

Lab 12: Microcontrollers I

Prelab

We will be using the SparkFun RedBoard microcontroller, which is an Arduino UNO clone.

- Watch this introduction to Arduinos: <https://www.youtube.com/watch?v=BtLwoNJ6klE>.
- Q. What is the difference between a digital input and an analog input?
- Q. What does he list as the 4 main functions of the Arduino?
- Q. What commands would you use to set pin 13 to zero volts and to 5 volts?
- Q. What does he list as the 5 fundamental commands?
- From Moodle download the SparkFun Inventor's Kit (SIK) guidebook pdf for reference. There will also be a paper copy at each desk.
- Download and install the Arduino software (not the web editor or the Windows Store App, use the "Windows Installer" link for Windows) from www.arduino.cc/en/main/software on at least one partner's computer (Windows, Mac, or Linux) which has a type A (rectangular) USB port. (Mac or Linux tend to have the fewest issues). As of April 2018 the current version is 1.8.5.
- From Moodle download the example code zip file, SIK_Guide_Code.zip.
- Unzip the file. Open the resulting folder and find the list of folders for circuits 1 through 16. Following the instructions below, find the Arduino "examples" folder.

The most reliable way to find your examples folder is to look for where the built-in examples are kept. In the Arduino software and open the "Blink" example program by going to:

File -> Examples -> 01.Basic -> Blink

Once it's open go to **Sketch -> Show Sketch Folder**

The examples folder you want should be up two levels from the folder that just opened.

- Create a folder called **SIK.Code** inside the examples folder. Copy all of the circuit folders from the file you unzipped (1 to 16) into that directory.

Once you've copied the new examples into the **SIK.Code** folder, restart the Arduino software (if it's open) and you should be able to see the circuits listed under:

File -> Examples -> SIK.Code.

Part I: Setup and Testing

Construct all of these circuits on the small breadboard which comes in the kit.

- Follow the instructions on pages 5 – 8 of the SIK guide to connect the board to your computer.

If the port doesn't show up, try installing new drivers following the guide for your OS:

<https://learn.sparkfun.com/tutorials/how-to-install-ftdi-drivers/windows---in-depth>

<https://learn.sparkfun.com/tutorials/how-to-install-ftdi-drivers/linux>

<https://learn.sparkfun.com/tutorials/how-to-install-ftdi-drivers/mac>

- Construct circuit #1 (Blinking an LED, pp. 19-24) from the SIK guide.
- Open the sketch, take a look through the code, and upload it to the board following the instructions in the guide.
- Verify that it works. Make changes by altering values in the code and re-uploading to see how your changes effect the LED.

If you can upload but it doesn't work, see the warning below for Window's users.

- Q. An arduino program is called a "sketch." Each sketch must have a `setup()` section and a `loop()` section. What is the difference between these sections?

Warning: If you are using Windows 10 (and possibly older versions and other operating systems), you may encounter issues with your computer sending a reset signal over and over to the board. This results in the blue LED flashing in bursts of three. If you are having this issue, try the following.

- Connect a 10uF capacitor between RESET and GND (if using an electrolytic cap, the negative leg goes to ground).
- To upload a sketch, click upload and when the IDE says "uploading," hit the reset button on the RedBoard. Sometimes it takes a couple tries.
- If using the serial monitor, you will probably see a bunch of junk text after an upload. Hit reset again to fix it.
- If you are still having issues, unplug the board, plug it back in, and try the same steps again.

Part II: Multicolored LEDs and PWM

- Read through and complete circuit #3 (RGB LED, pp. 29-32) from the guide.
- Once it is working, edit the code to only run the `showSpectrum()` function (you only need to edit two lines). Then edit the timing to slow down the color changes by a factor of 5.
- Connect channel 1 of your oscilloscope to the red leg of the LED and channel 2 to the blue leg.

- Q. Describe what you see. Summarize what PWM is and how it is used (See page 31).

Part III: Servos and Sensors

- Construct circuit #8 (A Single Servo) and run the example `SIK_circuit08-1_servoSweep`.
- change the delay values in the `for()` loops and observe the effects on the servo's rotation speed.
- Add circuit #9 (Flex Sensor) to the breadboard and run the example 9 code.

In #9 you will probably notice that the servo doesn't rotate a full 180 deg between the flex sensor bent fully one way and bent fully the other.

- Open the serial monitor (see page 47) and check what range of resistance values are being read by the arduino. Edit the code to more accurately map this range onto the servo's range.

Q. What changes did you have to make?

Part IV: Electronic switches: Transistors and Relays

4.1 Arduino controlled

Be careful to use a transistor and not the temperature sensor, they look nearly identical!

