

Final Projects

Project proposal ideas and hints (<http://cs107e.github.io/project/>)

Form project teams NOW - 1-3 people (2 best)

Google form due 11:59 pm Thursday November 17th

- **Include teammates for shared git repo**
- **Includes short project proposal**

Two labs

- **1st - refine project ideas and solid milestones**
- **2nd - finish milestones and final push**

Class demos (3:30pm Wed Dec 14)

Final code and writeup submission (due Noon on Thursday, Dec. 15th)

\$20 parts budget per person; need to submit receipts

Assignments / Extensions

Starting this week: Maria will do regrades every night. All regrades are due the Tuesday after Thanksgiving Break.

Full system bonus is also due the Tuesday after Thanksgiving Break.

CR/NC Deadline

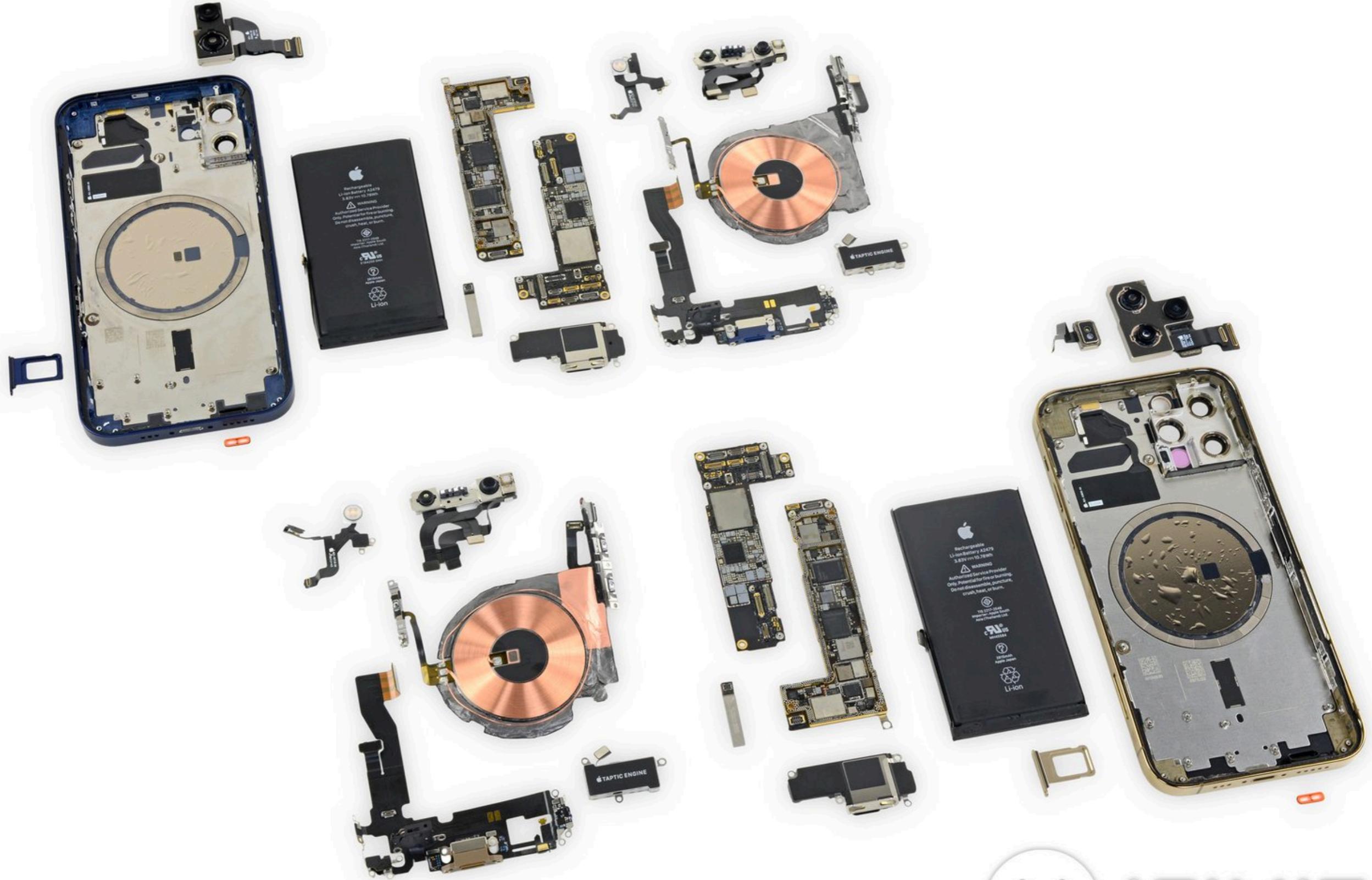
In order to pass the course:

- You must pass 80% of the P1 tests for the assignments.**
- You must pass the final project.**

CS107e

Sensors

Apple iPhone 12 Teardown



How many sensors?



Apple iPhone 12

How many sensors?

3 12MP Ultra wide, wide,, and telephoto cameras

LIDAR TrueDepth camera

12MP front TrueDepth camera with FaceID

HapticTouch multi-touch vibrating display

Microphones (2 at top, 2 at bottom)

Proximity sensor

Ambient light sensor

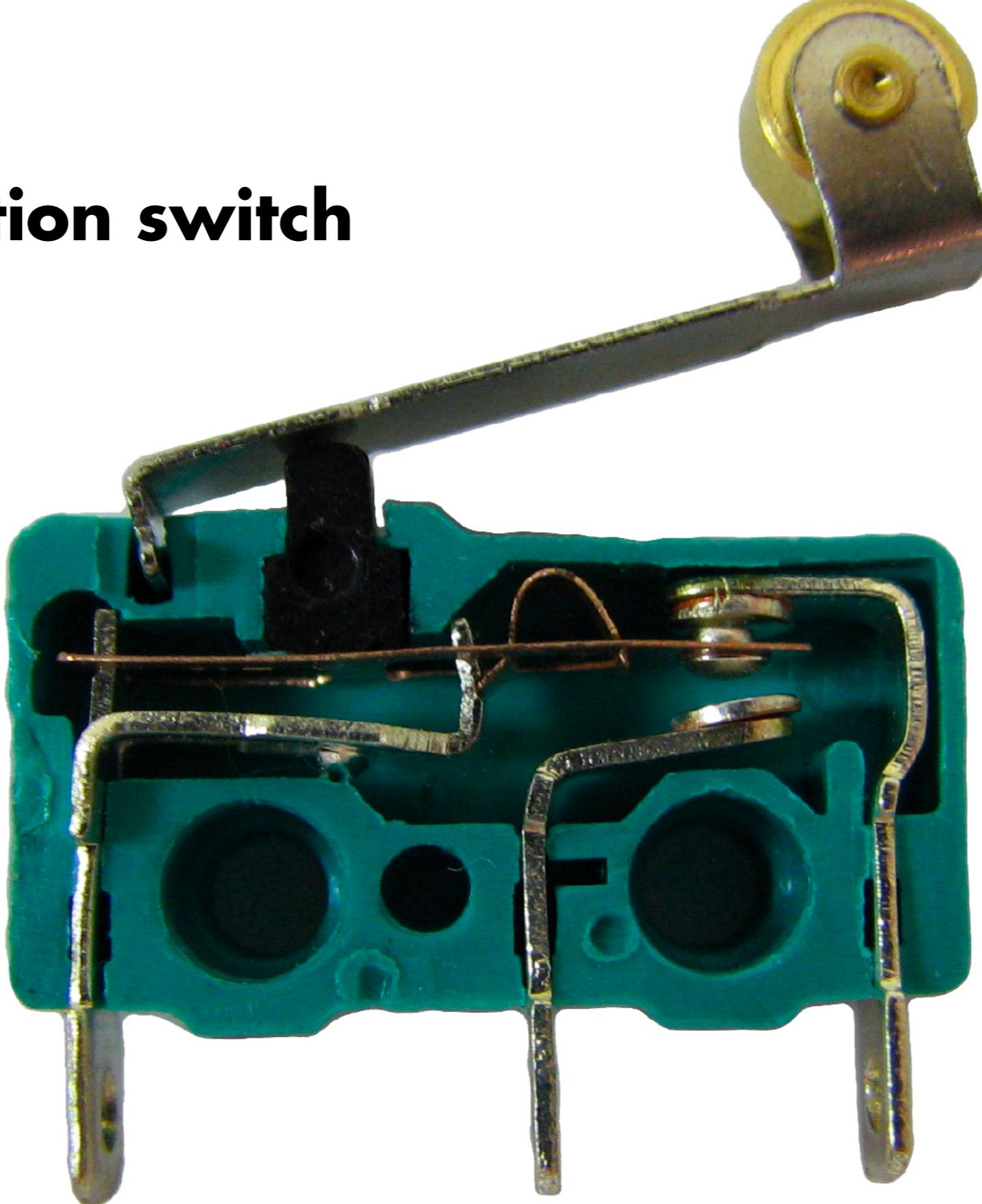
Accelerometer

Gyroscope

Compass (magnetometer)

Barometer (altimeter)

