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

C. **Reference**

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

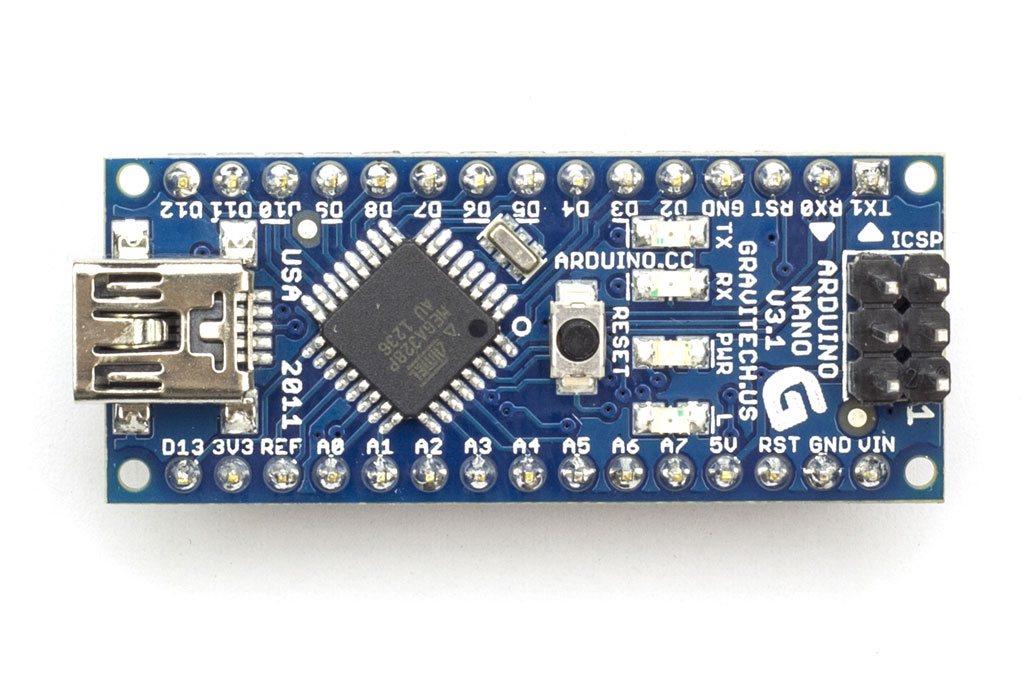
* ***Atmel Studio Tutorial (PowerPoint)***
* ***Computer Architecture, Microprocessors and Assembly Language (PowerPoint)***

D. **Introduction**

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

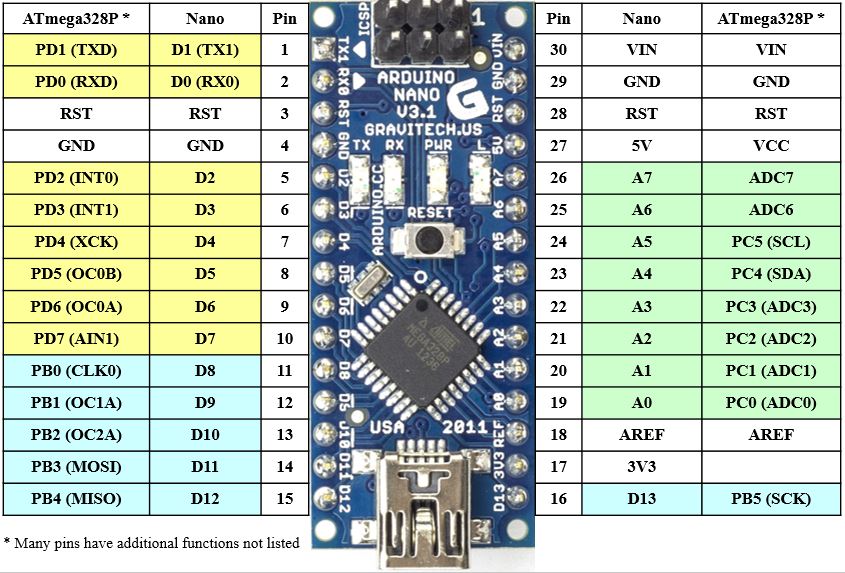
In this lab we will be using AVR Assembly Language to program the ATmega328P microcontroller on the Arduino Nano. The pin numbers on the Arduino are different from the pin numbers on the ATmega328P, so a ***Pin Mapping Table*** is shown below.

