Programming a Microcontroller with MicroPython

In this exercise you will learn how to run code on a microcontroller, which is a small computer chip that can be used to collect data or control equipment. In this case, you will make the LED on the microcontroller blink in a specific pattern.

A *microcontroller* is a small computer that generally can only run one program at a time, unlike a micro*processor*, which is a small computer that can run multiple programs at the same time. Microcontrollers are useful for collecting data and controlling equipment and other specialized tasks. The program in a microcontroller starts when the power is turned on and keeps running no mater what, which means that it should be able to deal with errors or unusual conditions gracefully. Since the microcontroller may be built in to another device, this kind of computer is sometimes called an "*embedded*" computer or system.

Arduino and Pico

Perhaps the most well-known set of microcontrollers used for teaching and research are named Arduino.[1,2]

MicroPython with Thonny

For this exercise you will want to use Thonny as your IDE. If you have been using IDLE up to now that is fine, but Thonny makes it particularly easy to run Python code on a Pico.

[1] `https://www.arduino.cc/education`

[2] https://en.wikipedia.org/wiki/Arduino

Morse Code

(Explain the basic idea. Explain timing of dots, dashes, and intervals.)

Assignment

To complete this exercise you must write a Python script which runs on a Raspberry Pi or Arduino Pico microcontroller and does the following:

1. When the device is started the LED blinks the Morse code for "H" and "I" and then pauses for 3 seconds. (Optionally, it could blink "Y" and "O" instead.)
2. The LED then blinks on and off 100 times, with the interval between blinks starting at 1.00 seconds and decreasing by 0.01 seconds after each blink.
3. The script waits 7 seconds and then repeats the same 100 blinks, but it does not output the initial Morse code greeting. It waits another 7 seconds, and then repeats again, until power is removed from the device.
4. Demonstrate that your code works when the Pico is not connected to a computer by recording a brief video showing you plugging the Pico into a power source (not a computer) and recording the introductory message and at least two cycles of 100 blinks.
5. Submit your code and video to your instructor.