

## Design Assignment 2

CpE 301

Due: Monday, Feb. 13<sup>th</sup>, 2017 by 11:59pm

Spring 2017 (Dr. Harris)

### Introduction

The purpose of this design assignment is to give you experience using timers and interrupts and to gain practice writing programs in C. This document will guide you in completing the following tasks.

1. Writing assembly and C programs to toggle LEDs at defined intervals.
2. Demonstrating your program in hardware.

### 1. Writing assembly and C programs to toggle LEDs at defined intervals

In this section, you will write AVR assembly and C programs to use timers and interrupts on the ATmega328p. You will write two assembly programs and two C programs. Each of the four programs will perform the same functions. They will:

1. Generate a waveform on PORTC.0 with a 50% duty cycle and a 0.5 second period.
2. Implement an 8-bit counter that increments on each rising edge of the waveform from part (1). Use this counter for parts 3 and 4 below.
3. Display the value of the counter on PORTB.0-7. In hardware these pins will be connected to the lower 8 bits of the 10-bit LED bar included in your kit. (The counter overflowing is no problem.)
4. Toggle PORTC.5 for every 5th rising pulse and PORTC.6 on every 10th rising pulse of the counter. In hardware you will connect the two most significant bits of the 10-bit LED bar to the PORTC.6 and PORTC.5 pins on the ATmega 328p. Connect PORTC.6 to the msb of the LED bar.

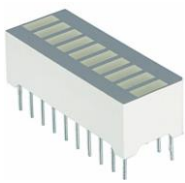
Write the following four programs:

- **DA2\_1:** An **assembly program** called DA2\_1 that implements the above functions using timers but no interrupts.
- **DA2\_2:** An **assembly program** called DA2\_2 that implements the above functions using timers as before but that uses interrupts to update PORTC.5-6.
- **DA2\_3:** A **C program** called DA2\_3 that implements the above functions using timers but no interrupts.
- **DA2\_4:** A **C program** called DA2\_4 that implements the above functions using timers as before but that uses interrupts to update PORTC.5-6.

### 2. Demonstrating your C program in hardware

Now demonstrate your assembly and C programs in hardware. First, connect the 10-bit LED bar (see Figure 1 in your kit) to PORTB.0-7 and PORTC.5-6. Remember to connect the ATmega328p chip pins to the anode (the positive polarity pin of the LED) of each pin on the LED bar. Each of

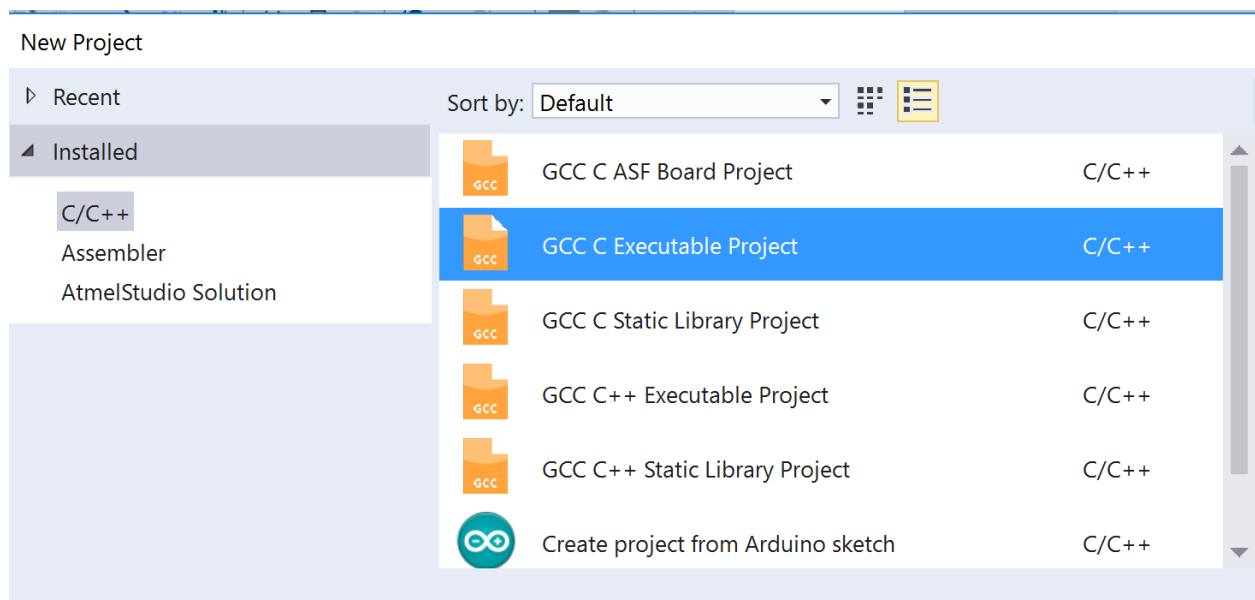
the pins on the other side of the LED bar (cathodes) should connect through 10 330 Ohm resistors to ground. The LED anode side is typically on the side or the LED bar that has writing on it, but you should test this out to verify. It is a good idea to verify that none of the LEDs are burned out before connecting them to the microcontroller.



**Figure 1. 10-bit LED bar**

Program the ATmega 328p with each of the four programs and verify each of them in hardware.

Recall that setting up a C project in Atmel Studio is similar to setting up an assembler program. Select File → New → Project as you have done before, select C/C++ and GCC C Executable Project, as shown in Figure 2.



**Figure 2. New C Project**

## What to Turn In

The following must be submitted via WebCampus by the due date/time to receive credit for the assignment. Messy, difficult to understand, or disorganized work will receive no credit.

**Total points available: 100**

0. **Time:** Indicate the amount of time this assignment took in hours. This will not affect your grade (unless omitted) but will help gauge the workload for this and future semesters. **[-5 points if omitted]**
1. **A pdf or word document** that contains details about your design in Part 2. Be sure to submit them **in this order**:
  - a. A 1-paragraph description of each design. **[20 pts]**
  - b. The code for each of the four programs. The code must have been built (i.e., compiled/assembled) and working. The code should be clear, well-formatted, and well-documented. **[40 pts]**
  - c. A flow chart of each program (hand-drawn is fine as long as it's legible). **[15 pts]**
  - d. A schematic of your hardware (again, hand-drawn is fine as long as it's legible). This should show the chip and the external parts. **[15 pts]**
  - e. A link to a ~1-2 minute Youtube video showing each of the 4 programs working in hardware. **[10 pts]**