UCDAVIS
UNIVERSITY OF CALIFORNIA

Internet of Things

Senior Design Project Course

Sensing – Part 1

Lecturer: Avesta Sasan

University of California Davis

Flow of Data in Internet of Things (Review)

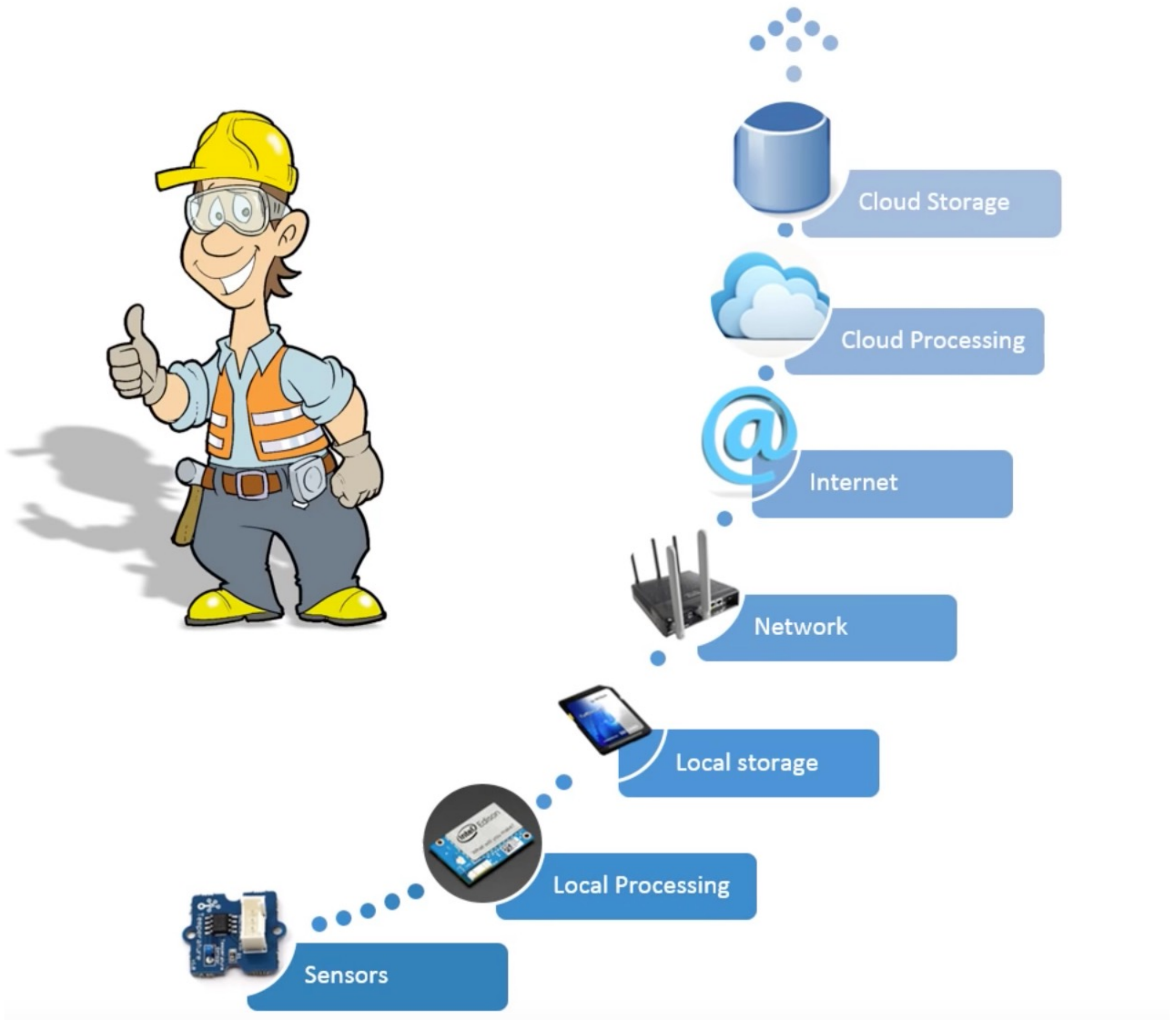


Image source: http://www.cchc.cl/informacion-a-la-comunidad/industria-de-la-construccion/personaje/

Where to Compute? (Review!)

- **Device centric:**
  - microcontroller in an IoT device can be exploited to perform the computation.
  - challenges are
    - scarce resources on IoT devices
    - runtime decision
- **Gateway centric**
  - gate- way devices which are used to settle the heterogeneity between different networks and Internet usually have more computational power
    - This scheme has been used for medical and healthcare monitoring application
    - challenge
      - guarantee the availability and deadline constraints

F. Samie, L. Bauer and J. Henkel, "IoT technologies for embedded computing: A survey," *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, Pittsburgh, PA, 2016, pp. 1-10.

Where to Compute? (Review!)

- **Fog centric**
  - Fogs provide more computational power compared to IoT embedded and gateway devices and have less latency compared to the cloud servers
- **Cloud centric**
  - massive data storage volume, huge processing resources
  - challenges
    - Size of data (big data)
    - scalability
    - high energy cost
    - latency
    - bandwidth
    - availability

F. Samie, L. Bauer and J. Henkel, "IoT technologies for embedded computing: A survey," *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, Pittsburgh, PA, 2016, pp. 1-10.

