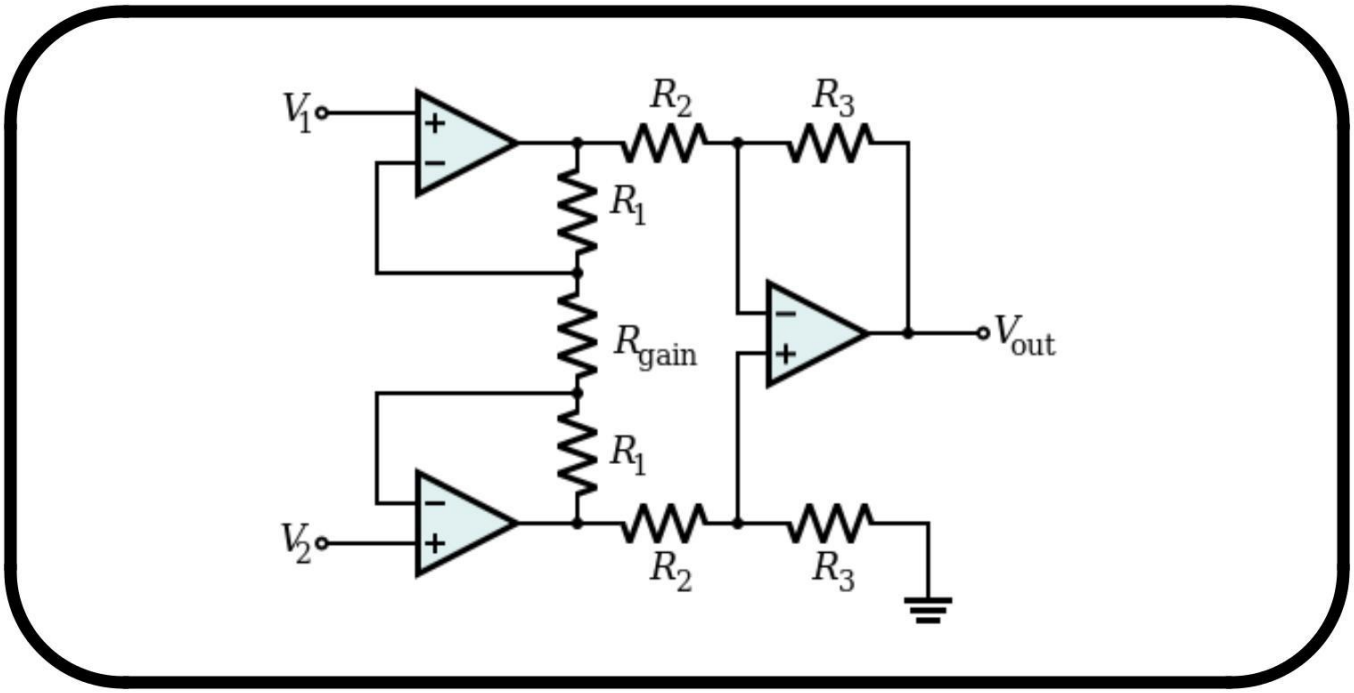
MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF PHYSICS

PRINCIPLES OF MEASUREMENT AND INSTRUMENTATION I

LABORATORY MANUAL

Experiment 2



Prepared by

Tarek Elsebaei

Revised by

Deniz Orhun Boztemur

EXP.2 Arduino-Based Measurement Devices

Preliminary Work

- Explain the difference between microprocessor, microcomputer and microcontroller.
- Explain the difference between an analog signal and a digital signal.
- Explain the basic principle of the PWM. (Pulse Width Modulation).
- Describe the purpose of the following functions in Arduino:

analogRead() – analogWrite() – digitalRead() – digitalWrite() – pinMode() – Serial.begin() –

delay() – millis() – Serial.print() – Serial.println() – Serial.parseInt() – map() – random()

- What does Clock Speed mean for a microcontroller? How is it related to the time response of the
- microcontroller?
- Explain INTERRUPT on Arduino with an analogy.
- What is the resolution of Analog-to-Digital Converter in Arduino UNO?
- **Writing the codes before the experiment and handing them in with the preliminary report will get you 10 bonus points!**

Purpose

To explore Arduino Uno and Arduino based circuits. To learn the basic Arduino programming.

Theory

Microcontrollers are being used as simple solutions to tasks that previously utilized transistor-transistor-logic. With the profusion of these devices being used in industry as well as in hobby electronics, the ability to program them can be very useful for any project that may be attempted. While microprocessors, microcomputers and microcontrollers all share certain characteristics and the terms are often used interchangeably, there are certain distinctions that are used to classify them into separate categories.

