University of California, Davis

Electrical and Computer Engineering

California State Summer School for Mathematics & Science

Cluster 8: Internet-of-Things

Lab 2

Serial Input/Output, Potentiometer and LED Testing

Introduction:

a. Purpose/ Scope

In this lab we will continue the exercises from the previous lab. Last time, you were introduced to the microcontroller, how to write a simple program, and how to interface with a breadboard. This week, we will continue to look at features of the MCU and write a program to use your microcontroller as a voltmeter. You will also learn how to use the ADALM2000 (also known as M2K) from Analog Devices (AD). The M2K is a powerful tool which can be used as an oscilloscope, signal generator, multimeter, and power supply. You will also test light-emitting diodes (LEDs) which are very common optoelectronic devices.

b. Resources/Lab Materials

Lab Materials:

- 1x TI MSP432P401 LaunchPad Kit (Red kit)
- 1x AD ADALM2000 (M2K)
- 1x Breadboard
- 1x Breadboard Booster Pack
- 2x red LEDs
- 1x green LED
- 1x blue LED
- 1x Wire Jumpers
- 2x 330 Ohm resistor
- 2x 1k Ohm resistor
- 1x 10k Ohm resistor
- 1x Potentiometer (POT)

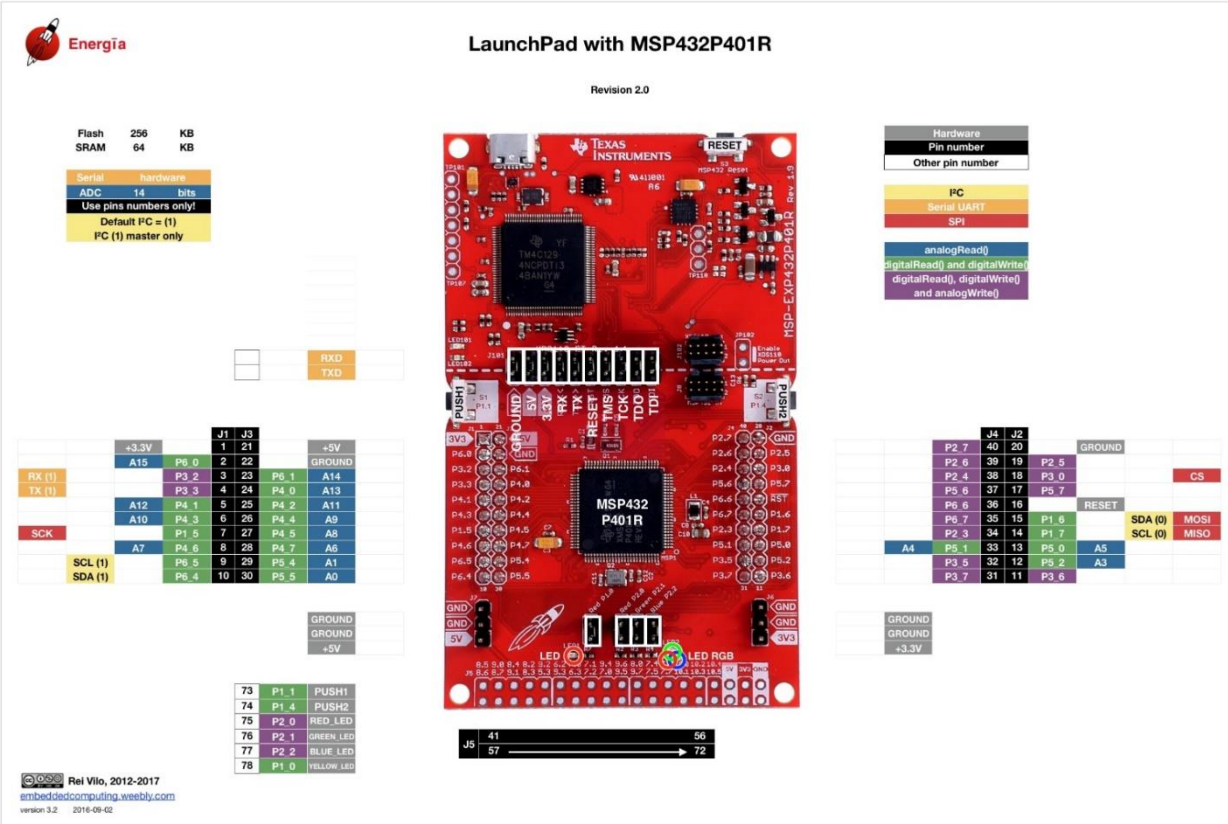


Fig. 1. Pin layout of MSP432

Exercise 1: Using the UART to Monitor Program Execution

Description: In this exercise, you will learn how to utilize the communication interface known as the "UART" (pronounced you-Art) to communicate between the PC and the MCU. This is useful if you want to know values that you are reading into the MCU. For this class we will use this extensively to measure voltages in our circuits, but this can even be used to display messages in different parts of your code to inform you that a certain event has happened which is particularly helpful when troubleshooting. On a superficial level, the serial communication functionality will appear to be nothing more than writing text to a monitor, however, its utility will become more evident as you progress.

