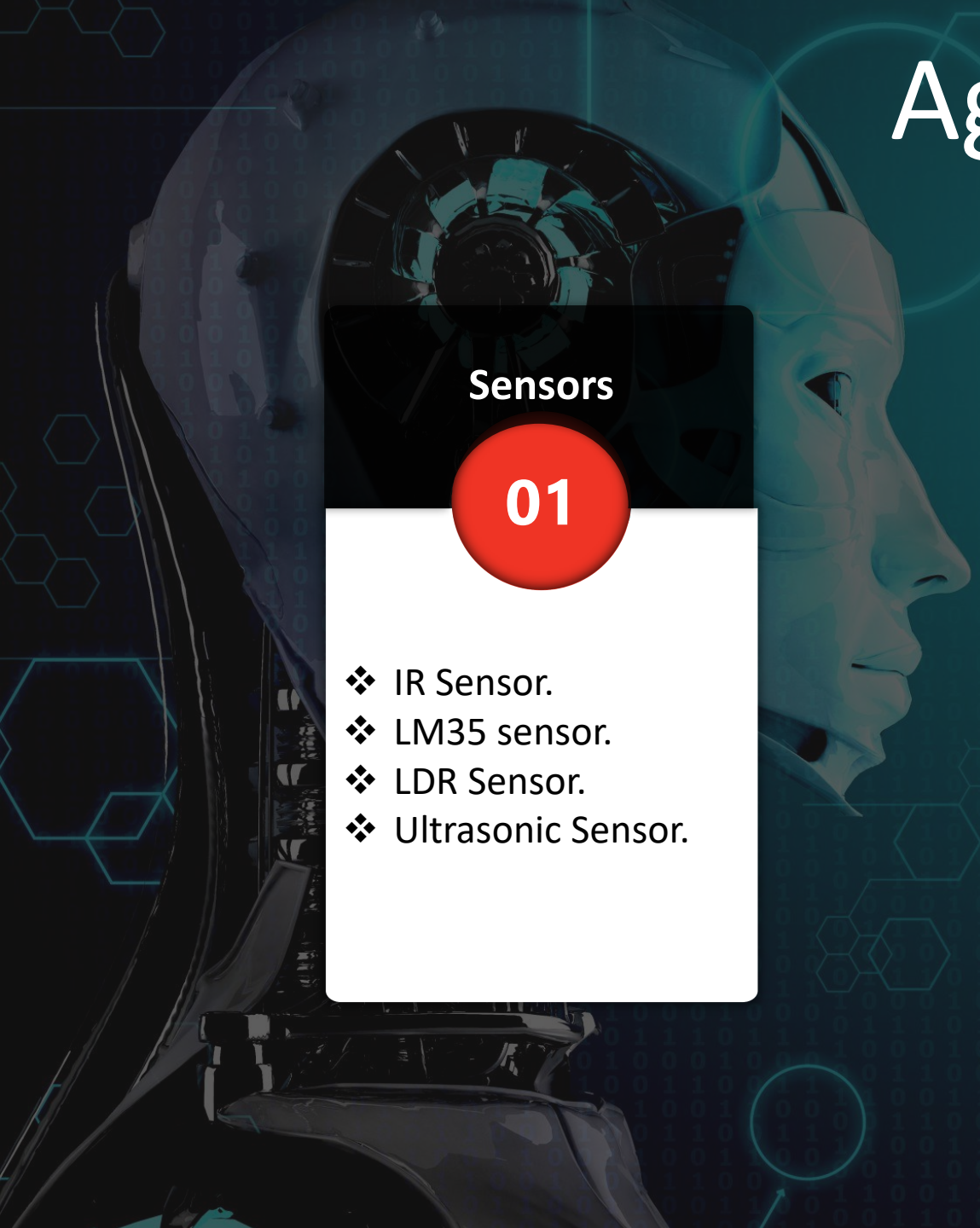
Session 3

--

Agenda



Sensors