***ATmega328P***



***Arduino Nano***

**Table 1: Arduino Nano – Atmega328P Pin Mapping Table**



**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***

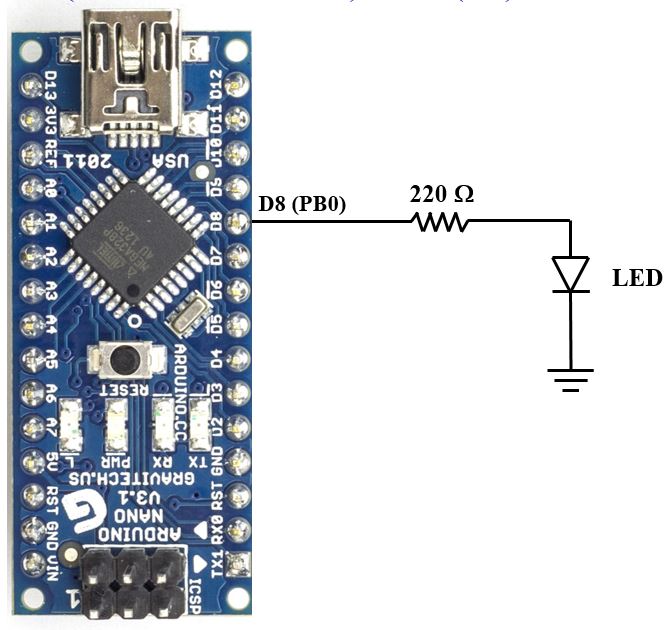
**Lab 8 Circuits and Programs**

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

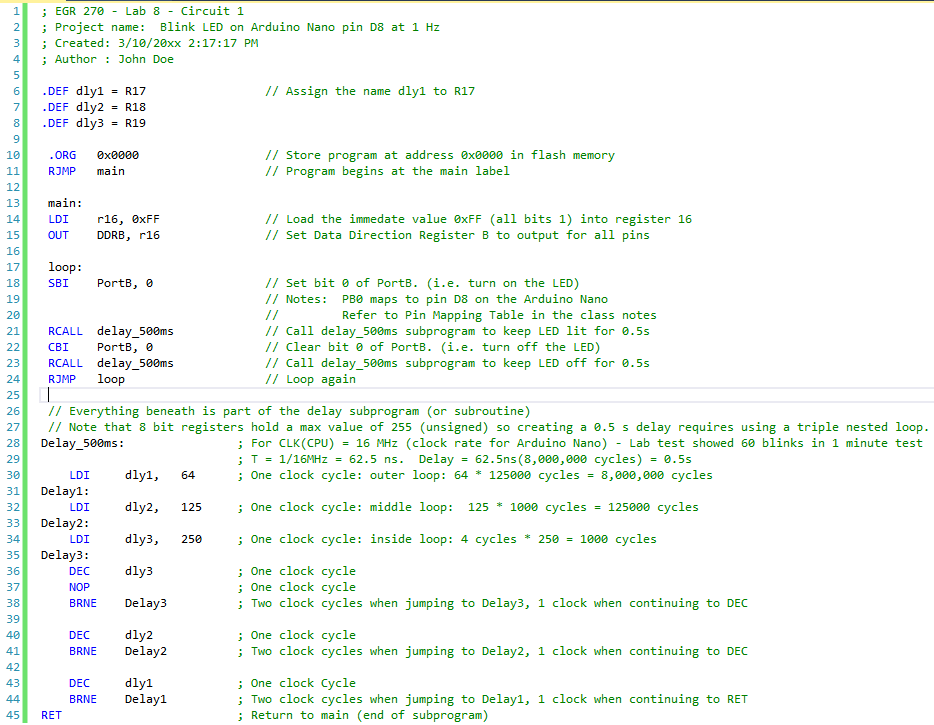
These circuits and programs will be discussed on the following pages.

