CPE 301L
Microcontroller Based System Design
Laboratory Exercise #3

Atmel Studio Tutorial Part 2

Department of Electrical and Computer Engineering
University of Nevada, at Las Vegas

Goals:

Understand the basic functions the compiling and debugging in AVR Studio 5.
Understand the basic functions of the AVRISP mkII.

Equipment Usage:

For this lab the following equipment will be used:

- Atmel Studio 6
- AVR RISP mkII
- ATmega328P

Background:

User Guide for AVRISP mkII:

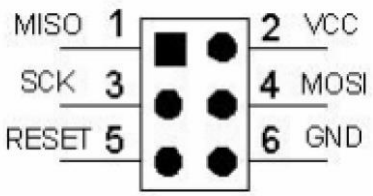
http://people.ece.cornell.edu/land/courses/ece4760/AtmelStuff/avrispmkii_ug.pdf

or

http://www.atmel.no/webdoc/index.html (select user guide under AVR ISP mkII section)

Connecting to the AVRISP mkII:

For the majority of these labs we will be using the AVRISP mkII. The figure below displays the required connections needed for the ISP.



Using the ATmega328P, the same ports must be attached to the ISP. Failure to connect to the proper ports will result in an error displayed when the AVRISP mkII attempts to read the attached device. You will have to refer to the datasheet of each IC to see the location of each port (SCK, MISO, MOSI) since they will be different for each chip

Assembly Code:

Assembly is a low-level programming language that allows users to generate expressions that are equivalent to machine langue instructions. Because assembly is essentially machine language, it is often used instead of high-level programming languages (such as C, C++, Java, etch) when programming devices such as microcontrollers and microprocessors. Simple instructions such as c = a+b are much longer in assembly code. The benefit of this is that assembly language provides a programmer more control over the instructions.

Basic ASM Functions

Mnemonics	Operands		Description	Operation
ARITHMETIC AND LOGIC INSTRUCTIONS				
ADD	Rd, Rr		Add two Registers	Rd ← Rd + Rr
SUB	Rd, Rr		Subtract two Registers	Rd ← Rd - Rr
AND	Rd, Rr		Logical AND Registers	Rd ← Rd • Rr
OR	Rd, Rr		Logical OR Registers	Rd ← Rd v Rr
NEG	Rd		Two's Complement	Rd ← 0x00 − Rd
INC	Rd		Increment	Rd ← Rd + 1
DEC	Rd		Decrement	Rd ← Rd − 1
BRANCH INSTRUCTIONS				
RJMP	k		Relative Jump	PC ← PC + k + 1
RCALL	k		Relative Subroutine Call	PC ← PC + k + 1
RET			Subroutine Return	PC ← STACK
BREQ	k		Branch if Equal	if (Z = 1) then PC ← PC + k + 1
BRNE	k		Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1
DATA TRANSFER INSTRUCTIONS				
MOV	Rd, Rr	Move Between Registers		Rd ← Rr
LDI	Rd, K	Load Immediate		Rd ← K
ST	X, Rr	Store Indirect		(X) ← Rr
IN	Rd, P	In Port		Rd ← P
OUT	P, Rr	Out Port		P ← Rr
PUSH	Rr	Push Register on Stack		STACK ← Rr
POP	Rd	Pop Register from Stack		Rd ← STACK

Prelab:

Watch the videos explaining how to create and simulate programs using Atmel Studio

- Create a New C Project for GCC in Atmel Studio 6:
  http://www.atmel.com/System/Overlay/Video.aspx?uri=tcm:26-39807
- Build a Project in Atmel Studio 6:
  http://www.atmel.com/System/Overlay/Video.aspx?uri=tcm:26-39808
- Using the Simulator in Atmel Studio 6:
  http://www.atmel.com/System/Overlay/Video.aspx?uri=tcm:26-39810

Lab Experiments:

Setup: to setup the circuit for this lab you should program your chip first while it is still connected to its development board. Once the IC is programmed, remove it and place it on a breadboard and set all VCC (also AVCC for 328P) connections to 5V and all GND connections to 0V.

