EGR 270
Fundamentals of Computer Engineering
File: N270L8

Lab # 8
The ATmega328P (Arduino Nano) and Assembly Language Programming

Lab Format

- This is a **Individual Lab** so each student must design and test their own circuits.
- Students are free to assist each other in all labs.
- Each student must complete the Preliminary Work Section **before** lab begins. Preliminary Work will be checked in lab and will be part of the lab report grade.
- Each student must submit his or her own lab report.
- Lab reports will not be accepted until all required circuits have been demonstrated to the instructor.

A. Objectives

The objectives of this laboratory are introduce the student to:

- assembly language programming
- Atmel Studio 7 (for assembling and simulating programs)
- ATmega328P Microcontroller (Arduino Nano)
- Breadboarding the Arduino Nano and using it to control LEDs and 7-segment displays

B. Materials

Breadboard	Atmel Studio 7
5V Power Supply	Common-anode 7-segment display (LDS-3221 or other)
Arduino Nano	Seven 220 ohm resistors

C. Reference

Refer to the following items (available on the instructor's web page):

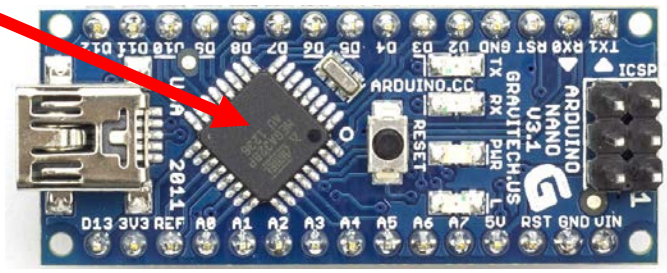
- ***Atmel Studio Tutorial (PowerPoint)***
- ***Computer Architecture, Microprocessors and Assembly Language (PowerPoint)***

D. Introduction

Refer to the references above for more detail on the items shown in this section.

In this lab we will be using AVR Assembly Language to program the ATmega328P microcontroller on the Arduino Nano. The pin numbers on the Arduino are different from the pin numbers on the ATmega328P, so a ***Pin Mapping Table*** is shown below.

