The Hong Kong Polytechnic University

Department of Electrical and Electronic Engineering

EIE3105 Integrated Project

Laboratory Exercise: PWM and Input Capture

(Deadline: Check the course information)

Objective:

To develop C programs to generate PWM pulses and capture the input using an AVR and ARM microcontrollers.

Equipment:

Atmel Studio 7 (software)
Atmega328p (hardware)
STMClubIDE (software)
STM32F103RBT6 (hardware)

Procedure:

Section A: Generate a wave (PWM pulses) by using PWM (Pulse Width Modulation) in an Atmega328p (AVR) microcontroller.

Write a C program to generate a wave using PWM mode. Use Timer 0 in Fast PWM mode, and set the wave frequency to 500 Hz. The duty cycle should be 50%. You may connect an LED to the pin that generates the wave. You can adjust the duty cycle to control the LED's brightness. Note that you will not see the LED flash because the frequency is too high.

Section B: Generate a wave (PWM pulses) from the AVR microcontroller and capture it back to measure its pulse width.

You are required to write a C program that has two parts.

1. It gets an integer from Tera Term and sets it as the pulse width (in clock cycles) of a wave generated by the AVR microcontroller (see Figure 1). The frequency of the wave should be 500 Hz. The received integer should be echoed in Tera Term. You can assume the integer is between 10 and 99.

2. It gets the wave from its ICP1 pin and measures the pulse width of the wave. Then it sends the measured width value to Tera Term again.

If your application executes properly, the pulse width setting should be equal to or close to the measured pulse width.

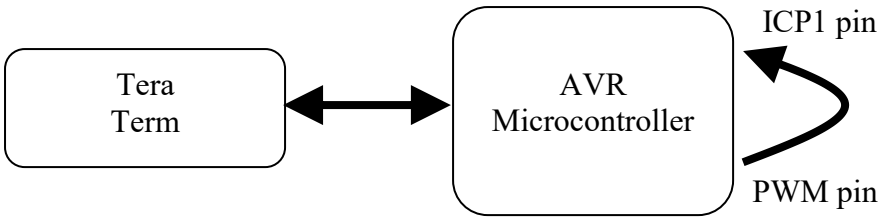

Figure 1

Section C: Generate a wave (PWM pulses) from the ARM microcontroller and capture it back to measure its pulse width.

Repeat Section B, but this time, you use the ARM microcontroller. You can use the program code in the lecture notes. The frequency of the wave should be 500 Hz. The received integer should be echoed in Tera Term. You can assume the integer is between 10 and 99.

Section D: Generate a wave (PWM pulses) from the AVR microcontroller and capture it from the ARM microcontroller to measure its pulse width.

Repeat Section B, but you use the ARM microcontroller to capture the wave this time.

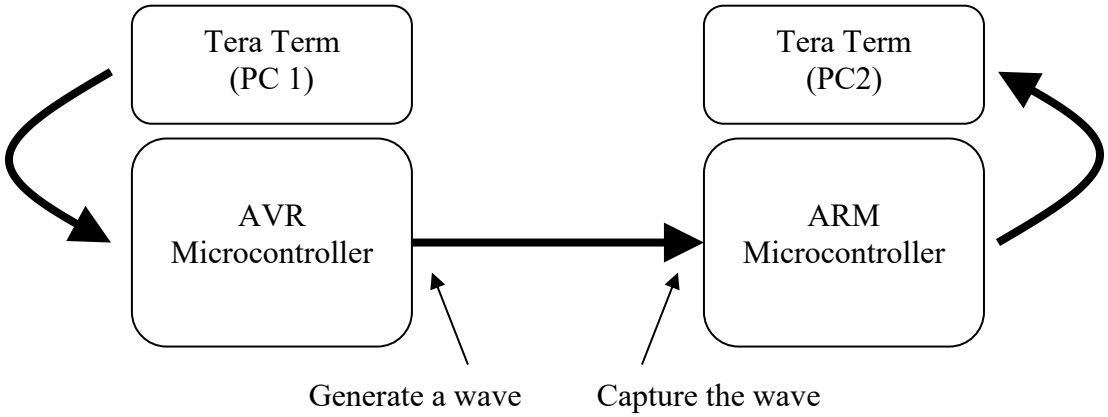

Figure 2

Section E: Generate a wave (PWM pulses) from the ARM microcontroller and capture it from the AVR microcontroller to measure its pulse width.

Repeat Section E, but this time the ARM microcontroller generates a wave, and the AVR microcontroller captures it and measures its pulse width.

Instructions:

1. You are required to make demonstration videos to demonstrate Sections A to E.
2. Zip all programs (including the whole projects) in Sections A to E and your demonstration videos into a single file and submit it to Blackboard.
3. Deadline: **Check the course information**.

Lawrence Cheung
January 2026