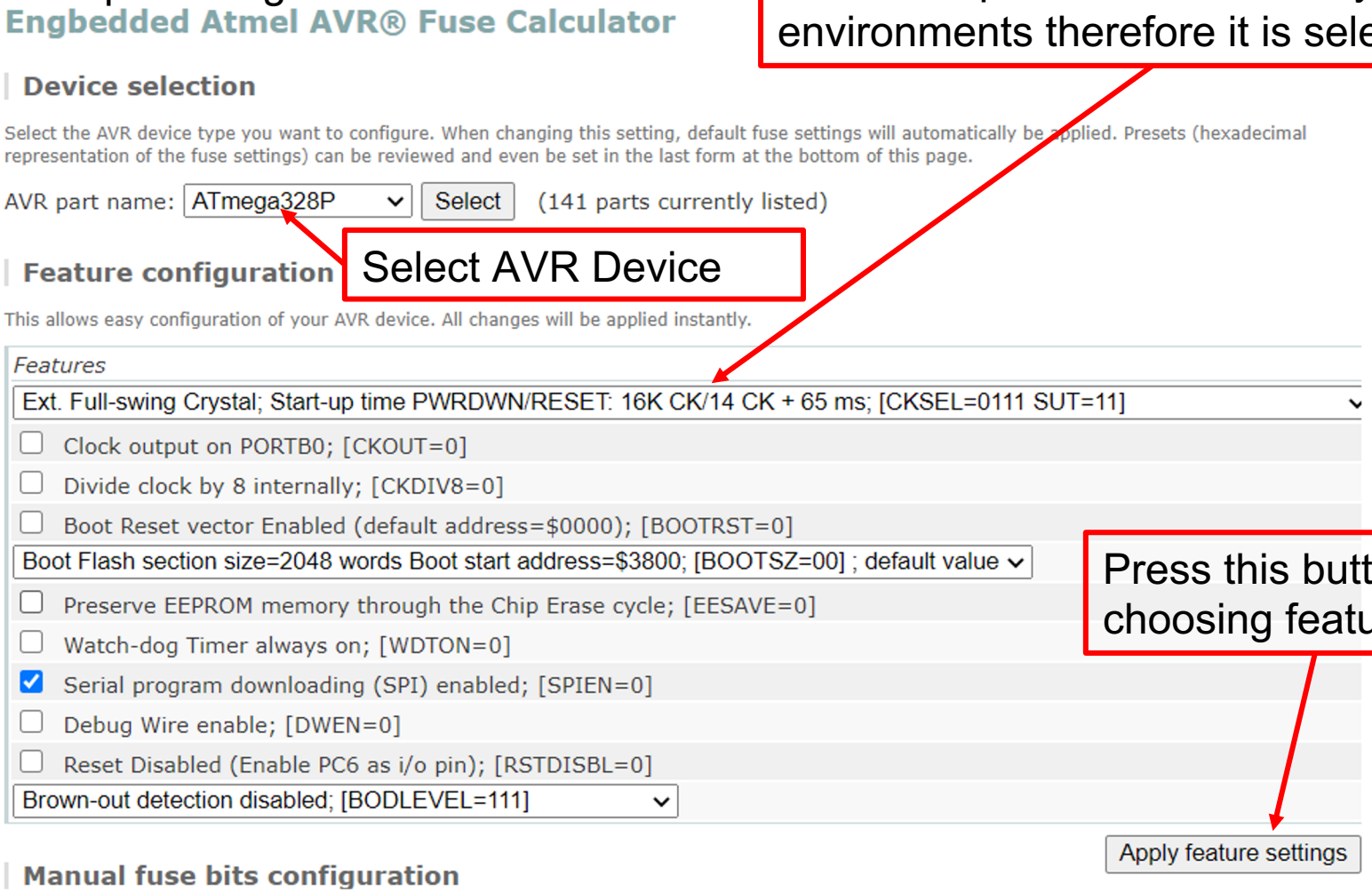
  - Make sure that **"Serial program downloading (SPI) enabled"** is ticked. If not enabled, ISP will not be able to program chip.

  - Make sure that **"Reset Disabled (Enable PC6 as i/o pin)"** is **not** ticked. If reset is disabled, programming with SPI cannot be initiated.

  - The last two options allow serial programming of the MCU, otherwise ISP will not work for the programmed chip.

Set Fuse bits for ATmega328P

- Example configuration of the calculator



Clock Settings (Ext. Full-swing Crystal) has better performance in noisy environments therefore it is selected.

