

Controlling LED lights by micro:bit: smart lamp code examples

Design and build a smart lamp

This document shows several code examples for controlling LEDs by a micro:bit (microbit.org). The codes are written using the block-based editor MakeCode (makecode.microbit.org).

Controlling a LED by micro:bit buttons

- Figure 1 shows how a LED can be connected to a micro:bit, together with a code to turn it on and off using buttons A and B, respectively. Note that the cathode (shorter leg) is connected to the ground (GND).
- The command 'digital write pin', sets a pin either to digital value 1 (3 V) or to digital value 0 (0 V).
- The micro:bit therefore functions as a programmable switch.

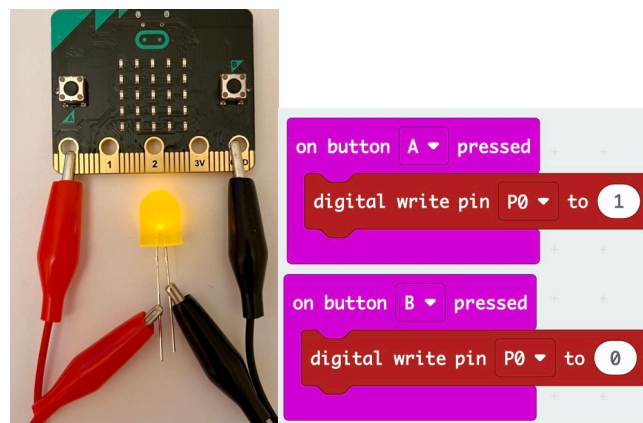


Figure 1: Connecting a LED to micro:bit and controlling it with buttons A and B

Image courtesy of the authors

The code in figure 1 can be modified to switch the LED on and off using only a single button. To do this, the micro:bit can record the current state of the LED in a variable.

This is shown in Figure 2.

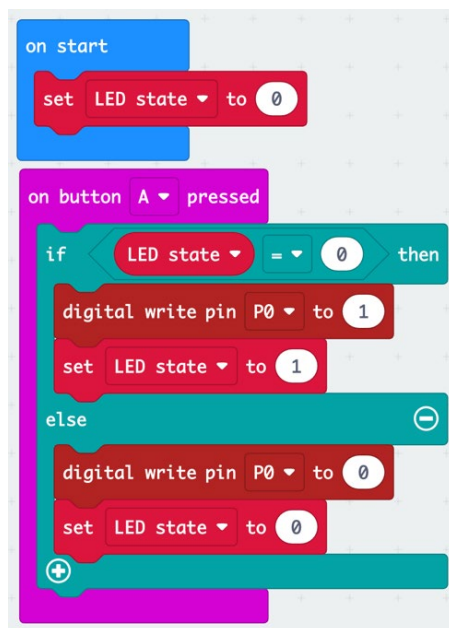


Figure 2: Controlling LED by a single micro:bit button

Image courtesy of the authors

Controlling a LED by environmental light or other sensor values

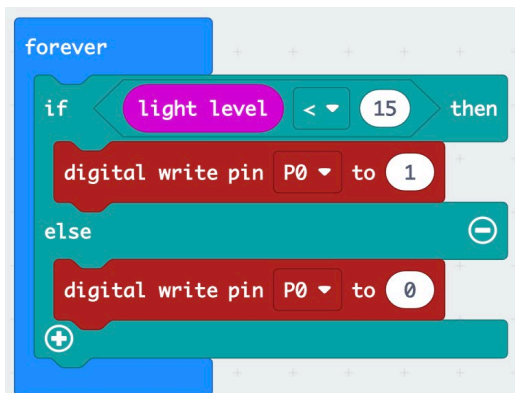


Figure 3: Controlling a LED with environmental light

Image courtesy of the authors

Figure 3 shows a code example that uses the built-in light sensor on the micro:bit to control the LED. If the light level is below a certain threshold (between 0 and 255), the light is turned on. This can be used to make a lamp that turns on automatically when it gets dark.

