Lab 2

Introduction to AVR Microcontroller Hardware Circuitry

Objectives

- Build and understand the minimal circuit required to start using a microcontroller
- Learn to program (download machine code to program memory of) a microcontroller using Arduino board

Software

- Microchip Studio (Version 7)
- Proteus ISIS
- Arduino IDE

Hardware

Name	Value	Quantity
Arduino Nano With USB Cable	-	1
Breadboard	-	1
LED	-	1
Resistor	470Ω	1

Table 2.1 - List of Components

Theory

For the purpose of this lab, students need to use a microcontroller. A microcontroller is a programmable IC that has many functionalities built-in the chip itself. Furthermore, microcontrollers have the ability to send and receive both digital and analogue data via pins. The main outcome of these lab sessions is that students should be able to **program** microcontrollers for a variety of use cases by the end of the semester.

The microcontroller that is recommended for this course is the AVR ATmega328P. The ATmega328P is a low-power CMOS 8-bit microcontroller. The AVR core has a rich (131) instruction set. Most instructions require single clock cycle for execution. It has 32K bytes of in-system programmable flash program memory, 1K bytes EEPROM, 2K bytes SRAM, 23 general purpose I/O pins and 32 general purpose registers. Some of the features are as follows:

- 3 x flexible Timer/Counters with compare modes
- 6 x PWM Channels

- Internal and External Interrupts
- A serial programmable USART
- Two Master/Slave SPI Serial Interface
- 8 Channel 10-bit ADC

ATmega328P Pin configuration

- The 23 digital I/O pins are grouped into 3 ports named as Port B, C and D:
- Port B and D have 8 pins each whereas Port C has 7 pins
- Pins 4 and 6 are required to connect with +ve (VCC) and Pins 3, 5 and 21 need to connect with ground (GND)
- Pin 18 is the supply voltage pin while Pin 20 is the analogue reference pin for the built-in ADC

The Pin configuration of ATmega328P can be seen in Figure 2.1.

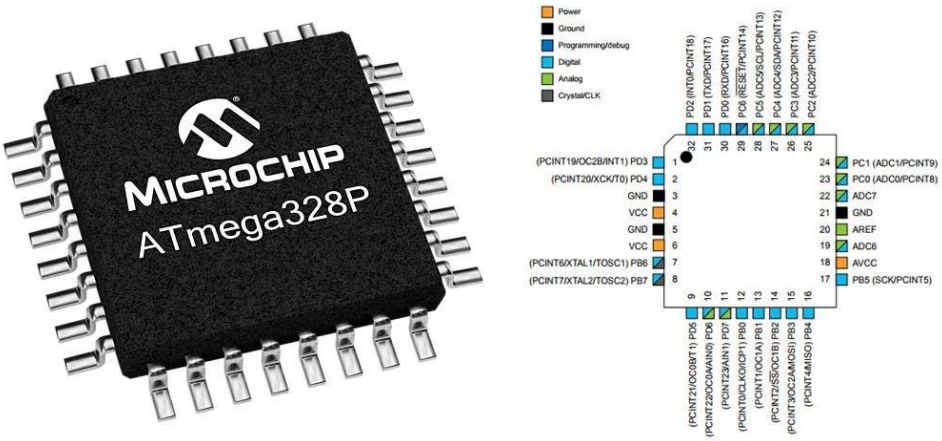

Figure 2.1 – Basic Pin-out of Atmega328P Microcontroller

System Clock and Clock Options

The system clock frequency refers to the frequency generated from the system clock pre-scaler. The AVR microcontroller unit (MCU) sports internal clock as well as external clock source. System can run on 1MHz internal clock or external crystal oscillator circuits. The device is shipped with internal RC oscillator at 8.0MHz and with the fuse CKDIV8 programmed, resulting in 1.0MHz system clock.

The ATmega328p can operate up to 20 MIPS at 20-MHz.