Snap-action switch



Common

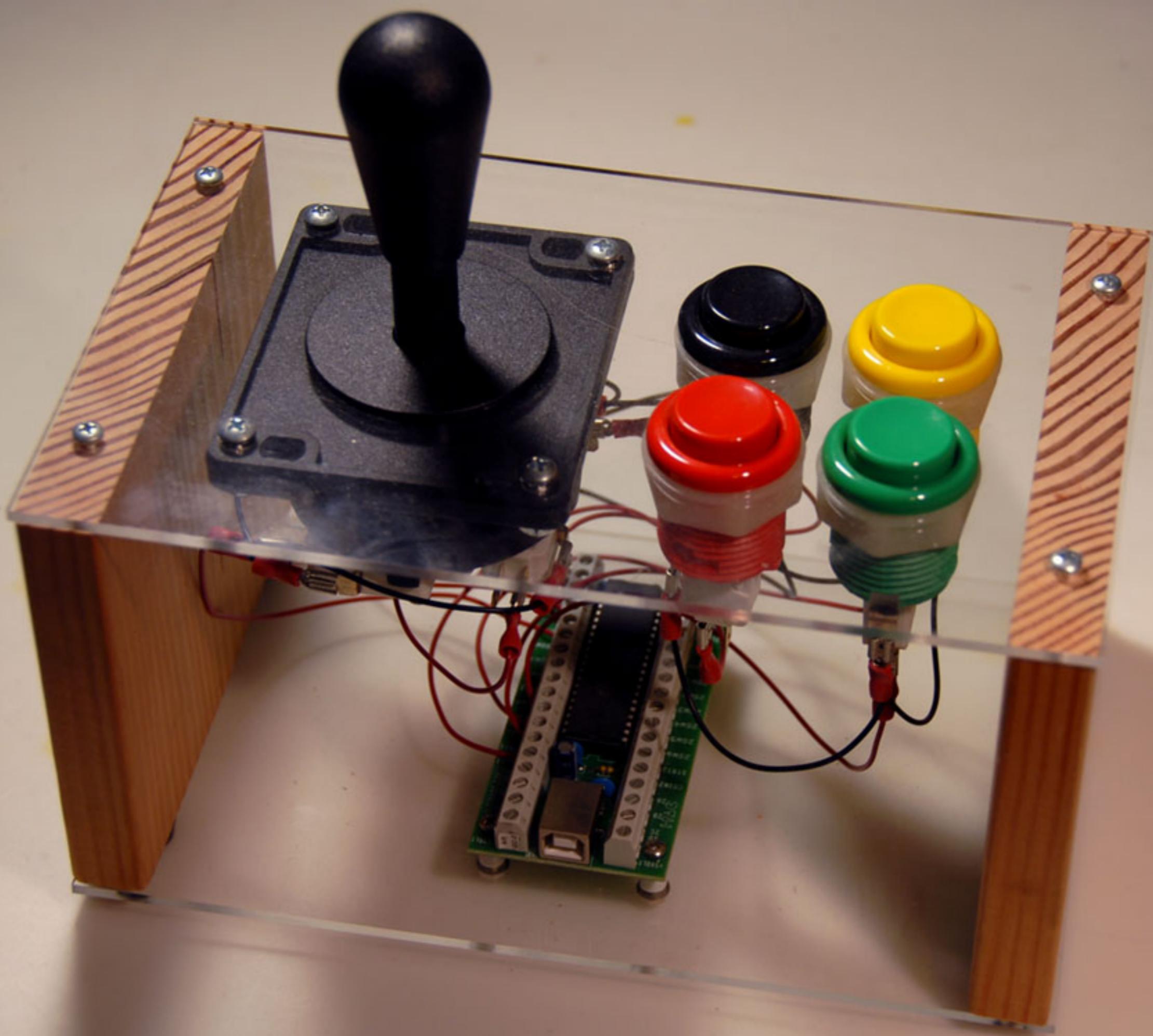
NO

NC

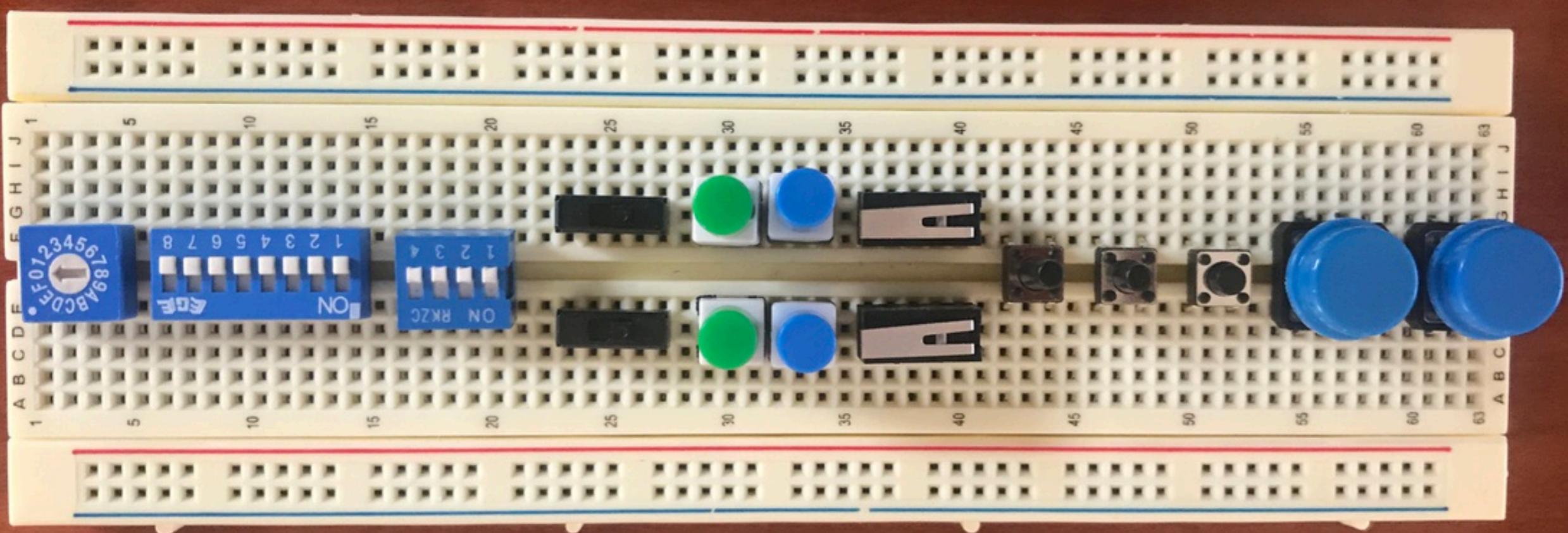
Happ Pushbutton



Happ Joystick



Buttons and Switches



Rotary Encoder

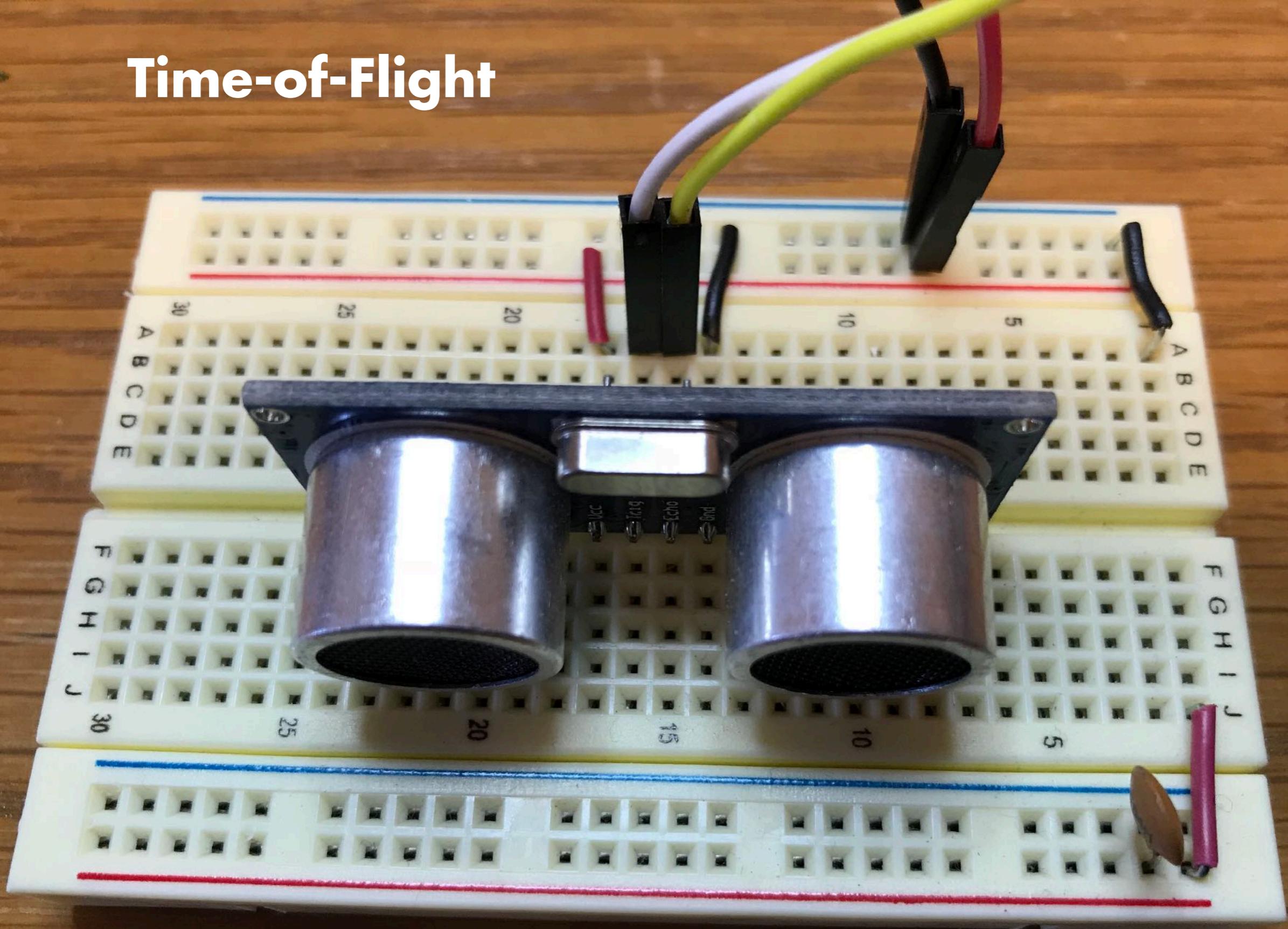


- A rotary encoder has a knob that turns, and "clicks" after small rotations.
- You can programmatically determine which direction the rotation occurred
- Pushing down on the knob clicks a button
- <https://lastminuteengineers.com/rotary-encoder-arduino-tutorial/>

Data Sheet and example code:

[https://www.handsontec.com/dataspecs/module/
Rotary%20Encoder.pdf](https://www.handsontec.com/dataspecs/module/Rotary%20Encoder.pdf)

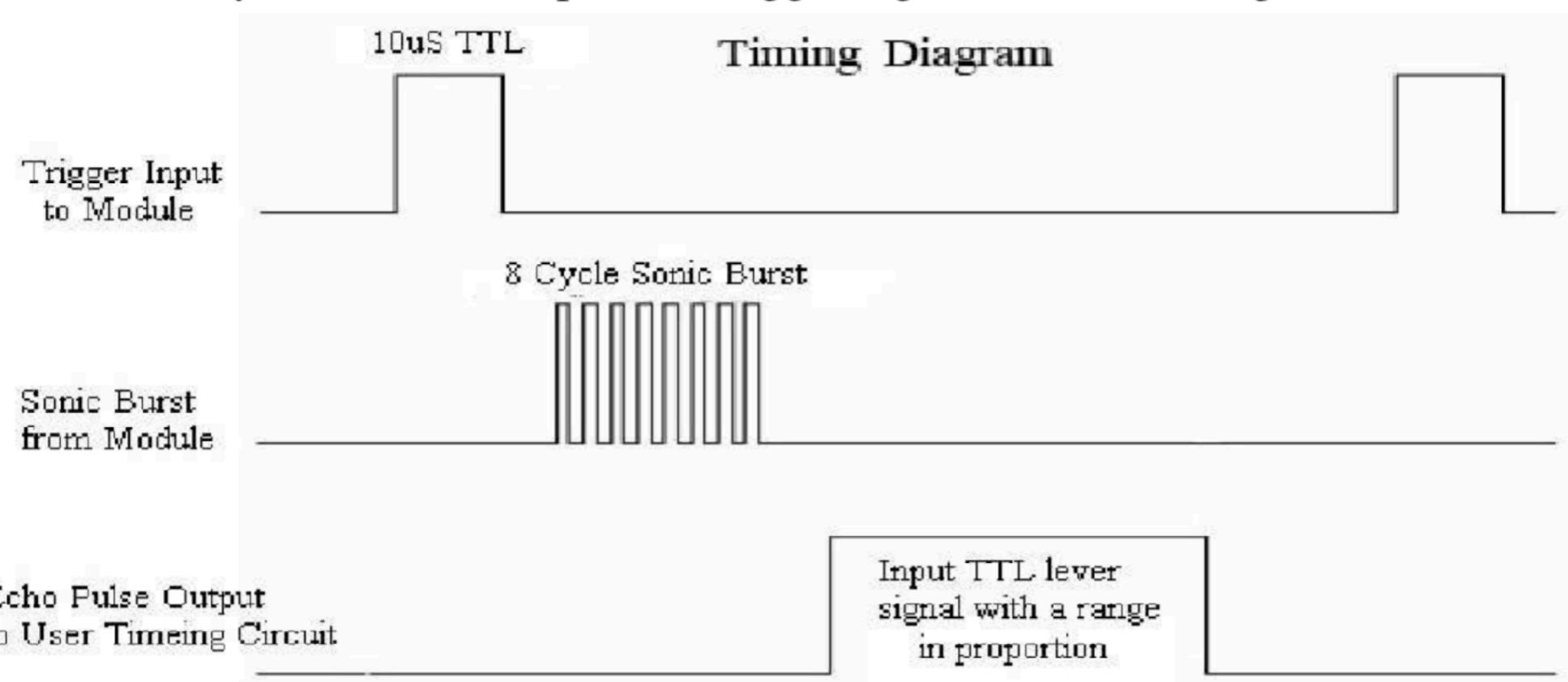
Time-of-Flight



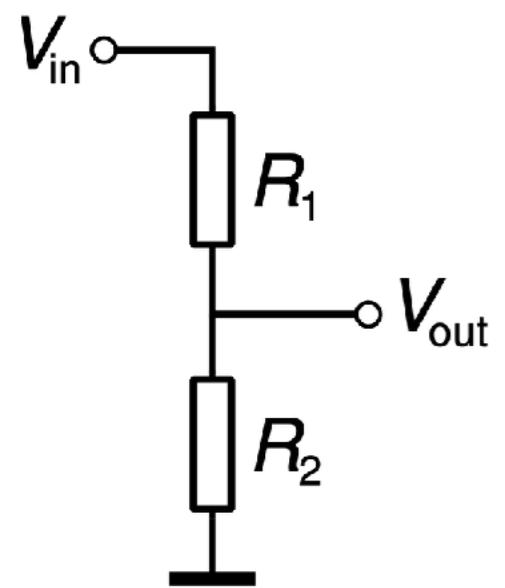
HC04 Ultrasonic Sonar



The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $uS / 58 = \text{centimeters}$ or $uS / 148 = \text{inch}$; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Analog to Digital (ADC)



