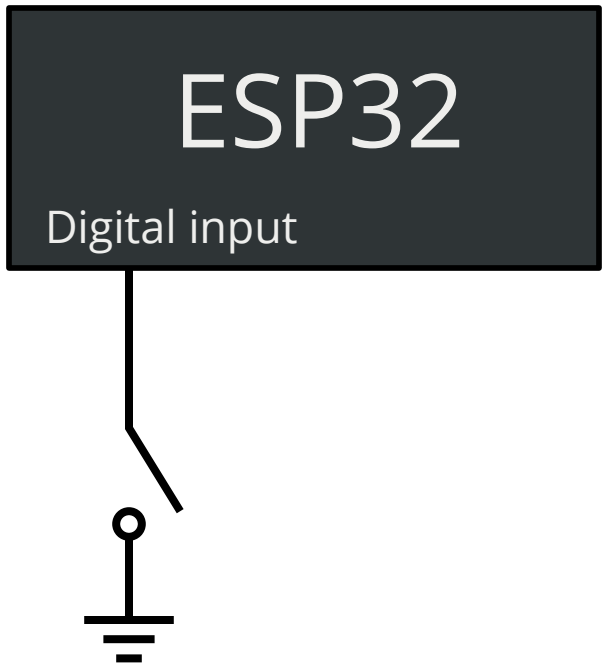
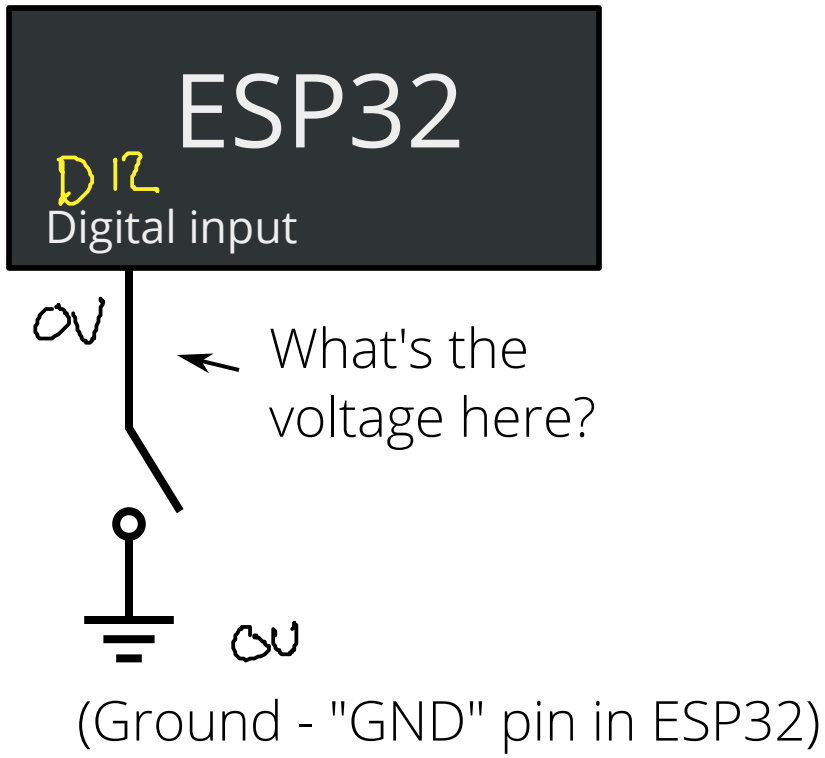
Build this circuit on your breadboard:

