CPE 301: Sensors

Sensors

A sensor is a device or module (or subsystem) that can sense or detect **physical phenomena** in the environment, such as light, pressure, temperature, and motion.

The input of a sensor can be any physical phenomenon, but the output should be a signal within the human-readable form.

A sensor detects the changes or events around it and sends the data to any smart devices or embedded system.

Sensors

Vision sensors, motion sensors, position sensors, photoelectric sensors, radiation sensors, temperature sensors, pressure sensors, proximity sensors, particle sensors, metal sensors, level sensors, electrical sensors, humidity sensors, flow sensors, leak sensors, flame sensors, contact sensors, and non-contact sensors.

How does a sensor work?

The **input** will be the physical quantity we will be measuring and the **output** will be the change in electrical property of the sensor element.

Sensors react to changing physical quantity by altering their electrical properties.

Digital Sensors could be as simple as an on/off sensor. Either there is light or not. 1 or 0. True or False.

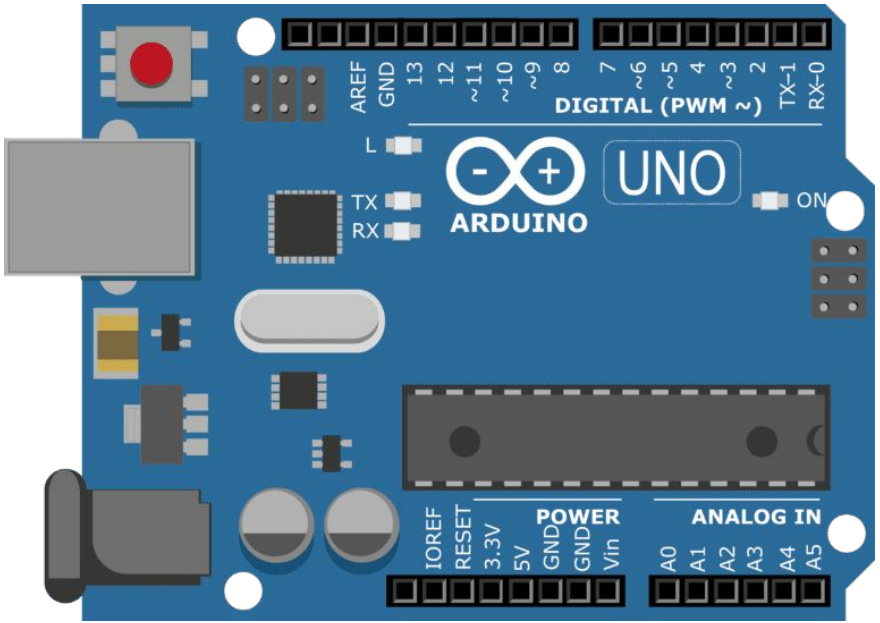
What if you want to measure ambient light using LDR or temperature using a temperature sensor?

Analog Sensor is simple. It has a voltage output that can vary with respect to the external physical quantity.

Connecting Sensors to Arduino

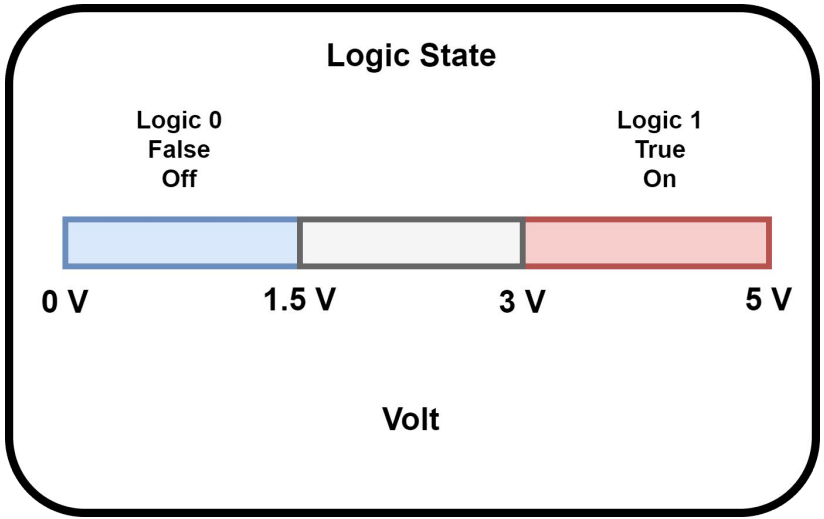
GPIO Pins

Arduino has several digital Input/Output pins, generally known as GPIO pins or General Purpose Input Output Pins that are designed either to provide input to the processor or get output from the processor. We will be making use of these pins to read the data from sensors.



