Physics 124: Lecture 7

Sensors

(always incomplete)

adapted from T. Murphy's lectures

Sensing Categories

- Voltage
  - starting easy: analog in
- Distance
  - acoustic or light
- Speed
  - hard; usu. via distance
- Acceleration
  - accelerometers
- Light Level
  - phototransistors, photodiodes
- Image
  - CCD cameras
- Object Passage
  - photogate (light source/sense)
- Sound Level
  - microphone to rectifier?
- Temperature
  - RTD, thermistor, AD-590
- Magnetic Flux
  - coil and EMF
- Pressure
  - pads?
- Mass
  - spring stretch?
- Strain
  - strain gauge
- Radiation

http://en.wikipedia.org/wiki/List_of_sensors for overwhelming list

Voltage

- Crudest version is digital: HIGH or LOW: 1-bit resolution
  - lots of digital inputs to handle this
  - option for internal pull-up resistor to V_{cc}
- Analog in provides 10-bit (0–1023) on Arduino
  - considered on crude-to-modest side: 5 mV in 5 V
  - high-end is ~~16-bit(65536 values)~~24 bit (16777216) or 28 bit!
    - seldom meaningful to carry more precision than this
  - 12-bit is also common, and 4× improvement over 10-bit
  - 8-bit is painful: 0.02 V in 5 V
    - but fine for some applications
- Voltage is seldom what you fundamentally want to know, but is often the electronic analog of a physical quantity of greater interest
  - generally, "converter" can be termed *transducer*

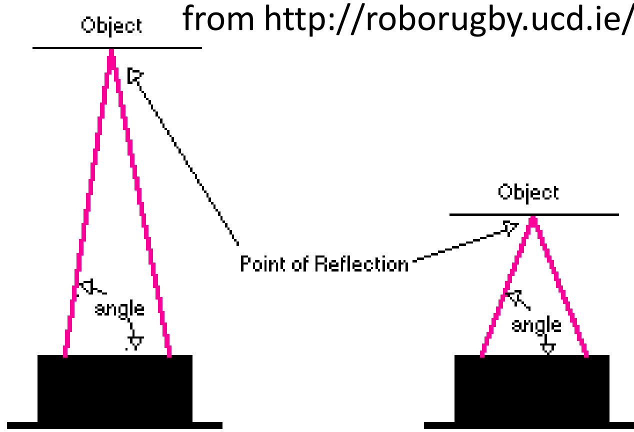
Distance

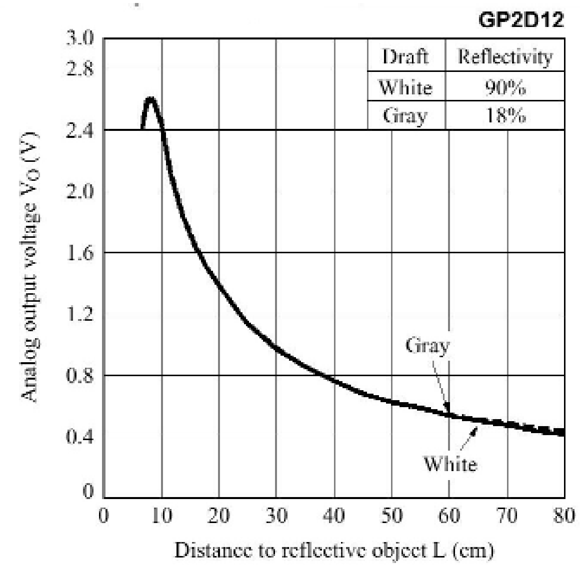
- Popular Phys124 metric
  - collision avoidance; parallel park; target approach
- Acoustic variety
  - ultrasound burst and time-of-flight measurement
  - Parallax Ping unit is integrated unit, $30
    - 2 cm to 3 m (dep. on surface type)
  - must send 2 μs pulse on SIG pin
  - then listen for return pulse
    - duration of pulse is round-trip time
  - must switch same pin between input/output
  - use `pulseIn()` to measure input duration
- Other modules in lab to roll your own acoustic sensor

Distance via Light?

