THE HONG KONG POLYTECHNIC UNIVERSITY

Department of Electronic and Information Engineering

EIE3105 Integrated Project (Part I)

Laboratory Exercise 5: Interfacing

(Deadline: Check the course information)

Objective:

To develop C programs to cooperate with different external devices under the Arduino platform.

Equipment:

Atmel Studio 6
The Arduino Starter Kit

Procedure:

Section A: Send pulses (Generate a wave) by using PWM (Pulse Width Modulation)

Write a C program to send pulses (generate a wave) by using PWM mode. You should use Timer 0 in Fast PWM mode and the frequency of the generated wave should be 500 Hz. The duty cycle of the wave should be 50%. You may connect a LED to the pin which generates the wave. You can change the duty cycle to control the brightness of the LED.

Section B: Capture a wave from an AVR microcontroller and measure its pulse width

You are required to write two C programs.

1. Write a C program in AVR microcontroller 1 to get an integer from PC terminal 1 (e.g., Tera Term) and set it as the pulse width (in clock cycles) of a wave generated by AVR microcontroller 1 (see Figure 1). The frequency of the wave should be 500 Hz. The received integer should be displayed on PC terminal 1. It means, after you receive the integer from the keyboard through PC terminal 1, you should send the integer to PC terminal 1 so that is can be displayed on the screen. You can assume that the integer must be between 10 and 99.

2. Then connect AVR microcontroller 1 to AVR microcontroller 2 so that AVR microcontroller 2 can capture the wave generated by AVR microcontroller 1.

3. Write a C program in AVR microcontroller 2 to measure the pulse width of the wave generated by AVR microcontroller 1. Then send the pulse width (in clock cycles) to PC terminal 2 (i.e., Tera Term) to display the value of the measured width. To verify whether your application can be executed properly, the integer displayed on PC terminal 1 should be equal or very close to the integer displayed on PC terminal 2. Note that AVR Microcontroller 1 and 2 are required to share a common ground.

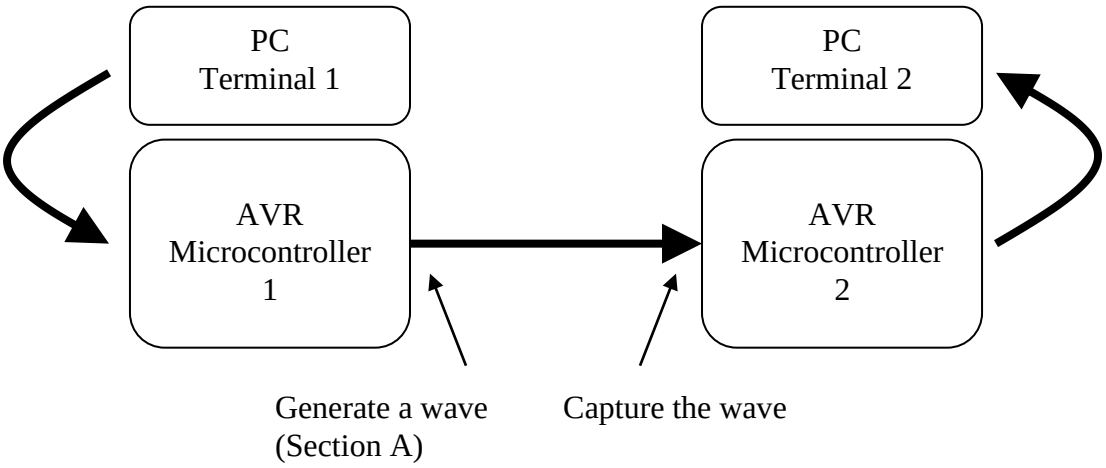

Figure 1

Note that you can test and debug your program by using your AVR only (see Figure 2). You can connect the PWM pin to ICP1 pin and use the PC terminal to display the pulse width.

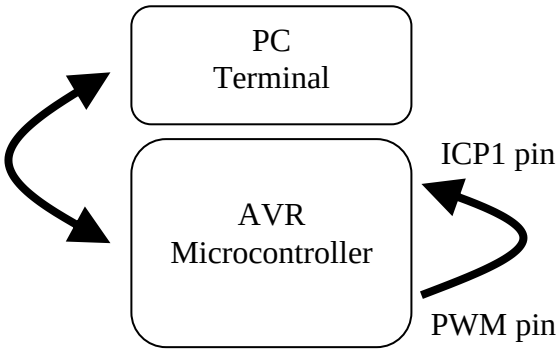

Figure 2

Section C: Capture the light intensity of photo-resistors to control a Tri-color LED

A photo-resistor (see Figure 3 and 4) is a sensor that changes its resistance depending on the amount of light that hits them. It is also known as photocells or light-dependent resistors. You should connect one end of the resistor to the Arduino ADC pin then measure the change in resistance by checking the voltage on the pin. You should write a simple program to check whether you can identify the change of the voltage of the photo-resistor or not.

A tri-color LED (see Figure 3 and 5) with 4 legs is a common cathode RGB LED. The LED has separate red, green and blue elements inside and one common ground (the cathode). The brightness of its three colors can be changed by applying 500 Hz PWMs with different duty cycles on these three color pins. You may freely choose which PWM pins to generate PWMs. Note that you should use AVR PWM pins to generate 500Hz-PWMs to the LED (i.e., you should not write a program to control the high and low of the output pin.).