Procedure:

Part I: "Hello, World!"

It is a tradition for the first program someone writes when learning how to code to be a simple program that displays the message, "Hello, World!" on the screen. While we have already written a few programs, now is just as good a time as any to write this program.

1. There are two commands that are particularly important:

```
Serial.begin([baud rate])

Serial.println([message])
```

Copy the program written below into Energia but **do not execute** it.

```
//Exercise 1: UART monitor

void setup(){
  //Initialize serial communication to 115200 bits per second.
  Serial.begin(115200);
}

void loop(){
  Serial.println("Hello, World!");
  delay(10000); //prints statement to monitor every 10s
}
```

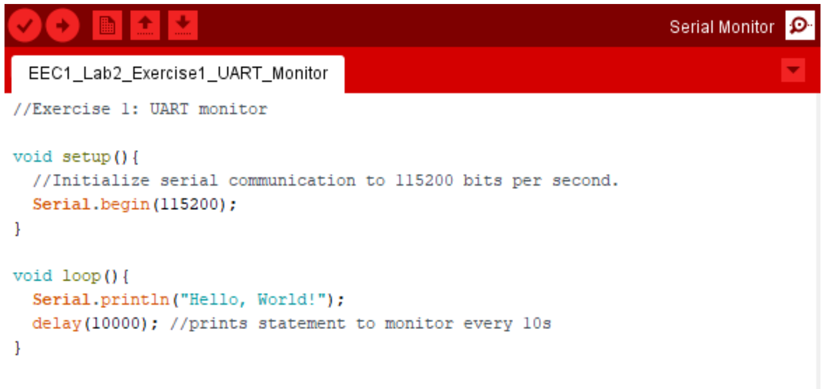
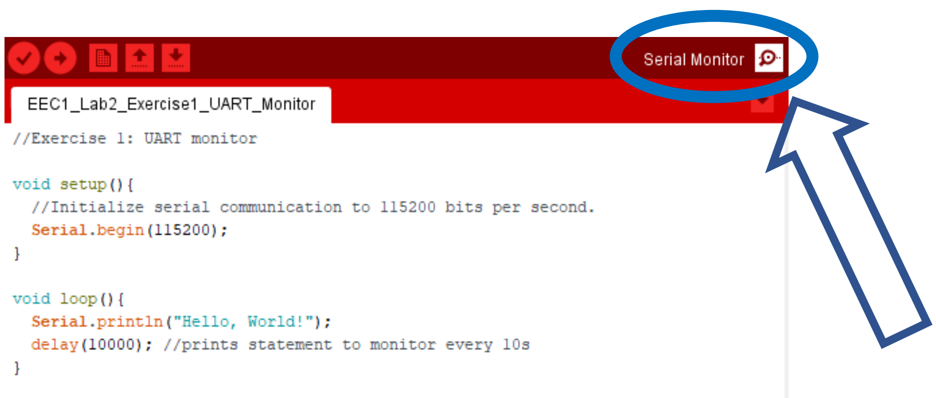


Fig. 2. Code for exercise 1

The first command, `Serial.begin()` in the setup block tells the MCU that we will be engaging in serial communication with it and we specify the rate of communication between us using what is known as a baud rate. In our program, we are specifying a baud rate of 115200 which corresponds to a maximum serial transmission of 115,200 bits per second.

The second command, `Serial.println()` sends a message that you will be able to view in the serial monitor in a moment. This function displays the message or contents of what is given to the function on screen and then jumps to the next line. This command will be important for the remaining exercises and future labs.

2. Before executing the code, open the UART monitor shown in the image below and set the baud rate in the drop-down menu to 115200. This screen is what you will use to display any communication.



