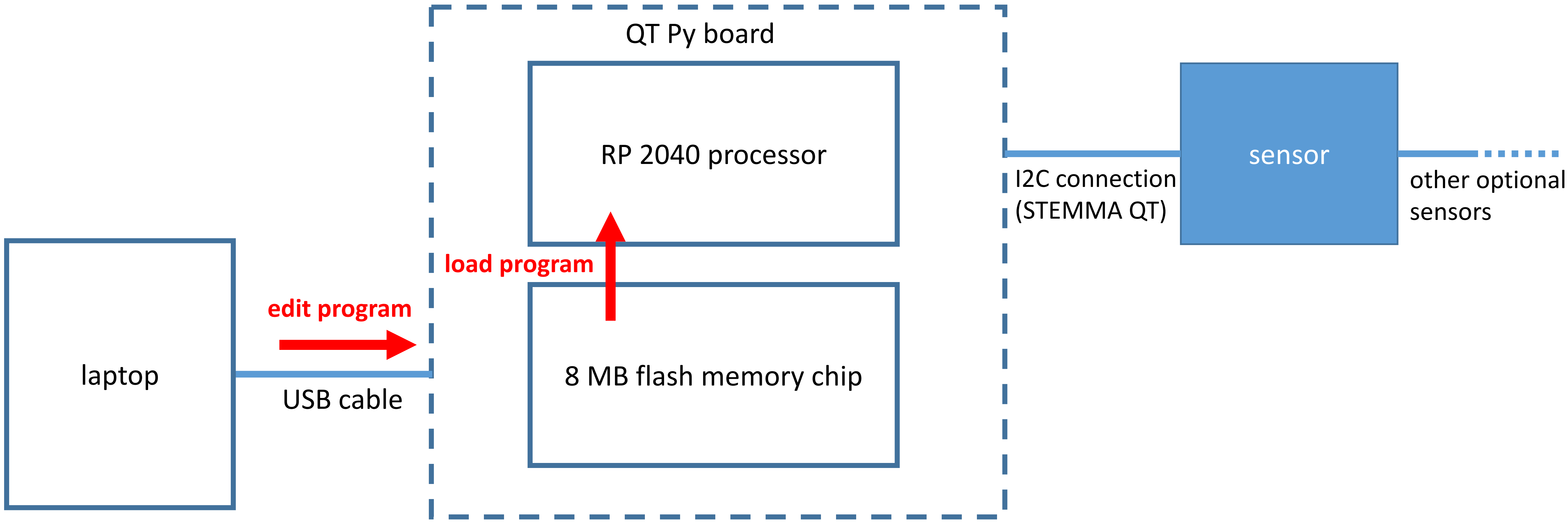
website: vanderbi.lt/codegraf

QT Py RP2040 memory limitations

DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

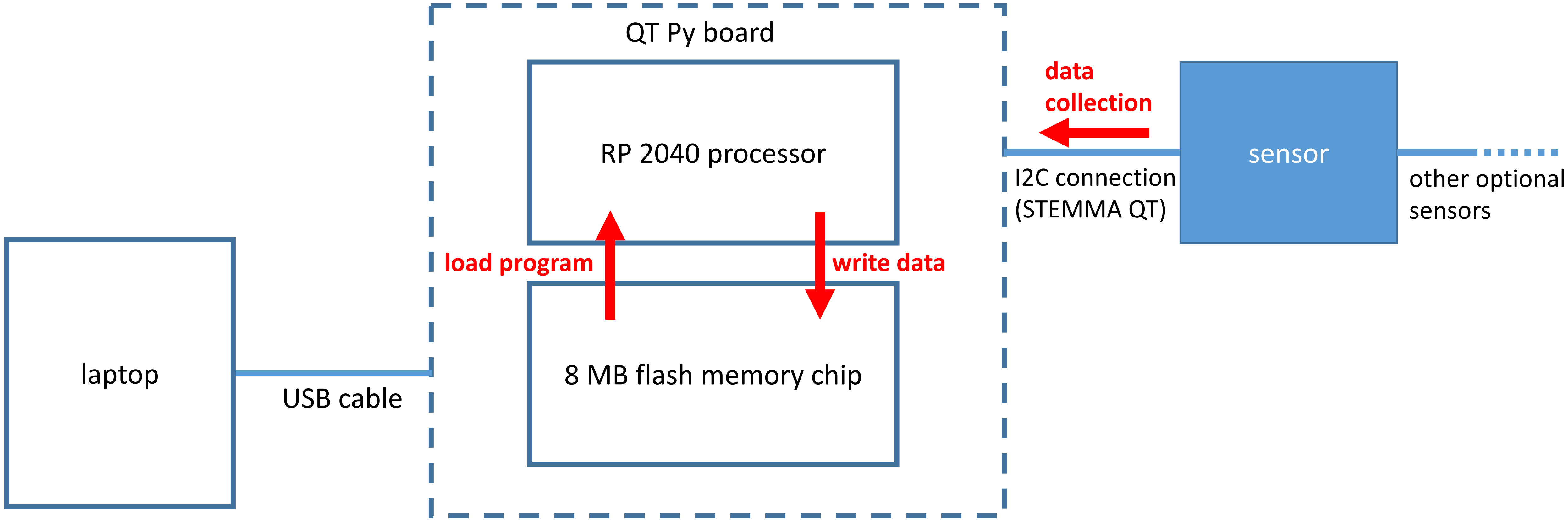
Memory in read only mode (default)



The processor can only read from the memory chip.