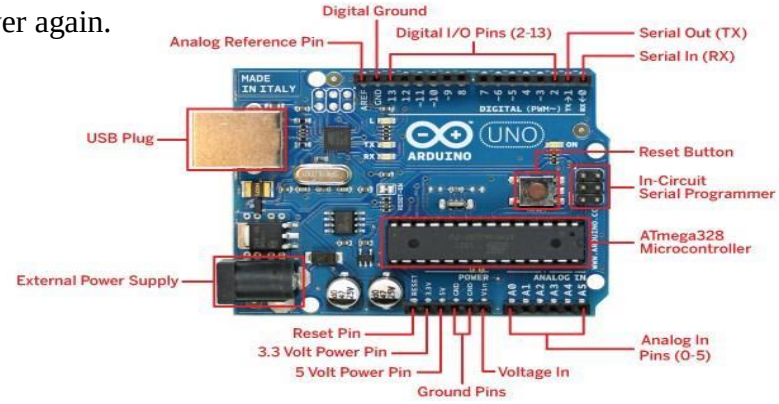
The simplest of the three categories is the **microprocessor**. Also known as a CPU (Central Processing Unit), these devices are generally found at the heart of a much larger system such as a desktop computer and are primarily used as data processors. They generally consist of an arithmetic logic unit (ALU), an instruction decoder, a number of registers and digital input/output (DIO) lines. Some processors also include memory spaces such as a cache or stack which can be used for more rapid temporary storage and retrieval of data than having to access system memory. Additionally, the processor must connect 1 to some form of data bus to access the memory and input/output peripherals external to the processor itself.

A **microcontroller** is, in some ways, a cross between a microprocessor and a microcomputer. Like microprocessors, the term microcontroller refers to a single device; however, it contains the entire microcomputer on that single chip. Therefore, a microcontroller will have a processor, on-board memory as well as a variety of IO devices. A microcontroller is a computer present in a single integrated circuit which is dedicated to perform one task and execute one specific application. It contains memory, programmable input/output peripherals as well as a processor. Microcontrollers are mostly designed for embedded applications and are heavily used in automatically controlled electronic devices such as cellphones, cameras, microwave ovens, washing machines, etc.

The **Arduino** environment has been designed to be easy to use for beginners who have no software or electronics experience. With Arduino, you can build objects that can respond to and/or control light, sound, touch, and movement. Arduino has been used to create an amazing variety of things, including musical instruments, robots, light sculptures, games, interactive furniture, and even interactive clothing. Arduino is an open-source electronic platform that use Atmel microcontroller chips based on easy-to-use hardware and software. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino Programming Language (based on Wiring), and the Arduino software IDE, based on Processing.

Arduino Uno

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.



Technical Specifications

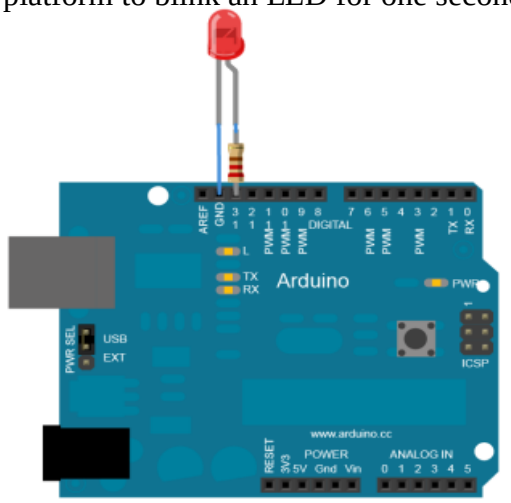
Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage(recommended)	7-12V
Input Voltage(limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Equipment