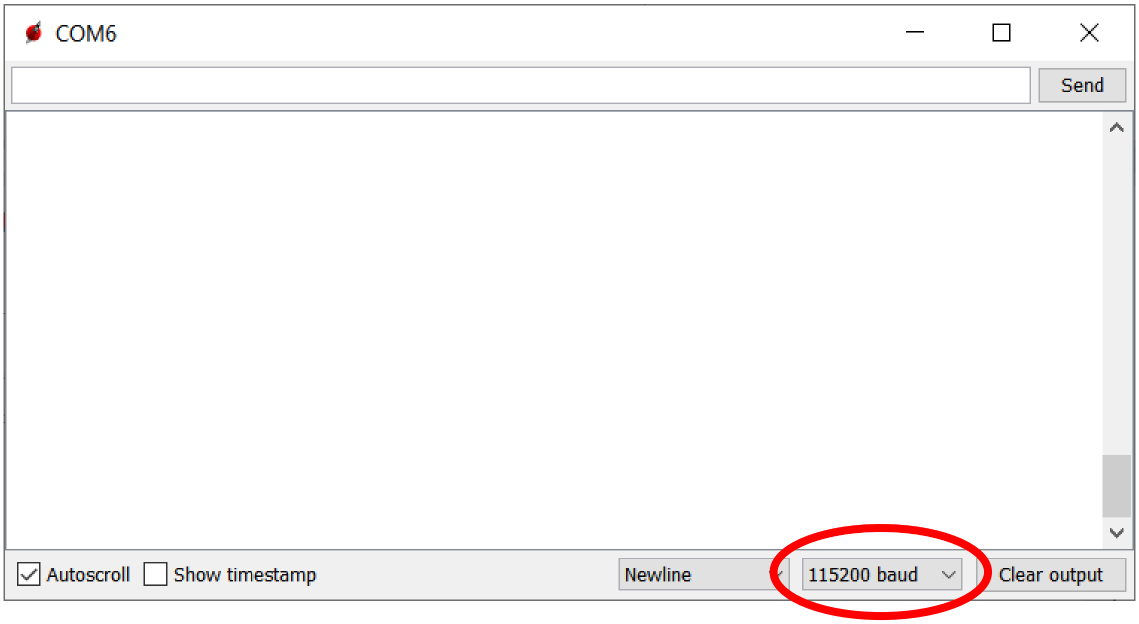


Fig. 3. Image to show how to open up the serial monitor and select the correct baud rate

3. Execute the program and you will see your message print to screen. It may take a few seconds before it appears as the program is starting up. After the first print, the code is set to delay for 10000 ms before executing again which is to ensure your UART monitor isn't filled with the print statement.

Hint: if you don't see anything, make sure your baud rate is correct, you may also have to try another COM port.

<u>Part II:</u> Print the colors switched between during exercise 2 from lab 1

<u>Procedure:</u> Using the information in part A, go back to exercise 2 from lab 1 where you wrote a program to switch between LED colors on the MCU board. In your code, first initialize serial communication in the setup block. In the loop block, have your code only switch back and forth between the red LED and the green LED. While each LED is on, print the word "Red" or "Green" to the UART monitor depending on which LED is on.

<u>Exercise 2:</u> Measure a Voltage and Monitoring it with the UART

<u>Description:</u> In this exercise you will be reading in a voltage from a variable resistor known as a potentiometer (trim-pot) using the analogRead() function and print it's value to the UART monitor. A trim-pot is a three pin device with a knob that allows you to change the resistance. The knobs you see in many electrical devices are actually one of these trim-pots and are used in embedded systems to determine the intensity of a parameter set by the user. Thus, we can see that while this is a simple system, it is also a very practical one with wide applicability. Remember it next time you increase the volume on the radio.

Fig. 4. Precision trim-pot

<u>Procedure:</u>

1. Build the circuit illustrated below and copy the code that follows into Energia. The blue square in figure 5 is a depiction of the potentiometer shown in figure 4.