On-board Controller

While it is possible to use a chip directly, it is sometimes much easier to use and program them when the controller is mounted on a board. It so happens, that the ATmega328p comes mounted on a number of boards. Students are recommended to use one of the two most popular variants of these boards, namely:

1. Arduino Nano
2. Arduino Uno

The Arduino Uno is more flexible to use with a removable microcontroller, the Nano is a cheaper alternative that matches most relevant capabilities of the Uno.

Arduino Nano

An Arduino Nano is small, cheap and breadboard friendly platform with the ATmega328p surface mounted on it. The Arduino Nano is programmed using the Arduino Software (IDE). To program Arduino Nano in offline mode, you need to install the Arduino Desktop IDE. Arduino after connecting a PC via USB cable. This also provides power to the board, as indicated by an LED (On the bottom of the Arduino Nano 2.x and the top of the Arduino Nano 3.0). The Nano can be seen in figure 2.2 while the pinout of the Nano is illustrated in figure 2.3.

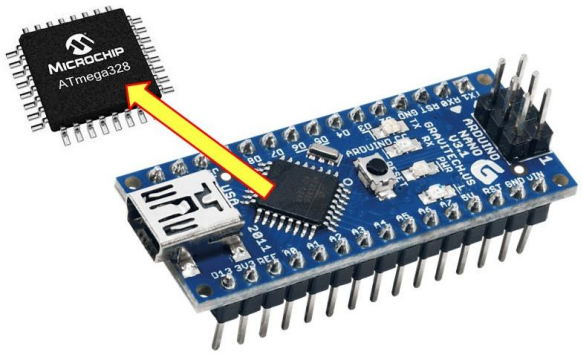

Figure 2.2 – An Arduino Nano Board

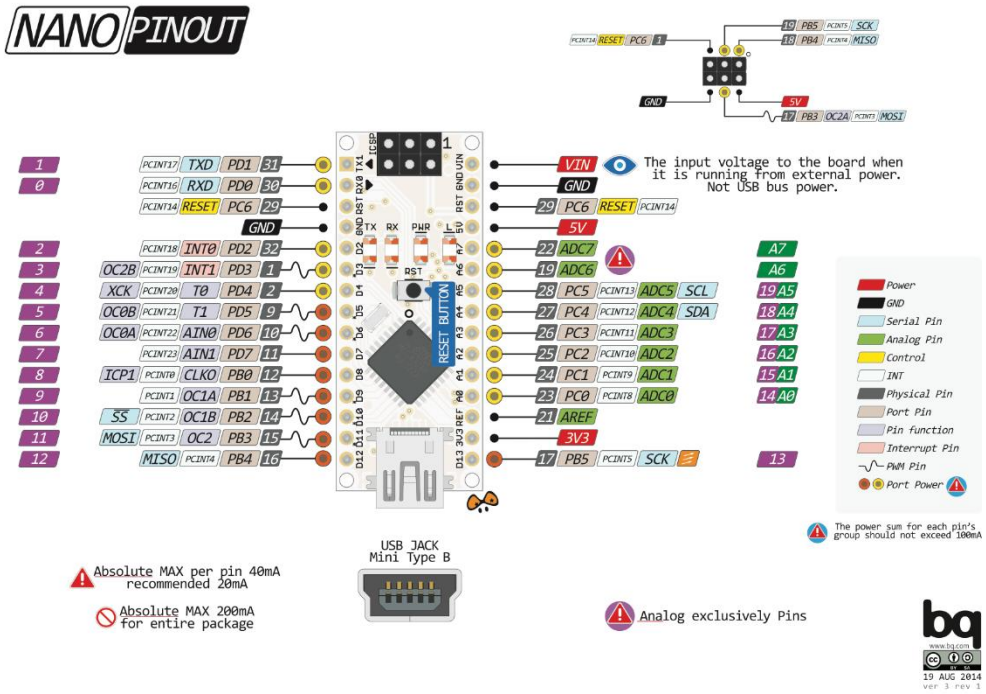

Figure 2.3 – Basic Pin-out of the Arduino Nano Development Board

The Arduino Nano has the following salient features:

1. Automatic reset during program download
2. Power OK LED
3. TX, RX, L LED
4. Auto sensing/switching power input
5. Small mini-B USB for programming and serial monitor
6. ICSP header for direct program download

7. Standard 0.1" spacing DIP (breadboard friendly)
8. Manual reset switch

Arduino Uno

Please review figure 2.4 and 2.5 for Arduino Uno. Do note that both boards use the AVR Atmega328P microcontroller.