- Electronic Board
- Arduino UNO
- Resistors
- LED
- Push Button
- Potentiometer

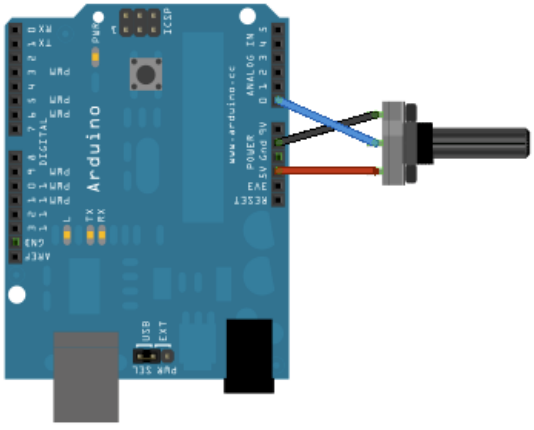
Procedure

1. Write a program on Arduino platform to blink an LED for one second by using the delay function.



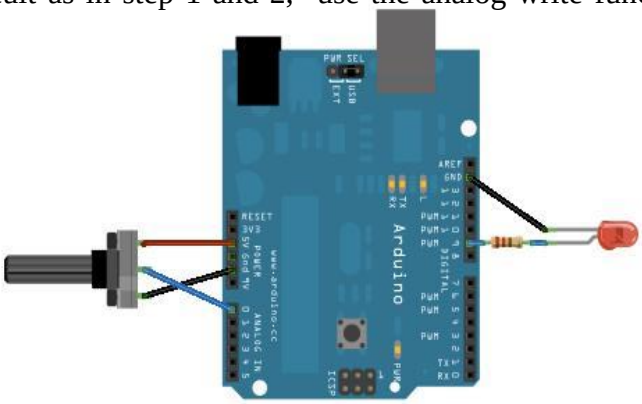
- Write a program to blink the LED without using the delay function
- Explain the reasons behind using a current-limiting resistor for the LED

2. By using the 1k ohm potentiometer implemented in the electronic board, connect one terminal to 5v and the other terminal to ground and connect the output of the potentiometer to one of the Arduino pins. Use analog read function and serial print to read the output.

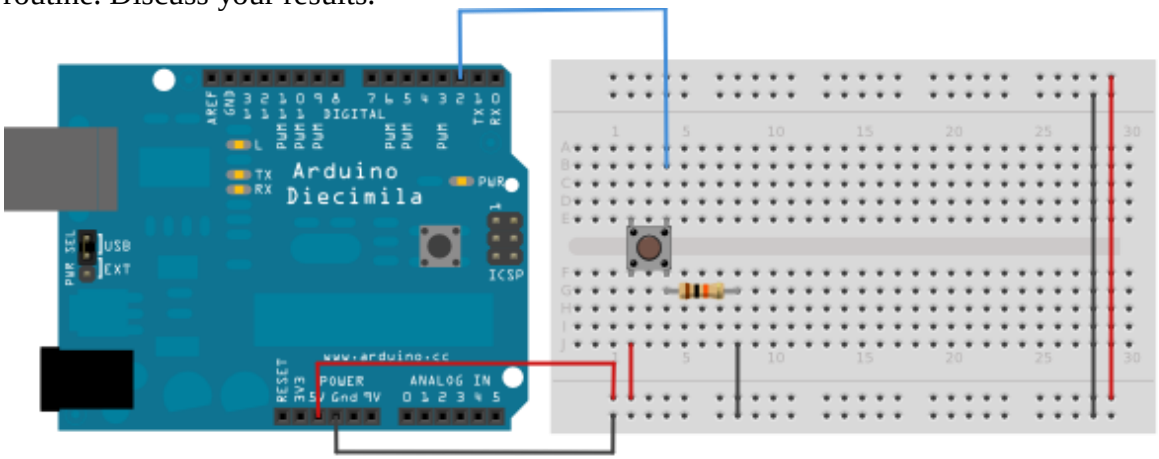


- What are your results? And what is the resolution of your Arduino analog-to-digital converter?
- By using map function in Arduino make your results scale to be from 0 to 5
- Write a program to read a float type of the potentiometer output from 0 to 5

3. By using the same circuit as in step 1 and 2, use the analog write function in Arduino to control the brightness of an LED.

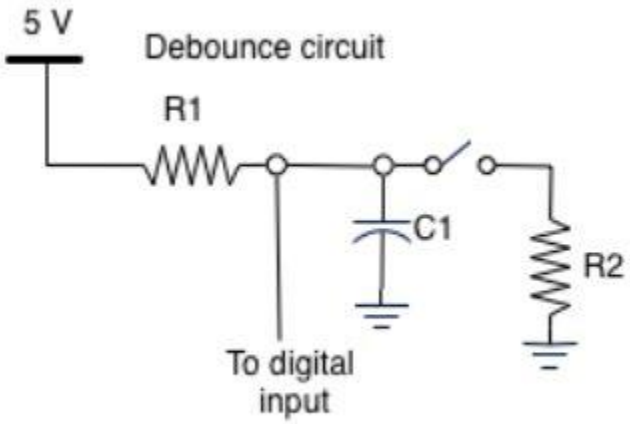


4. Write a program that can count the number of clicks on a push button by using Arduino interrupt service routine. Discuss your results.



Hint: choose the pin mode of the interrupt pin to be INPUT_PULLUP and add the reason of that to your report.

5. Solve the problem by constructing the circuit below.



- How the capacitor can solve this problem? (C_1 = 100nF)
- Compare your results with the previous part.