**The Hong Kong Polytechnic University**

**Department of Electronic and Information Engineering**

**EIE3105 Integrated Project (Part I)**

**Laboratory Exercise 1: Introduction to AVR**

**(Deadline: Check the course information)**

## Objective:

1. To familiarize students with the use of an AVR microcontroller.
2. To develop assembly and C programs under the Arduino platform.

## Equipment:

Atmel Studio 6 (software)
The Arduino Starter Kit (hardware)



A cable to connect the microcontroller to a computer

Components

An ATmeg328p microcontroller and a breadboard

## Important Notices:

1. You must read the ATmega328p datasheet carefully before you do the laboratory exercises. You have three ways to access the datasheet:
   **a.** Download the pdf file from Blackboard.
   **b.** Use a web searching engine (e.g., Google) to search the keyword "ATmega328p datasheet" and get the pdf file.
2. You leave your student identity card to our technicians to borrow one box of the kit. They will return your card to you when you return the box. Note that you need to return the box 5 minutes before the end of the laboratory session; otherwise, our technicians will pass your card to the instructor and you will have penalties when it happens.

**Introduction:**

This experiment introduces some simple applications of using an AVR microcontroller to perform some arithmetic and I/O operations. Students are required to develop assembly and C programs and build up some simple electronic circuits for such simple applications.

Atmel Studio 6 is the integrated development platform (IDP) for developing and debugging Atmel AVR microcontroller (MCU) applications. The Atmel Studio 6 IDP gives you a seamless and easy-to-use environment to write, build and debug your applications written in C/C++ or assembly code. Atmel Studio 6 supports all 8- and 32-bit AVR MCUs. It also connects seamlessly to Atmel debuggers and development kits.

Arduino is an open-source physical computing platform based on a simple I/O board and a development environment that implements the Processing/Wiring language. Arduino can be used to develop stand-alone interactive objects or can be connected to software on your computer.

The Arduino Start Kit walks you through the basics of using the Arduino in a hands-on way. You will learn through building several creative projects. The kit includes a selection of the most common and useful electronic components.

**Procedure:**

*Section A: Create your first simple application*

1. Open Atmel Studio: Double click Atmel Studio 6.2.

2. Go to main menu. Select "Tools" → "External Tools". Type in "Arduino Uno" in the field "Title". Moreover, type in the following paths into the fields "Command" and "Arguments":

Command

```
C:\Program Files\Arduino\hardware\tools\avr\bin\avrdude.exe

-C"C:\Program Files\Arduino\hardware\tools\avr\etc\avrdude.conf"
-v -v -v -patmega328p -carduino -P\\.\COM4 -b115200 -D -
Uflash:w:"$(ProjectDir)Debug\$(TargetName).hex":i
```

Arguments

Note that "COM4" in the above path shows the port that the Arduino Uno plugs in your computer. Thus, later, when you plug in the Arduino Uno to your computer, you need to check the port is "COM4" or not (Double click the icon "Computer" and Select "Properties" → "Device Manager" → "Ports". Check the position of the port for the Arduino Uno). You may need to change the port number if it is not "COM4". Moreover, you may go to the course web site to download the text file "ArduinoUno.txt" which stores these two paths. Then you can copy and paste them into such two fields. Note that this file can also be found in "K:\Arduino". Finally, in some computers, the directory "Arduino" may not be installed in the directory "Program Files" but "Program Files (x86)". If it happens, please change the name of the directory "Program Files" in the command and the argument accordingly.

After that, click the checkboxes "Use Output window" and "Prompt for arguments" and then click the button "OK". When you finish the setting, you will see a new field "Arduino Uno" is created under the menu item "Tools". Note that the above setting is required to go through once it is the first time to open Atmel Studio.
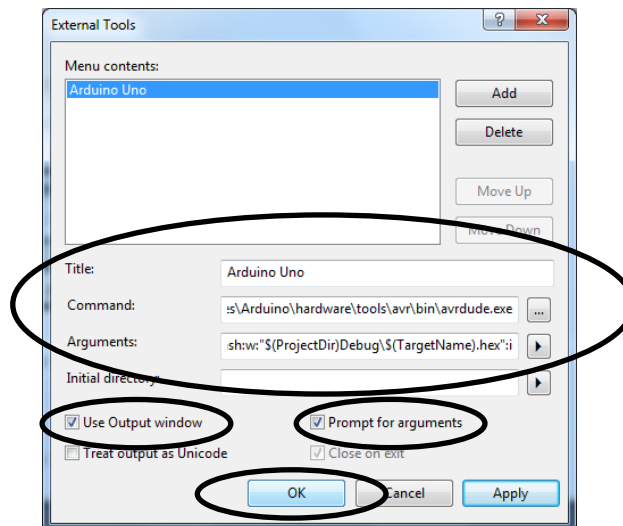