$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in}$$

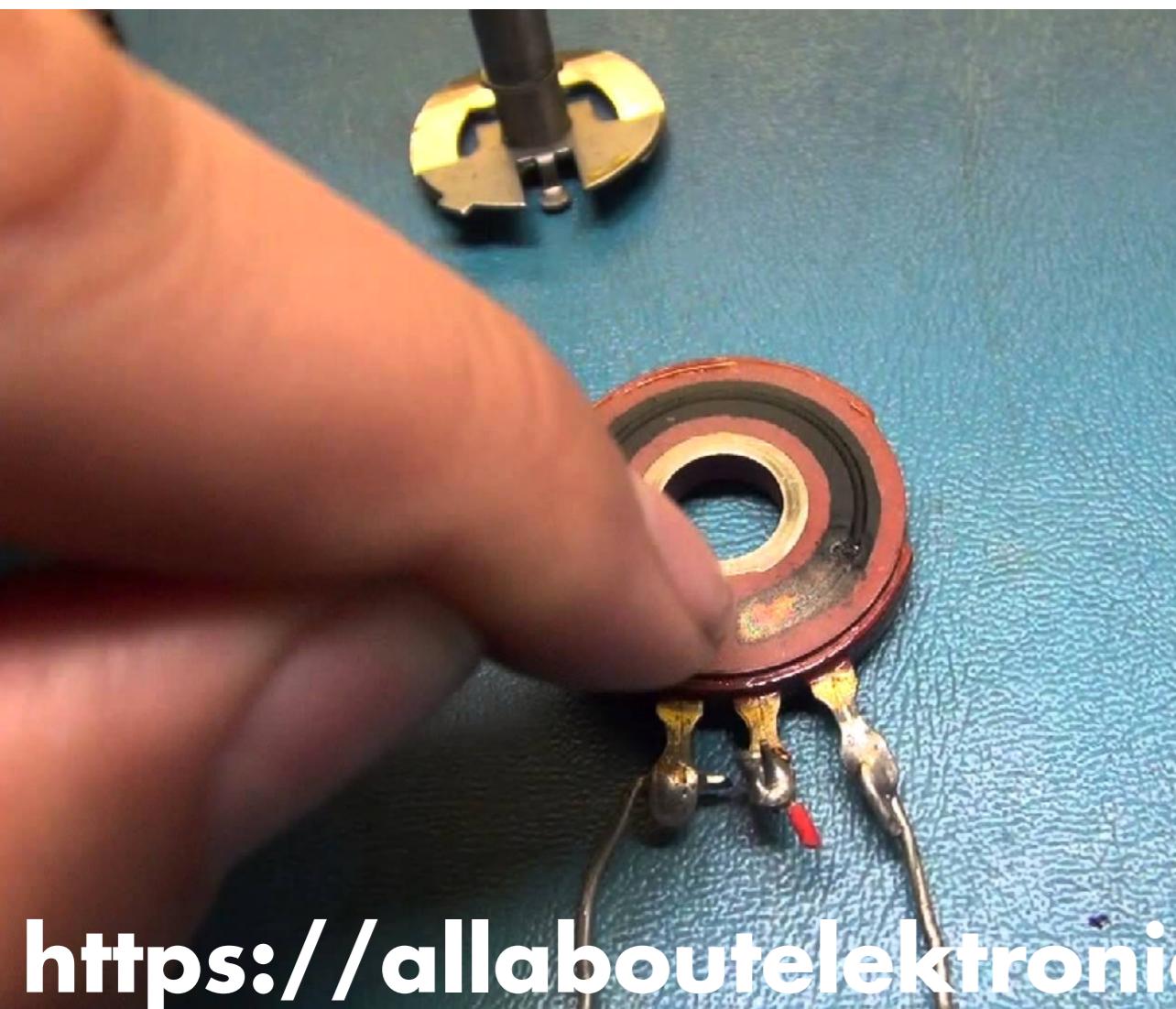
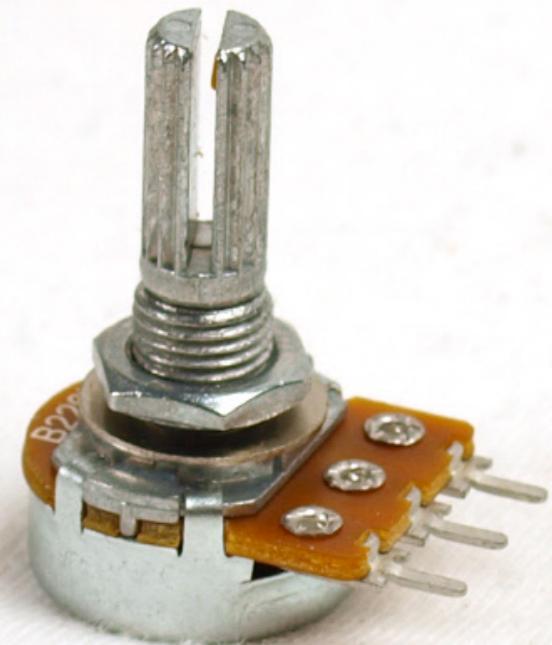




Image © Avon Fox
www.the-liberator.net
Images may be used under terms
with this watermark intact.

Atari 2600 Paddle

How would you measure the voltage?

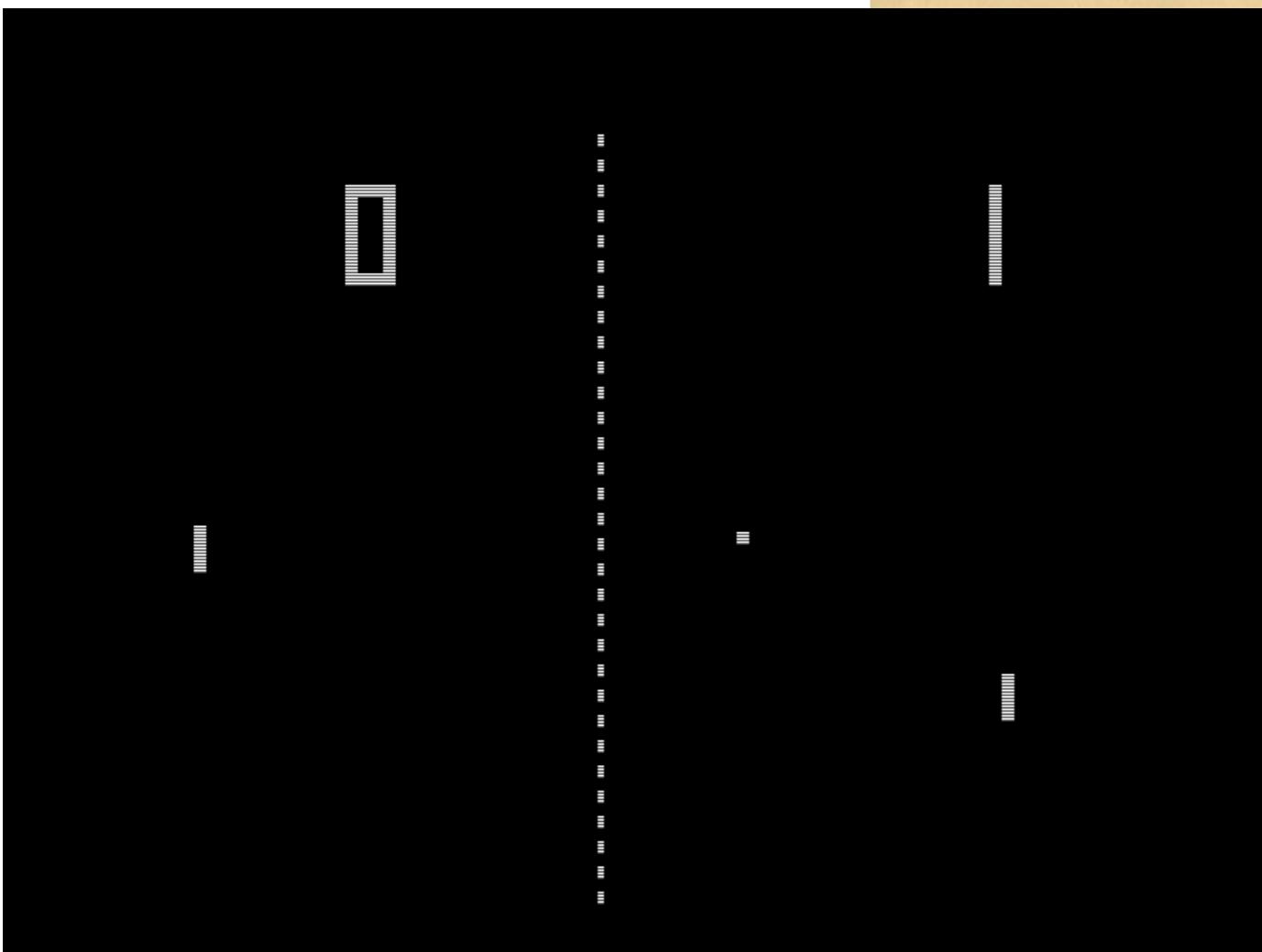


THE NEWEST 2 PLAYER
VIDEO SKILL GAME

PONG

from ATARI CORPORATION
SYZYGY ENGINEERED

The Team That Pioneered Video Technology



tract Mode
matically
of Ball
ng Paddle
te Controls
TE TV and
Long,

MPUTER

ROFITS
• Location
Suitable
d Locations

FROM YOUR LOCAL DISTRIBUTOR

Maximum Dimensions:

WIDTH -26"

HEIGHT - 50"

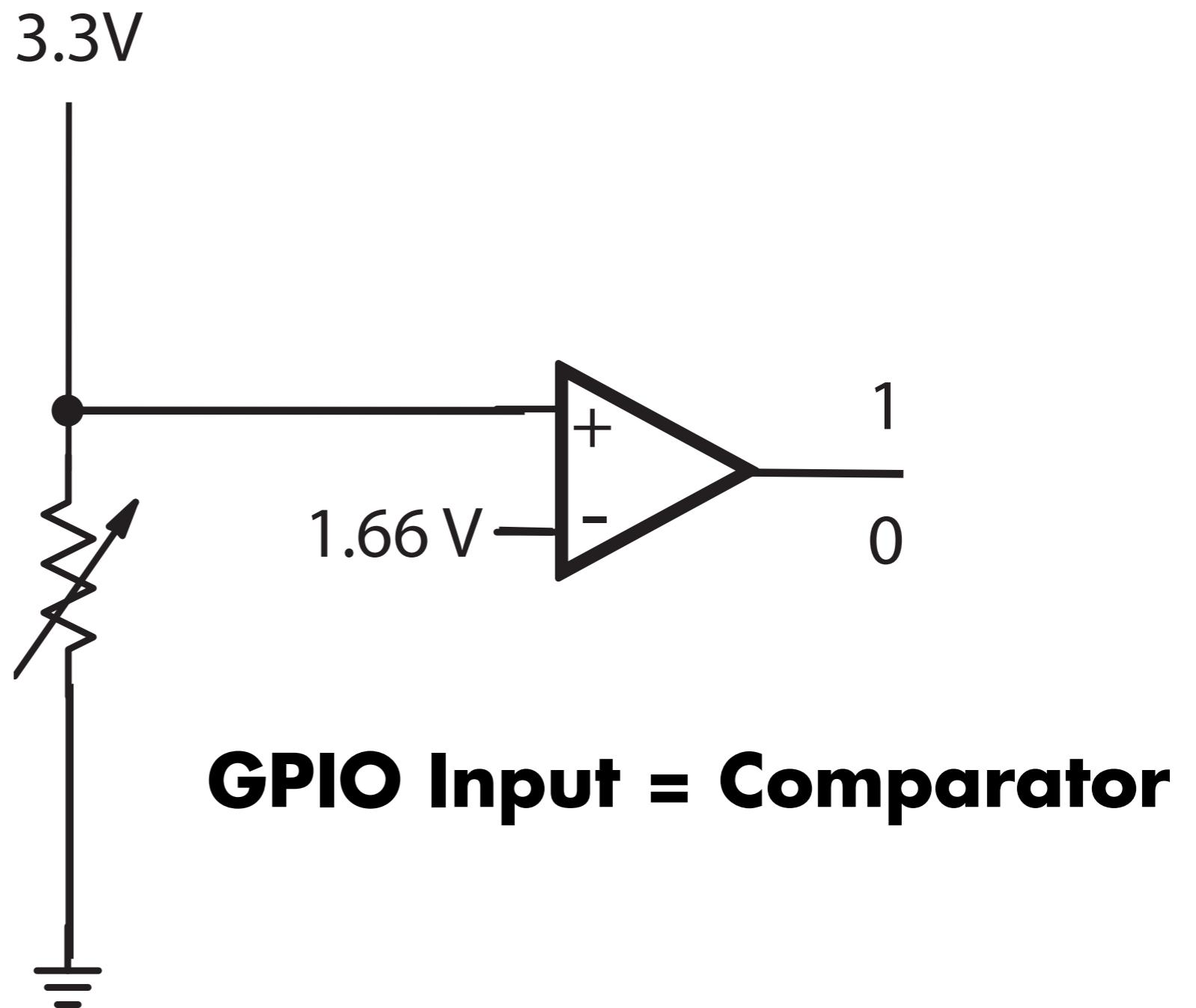
DEPTH - 24

SHIPPING WEIGHT:

