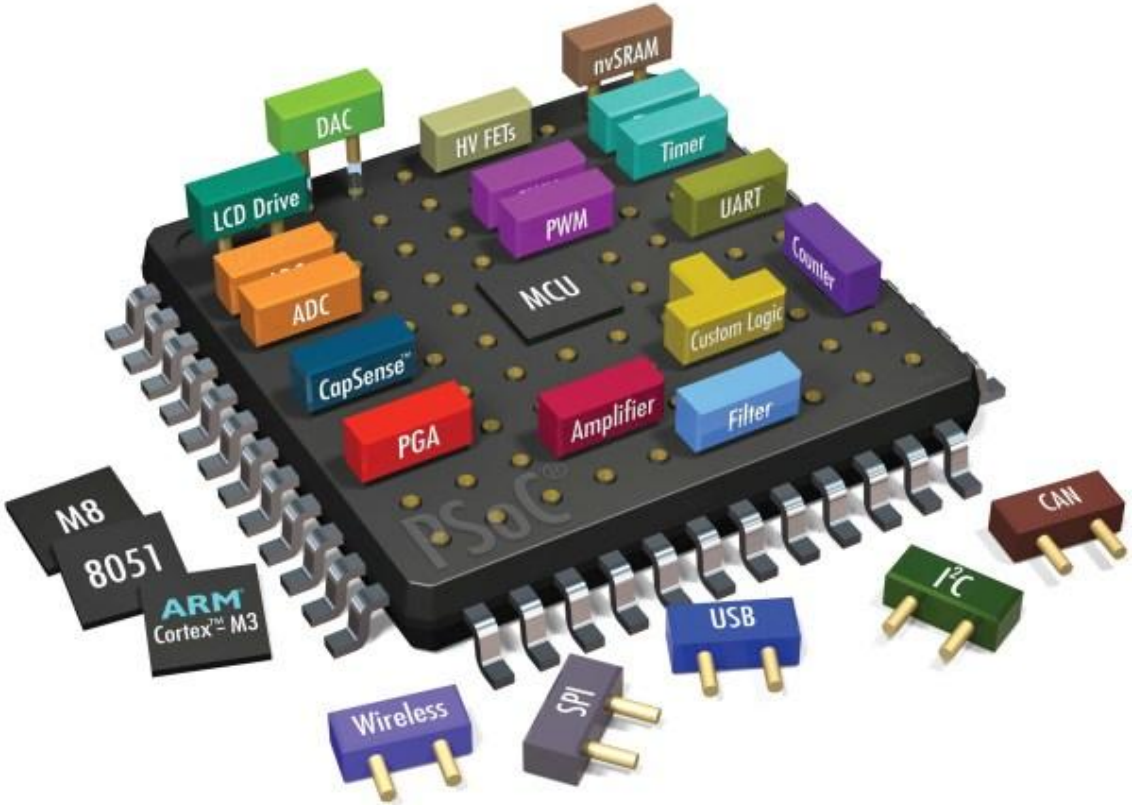
EMBEDDED SYSTEMS & IOT

Techmaniacs Eduservices LLP

What is Embedded Systems

- An **embedded system** is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints.
- It is *embedded* as part of a complete device often including hardware and mechanical parts.
- typical embedded computers have low power consumption, small size, rugged operating ranges, and low per-unit cost.

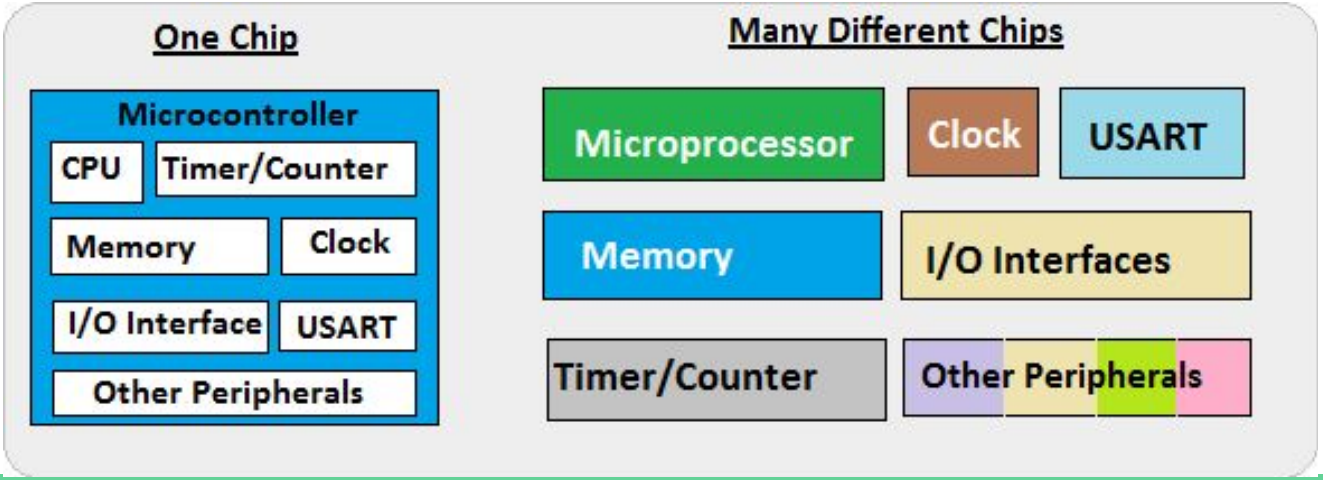
Basic Microcontroller



Basic Features

- **User interface**
  - Embedded systems range from no user interface at all, in systems dedicated only to one task, to complex graphical user interfaces that resemble modern computer desktop operating systems.
  - Some systems provide user interface remotely with the help of a serial (e.g. RS-232, USB, I^2C, etc.) or network (e.g. Ethernet) connection.

