

Topic - Digital Loops & Digital I/O

Objectives

- Write assembly code in AVR Studio 5.
- Build and download a hex file to an AVR microcontroller.
- Calculate execution time for an assembly code.

Preparation

- Review the tutorials on the AVR microcontroller digital I/O ports on [AVR-Tutorials.com](http://www.avr-tutorials.com/digital/about-avr-8-bit-microcontrollers-digital-io-ports):
<http://www.avr-tutorials.com/digital/about-avr-8-bit-microcontrollers-digital-io-ports>
<http://www.avr-tutorials.com/digital/digital-input-output-assembly-programming-atmel-8-bits-avr-microcontrollers>
- Review the tutorials on calculating execution time for assembly code on [AVR-Tutorials.com](http://www.avr-tutorials.com/assembly/calculating-execution-time-sequential-code):
<http://www.avr-tutorials.com/assembly/calculating-execution-time-sequential-code>
<http://www.avr-tutorials.com/assembly/calculating-execution-time-code-single-loop>

Laboratory Procedures

1. Obtain a breadboard and the necessary components and construct the circuit in **Figure 1**.

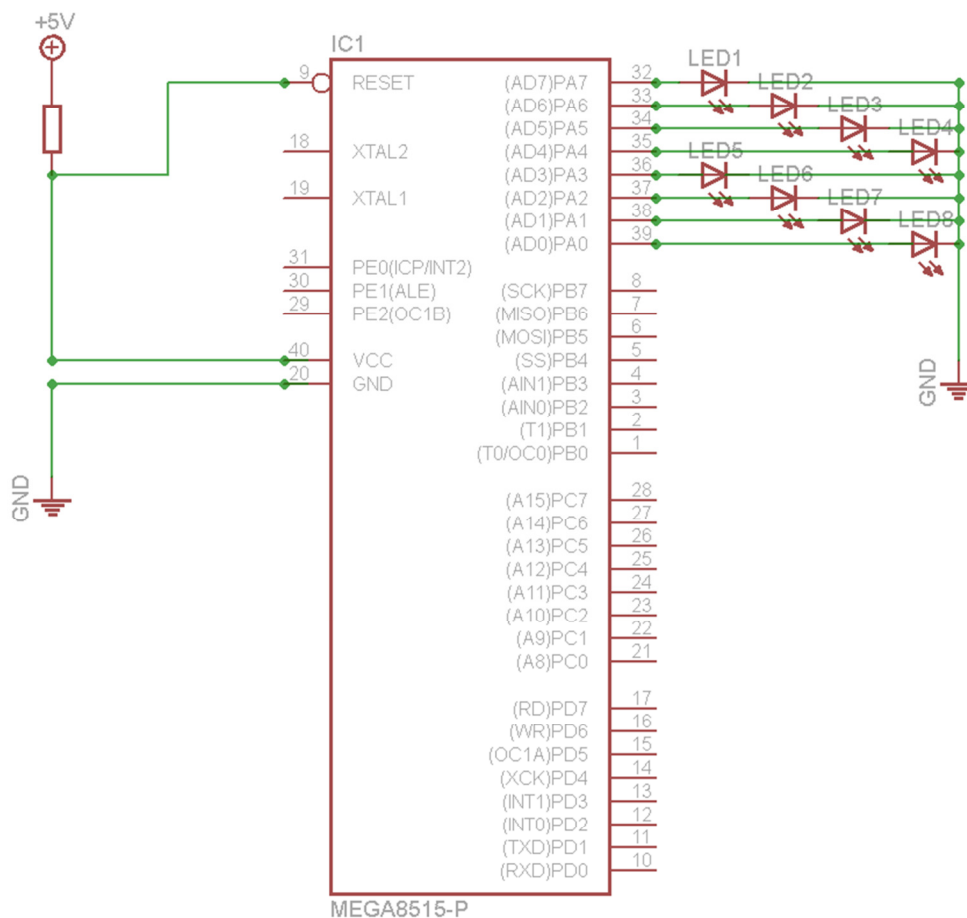


Figure 1

2. Analyze the code in **Figure 2**. How would the LEDs behave if this code was downloaded to and running on the microcontroller?
-
-
-

```

.include      "m8515def.inc"

                LDI    R16, 0xFF
                OUT    DDRC, R16

loop:           LDI    R16, 0x00
                OUT    PORTC, R16

                LDI    R16, 0xFF
                OUT    PORTC, R16

                RJMP   loop
    
```