Select AVR Device

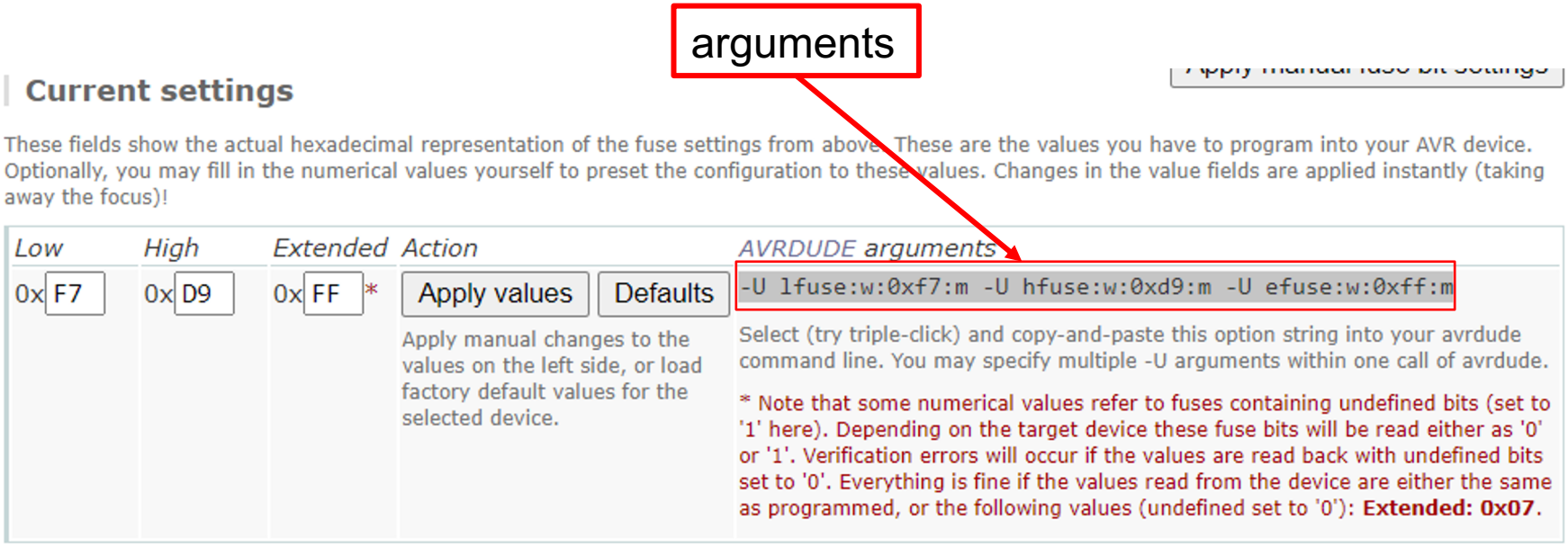
Press this button after choosing features

GUC German University in Cairo

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Set Fuse bits for ATmega328P

- Copy the AVRdude arguments to use them and set the fuse bits.



- Open the command prompt and navigate to the directory of avrdude.

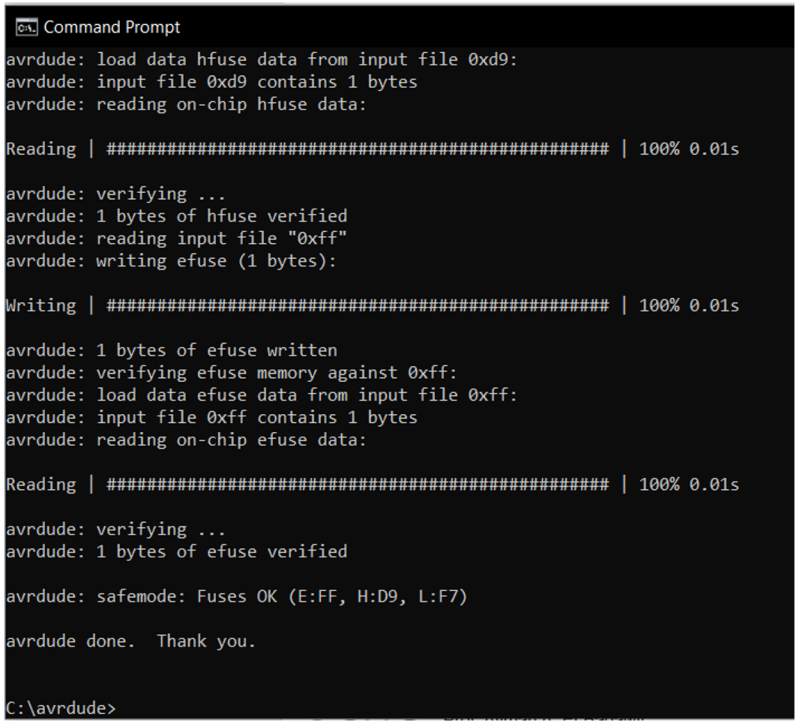
Set Fuse bits for ATmega328P

- While in the avrdude directory, type the following command while USBasp is connected to PC and atmega328p:
  avrdude –c usbasp –p m328p <fuse settings copied from calculator>

- Example:

avrdude -c usbasp -p m328p -U lfuse:w:0xf7:m -U hfuse:w:0xd9:m –U efuse:w:0xff:m

- The command prompt should show positive results indicating that the fuse bits were successfully modified.