150 Lb.

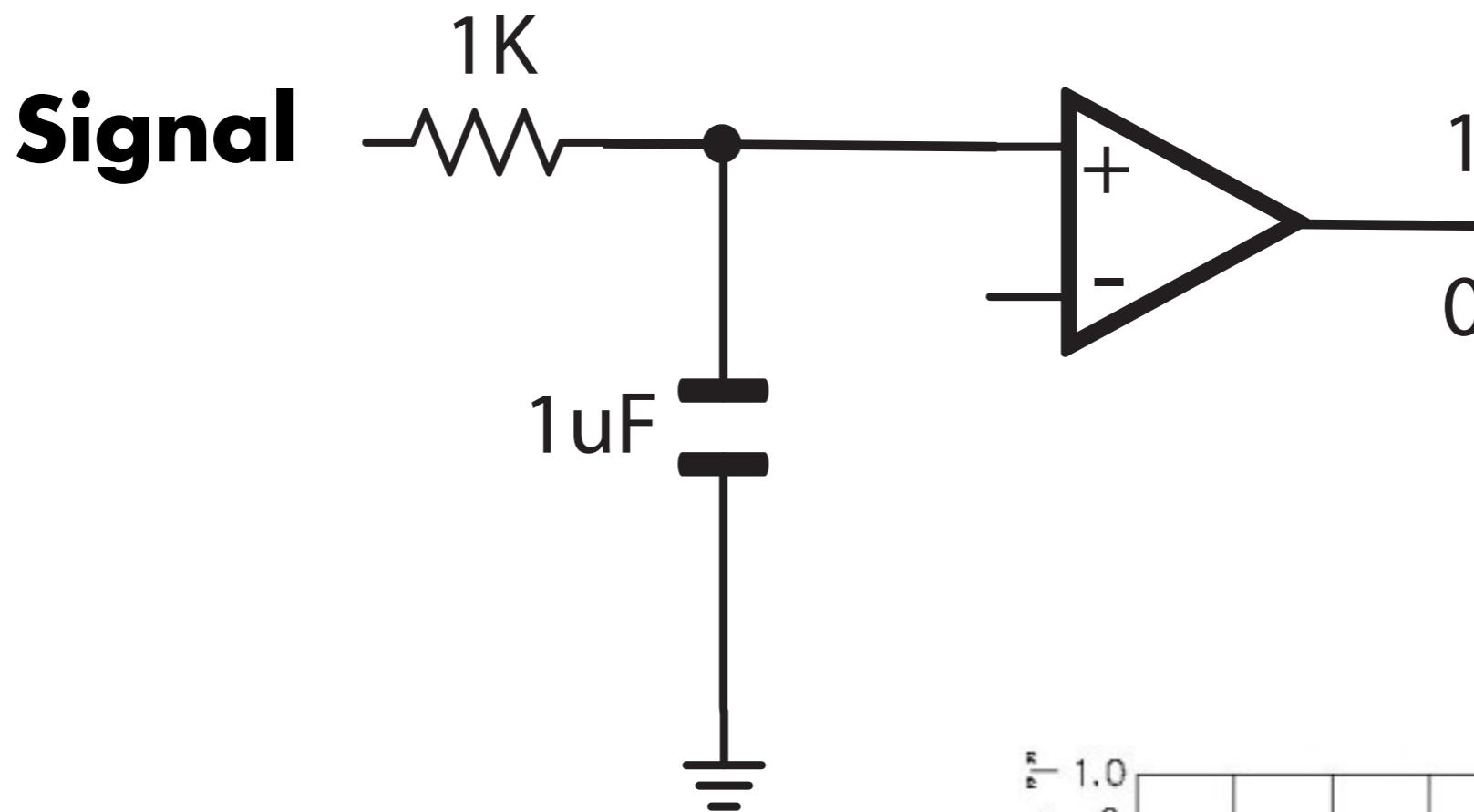


**Potentiometer
(Voltage divider)**

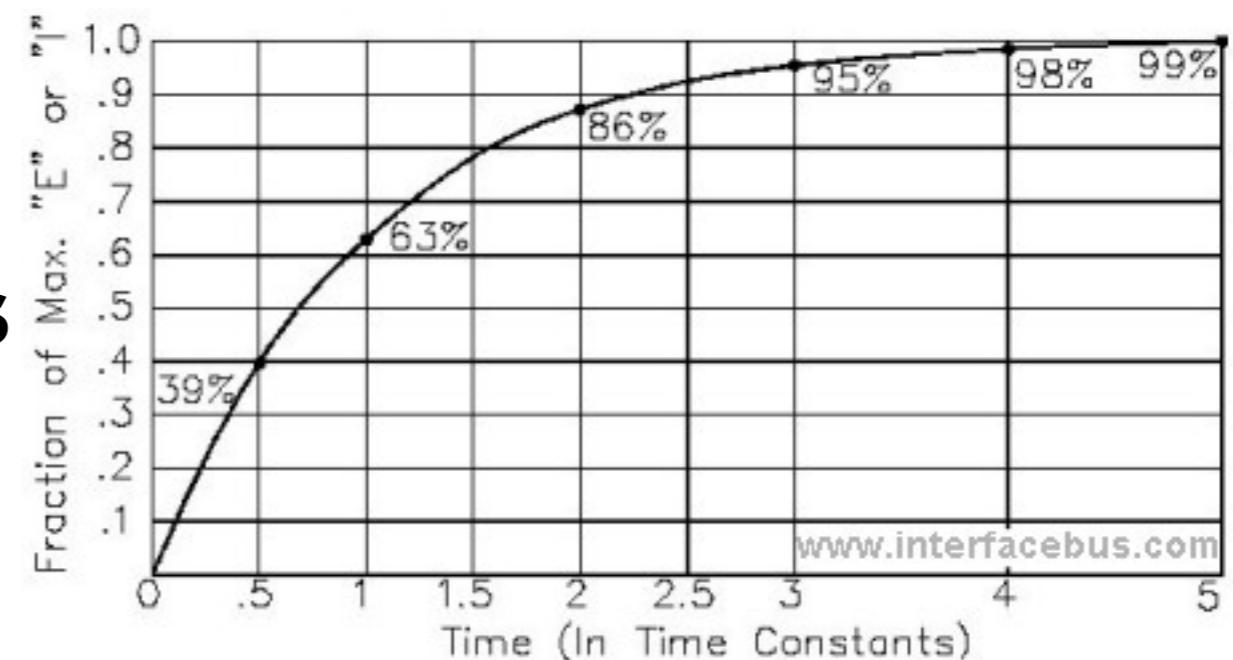


Charging Circuit

The time to fire depends on the input signal voltage

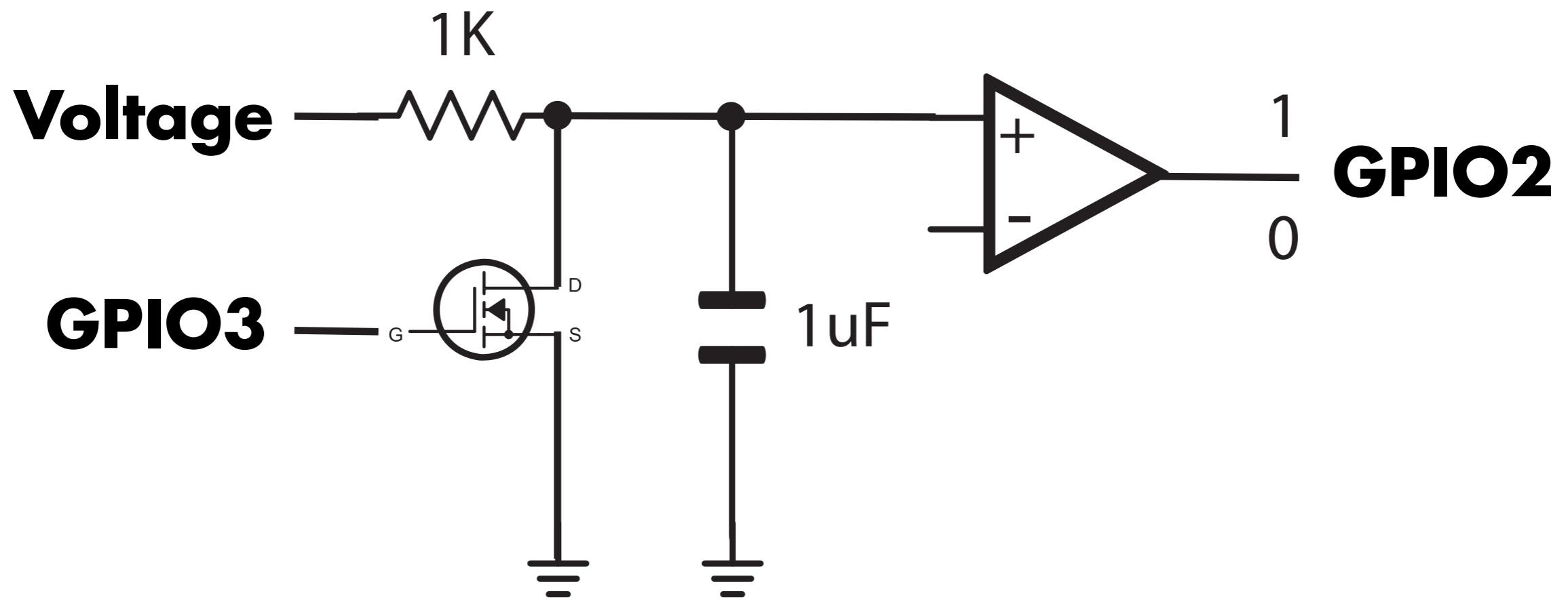


$$RC = 10^3 \times 10^{-6} = 1000 \mu\text{secs}$$

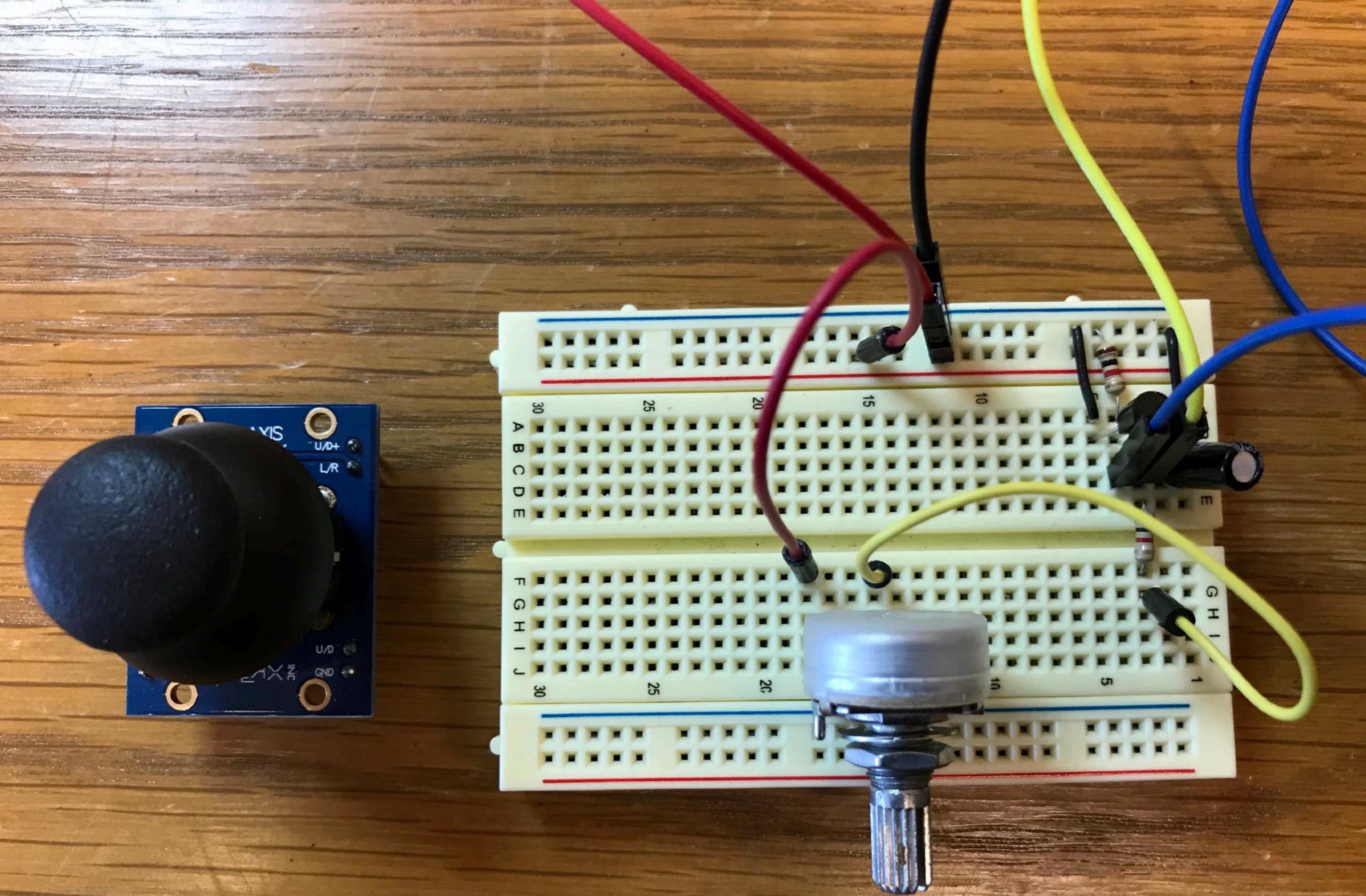


ADC

1. Turn on transistor, discharge capacitor
2. Turn off transistor, charge capacitor



RC = 1000 usecs



joystick



How do we figure out how to use it?

We find the part number (often written on the part), and we read the data sheet!

joystick



How do we figure out how to use it?

We find the part number (often written on the part), and we read the data sheet!

