**CG1112 Engineering Principles & Practices II**

**Week 5 – Tutorial 2 – Part 1**

Objectives:

* Explore the Atmel microcontroller in more detail.
* Review GPIO Peripheral Block
* Understand Energy-Saving Concepts

**Q1.**

Most Microcontroller Pins have multiple functionalities. It is up to the designer to plan ahead and be aware of the peripherals that are needed and select the appropriate pins to be used in the project. In the studio we have used PORT B pins for the LED’s and Switches.

PB7 and PB6 are currently mapped to the external crystal oscillator (XTAL1 and XTAL2 for the Uno Board).



You have decided to create your own board using the Atmega328p and want to make use of all 8-bits of PORTB. Is it possible? What are the factors to consider?

(Hint: Refer to “Section 13: System Clock and Clock Options” in the Atmega datasheet)

**Q 2.**

In our studio we configured LED’s in “Active-High” logic, i.e. we applied a Logic ‘1’ to Turn it ‘ON’ and a Logic ‘0’ to turn it ‘OFF’.

1. Draw a circuit connection to connect an LED in “Active-Low” logic to PORTB Pin 1, i.e. you need to apply a Logic ‘0’ to turn it ‘ON’ and a Logic ‘1’ to turn it ‘OFF’.
2. Write the code to set the DDRB register value according to your schematic.
3. In a design such as this, we say that the microcontroller is “sinking” current. That means, that current is now flowing into the pin of the device. The manufacturer specifies a limit to the amount of sinking current. Anything more than this, could lead to a “smoke-effect” + “water-sprinkler” effect in the lab.

Refer to Table 32-2 in the Atmega328p datasheet and state the maximum sinking current when the device is operating at 5V.

1. The voltage drop across the LED is 0.7V. Choose an appropriate R value to prevent a “smoke-effect”.

**Q 3.** Power Consumption is a critical factor in Embedded Systems and it is important to minimize is to as to extend its usage before recharging the batteries. You read from Section 14 of the datasheet that the AT328P has some energy-saving features that you want to implement for “Alex” your robot. Mainly, you wish to put the robot in “Standby Mode” when the robot is idle and not doing anything useful.

1. Which is the main register that controls these features and what value should be written to it?
2. In Standby Mode, which events can trigger the device to “wake-up” and resume full-functionality?
3. How many clock cycles does it take for the device to come out of Standby mode to full operation?
4. Based on your initial assessment, you feel that you may not need to use the ADC module for this project. As such, you want to disable it so that it doesn’t consume any additional power. How can this be achieved?