- Not time-of-flight; forget about it! Leave that to pros
- Clever sensing of angle between emitter and receiver

from http://roborugby.ucd.ie/distsensor.html





- Detector is linear array behind lens
  - angle maps to position, indicating distance
- Smarts on board, so GND, +5 V in; analog voltage out proportional to distance, though not linearly so
- Also a proximity version: logic out dep. on “too close”

Laser range finder

- Simple version: laser and ccd camera. Distance is calculated by triangulation between centroid of laser beam, camera, and object. (~$100)

- Time-of-flight system, not affected by speed, wind, pressure changes, noise ambient light or air temperature (>$280)

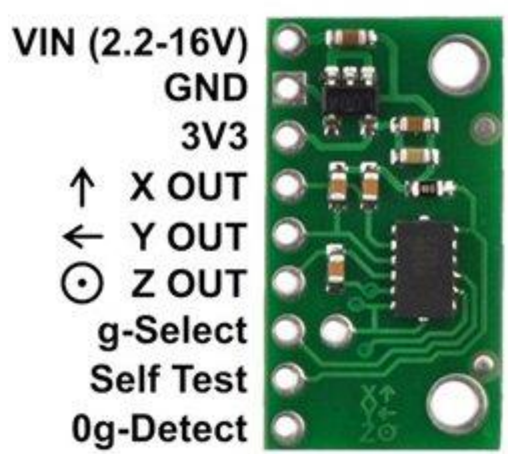
Images from parallax.com

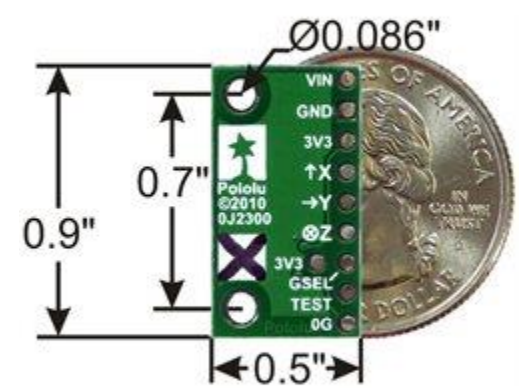
Measure Speed?

- Galileo and Einstein would both agree that this is hard to directly sense
- Options
  - measure distance and rate of change
    - noise in distance measurement can make for ratty/spiky velocity
  - Doppler?
  - measure rotation rate of wheel or axle engaged in motion
    - what speedometers do
    - can use photogate for once/revolution knowledge

Acceleration

- This is something we *can* directly sense
- Recent rapid advances; driven by MEMs and smartphones
  - 3-axis accelerometer based on micro-cantilevers capacitively sensed
  - for example: bitty MMA7361L unit, $15
    - centers output on ½ of 3.3 V
    - default roughly ±1.5g, but can config. for ±6g
    - zero-g detection and digital flag