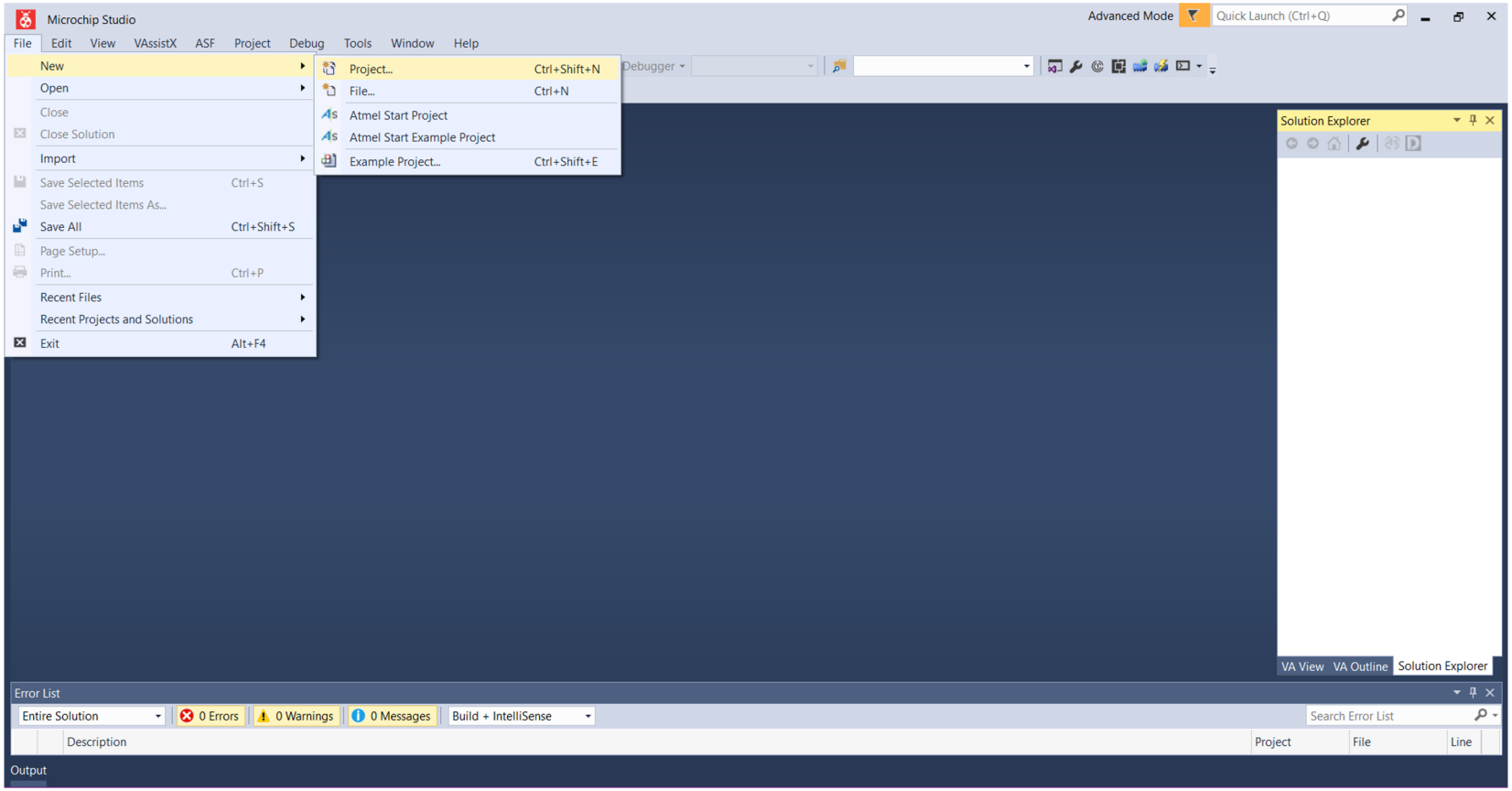


Contents

- **AVR Microcontroller Programming Methods**
  - Self-programming
  - **In-System Programming (ISP)**
    - Hardware Connection
    - Software Installation
    - Fuse Bits Setup
    - **Create Project**
    - Download Program