Similarly, we can use values from the other built-in sensors of the micro:bit to control the LED. This makes it possible to make a smart lamp that responds to changes in temperature, orientation, sound level, etc. Explore the different possibilities in the MakeCode input tab.

Furthermore, the LED brightness can be set to 1024 different levels (0–1023) using the command ‘analog write pin’ rather than ‘digital write pin’. The code example in figure 4 increases the LED brightness as the environmental light level decreases.

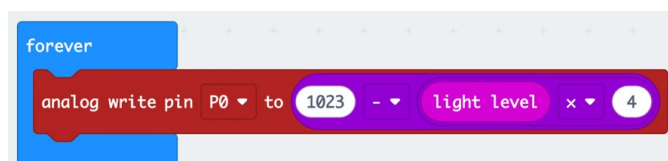


Figure 4: Adjusting LED brightness according to environmental light

Image courtesy of the authors

Traffic light – controlling several LEDs

Several LEDs can be controlled separately by connecting them to different micro:bit pins. Figure 5 shows the connection of three LEDs to act as a traffic light. LED anodes are connected to pins 0–2, while the cathodes are connected together to the ground pin (GND).

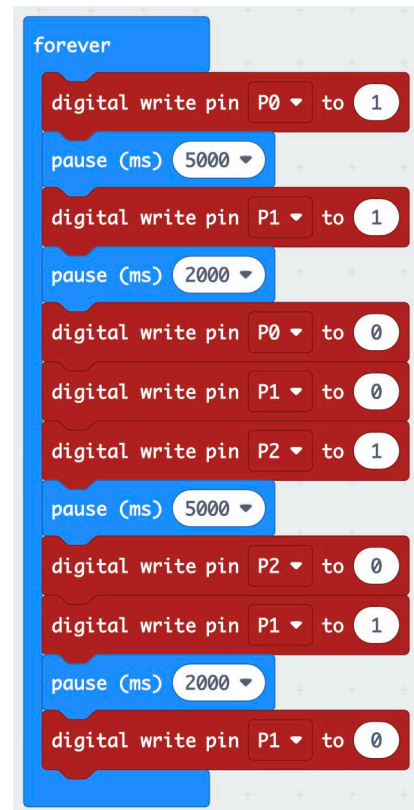
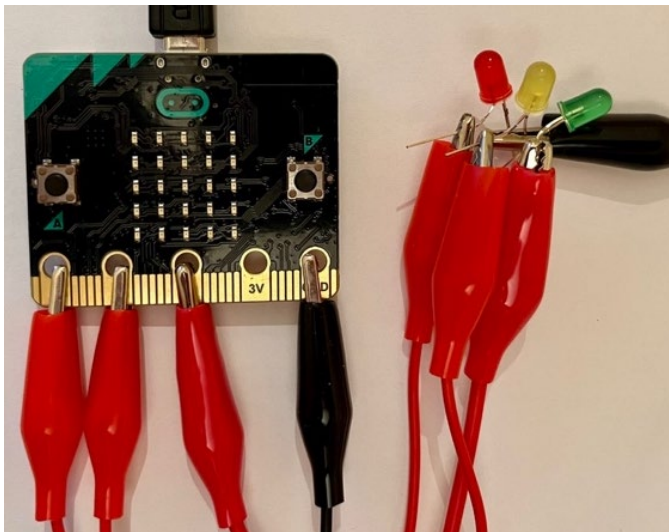


Figure 5: Connecting and coding a traffic light

Image courtesy of the authors

Blinking LED

Figure 6 shows a code that makes the LED blink when the micro:bit is tilted to the left. The blinking frequency can be changed by adjusting the pause duration.