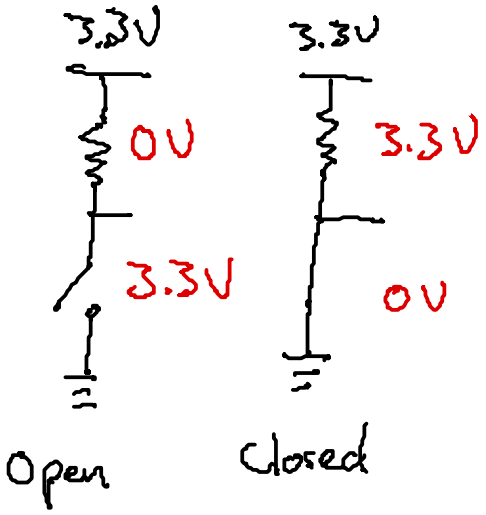


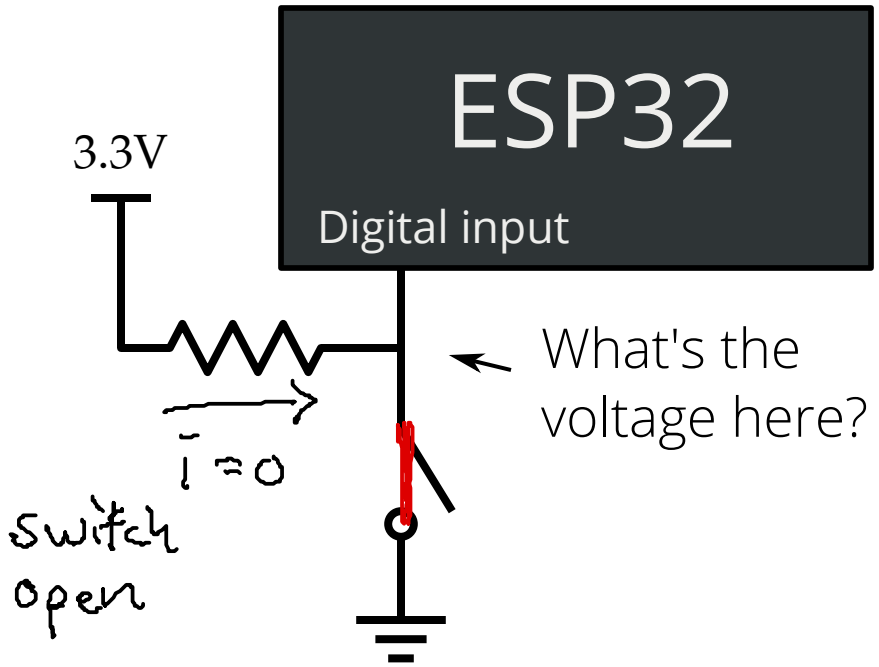
(Ground - "GND" pin in ESP32)

Build this circuit on your breadboard:



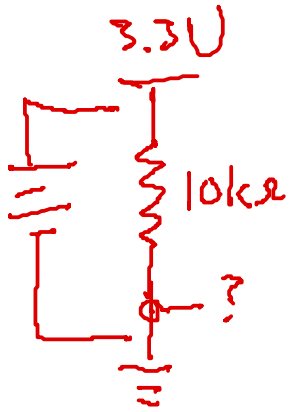
Build this circuit on your breadboard:





$V = iR$

$\Rightarrow V = 0$



(Ground - "GND" pin in ESP32)

EN 1: Engineering in the Kitchen

Steven Bell

30 September 2021

Tufts UNIVERSITY

Reading a switch

```
from machine import Pin # import necessary library

switch = Pin(12, Pin.IN, Pin.PULL_UP)

print(switch.value())
```

Use pull-up resistor

Constants in Python

We often have "special numbers" in our code

It helps to give them descriptive names instead of reusing the number everywhere.

```
PRESSED = 1

DELAY_TIME = 3000 # milliseconds

HELLO_MESSAGE = "Hello, microPython!"
```

`if` statements

```
if CONDITION:
  # Stuff to do if CONDITION is true
  # This is specified using indentation
else:
  # Stuff to do if CONDITION is false

# Stuff that is outside of the if-statement (un-indented)
```

CONDITION can be lots of things:

```
a == 3 # Check if a is equal to 3
x > y  # Check if x is greater than y
button.value() == 1
```

`if/elif` statements

"elif" is a contraction of "else if"

```
if FIRST_CONDITION:
  # Stuff to do if FIRST_CONDITION is true
elif SECOND_CONDITION:
  # Stuff to do if SECOND_CONDITION is true
else:
  # Stuff to do if neither condition is true
```

Write a program that turns on an LED when a switch is pressed.

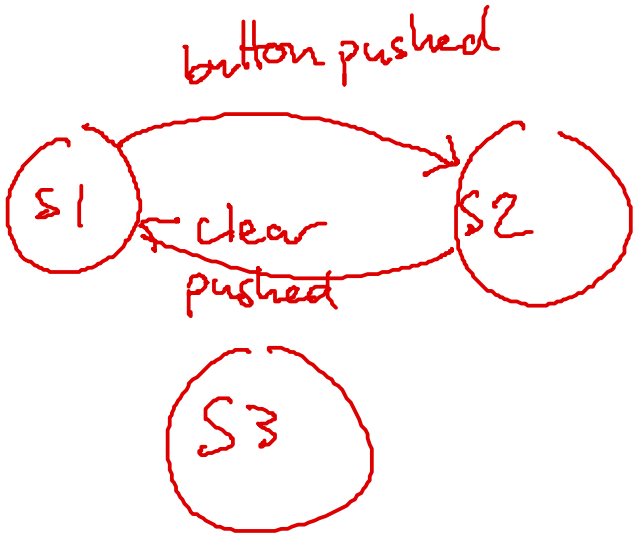
(The ESP32 should read the switch and control the LED, don't just put the switch in series with the LED)