Create New Project in Microchip Studio

- Open Microchip Studio
- Select **File>New>Project**

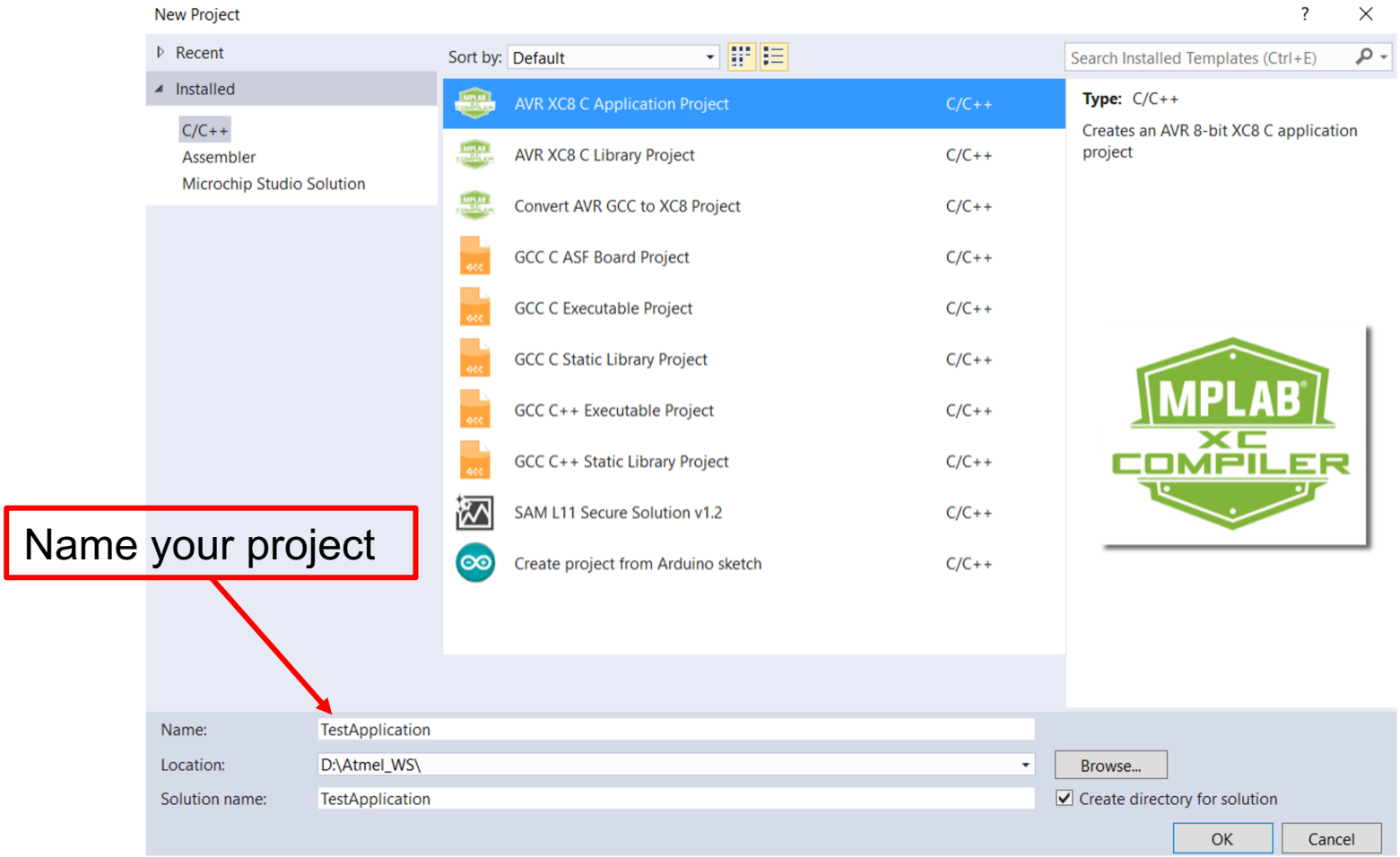


GUC German University in Cairo

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Create New Project in Microchip Studio

