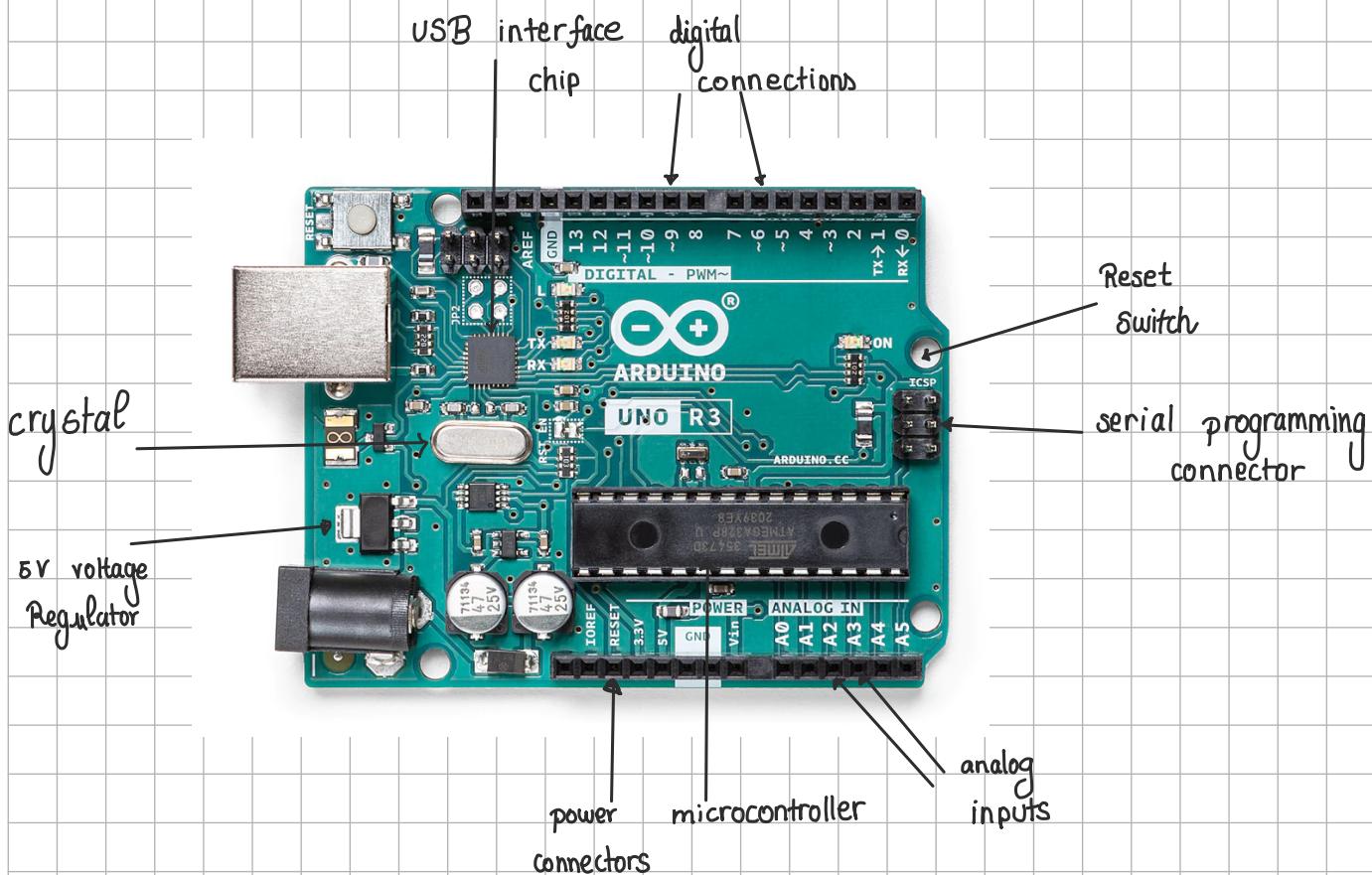


Circuit Summary

- a tour in the Arduino Board
- Input and outputs (Digital_analog)
- servomotor
- ultra sonic Sensor
- different input/output

A tour in Arduino Board



Power Supply

directly below the USB connector is 5 volt voltage regulator
this would regulate +12V into a constant 5V

Power connections

at the bottom, you will find the power connections starting from Reset (it is similar to the reset button) it resets the microcontroller so that it begins its program from the start.

the rest of the pins in this section just provide different voltages (3.5V, 5V, GND)
GND just means zero voltage.

