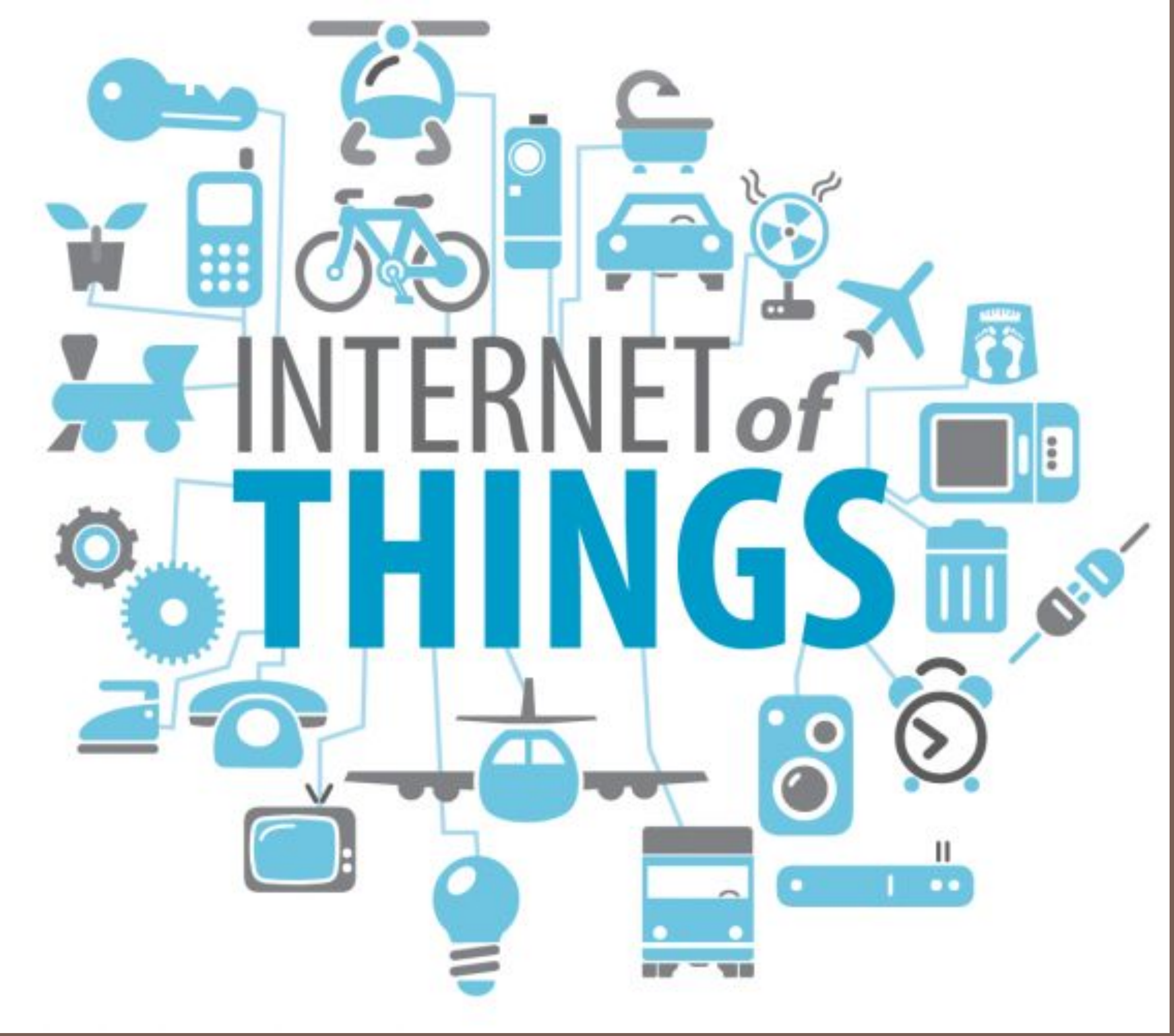


IMPLEMENTING IOE

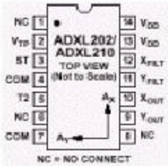
Week 5

Assist. Prof. Rassim Suliyev - SDU 2017

Getting Input from Sensors

- Sensors give report on the world around it
- Sensors convert physical input to an electrical signal
- The electrical signal depends on the kind of sensor and how much information it needs to transmit
  - Some use substance that alters their electrical properties in response to physical change
  - Others are sophisticated electronic modules that use their own microcontroller to process information