The laptap can write to the memory through the USB.

Memory in write mode

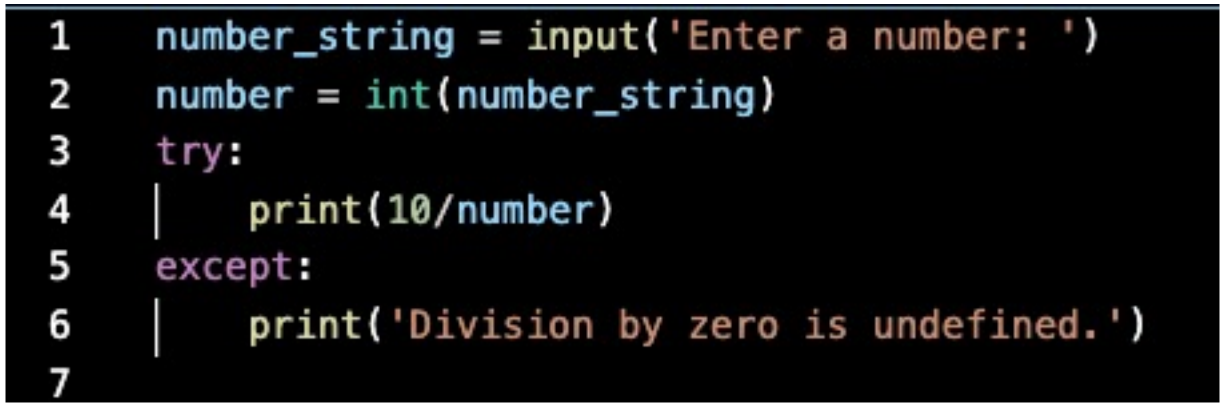


The processor can write to the memory chip.

The laptop cannot access the memory, although it can monitor what's going on through the serial console.

