

CS Makers – Variable Blinking LED

For this project we will build on what we learned from the “Blink a LED” circuit we built previously. For this project, we will add a control circuit to our one LED circuit. The control circuit will vary the rate of blinking. As before, we will need to build the circuit and then write a control program.

You will need one (1) LED of any color. You will also need one (1) resistor. You need a resistor that is at least 150 Ω (Ohms) and no more than 330 Ω . Remember it is important to use resistors to make sure that we don't damage either the LED or the Micro:bit. The circuit diagram shown in Figure 1 illustrates Pin 0 of the Micro:bit is connected to a LED and Pin 1 is connected to a potentiometer.

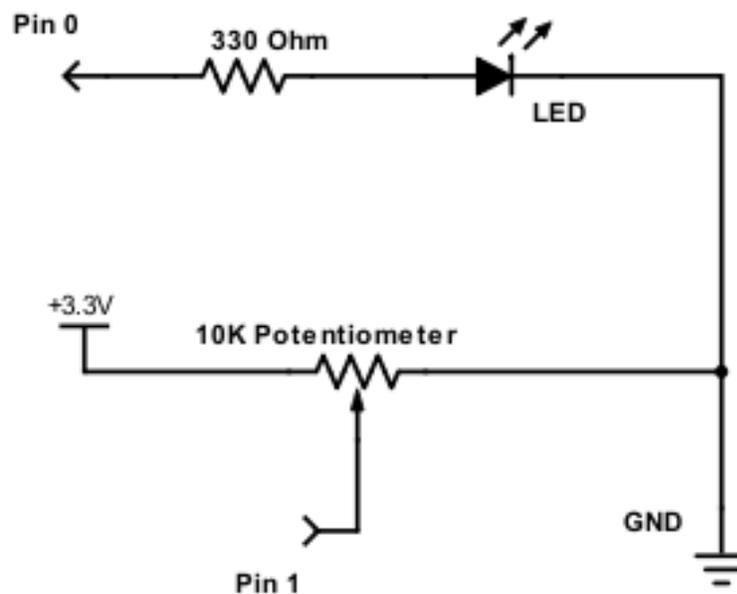


Figure 1. Potentiometer and LED Circuit

Parts required:

- 1 Micro:bit
- 1 LED
- 1 330 Ω resistor
- 1 10K potentiometer
- 1 Breadboard
- 4 Alligator (crocodile) clips and 5 hookup wires (jumpers)
- 1 Battery pack and batteries –OR– USB cable for power

Remember that wires and resistors will conduct electricity in any direction. This means it does not matter which way you connect alligator clips (a.k.a. crocodile clips) or resistors. However, a LED will only conduct electricity in one direction. The schematic shown in Figure 1 tells us which way to connect the LED once you know which wire on the LED is marked positive (+) and which one is marked minus (-). Figure 2 shows us how to distinguish the wires on a LED. Notice the longer wire is the positive.

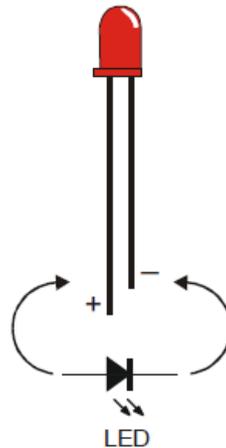


Figure 2. Light Emitting Diode

The holes in the breadboard are exactly the right size for LEDs, resistors, jumper wires, and other electronic parts. Remember from the *Traffic Light* activity that some of the holes are connected to each other so that we can attach electronic components together without soldering, using electrical tape, or other fasteners. However, not all the holes are connected together and knowing which ones are connected and which ones are not is important so that our circuits will work properly. Figure 3 shows our breadboard.



Figure 3. Breadboard for prototyping circuits

You can find out more about how breadboards work and how to use them to prototype circuits in the *Traffic Light* activity or on the Sparkfun web site <https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard>.