- **Processors in embedded systems**
  - Embedded processors can be broken into two broad categories.
  - microprocessors (μP) use separate integrated circuits for memory and peripherals.
  - Microcontrollers (μC) have on-chip peripherals, thus reducing power consumption, size and cost.
  - Both Von Neumann as well as various degrees of Harvard architectures are used.
  - RISC as well as non-RISC processors are found.
  - Word lengths vary from 4-bit to 64-bits and beyond



- **Reliability**

  Embedded systems often reside in machines that are expected to run continuously for years without errors, and in some cases recover by themselves if an error occurs.

  - The system cannot safely be shut down for repair, or it is too inaccessible to repair. Examples include space systems, undersea cables, navigational beacons.
  - The system must be kept running for safety reasons. "Limp modes" are less tolerable. Often backups are selected by an operator. Examples include aircraft navigation, reactor control systems, safety-critical chemical factory controls, train signals.
  - The system will lose large amounts of money when shut down: Telephone switches, factory controls, bridge and elevator controls, funds transfer and market making, automated sales and service

Introduction to Robotics and Robots

- Robot used in English describes any construct that automates some behavior. For example, a garage door opener automates the behavior of opening a door.
- Robotics can be defined as the science or study of the technology primarily associated with the design, fabrication, theory, and application of robots.

Types of Robots

- Machine Pet
  - A machine, capable of moving in some way, that can sense its surroundings and can act on what it senses autonomously. Most of these robots have no real useful purpose, other than to entertain and challenge.
- Autonomous Machine
  - A machine with sensors and actuators that can do some sort of work "on its own". Most industrial and commercial robots fall in this category.

What is Arduino

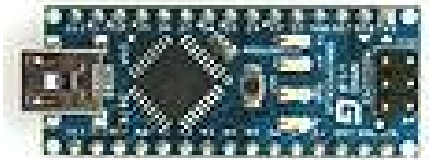
- Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world.
- Arduino board designs use a variety of microprocessors and controllers.
- Easy to use design, no extra components needed.

Some Boards

UNO R3

Mega

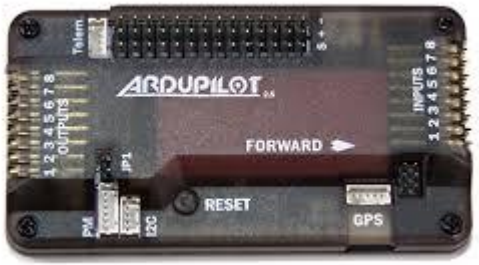
UNO R3 SMD



Nano

Applications

Arduboy



Ardupilot

Gameduino

Specification of Arduino UNO

https://store.arduino.cc/usa/arduino-uno-rev3

Arduino IDE

- The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java.
- A program written with the IDE for Arduino is called a *sketch*. Sketches are saved on the development computer as text files with the file extension *.ino*.
- The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Program structure

A minimal Arduino C/C++ program consist of only two functions:

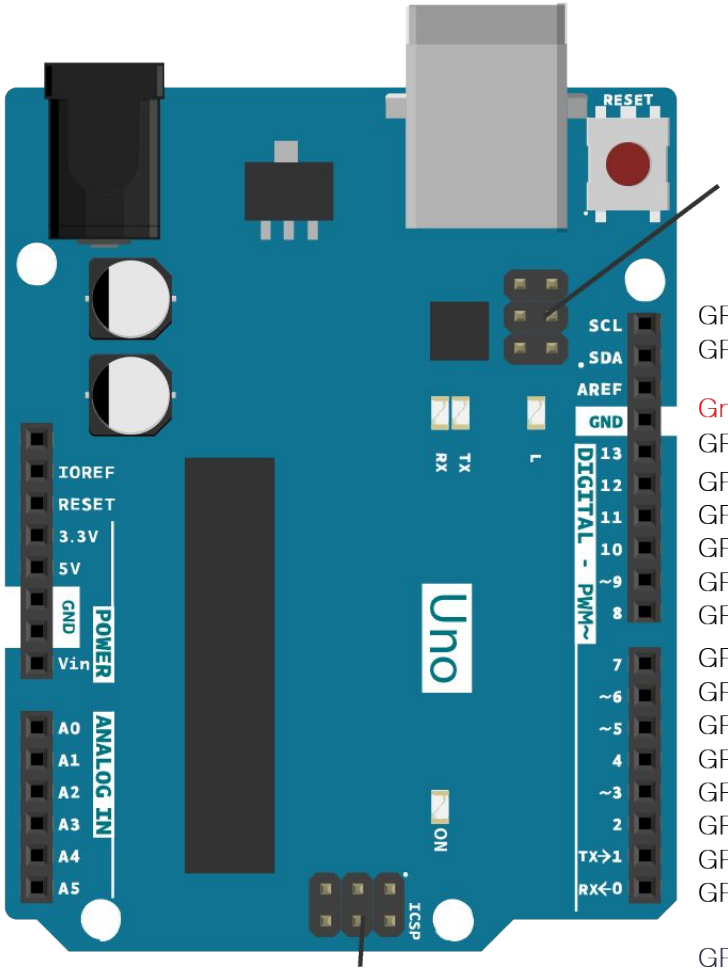
- *setup()*: This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.
- *loop()*: After *setup()* has been called, function *loop()* is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

IORef: 5V

Vin: 7-12V DC max.