`try` … **`except`** … code blocks

- Normally, when an error is thrown, code execution is terminated.
- If errors are handled, the script can continue to operate.
- A **`try`** code block specifies the code that might throw an error.
- An **`except`** code block specifies the code to be run if an error is thrown.



```
number_string = input('Enter a number: ')
number = int(number_string)
try:
    print(10/number)
except:
    print('Division by zero is undefined.')
```

CircuitPython error trapping code example

define code block where error might occur

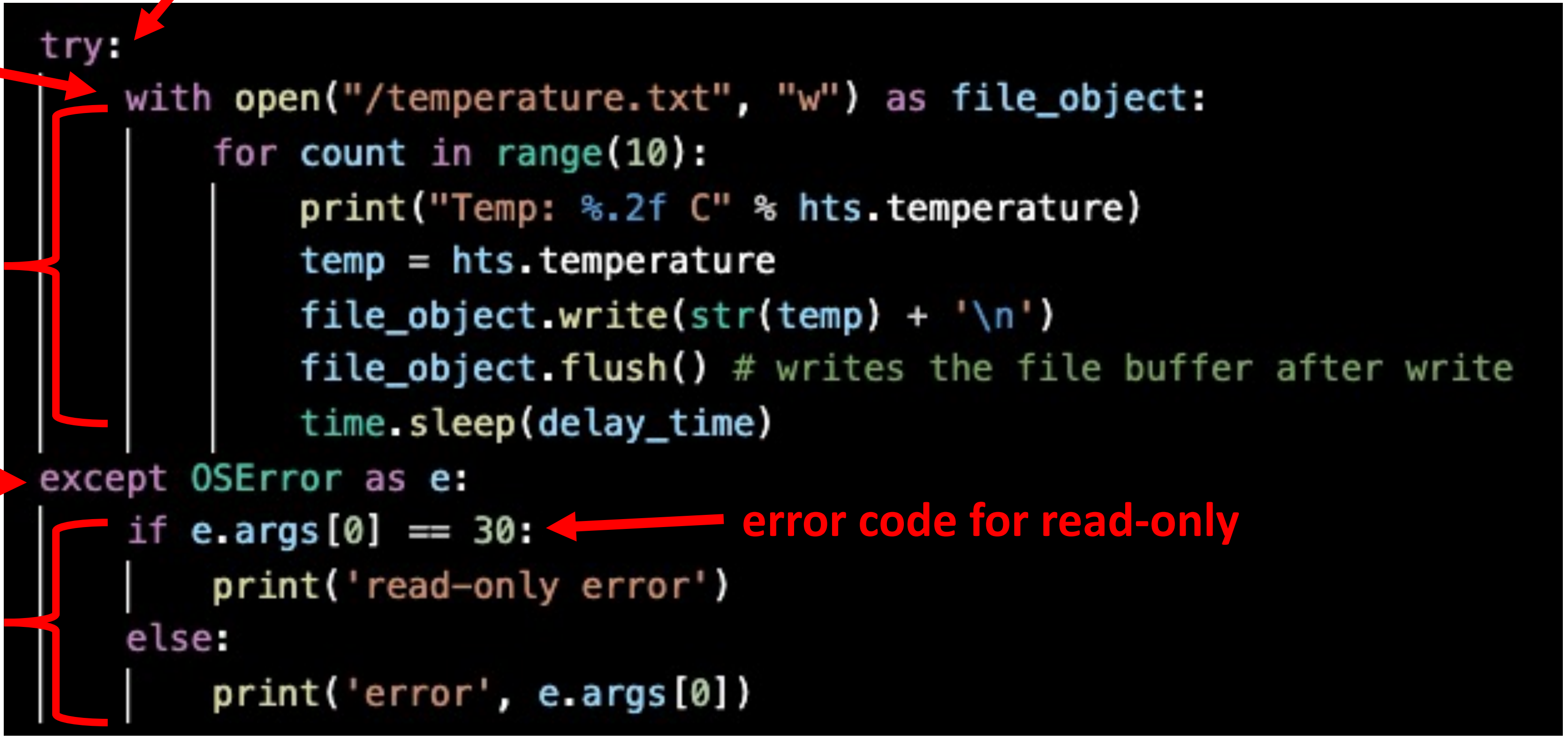
trying to open a file in read-only mode will throw an error

indented code block for `try`

capture the type of error

error code for read-only

error handling code



```
try:
    with open("/temperature.txt", "w") as file_object:
        for count in range(10):
            print("Temp: %.2f C" % hts.temperature)
            temp = hts.temperature
            file_object.write(str(temp) + '\n')
            file_object.flush() # writes the file buffer after write
            time.sleep(delay_time)
except OSError as e:
    if e.args[0] == 30:
        print('read-only error')
    else:
        print('error', e.args[0])
```