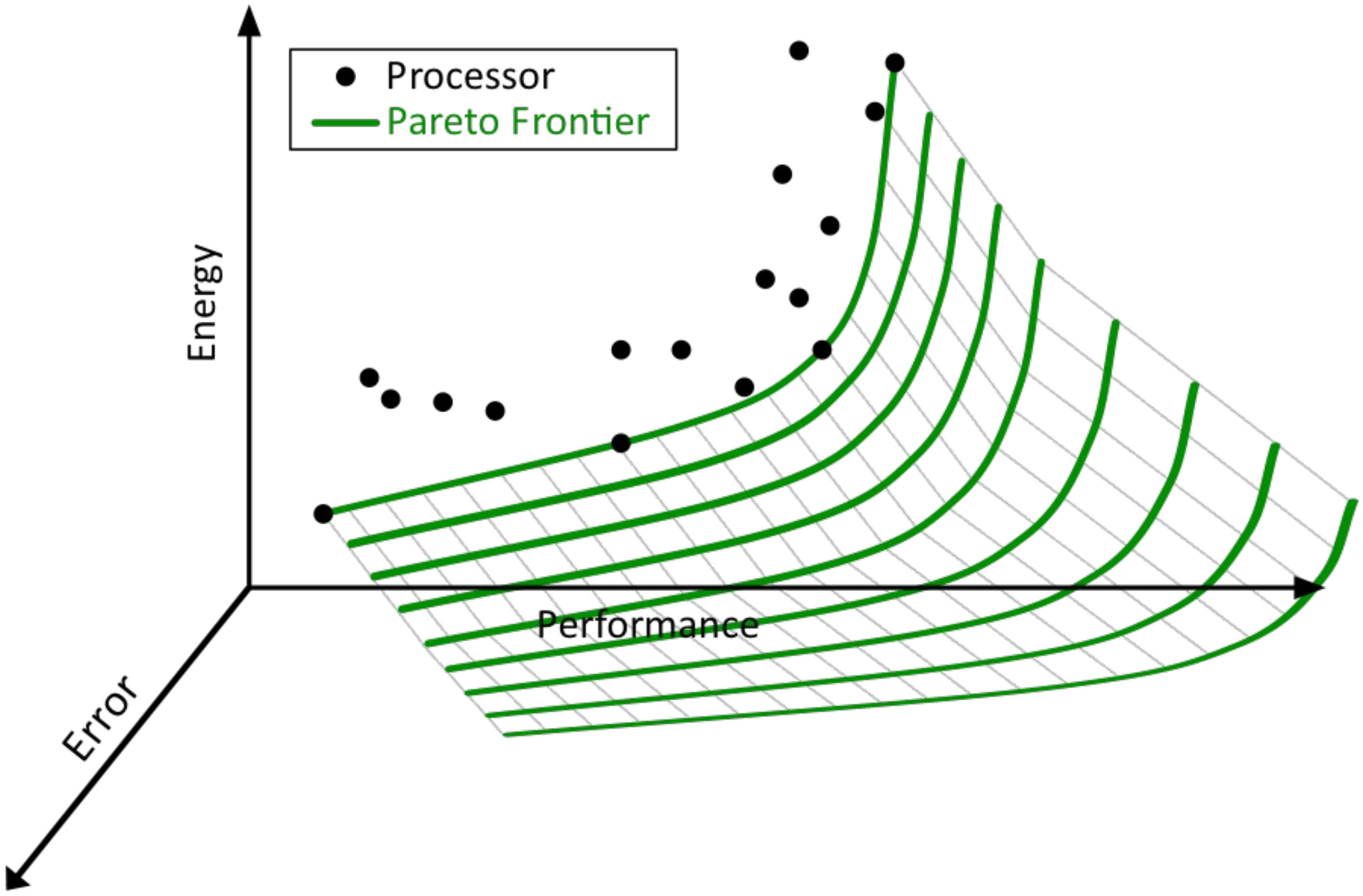
Approximate Computing

- 100% accuracy is not always required!
  - Example: Multimedia
    - 8 bits represent 256 shades of gray
    - Human can only recognize 27 shades of gray

- **Definition:** a computation which returns a possibly inaccurate result rather than a guaranteed accurate result, for a situation where an approximate result is sufficient for a purpose.
- leverages inherent resilience of applications
- relaxes the requirement of exact equivalence between the specification and implementation [1]
- Advantages: Power, Performance, Area (PPA)
- We will cover this in more details!

[1] F. Samie, L. Bauer and J. Henkel, "IoT technologies for embedded computing: A survey," *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, Pittsburgh, PA, 2016, pp. 1-10.

Approximate computing design space



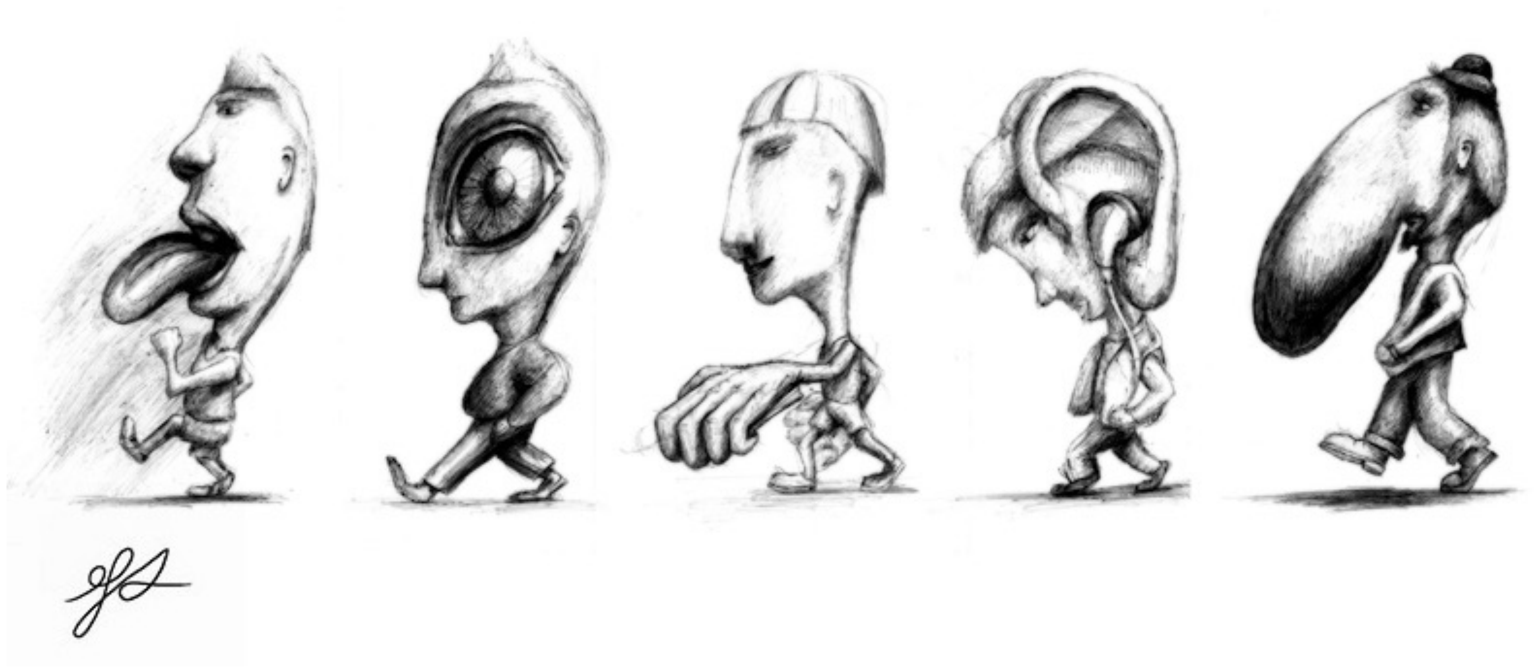
Approximate computing in IoT

- **Data acquisition**
  - reducing the quality of input
  - to reduce the energy consumption or delay
- **Data processing**
  - designing inexact hardware units
    - Approximate adders, multipliers, DCT, FFT
  - designing inexact software
    - Loop skipping, proliferation, stage skipping
- **Data Storage:**
  - Use tolerable error to
    - reduce the size of memory
    - reduce number of memory accesses
    - reduce energy consumption
    - Improve performance

Approximate Computing to improve the PPA is a Good Topic for your Research project!

