Laboratory 6 ***AVR Timers***

Instructor: Dr. Lihong Zhang

1 Objectives

With the completion of this lab, you should be able to:

- Work with different timers of AVR microcontroller.
- Use AVR timers to make delay in program or as a counter to count waveforms.

2 Introduction

The feature of timers is useful for generating periodic signals, including square waves and rate waves. Other features include the ways to generate a consistently timed interrupt requests, to generate clock signals for other system devices (such as for serial communications), to make Pulse-Width-Modulation (PWM) waves, and to count wave cycles as a counter. In this lab, you will work with AVR microcontroller timers. You will make a square wave with a programmable frequency to blink a LED. And you will also write an assembly language program to count the number of pulses by using timer/counter.

3 AVR Microcontroller Timer

AVR ATmega32 microcontroller has 3 general-purpose timers with a variety of features. In this lab you will only use **Timer0** and **Timer1**.

3.1 Short Delays

In this part of lab, you have to write an assembly language program which is to blink an LED using **Timer0**. Your program should make a square wave on PORTD.0 in the frequency of as close as 5Hz and connect PORTD.0 to one of the built-in LEDs. The easiest way is to connect PORTD and LEDS headers on the STK600 board so that LED0 is to blink. Demonstrate your work to a TA who will sign on your program if she or he is satisfied.

Hints: (1) You may want to put your instructions related to Timer0 in a delay subroutine. (2) You should configure PORTD.0 as an output pin. (3) By default, STK600 AVR microcontroller uses 1MHz clock as $Clk_{I/O}$ (produced by AVR internal calibrated RC oscillator, which will be discussed later in the term).

Question1: Explain how to configure **Timer0** to fulfill the task above with proper initial numbers. Justify your calculation by providing the information such as clock frequency, working mode, prescaler, etc.

Then you should modify your program to output a square wave to PORTA.0 in the frequency of 1KHz. Use an oscilloscope to verify the output wave and demonstrate its correctness to a TA, who will sign on your program. To measure

the signal of PORTA.0, you should use a 10-wire cable (the one with a DIP connector on one end) to bring PORTA of the STK600 to a breadboard. Then use a wire to measure the signal from the breadboard.

Question2: Explain how to configure **Timer0** to fulfill the task above with proper initial numbers. Justify your calculation by providing the information such as clock frequency, working mode, prescaler, etc.

3.2 Long Delay

In the second part of the lab, you will write an assembly language program which is to blink an LED using **Timer1**. Your program should make a square wave on PORTD.0 in the frequency of as close as 1/4 Hz and connect PORTD.0 to one of the built-in LEDs. The easiest way is to connect PORTD and LEDS headers on the STK600 board so that LED0 is to blink. Demonstrate your work to a TA who will sign on your program if she or he is satisfied.

Hints: (1) You may want to put your instructions related to Timer1 in a delay subroutine. (2) You should configure PORTD.0 as an output pin.

Question3: Explain how to configure **Timer1** to fulfill the task above with proper initial numbers. Justify your calculation by providing the information such as clock frequency, working mode, prescaler, etc.

3.3 Counting Pulses

Note: In this part of the lab, it is required that PORTD and LEDS headers on the SKT600 should be connected by a 10-wire cable.

You will write an assembly language program that counts the number of pulses. Pulses are produced from key pressing. You should connect one of the built-in push-button switches to the input of **Timer0**. By reading ATmega32 datasheet, you can find T0, the input of Timer0, is actually PB0. Therefore, the most convenient way is to connect PORTB and SWITCHES headers together by using a 10-wire cable. And SW0 is to function as a pulse generator. With every pressing, one pulse is generated. It will make **Timer0** content (i.e., TCNT0) increase by one. The binary content of **Timer0** (i.e., TCNT0) should be displayed on PORTD as well as the built-in LEDs of the STK600.

Demonstrate your work to a TA who will sign on your program if she or he is satisfied.

3.4 Return of the Rat

In laboratories 4&5, you wrote a game called *Rat-bashing*. The easiest way to generate a sequence of rats was to pre-program them in an array. Unfortunately,

such a scheme means that the sequence is the same every time the program is run.

One way to create a seemingly random sequence would be to use a counter on an AVR timer to cycle through a sequence of values; the program could then read the count value to generate a rat in a pseudo-random location.

Explain in more detail how the method described above would work. Briefly describe the changes that your *Rat-bashing* program would require to incorporate such a feature. However, you don't need to program for this task.

4 Submission

At the end of the lab, submit print-outs of your *.ASM files*. Include answers to all the questions and test results in your lab report.

Before leaving the lab, deconstruct your circuit and return your components to the provided kit.