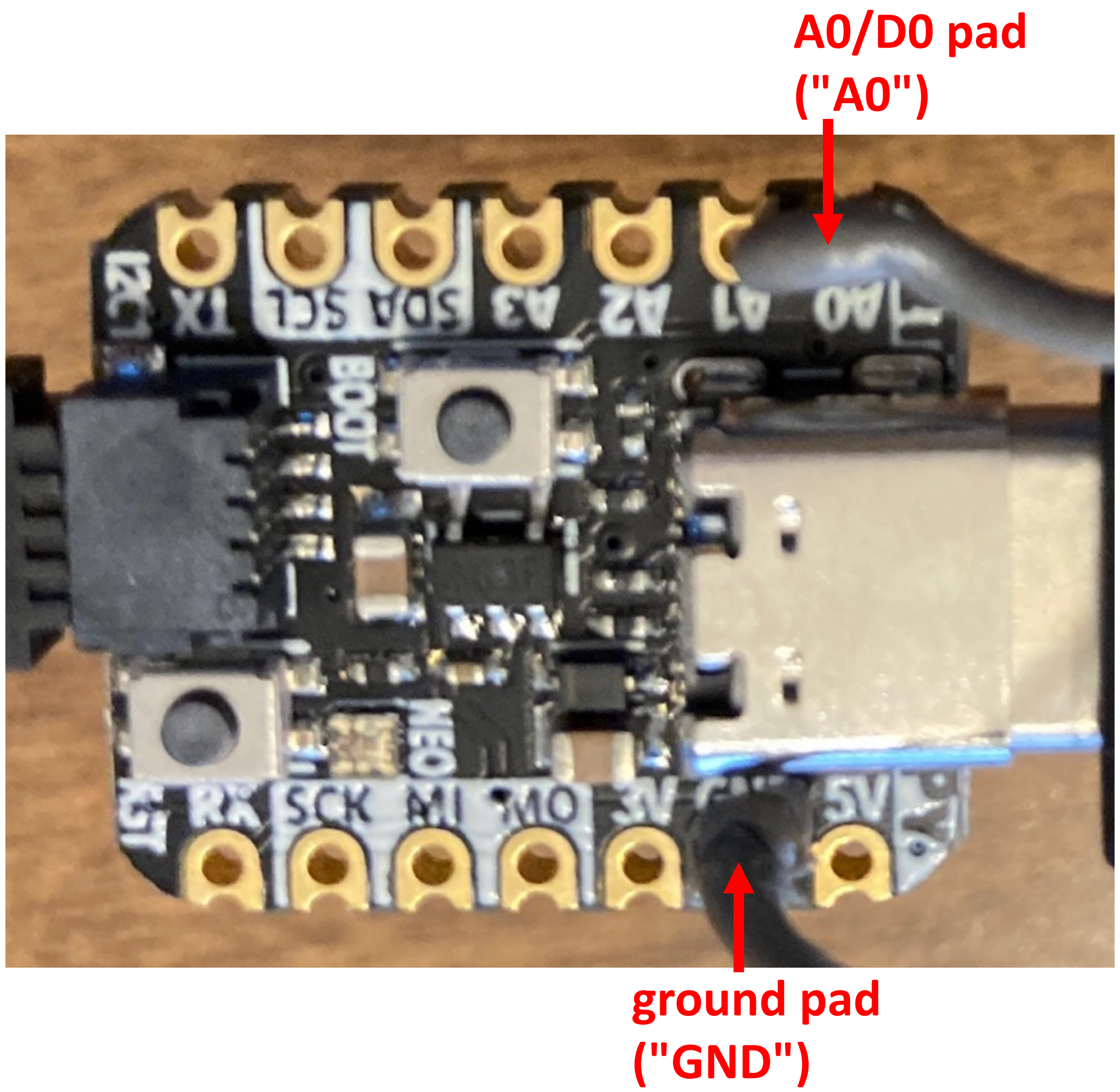
website: vanderbi.lt/codegraf

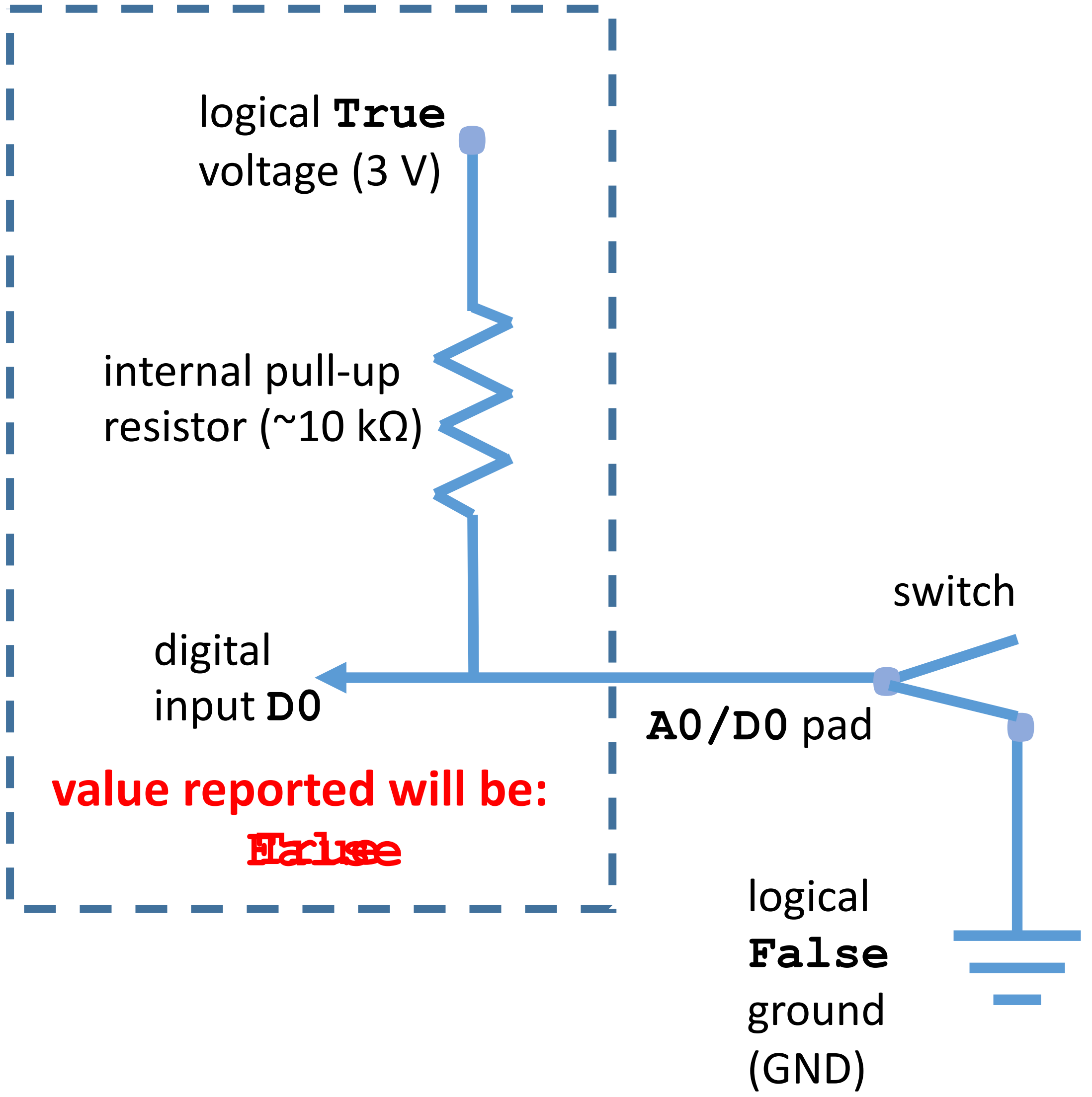