Gas Sensor



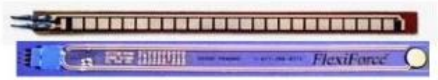
Accelerometer


Gyro


Piezo Bend Sensor

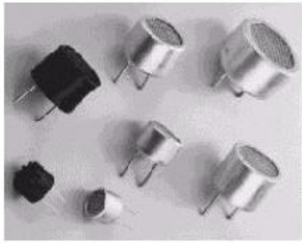

Thyristor


Digital Infrared Ranging



Resistive Bend Sensors


Magnetic Reed Switch


Piezo Ultrasonic Transducers

Methods to provide information

- Digital on/off - switch a voltage on and off
  - Tilt sensor
  - Motion sensor
- Analog - provide an analog signal (voltage)
  - Temperature sensor
  - Light intensity sensor
- Pulse width - measure the duration of a pulse
  - Distance sensors

Methods to provide information

- Serial - provide values using a serial protocol
  - RFID reader
  - GPS module
- Synchronous protocols: I2C and SPI
  - The I2C and SPI digital standards were created for microcontrollers to talk to external sensors and modules
  - These protocols are used extensively for sensors, actuators, and peripherals
  - E.g: compass module, LCD display

Consumer devices

- Contain sensors but are sold as devices in their own right
- Provide sensors already incorporated into robust and ergonomic devices
- They are also inexpensive as they are mass-produced
  - PS2 mouse
  - PlayStation game controller

Data sheets

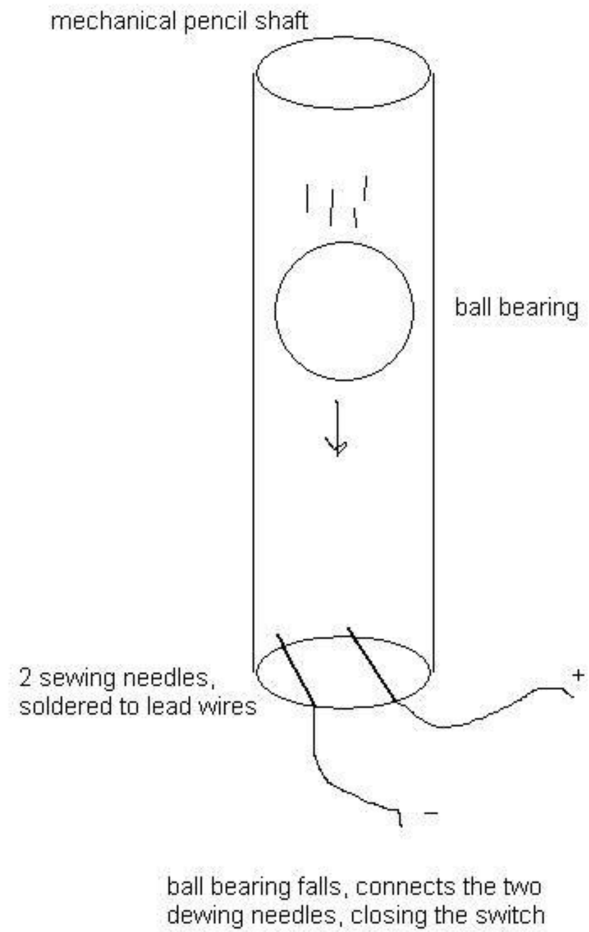
- Contains information about a sensor’s output signal
- Available from the company from which you bought the device
- Google search of the device part number or description
- Are aimed at engineers designing products to be manufactured
- Usually provide more detail than you need
- Information on output signal will usually be in a section:
  - Data format, interface, output signal, or something similar
- Check the maximum voltage!!!

Noises

- Reading sensors from the messy analog world is a mixture of science, art, and perseverance
- Use trial and error method to get a successful result
- Common problem:
  - Sensor just tells you a physical condition has occurred
  - But not what caused it
- Skills to acquire with experience:
  - Putting the sensor in the right context
  - Location, range, orientation
  - Limiting its exposure to things that you don't want to activate it
- Separating the desired signal from background noise
  - Use a threshold to detect when a signal is above a certain level
  - Take the average of a number of readings to smooth out noise spikes