Serial: Serial is attached to pins 0 and 1, and to the USB-Serial micrcontroller on board.

The Uno has a second microcontroller on board to handle USB-to-serial communications.
This is the ICSP header for that microcontroller.



		GPIO18	ADC4	SDA		
		GPIO19	ADC5	SCL		
			AREF			
		Gnd				
IORef		GPIO13		SCK		LED
Reset		GPIO12		MISO		
+3.3V		GPIO11		MOSI	PWM11	
+5V		GPIO10		CS	PWM10	
Gnd		GPIO9			PWM9	
Gnd		GPIO8				
Vin		GPIO7				
	ADC0	GPIO14			PWM6	
	ADC1	GPIO15			PWM5	
	ADC2	GPIO16				
	ADC3	GPIO17			PWM3	INT1
SDA	ADC4	GPIO18				INT0
SCL	ADC5	GPIO19		TX		
		GPIO0		RX		
Comm.	ADC	GPIO		GPIO	ADC	Comm. PWM Interrupts

ICSP:

Reset	SCK	MISO
Gnd	MOSI	+5V

Fun with LEDs

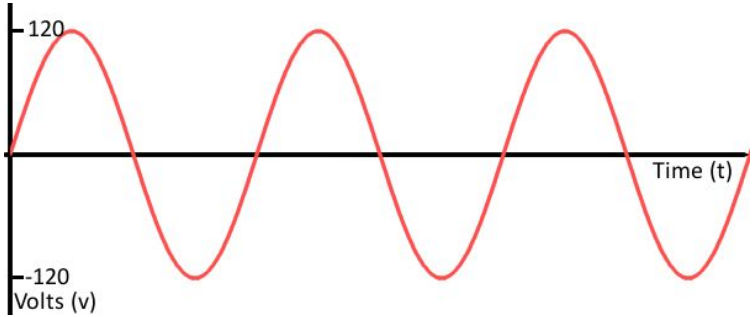
- Linear Pattern, one after another.
- Rising
- Binary counter
- Accumulation
- SHM oscillations

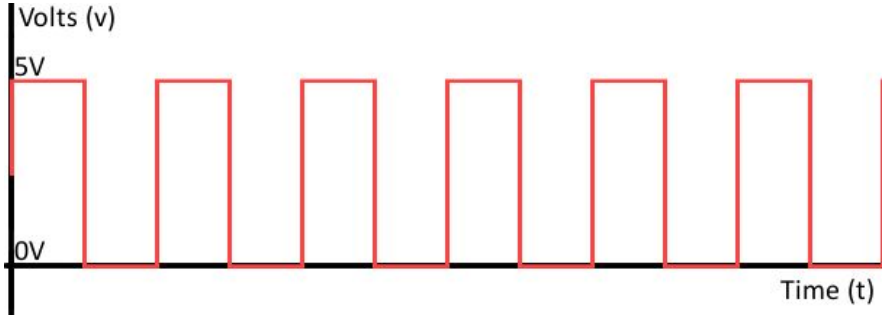
Sensors

- Analog Sensors-

  Sensors which give output in varying voltage levels are known as analog sensors
- Digital Sensors-

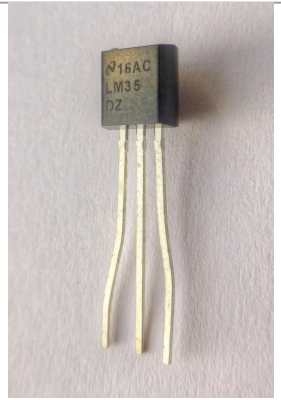
  Sensors whose output voltage varies between certain fixed voltage levels are known as Digital Sensors

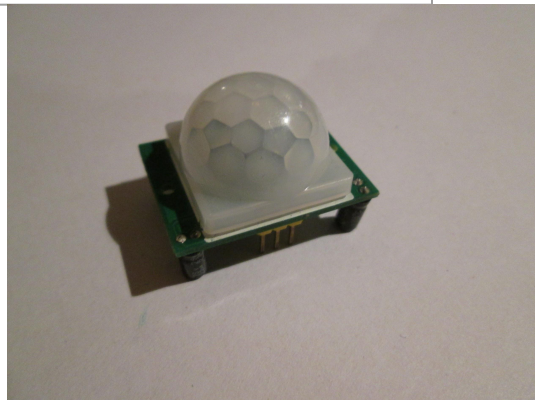




List of Sensors

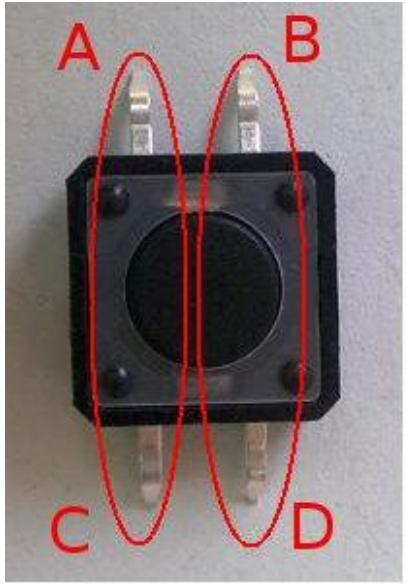
Analog Sensors	Digital Sensors
• Potentiometer • LM35	• Push Button • IR • LDR • Ultrasonic • PIR