Analog Inputs:

the pins labelled as P0 - P5 are called analog, they can be used to measure the voltage connected to them. they measure voltage and not current. since they have large internal resistance, then only a low current could pass to them.

they can be used as digital inputs or outputs.

Digital Connections

take a look at the top you will find pins labeled Digital 0 to 13 these can be either used as either inputs or outputs. they can be used as either inputs or outputs.

When used as outputs, they behave rather like the power supply voltages. they are all 5V if you turn them off at 0V. the first two 0 and 1 are labelled RX and TX for receive and transmit. they are reserved and used indirectly to receive and transmit connections for USB link.

they supply 40 mA at 5V.

microcontroller

it is the CPU of the Arduino, it controls everything that goes within the device, it fetches instruction from the Ram

there is also an EEPROM memory used to store the data that you don't want to lose in any event.

(/)

crystal

this is a quartz crystal oscillator, it ticks 16 million times a second on each tick, the microcontroller can perform one operation.

Arduino family

UNO



the older boards look very similar to the Arduino Uno
they can supply more current on its 3.3 V supply.
they are equipped with ATmega328 (they have more memory).

Mega

it boasts a huge collection of input output
ports

54 input /output

128 KB of flash memory

8 KB of RAM

4 KB of ROM.



Nano



it is a very useful device , it can be plugged
into the breadboard as if it were a chip.

there are also circular ones "lilypad Arduino".

C language basics

let's take a look at the code:

1. digital write (13, HIGH);
2. delay (500);
3. digital write (13, LOW);

the first line would set the output of pin 13 to HIGH , this is the pin is linked with a led so this means the led lights up

the second line tells it to wait for 0.5 s

the third line tells it to set the LED back off.

digitalwrite has two arguments :
1. the pin which is linked to bread
2. LOW / HIGH to set the voltage to either zero / 5v.

there are two functions that you must define setup and loop it's a way to separate between things that are done once and things that are done for multiple times the setup will run once while the loop will run recursively.

take a look at this code:

```
void setup () {  
    pinMode (13, OUTPUT); // this will  
    digitalWrite (13, HIGH); // run once.  
}  
void loop () {  
}
```

we are familiar with the digital write , now, we have encountered a new builtin function pinMode (); it sets a pin to either output or input.

when you run this code you will find that the led will turn on and stays on until the program is stopped from executing.

now, let's take a look at the following code

```
void setup () {  
    pinMode (13, OUTPUT);  
}  
void loop () {  
    digitalWrite (13, OUTPUT);  
    delay (500);
```

```
digitalWrite(13, low);  
}
```

this would be like

1. run the setup and set the pin 13 to OUTPUT
2. now we will run the loop
 - a. set the voltage of the output to HIGH
 - b. wait for 0.5 s
 - c. set the voltage of the output pin 13 to LOW

3. run the loop again.

variables

to declare a led pin we can do the following

```
int ledPin=13; // it shall be declared as global to be used in all the  
// functions.
```

for naming we use CamelCase

serial:

```
serial.begin(9600); // declared within the set up.
```

```
serial.print(a);
```

this code allows to print code on the monitor.

define

```
#define ledPin 13
```

this is a pre compiled directive and it runs before compilation.

it is used to declare a pin.

functions.

they are very similar to what we have seen in Cpp,
to read from a USB we can use

```
ch = serial.read(); // what it reads will be stored within ch.
```

Input and Outputs

Arduino is about physical computing thus you need how to use various options for read and write.

outputs can be digital which just means switched between 0V or 5V
or analog which allows you to set it to any value between 0 and 5V.

likewise, inputs can be either digital (determining whether a button is pressed or not)
analog (light sensor).