**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.

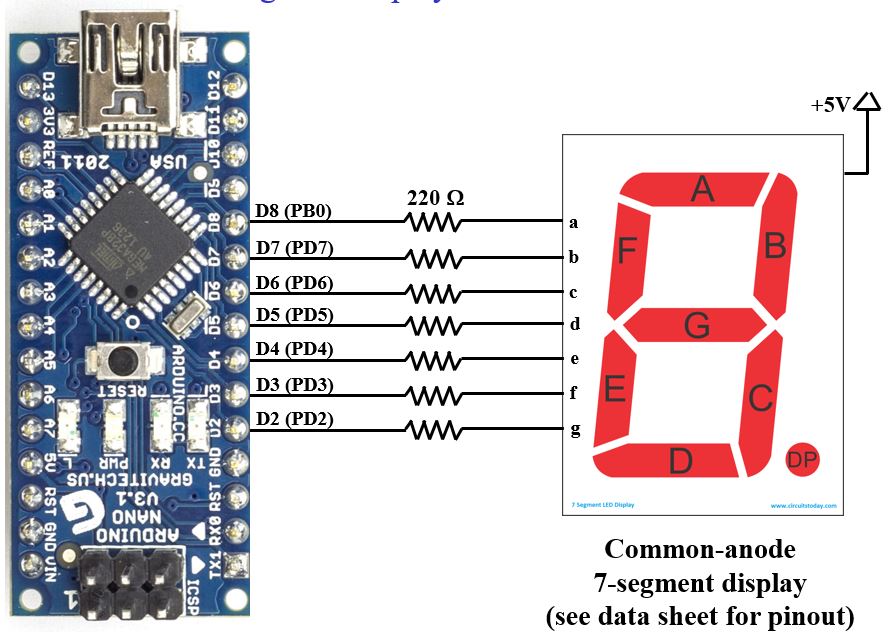


**Lab 8 Program 1:**  This program has already written and is discussed in ***Atmel Studio Tutorial***

Download this program (Blink LED on Arduino Nano pin D8 at 1 Hz) from the course website.



**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.



**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.

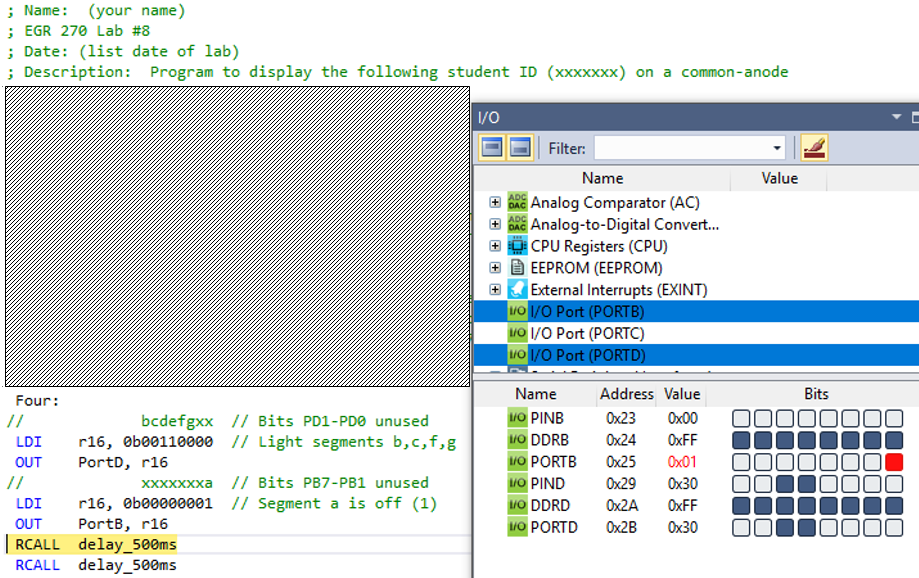
6. **Table 2: Output Port – 7-segment Display Table:** Include a completed Table 2.

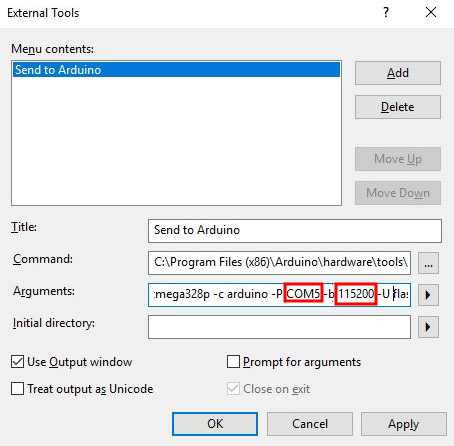
7. **Program #2**: Include a printout of Program #2. The program should contain an initial block of comments with name, course, lab number, date, and a description of the program. Additionally, include comments throughout the program.

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).





**F. Laboratory Work**

1. **Program#1/Circuit #1**

1. Construct Circuit #1 and connect the Arduino Nano to the computer using the USB cable.
2. Download Program #1. Open the project with Atmel Studio 7 and build the program. Correct any errors.
3. 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.
4. 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.)

1. 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.
2. Demonstrate proper circuit operation to the instructor.

2. **Program#1/Circuit #2**

1. Construct Circuit #2 and connect the Arduino Nano to the computer using the USB cable.
2. 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.
3. 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.
4. 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.
5. Verify that the student ID is correctly displayed and that it repeats.
6. What happens if you press the reset button on the Nano while the program is running? Record the result.
7. 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