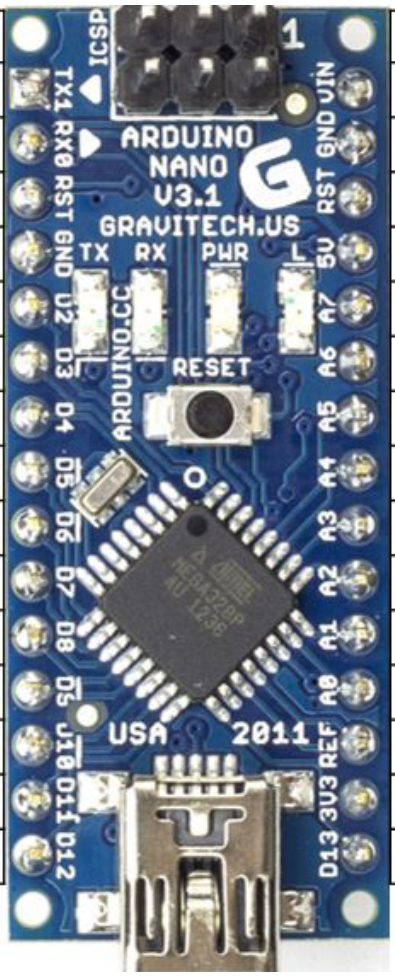
ATmega328P



Arduino Nano

Table 1: Arduino Nano – Atmega328P Pin Mapping Table

ATmega328P *	Nano	Pin	Pin	Nano	ATmega328P *
PD1 (TXD)	D1 (TX1)	1	30	VIN	VIN
PD0 (RXD)	D0 (RX0)	2	29	GND	GND
RST	RST	3	28	RST	RST
GND	GND	4	27	5V	VCC
PD2 (INT0)	D2	5	26	A7	ADC7
PD3 (INT1)	D3	6	25	A6	ADC6
PD4 (XCK)	D4	7	24	A5	PC5 (SCL)
PD5 (OC0B)	D5	8	23	A4	PC4 (SDA)
PD6 (OC0A)	D6	9	22	A3	PC3 (ADC3)
PD7 (AIN1)	D7	10	21	A2	PC2 (ADC2)
PB0 (CLK0)	D8	11	20	A1	PC1 (ADC1)
PB1 (OC1A)	D9	12	19	A0	PC0 (ADC0)
PB2 (OC2A)	D10	13	18	AREF	AREF
PB3 (MOSI)	D11	14	17	3V3	
PB4 (MISO)	D12	15	16	D13	PB5 (SCK)



* Many pins have additional functions not listed

Output Ports on the ATmega328P

Recall that the outputs are arranged into 3 ports (Port D, Port B, Port C):

Output Port	Notes