Controlling read and write state

DISC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

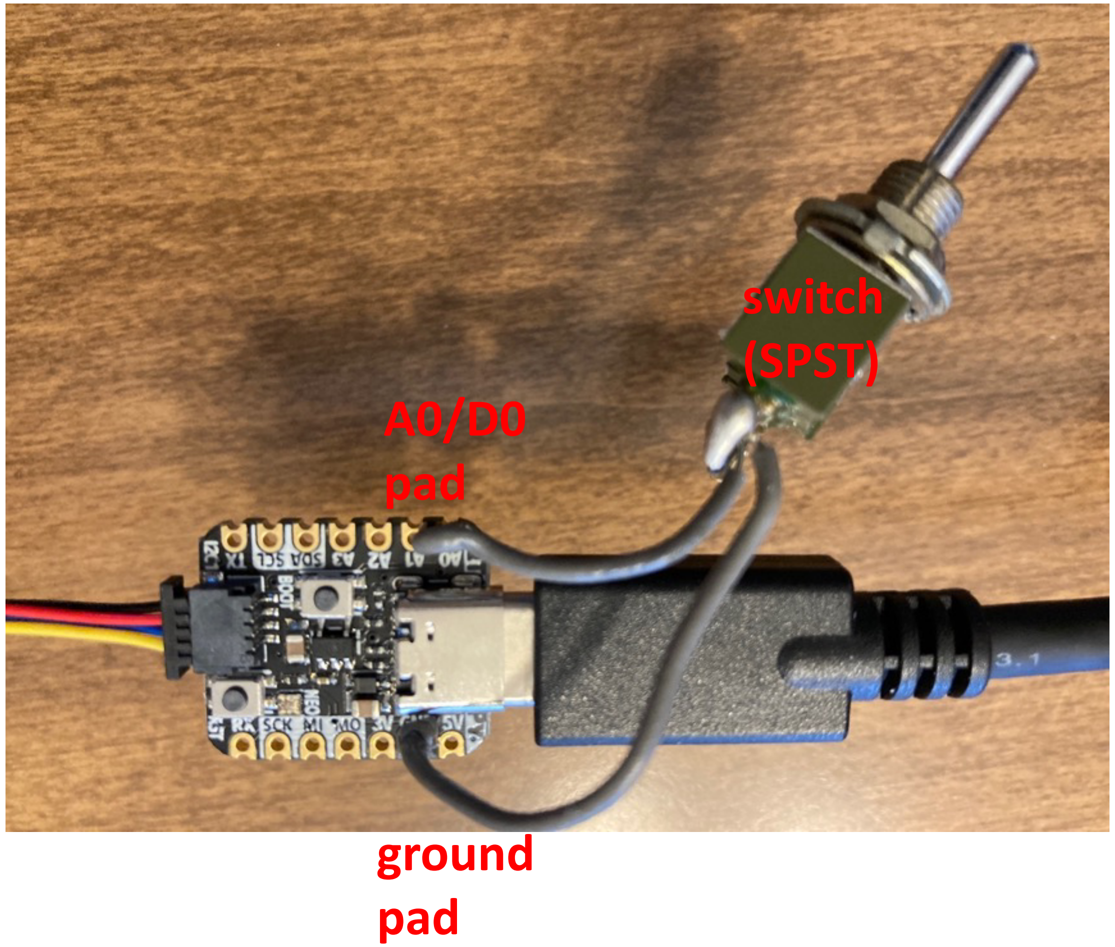
Jean & Alexander Heard LIBRARIES