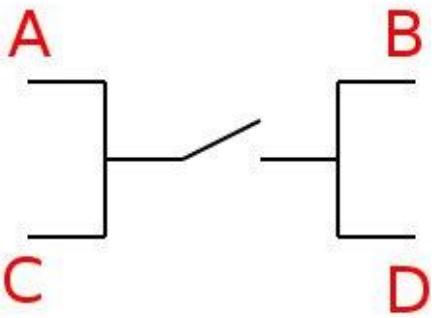




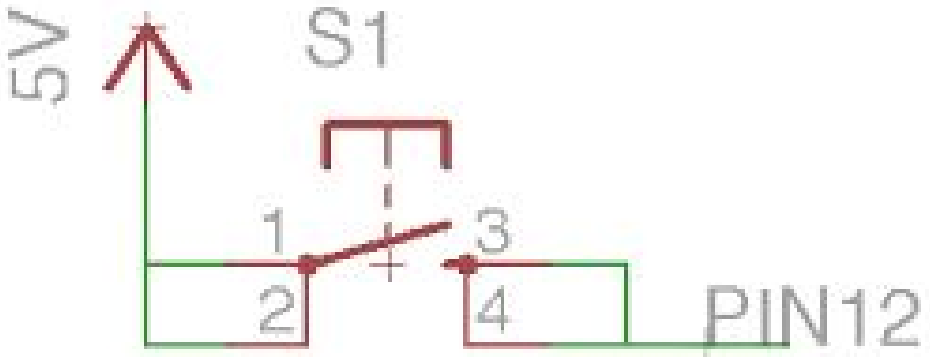
The Push Button

- Basic structure

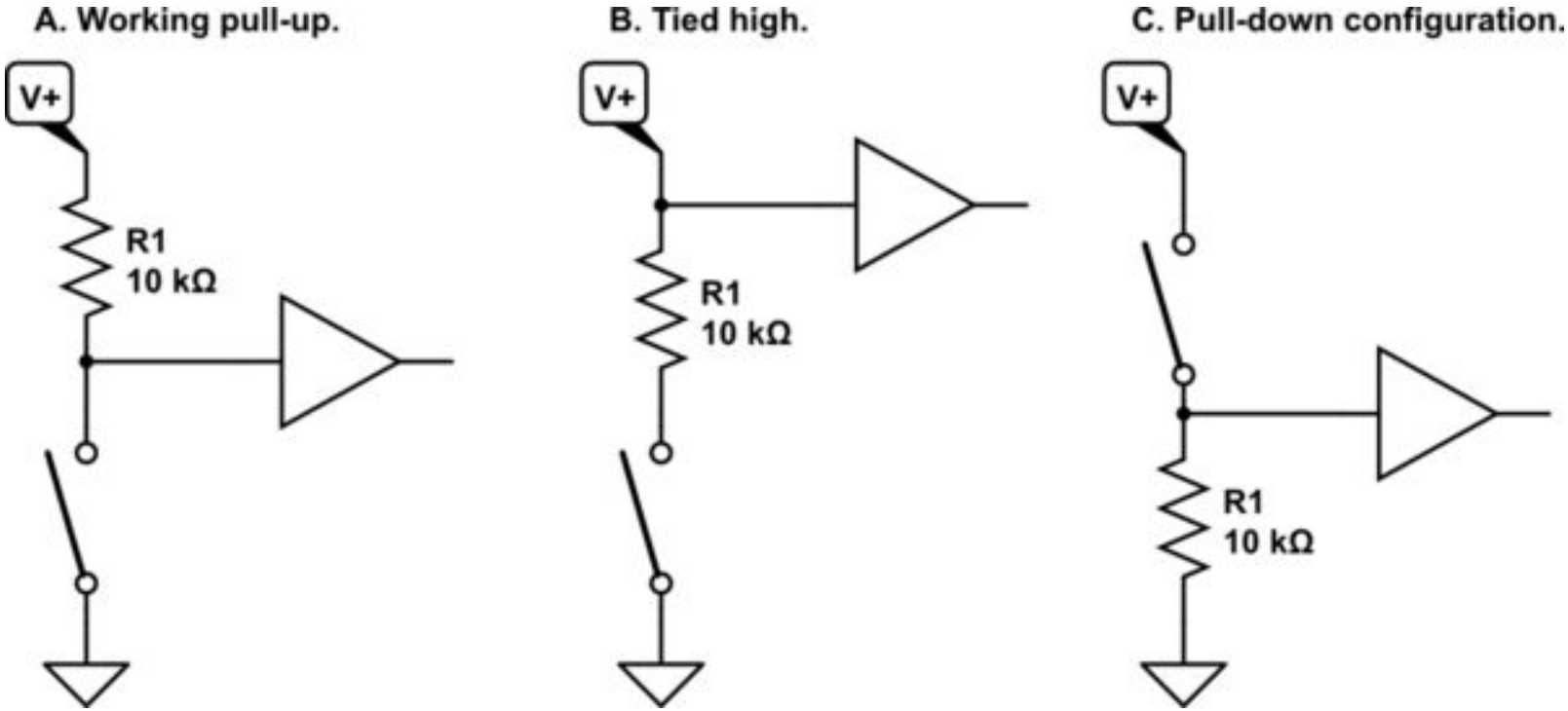




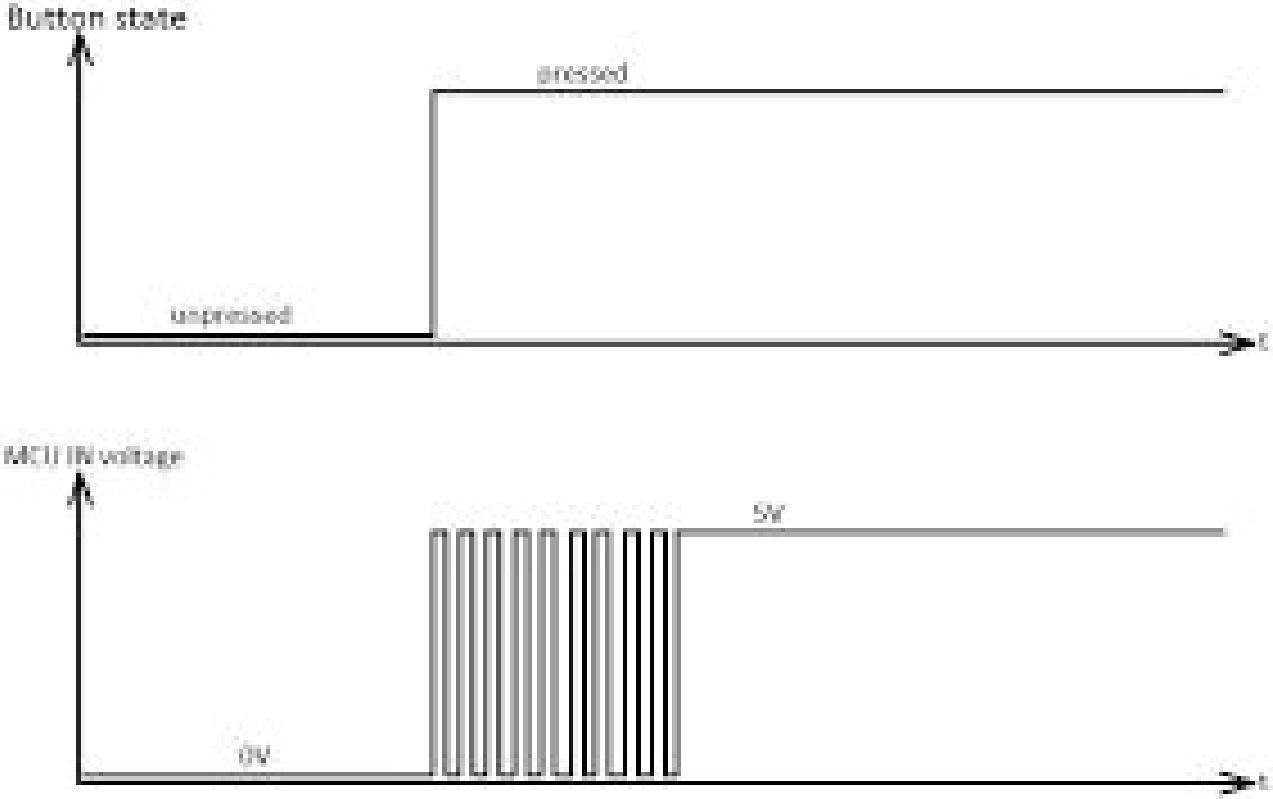