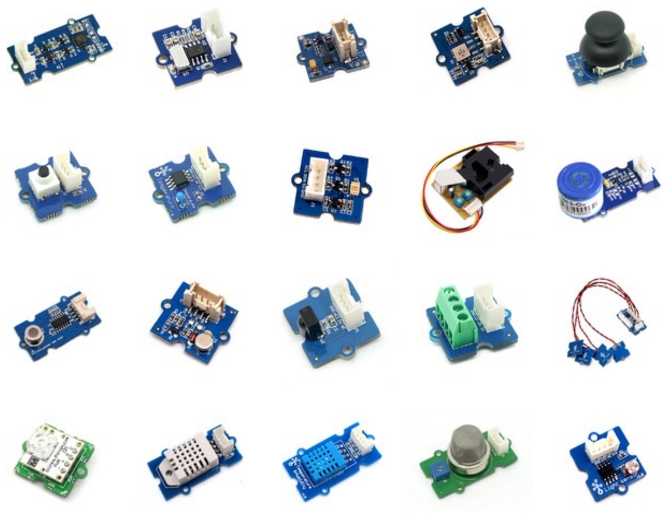
[1] F. Samie, L. Bauer and J. Henkel, "IoT technologies for embedded computing: A survey," *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, Pittsburgh, PA, 2016, pp. 1-10.

Sensing



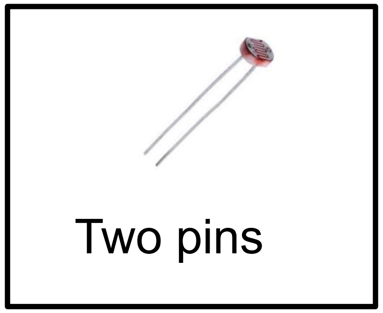
Sensors

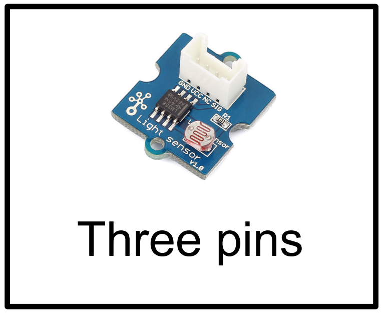
- Measure values
- Send raw data
- For IoT devices it is desired that sensors consume very Low power



Sensor Types

- **Analog**
  - 2 or 3 pins
  - Use pin functions
  - Following article show how analog sensor could be read by Arduino processor:
    - https://www.arduino.cc/en/Tutorial/AnalogInput

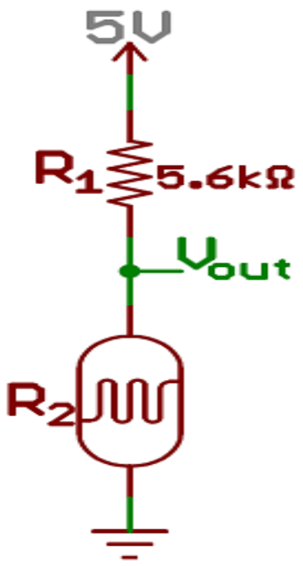

Two pins

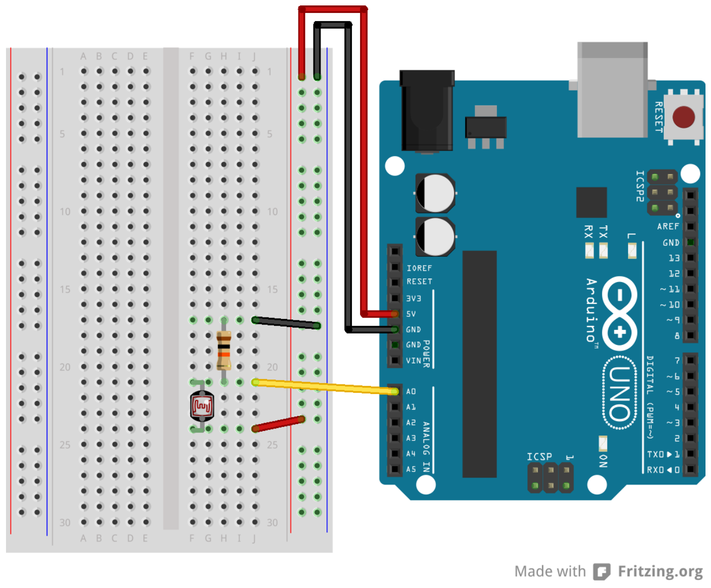

Three pins

- **Digital**
  - Use some digital protocol
  - Use libraries
  - Following article show how digital sensors could be read by Arduino processor:
    - https://www.arduino.cc/en/Tutorial/DigitalReadSerial

Measuring Analog

- In most cases we build a Voltage Divider
- We measure the voltage in V_{out}
- Rule of Thumb :
  - To minimize readout errors read many values and average them





$$V_{out} = \frac{R_2}{(R_1 + R_2)} \cdot V_{supply}$$

Measuring Analog (photoresistor example)

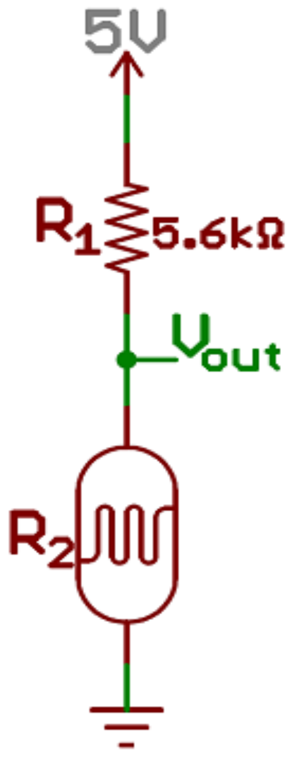
- A resistor divider to allow the **high impedance Analog input** to measure the voltage.

