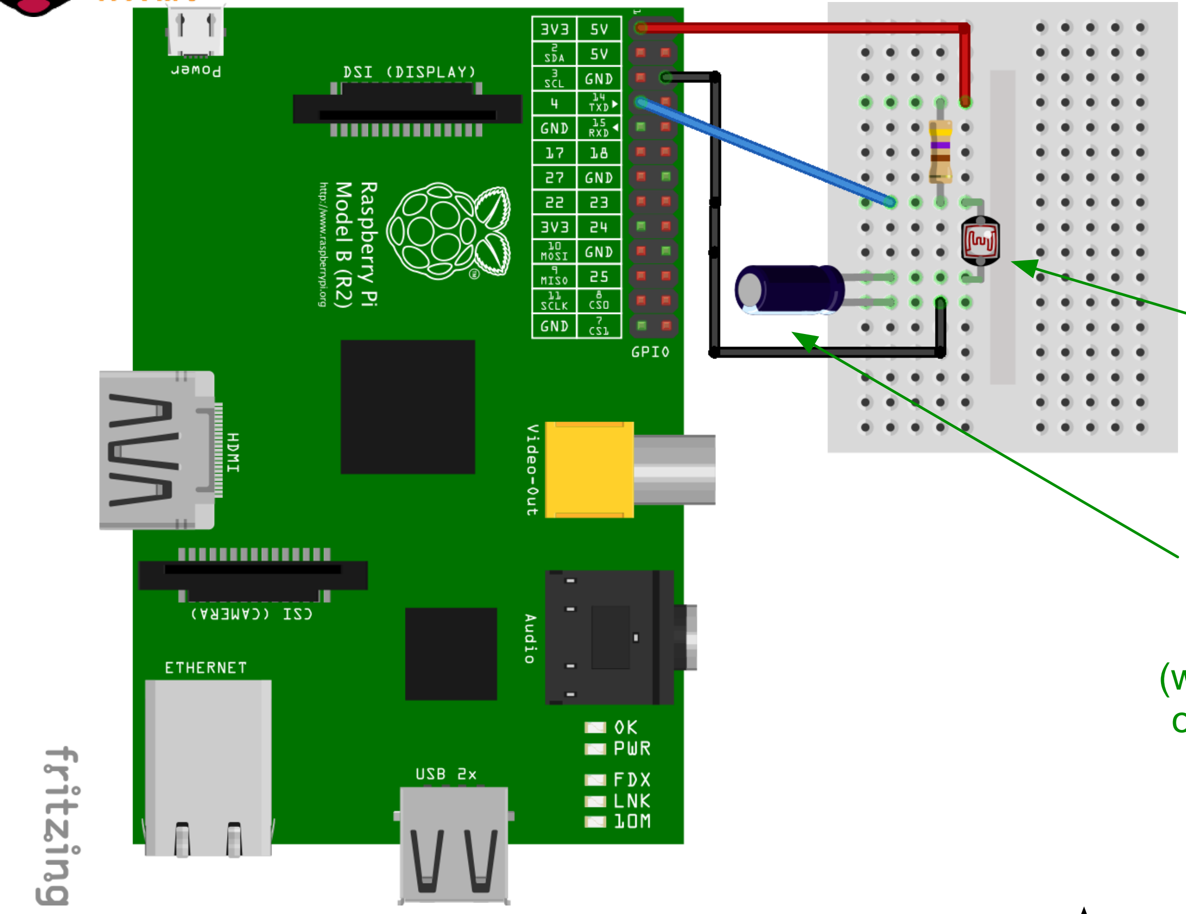
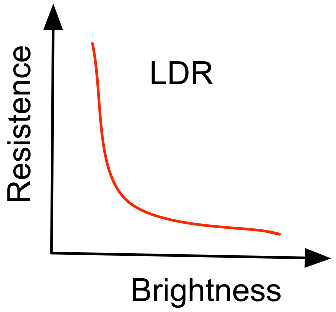
TOO BRIGHT WITH python



Light Dependent resistor (LDR)

0.47 uF capacitor

(white stripe on one side)