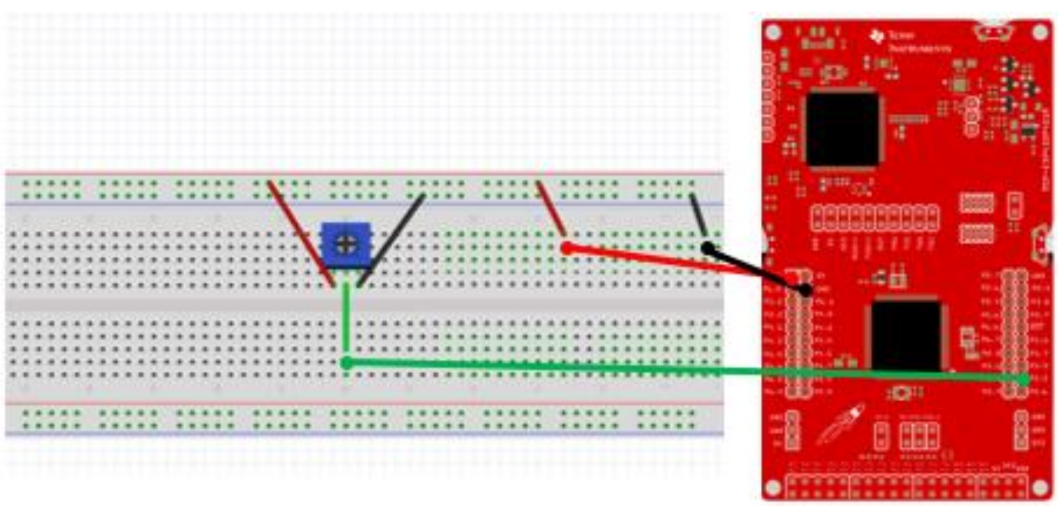


Fig. 5. Diagram of circuit for exercise 2

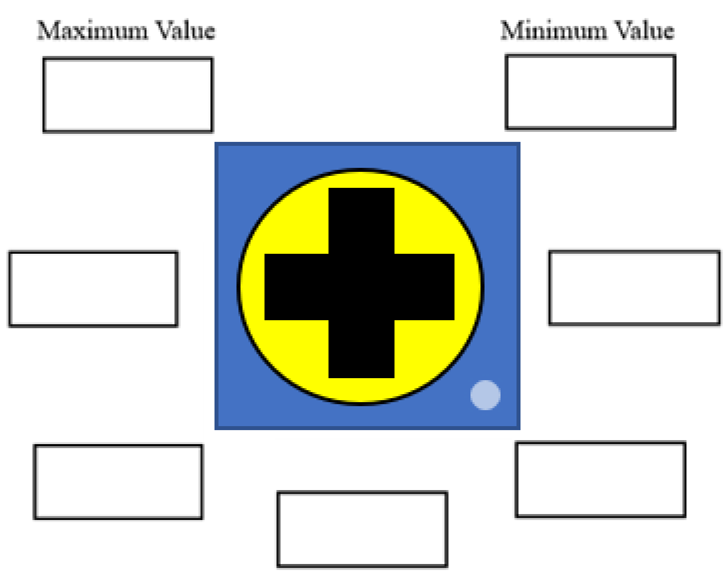
```
//Exercise 2: Measure a Voltage and Monitoring it with the UART

void setup() {
  Serial.begin(115200);
}

void loop() {
  int sensorValue = analogRead(A3);
  //print out the digital equivalent of the value read
  Serial.println(sensorValue);
  //1 second delay between readings
  delay(1000);
}
```

2. Execute the program and open the UART monitor. If you have done the previous step correctly, the maximum value you should read when manipulating the trim-pot knob is 1023. Now answer the following questions on Canvas:

QUESTIONS:

1. The analog input reads values into the analog-to-digital converter (ADC) which converts an analog signal into a discrete set of binary values. Given the maximum value observed on the UART monitor and our discussion of binary numbers in the lecture, how many bits is the signal converted into?

2. In a number of cases, we would like the response of the system to an input such as a knob, to be linear. This means that when the knob is all the way to one end it should be off, when it is at the other end it should be at maximum intensity, and when it is in the middle it should be at half intensity. However, this is not always the case. Below you will see a drawing of the potentiometer at different positions. Place the knob in each of these positions and record the value displayed on the UART monitor. Is the trim-pot linear?



Exercise 3: Using your ADALM2000 (M2K) as a voltmeter

Links:
Scopy software downlink: Link
Video tutorial for ADALM2000 (M2K) and Scopy Software: Link

Description: This exercise is arguably the most important of the lab. It is standard when working with electrical circuits to measure the currents and voltages at different components. In a laboratory setting, you will have easy access to tools such as voltmeters and oscilloscopes, however, here you will use the M2K's voltmeter function.

Procedure:
1. Download the Scopy software compatible with your computer from the link provided above. Run the .exe file to install the software. Be sure to check the box that says Install drivers for ADALM2000.
2. Watch the video tutorial to familiarize yourself with other features of the ADALM2000.
3. Connect your ADALM2000 via USB port shown in Figure 6 below.

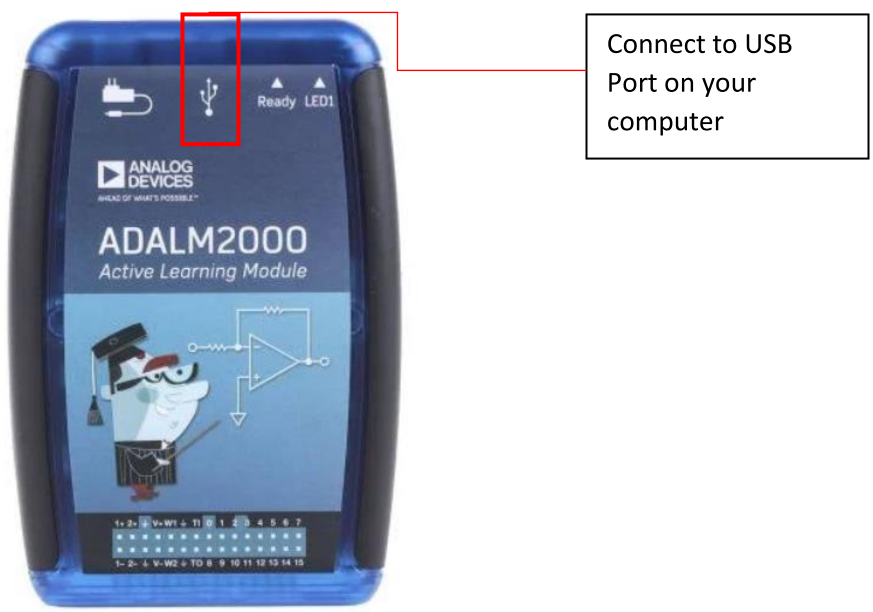


Fig. 6. ADALM2000 (M2K)

4. IMPORTANT: Connect and calibrate the ADALM2000 while it is not connected to any circuit or device. Open the Scopy software on your computer, your M2K icon should appear. Then click on Connect which will also calibrate the ADALM2000. The left-hand side lists the tools available (see Figure 7.)