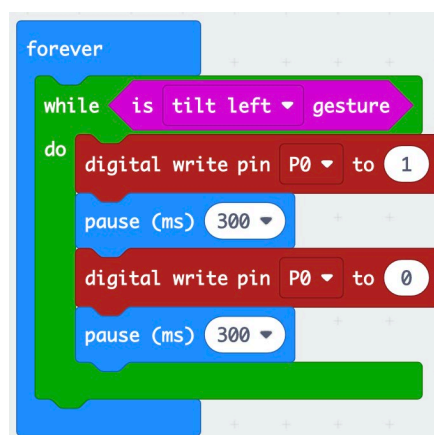


Figure 6: Blinking LED on left tilt

Image courtesy of the authors

Remote controlled LED

We can control a LED remotely by connecting it to one micro:bit (receiver) and controlling it with another one (transmitter) (see figure 7). The remote controller (transmitter) sends two different numbers (1 and 0) to turn the LED on and off, respectively. The receiver code is in figure 8.

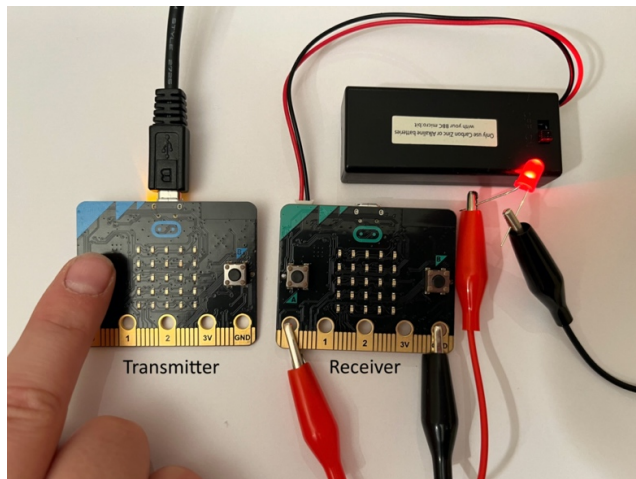


Figure 7: Remote-controlled LED

Image courtesy of the authors

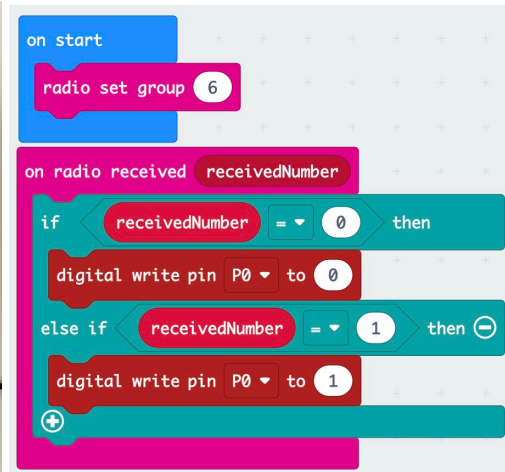


Figure 8: Receiver code

Image courtesy of the authors

Any sensor on the remote controller can be used to control the LED, for example, buttons, tilting, or a light sensor, as illustrated in figure 9. Explore the MakeCode input tab to see all the possibilities.

Note that the transmitter and receiver must operate on the same radio channel to be able to communicate with each other. The channel is set at the start of the code. Any number between 0 and 255 can be used, but each lamp must operate on a different channel to avoid interference. We used number 6 as an example.