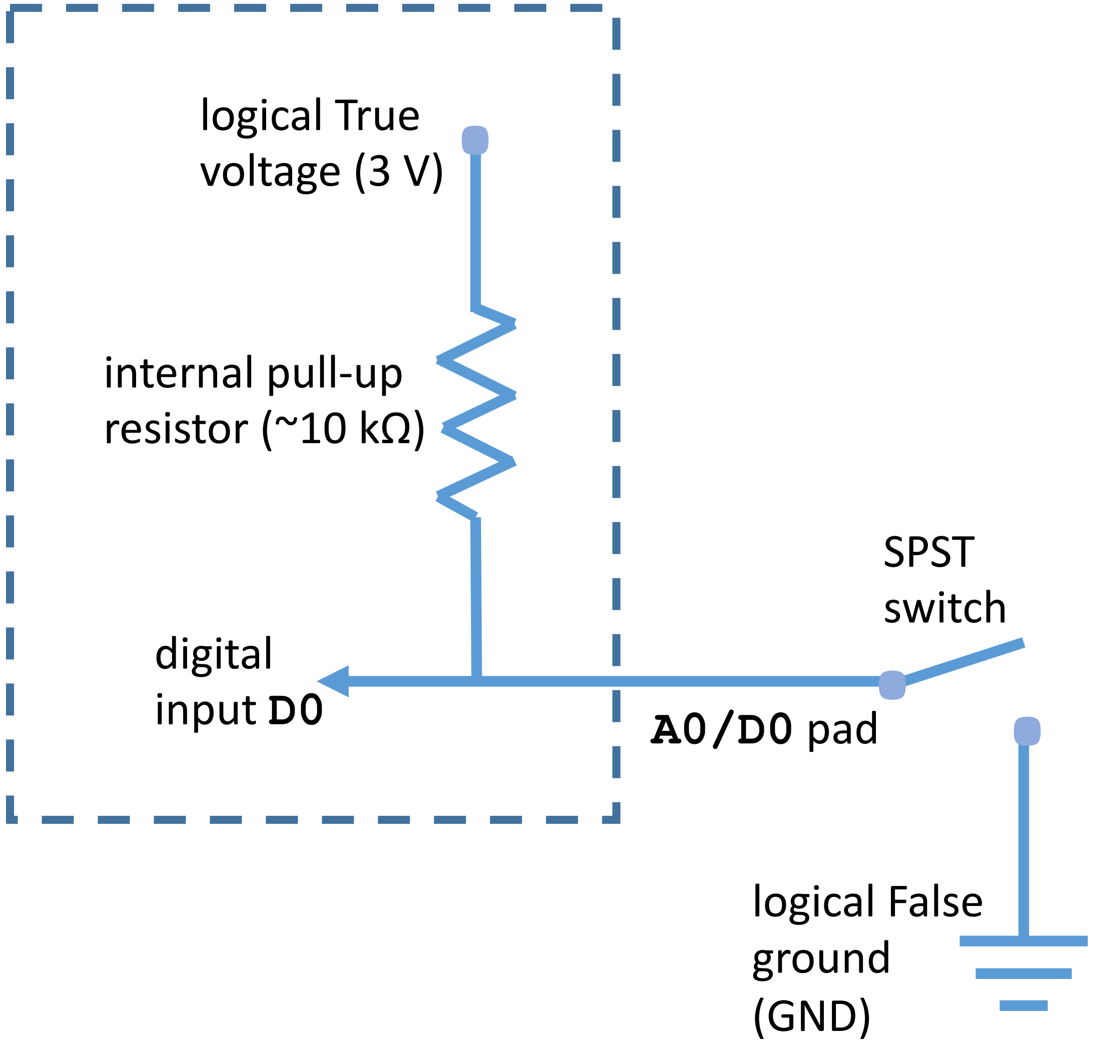
Controlling the input voltage of digital input