Port D (in Yellow above)	Avoid PD0,PD1 (used for Rx,Tx) 6 free digital pins (PD2-PD7)
Port B (in Blue above)	5 free digital pins (PB0-PB4)
Port C (in Green above)	8 free analog pins (A0-A7) - not used in this course

There are 11 free digital pins to use for this lab: PD2-PD7, PB0-PB4

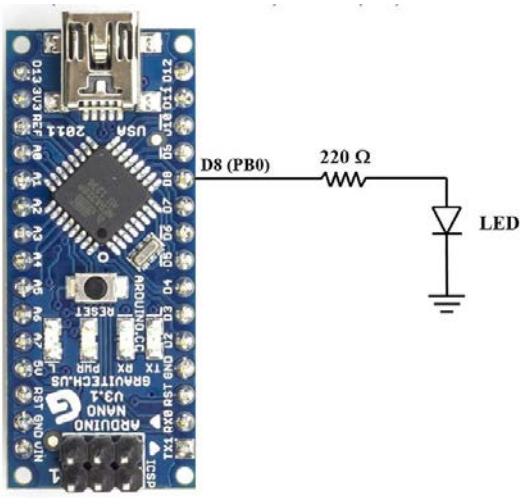
Lab 8 Circuits and Programs

In this lab two circuits will be constructed and two assembly language programs will be used with the circuits.

- Circuit 1 – Blink an LED on pin D8 (PB0)
- Circuit 2 – Display the 7 digits of your student ID in order on a 7-segment display

These circuits and programs will be discussed on the following pages.

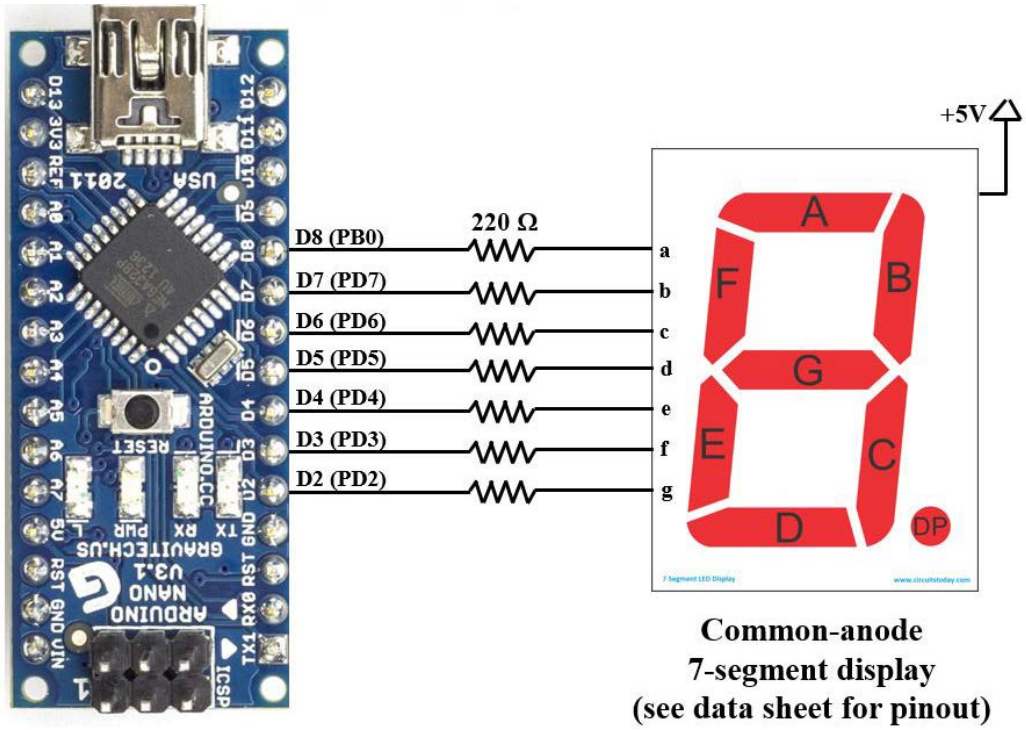
Lab 8 Circuit 1: PB0 (D8) is used in Circuit 1 below. Note that we could have used any of the 11 free digital pins.