Digital output

```
int outPin = 4;  
void setup() {  
    pinMode(outPin, OUTPUT);  
    Serial.begin(9600);  
    Serial.println("Enter 1 or 0");  
}  
void loop()  
{  
    if(serial.available > 0)  
    {  
        char ch = Serial.read();  
        if(ch == '1') {  
            digitalWrite(outPin, HIGH);  
        }  
        else if(ch == '0') {  
            digitalWrite(outPin, LOW);  
        }  
    }  
}
```

Digital inputs

one of the main use of digital inputs is to detect when a switch has been closed

the digital input can be either on or off if the voltage is less than 2.5V it will be 0 and if it is above it will be on 1.

take a look at this code

```
int reading = digitalRead (inputPin);
```

you can use the value to either switch on or off a LED.

Analog output

a few of digital pins namely 3,5,6,9, 10,11 can provide variable output they have ~ next to them. Which means we can control the amount of power. thus, we can consider them as analog outputs.

```
void setup ()  
{  
    pinMode (outputPin, OUTPUT);  
    Serial.begin (9600);  
    Serial.println ("Enter volts 0 to 5");  
}  
  
void loop ()  
{  
    if (Serial.available () > 0)  
    {  
        char ch = Serial.read ();  
        int vol = (ch - '0') * 51;  
        analogWrite (outputPin, Volts);  
    }  
}
```

Analog input:

they give a value between 0 and 1023 depending on the voltage of that Pin

we will use A0-A5 as Analog Pins.

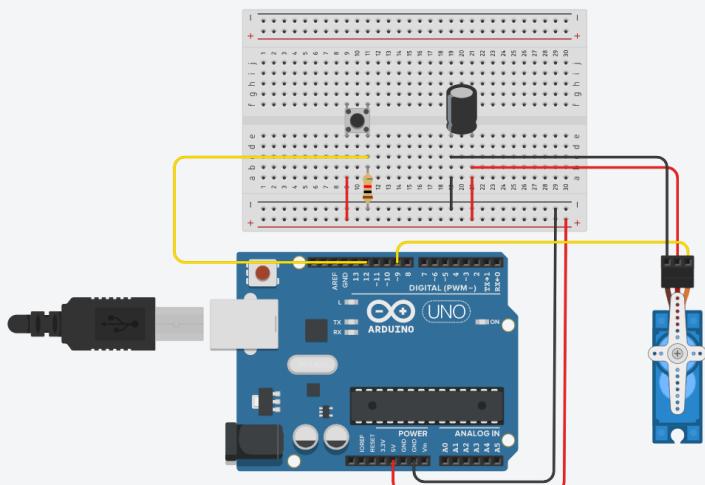
```
int analogPin = 0;  
void setup ()  
{  
    Serial.begin (9600);  
}
```

```

void loop() {
    int reading = analogRead(A0);
    float voltage = reading / 1024.0;
    Serial.print("Reading ");
    Serial.print(reading);
}

```

servo motor



how to link it ?

red 5v

yellow g~ (any analog output).

black GND

```

#include <Servo.h>
Servo myservo;
void setup()
{
    myservo.attach(9);
}
void loop()
{
    myservo.write(90); // pos is a degree of rotation.
}

```

ultra sonic



VCC to 5V

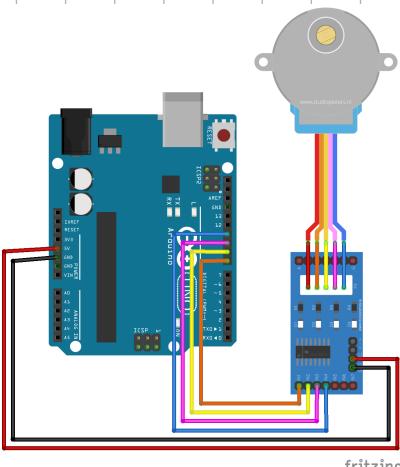
GND to GND

Trig to 9 } / any analog pin you want.
echo to 10.