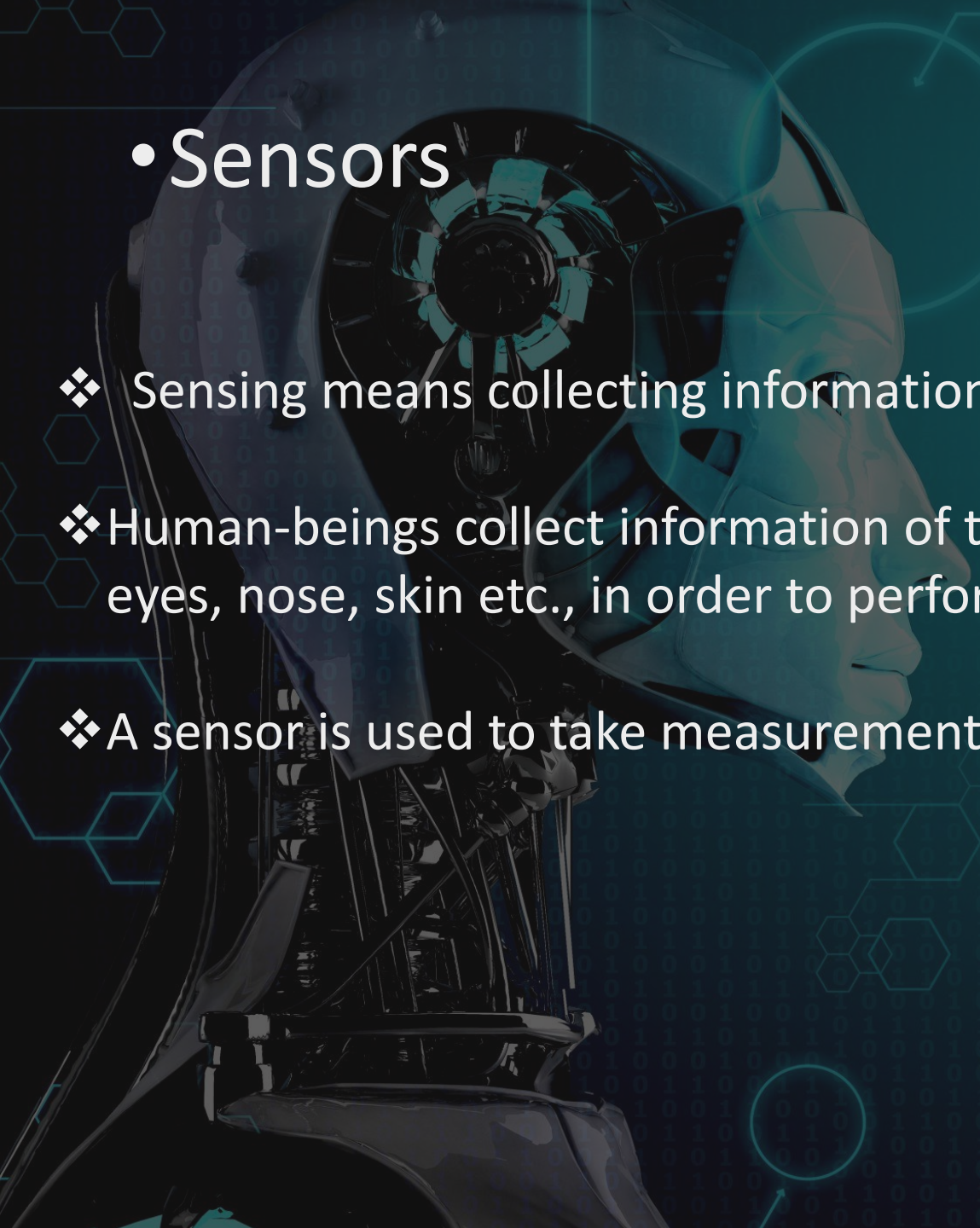
01

- IR Sensor.
- LM35 sensor.
- LDR Sensor.
- Ultrasonic Sensor.