- Select AVR XC8 Application project

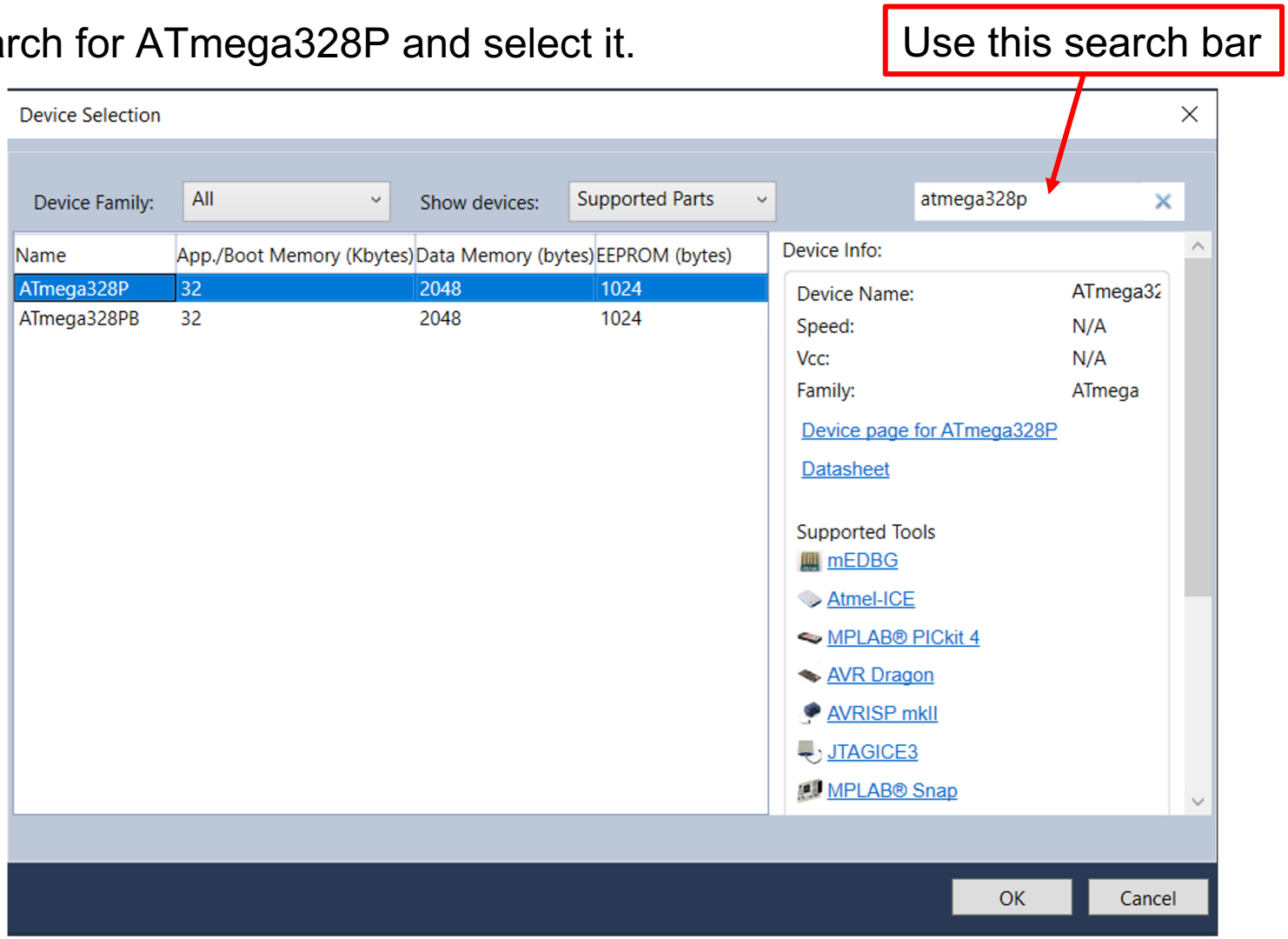


GUC German University in Cairo

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Create New Project in Microchip Studio

- Search for ATmega328P and select it.



Create New Project in Microchip Studio

- Write your code in “main.c” (The code shown toggles pin PD0 every 1 second).