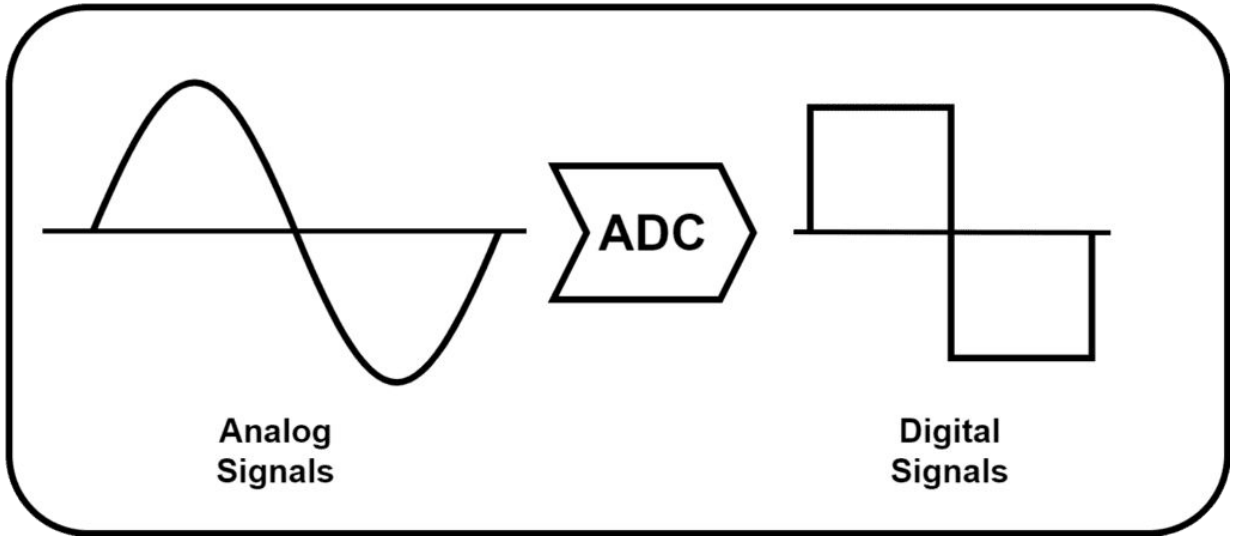
Arduino Digital Read

Microcontrollers in Arduino are capable of detecting binary signals or Digital signals – 0 or 1. That is, for a 5V Arduino board, it understands 0V as a logic 0 and a voltage above 3V as a logic 1.



Arduino Analog Read

To measure the value of analog signals, the Arduino has a built-in analog to digital converters or ADC at certain pins. These ADC circuits turn the analog voltage into a digital value that the Arduino can read. These pins are what we call the analog pins. This is where we connect the analog sensor input.



Temp/humidity sensor

DHT11 humidity and temperature sensor.

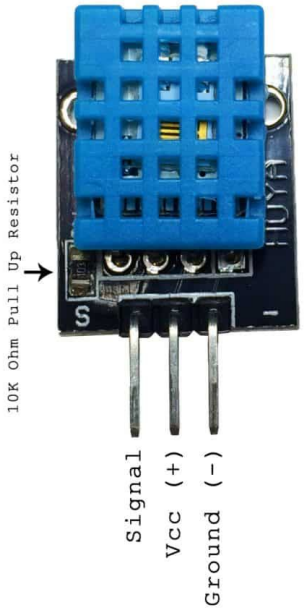
It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and sends a digital signal on the data pin.