**Uh-oh. This part doesn't have a number.
So...we Google the pins.**

joystick



Specifications

| | Joystick Module |
|-------------------|-----------------------|
| Switch Protocol | Digital |
| X,Y Protocol | Analog |
| Operating Voltage | 3VDC - 5VDC |
| Dimensions | 47(L) x 25(W) x 32(H) |

Pinout

| Module | Duinotech | Function |
|--------|-----------|--------------------|
| GND | GND | Ground Connection |
| +5V | 5V | Power |
| VRX | A0 | X Output |
| VRY | A1 | Y Output |
| SW | D7 | Push Button Output |

joystick



Specifications

| | Joystick Module |
|-------------------|-----------------------|
| Switch Protocol | Digital |
| X,Y Protocol | Analog |
| Operating Voltage | 3VDC - 5VDC |
| Dimensions | 47(L) x 25(W) x 32(H) |

**Here we go again with
an analog circuit!**

Pinout

| Module | Duinotech | Function |
|--------|-----------|--------------------|
| GND | GND | Ground Connection |
| +5V | 5V | Power |
| VRX | A0 | X Output |
| VRY | A1 | Y Output |
| SW | D7 | Push Button Output |

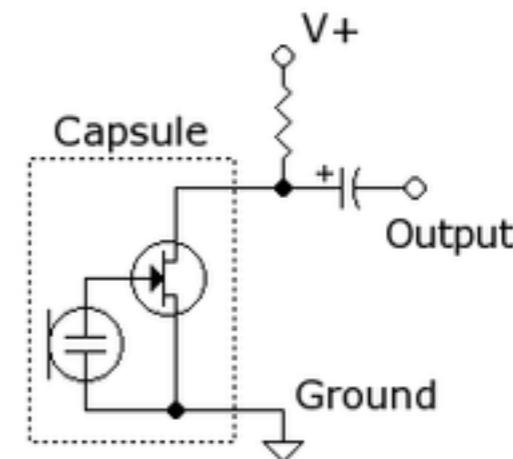
**The Raspberry Pi GPIO pins are all digital, but this joystick
is analog!**

We can use our code from the potentiometer demo!

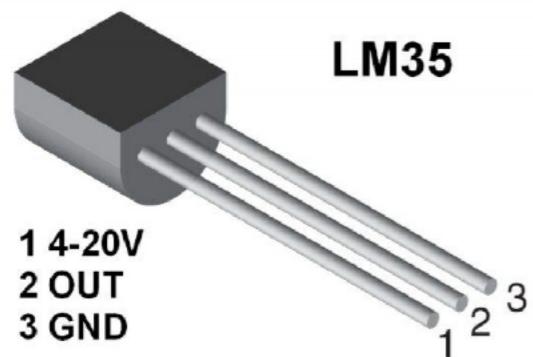
Analog Sensors



**Phototransistor
(light)**



**Electret Microphone
(pressure)**



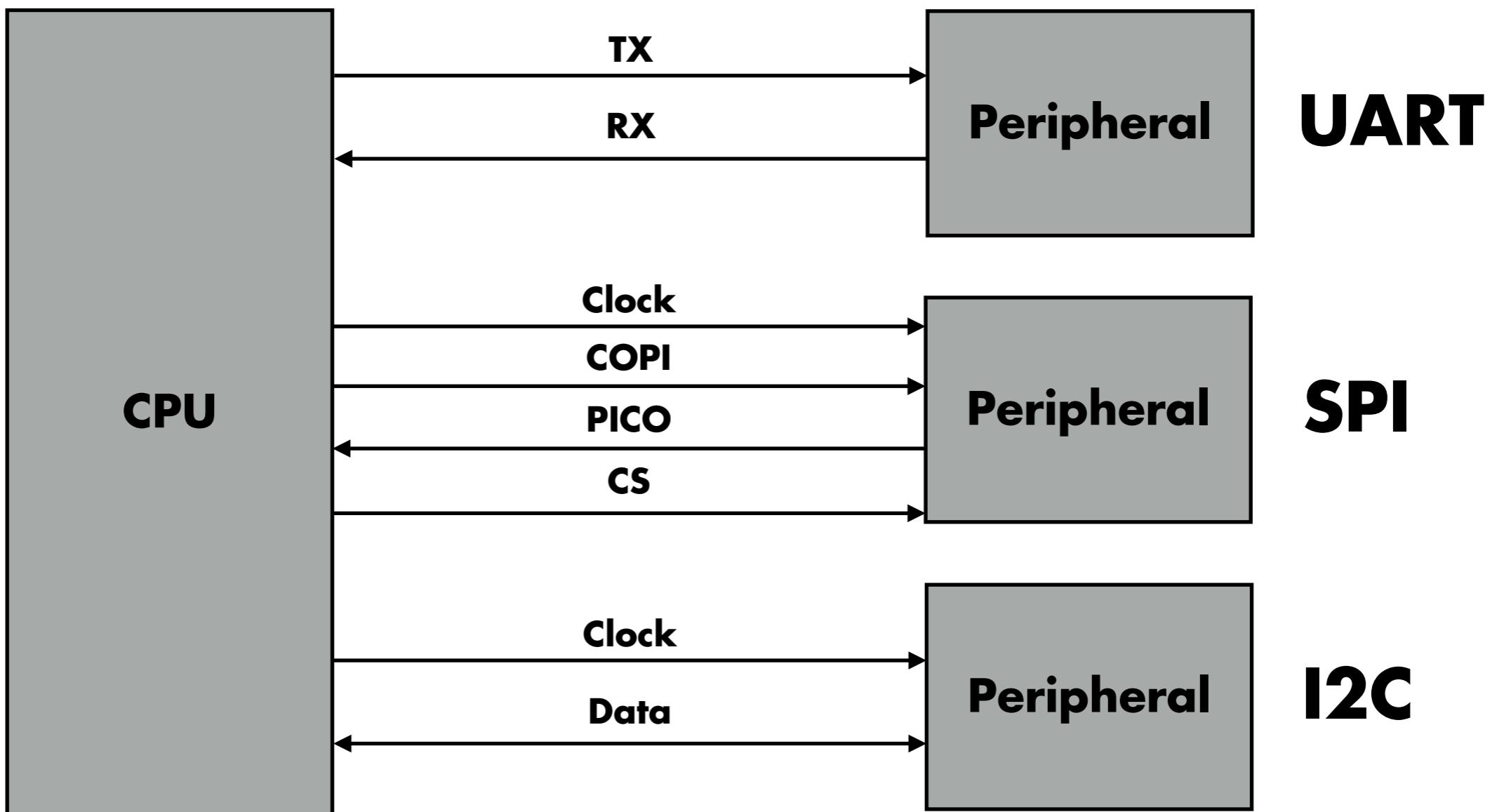
(temperature)



**Analog Hall Effect
(magnetic field)**

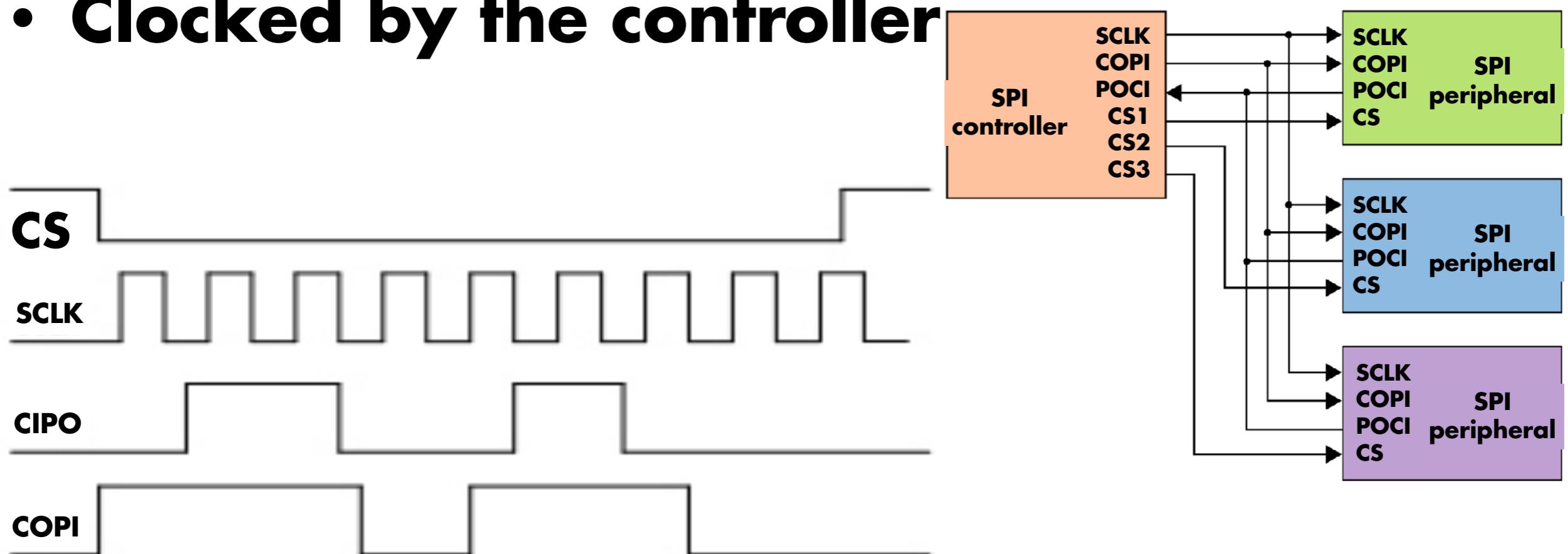
Smart Sensors

Bus Protocols

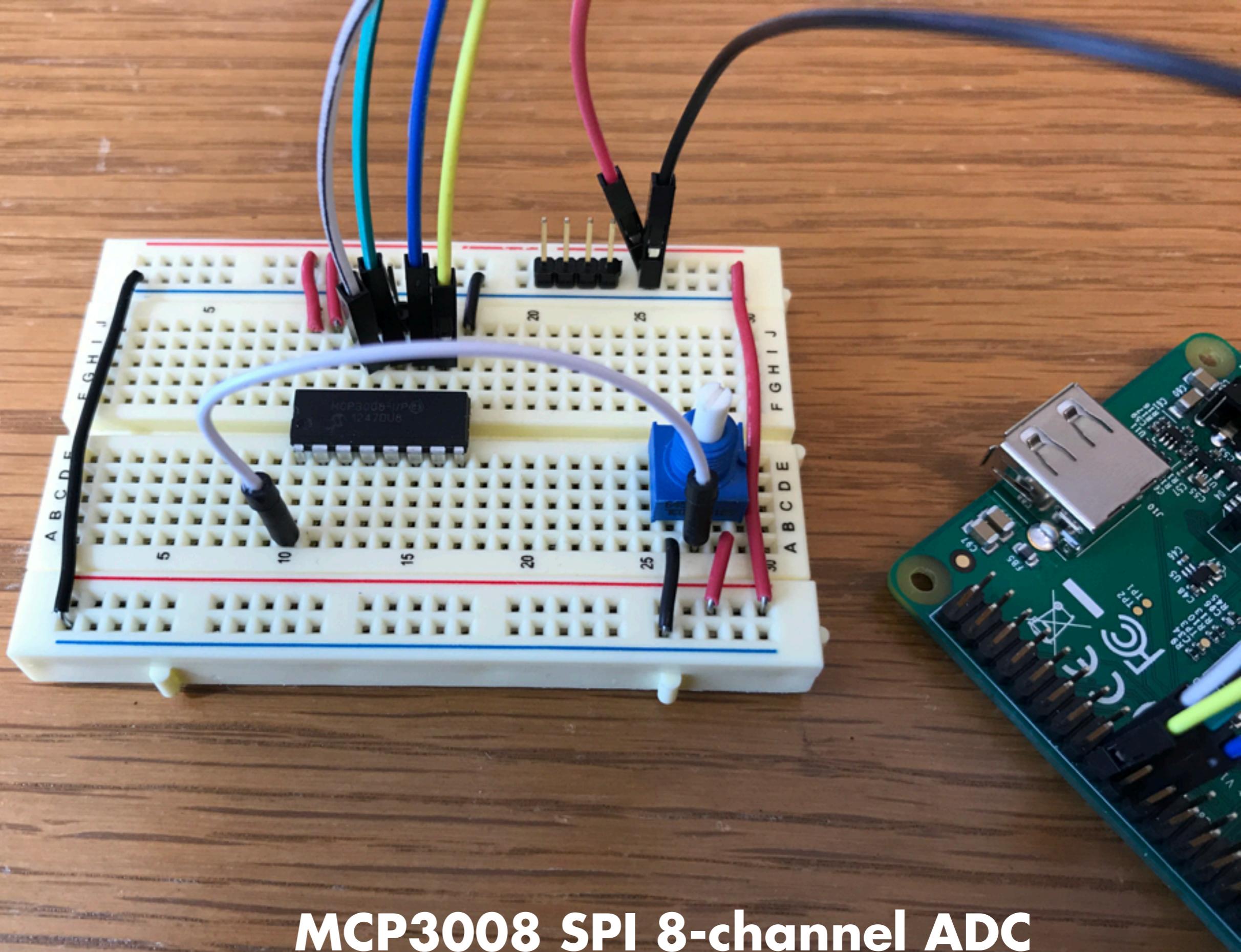


SPI

- Shared **CLK, COPI, POCl** lines
- Active low chip select (**CS**) lines to specify which peripheral is active
- Clocked by the controller

