Experiment 2

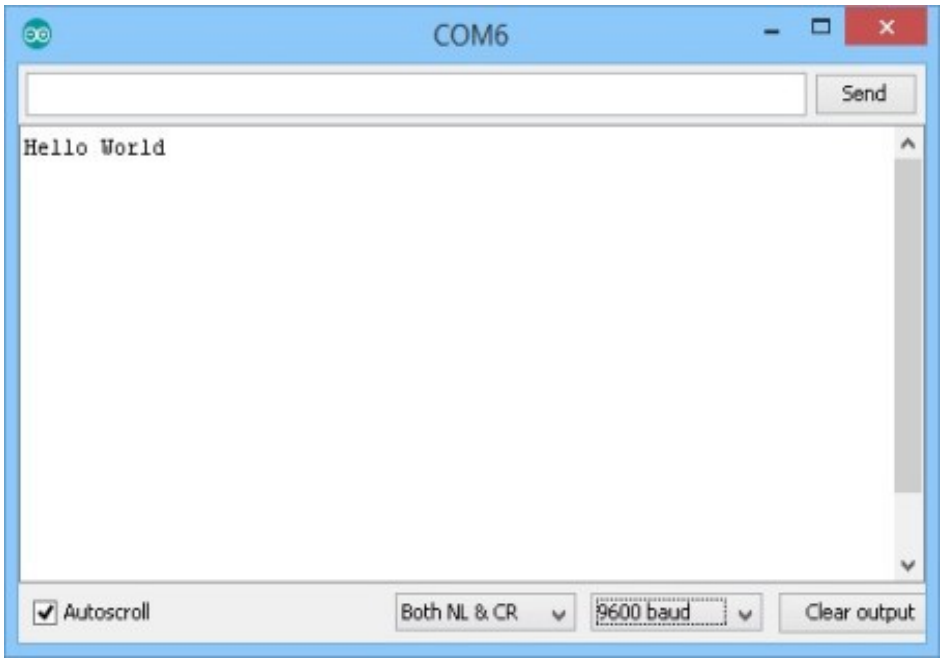
Reading a Potentiometer / Utilizing Varying Values

Outline

In this experiment, it is expected from you to,

1. Get familiar with the concept of serial monitor
2. Learn how to read values (from an analog pin)
3. Learn the potentiometer structure and usage
4. Assembling the circuit with potentiometer
5. Test the circuit

1. Serial Monitor



Serial Monitor

Serial monitor used to visualize serial communication of Arduino board. Serial communication used between the Arduino board and a computer or other devices. All Arduino boards have at least one serial port (also known as a UART or USART), and some have several.

So serial monitor can be used to,

1. Visualize the output coming from Arduino board
2. To send to Arduino board as input

To be able to fully utilize the serial monitor, you need to setup the serial communication. For that,

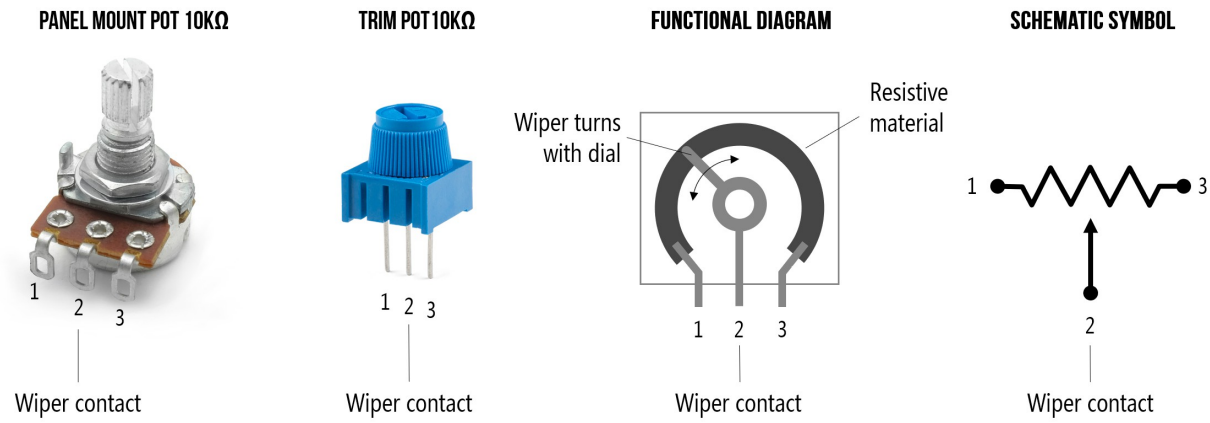
1. You need to assign your communicaiton's baud rate by using begin(int baudRate) funtion during setup
2. Use serial appropriate functions such as println(), read()
3. Set your serial monitor's baud rate same as in the begin() function

2. Reading Value From Analog Pin

The ATmega controllers used for the Arduino contain an onboard 6 channel (8 channels on the Mini and Nano, 16 on the Mega) analog-to-digital (A/D) converter. The converter has 10 bit resolution, returning integers from 0 to 1023. While the main function of the analog pins for most Arduino users is to read analog sensors, the analog pins also have all the functionality of general purpose input/output (GPIO) pins (the same as digital pins 0 - 13).

Analog pins are specialized pins to input/output analog signals. They have different naming system from digital pins starting from A0. To be able to read an input value you need to use analogRead() function.