Acceleration and more

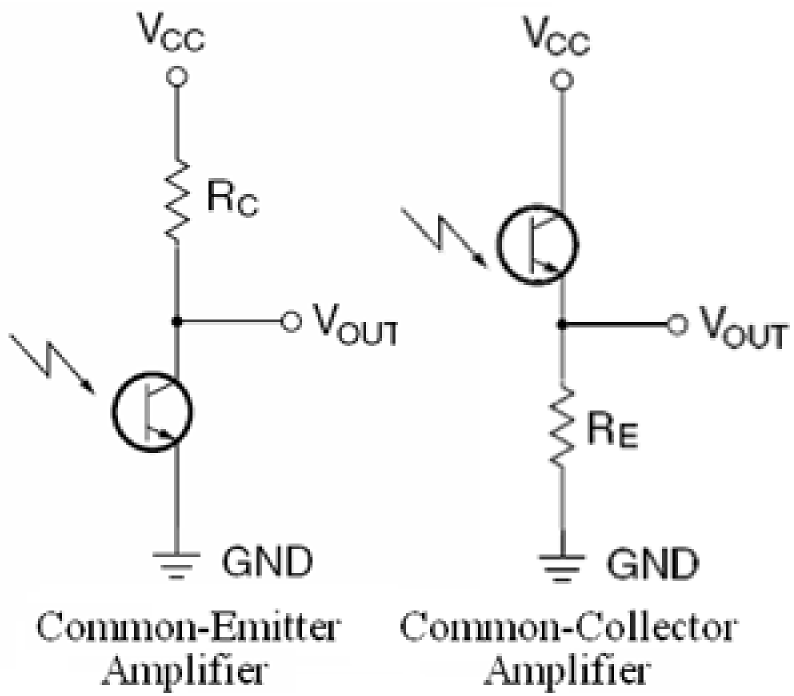
Tilt sensor

Gyroscope (yaw, pitch, roll)
& temperature

Images from parallax.com

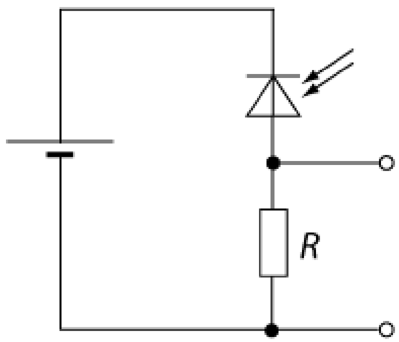
Light Level

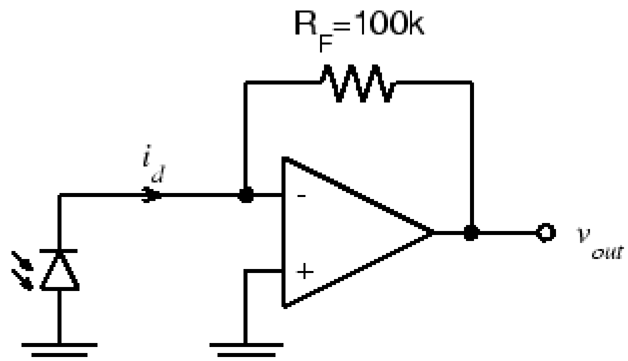
- Lots of options: phototransistor, photodiode most common
  - photons knock electrons loose, which either constitute a base current (phototransistor) or direct into current (photodiode)
- Phototransistor (right) effectively has some gain already
  - 10 kΩ usually about right

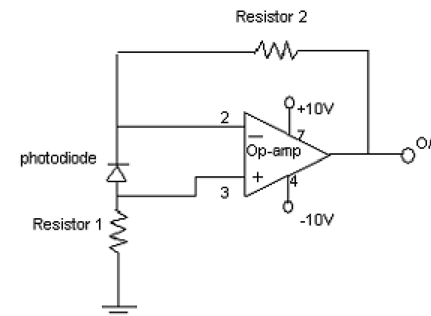


Photodiode Read Out

- Many options for photodiode
  - reverse bias, developing voltage across resistor
  - zero bias, in op-amp feedback mode