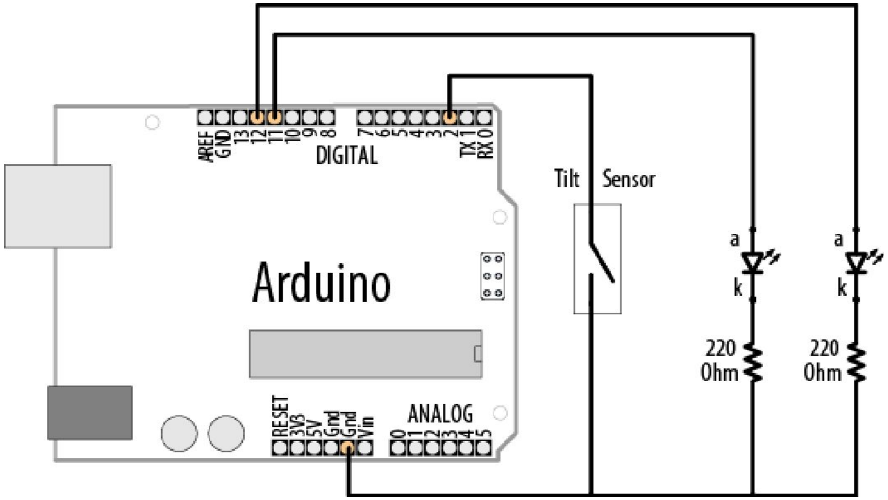
Detecting Movement

- Detect when something is moved, tilted, or shaken
- Tilt sensor - switch that closes a circuit when tilted
- Ball bearing in a box with contacts at one end
- Sensitive to small movements of around 5 to 10 degrees



Detecting Movement

```
const int tiltSensorPin = 2; //pin the tilt sensor is connected to
const int firstLEDPin = 13; //pin for one LED
const int secondLEDPin = 12; //pin for the other
void setup(){
  pinMode (tiltSensorPin, INPUT); //the code will read this pin
  digitalWrite (tiltSensorPin, HIGH); // and use a pull-up resistor
  pinMode (firstLEDPin, OUTPUT); //the code will control this pin
  pinMode (secondLEDPin, OUTPUT); //and this one
}
void loop(){
  if (digitalRead(tiltSensorPin)){ //check if the pin is high
    digitalWrite(firstLEDPin, HIGH); //if it is high turn on firstLED
    digitalWrite(secondLEDPin, LOW); //and turn off secondLED
  } else{ //if it isn't do the opposite
    digitalWrite(firstLEDPin, LOW);
    digitalWrite(secondLEDPin, HIGH);
  }
}
```



Shake detection

```
const int tiltSensorPin = 2;
const int ledPin = 13;
int tiltSensorPreviousValue = 0;
int tiltSensorCurrentValue = 0;
long lastTimeMoved = 0;
int shakeTime = 100;
void setup(){
  pinMode (tiltSensorPin, INPUT);
  digitalWrite (tiltSensorPin, HIGH);
  pinMode (ledPin, OUTPUT);
}
void loop(){
  tiltSensorCurrentValue=digitalRead(tiltSensorPin);
  if (tiltSensorPreviousValue != tiltSensorCurrentValue){
    lastTimeMoved = millis();
    tiltSensorPreviousValue = tiltSensorCurrentValue;
  }
  if (millis() - lastTimeMoved < shakeTime){
    digitalWrite(ledPin, HIGH);
  } else{
    digitalWrite(ledPin, LOW);
  }
}
```

millis() function

- Returns long type value - number of milliseconds since the current sketch started running
- Will overflow (go back to zero) in approximately 50 days
- Determine the duration of the event by subtracting the pre-stored start time from the current time

```
long startTime = millis();
…do something…
long duration = millis() - startTime;
```

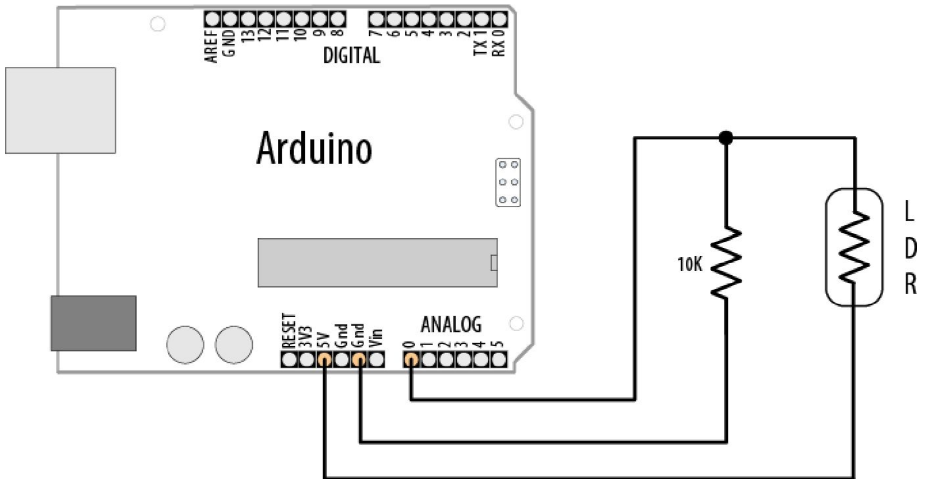
Similar digital output sensors

- Mechanical switch sensors can be used in similar ways
- Float switch can turn on when the water level in a container rises to a certain level
- The way a ball cock works in a toilet cistern
- A pressure pad can be used to detect when someone stands on it