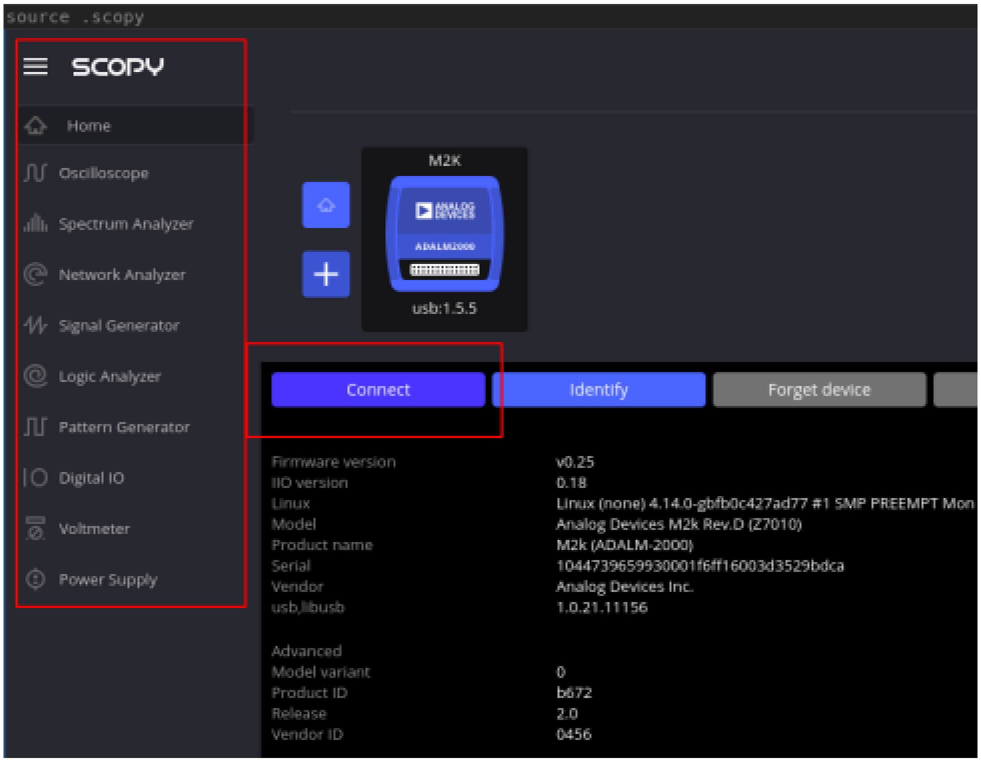


Fig. 7. Scopy Connect and Tools Menu

5. There are two wire harnesses in your ADALM2000 kit. Attach the smaller of the two on the left side of the ADALM2000 as shown in Figure 8a to the highlighted pins shown in Figure 8b.

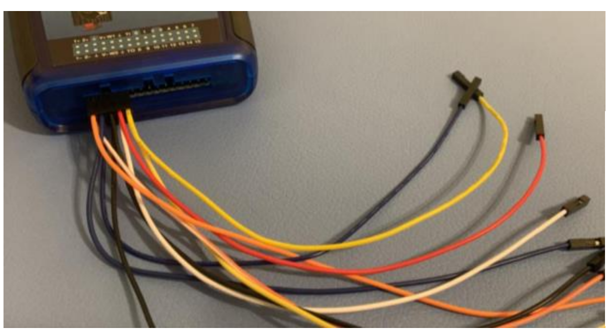


Figure 8a. ADALM2000 extension wires. Figure 8b: ADALM2000pins and labels.

The 10 pins on the left side are the analog pins on the M2K. Below is a description of each pin.

1+ and 1-	A positive and negative probe for an Analog Input (voltage) for Channel 1
2+ and 2-	A positive and negative probe for an Analog Input (voltage) for Channel 2
V+	A voltage source capable of up to 5 volts above the computer's ground reference
V-	A voltage source capable of up to 5 volts below the computer's ground reference
W1 and W2	Two analog output signals, each relative to the computer's ground

6. Modify your circuit such that the trim-pot and the jumper wire was connected to the analogRead() pin are no longer included. From the ADALM2000, connect the 1- pin to ground while the 1+ pin will be your voltage probe that you will place at different locations of the circuit to measure the voltage. This is shown more explicitly in Figure 10.