Assemble the circuit on a breadboard as shown in Figure 4. You should use alligator clips to connect the wires to the Micro:bit. You must wire the circuit exactly as shown on the breadboard. Notice in Figure 4 that the resistor and potentiometer are placed across the ravine of the breadboard so that one end of the resistor is on either side (remember the two halves of the breadboard on either side of the ravine are NOT connected). Note the potentiometer has three (3) connections so pay attention to the orientation. Figure 4 shows which rows and columns are used on the breadboard with faint green shading. In this image, the negative power rail is blue and is used to connect the negative side of the LED to GND. The positive wire on the LED is connected to the resistor (see Figure 1).

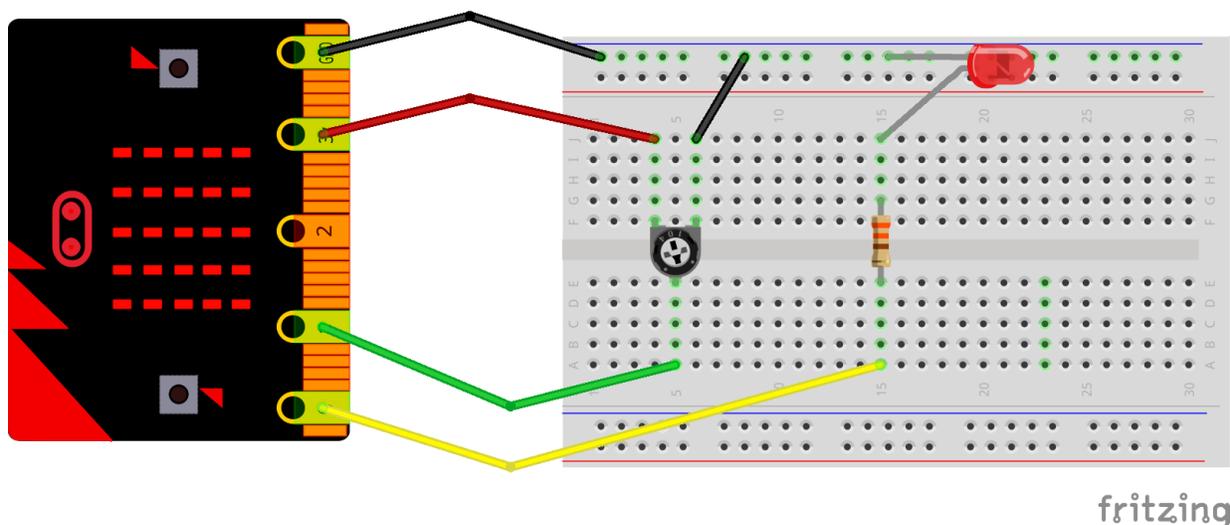


Figure 4. Variable Blink LED circuit

Now we are ready to design our program to control our LED with the potentiometer. The program is similar to our previous project where we blinked a LED on and off. However, this time we will use the potentiometer to control how fast the LED blinks.

A potentiometer is a kind of resistor that can be changed. By turning the knob on the potentiometer, we are changing the resistance. We will connect one side of the potentiometer to the power pin (labeled 3V) on the Micro:bit and the other side to Pin 1 of the Micro:bit. In our program, we will read how much voltage is entering Pin 1 and use that value to calculate how long to wait between blinks of the LED.

Our program will need to initialize two variables, one to store the potentiometer voltage and one to store the blink rate. Our program will have two (2) **forever** blocks. One forever block will read the voltage at Pin 1 then compute the blink rate and the other forever block will blink the LED. Figure 5 shows our algorithm as a flowchart.