Lab 8 Program 1: This program has already written and is discussed in ***Atmel Studio Tutorial***
Download this program (Blink LED on Arduino Nano pin D8 at 1 Hz) from the course website.

```
; EGR 270 - Lab 8 - Circuit 1
; Project name:  Blink LED on Arduino Nano pin D8 at 1 Hz
; Created: 3/10/20xx 2:17:17 PM
; Author : John Doe

.DEF dly1 = R17                 // Assign the name dly1 to R17
.DEF dly2 = R18
.DEF dly3 = R19

 .ORG   0x0000                  // Store program at address 0x0000 in flash memory
 RJMP   main                    // Program begins at the main label

 main:
 LDI    r16, 0xFF               // Load the immedate value 0xFF (all bits 1) into register 16
 OUT    DDRB, r16               // Set Data Direction Register B to output for all pins

 loop:
 SBI    PortB, 0                // Set bit 0 of PortB. (i.e. turn on the LED)
                                // Notes:  PB0 maps to pin D8 on the Arduino Nano
                                //         Refer to Pin Mapping Table in the class notes
 RCALL  delay_500ms             // Call delay_500ms subprogram to keep LED lit for 0.5s
 CBI    PortB, 0                // Clear bit 0 of PortB. (i.e. turn off the LED)
 RCALL  delay_500ms             // Call delay_500ms subprogram to keep LED off for 0.5s
 RJMP   loop                    // Loop again

 // Everything beneath is part of the delay subprogram (or subroutine)
 // Note that 8 bit registers hold a max value of 255 (unsigned) so creating a 0.5 s delay requires using a triple nested loop.
Delay_500ms:                    ; For CLK(CPU) = 16 MHz (clock rate for Arduino Nano) - Lab test showed 60 blinks in 1 minute test
                                ; T = 1/16MHz = 62.5 ns.  Delay = 62.5ns(8,000,000 cycles) = 0.5s
    LDI     dly1,   64          ; One clock cycle: outer loop: 64 * 125000 cycles = 8,000,000 cycles
Delay1:
    LDI     dly2,   125         ; One clock cycle: middle loop:  125 * 1000 cycles = 125000 cycles
Delay2:
    LDI     dly3,   250         ; One clock cycle: inside loop: 4 cycles * 250 = 1000 cycles
Delay3:
    DEC     dly3                ; One clock cycle
    NOP                         ; One clock cycle
    BRNE    Delay3              ; Two clock cycles when jumping to Delay3, 1 clock when continuing to DEC

    DEC     dly2                ; One clock cycle
    BRNE    Delay2              ; Two clock cycles when jumping to Delay2, 1 clock when continuing to DEC

    DEC     dly1                ; One clock Cycle
    BRNE    Delay1              ; Two clock cycles when jumping to Delay1, 1 clock when continuing to RET
RET                             ; Return to main (end of subprogram)
```

Lab 8 Circuit 2: Port B (PB0) and Port D (PD2-PD7) will be used in Lab 8 to connect a 7-segment display to the Arduino Nano. Note that 7 of the 11 free digital pins will be used.