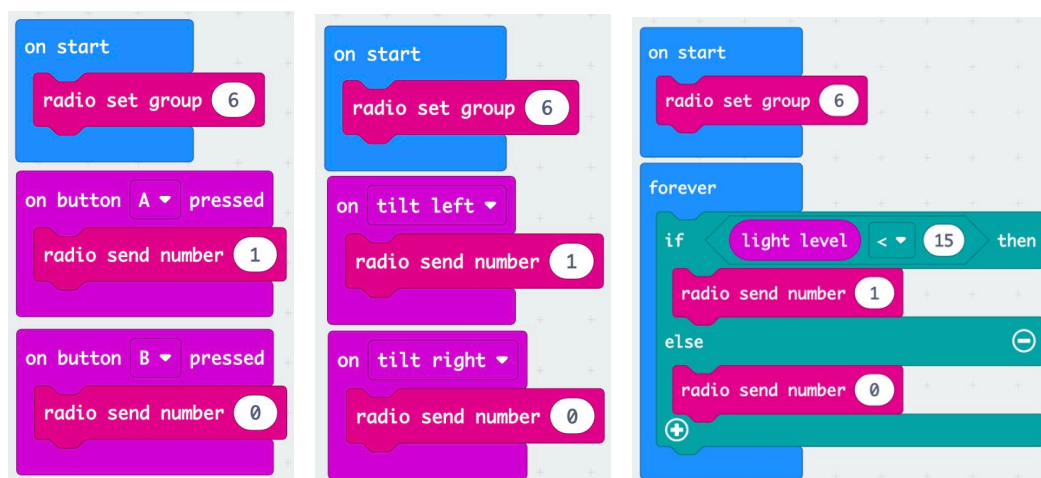


Figure 9: Transmitter codes that use buttons, tilting, and a light sensor to control the LED

Image courtesy of the authors

Alarm clock for the hearing impaired

Figure 10 shows a code for an alarm clock that uses light rather than sound to indicate that the set time has elapsed. Buttons A and B are used to set the sleeping length (8 hours is default). The alarm clock is activated by pressing buttons A and B together.

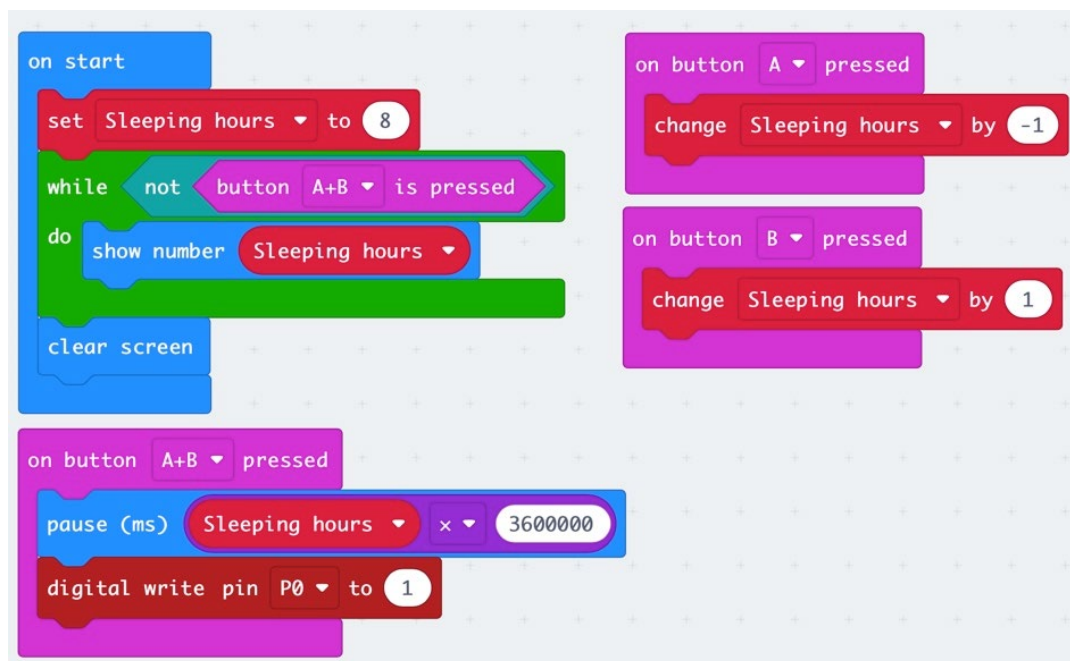


Figure 10: Code for an alarm clock for the hearing impaired. The LED turns on when the set time has elapsed.

Image courtesy of the authors