take a look at the code

```
void setup()
{
    pinMode (trigPin, OUTPUT);
    pinMode (echoPin, INPUT);
    Serial.begin (9600);
}

void loop ()
{
    digitalWrite (trigPin, LOW);
    delayMicroseconds (2);
    digitalWrite (trigPin, HIGH);
    delayMicroseconds (2);
    digitalWrite (trigPin, LOW);
    duration = pulseIn (echoPin , HIGH);
    distance = (duration * 0.343) / 2;
    Serial .println (distance);
}
```



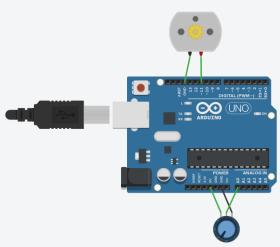
Step motor

we can clearly see from the picture that
 VCC to 5v
 GND to ground
 IN1, IN2, IN3, IN4 to pins

now, we will take a look at the code:

```
#include <stepper.h>
// define the number of steps per rotation
const int stepsPerRotation = 2048;
// create an object of stepper
Stepper myStepper = Stepper(stepsPerRotation, Pin1, Pin2, Pin3, Pin4);
void setup()
{
}
void loop()
{
    myStepper.setSpeed(5);
    myStepper.step(stepsPerRotation);
    delay(1000);
    myStepper.setSpeed(10);
    myStepper.set(-stepsPerRotation);
    delay(1000);
}
```

DC motor

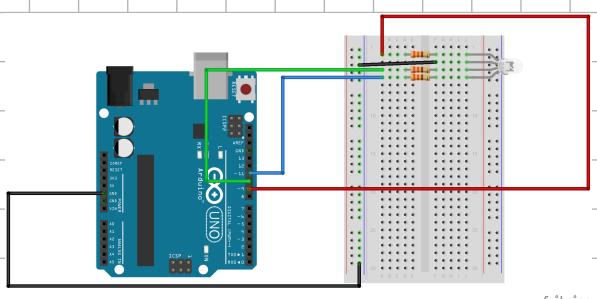


the black to GND
 the red to a pin

take a look at the code:

```
int basePin = 5;  
int speed;  
void setup(){  
    pinMode (basePin, OUTPUT);  
    Serial.begin (9600);  
    while (!Serial)  
    {  
        Serial.println ("enter the speed of the motor");  
    }  
}  
void loop()  
{  
    if (Serial.available > 0)  
    {  
        speed = serial.parseInt();  
        if (speed >= 0 && speed <= 255){  
            analogWrite (basePin, speed);  
        }  
    }  
}
```

RGB LED



first linked with pin (red)

second linked with GND

third and fourth linked with pin
(green) (blue)

take a look at the code:

```
const int greenPin = 9;  
const int bluePin = 10;  
const int redPin = 11;  
  
void setup() {  
    pinMode (greenPin, OUTPUT);  
    pinMode (bluePin, OUTPUT);  
}
```

```

pinMode (redPin , OUTPUT);
}
void loop(){
;
SetColor (red, green, Blue);
;
}

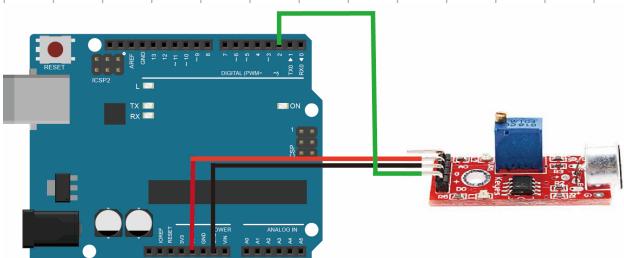
void setColor( int a, int b, int c){

analogwrite ( redPin, a );
analogwrite ( greenPin , b );
analogwrite ( bluePin, c );
}

}

```

Big sound



VCC to 5v
GND to Ground
Ao to Ao

// code

```

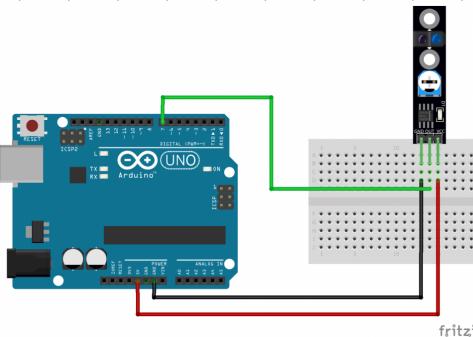
const int SoundSensorPin = A0;

void Setup(){
pinMode (soundSensorPin, INPUT);
Serial.begin (9600);
}

void Loop(){
int soundvalue = analogRead (SoundsensorPin);
serial.println (sound value);
}