Detecting Light

- Detect changes in light levels
  - Something passes in front of a light detector
  - Detecting when a room is getting too dark
- Use a light dependent resistor (LDR)
  - Changes resistance with changing light levels
  - Produces a change in voltage

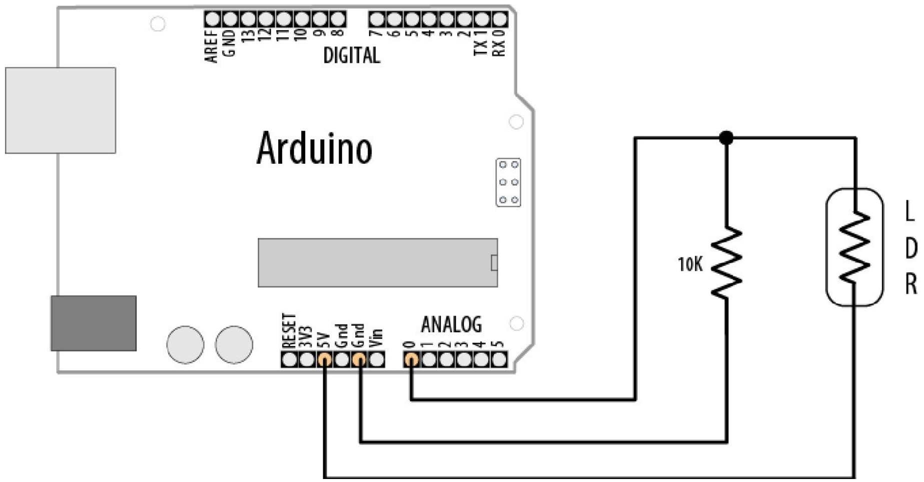
This circuit is the standard way to use any sensor that changes its resistance based on some physical phenomenon



Detecting Light

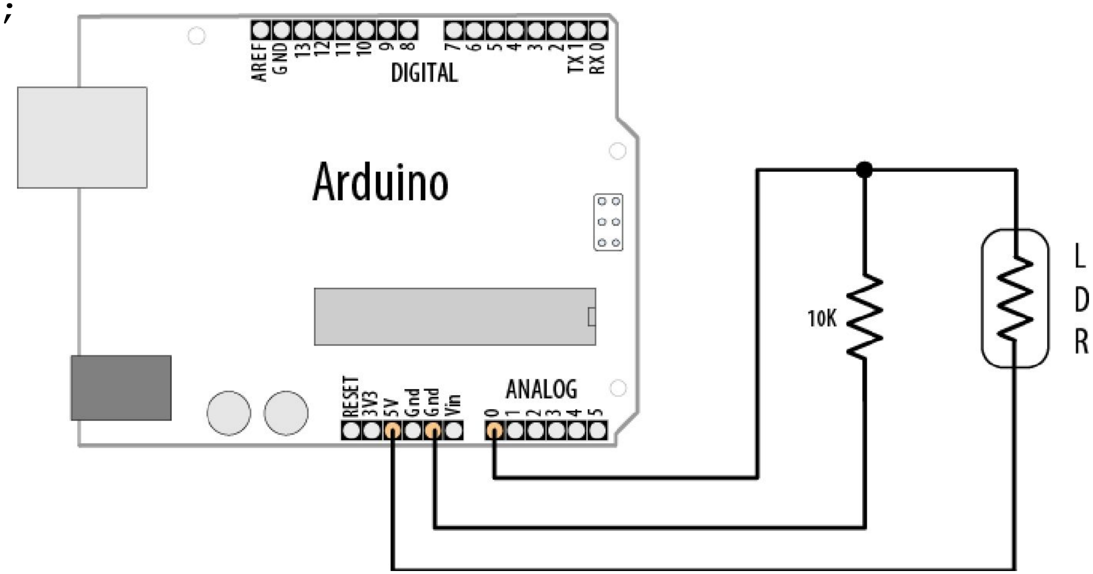
- Not full range of possible values (0-1023)
  - Since voltage will not be swinging between 0-5 V
- LDR - simple kind of sensor called a resistive sensor
  - Range of resistive sensors respond to changes in different physical characteristics

It is important to check the actual values the device returns in the situation you will be using it. Then you have to determine how to convert them to the values you need.