Why high impedance?

- **V_{out}** do not draw any current
  - By Ohm's law the voltage measured on the other end of a resistor connected to 5V is always 5V, regardless the resistor's value.

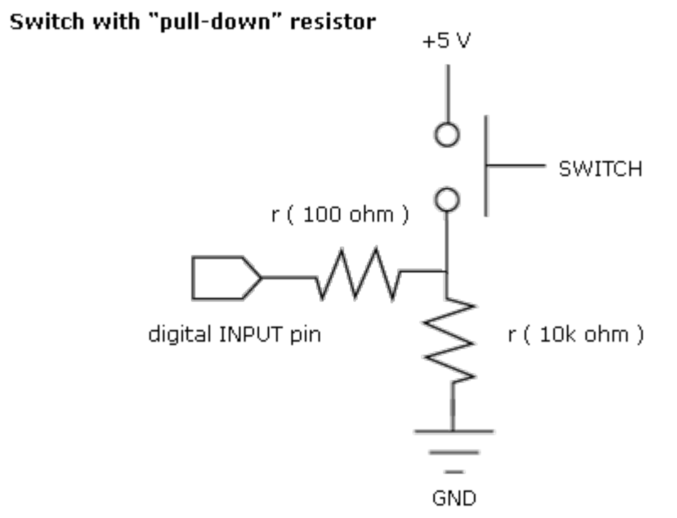
- To get a voltage proportional to the photoresistor value, a voltage divider (resistor divider) is necessary.

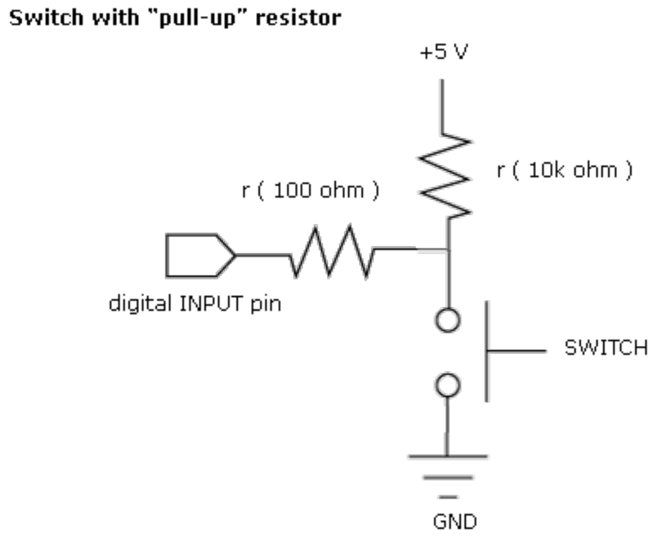
- This circuit uses a variable resistor, a fixed resistor and the measurement point is in the middle of the resistors.

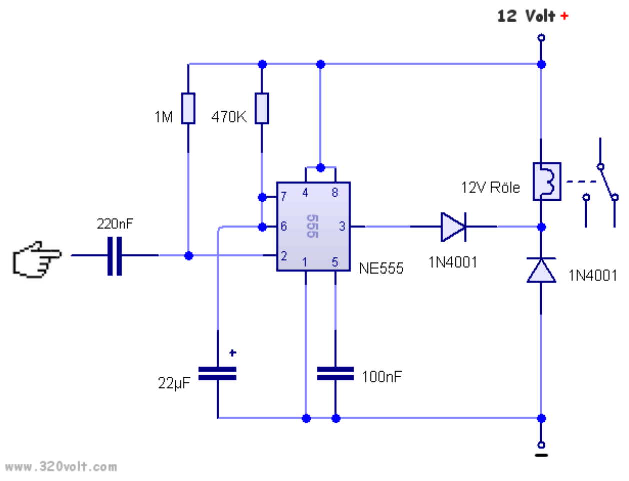


Button is a Sensor!

- Button is an analog sensor!
- We measure **change in resistance:**
  - Infinite → if released
  - 0 → if pressed







Button Debounce

When you push the switch, it initially makes contact with the other metal part, but just in a brief split of a microsecond. Then it makes contact a little longer, and then again a little longer. In the end the switch is fully closed. **The switch is bouncing between in-contact, and not in-contact.**

- **Problem:** When the button is pressed
  - Signal is bouncing, Fast reading results in 0s and 1s
- **Solutions:**
  - Read more values and average them (analogue solution)
  - Use a debouncer (RC based or SR based) → **converting it to a digital input**!

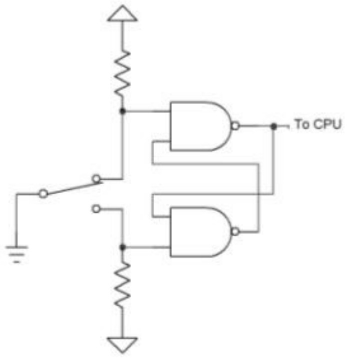


Figure 1: The SR debouncer

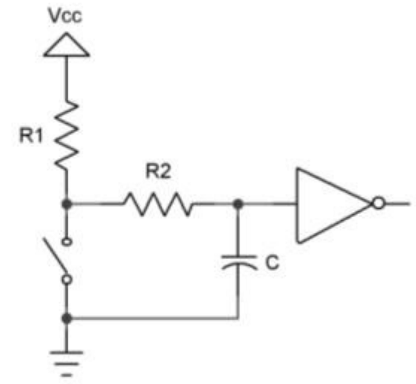
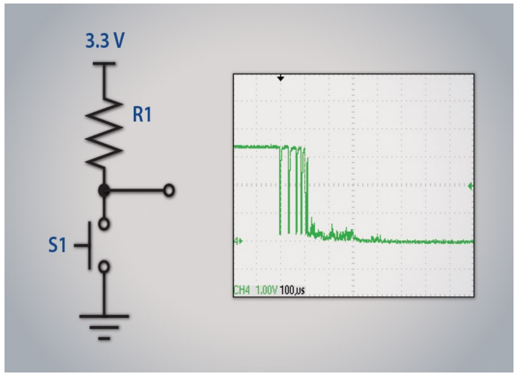
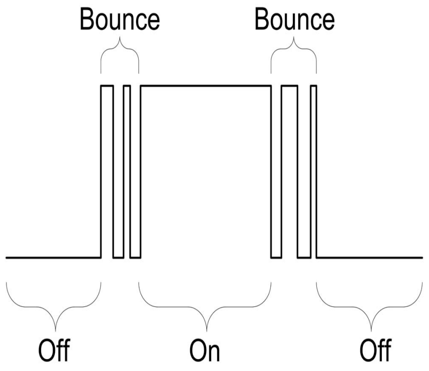


