OBJECTIVES

- Understand general concepts regarding pulse-width modulation (PWM).
- Learn how to perform pulse-width modulation techniques with the *ATxmega128A1U*.

INTRODUCTION

Pulse-**w**idth **m**odulation, or **PWM**, is a method for describing digital signals through varying duty cycles. In general, pulse-width modulation is commonly used to mimic analog values in digital circuits. By modulating the duty cycle of a digital signal, it is possible to emulate values in between some minimum (e.g., 0 V) and some maximum (e.g., *Vcc*). Some common use cases for this technique include controlling the speed of a motor as well as controlling the intensity of an LED. Within the *ATxmega128A1U*, pulse-width modulation is available through modes within the timer/counter systems.

HOMEWORK STRUCTURE

In this homework, you will learn how to utilize a pulse-width modulation mode within an appropriate timer/counter module of the *ATxmega128A1U* to control the intensity of the LEDs within the RGB LED package located on the *OOTB µPAD*. Ultimately, light intensity will be shown to directly affect the appearance of the color emitted from RGB LEDs.

REQUIRED MATERIALS

- Atmel XMEGA AU Manual (doc8331)
- Atmel *ATxmega128A1U* Manual (doc8385)
- Timer/Counter Application Note (doc8045)
- *OOTB µPAD v2.0* with USB A/B cable and accompanying schematic
- *OOTB Memory Base* with accompanying schematic
- *OOTB Switch & LED Backpack* with accompanying schematic

SUPPLEMENTAL MATERIALS

- **D**igital **A**nalog **D**iscovery (**DAD**) kit with *WaveForms* software

HOMEWORK PROCEDURE

An RGB LED is a single package consisting of a red LED, a green LED, and a blue LED. There are active-high RGB LED packages as well as active-low RGB LED packages. Active-high RGB LEDs share a common cathode (negative terminal), while active-low RGB LEDs (shown in Figure 1) share a common anode (positive terminal).

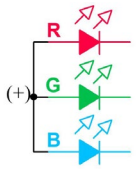

Figure 1: Active-low RGB LED

NOTE: Your *OOTB µPAD* Development Board is equipped with an active-low RGB LED, labeled ***D4*** below the female headers at location ***J3***.

In general, activating any one of the terminals of an RGB LED package will display a corresponding color. Additionally, by simply activating different combinations of terminals within an RGB package, it is possible to create the appearance of complementary colors. For example, by illuminating both red and blue LEDs at the same time, an RGB LED package should appear to emit a magenta color.

Beyond the above, it is also possible to exploit the intensity of an LED to produce an even wider variety of colors. More specifically, by turning an LED on and off rapidly, the percentage of time that the LED is powered on, or the duty cycle of the LED, can dictate the appearance of color emitted from that LED. This technique has actually led to the standard for how digital colors are defined.

In computer displays, it is common for RGB values to be representable via 24-bit values, where each of red, green, and blue colors are equally represented by 8-bits. This ultimately means that the range of intensity for each of the color values is from 0 to 255 in decimal, or $00 to $FF in hexadecimal. To control these intensities, the technique referred to as pulse-width modulation (PWM) is used.

As alluded to previously, pulse-width modulation is a technique for emulating analog voltages with digital signals. Instead of providing a continuous range of voltage with a variable power supply, PWM can be utilized to vary the average output voltage between some minimum (usually 0 V), corresponding to a 0% duty cycle, and some maximum (usually Vcc), corresponding to a 100% duty cycle.

Within the *ATxmega128A1U*, the **C**ompare or **C**apture (CC) channels within a timer/counter system can be utilized to easily perform pulse-width modulation via specific modes. Below, you will begin to learn how to utilize these modes for the purposes of interfacing with the RGB LED package located on the *OOTB µPAD*.

1.1. Carefully read through § 14 (*TC0/1 – 16-bit Timer/Counter Type 0 and 1*) in the 8331 manual, as well as any pertinent sections in the 8045 manual. Make sure that you understand how to utilize the various waveform generation modes as well as the appropriate configuration registers.