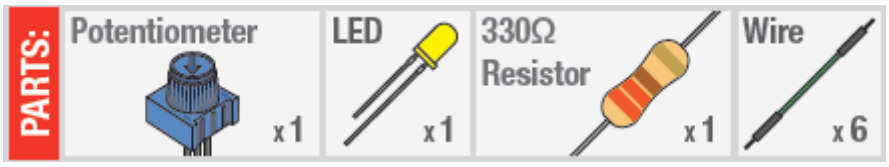
3. Structure of Potentiometer



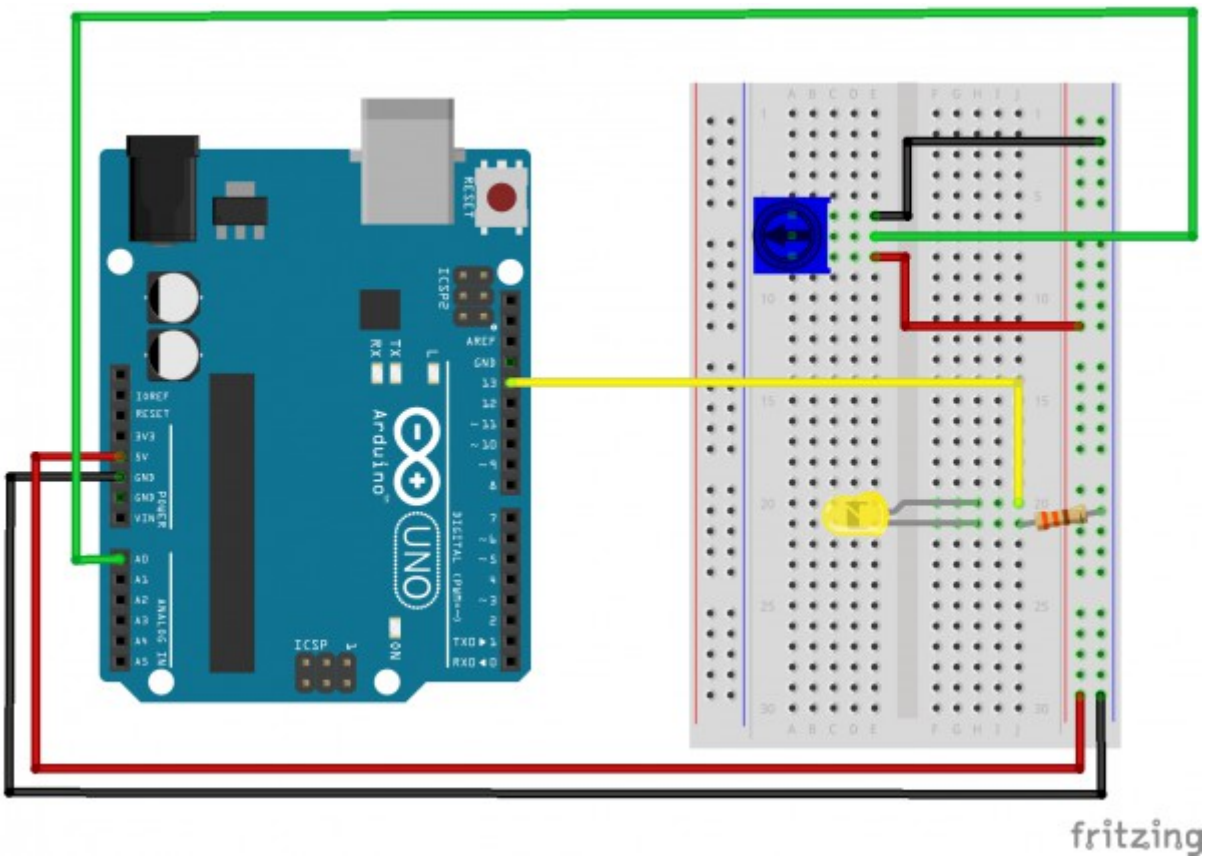
Structure of Potentiometer

Potentiometers are 3 terminal resistors which used to get variable resistance values. As shown in the structure by using any other terminals with the terminal numbered with 2 will make you achieve this.

4. Assembling the Circuit



Required Parts



Fritzing Diagram of the Circuit

1. Select your resistor (330 Ω) by using the color code table
2. Connect your LED according to anode/cathode direction
3. Connect your potentiometer and care to not insert the terminals on shared strip
4. Verify and upload your code to the arduino board
5. Observe the result and compare it with the expected outcome

Expected Outcome: LED should be in turned on and turned off state for some time *t* for both state continuously (*Blinking*) and the duration *t* will vary according to potentiometer's value.

5. Testing

1. Measure the minimum and maximum values of your potentiometer by using multimeter.
    - Since there are 3 terminals on potentionmeter you need to select 2 of them each time and note the values for each pair.
    - For testing purpose chose one of the fixed terminals (numbered with 1 and 3 on structure display) and measure the resistance value by completing the circuit with the varying terminal (numbered with 2 on structure display). Note the value you are getting and then switch the fixed terminals (1 -> 3 or 3 -> 1). Note the new value again. Comment on your findings.
2. You already learned that values we get from analog pins are mapped between [0 1023]. Try to remap this range to [500 2000] so we can use the potentiometer value to determine blinking speed for our own choice.

Modification

User 3 LEDs to encode 8 bits [000, 001, 010, 011, 100, 101, 110, 111]

111: All LEDs are in on state. Potentiometer should be at it's maximum value.

000: All LEDs are in off state. Potentiometer should be at it's minimum value.

Print your potentiometer value to serial port.