Common-anode 7-segment display (see data sheet for pinout)

Lab 8 Program 2: Write an assembly language program for the Arduino Nano that will display the 7 digits of your student ID in order (and repeat). Each digit should be lit for 1 second. Assume that Circuit 2 above will be used and that a common-anode display will be used.

Before this program can be written it is necessary to determine the PortB and PortD values that must be sent to the Arduino Nano to light each segment. Table 2 below can be used to determine these values. One example has been provided below. In order to light the digit 4, segments b,c,f and g should be ON (0). Segments a, d, and e should be OFF (1). You will need to complete the rest of the table as part of the Preliminary Work for this lab.

Table 2: Output Port – 7-segment Display Table

Note that 0 is used for any unused output port bits below.

	Segment (output)							PortB	PortD	PortB	PortD
Digit	a PB0	b PD7	c PD6	d PD5	e PD4	f PD3	g PD2	(binary) 0b0000000?	(binary) 0b000000??	(hex) 0x??	(hex) 0x??
0											
1											
2											
3											
4	1	0	0	1	1	0	0	0b00000001	0b00110000	0x01	0x30
5											
6											
7											
8											
9											

Possible Outline for Program 2

- Detailed comments
- Define PortB and PortD as all outputs.
- Add sections (similar to the code below for digit 4) for each digit in Student ID
- Add Delay_500ms subprogram
- Loop back to first digit

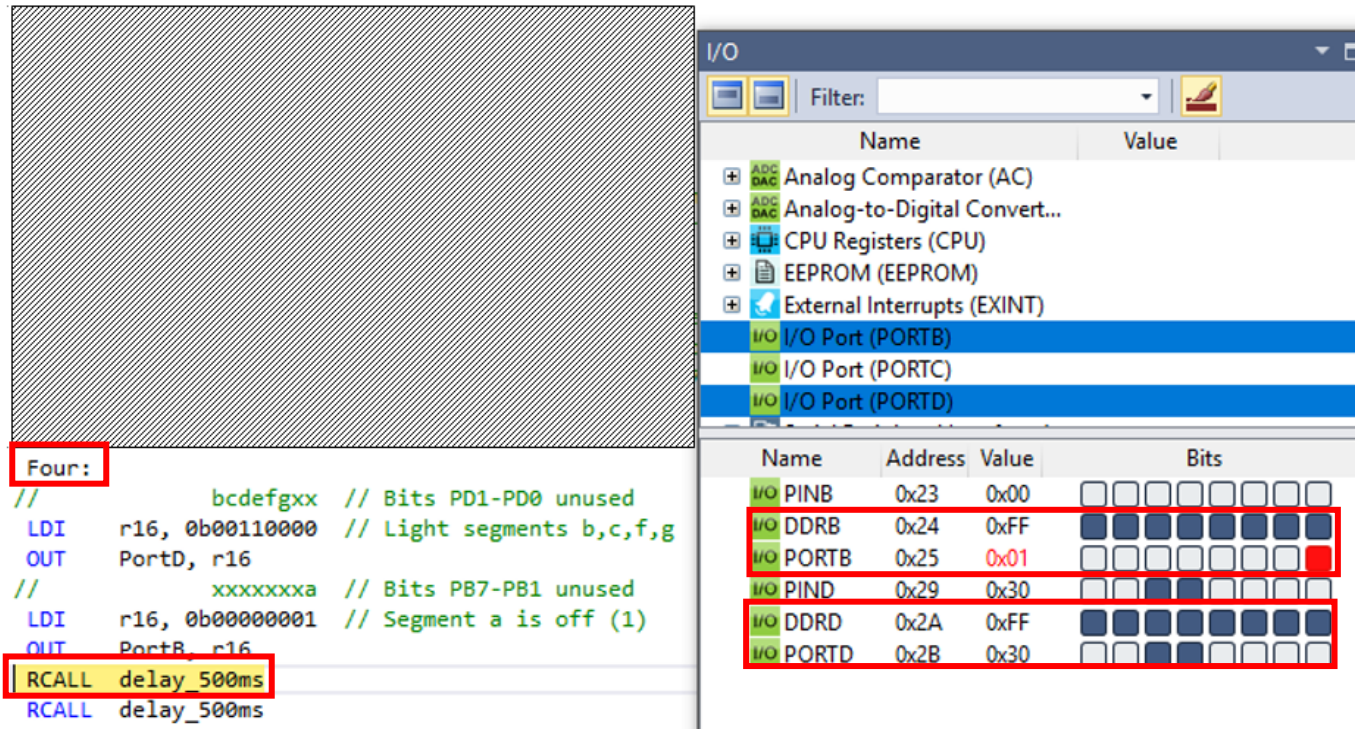
```
Four:
//              bcdefgxx  // Bits PD1-PD0 unused
 LDI    r16, 0b00110000  // Light segments b,c,f,g
 OUT    PortD, r16
//              xxxxxxxa  // Bits PB7-PB1 unused
 LDI    r16, 0b00000001  // Segment a is off (1)
 OUT    PortB, r16
 RCALL  delay_500ms
 RCALL  delay_500ms
```