Keypad

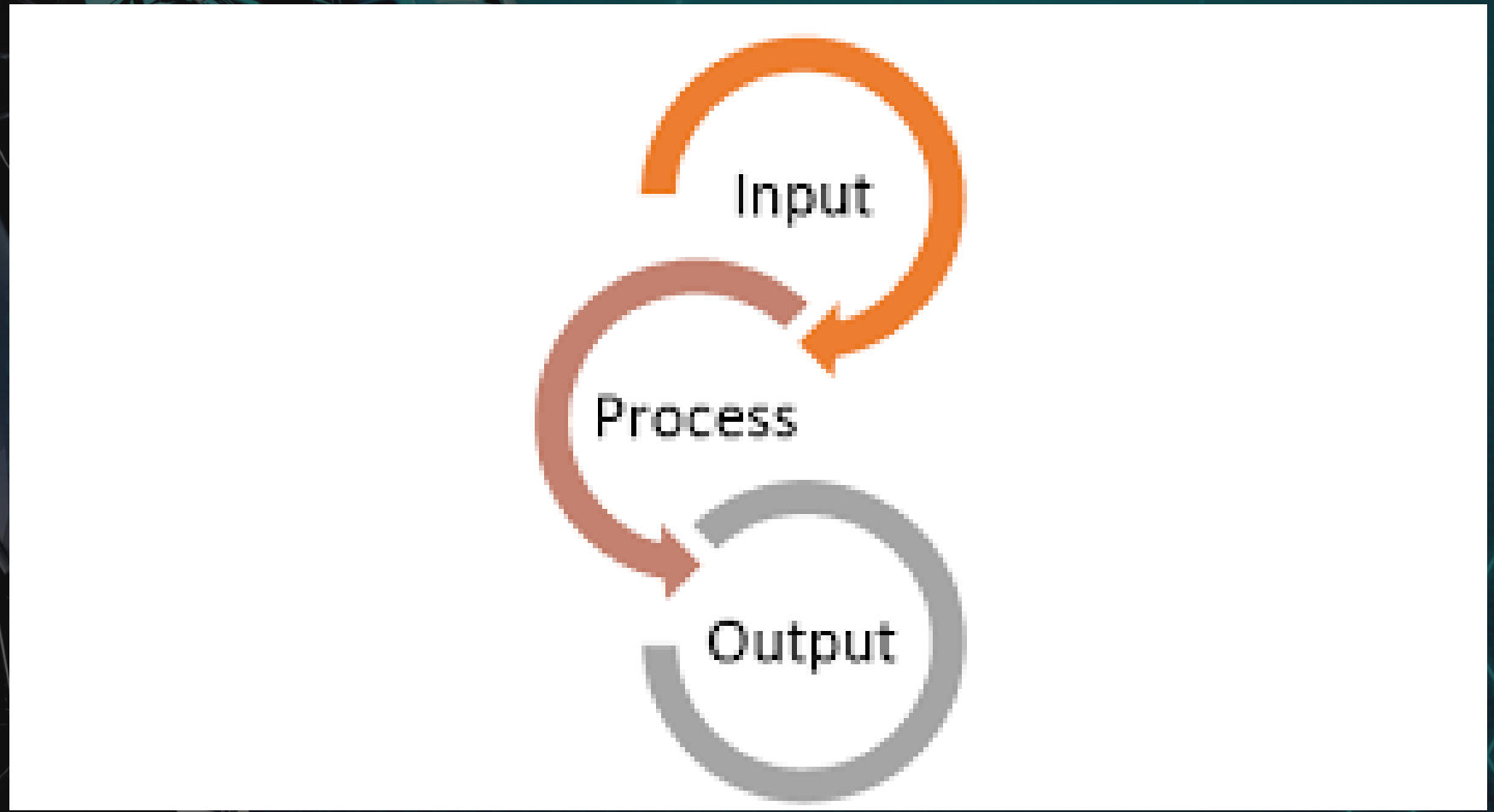
02

- Connection
- Functions



• Sensors

- Sensing means collecting information.
- Human-beings collect information of the surroundings sensors, namely eyes, nose, skin etc., in order to perform various tasks.
- A sensor is used to take measurement of physical variable.