main.c*

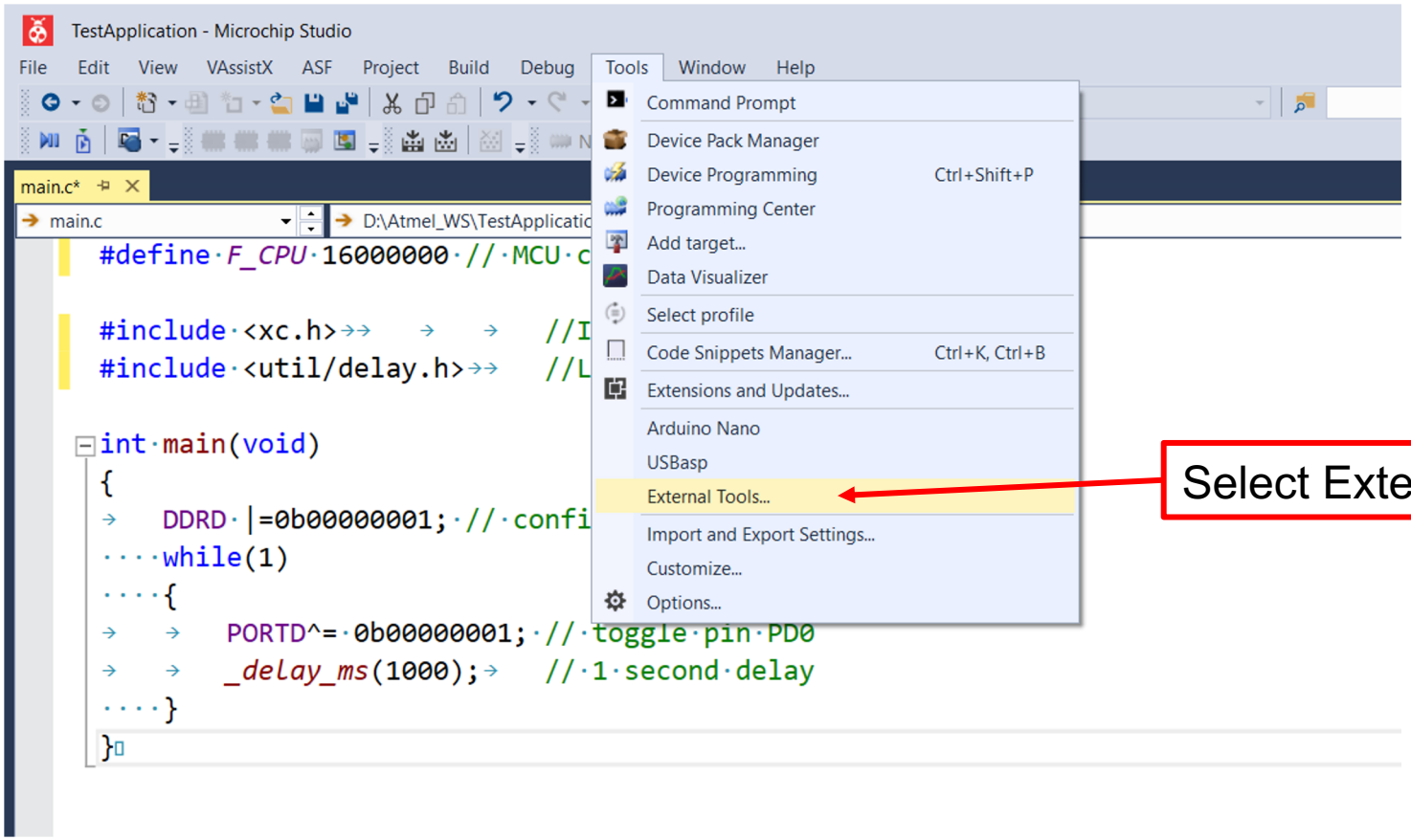
main.c — D:\Atmel_WS\TestApplication\TestApplication\main.c

```
#define F_CPU 16000000 // MCU clock frequency in Hz

#include <xc.h>          //Include for AVR references
#include <util/delay.h>  //Library for delay function

int main(void)
{
    DDRD |=0b00000001; // configure pin PD0 as output
    while(1)
    {
        PORTD^= 0b00000001; // toggle pin PD0
        _delay_ms(1000);    // 1 second delay
    }
}
```