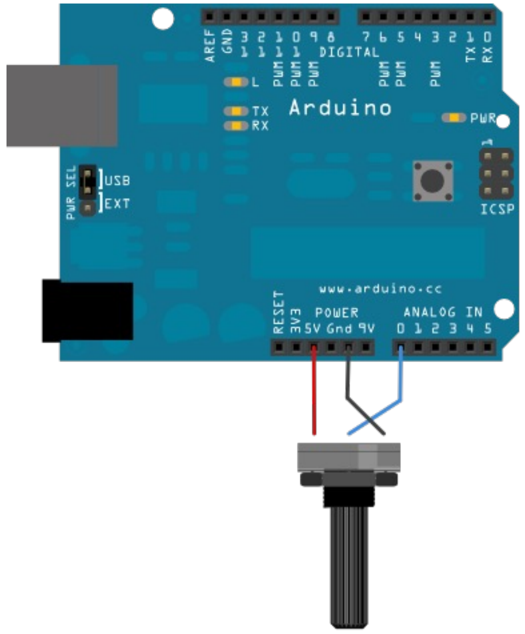
Figure 2: An RC debouncer

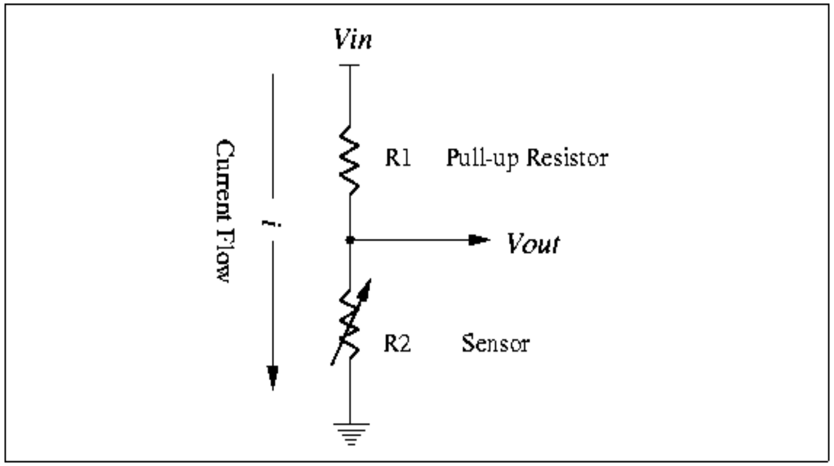


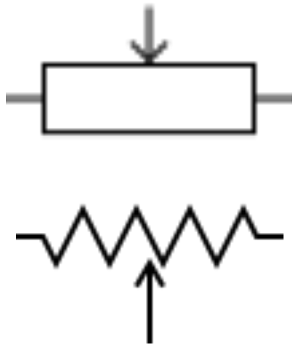


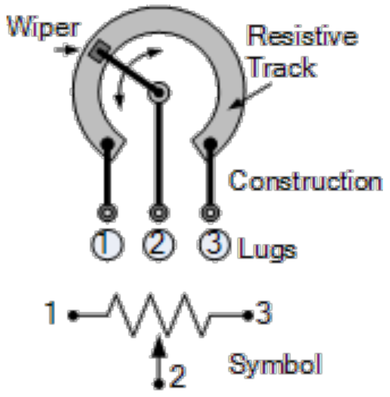
Potentiometer

A **potentiometer**, informally a **pot**, is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider.





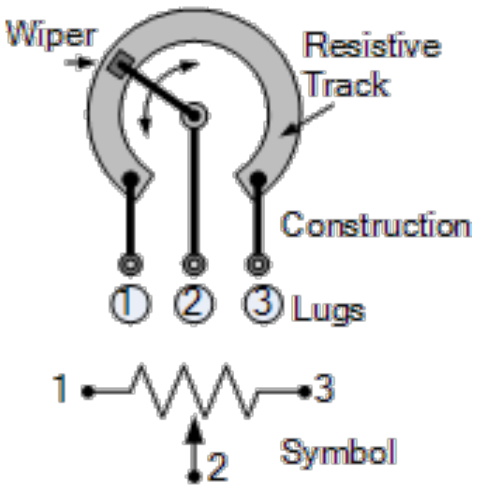


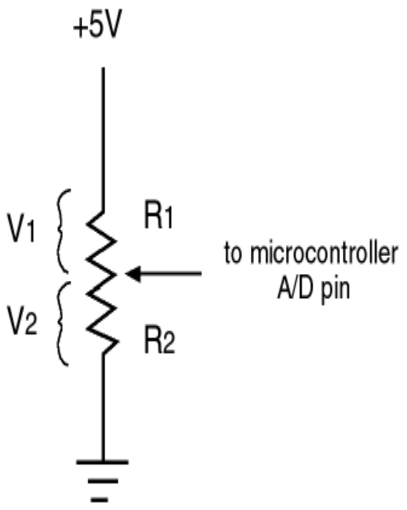


Is this an analogue or a digital sensor?