Remember to use the wire extensions that plug into the ADALM2000 pins so that you have a solid mechanical connection to all pins. Do not use jumper wires to connect directly to the pins, use the extensions as shown in Figure 8a. Also make sure to use the headers that came with your ADALM to make solid mechanical contact between the M2K wires and the breadboard. See Figure 9.

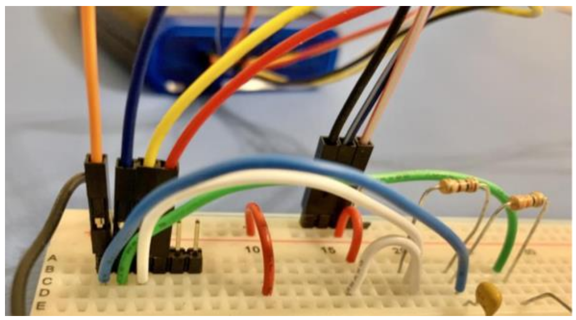


Fig. 9. Extension wires plugged into headers on a breadboard.
(Note: this photo is NOT a circuit from this class.)

2. To test your voltmeter, construct the following simple circuit on your breadboard:

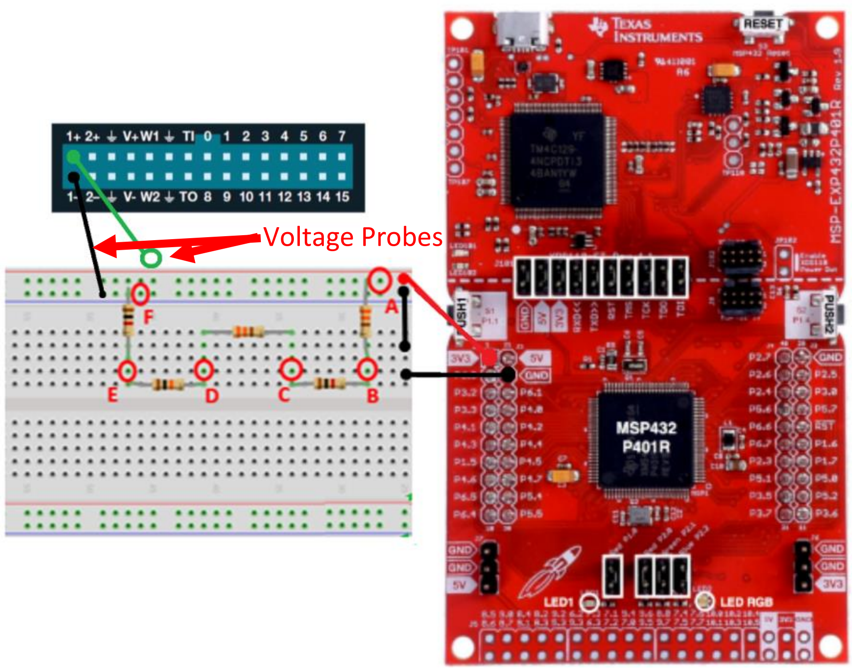


Fig. 10. Diagram of circuit to test voltmeter. The green wire from the ADALM2000 1+ pin hanging is the voltage probe.

Notice the green and black wire dangling from the ADALM2000. These are your voltage probes that you will be using for the following step. Each end of the resistors is labeled points A through F. It is often useful to measure the difference in voltage from one end of the device to the other to see how much energy was consumed by the resistor. In this step you will place the voltage probe at different labeled points and record the difference in the table shown below.

Fig. 11. Scopy Voltmeter Settings

3. In Scopy, click on the Voltmeter tab then apply the settings shown in Figure 11. On the righthand side, under Channel 1, select DC (Direct Current), set Range to Auto, turn off History. To test that your voltmeter is working properly, if you measure the value CD (voltage at C - voltage at D). To do this, connect your 1+ pin (green wire voltage probe) to point C and your 1- pin (black wire voltage probe) to Point D. Then click Run on the top right of the Voltmeter to measure the voltage. If your voltmeter reads ~0.08 volts, it is working properly.
Repeat this step by moving the 1+ and 1- pins to the measurement points to complete the table below.

Questions: Answer the following question and complete the table.

1. Complete the table below:

Measurement Points:	Voltage:
AB	
BC	
CD	0.08
DE	
EF	
AC	
CF	
AF	

<u>Exercise 4:</u> LED Testing

In this lab, you will need to test three light-emitting diodes (LEDs): red, blue, and green. You will be using the ADALM2000 as a voltmeter and the MCU as the power source throughout this exercise. LEDs take in electrical energy and then turn it into light. The electromagnetic spectrum, seen below, shows how the color of the wave is related to how much energy is in the wave.