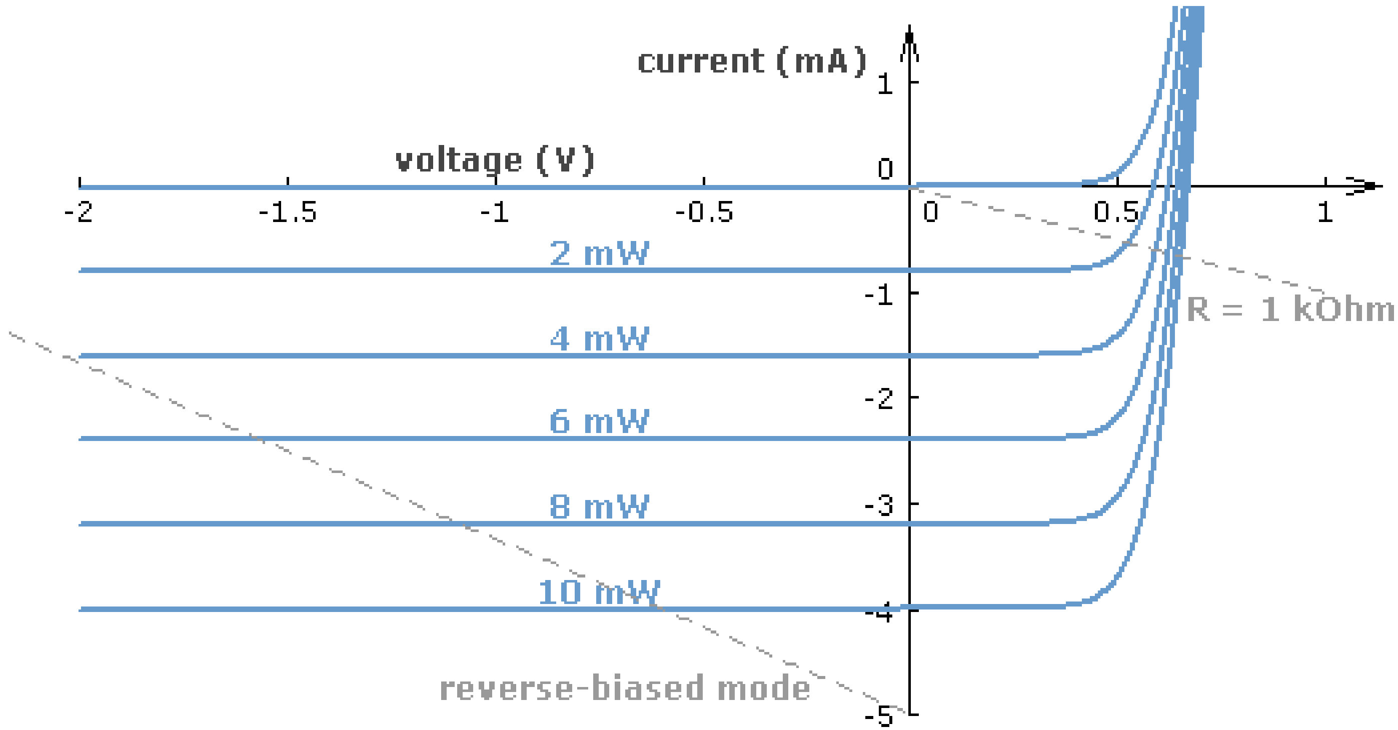






- Typically < 0.4 A per Watt incident
  - stream of photons at 550 nm → 0.447 A at 100% Q.E.
  - so 1 mm^2 detector in full sun (1000 W/m^2) is 1 mW
  - thus at best 0.5 mA current (puny)
  - tend to want pretty large resistor to build up voltage

Photodiode IV Curve



- At zero or reverse bias, current is proportional to incident light power
  - note approximate relation: $I \approx 0.4P$
  - matches quantum expectations

Object Passage

- We often need to know if something is physically present, has passed through, count rotations, etc.
- Can have simple scheme of light source and light detector, where the something of interest passes between
  - termed a *photogate*
  - interruption of light level pretty unmistakably sensed
  - pulse duration, via `pulseIn()`, may even speak to velocity
- Magnetic
  - as in bicycle speedometers

Temperature

- Exploit temperature dependence of materials
  - RTD: resistive temperature device
    - usually laser-etched platinum spiral, often 1000 Ω + 3.85×(T °C)Ω
    - linear, good absolute calibration
    - but a resistor: need to fashion accurate current source and read off voltage (make ohmmeter)
  - thermistor: exploits conduction electron density as e^{T}
    - nonlinear, due to exponential dependence on T
  - AD-590: Analog Devices
    - supply 5 V and a route for current (resistor), and output current is proportional to temperature
    - measure current as voltage across provided resistor
- Caution: resistors often 200 ppm per °C
  - for accuracy, may want low "tempco" resistors

Sound Level

- Microphone is transducer for acoustic vibrations into voltage
  - usually membrane that vibrates is part of capacitor
  - can rectify resulting waveform, low-pass, and measure level

Magnetic Flux

- A loop of wire (or many loops) will develop EMF according to changing magnetic field
  - can amplify, rectify, etc.
- A Hall sensor can measure DC magnetic field

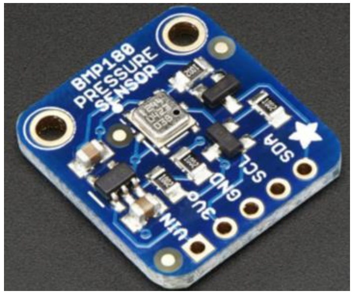
Triple axis magnetometer

Triple-axis accelerometer + magnetometer, $15

Pressure

- Pressure pads: 2 conductors separated by carbon film, squeezes out; so more conductivity: bite pads
- Capacitive pressure deflects membrane (lab pressure meter)
- Party-roller paper tube