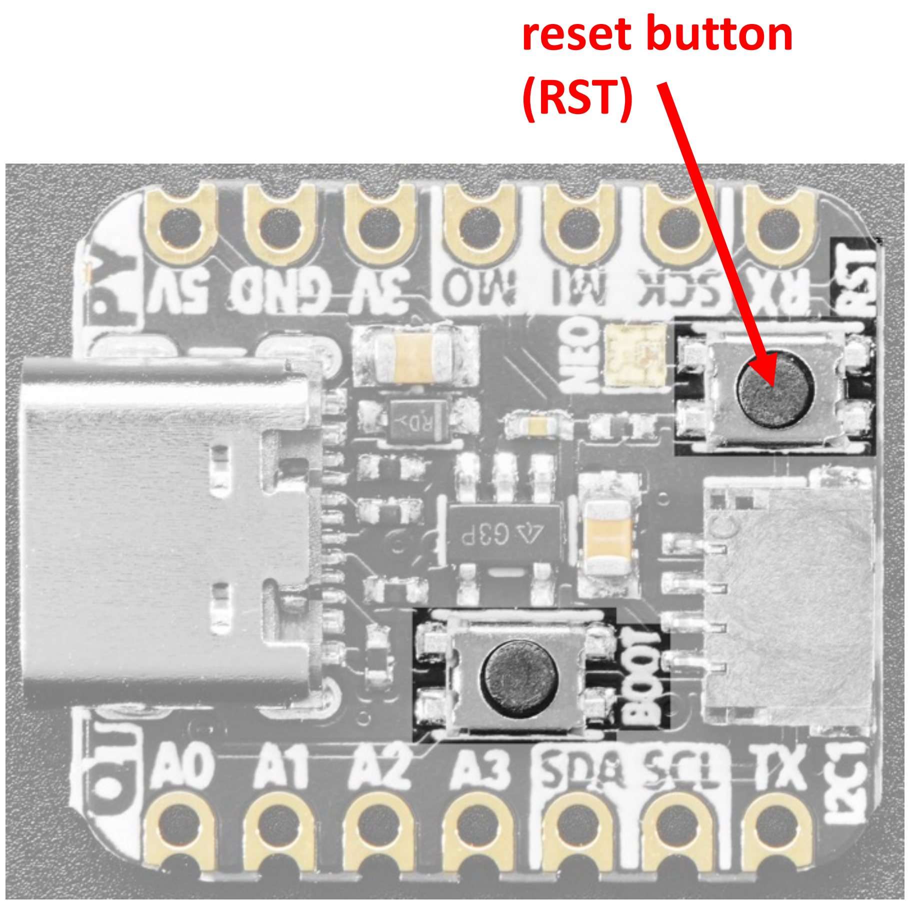


External switch wiring (soldering required)




logical True voltage (3 V)
internal pull-up resistor (~10 kΩ)
digital input D0
SPST switch
A0/D0 pad
logical False ground (GND)

What does the `boot.py` script do?



The **`boot.py`** script is a special file in CircuitPython

It is executed when:

- the board is powered up.
- when you do a "hard reset" by pushing the reset button on the board

It does not run of a "soft reset" from the serial console.

It runs before **`code.py`**

`boot.py` script to control read/write

Set the mode of A0/D0 ADC pin to digital by instatiating a `DigitalInOut` object using the D0 pin object

Set the direction of A0/D0 ADC pin to input

Set up internal resistor to pull-up configuration

```
import board
import digitalio
import storage

switch = digitalio.DigitalInOut(board.D0)
switch.direction = digitalio.Direction.INPUT
switch.pull = digitalio.Pull.UP

# Connecting D0 to ground makes switch.value False
storage.remount("/", switch.value)
```

If second argument of `.remount()` is `True`, memory is read-only
If second argument of `.remount()` is `False`, CircuitPython can write