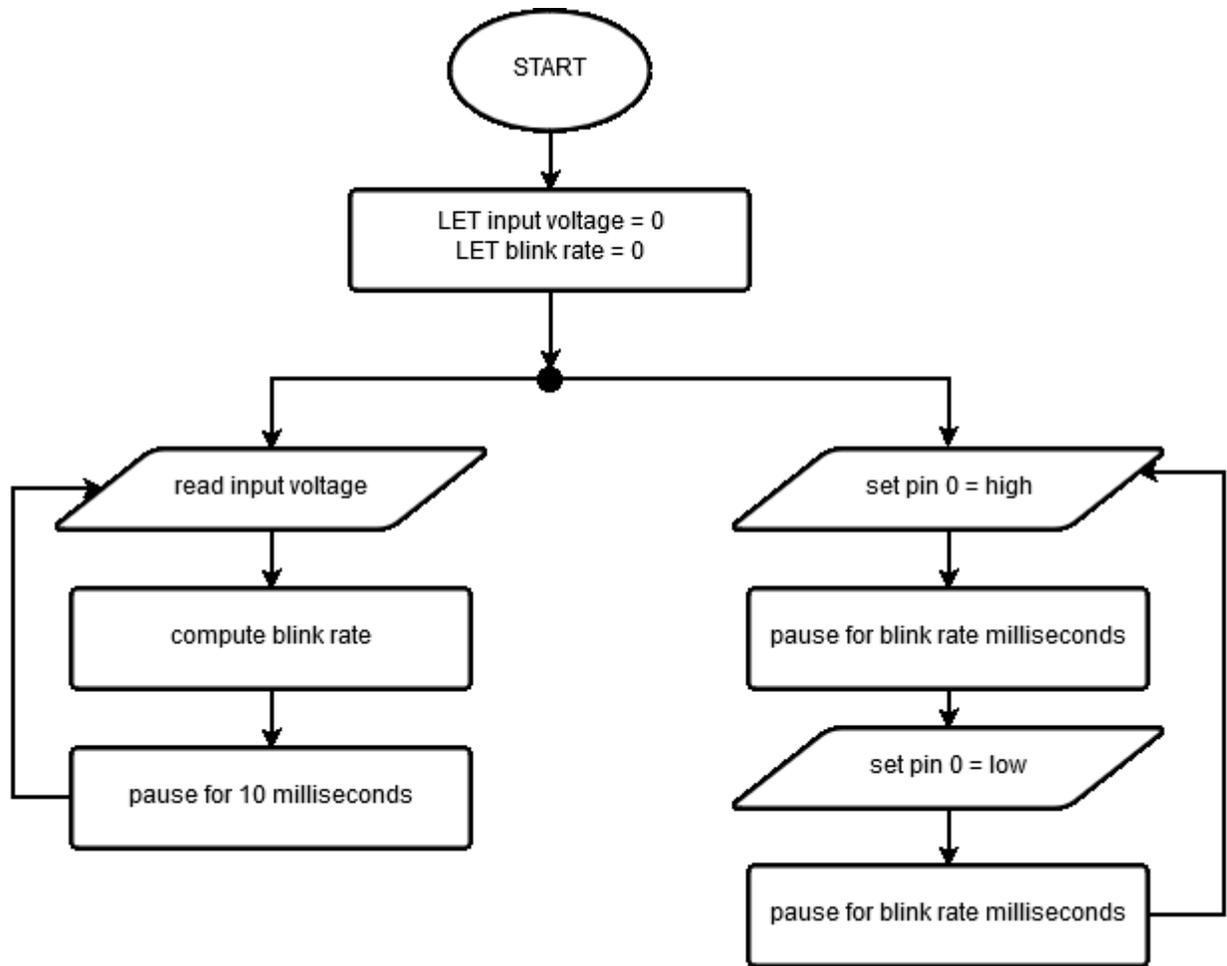


Figure 5. Variable Blink LED algorithm

Open up the Javascript Blocks Editor for the Micro:bit and create your program. Recall that most microcontroller programs have two sections, a start (or initialization) section and a forever section (the part that repeats forever). We need to initialize two variables in the **start** block. The remainder of our Micro:bit program will be inside two **forever** blocks.

Recall that on and off (which is the same as high and low) is often denoted by the numbers one and zero in computer science. On or high is denoted as the number one and off or low is denoted as the number zero.

0 (zero) = Off

1 (one) = On

We will use the same block that we used to toggle the single Pin in our “Blink a LED” program to toggle (turn on and off) our LED in this program. Access the “Pins” drawer under “Advanced” to find the digital write pin block (see Figure 6).



Figure 6. Digital write block in the Pins drawer

In addition to the “digital write pin” block you will also need the “pause” block and the “analog read pin” block. These are all in the “Pins” drawer.