Figure 2.4 – An Arduino Uno Board

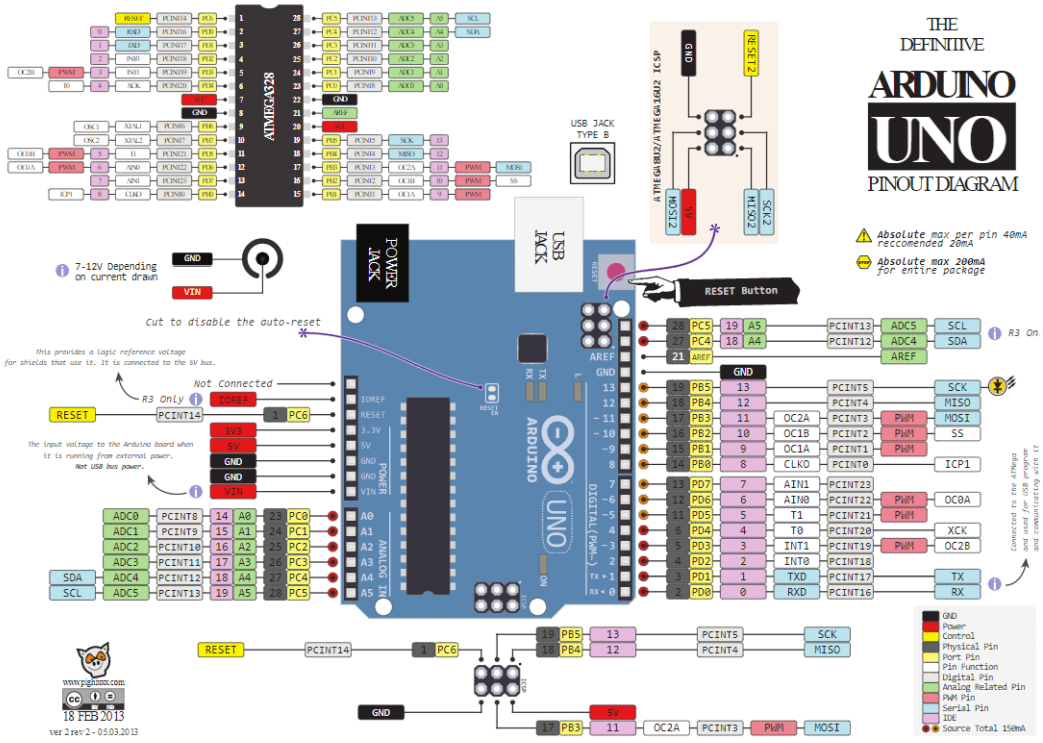

Figure 2.5 – Basic Pin-out of the Arduino Uno Development Board

AVRDUDE (Installed automatically with Arduino IDE)

AVR Downloader Uploader (AVRDUDE) – is a program for downloading and uploading the on-chip memories of Atmel's AVR microcontrollers. It can program the program memory and the EEPROM. Where supported by the serial programming protocol, it can program fuse and lock bits. AVRDUDE is a command line program and can be used with Microchip Studio for programming AVR microcontrollers.