- Floating state of pin
  -



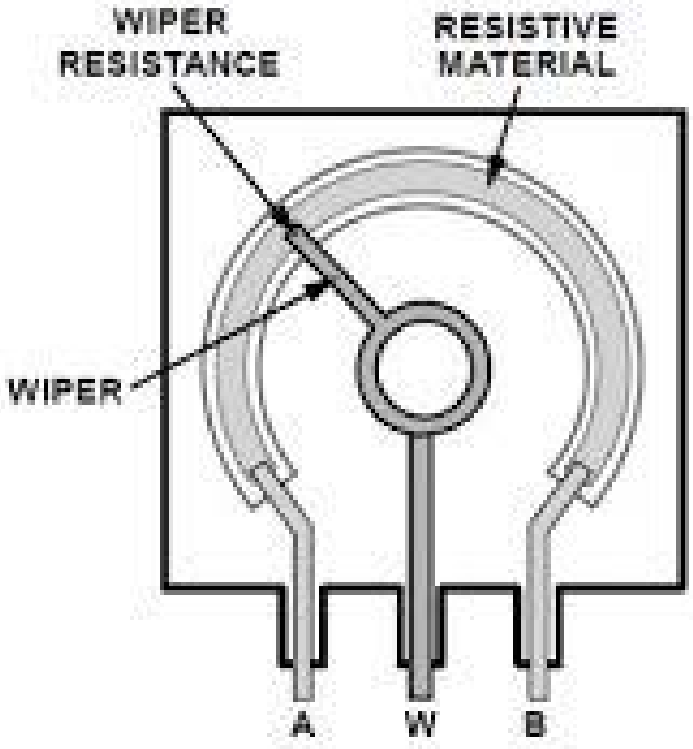
- Pull up and pull down resistors



- Debouncing



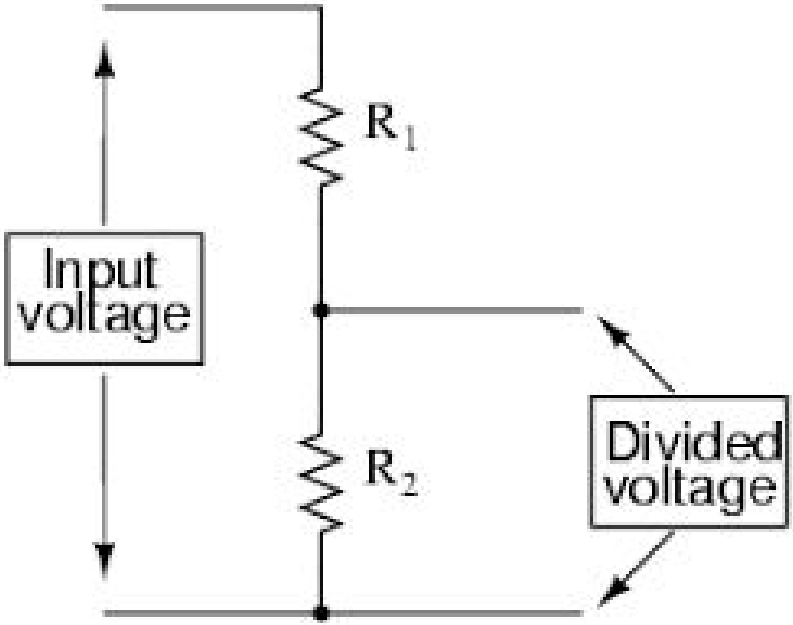
The POT