When a room is getting dark

```
const int ldrPin = A0;
const int ledPin = 13;
const int darknessThreshold = 500;
void setup(){
  pinMode (ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop(){
  if(analogRead(ldrPin) < darknessThreshold){
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

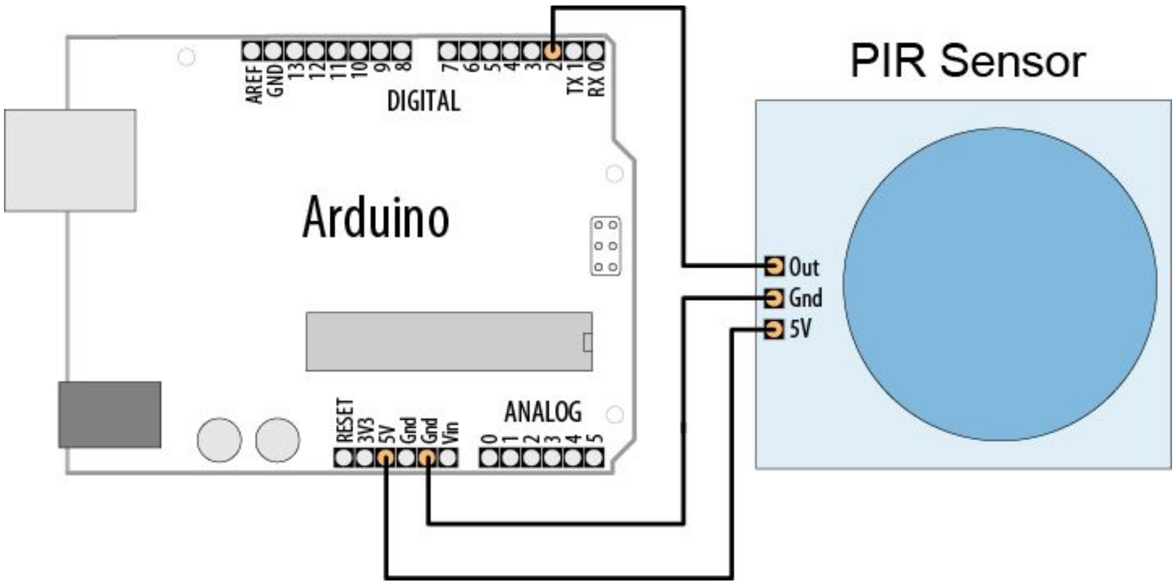


Detecting motion