The AVR microcontroller connects to the LED and three photo-resistors (see Figure 6). Write a C program to get the light intensity of those resistors through ADC (Analog-to-Digital Converter). Note that you should use an interrupt to get the data. Then use such data to

generate three waves by using PWM (use Timer0, Timer1 and Timer2 in Fast PWM mode) to control the brightness of three colors of the LED. Note that the brightness of three colors of the LED is depended on the light intensity of those resistors. It means if the light (from the surrounding) received by those resistors is bright, the brightness of three colors of the LED is high and vice versa. For simplicity, you can set a threshold the light received by those resistors. If the light received by a photo-resistor is bright (normal), you can set the duty cycle of the corresponding PWM signal as 100%; otherwise (the photo-resistor is covered by hand), yo use set the duty cycle as 0%.

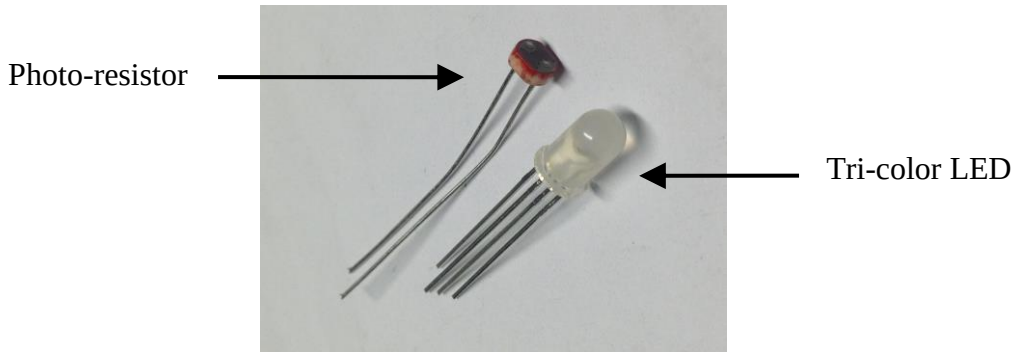

Figure 3

Figure 4

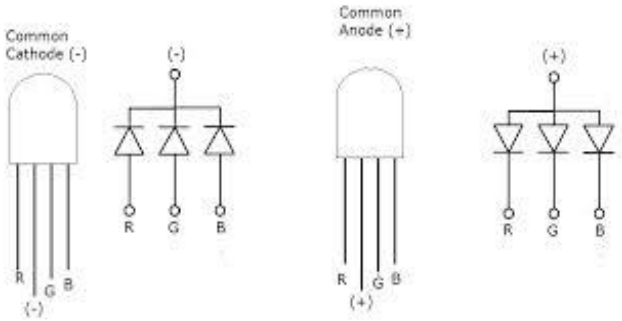

Figure 5

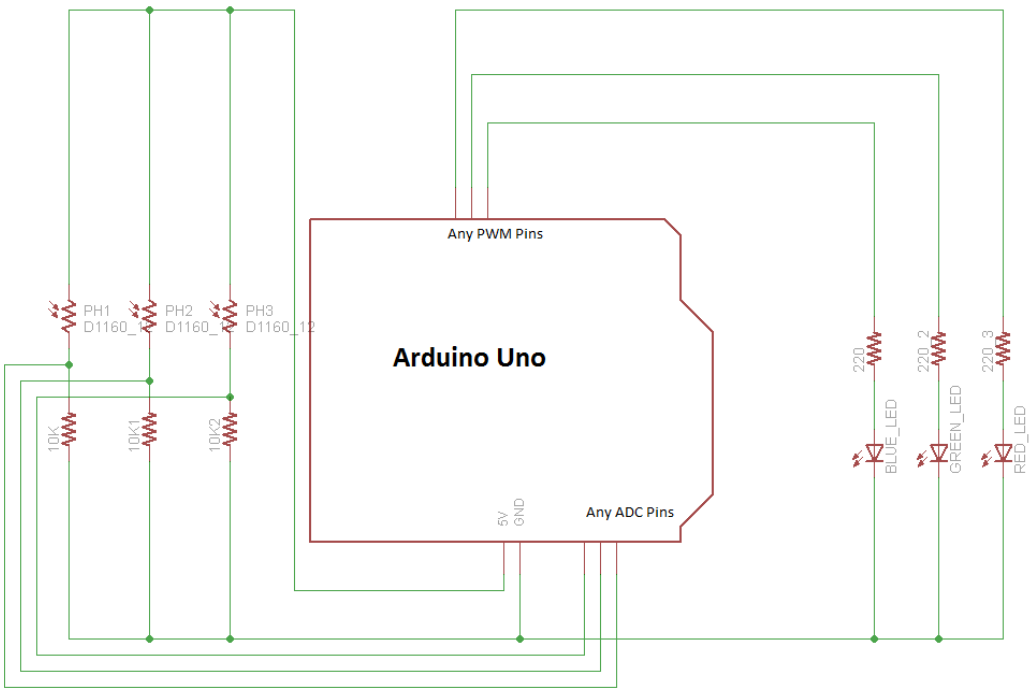

Figure 6

<u>Demonstrate your applications in Section B and C to our tutor or technicians.</u>

Instructions:

1. You are required to demonstrate your programs to our tutor or technicians.
2. It is a mini-project. You are required to submit a report for this lab. Your report should include introduction, procedures, results and analysis, discussion and conclusion.
3. Zip all programs (including the whole projects) in Section B and C, and your report (a word document) to a single file. Submit it to Blackboard.
4. Deadline: **<u>Check the course information</u>**.

Ivan Lau
Lawrence Cheung
August 2016