

# Arduino as In-Circuit Serial Programmer

## Contents

<b>1</b>	<b>Acknowledgements</b>	<b>2</b>
<b>2</b>	<b>Introduction</b>	<b>2</b>
<b>3</b>	<b>Software Resources</b>	<b>2</b>
<b>4</b>	<b>Steps to burn your program</b>	<b>3</b>
4.1	Step 1: Setting up Arduino Board . . . . .	3
4.2	Step 2: Connect Arduino with microcontroller . . . . .	3
<b>5</b>	<b>Step 3: Burn Hex file in microcontroller</b>	<b>3</b>

# 1 Acknowledgements

This tutorial is prepared by Mohammad Azfar Tariq and Muhammad Usman under the supervision of Dr. Rehan Ahmed for the course EE-222 Microprocessor Systems. Reporting any error or discrepancy found in the text is appreciated.

## 2 Introduction

Dispatching microcontroller from circuit and placing it on a high voltage programmer (MAXLOADER or SUPERPRO) again and again is tiresome. To avoid this inconvenience in-circuit serial programming or generally, in-system programming is used. It is a technique where a programmable device (like a microcontroller) can be programmed while it is still in a circuit.

This is usually done using a host device. In our case we will be using ATmega328P (a family member of ATmega16A) mounted on Arduino UNO board. The communication protocol used for this purpose is Serial Peripheral Interface (SPI). To learn what SPI actually is consult SparkFun website<sup>1</sup>.

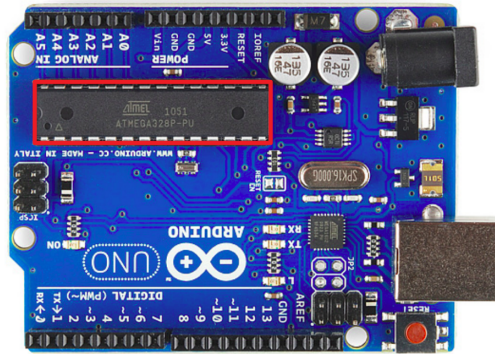


Figure 1: In red boundary ATmega328P

## 3 Software Resources

There are two software programs we need to undertake the task in hand

- Arduino IDE<sup>2</sup>
- WinAVR

Download WinAVR from this website<sup>3</sup>

<sup>1</sup><https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all>

<sup>2</sup><https://www.arduino.cc/en/Main/Software>

<sup>3</sup><https://sourceforge.net/projects/winavr/files/WinAVR/20100110/>

## 4 Steps to burn your program

### 4.1 Step 1: Setting up Arduino Board

- Start Arduino IDE
- Move to Files > Examples > ArduinoISP > ArduinoISP.
- A sketch will be opened. upload the sketch on Arduino after selecting the appropriate port.

### 4.2 Step 2: Connect Arduino with microcontroller

Patch the circuitry as;

Arduino pin	$\mu$ C pin
11	MOSI
12	MISO
13	SCK
10	RST
VCC	VCC
GND	GND

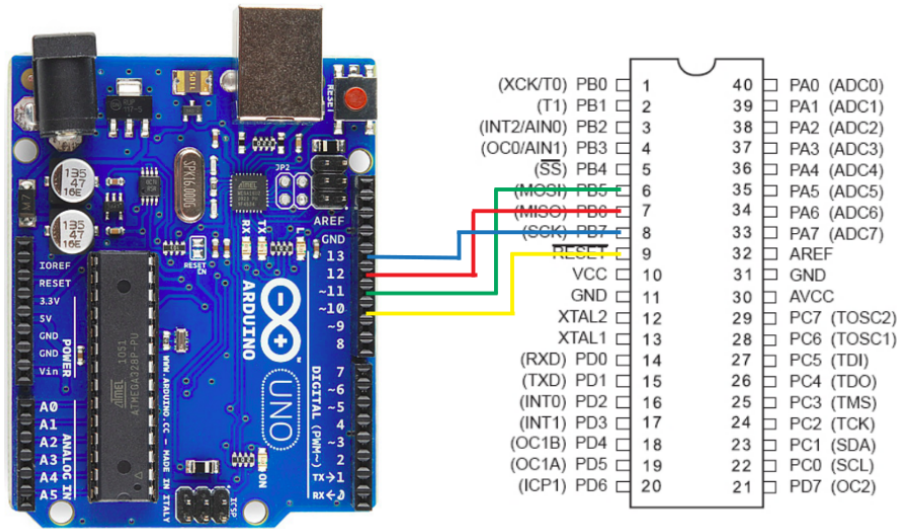


Figure 2: Circuit

## 5 Step 3: Burn Hex file in microcontroller

1. Keep in mind the COM port your arduino is connected. You can check it in Arduino IDE Tools > Ports.

2. Open Command Prompt.
3. Move to the folder that contains relevant hex file. This can be achieved by entering `cd` space folder path in CMD prompt. For example to move in folder “System32”, which is present in folder “Windows”, which in turn is present in folder “C” we have to type,

```
1 cd C:\Windows\System32
```

4. Now run the following command replacing COM13 with your COM Port name and abc.hex with the name of your hex file.

```
1 avrdude -c arduino -p m16 -P COM13 -b 19200 -U flash:w:abc.hex
```