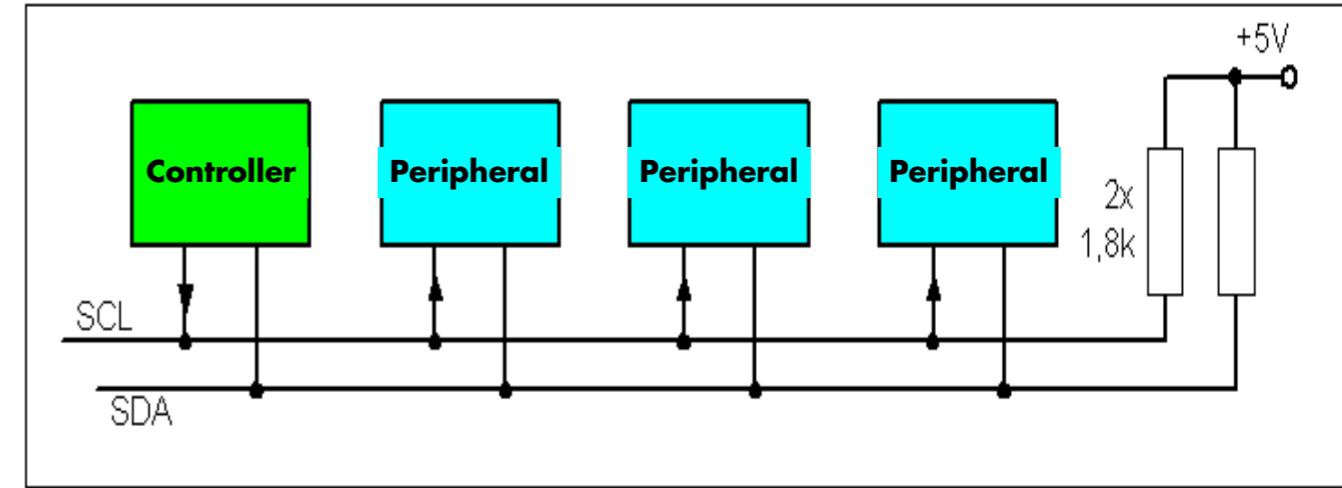


Figures from https://upload.wikimedia.org/wikipedia/commons/thumb/f/fc/SPI_three_slaves.svg/2000px-SPI_three_slaves.svg.png (top), <http://www.tequipment.net/RigolSD-SPI-DS4.html> (bottom)

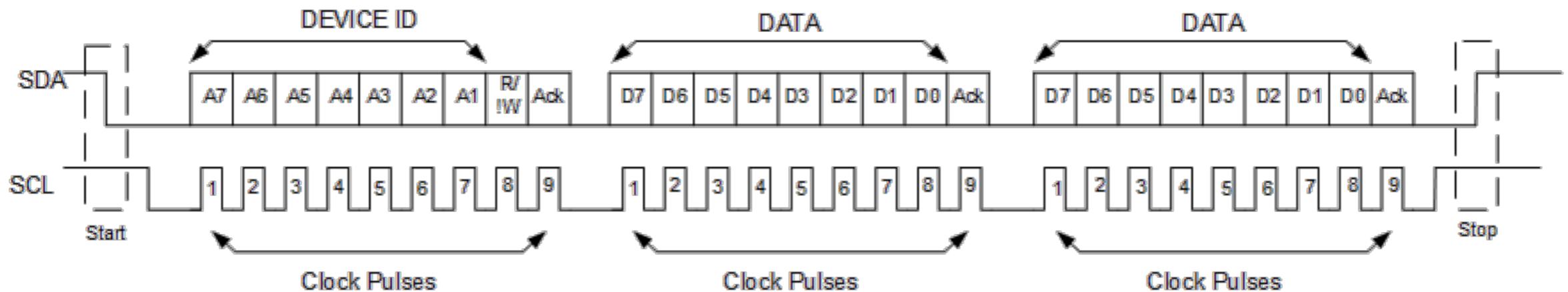


MCP3008 SPI 8-channel ADC

I2C

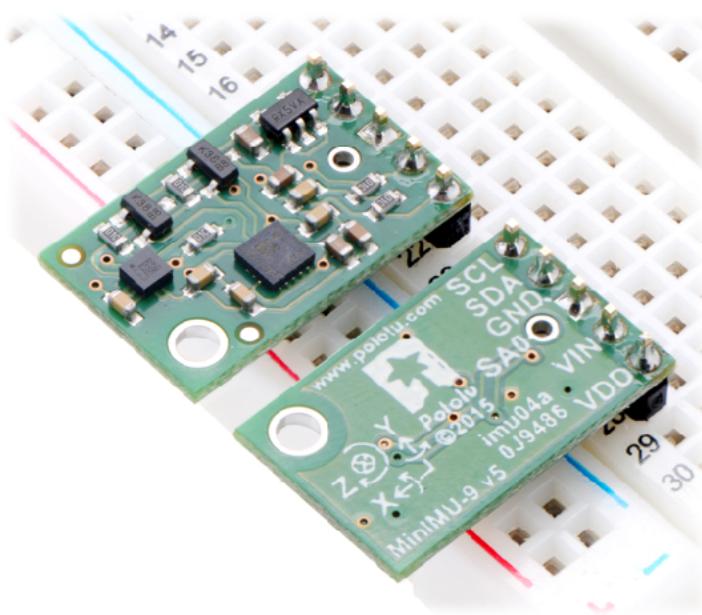


- **CLK & DATA lines (with pull-ups)**
- **Clocked by controller, both controller and peripheral send data**
- **Shared bus, peripheral identified by 7 (or 10) bit address**

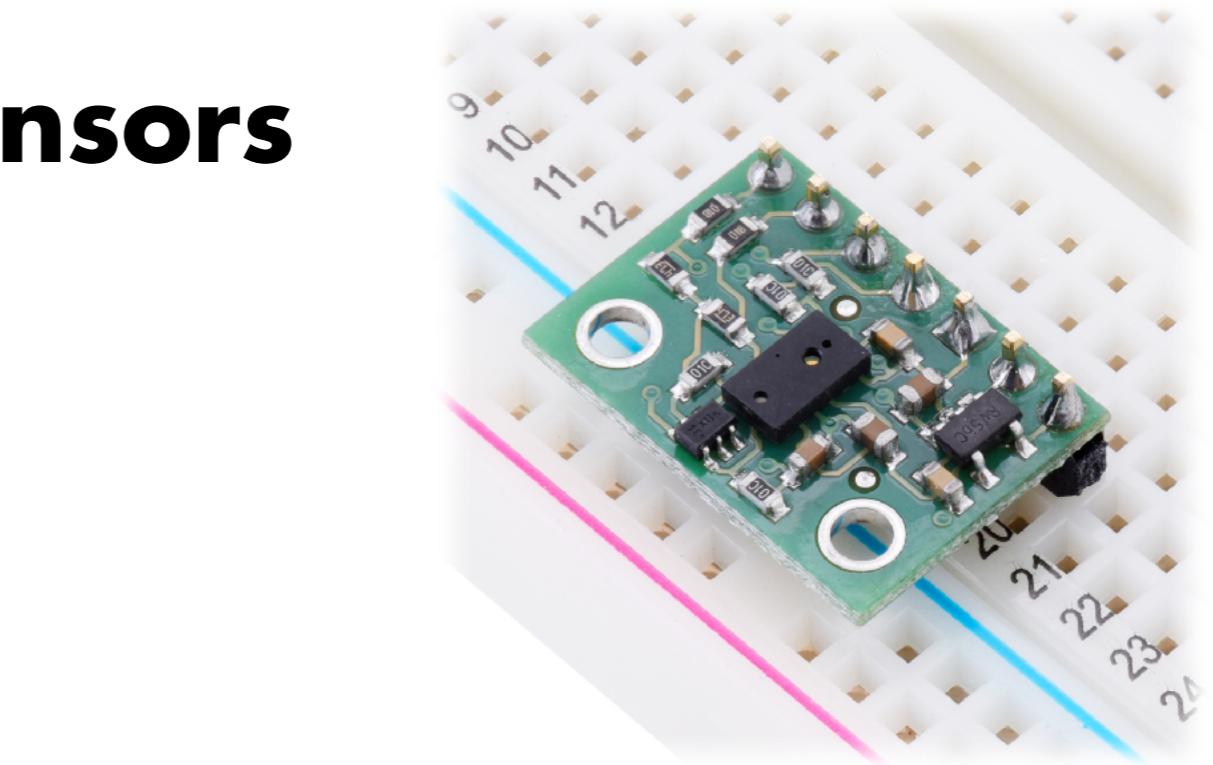


Figures from <http://www.cs.fsu.edu/~baker/devices/notes/graphics/i2cbus3.gif> (top)
https://learn.digilentinc.com/Documents/chipKIT/chipKITPro/P08/Fig_1_Waveform.png (bottom)