```
const int ldrPin = A0;
const int ledPin = 13;
const int movementThreshold = 20;
int previousReading = 0;
int currentReading = 0;
void setup(){
  pinMode (ledPin, OUTPUT);
  Serial.begin(9600);
  previousReading = analogRead(ldrPin);
}
void loop(){
  currentReading = analogRead(ldrPin);
  if(abs(currentReading - previousReading) >
movementThreshold){
    previousReading = currentReading;
    digitalWrite(ledPin, HIGH);
    delay(1000);
    digitalWrite(ledPin, LOW);
  }
}
```

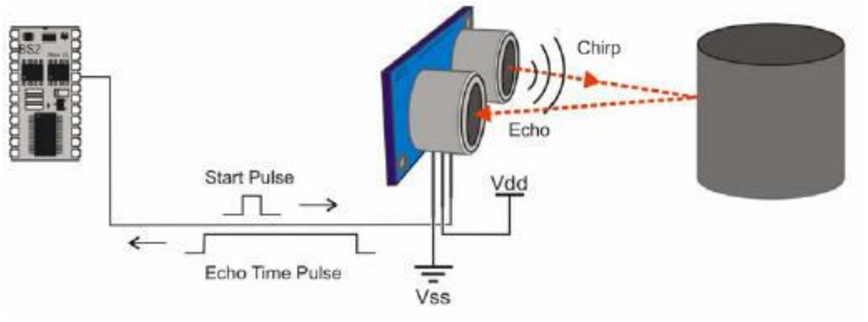
PIR motion detection

- PIR - passive infrared sensor
  - measures infrared light radiating from objects
- Made from pyroelectric materials
  - which generate energy when exposed to heat



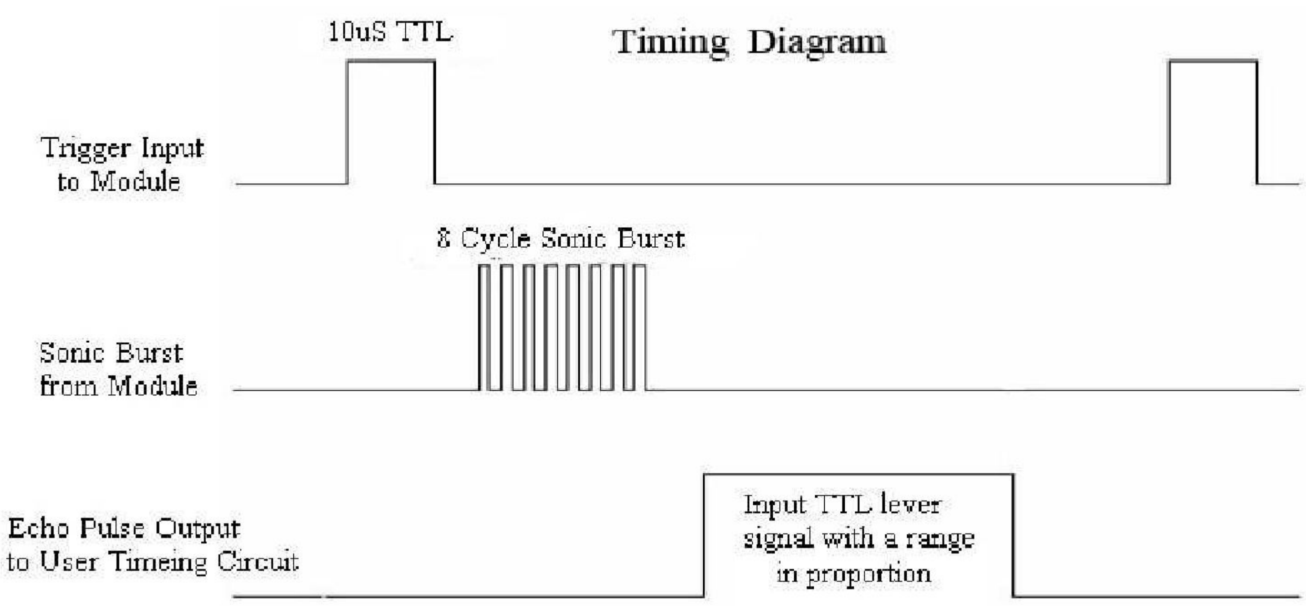
Measuring Distance

- Ultrasonic distance sensor
  - Measure the distance of an object (2 cm - 3 m)
  - Modules includes ultrasonic transmitters, receiver and control circuit
- Basic principle of work:
  1. Trigger trig pin for at least 10us with a high level signal
  2. The module automatically sends eight 40 kHz and detect whether there is a pulse signal back
  3. Time of high output duration at echo pin is the time from sending ultrasonic to returning.



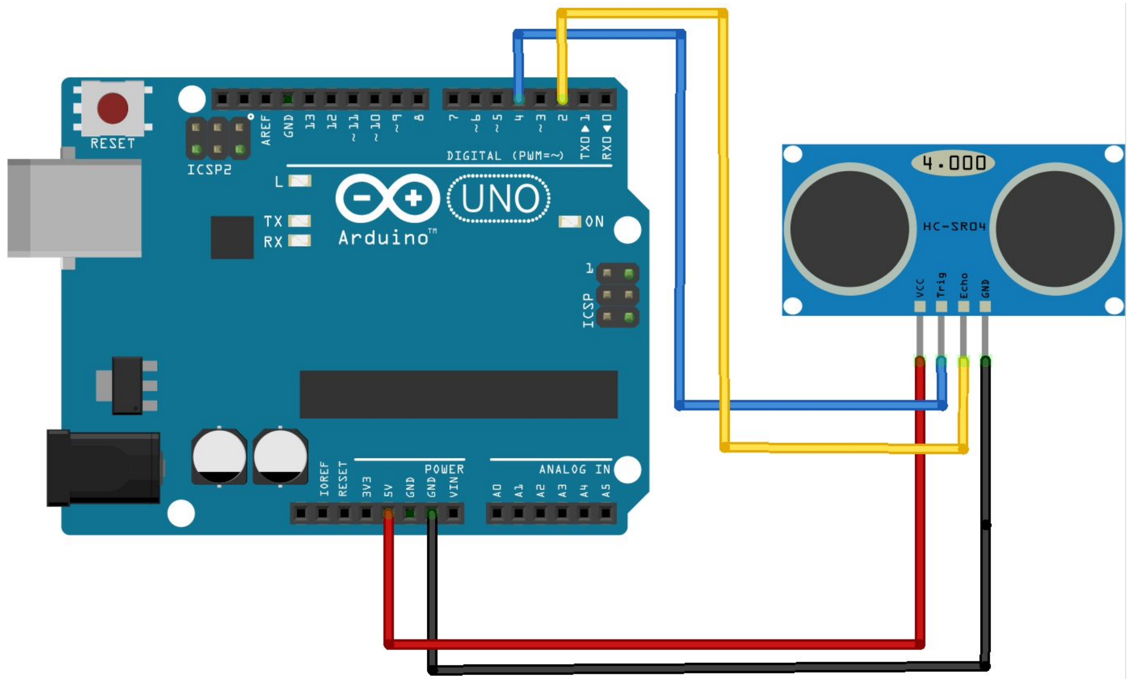
How it works

- Ultrasonic sensors provide a measurement of the time it takes for sound to bounce off an object and return to the sensor
- Pulse width is proportional to the distance the sound traveled
- The speed of sound is 340 meters per second - 29 microseconds per centimeter
- Roundtrip = microseconds / 29
- Distance in centimeters is: microseconds / 29 / 2



Measuring Distance