Altimeter
& temperature

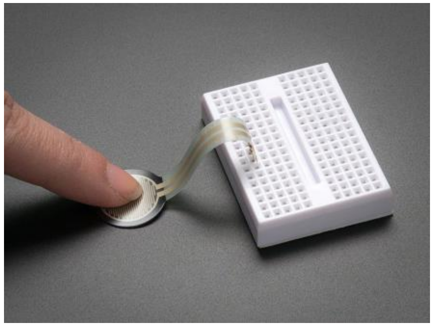


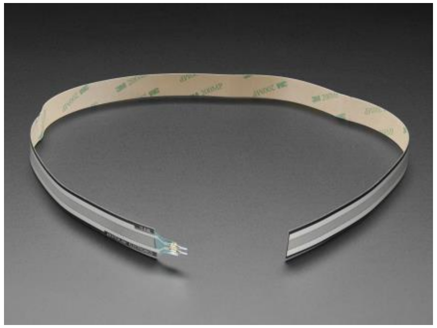
Barometric pressure
Temperature
Altitude
(0.03hPa, 0.25m resolution?)
$10

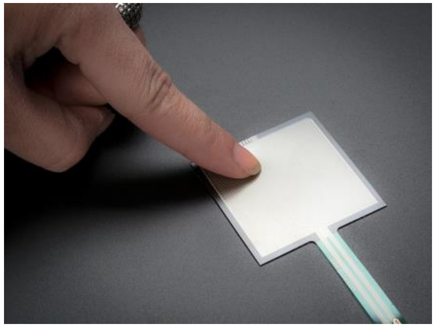
1 hPa, 10m resolution
$8

Mass/Weight

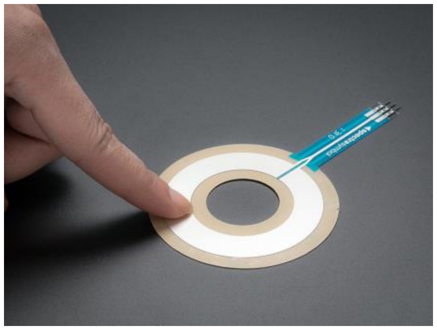
- “Spring” stretch plus flexometer (strain gauge)
- FSR: Force-Sensitive resistor

