Name: _____

Decisions, Decisions

Professor Bob Brown – Kennesaw State University
Bob.Brown@Kennesaw.edu

Sequence and Selection

Our code blocks have run one after another. That's called *sequence*, and it's the natural way that computers work.

Computers can also make decisions. That's called *selection*, because we *select* what to do next instead of just doing the next thing in sequence. The most common way of doing selection is with an IF statement or IF block. You can find the IF blocks in the "Logic" bin of the MakerCode screen.

Red Light / Green Light

Today we will make a device that can display a red light or a green light under the control of the player. You could use this in a game instead of calling out "Green light!" or "Red light!" The other players would have to watch closely and pay attention. We will use some of the parts in the kit to build the device. We will also write a program. As always, our program needs an algorithm.

Algorithm

- When the program starts, turn the red light on.
- If button C on the button pad is pressed:
  - Turn the red light off.
  - Turn the green light on.
- If button E on the button pad is presses:
  - Turn the red light off
  - Turn the green light on.

Assemble your Equipment

In addition to the Micro:Bit and USB cable, you will need the following:

- Break-out board: This makes it easy to connect to the pins on the Micro:Bit.
- Three red, yellow, and black jumper (connector) wires.

KENNESAW STATE UNIVERSITY
COLLEGE OF COMPUTING AND SOFTWARE ENGINEERING

Last update: 2020-03-01

- Button pad: You will use this to control the LEDs
- Red and green LEDs.

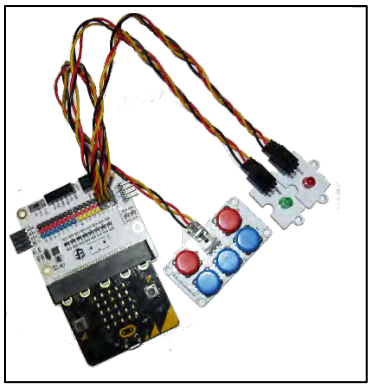
The breakout board has numbered pins; some are black, red, and yellow and some are black, red, and blue. We will use the black, red, and yellow pins and match the colors of the jumper wires to the colors on the pins.

The Micro:Bit plugs in to the break-out board with the A/B buttons and the LEDs facing up. There's a sketch on the breakout board to remind you. When the Micro:Bit is plugged in correctly and the battery or USB cable is attached, the power light on the break-out board will light up.

The ends of the jumper wires are different. One end is plain and goes to the break-out board. The other end has a tab, and goes to devices like the button pad and the LEDs.

Wiring Your Red Light / Green Light Device

- Connect the tab end of a jumper wire to the red LED; it will only go one way.
- Connect the other end to pin 2 on the break-out board. Match the wire colors with the colors on the pins.
- In the same way, connect the green LED to pin 1 on the break-out board.
- Connect the button pad to pin zero.



We wired the "inside" pins first because it's easier that way.

Your assembled device will look like the picture.

Pins and Devices

We humans think about buttons and lights, but the Micro:Bit only knows about pin numbers, so we have to remember which pin is connected to which device. I've made a little table to help us remember that:

Device	Pin
Button pad	0
Green LED	1
Red LED	2

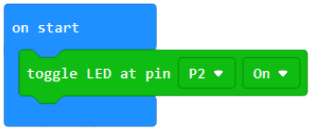
The Red Light / Green Light Program

Go to https://makecode.microbit.org and click the New Project button. Name your project "redgreen."

To use the parts from the Tinker Kit, we have to load some extra blocks. In the Parts Bin, click *Advanced*. Scroll down and click *Extensions*. Click *tinkercademy-tinker-kit*. You will have a new set of blocks called </> Tinkercademy. It will be just before *Advanced* and bright green.

This time, we're going to use both the *on-start* and *forever* blocks, so leave them in the workspace.

From our algorithm we know that when the program starts, the red LED should come on. From the table of devices and pins, we can remember that we connected the red LED to pin 2. To make the red LED come on when the program starts, drag a *toggle LED* block from the Tinkercademy bin and place it the *on-start* block. Set the on/off and pin values.



Get an *if* block from the Logic bin and drag it into the *forever* block. It will start out by saying *if true*. From the Tinkercademy bin, get the *if key A is pressed…* block and drag it to replace *true* in the *if* statement. From the algorithm, we know that we want key C, and from the table, we know that the button pad is on pin 0. Set up the *if* block with those values.

Inside the *if* block, we need to do two things. Find out what they are from the algorithm and look up any pin information you may need from the table. Get the necessary *toggle* blocks from the Tinkercademy bin.

Thought Challenges

- What happens if button C is *not* pressed?
- What happens if the green light is already on?

Programming Challenge

Set up button E according to the algorithm. Refer to the table of pins as needed. You will need another *if* block. Be sure it is not inside the first *if* block. (It does have to be inside the forever block.)

Save your program, load it onto the Micro:Bit, and test it.

Vocabulary

Selection: Making a choice of what a program will do next. Today our choice was to do something or not. It's also possible to choose between two or more actions.

Sequence: Doing things one after another, in the order they're written. Doing instructions in the order they're written is the natural order of operation for computers. Special instructions like *if* can change that order.

Remember all this material is on line at drbrown.net/mcnair