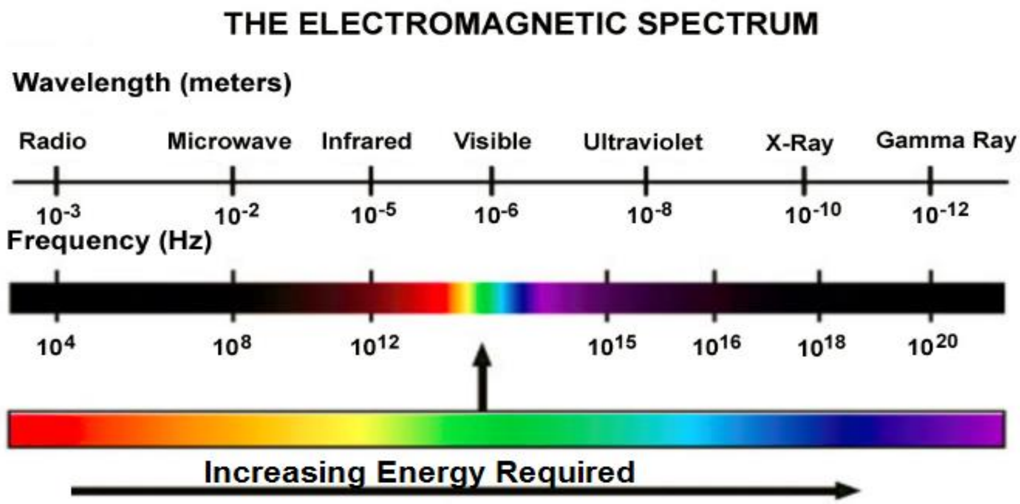


Fig. 12. Diagram of the electromagnetic spectrum

Step 1: Setup a circuit as shown in the figure below and run Scopy's Voltmeter. The circuit will use the 3.3V supply from the microcontroller and will be connected to the potentiometer in order to scale the voltage between 0 and 3.3V.

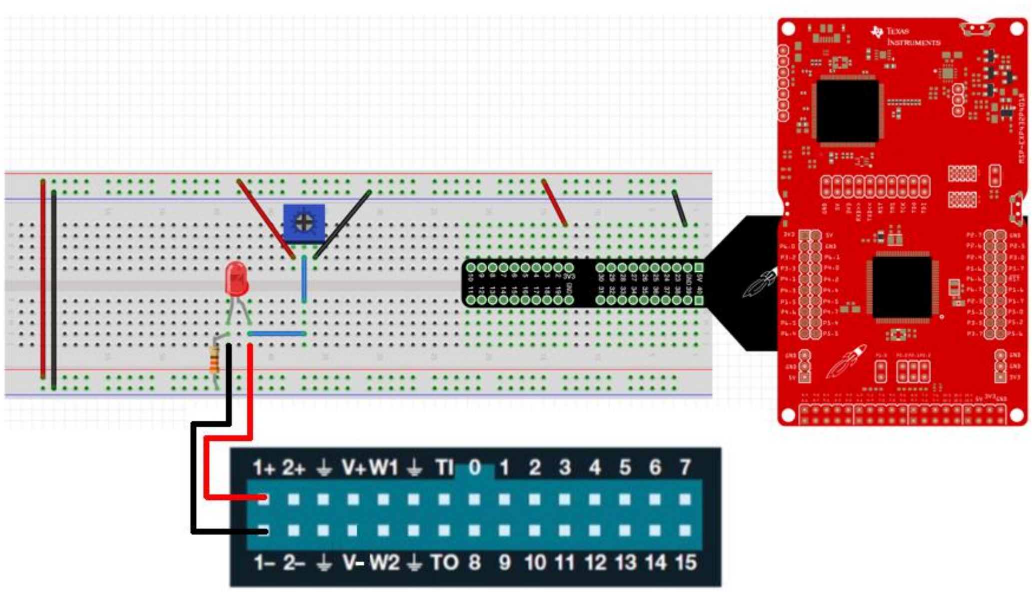


Fig. 13. Diagram of circuit for exercise 4

Step 2: Turn the potentiometer such that 0V is being applied to the LEDs. Now slowly increase the voltage delivered to the LEDs. In this exercise, we are interested in what the value of the voltage is as soon as the LED starts producing light.

SUBMISSIONS: Complete the table and questions below.

1. Write down the exact voltage values when the LEDs start producing light (i.e. you can first see it).

LED	**RED**	**GREEN**	**BLUE**
On Voltage			

2. Which of the colors of light requires the least energy according to the diagram of the electromagnetic spectrum? Which requires the most?

3. Which LED required the least voltage to turn on? Which required the most?

4. Why do you think the LEDs turn on at different voltages?

5. What changes do you notice in the LEDs as the voltage increases above the "On Voltage"?

6. Now, connect two RED LEDs in ***parallel*** as shown below. Connect the M2K's 1+/1- pins in parallel with the LEDs. Do you notice any changes in the turn on voltage or the intensity vs. when just one LED is connected? What are the changes? And why do you think they occur (or don't occur)?