State machines

Consist of a set of states and transitions between them

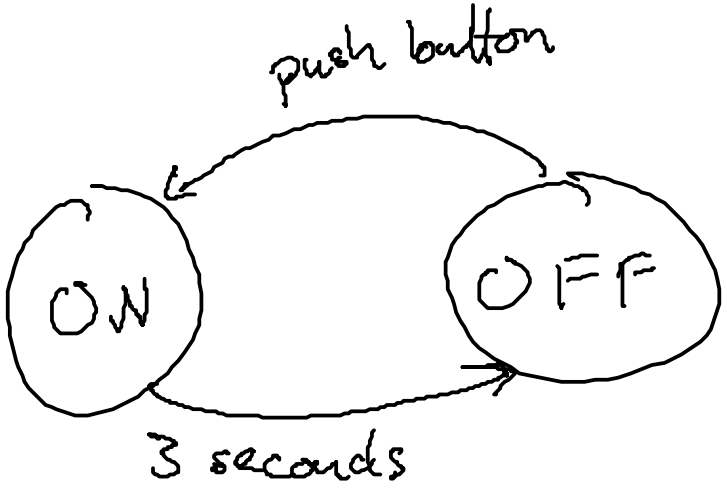
This is a formal way to think about complicated behavior

And it translates very nicely to code

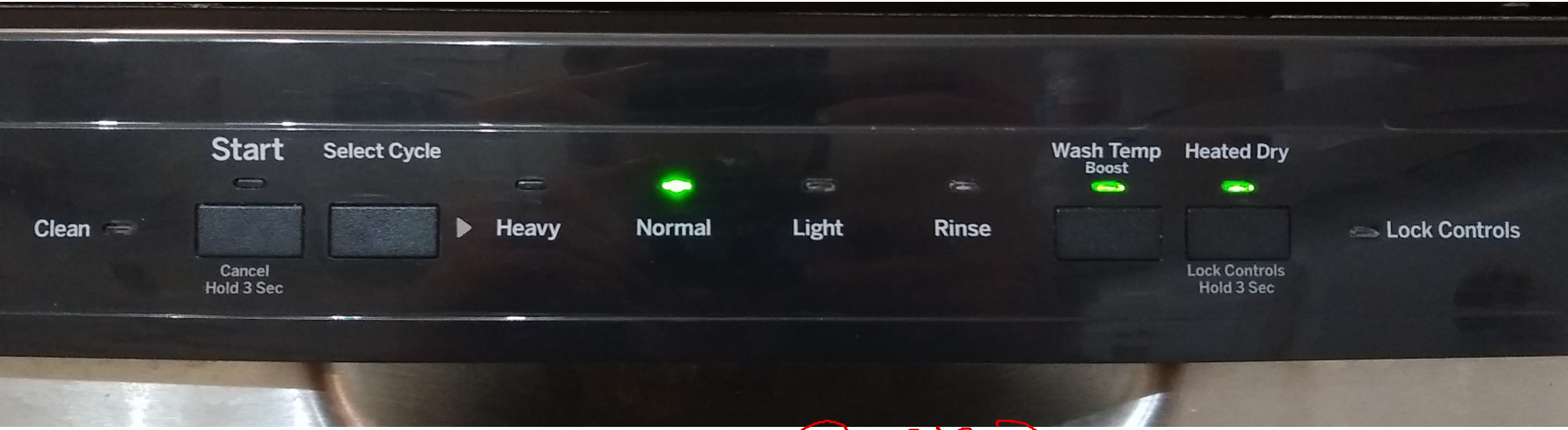


State machine example

Let's make something that turns on an LED for 3 seconds every time you press a button.



My dishwasher control panel



Draw one (or more) state machines to represent its behavior.

Project 2: building a microwave control panel

You will design and build your own microwave control panel.
It must:

1) Allow a user to specify different amounts of time to cook
2) Indicate the time remaining in some way
3) Turn off the microwave if the door is opened

(https://twitter.com/i/status/1418413938115371011)

For Tuesday

Email me if you want to keep your project partner for project 2

Complete "proof-of-concept" by Tuesday (see website)

Make revisions to your disassembly guides