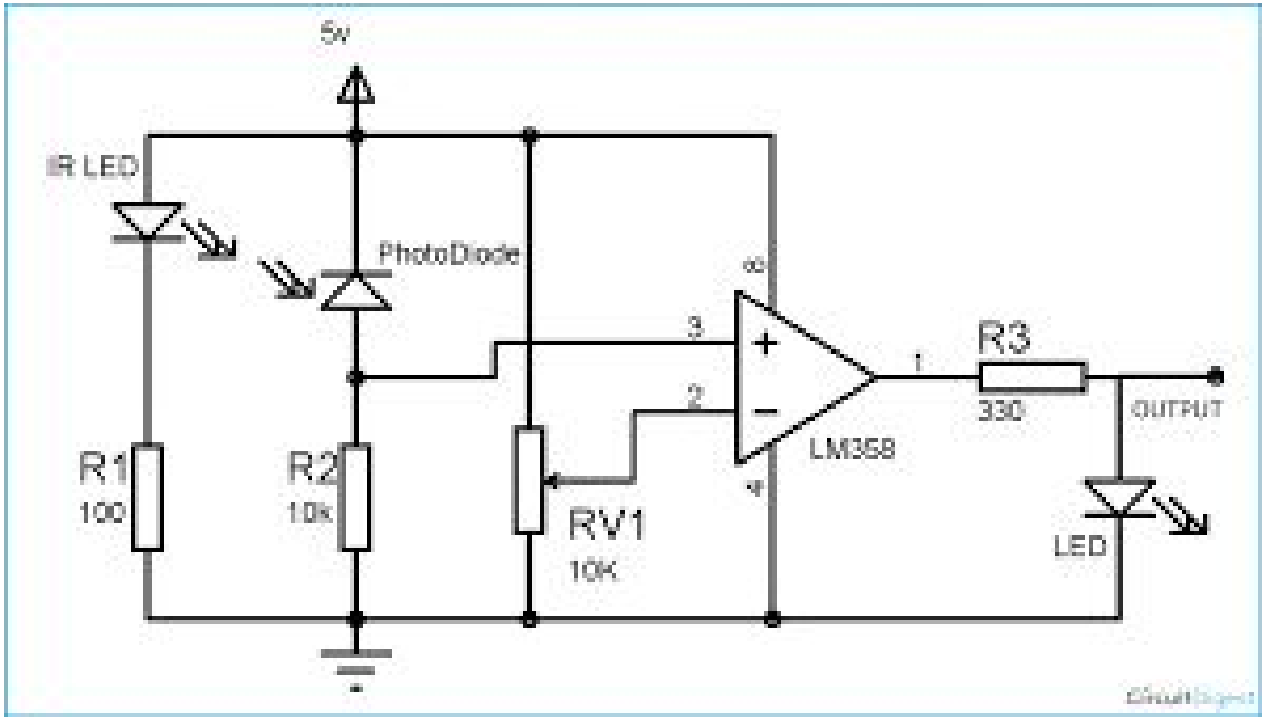
LM35 (Temperature Sensor)

```
int val;

int tempPin = 1;



void setup()

{

Serial.begin(9600);

}

void loop()

{

val = analogRead(tempPin);

float mv = ( val/1024.0)*5000;
```