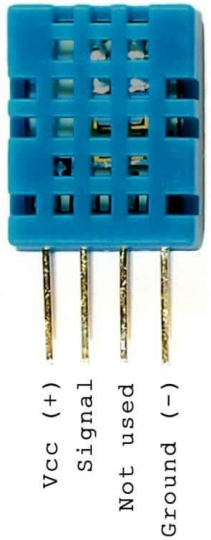
Here are the ranges and accuracy of the DHT11:

- Humidity Range: 20-90% RH
- Humidity Accuracy: ±5% RH
- Temperature Range: 0-50 °C
- Temperature Accuracy: ±2% °C
- Operating Voltage: 3V to 5.5V

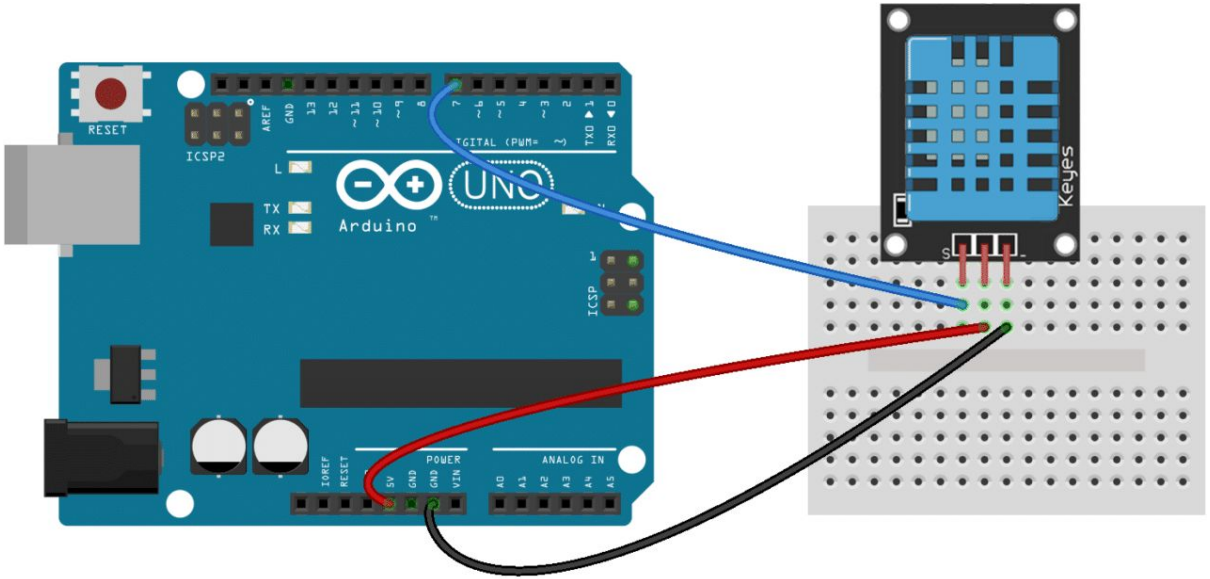
DHT11 sensor

The DHT11 measures *relative humidity*. Relative humidity is the amount of water vapor in air vs. the saturation point of water vapor in air. At the saturation point, water vapor starts to condense and accumulate on surfaces forming dew.