1.2. Explore the *OOTB µPAD v2.0 Schematic*. Locate the on-board RGB LED package and determine which port pins are connected to this package.

1.3. Refer to § 33.2 (*Alternate Pin Functions*) in the 8385 manual to determine which pins on the same port as above are used for the relevant timer/counter signals. Unfortunately, the relevant timer/counter signals are not mapped to the appropriate pins utilized by the RGB LED package by default, however these pins can be remapped through software. Refer to § 13 (*I/O Ports*) within the 8331 manual to learn how to remap the relevant port pins.

HOMEWORK EXERCISES

i. How many TC0 channels are necessary to control all three of the LEDs within the on-board RGB LED package?
ii. In the context of the program specified above, would any difference (theoretically) result from setting the RGB period to be $FFFF, instead of $FF?

For the remainder of this homework, you will create an assembly program to continually allow various switches to specify an 8-bit duty cycle for each of the three LEDs within the RGB LED package on the *OOTB µPAD*. The program must meet the following specifications.

- By default, no LED within the RGB LED package should be illuminated.
- Following the relevant initializations, the eight DIP switches located on the *OOTB Switch & LED Backpack* should be used to specify an 8-bit duty cycle for an LED within the RGB LED package whenever a corresponding tactile switch on either the *OOTB Switch & LED Backpack* or *OOTB Memory Base* is pressed. In particular, whenever tactile switch S1 located on the *OOTB Switch & LED Backpack* is pressed, an 8-bit duty cycle for **only** the *RED_PWM* LED should be specified by the eight DIP switches, i.e., only the red LED should be illuminated with some intensity. Similarly, whenever tactile switch S2 on the *OOTB Switch & LED Backpack* is pressed, an 8-bit duty cycle for **only** the *BLUE_PWM* LED should be specified. Lastly, whenever tactile switch S1 on the *OOTB Memory Base* is pressed, an 8-bit duty cycle for **only** the *GREEN_PWM* LED should be specified.
- A pulse-width modulation mode for a timer/counter module within the *ATxmega128A1U* must be utilized to generate the relevant RGB LED signals.
- Since the RGB LEDs located on the *OOTB µPAD* are active-low, the logic levels of the I/O pins connected to the LEDs should be inverted, so that all duty cycles may match intuitively. (Recall that a *duty cycle of a signal* is usually defined as *the percentage of time that the signal is high*.) To invert the logic levels of the I/O pins, utilize a particular setting of the *PINnCTRL* configuration register within the relevant I/O port.

NOTE: When accessing, i.e., either reading from or writing to, a set of registers representing a value larger than 8-bits within the *ATxmega128A1U* (like when storing to or reading from the PER, CCX, and CNT registers within a timer/counter module), it is **necessary** to access each register within the set, from least significant byte to most significant byte, for a change to take effect.

1.4. Write an assembly language program, **`hw2.asm`**, as described above.

1.5. Take three pictures of your uPAD as described below.

1.5.1. The red LED at about 75% duty cycle with the other LEDs at 25% duty cycle.

1.5.2. The green LED at about 75% duty cycle with the other LEDs at 25% duty cycle.

1.5.3. The blue LED at about 75% duty cycle with the other LEDs at 25% duty cycle.

HOMEWORK PROCEDURE SUMMARY

1) Read any pertinent material to learn how to implement pulse-width modulation techniques with the *ATxmega128A1U*. Understand how to interface with the RGB LED package located on the *OOTB µPAD*.
2) Answer all homework exercises.
3) Write an assembly program, **`hw2.asm`**, to control the light intensity of the each of the LEDs within the RGB LED package located on the *OOTB µPAD* via switches located on both the *OOTB Switch & LED Backpack* and the *OOTB Memory Base*.
4) Take the three specified screenshots and include them in the document..
5) Submit a pdf file in the format specified in the *Homework Submission Template* on our website. Include the asm file and the three images in the submitted document.