Using AVRDUDE

1. AVRDUDE requires LibUSB drivers. LibUSB must installed along with Arduino IDE. If it is not installed, it has to be downloaded separately and installed.
2. Make sure that the computer/laptop running Microchip Studio is connected to a Nano or Uno via USB cable. The power LED on Arduino should turn on.
3. In device manager, a COM Port will be listed under Ports (COM and LPT). If the device is not detected, installing CH340 drivers may help. Get the port number for your Arduino.
4. In Microchip Studio, under '***Tools***', select '***External Tools***'. Add the title '***Arduino Nano***'. Type in the path to avrdude.exe in the '***Command***' text box. The default path is:

   C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avrdude.exe

5. In the '***Arguments'*** box type in the following

   -C"C:\Program Files (x86)\arduino\hardware\tools\avr\etc\avrdude.conf" -v -patmega328p -carduino -PCOM5 -b57600 -D -Uflash:w:"$(ProjectDir)Debug\$(TargetName).hex":i

6. The above arguments tell Avrdude four important things,
   a. Path to the generated hex file
   b. Name of the target microcontroller (-partmega328p)
   c. Download speed (57600) in bits per second.
   d. Port assigned to Arduino. (COM5 in this case. Yours may vary)
7. The above setup is required only once. When you want to program your Arduino, you will just select '***Arduino Nano***' under '**Tools**' and the program will be downloaded to your microcontroller.

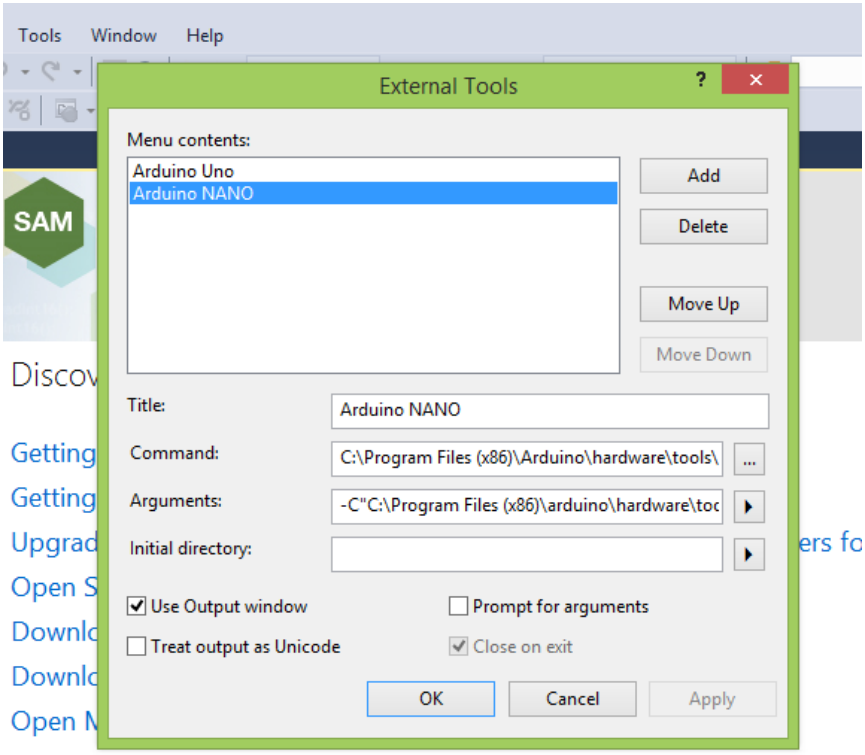

Figure 2.6 – Burning a .hex file using AVRDUDE and Microchip Studio

Pre-Lab Tasks

Task-1

Please review the theory section of this lab comprehensively.

Task-2

Install AVRDUDE on to your computer/laptop. It gets installed automatically when you install Arduino IDE)

Task-3

Please download the following datasheets from the internet:

- ATmega328P
- Arduino Uno or Arduino Nano (depending on what you are planning to buy)

Please review the Uno/Nano documents as per your need.

In-Lab Tasks

Task-1

Set up your Arduino Nano or Uno with a resistor and LED as per the following figure.