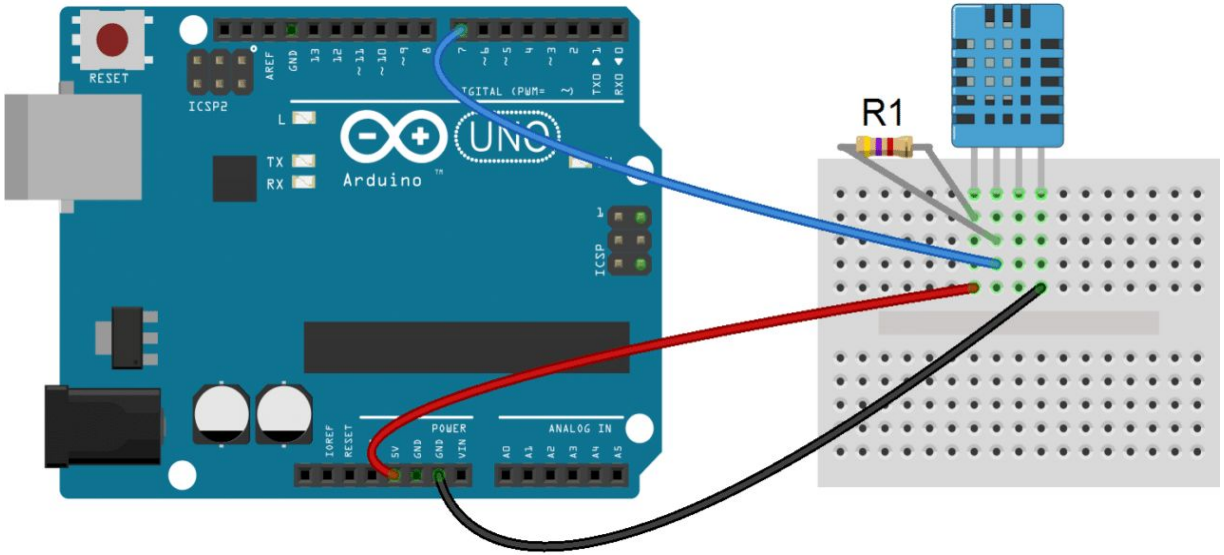




CONNECTING A THREE PIN DHT11



CONNECTING A FOUR PIN DHT11



Example Code:

```
#include <dht.h> //install the DHTLib library

dht DHT;

#define DHT11_PIN 7

void setup(){

  Serial.begin(9600);

}
```

```
void loop(){

  int chk = DHT.read11(DHT11_PIN);

  Serial.print("Temperature = ");

  Serial.println(DHT.temperature);

  Serial.print("Humidity = ");

  Serial.println(DHT.humidity);

  delay(1000);

}
```

Please make sure to check the final project's rules about using library.
https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/

Water Sensor

Connecting a water sensor to an Arduino is a great way to detect a leak, spill, flood, rain, etc. It can be used to detect the presence, the level, the volume and/or the absence of water. While this could be used to remind you to water your plants, there is a better Grove sensor for that.