I2C Sensors



**Accelerometer
Gyroscope
Magnetometer**



VCSEL Time of Flight

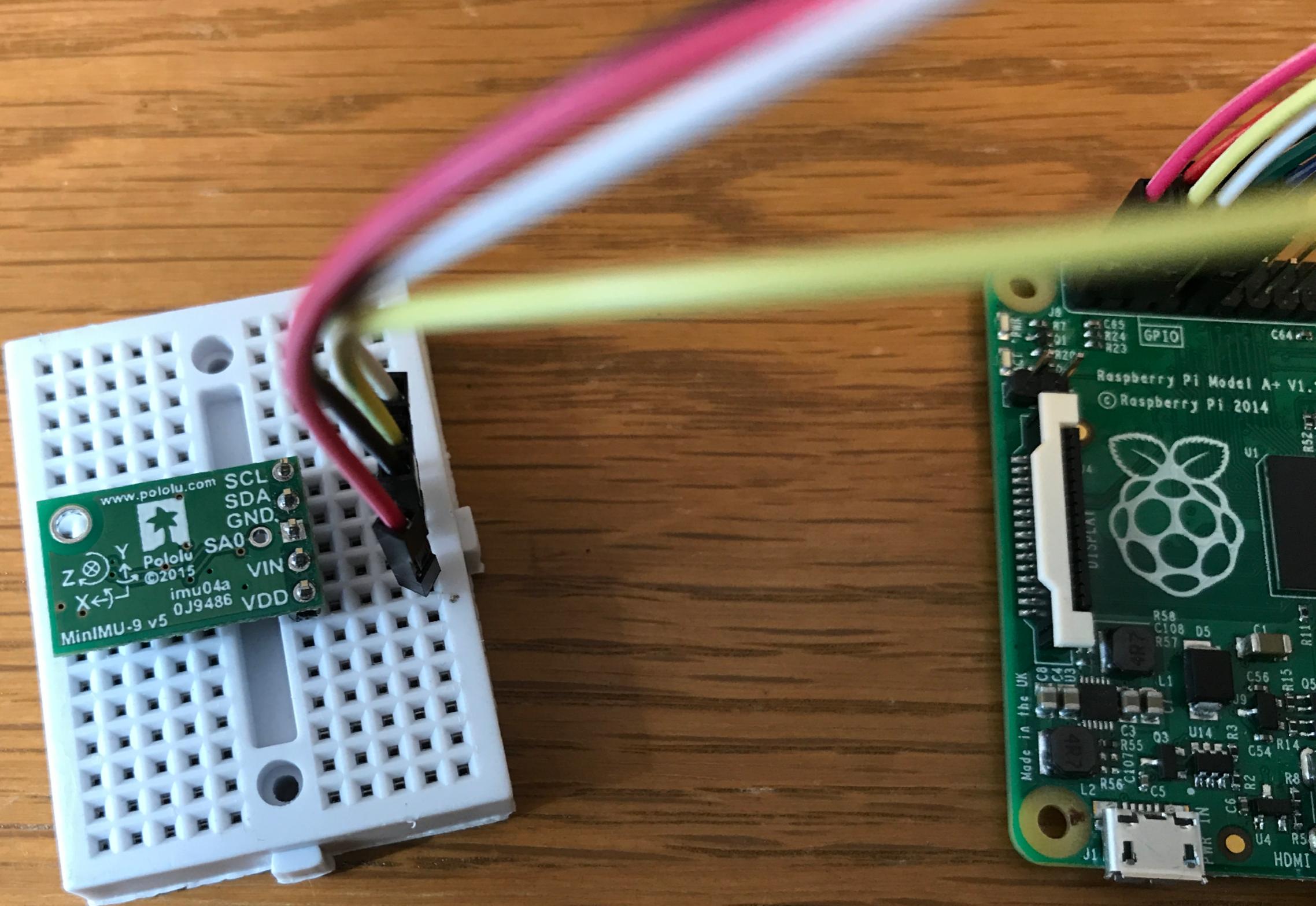


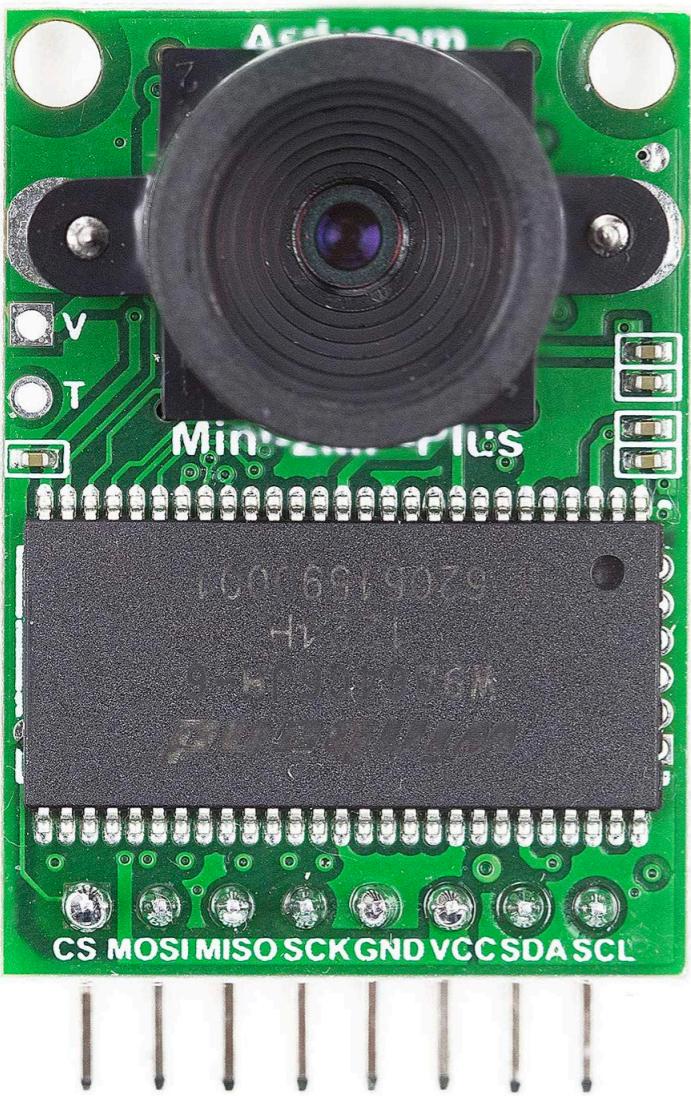
**Temperature,
Humidity,
Pressure**



Arducam (SPI and I2C)

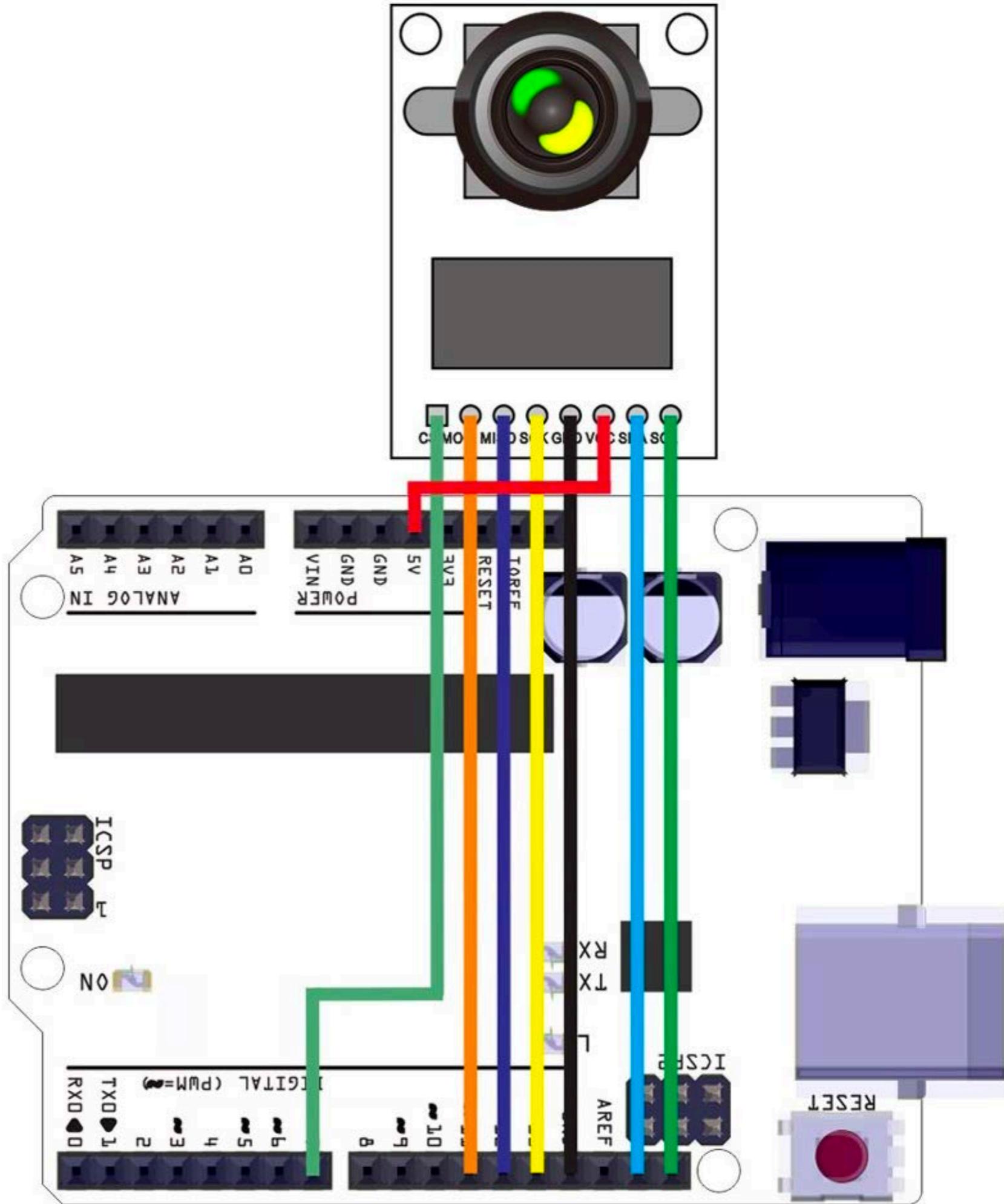
I²C IMU (accelerometer, gyroscope, compass)



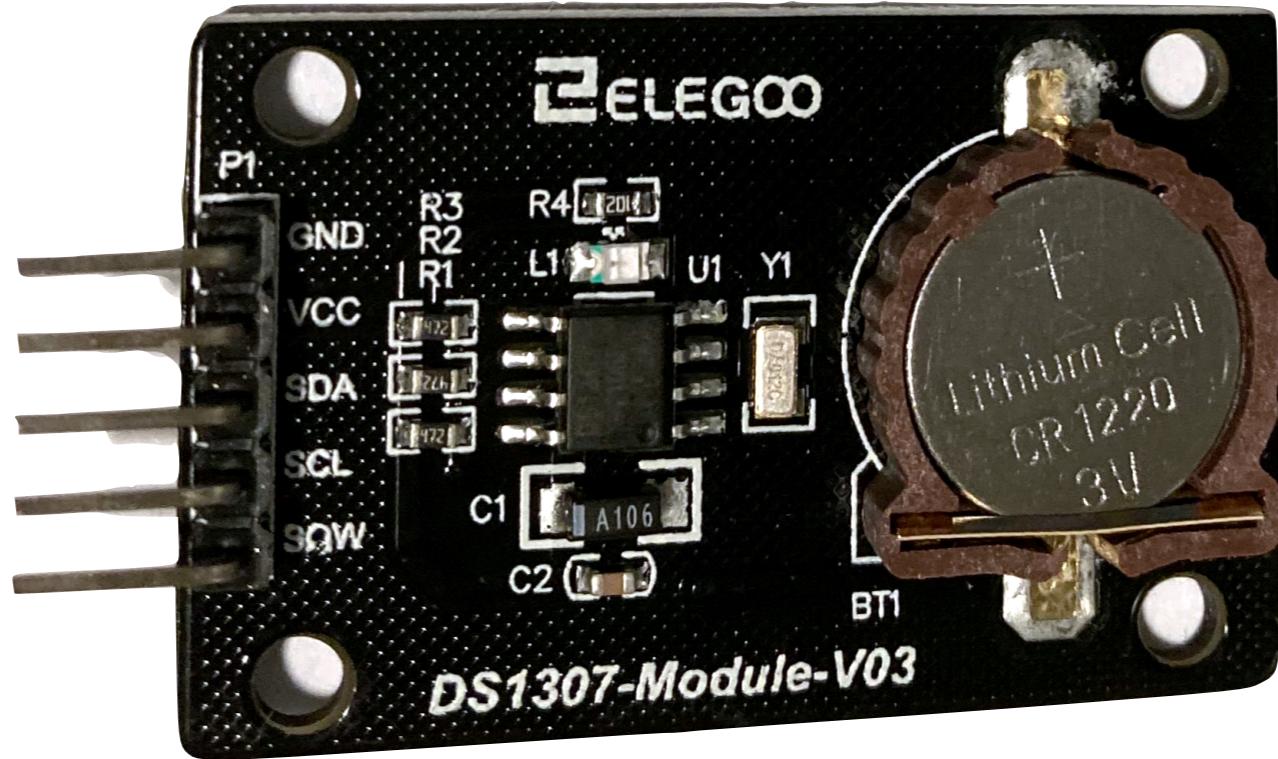


Arducam

SPI + I2C

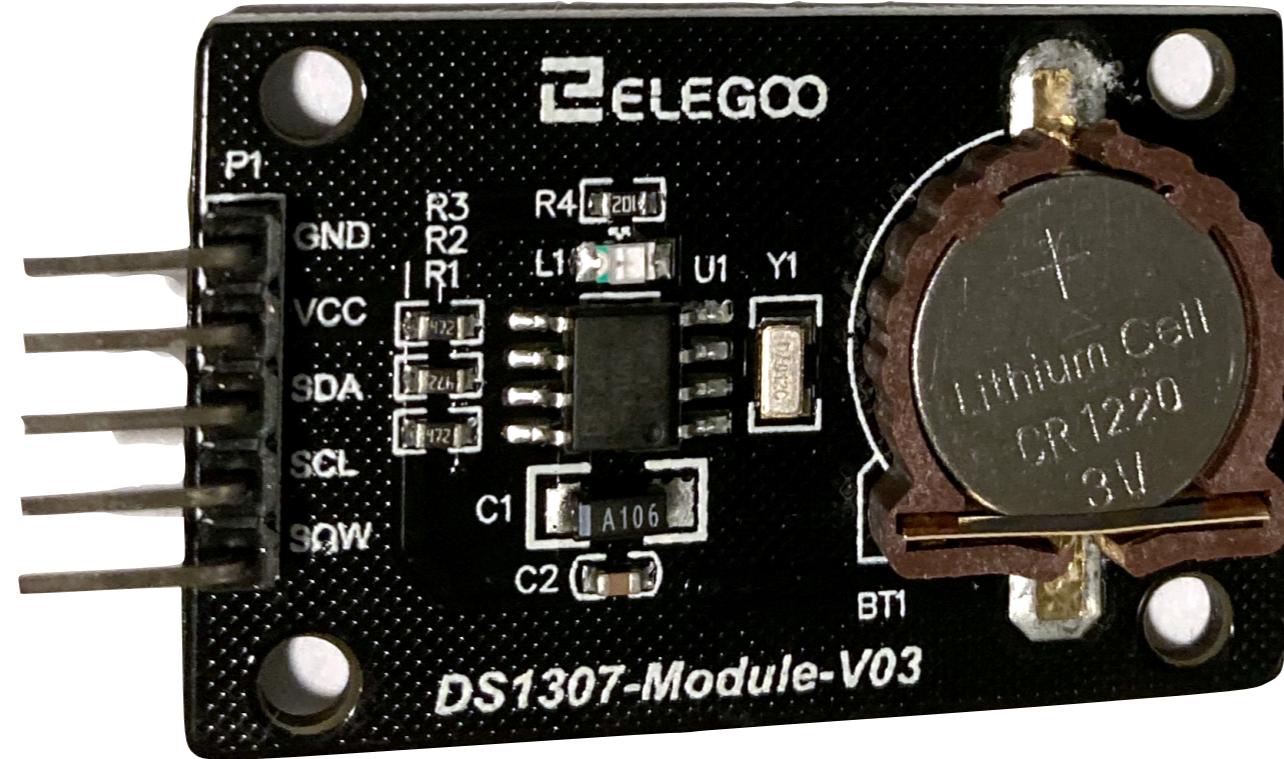


I2C Example: DS1307 Real Time Clock Module



- Five pins: VCC/GND, SDA, SCL, SQW
 - SQW is a "square wave," which we won't demonstrate here. The other pins are for I2C
- Battery – keeps the time going when power is off!
- Has 56 bytes of RAM that you can also use!