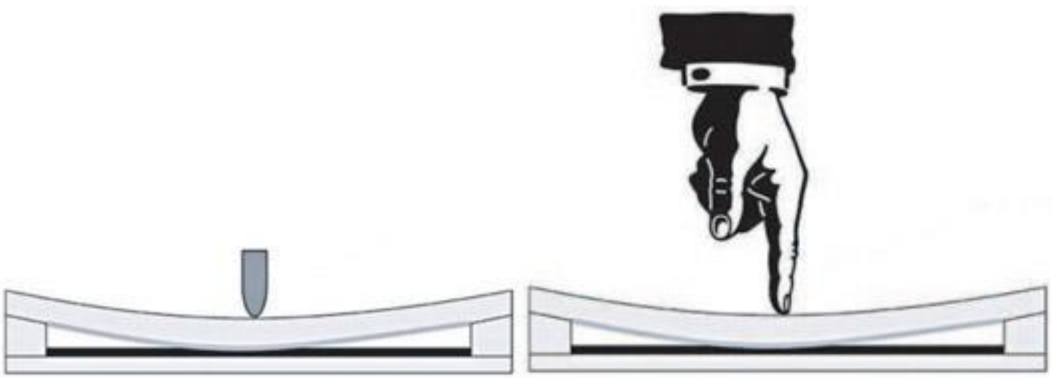






Soft potentiometers

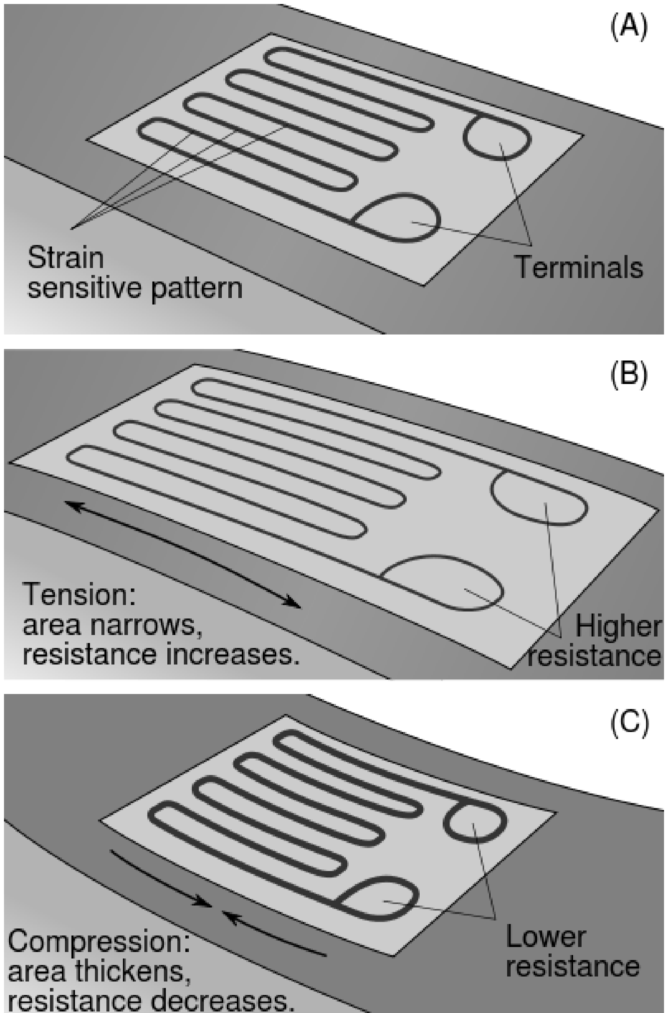




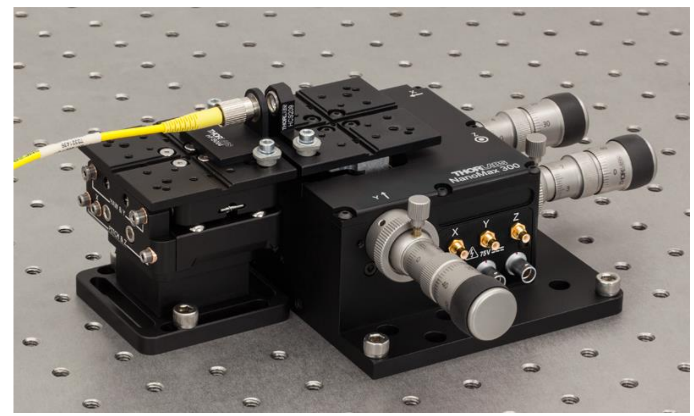
http://www.spectrasymbol.com/potentiometer/softpot/how-it-works-softpot