- Construct and run circuit #13 (Relays). (Note that some kits have a replacement relay with a different pinout, follow the label on the back of the relay.)
- Change the switching speed in the code to a few different values. As a rule of thumb, the switching rate for these hobby quality mechanical relays should stay under about 20Hz (so don't set the delay less than 50 ms).

4.2 Switch controlled - with protection diode

We were using the transistor as an electronic switch. We can also control the relay using a mechanical switch.

- Remove the arduino and transistor from the circuit.
- Using a **FLOATING** DC power supply (your HP/Agilent Bench power supply with current limit set to 150 mA) and a push button switch, set up the circuit in figure 1.

The diode protects the circuit from high reverse voltages.

- Q. Explain what causes these voltage spikes.
(I highly recommend watching this video: <https://youtu.be/LXGtE3X2k7Y>)

Now we'll investigate these spikes with the oscilloscope.

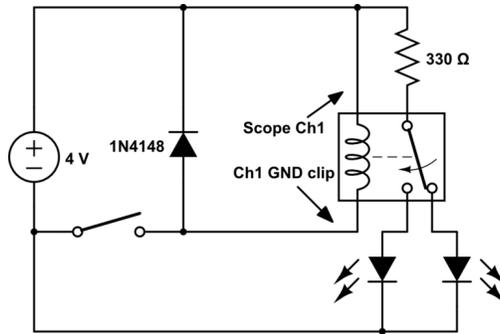


Figure 1: A relay controlled by a push button switch.

- Connect a single oscilloscope probe (**in 10x mode**) across the coil **with the ground clip on the same side as the push button.**¹
 - Set the voltage on the scope to 1 V/div, the time to 25 ms/div, and the trigger to 2 volts.
 - Set the scope to capture one trigger event with the “Single” (or “Single SEQ”) button. The scope will now wait until it gets a trigger event, then keep that data on the screen so you can look at it more closely.
 - Quickly tap (push and release) the button on the breadboard.
 - Adjust the scope settings and retry the capture until you can see the rising **AND** falling edges of the voltage.
- Q. What do you notice about the falling edge of the voltage? Measure the reverse voltage value. This number should look familiar, where does it come from?
- Q. Include a picture of the output in your report.

4.3 Switch controlled - without protection diode

Now let’s see what would happen without the protection diode. This won’t damage anything since we aren’t using the transistor, but we will be able to see the spikes the transistor would have had to handle.

- Your power supply should be able to protect itself but just to be safe let’s take that diode and put in on the positive voltage rail to prevent reverse voltage from going to the power supply.
- The circuit should now look like figure 2.

The scope should still be connected across the relay coil, (**still in 10x mode**).

- Zoom in on the time axis (10 μ s/div) and out on the voltage axis (50 V/div). Set the trigger to around -60 volts.
- Q. Use single capture mode again and use the cursors to measure the reverse voltage spikes on several different captures. Include sketches or pictures of a few of them.

¹Since we are using a floating supply, the oscilloscope ground clip doesn’t need to be at the same potential as the negative side of the power supply.

- Q. If we were still using the transistor, are these spikes large enough to damage it? Take a look at the datasheet to find out: <http://www.onsemi.com/pub/Collateral/P2N2222A-D.PDF>.

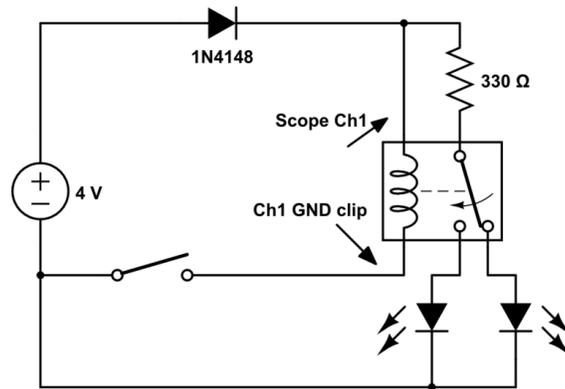


Figure 2: A relay controlled by a push button switch.

The next time you are using an electric kettle or a coffee maker, listen for the click when the relay turns on or off!

Part V: Hello World!

- Construct circuit #15 (LCD). There are a lot of connections; use wire colors wisely to help keep things straight.
- Q. Edit the code to display your own message. Take a picture of the completed circuit and the custom message to include in your report.

If you have extra time, try to get the display to show a scrolling message. Check out the Liquid-Crystal library <http://www.arduino.cc/en/Reference/LiquidCrystal> for what commands to use.



Figure 3: <https://xkcd.com/228/>