Create your program using the flowchart shown in Figure 5 then follow the same steps you learned in the “Introduction to the Micro:bit” lesson to load your program into your microcontroller. A picture of the completed circuit is shown in Figure 7. Once your program is loaded and running, try turning the potentiometer to change how fast the LED blinks.

If your program does not work as expected, carefully review your circuit to make sure it matches Figure 4 and verify that your program matches Figure 5. The complete program is shown on the next page in Figure 8.

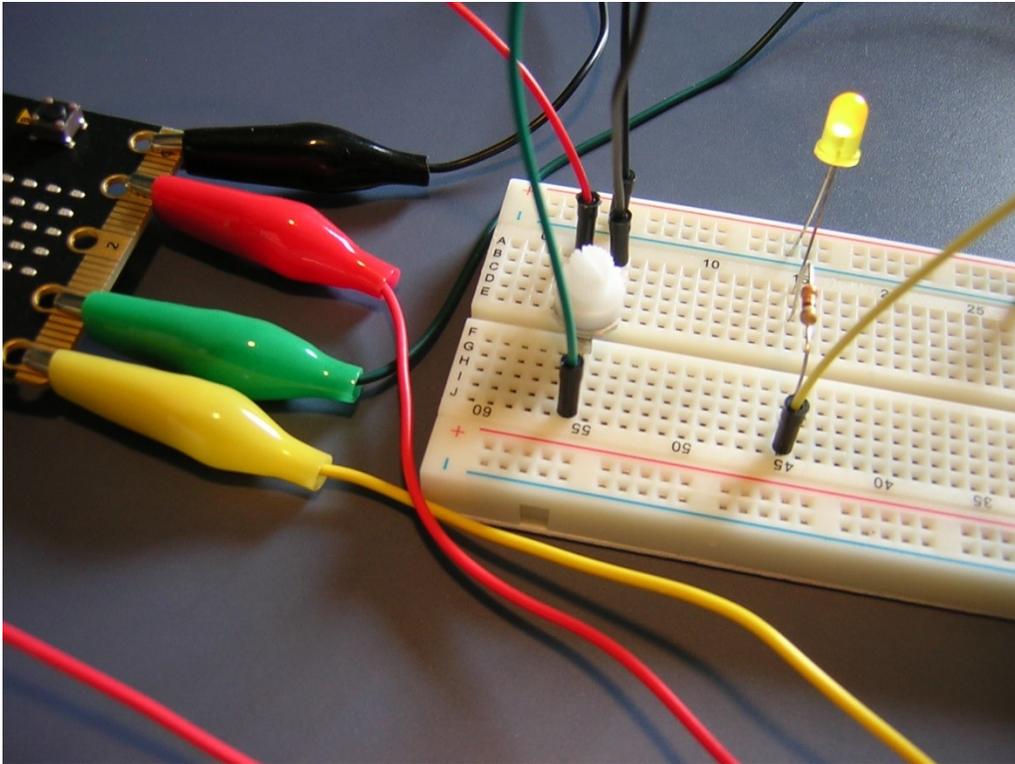


Figure 7. Completed Circuit on Breadboard

OPTIONAL – Theory of Operation

How it works: Three volts is fed into the potentiometer from the 3V pin on the Micro:bit. As the potentiometer knob is turned (a.k.a. wiper), the resistance changes across the potentiometer. Turn the knob all the way to one end and you have maximum resistance and nearly zero volts exiting the potentiometer. Turn the knob all the way to the other end and you have minimum resistance and nearly 3 volts exiting the potentiometer to Pin 1. The amount of voltage exiting the potentiometer ranges from zero to 3 and is used as input to Pin 1. The Micro:bit has a circuit called an Analog-to-Digital-Converter (ADC) on Pin 1. The ADC converts the analog signal (voltage) into a digital signal (an integer in the range 0...1023). The “analog read pin” block is used to access the value from the ADC. The converted value, as an integer, is added to 100 ms so the fastest our LED will blink is 100 ms and the slowest it will blink is 1,123 ms.