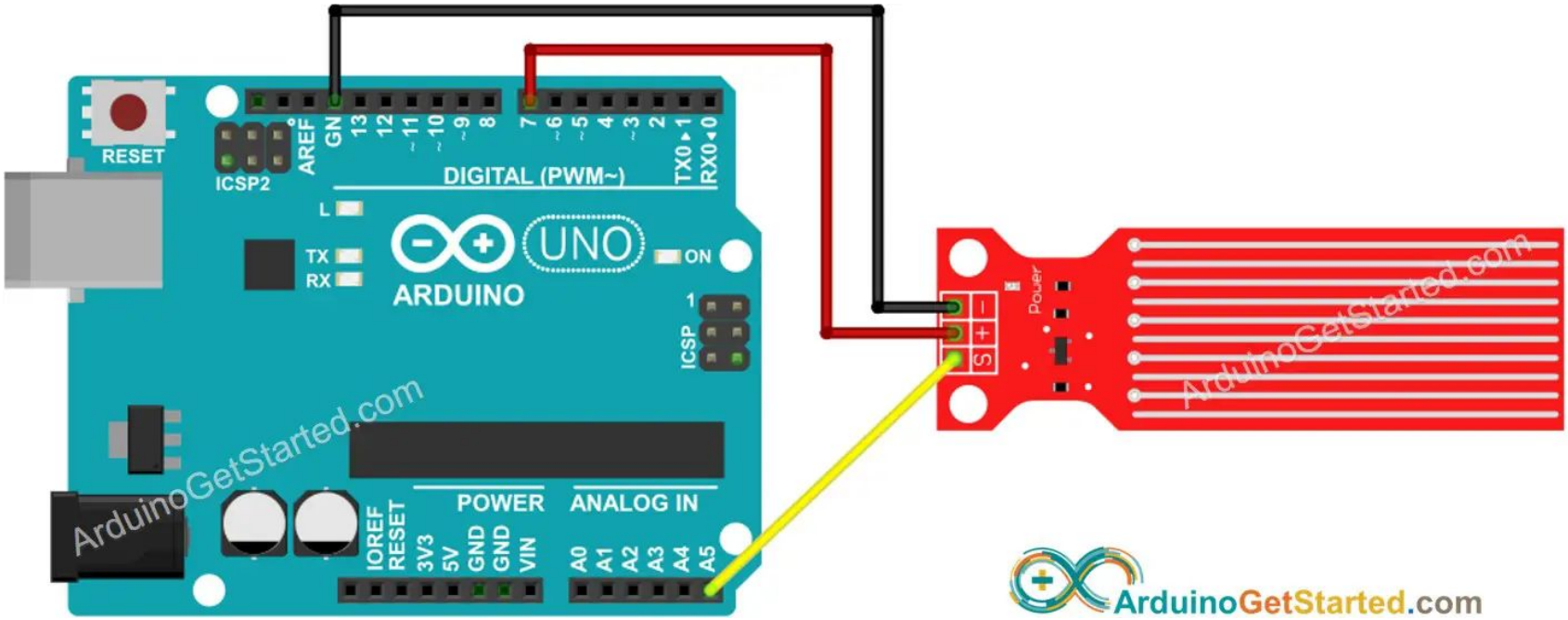
Water Sensor

Water sensor has three pins -

Signal (S) pin: one analog output that will be connected to the analog input of the arduino

V_{out}(+), and GND (-).

The more water the sensor is immersed in, the higher the output voltage will be in the signal pin.


RESET
ICSP2
AREF
GN
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
1
ICSP
POWER
ANALOG IN
IOREF
RESET
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5
S
+
-
Power
ArduinoGetStarted.com
ArduinoGetStarted.com

```
#define POWER_PIN  7
#define SIGNAL_PIN A5
int value = 0; // variable to store the sensor value
void setup() {
  Serial.begin(9600);
  pinMode(POWER_PIN, OUTPUT);    // configure D7 pin as an OUTPUT
  digitalWrite(POWER_PIN, LOW); // turn the sensor OFF
}
void loop() {
  digitalWrite(POWER_PIN, HIGH);  // turn the sensor ON
  delay(10);                      // wait 10 milliseconds
  value = analogRead(SIGNAL_PIN); // read the analog value from sensor
  digitalWrite(POWER_PIN, LOW);   // turn the sensor OFF

  Serial.print("Sensor value: " );
  Serial.println(value);
  delay(1000);
}
```

Please make sure to check the final project's rules about using library.

https://arduinogetstarted.com/tutorials/arduino-water-sensor

https://circuitcellar.com/research-design-hub/basics-of-design/sensor-solutions-for-embedded-systems/

https://rootsaid.com/arduino-sensors/

https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/

https://arduinogetstarted.com/tutorials/arduino-water-sensor

Git vs Github

Git is an open-source, version control tool created in 2005 by developers working on the Linux operating system; GitHub is a company founded in 2008 that makes tools which integrate with git. You do not need GitHub to use git, but you cannot use GitHub without using git.

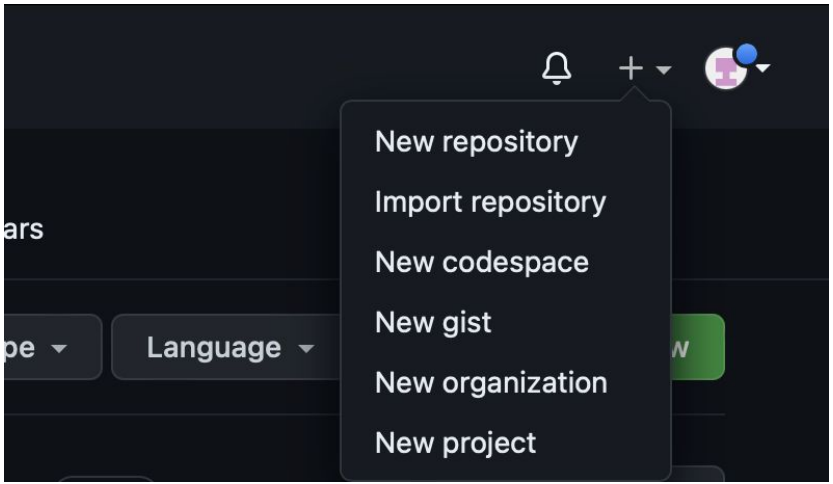
https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners

https://docs.github.com/en/repositories/viewing-activity-and-data-for-your-repository/viewing-a-projects-contributors

Go to the link: https://github.com/ . Fill the sign up form and click on “Sign up for Github”.

Click on Create a new repository



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * **Repository name** *

anima-unr / Github-tutorial

Great repository names are short and memorable. Need inspiration? How about literate-succotash?

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.

.gitignore template: None

Choose a license
A license tells others what they can and can't do with your code. Learn more.

License: None

You are creating a public repository in your personal account.

Create repository