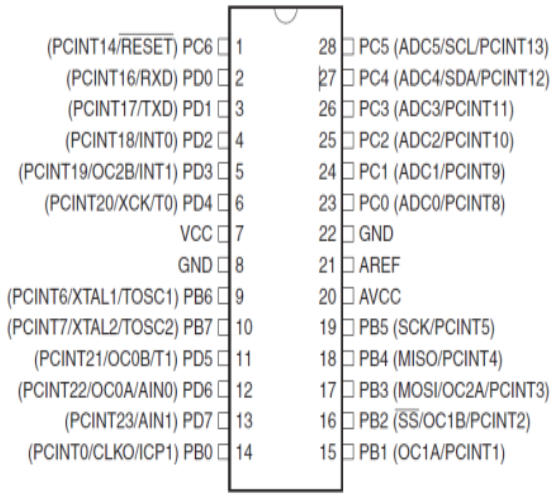


Figure 3.1: Microcontroller pin configuration

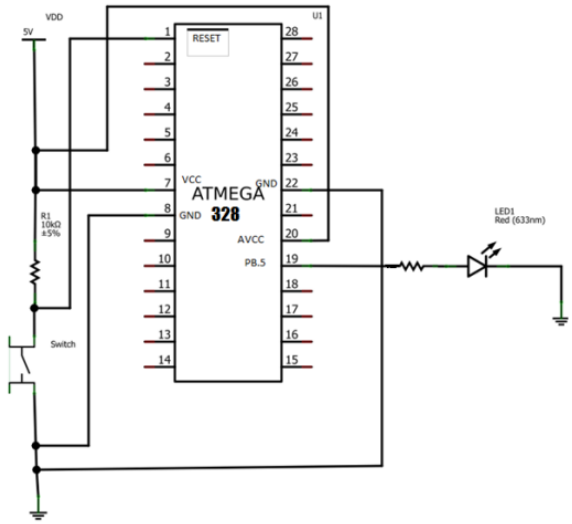


Figure 3.2 Circuit Example

Experiment 1: Program the following assembly code using Atmel Studio 6 into your Atmega 328P. The code should light up an LED connected to pin (PB.4)

```
        .org 0
        SBI         DDRB,4          ;set PB4 as an output
        LDI         R17,16          ;value is 16 to light up bit 4
        OUT         PORTB,R17       ;sends value of R17 to corresponding bit
```

Experiment 2:

Program the following C code using AVR studio with your Atmega 328P

```
#include <avr/io.h>

int main ()
{

       DDRB = 0xFF;          //set all of portB to output
    while (1)
       PORTB = 0x88;         //set PB7 and PB3 high
    return 0;
}
```

Experiment 3:

Modify the given assembly code and program the ATmega328P to light up the LEDs connected to pin PB5 and PB3 simultaneously. Modify the given C code for the ATmega328P light LEDs connected to pins PB6, PB4, PB2 simultaneously.

Post-Lab Deliverables:

1) Copy of your modified code
2) Questions:
    a. What does the red stripe of the ISP cable indicate?
    b. List and explain the different color codes of the AVRISP mkII
    c. At what frequency should the programmer be set at to read/write to the device?
    d. Write code in assembly that will perform the following equation: 100+53-27
    e. Explain what is happening on each line of the following code. If you were to execute this code what would be the final decimal value stored in R20?

```
        LDI     R20, 0x75
        LDI     R21, 0x05
        LDI     R22, 0x24
        ADD     R20, R22
        SUB     R22, R21
        ADD     R20, R21
        SUB     R22, R20
        ADD     R20, R22
        MOV     R20, R21
        RJMP    DONE
        ADD     R21, R20
        SUB     R21, R22
DONE:   SUB     R20, R21
END:    RJMP    END
```