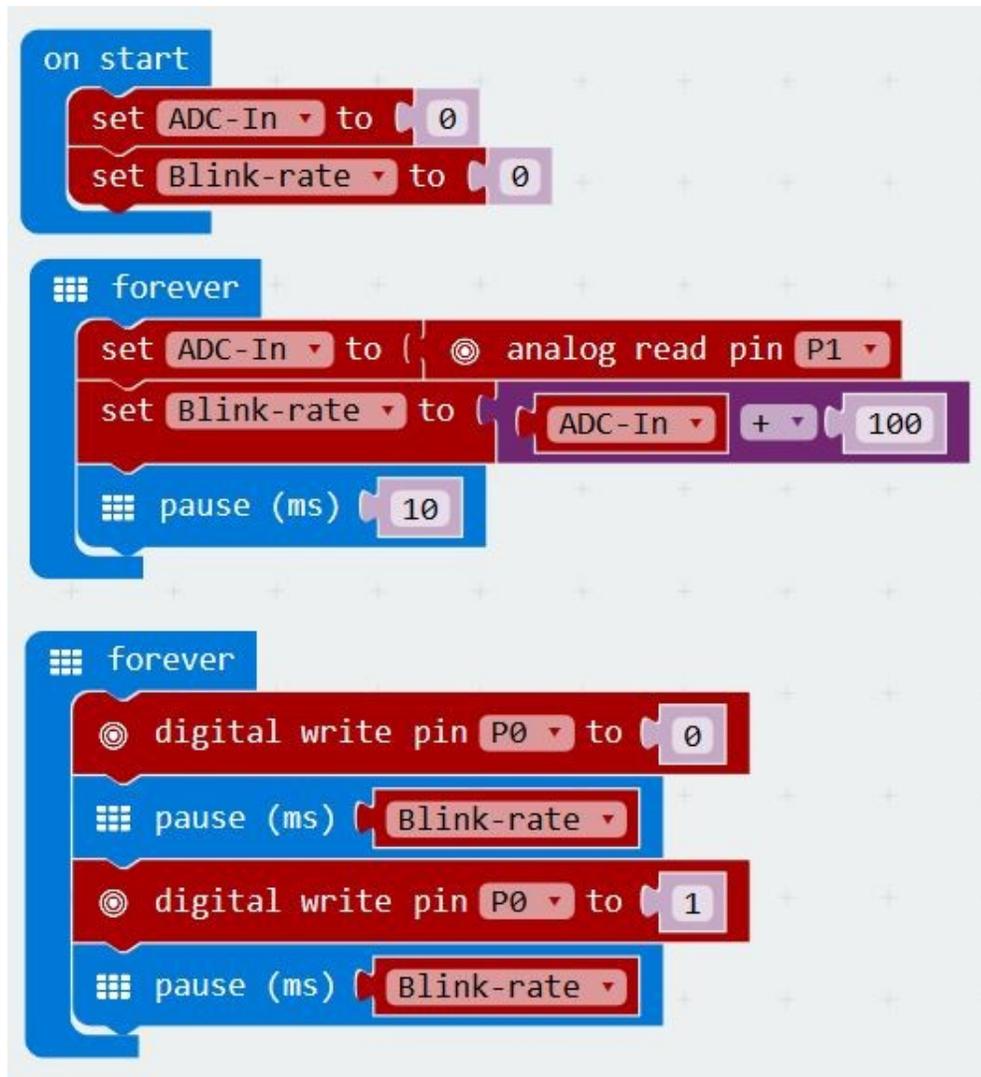


Figure 8. Variable Blink a LED program

CS Makers - The Alabama Alliance for an Inclusive Middle Grades Computer Science Preparation through Makerspaces in the Alabama Black Belt Region
NSF award #1744467

Computer Science topics – Algorithms, flowcharts

Figure 2 - <https://learn.parallax.com/tutorials/robot/shield-bot/robotics-board-education-shield-arduino/chapter-2-shield-lights-servo-17>

Figure 3 – <https://www.sparkfun.com/>

Figure 6 - <https://microbit.hackster.io/anish78/traffic-light-system-using-bbc-micro-bit-da2f47>

All other figures and text James A. Jerkins, Ph.D., CISSP