Potentiometer is a Voltage Divider

- Variable resistance
- Connect the three pins
  - One pin to the supply voltage Vdd
  - The middle pin the **analog high impedance input**
  - One pin to the ground Vss



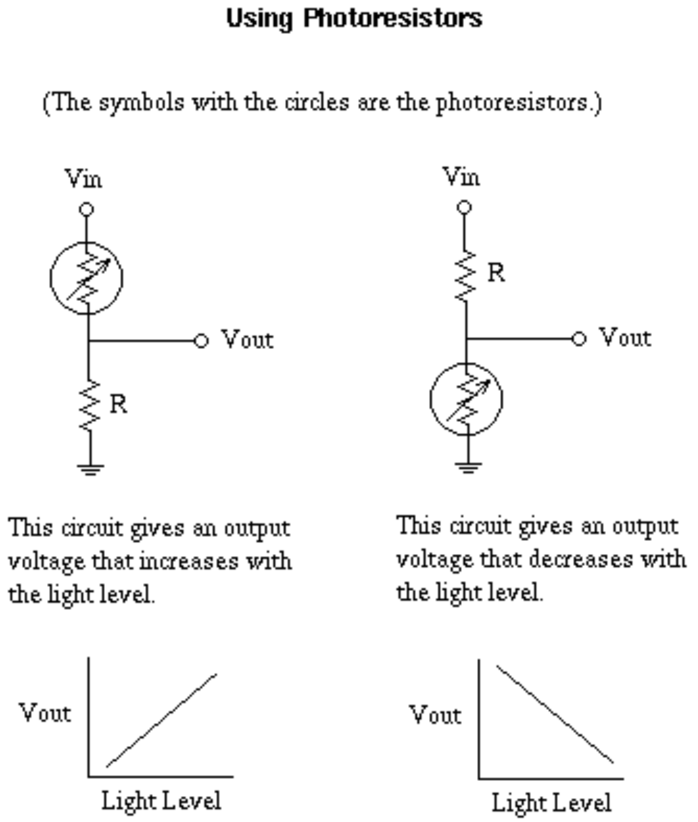


Potentiometer as Voltage Divider

Photoresistor

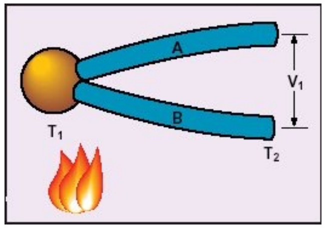
Is a light-controlled variable resistor. The resistance of a Photoresistor decreases with increasing incident light intensity

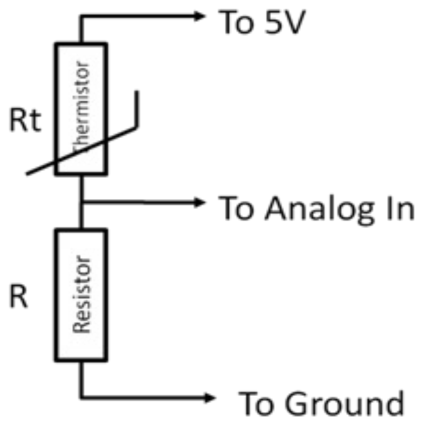
- Photo Resistor
  - 2 pins
  - R inverse proportional with the light

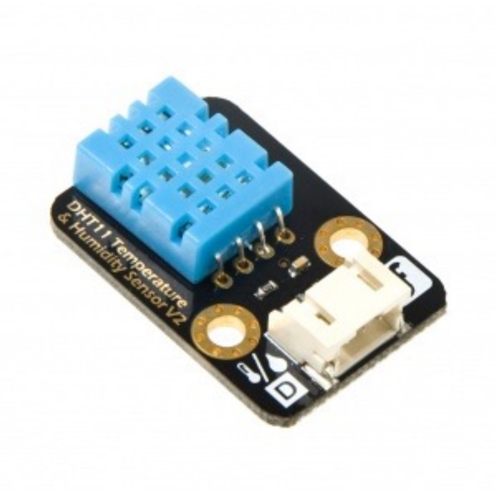

Using Photoresistors
(The symbols with the circles are the photoresistors.)
Vin
Vout
R
This circuit gives an output voltage that increases with the light level.
Vout
Light Level
Vin
R
Vout
This circuit gives an output voltage that decreases with the light level.
Vout
Light Level

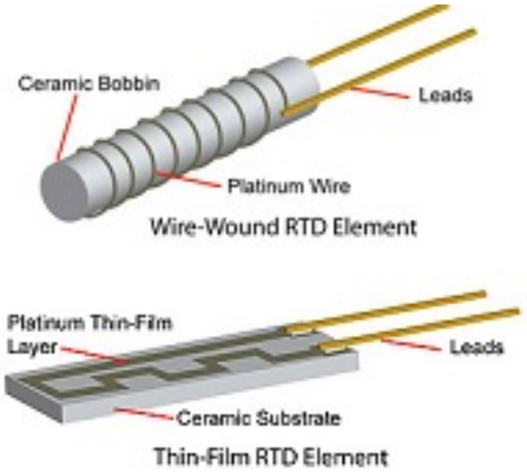
Temperature Sensor

- Measure the change in temperature
- There are few different classes of temp sensors:
  - Thermocouples
  - Resistance temperature detectors
  - Temperature-transducer ICs
  - **Thermistors**