```
const int trigPin = 4;
const int echoPin = 2;
const int ledPin = 13;
long value = 0;
int cm = 0;
void setup(){
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  digitalWrite(trigPin, LOW);
}
void loop(){
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  value = pulseIn(echoPin, HIGH, 50000);
  cm = value / 58; // pulse width is 58 microseconds per cm
  Serial.print(value); Serial.print(" , "); Serial.println(cm);
  digitalWrite(ledPin, HIGH);
  delay(cm * 10 ); // each centimeter adds 10 milliseconds delay
  digitalWrite(ledPin, LOW);
  delay( cm * 10); delay(20);
}
```

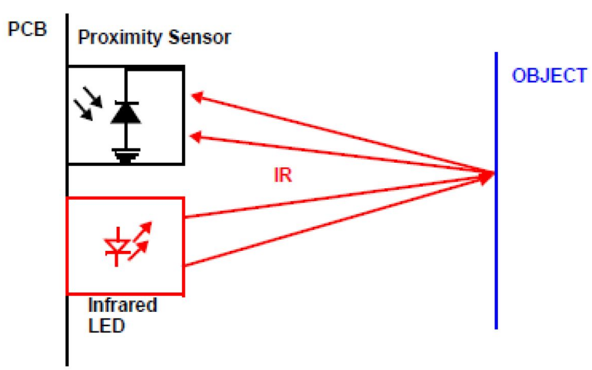


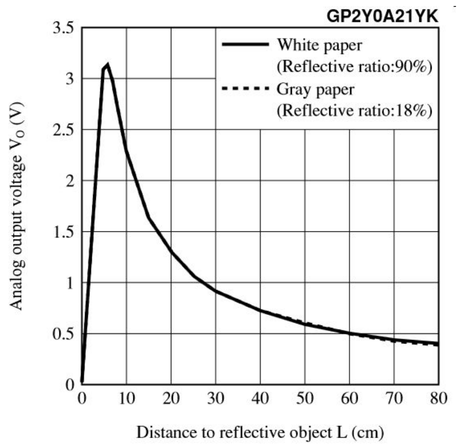
pulseIn() and delayMicroseconds()

- pulseIn(pin, value[, timeout])
  - Reads a pulse (either HIGH or LOW) on a pin
  - pin: the number of the pin on which to read the pulse
  - value: type of pulse to read: either HIGH or LOW
  - timeout (optional): the number of microseconds to wait for the pulse to be completed. Default is one second
  - Returns the length of the pulse in microseconds or 0 if no complete pulse was received within the timeout
- delayMicroseconds(us)
  - Pauses the program for the us amount of time (in microseconds) specified as parameter

IR distance rangers

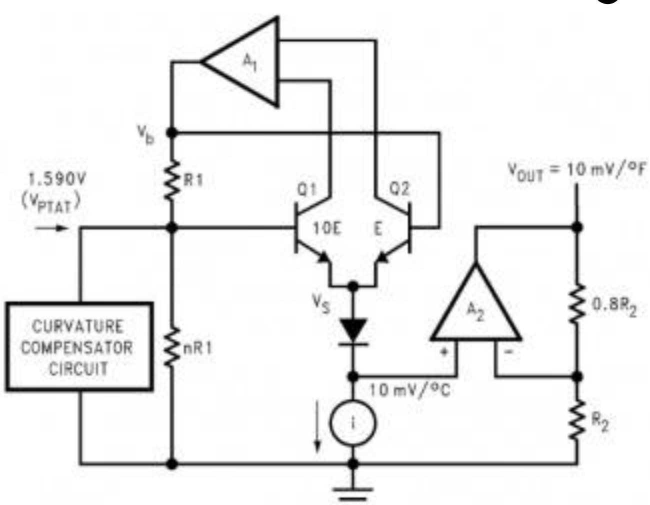
- Generally provide an analog output
- Have greater accuracy than ultrasonic sensors
- Range of 10 cm to 1 m or 2 m
- Output from the IR sensor is not linear (not proportional to distance)
- Distance values can be found by trial and error





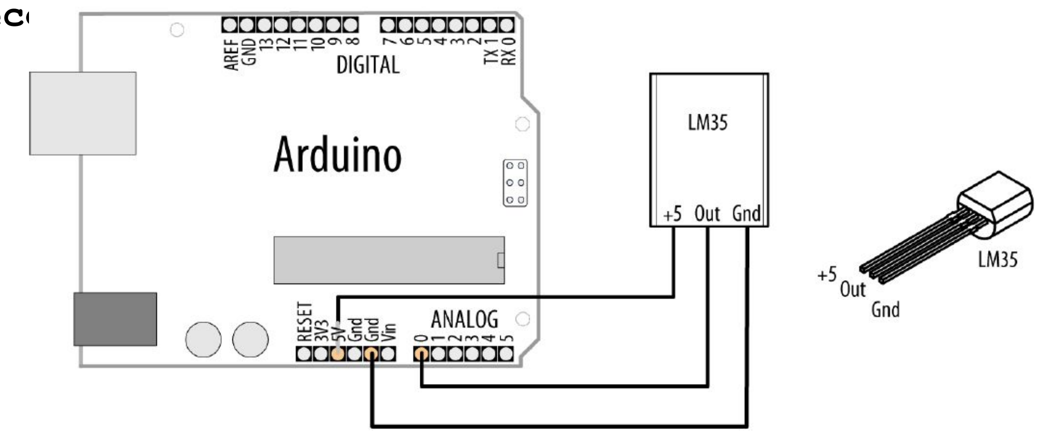