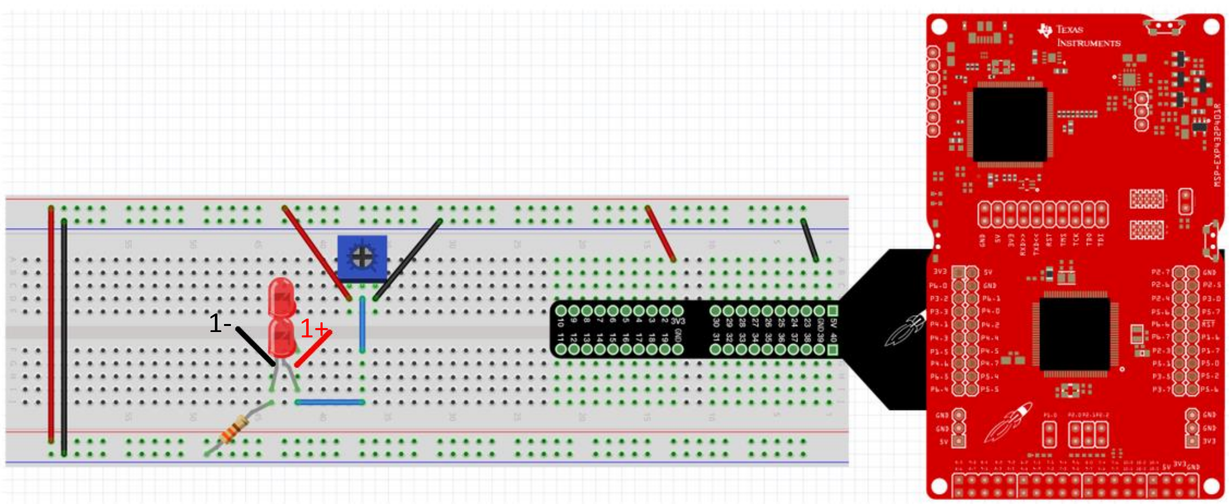


Fig. 14. Diagram for LEDs connected in parallel

7. Instead of connecting them in parallel connect them in ***series***. Note: for this question, change over from the 3.3 V supply to the 5 V supply, as highlighted in the red box in figure 15, and be careful not to turn the potentiometer knob too high as it can cause damage to the circuit. Now what changes do you notice in either intensity or turn on voltage? Again, explain why.

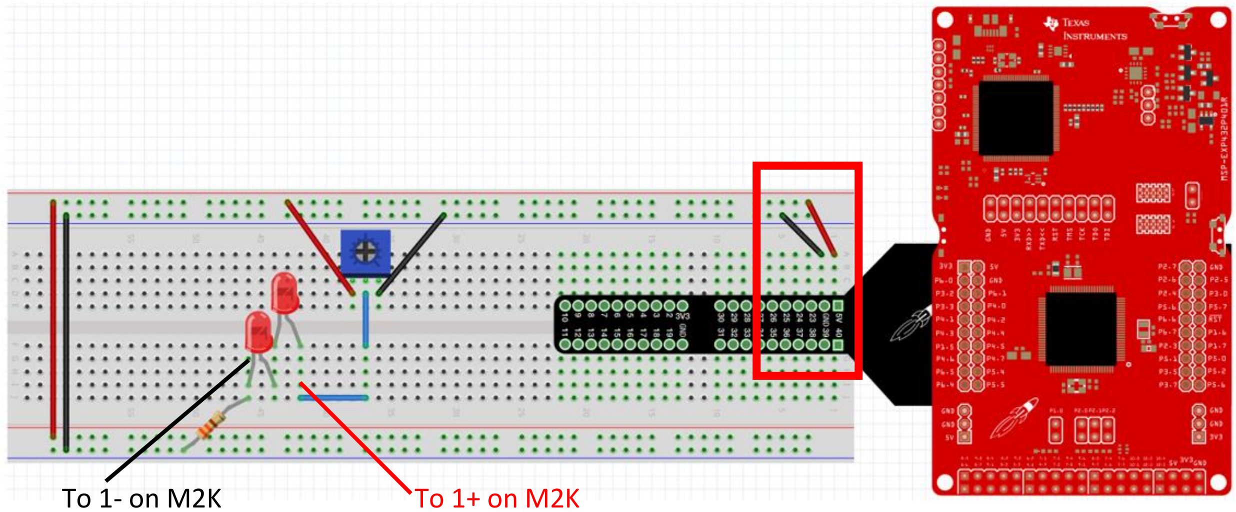


Fig. 15. Diagram for LEDs connected in series. Note the change from a supply of 3.3V to 5V