Strain

- Strain gauge can tell you about minute flexing of a structural beam/material



enable 5 nm resolution



Wikipedia.org + Thorlabs.com

Radiation: Geiger

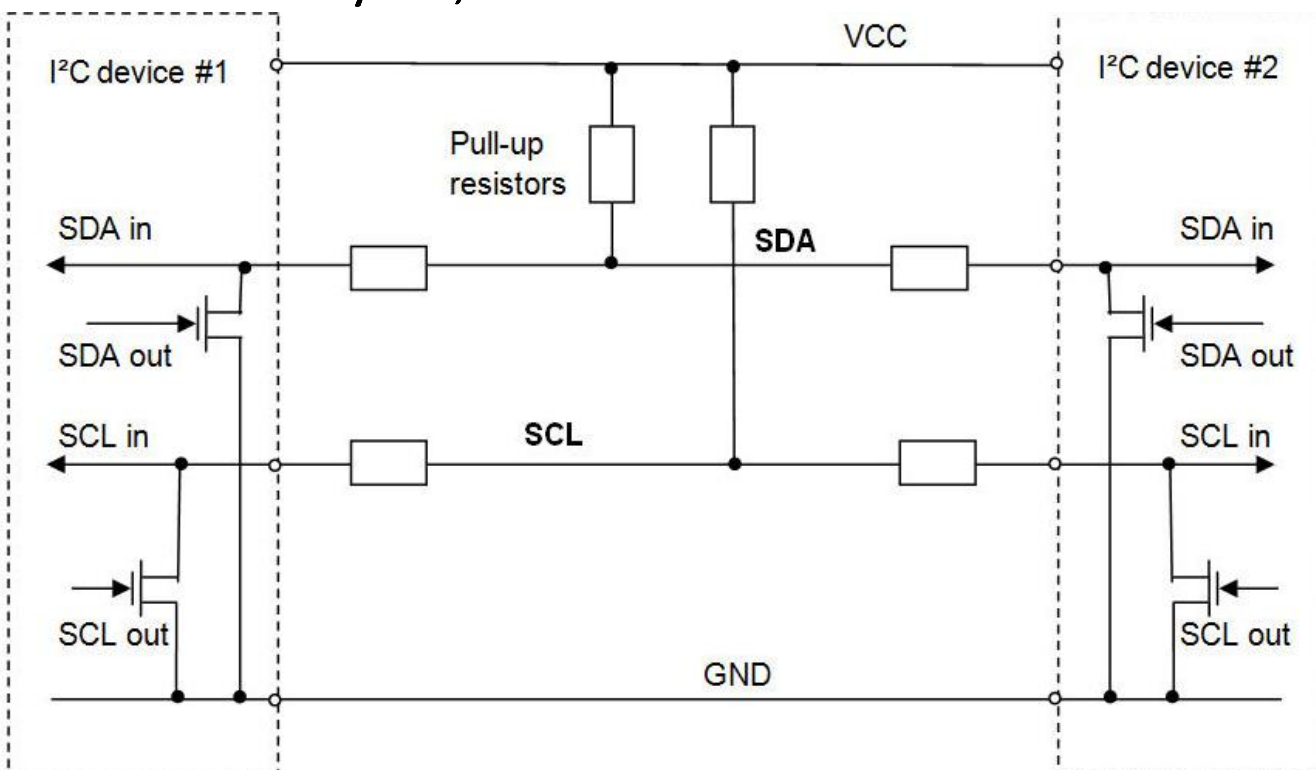
$100, Adafruit.com

Other Sensors

- Direction
  - HM55B Compass Module from Parallax ($30)
- Motion
  - infrared motion sensor
- CCD

Most sensors use I2C serial protocol

- I^2C is a multi-master protocol using 2 signal lines:
  - SDA: Serial data, SCL: Serial clock
  - Data rate 100 kbps, 400 kps, 3.4 Mbps
  - 7-bit slave addresses: each device has unique address
  - Data in 8-bit bytes, few control bits for comm control



http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/

I2C on the arduino

- SDA pin A4, SCL pin A5
- #include <Wire.h>
- Example: SRFxx Sonic Range Finder Reader

```
#include <Wire.h>

void setup() {
  Wire.begin();                // join i2c bus (address optional for master)
  Serial.begin(9600);          // start serial communication at 9600bps
}

int reading = 0;

void loop() {
  // step 1: instruct sensor to read echoes
  Wire.beginTransmission(112); // transmit to device #112 (0x70)
  // the address specified in the datasheet is 224 (0xE0)
  // but i2c adressing uses the high 7 bits so it's 112
  Wire.write(byte(0x00));      // sets register pointer to the command register (0x00)
  Wire.write(byte(0x50));      // command sensor to measure in "inches" (0x50)
  // use 0x51 for centimeters
  // use 0x52 for ping microseconds
  Wire.endTransmission();      // stop transmitting
```

← Writing part

Arduino.cc

```
  // step 2: wait for readings to happen
  delay(70);                   // datasheet suggests at least 65 milliseconds

  // step 3: instruct sensor to return a particular echo reading
  Wire.beginTransmission(112); // transmit to device #112
  Wire.write(byte(0x02));      // sets register pointer to echo #1 register (0x02)
  Wire.endTransmission();      // stop transmitting

  // step 4: request reading from sensor
  Wire.requestFrom(112, 2);    // request 2 bytes from slave device #112

  // step 5: receive reading from sensor
  if (2 <= Wire.available()) { // if two bytes were received
    reading = Wire.read();  // receive high byte (overwrites previous reading)
    reading = reading << 8;    // shift high byte to be high 8 bits
    reading |= Wire.read(); // receive low byte as lower 8 bits
    Serial.println(reading);   // print the reading
  }

  delay(250);                  // wait a bit since people have to read the output :)
}
```

← Reading part

Arduino.cc