```

tracking sensor



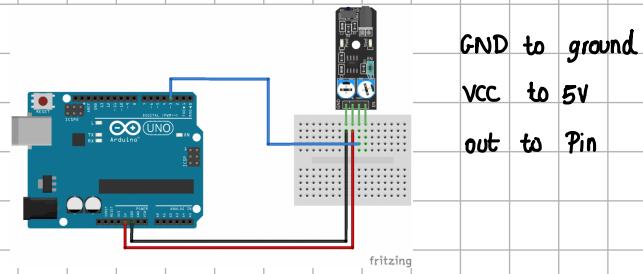
GND to GROUND

VCC to 5V

out to aPin

```
int SensorPin = 7;  
  
void Setup()  
{  
    pinMode (SensorPin , INPUT);  
    Serial.begin (9600);  
}  
  
void loop()  
{  
    int val= digitalRead (SensorPin);  
    if (val == HIGH)  
    { serial.println (" detected");  
    }  
    else  
    {  
        serial.println (" not detected");  
    }  
    delay (500);  
}
```

Infrared avoidance sensor



code

```
const int Pin=3;  
void setup(){  
    pinMode(Pin, OUTPUT);  
}  
void loop(){  
    int state = digitalRead(Pin);  
    if(state == HIGH){  
        Serial.println("the obstacle is not present");  
    }  
    else  
        Serial.println("the obstacle is present");  
}
```

Human touch sensor



+ to 5V
G to GND
A_o to A_o

```

int touchPin = A0;

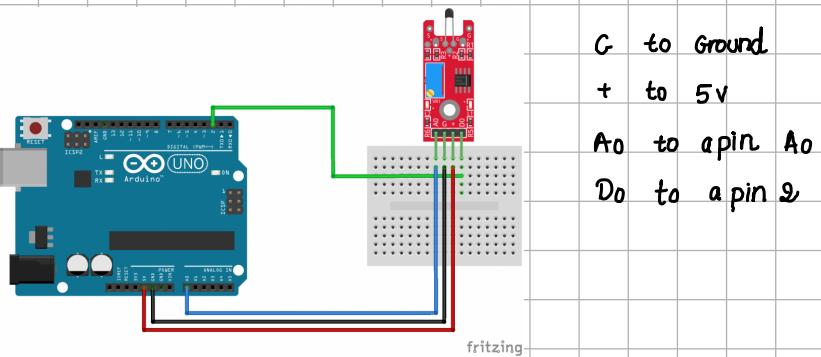
void Setup () {
    pinMode (touchPin, INPUT);
    Serial.begin (9600);
}

void loop ()
{
    int val = digitalRead (touchPin);
    if (val == HIGH)
        serial.println ("Human detected");
    else
        serial.println ("no Human detected");

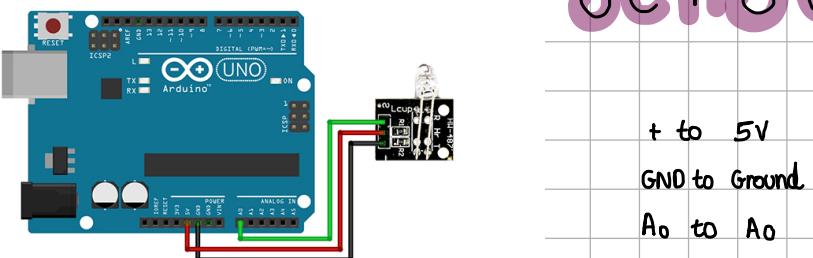
    delay (1000);
}