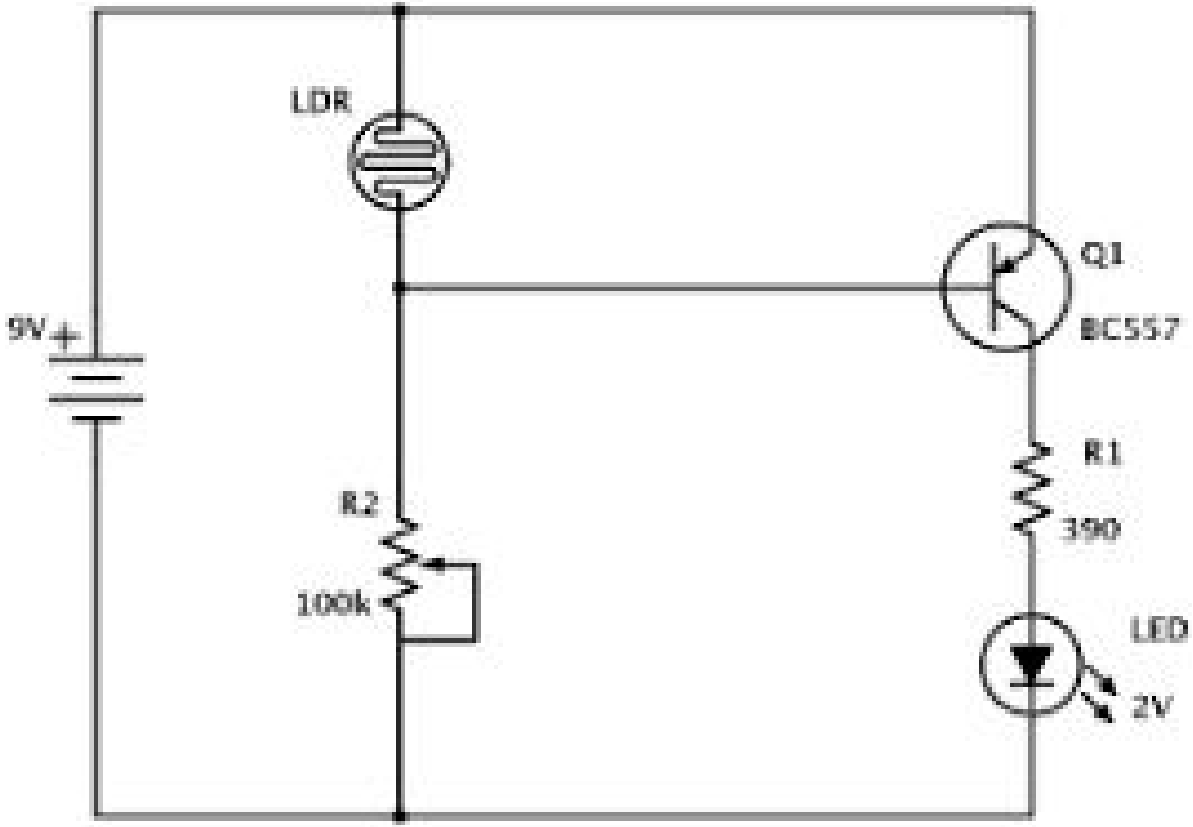
Potential Divider



IR Sensor

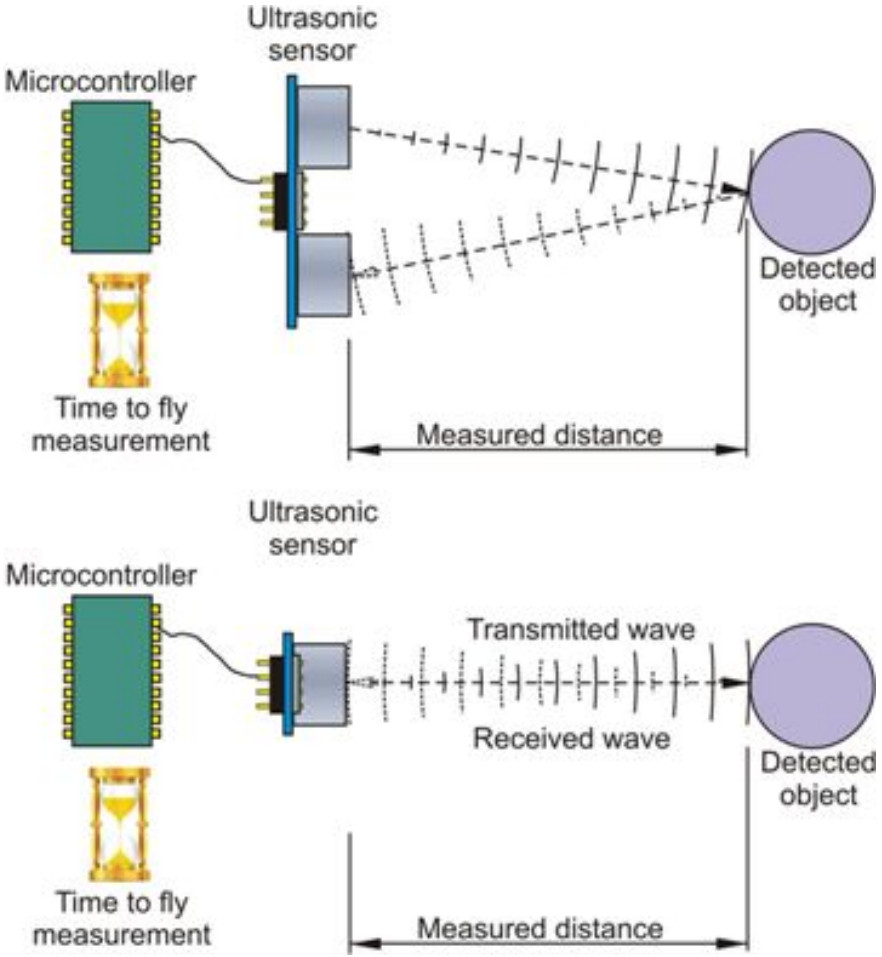

5v
IR LED
PhotoDiode
R1
100
R2
10k
RV1
10K
3
2
8
4
1
LM358
R3
330
OUTPUT
LED

LDR Sensor

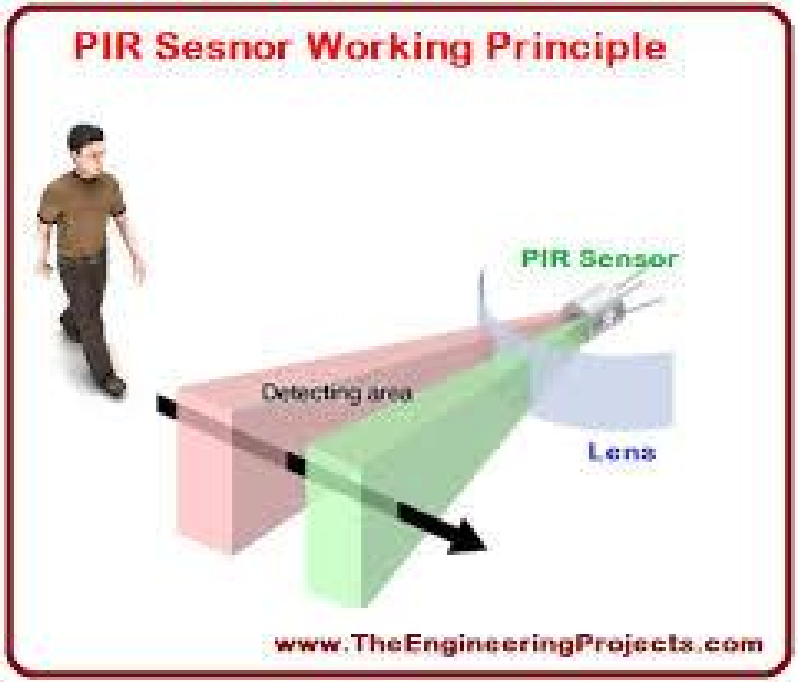


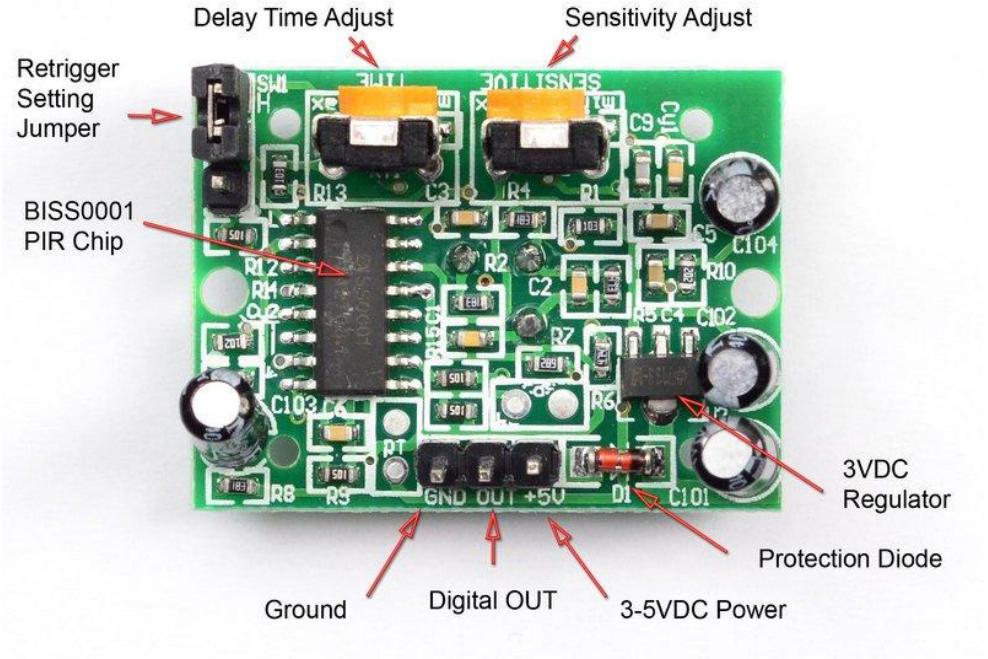
Ultrasonic Sensor

- Give a 10 micro second pulse on trigger pin
- Use pulseIn() to measure the time period of the pulse on echo pin.
- Convert to get distance



PIR Sensor


PIR Sesnor Working Principle
PIR Sensor
Detecting area
Lens
www.TheEngineeringProjects.com


Delay Time Adjust
Sensitivity Adjust
Retrigger Setting Jumper
BISS0001 PIR Chip
3VDC Regulator
Protection Diode
Ground
Digital OUT
3-5VDC Power
GND OUT +5V

Bonous

```
//Plays a song on a speaker

#include "pitches.h" //Header file with pitch definitions

const int SPEAKER=9; //Speaker Pin

//Note Array

int notes[] = {

NOTE_A4, NOTE_E3, NOTE_A4, 0,

NOTE_A4, NOTE_E3, NOTE_A4, 0,

NOTE_E4, NOTE_D4, NOTE_C4, NOTE_B4, NOTE_A4, NOTE_B4, NOTE_C4,
```