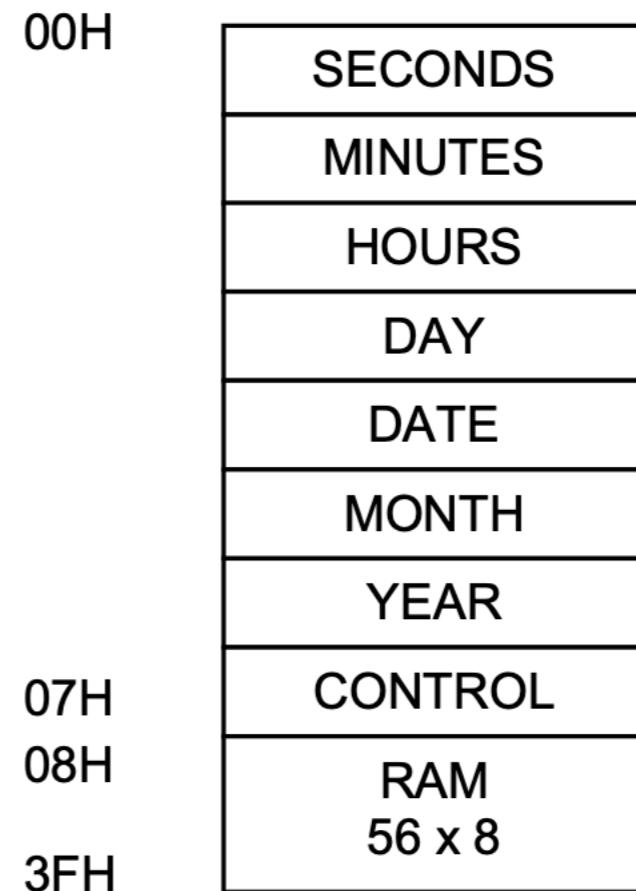
I2C Example: DS1307 Real Time Clock Module



- We need to read the data sheet to figure it out...
- <https://www.sparkfun.com/datasheets/Components/DS1307.pdf>

I2C Example: DS1307 Real Time Clock Module

DS1307 ADDRESS MAP Figure 2



- We send an I2C write with the address we want to read. If we send 0, we can read all 64 bytes

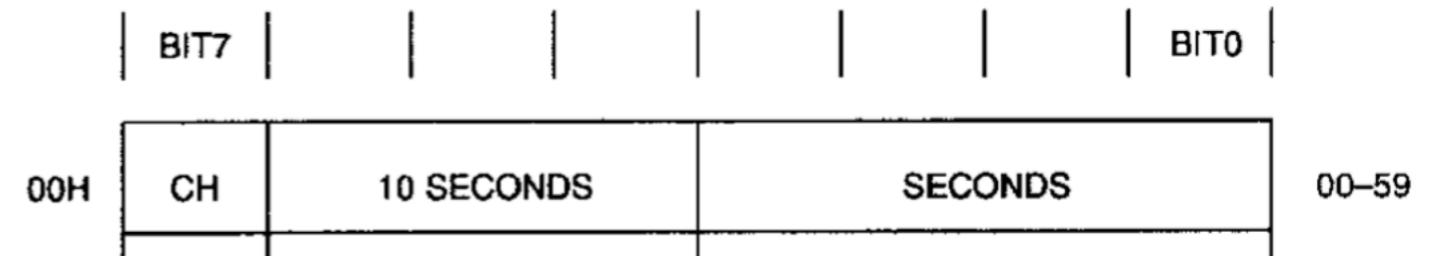
I2C Example: DS1307 Real Time Clock Module

DS1307 TIMEKEEPER REGISTERS Figure 3

- This looks confusing...let's see if we can figure it out (this sometimes takes trial and error!)
- The year is only two digits! (Y2K problem...)
- What would 0b01110011 at address 0 mean?

| | BIT7 | | | | | | | BIT0 | |
|-----|---------|------------|--------------|-------|---------|-------|-----|----------------------------|-------|
| 00H | CH | 10 SECONDS | | | SECONDS | | | 00-59 | |
| | 0 | 10 MINUTES | | | MINUTES | | | 00-59 | |
| | 0 | 12 24 | 10 HR A/P | 10 HR | HOURS | | | 01-12 00-23 | |
| | 0 | 0 | 0 | 0 | 0 | DAY | | 1-7 | |
| | 0 | 0 | 10 DATE | | | DATE | | 01-28/29 01-30 01-31 | |
| | 0 | 0 | 0 | 10 | MONTH | MONTH | | 01-12 | |
| | 10 YEAR | | | | YEAR | | | | 00-99 |
| 07H | OUT | 0 | 0 | SQWE | 0 | 0 | RS1 | RS0 | |

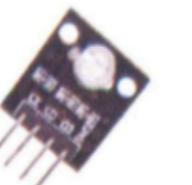
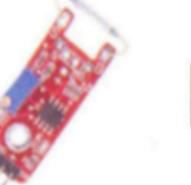
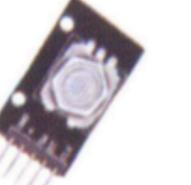
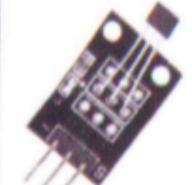
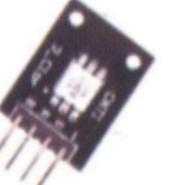
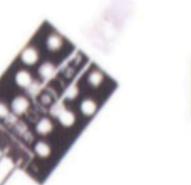
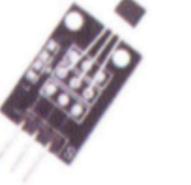
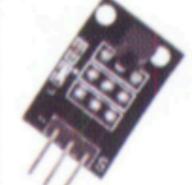
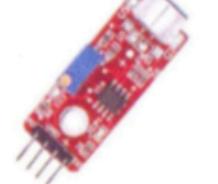
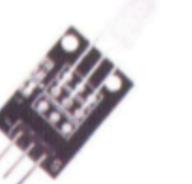
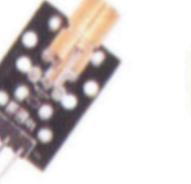
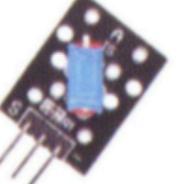
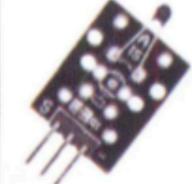
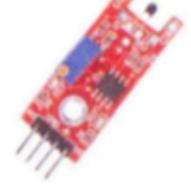
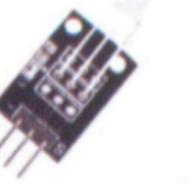
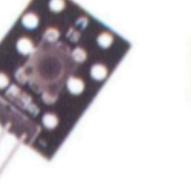
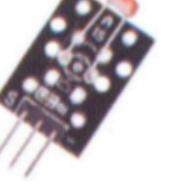
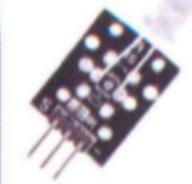
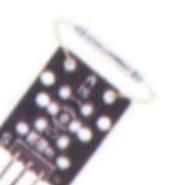
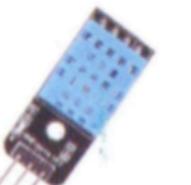
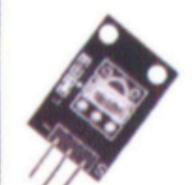
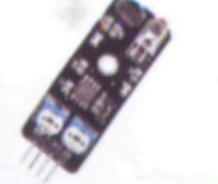
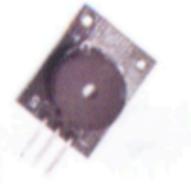
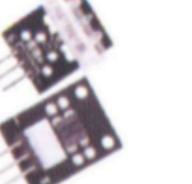
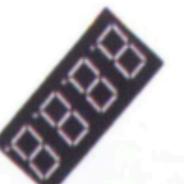
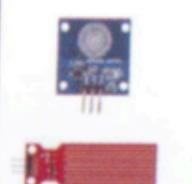
I2C Example: DS1307 Real Time Clock Module



- What would 0b01010011 at address 0 mean?
 - Bit 7: ignore (sets the oscillator on/off). Will be 0 when the oscillator is going (e.g., when it is telling time).
 - Bits 6-4 are the "tens" digit. In this case, 0b101 is 5, or "50"
 - Bits 3-0 are the "ones" digit. In this case, 0b0011 is 3
 - Therefore, the seconds is 53.
 - Why isn't it just the actual numerical 53?
 - This is called "Binary Coded Decimal", which is an older standard.
 - The hex value will actually represent a decimal value... e.g., 0b01010011 = 0x53.

What other sensors are available?

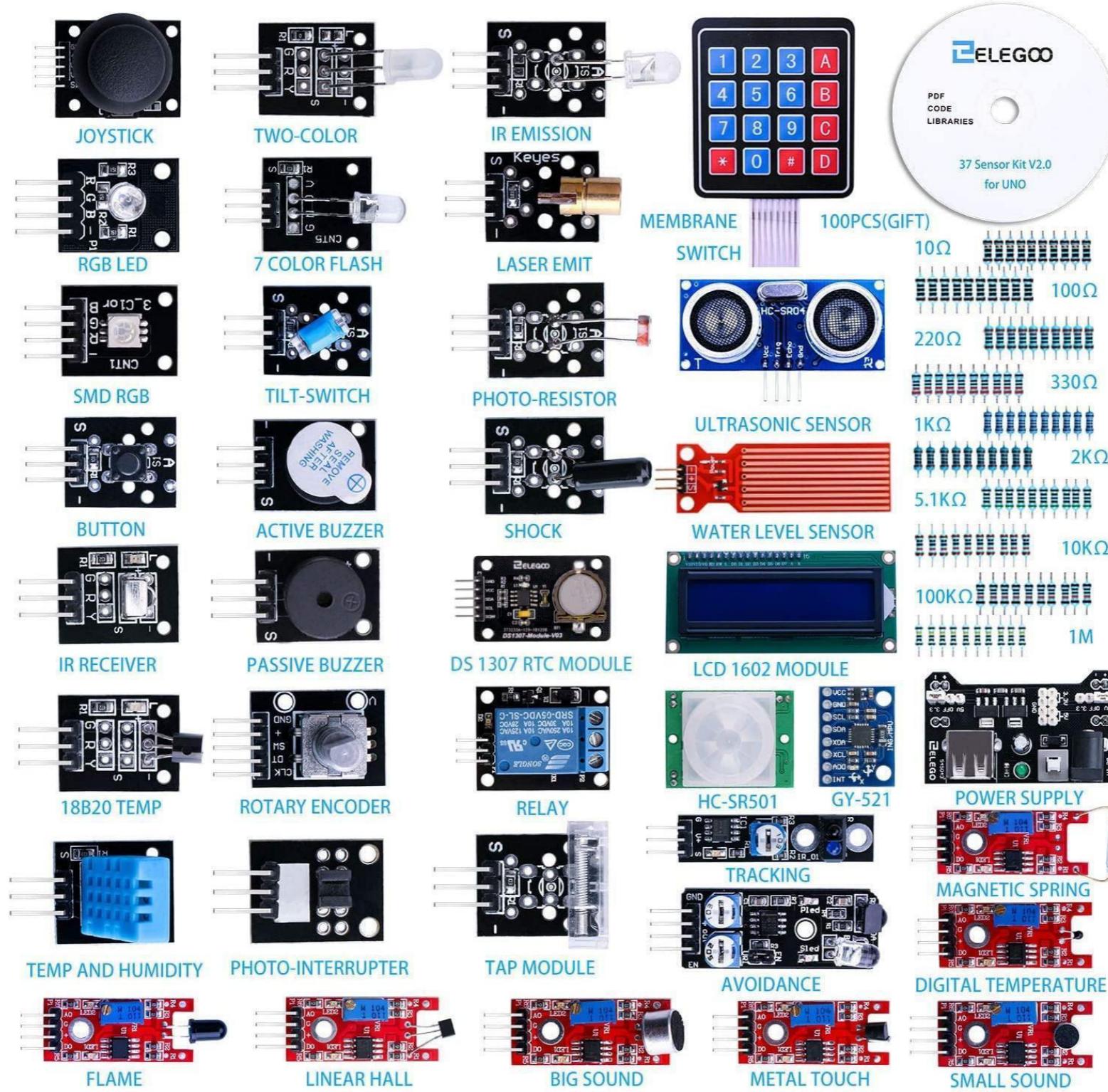
sensor kit

| | | | | | |
|--|---|--|---|---|---|
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

sensor kit #2

ELEGOO sensor kit

Download code: <https://www.elegoo.com/blogs/arduino-projects/elegoo-37-in-1-sensor-modules-kit-tutorial>



Sensing the World

Resistance (conduction, capacitance)-

Convert energy to voltage/current

- **Light (phototransistor)**
- **Sound/pressure/deformation (piezo, electret, strain gauge)**
- **Temperature (heat), humidity, pressure**
- **Electromagnetic fields (hall effect, compass, antenna)**

Smart sensors (sensor with a digital interface)

- **Acceleration/Orientation/Magnetic (force direction)**
- **Camera, IMU (inertial management unit), ...**