3. Go to the main menu and create a new project. Then select "C/C++" and then "GCC C Executable Project" to create a new project. Type in the project name "Lab1A". After that, select "ATmega328p".
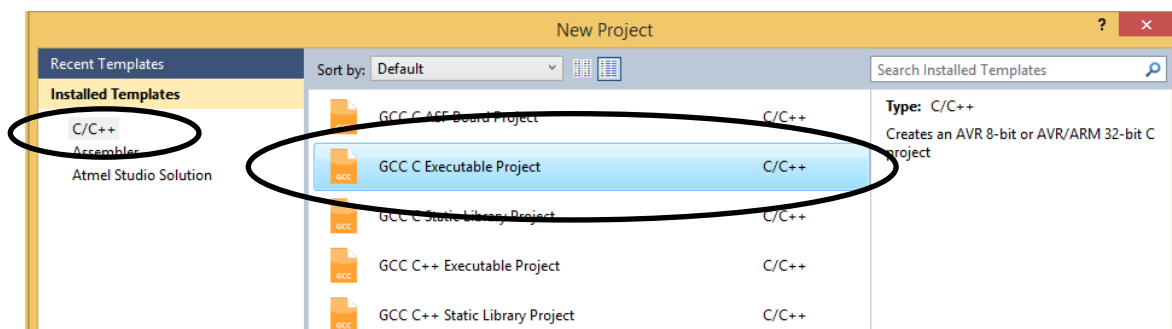


Note that the I/O ports of the AVR microcontroller are bit-accessible but the AVR C compiler does not support this feature. Thus, you should use a mask to set, clear and check some bits in the I/O ports.
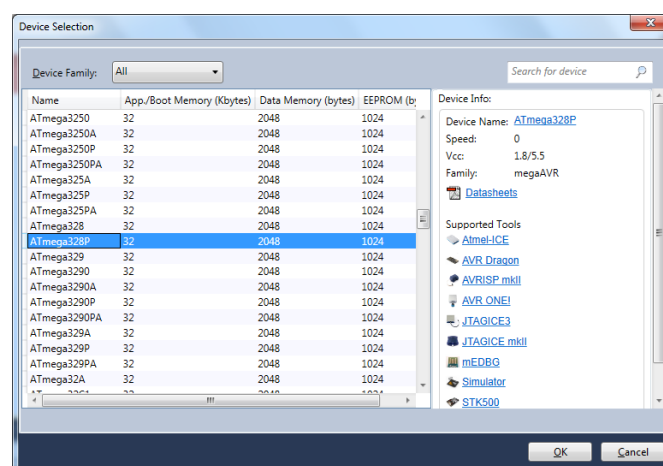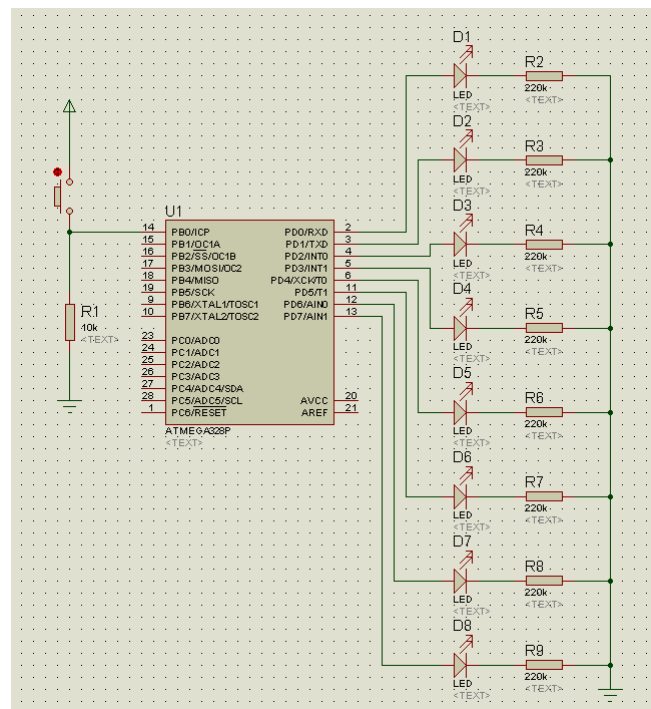
4. Type in the following program into "Lab1A.c".

```c
#include <avr/io.h>

int main(void)
{
    DDRD = 0xFF;
    while(1)
    {
        PORTD = 0x55;
    }
}
```

5. Go to the main menu. Select "Build" → "Build Solution" to compile the program.

6. Build the circuit below on the Arduino Uno and the breadboard. Refer to the pinout diagram in the appendix. Note that "220k" should be "220". It means 220Ω.