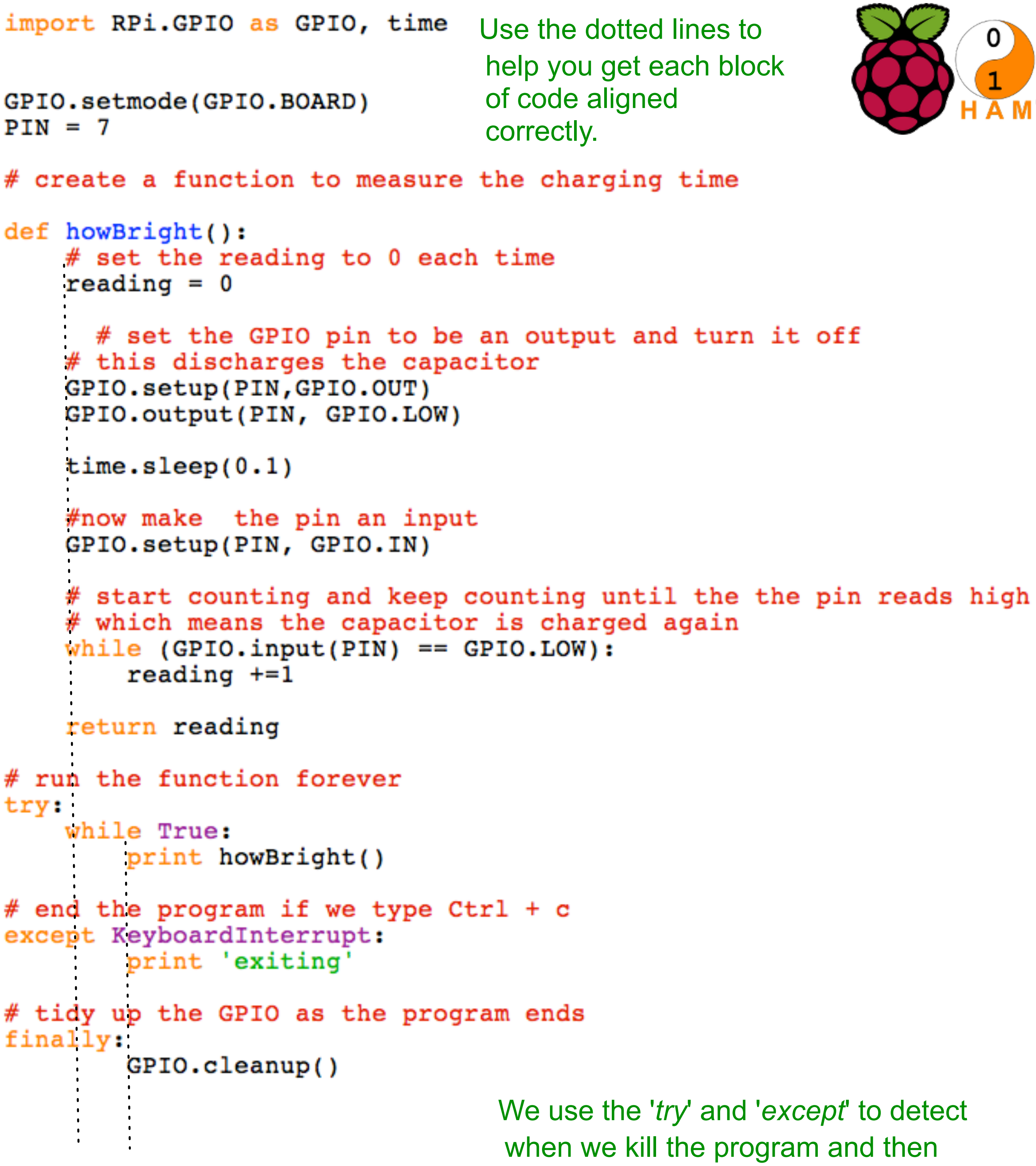


The LDR is an *analogue* sensor so it can have lots of different values. The Pi only has *digital* inputs (1 or 0) but we can use a clever trick - and a capacitor - to measure brightness.

A capacitor stores energy but takes time to charge up when connected to a source (in this case, the Pi).

The time taken for the capacitor charge depends on the resistance in the circuit and this depends on the amount of light shining on the LDR.

So by measuring how long the capacitor takes to charge we can get a rough idea of how light or dark it is!



```
import RPi.GPIO as GPIO, time


GPIO.setmode(GPIO.BOARD)
PIN = 7

# create a function to measure the charging time

def howBright():
    # set the reading to 0 each time
    reading = 0

      # set the GPIO pin to be an output and turn it off
    # this discharges the capacitor
    GPIO.setup(PIN,GPIO.OUT)
    GPIO.output(PIN, GPIO.LOW)

    time.sleep(0.1)

    #now make  the pin an input
    GPIO.setup(PIN, GPIO.IN)

    # start counting and keep counting until the the pin reads high
    # which means the capacitor is charged again
    while (GPIO.input(PIN) == GPIO.LOW):
        reading +=1

    return reading

# run the function forever
try:
    while True:
        print howBright()

# end the program if we type Ctrl + c
except KeyboardInterrupt:
        print 'exiting'

# tidy up the GPIO as the program ends
finally:
        GPIO.cleanup()
```

Use the dotted lines to help you get each block of code aligned correctly.

We use the '*try*' and '*except*' to detect when we kill the program and then call '*finally*' to close things down smoothly.

Can you add an LED to your circuit and have it turn on whenever it gets too dark?