Create USBasp Tool in Microchip Studio



Select External Tools

Create USBasp Tool in Microchip Studio

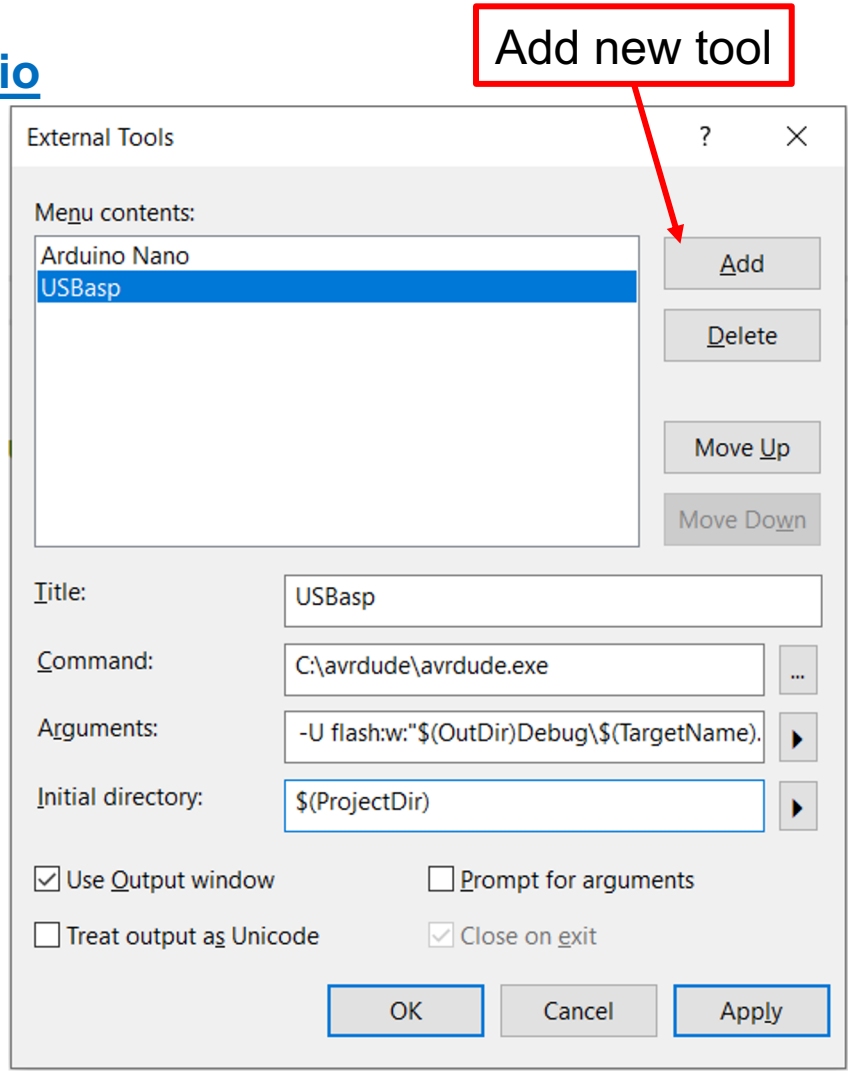
- Click on the “Add” button.
- The tool should be created with the following specifications:

1. Command should be **avrdude.exe** with absolute path. In this example it is “C:\avrdude\avrdude.exe”.

1. The arguments should be as follows: -c usbasp -p m328p -P usb -U flash:w:"$(OutDir)Debug\$(TargetName).hex"

1. Initial Directory: $(ProjectDir)

1. Mark “Use Output Window”



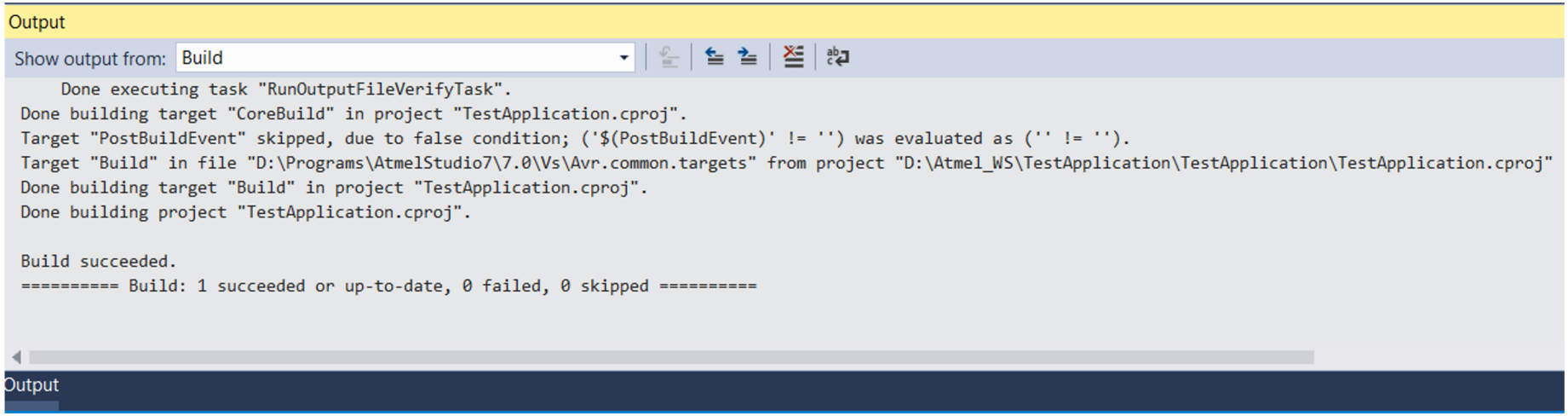
Note: -c argument defines programmer board, while –p defines the AVR device