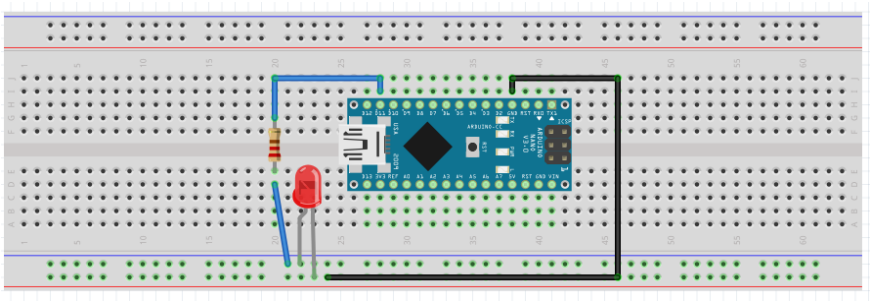

Figure 2.6 – Breadboard View Basic Circuit of ATmega328P

Task-2 – Write Assembly Code

Build the following program on Microchip Studio and **add comments against each line of code**

Assembly Code

```
;this program adds value 3 R20 ten times and
;passes it on to port B

.INCLUDE "M32DEF.INC"
      LDI R16, 10
      LDI R18, 0xFF
      LDI R20, 0
      LDI R21, 3

      OUT DDRB, R18
AGAIN:
      ADD R20, R21
      OUT PORTB, R20
      DEC R16
      BRNE AGAIN
```

Task-3 – Write C-Code

Build the following program on Microchip Studio.

C-Code

```
include <avr/io.h>
#define F_CPU 16000000UL
#include <util/delay.h>

int main(){
      DDRB = 0xFF;
      while(1){
            PORTB |= (1<<PB3);
            _delay_ms(1000);
            PORTB &= ~(1<<PB3);
            _delay_ms(1000);
      }
      return 0;
}
```

Task-4

Burn the .hex file from Task-2 and Task-3 onto your Arduino using AVRDUDESS, as per the instructions in the theory section. Connect and LED on PB3 and observe ana document the behaviour of the LED.

Post Lab

Task-1

Set up your Arduino Nano / Uno with a button and LED as per the following figure.

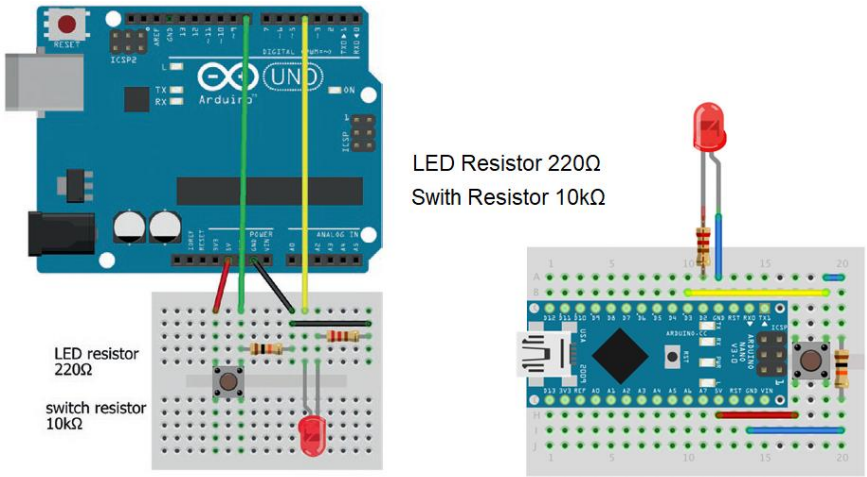

Figure 2.7 – Breadboard View Basic Circuit of ATmega328P

Task-2

Modify the program given in the 'In-Lab' task so that it toggles the LED only when a button is pressed. Connect the button to any port pin of your own choice.

Task-3

Burn the .hex file from Task-2 onto your Arduino, as per the instructions in the theory section. Observe the behaviour of the LED.

Critical Analysis / Conclusion (To be filled in by the student)

Lab Assessment (To be filled by the lab-instructor)		
Pre-Lab	/5	/25
In-Lab	/5	
Results	/5	
Viva	/5	
Critical Analysis	/5	
Comments:		
Instructor Name		Instructor Signature