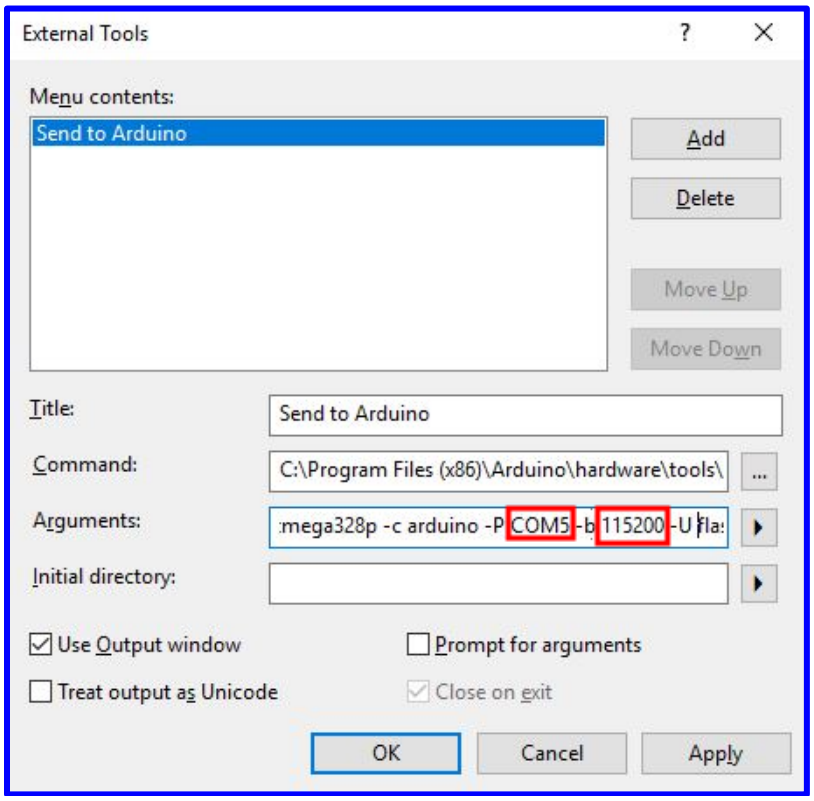
E. Preliminary Work

1. **Circuit #1**: Include a copy of Circuit #1.
2. **Program #1**: Download Program #1 from the course website. Include a copy of Program #1.
3. **Circuit #2:** Include a copy of Circuit #2.
4. **State Diagram for Program #2**: List the Student ID being used and include a state diagram for Program #2.
5. **Pinout**: Include a pinout for the 7-segment display to be used in lab.
6. **Table 2: Output Port – 7-segment Display Table:** Include a completed Table 2.
7. **Program #2**: Include a printout of Program #2. The program should contain an initial block of comments with name, course, lab number, date, and a description of the program. Additionally, include comments throughout the program.
8. **Simulation for Program #2**: Use Atmel Studio 7 to simulate Program #2. In particular,
   - After you begin ***Debug – Start Debugging and Break***, use ***Debug – Windows – I/O*** and then open windows to display I/O Port (PORTB) and I/O Port (PORTD). (Use Ctrl key to select multiple ports.)
   - After the instructions to load PortB and PortD have been executed for each digit, take a screen shot showing the code (including your name) and the PortB and PortD windows. Verify that values for PortB and PortD are correct. Include screen shots for at least 3 different digits in your report.
   - An example simulation for digit 4 is shown on the following page.

Simulation for Digit 4: The simulation below verifies that PortB = 0b00000001 = 0x01 and PortD = 00110000 = 0x30 as shown in Table 2. The simulation also shows that DDRB = DDRD = 0b11111111 = 0xFF (all outputs).

```
; Name:  (your name)
; EGR 270 Lab #8
; Date: (list date of lab)
; Description:  Program to display the following student ID (xxxxxxx) on a common-anode
```





F. Laboratory Work

1. **Program#1/Circuit #1**
   A. Construct Circuit #1 and connect the Arduino Nano to the computer using the USB cable.
   B. Download Program #1. Open the project with Atmel Studio 7 and build the program. Correct any errors.
   C. If any changes were made to Circuit #1 or Program #1 as shown in the Preliminary Work, record the changes and the reason for the changes.
   D. Use ***Tools – Send to Arduino*** to send the hex file to the Arduino Nano. You may need to press the reset button on the Nano to start the program.
   Note: Before doing this you may need to change some settings under Tools - External Tools (see the window on the previous page)
      - Change the baud rate to 57600 for the Arduino Nano (the Arduino UNO uses a rate of 115200).
      - Change the COM port number to the one assigned when you connected the Arduino Nano to the computer using the USB cable. To see what COM port was assigned, search in the Windows toolbar for Device Manager. Open Device Manager and look under Ports.)

   E. Count the number of times that the LED blinks in 60 seconds. Record the value and include it in your report. Calculate the % error from the expected value of 60 blinks in 60 seconds.
   F. Demonstrate proper circuit operation to the instructor.

2. **Program#1/Circuit #2**
   A. Construct Circuit #2 and connect the Arduino Nano to the computer using the USB cable.
   B. Use Atmel Studio 7 to create a project and enter your code (if not already done in Preliminary Work). Build the project. Correct any errors.
   C. If any changes were made to Circuit #2 or Program #2 as shown in the Preliminary Work, record the changes and the reason for the changes.
   D. Use ***Tools – Send to Arduino*** to send the hex file to the Arduino Nano. You may need to press the reset button on the Nano to start the program.
   E. Verify that the student ID is correctly displayed and that it repeats.
   F. What happens if you press the reset button on the Nano while the program is running? Record the result.
   G. Demonstrate proper circuit operation to the instructor.

F. Report

Remember that each lab report should have the following four sections. Also see additional notes below.

Title Page

Preliminary Work (include instructions)

Lab Results

- Include all measured results.
- Include step numbers and titles or headings that make it clear what is being shown.

Discussion/Conclusion

- Discuss each circuit/program tested in lab.
- Compare the implementation of a sequential circuit:
  - Using JK flip-flops (Lab 6)
  - Using VHDL and FPGAs (Lab 7)
  - Using a microcontroller and assembly language (Lab 8)
- Compare the simulation of each sequential circuit above:
  - Lab 6 – Simulation using PSPICE
  - Lab 7 – Simulation using Aldec Active-HDL
  - Lab 8 – Simulation using Atmel Studio 7