7. Go to the main menu. Select "Tools" → "Arduino Uno" to burn the program into the Arduino Uno. You should see some LEDs are on. Note that you should remove any connections to pin 0 (RX) and pin 1 (TX) of Arduino Uno before you burn your program into it; otherwise, you may not burn your program successfully.

8. Repeat the above steps but this time Port D is toggled between 0x55 and 0xAA with 1 to 2 seconds time delay. You should use looping to generate the delay and set the optimization level to the first one (level zero). Show the calculation how to generate the delay. Note that the clock frequency of the Arduino Start Kit is 16 MHz. When your program works properly, you should see all LEDs connected to Port D are flashing.
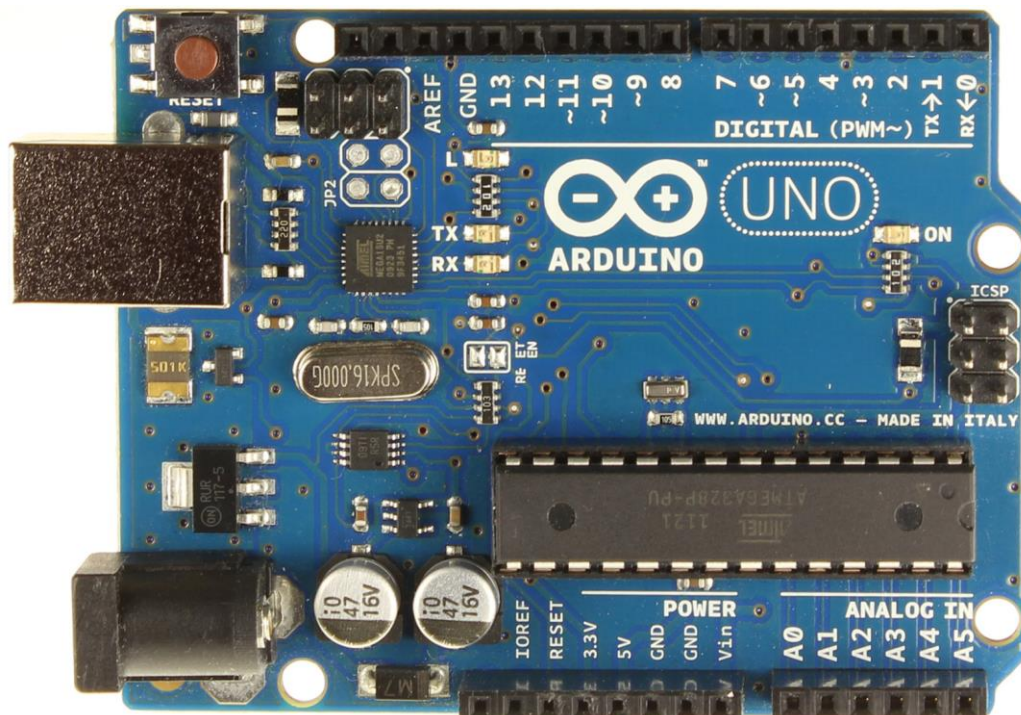
*Section B: Simple applications*

1.  Connect a switch to PB0 and the pin diagram is shown in Step 6 of Section A. Connect a LED to the switch. When the switch is pressed, the LED is on. When the switch is released, the LED is off.

2.  Use the circuit in Section A and the connection in Step 6. There are two states in the switch: State 0 and 1. When it is in State 0, all LEDs connected to Port D are off. When it is in State 1, all LEDs are on. At the beginning, the switch is in State 0. When the switch is pressed and it is in State 0, it goes to State 1. When the switch is pressed and it is in State 1, it goes to State 0.

*__Demonstrate Section A (Step 7 and 8) and B (Step 1 and 2) to our tutors or technicians.__*

**Instructions:**

1.  You are required to demonstrate your programs to our tutor or technicians.
2.  Zip all programs (including the whole projects) in Section A and B to a single file. Submit it to Blackboard.
3.  Deadline: **Check the course information**.

Appendix A: Arduino Uno Outlook

# Appendix B: Arduino Uno Pinout Diagrams

*Ivan Lau*
*Lawrence Cheung*
*August 2020*