Figure 2

3. Start AVR studio 5. Create a new assembly project selecting the ATmega8515 as the microcontroller for this project.
4. Type the code given in **Figure 2**, generate the hex file and download it to the microcontroller in the circuit from part 1. Ensure that the microcontroller is program to use its internal oscillator you can set it to 4MHz for this experiment.
5. Power-up your circuit and describe the behavior of the LEDs.
-
-
6. Does your expected observation from part 2 differ from that of part 5? Yes No
7. If you answer for part 6 is yes give an explanation for the difference.
-
-
8. Now modify your code to reflect the changes in **Figure 3**. Generate the new hex file and download it to the microcontroller.
9. Power-up your circuit and describe the behavior of the LEDs.
-
-

10. Give an explanation why the LEDs blink using one code and does not using the other code.

.....

.....

.....

.....

```
.include      "m8515def.inc"

              LDI   R16, 0xFF
              OUT   DDRC, R16

loop:         LDI   R16, 0x00
              OUT   PORTC, R16

              LDI   R17, 0xFF
dly1:         LDI   R18, 0xFF
loop1:        DEC   R18
              BRNE loop1
              DEC   R17
              BRNE dly1

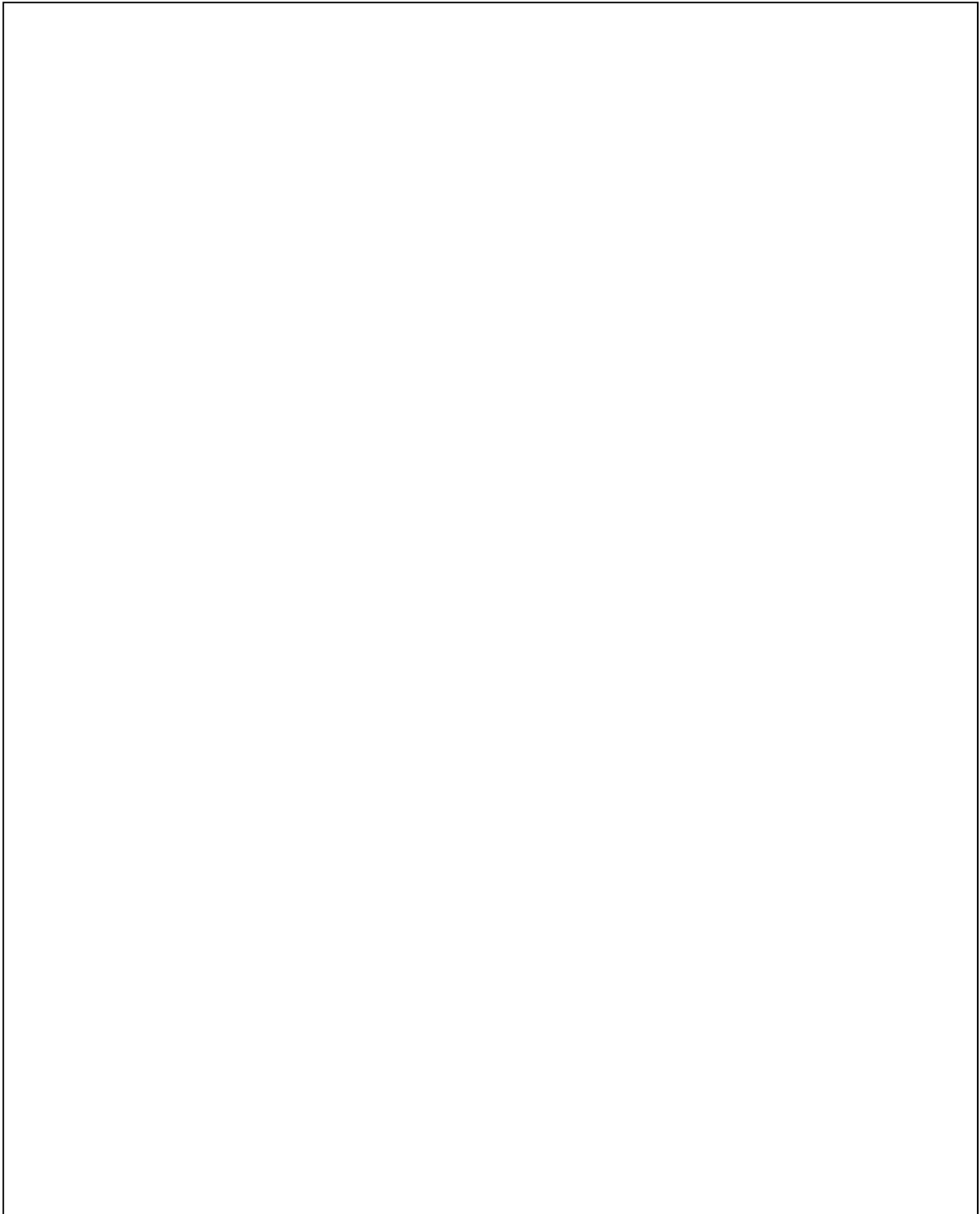
              LDI   R16, 0xFF
              OUT   PORTC, R16

              LDI   R17, 0xFF
dly2:         LDI   R18, 0xFF
loop2:        DEC   R18
              BRNE loop2
              DEC   R17
              BRNE dly2

              RJMP  loop
```