Measuring Temperature

- LM35 heat detection sensor
  - Produces an analog voltage directly proportional to temperature
  - 1 millivolt per 0.1°C (10mV per degree)
  - The sensor accuracy is around 0.5°C,
  - Functional range: from -40°C to 150°C



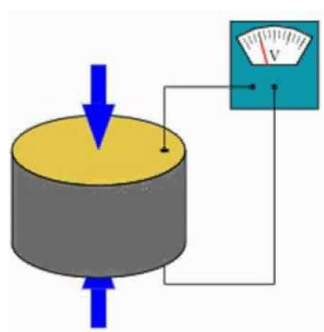
Measuring Temperature

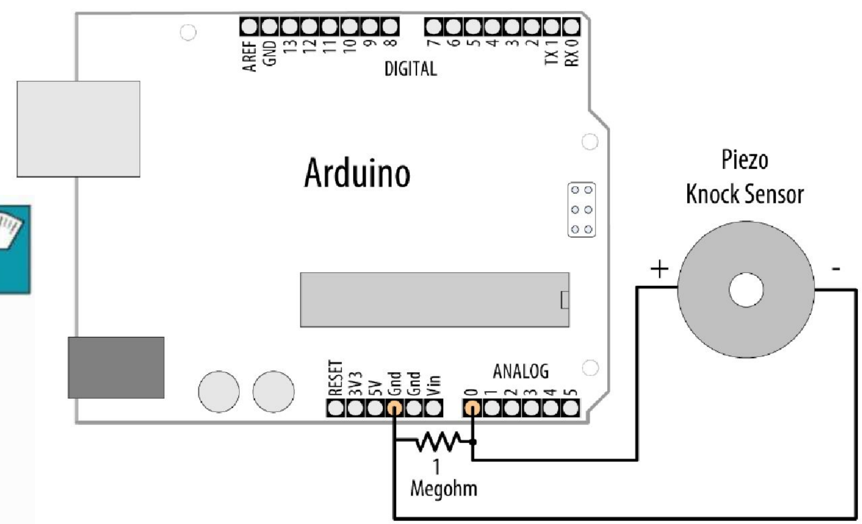
```
const int inPin = A0; // analog pin
void setup() {
  Serial.begin(9600);
}
void loop(){
  int value = analogRead(inPin);
  Serial.print(value); Serial.print(" => ");
  float millivolts = (value / 1024.0) * 5000;
  float celsius = millivolts / 10; // sensor output is 10mV per degree
Celsius
  Serial.print(celsius);
  Serial.print(" degrees Celsius, ");
  Serial.print( (celsius * 9)/ 5 + 32 ); // converts to fahrenheit
  Serial.println(" degrees Fahrenheit");
  delay(1000); // wait for one sec
}
```



Detecting Vibration

- Piezo sensor responds to vibration
- Produces a voltage in response to physical stress
- The more it is stressed, the higher the voltage
- Piezo is polarized (has + and -)
- A high-value resistor (1 megohm) is connected across the sensor





Detecting Vibration

```
const int sensorPin = A0; // the analog pin connected to the
sensor
const int ledPin = 13; // pin connected to LED
const int THRESHOLD = 1;
void setup(){
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop(){
  int val = analogRead(sensorPin);
  Serial.println(val);
  if (val >= THRESHOLD){
    digitalWrite(ledPin, HIGH);
    delay(100);
  } else{
    digitalWrite(ledPin, LOW);
  }
}
```