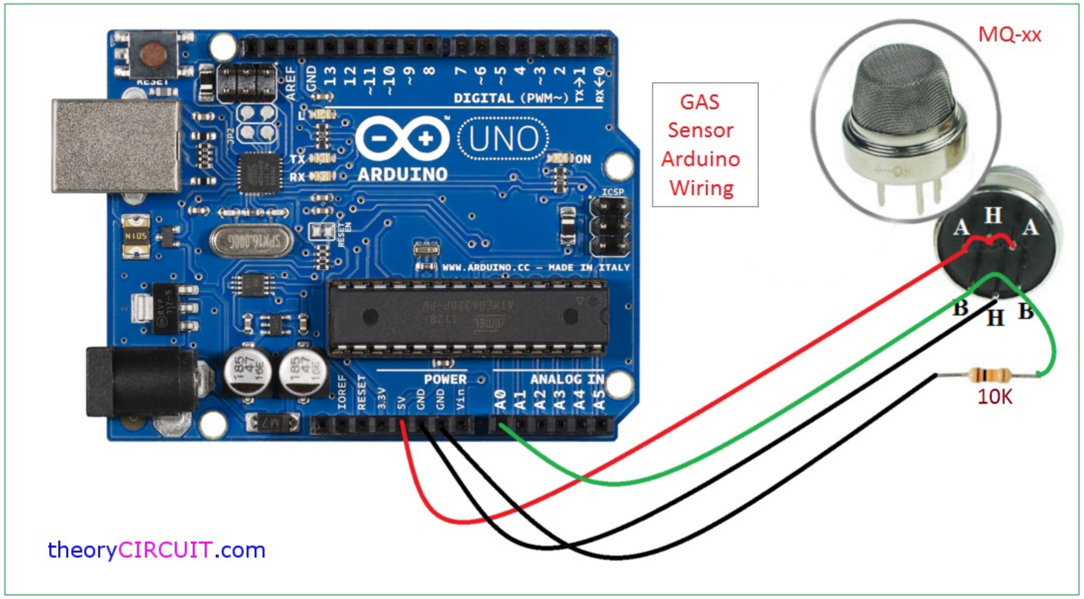


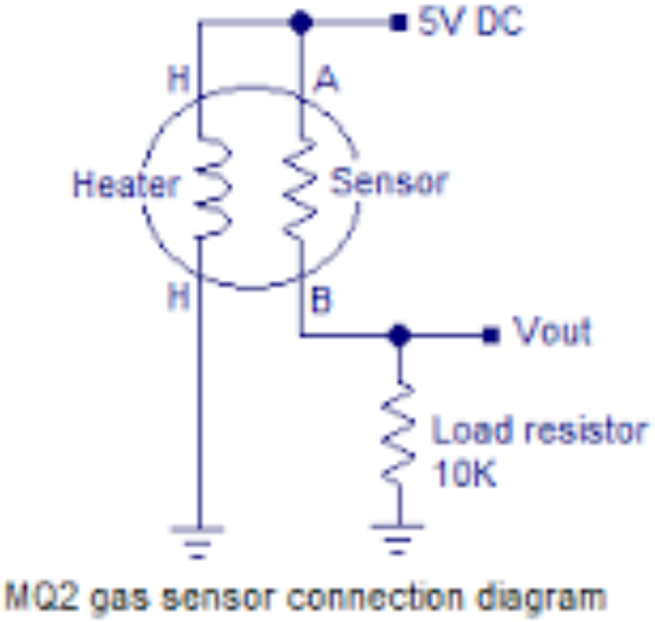


Thermistors are thermally sensitive resistors whose prime function is to exhibit a large, predictable and precise change in electrical resistance when subjected to a corresponding change in body temperature.

Gas Sensor

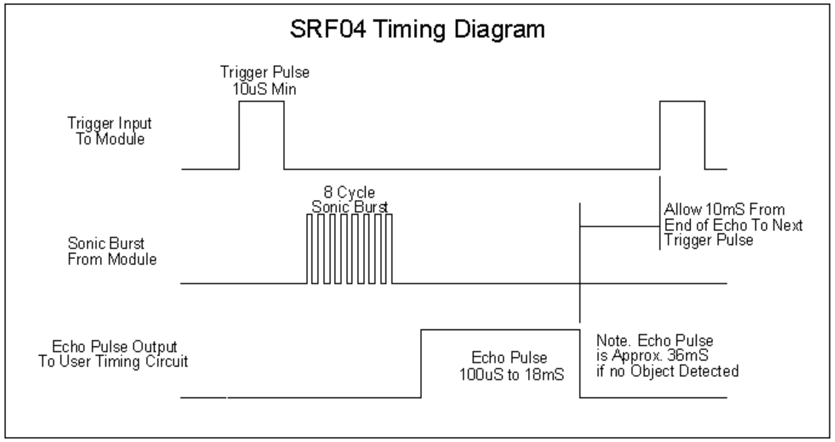
- Gas Sensor
  - Three pins
    - Vcc
    - Signal
    - Ground

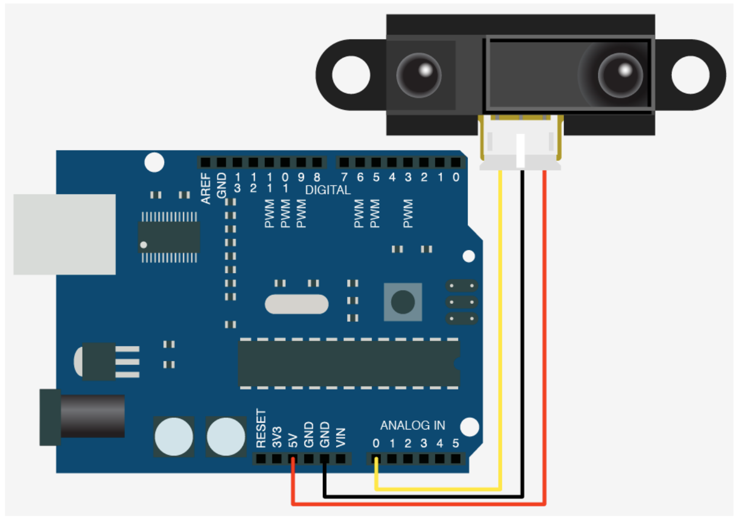


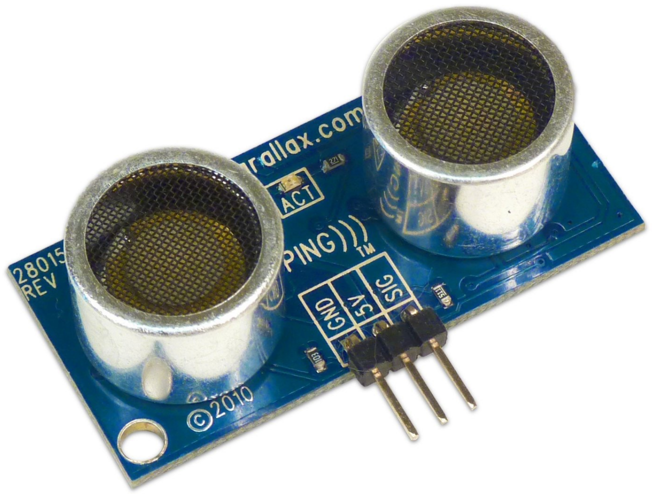


Distance Sensor

- SRF04 sensor
  - Ultrasonic technology
  - Sends a pulse
  - Real time system
  - Works on microcontrollers
- Interested to know how it works:
  - https://www.robot-electronics.co.uk/htm/srf04tech.htm


SRF04 Timing Diagram
Trigger Pulse 10uS Min
Trigger Input To Module
8 Cycle Sonic Burst
Sonic Burst From Module
Allow 10mS From End of Echo To Next Trigger Pulse
Echo Pulse Output To User Timing Circuit
Echo Pulse 100uS to 18mS
Note. Echo Pulse is Approx. 36mS if no Object Detected