```

temperature sensor



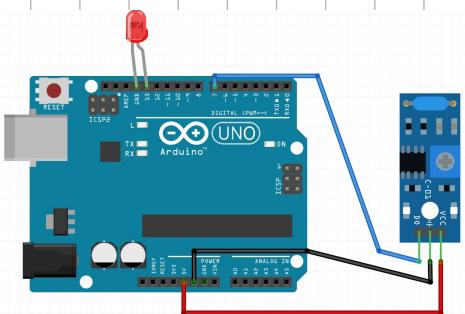
Heart beat sensor



take a look at the code

```
int pin = A0; float Pulse;  
void Setup(){  
    pinMode (pin, INPUT);  
    Serial.begin(9600);  
}  
void loop(){  
    pulse = 0;  
    for(int i=0; i<60; i++)  
    {  
        pulse += analogRead (pin);  
        delay (1000);  
    }  
    serial.println(" Your heart pulse is :");  
    Serial.println(pulse /60);  
}
```

vibration sensor



VCC to 5V
T to Ground

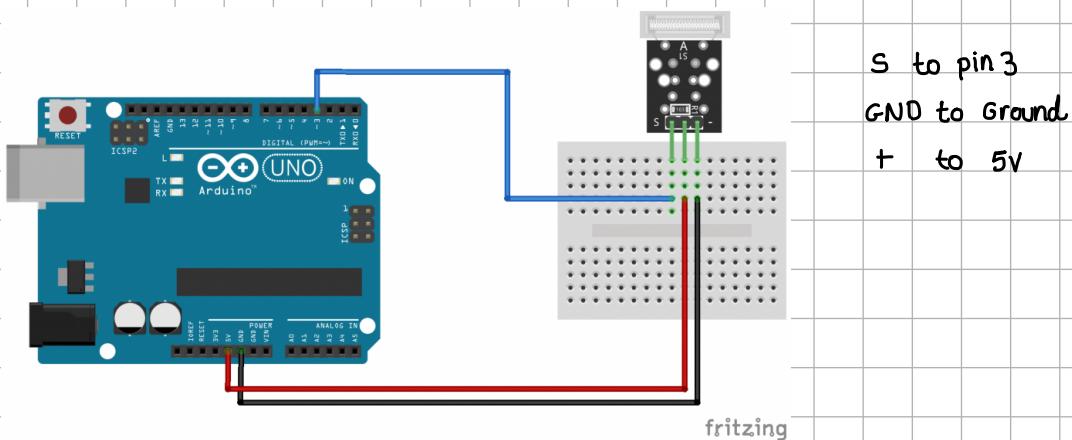
DO to a pin (any digital Pin).

here is the code

```
int Pin = 4;  
void setup()  
{  
    pinMode (Pin, INPUT); Serial.begin (9600);  
}  
void loop()  
{  
    int val;  
    val = digitalRead (Pin);  
    if(val == HIGH)  
    {  
        Serial.println (" vibration detected");  
    }  
}
```

```
delay(1000);  
}
```

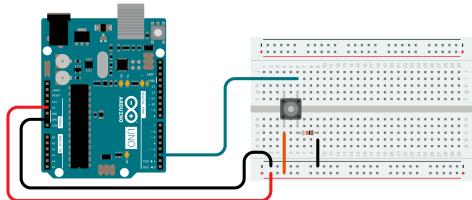
Knock sensor



here is the code

```
int Pin = 3;  
int val;  
void setup ()  
{  
    pinMode (Pin, INPUT); Serial.begin (9600);  
}  
void loop ()  
{  
    int val = digitalRead (Pin);  
    if (val == 1){  
        serial.Println ("a knock is detected");  
    }  
    delay (1000);  
}
```

Push button arduino



connect two wires to breadboard to 5 voltage and ground.

the third one goes from Pin 2 to one of the legs

the same leg connects through a pull down resistor to ground the other leg is connected to 5v

Here is the code

```
int Pin = 2;  
void Setup()  
{  
    pinMode(Pin, INPUT);  
    serial.begin(9600);  
}  
void Loop()  
{  
    int val = digitalRead(Pin);  
    if (val == 1)  
    {  
        Serial.println("button pushed");  
    }  
    else  
    {  
        Serial.println("button not pushed");  
    }  
    delay(1000);  
}
```

