Lab 12: Microcontrollers I

1 Setup

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

Pre-lab tasks

- Download the SparkFun Inventor's Kit guidebook to reference. There will also be a paper copy at each desk.
  cdn.sparkfun.com/assets/learn_tutorials/3/1/0/RedBoard_SIK_3.2.pdf
- Download and install the software from www.arduino.cc/en/main/software on at least one partner's computer (Windows, Mac, or Linux).
- Download the example code from www.sparkfun.com/sikcode. Unzip the resulting file and copy into the examples folder in the arduino program folder. See page 9 of the SIK guide pdf for help.

  Note that on Windows 10 64-bit (and probably 7 and 8) the path will be as shown below. If you are running 32-bit then just remove the " (x86)."

  `C:\Program Files (x86)\Arduino\examples\SIK_Guide_Code_32`

Connecting the board

Follow the instructions on pages 5 – 8 of the SIK guide to connect the board to your computer. (If you have issues, see the warning below.) **Set up circuit #1 (Blinking an LED) 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 and you can make changes by altering values in the code and re-uploading to see how your changes effect the LED.

There are two main logical sections to every sketch, what are they and how do they differ? (Note that a 'sketch' is what an arduino program is called.

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 it will probably spew junk 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.

Take a pictures of each completed circuit to include in your lab report.

2 Multicolored LEDs and PWM

Read through and complete circuit #3 (RGB LED) from the guide. Once it is working, edit the code to only run the showSpectrum() function (you only need to edit one line). Edit the timing to slow down the color changes by a factor of 5. Connect your oscilloscope from the red pin to the ground pin. *Describe what you see. Summarize what PWM is and how it is used.*

3 Servos and Sensors

Construct and run the code for circuits #8 (A Single Servo) and #9 (Flex Sensor) from the guide. For circuit #8, change the delay values in the for loop and observe the effects on the servo's rotation speed. 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 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. *Make a note of the changes you had to make.*

Take a picture of the final circuit to include in your lab report.

4 Electronic switches: Transistors and Relays

4.1 Arduino controlled

Construct and run circuit #13 (Relays). **Be careful to use a transistor and not the temperature sensor, they look nearly identical!** 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're 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 and, using a **FLOATING** DC power supply and a push button switch, set up the circuit in figure 1. Test that you can control the LEDs.

The diode protects the circuit from high reverse voltages. *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.

Connect the leads of the oscilloscope probe (**in 10x mode**) across the coil with the ground clip on on the same side as the push button.[1] Set the voltage on the scope to 1 V/div, the time to

[1] Note that since we are using a floating supply, the oscilloscope is not providing a return path and is thus taking a differential measurement.

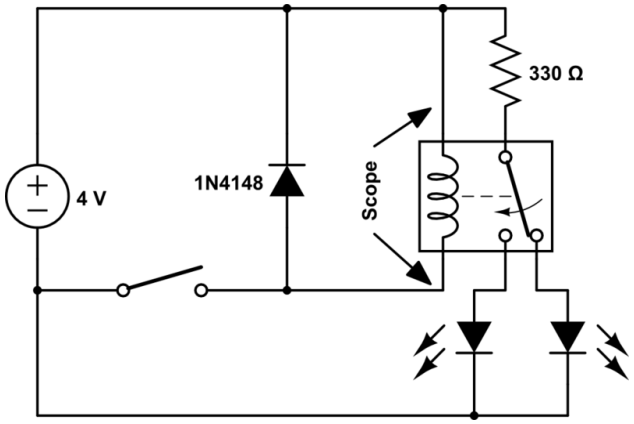

Figure 1: A relay controlled by a push button switch.

25 ms/div, and the trigger to around 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 push the button on the breadboard in and out, and keep adjusting the scope settings until you can see the rising **and** falling edges of the voltage. You may have to retry the capture a few times.

What is the peak to peak voltage? 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? 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 μs/div) and out on the voltage axis (50 V/div). Set the trigger to around -60 volts. *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.*

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_link/Collateral/P2N2222A-D.PDF`.

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

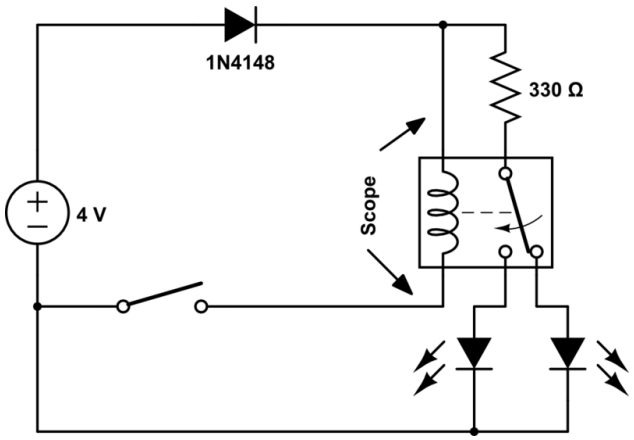

Figure 2: A relay controlled by a push button switch.

5 Hello World!

Construct circuit #15 (LCD). There are a lot of connections, use wire colors wisely to help keep things straight. 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 LiquidCrystal library http://www.arduino.cc/en/Reference/LiquidCrystal for what commands to use.

Figure 3: https://xkcd.com/567