Contents

- **AVR Microcontroller Programming Methods**
  - Self-programming
  - **In-System Programming (ISP)**
    - ❖ Hardware Connection
    - ❖ Software Setup
    - ❖ Fuse Bits Setup
    - ❖ Create Project
    - ❖ **Download Program**

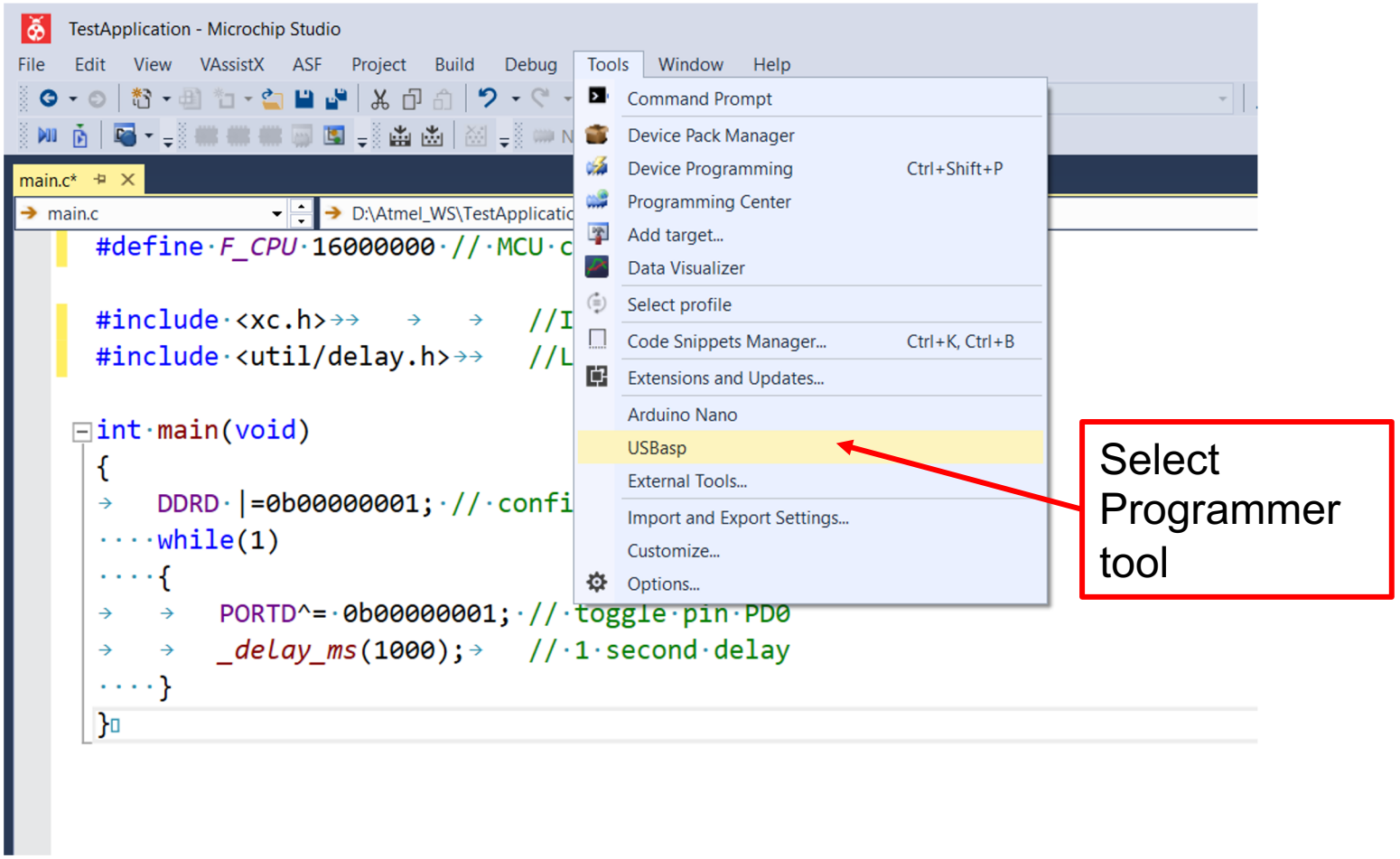
Download Code to MCU

- Build the project from **Build>Build Solution**
- The output window should show “Build Succeeded”.



Download Code to MCU

- To download the code, select the external tool created for USBasp:



GUC German University in Cairo

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Download Code to MCU

- Finally, check the output window to verify that the MCU has been programmed successfully.

- Now the LED connected to PD0 should be blinking as instructed in the code.