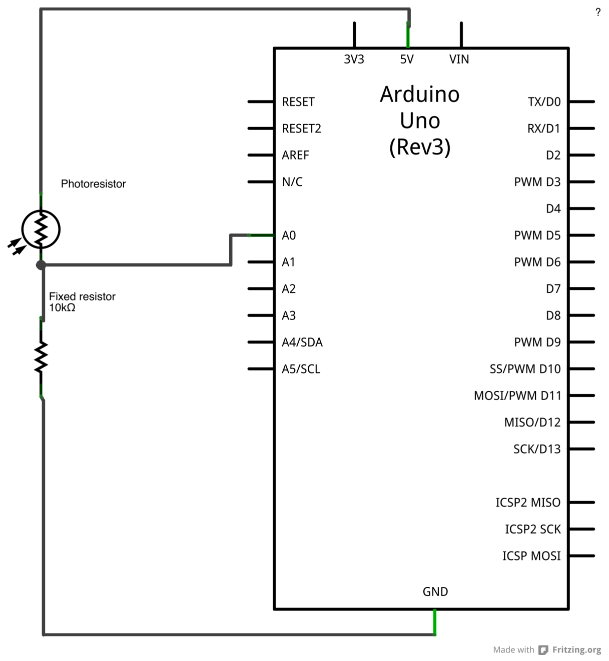
Reading Analogue Input (an example!):

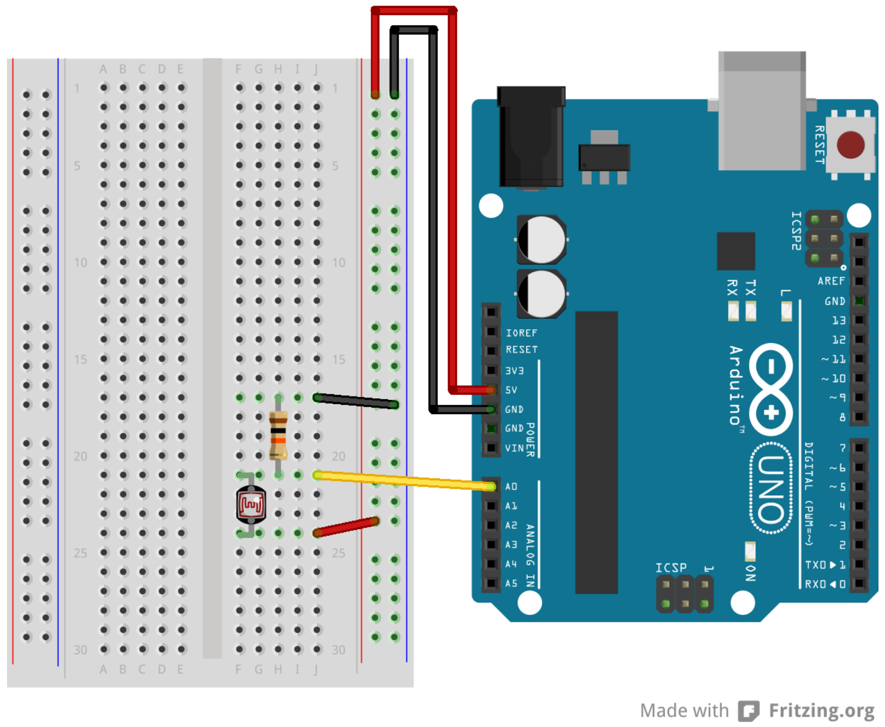
- **Objective**:
  - Use a variable resistor (a potentiometer or a photoresistor), we read its value using one analog input of an Arduino or Genuino board and we change the blink rate of the built-in LED accordingly.

- **What do we need:**
  - Arduino or Genuino Board (our microprocessor)
  - 10K ohm photoresistor and 10K ohm resistor
  - built-in LED on pin 13 (already available on this board!)

Example Continued:

- Build the voltage divider and connect the mid point to the analogue sensor!





$$V_{out} = \frac{R_2}{(R_1+R_2)} \cdot V_{supply} \mid V_{suuply} = 5V$$

Example Continued:

- **Microprocessor commands:**
  - **analogRead():**
    - converts the input voltage range, 0 to 5 volts, to a digital value between 0 and 1023
    - **Which circuit does this conversion?**
  - **delay(variable):**
    - processor busy wait for the duration specified by variable!
  - **digitalWrite():**
    - writes a HIGH (1,5V) or LOW (0,0V) to a digital pin.
  - Digital pin (13) in this board is connected to the on board LED.

Click here to get more details!

Click here to get more details!

Example Continued

- Program the microprocessor!

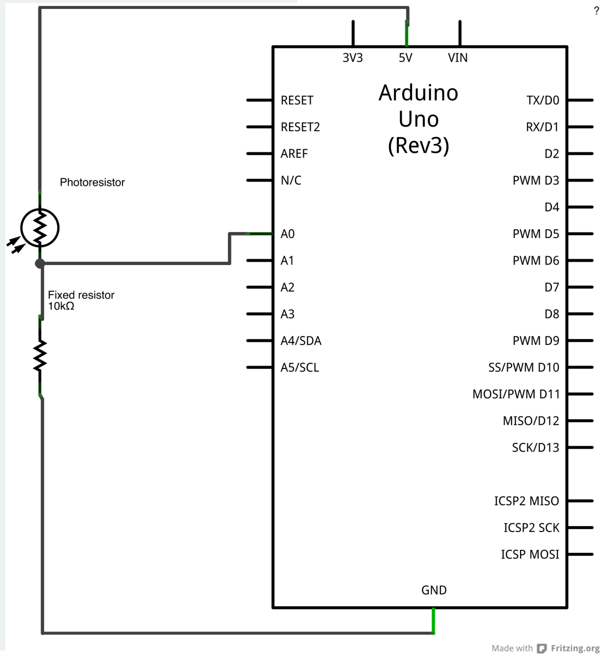
```
  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/AnalogInput
*/

int sensorPin = A0;    // select the input pin for the potentiometer
int ledPin = 13;      // select the pin for the LED
int sensorValue = 0;  // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  // turn the ledPin on
  digitalWrite(ledPin, HIGH);
  // stop the program for <sensorValue> milliseconds:
  delay(sensorValue);
  // turn the ledPin off:
  digitalWrite(ledPin, LOW);
  // stop the program for for <sensorValue> milliseconds:
  delay(sensorValue);
}
```


3V3
5V
VIN
Arduino Uno (Rev3)
RESET
RESET2
AREF
N/C
A0
A1
A2
A3
A4/SDA
A5/SCL
TX/D0
RX/D1
D2
PWM D3
D4
PWM D5
PWM D6
D7
D8
PWM D9
SS/PWM D10
MOSI/PWM D11
MISO/D12
SCK/D13
ICSP2 MISO
ICSP2 SCK
ICSP MOSI
GND
Photoresistor
Fixed resistor 10kΩ
Made with Fritzing.org