Figure 3

11. Calculate the execution time for the delay loop **dly1**. (Note: include the load instruction for R17).



12. Design and write an assembly code in the space provided below that if downloaded to the microcontroller in Figure 1 will simulate a lit LED running back and forth indefinitely.

--	--

13. Verify that your code works by typing the code in AVR Studio 5. Generate the hex file and download it to the microcontroller.
14. Ensure that you demonstrate the working of your code to the Lecturer or the Lab Personnel and obtain their verification signature.

Verification: _____

Code Worked: Yes No

Signed _____
Lecturer or Demonstrator

Disclaimer & Copyright Information

The content of this document is owned and copyrighted © by **AVR Tutorials @ AVR-Tutorials.com**.

This document including the codes or other content may not be reproduced in any way, shape, or form without the express written permission of AVR Tutorials.

Any website may link to this document on AVR Tutorials, but no page or documents from AVR Tutorials may be placed or embedded in a frame on an external website without the express written permission of AVR Tutorials.

AVR Tutorials is provided with no warranties of any kind. The information contained in documents and pages from the website is for general information purposes only. The information is provided by AVR Tutorials and while we endeavour to keep the information up to date and correct, we make no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability or availability with respect to the website or the information, products, services, or related graphics contained on the website for any purpose. Any reliance you place on such information is therefore strictly at your own risk.

In no event will we be liable for any loss or damage including but not limited to, indirect or consequential loss or damage, or any loss or damage whatsoever arising from loss of data or profits arising out of, or in connection with, the use of this document or information from the website.

Through this website you may be able to link to other websites which are not under the control of AVR-Tutorials.com. We have no control over the nature, content and availability of those sites. The inclusion of any links does not necessarily imply a recommendation or endorsement of the views expressed within them.

Your continued use of this document and the website will be deemed as implicit acceptance of the terms and conditions of this disclaimer in its entirety.