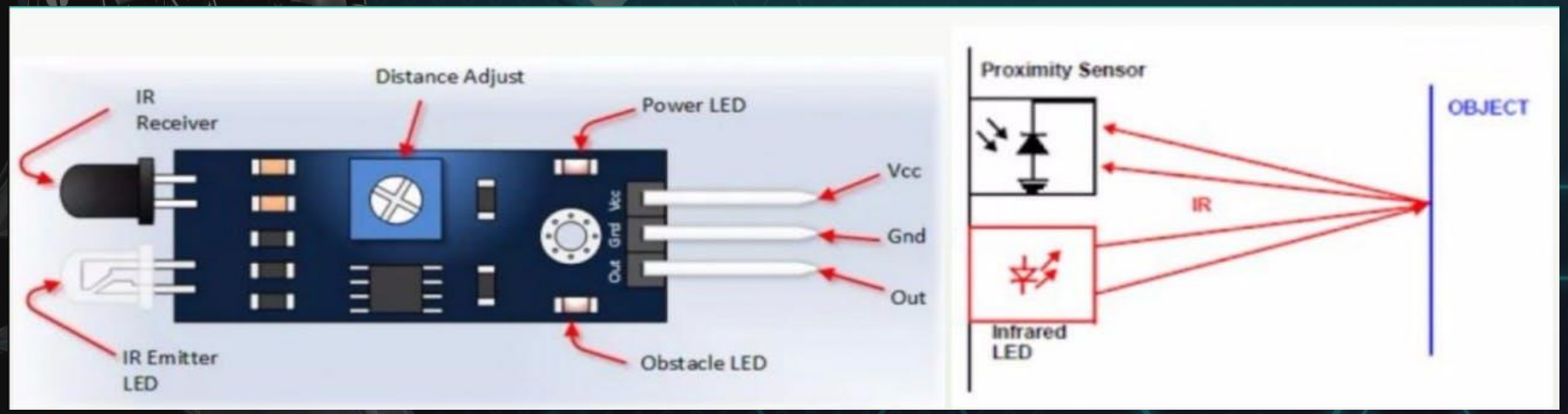

Input
Process
Output

• IR Sensor

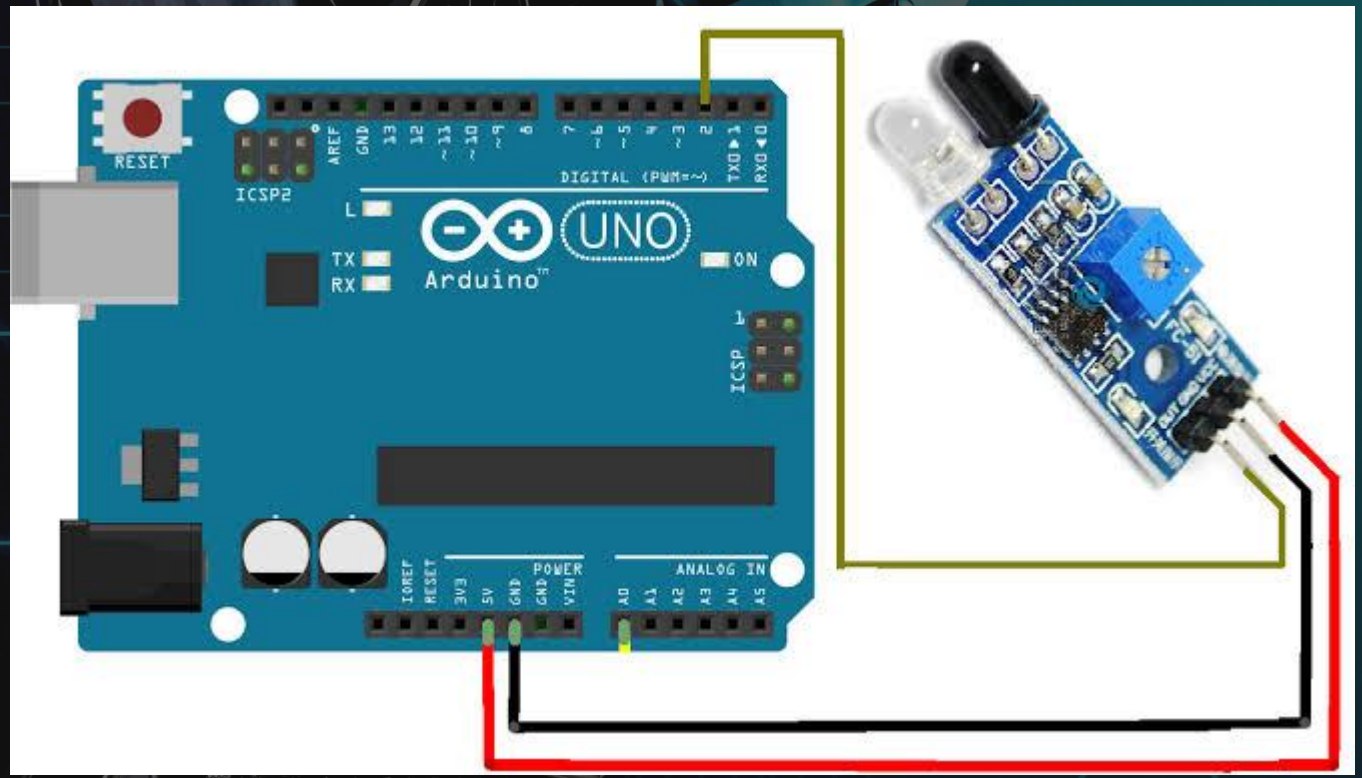
- An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment.
- IR is invisible to the human eye, as its wavelength is longer than that of visible light.
- Anything that emits heat gives off infrared radiation.

•IR Sensor

- ❖IR transmitter and receiver typically generate infrared using light emitting diodes (LEDs).
- ❖The main component of a receiver unit is usually a photodiode or phototransistor
- ❖A remote control flashes a pattern of invisible light, which is picked up and then turned into an instruction by the receiver module.



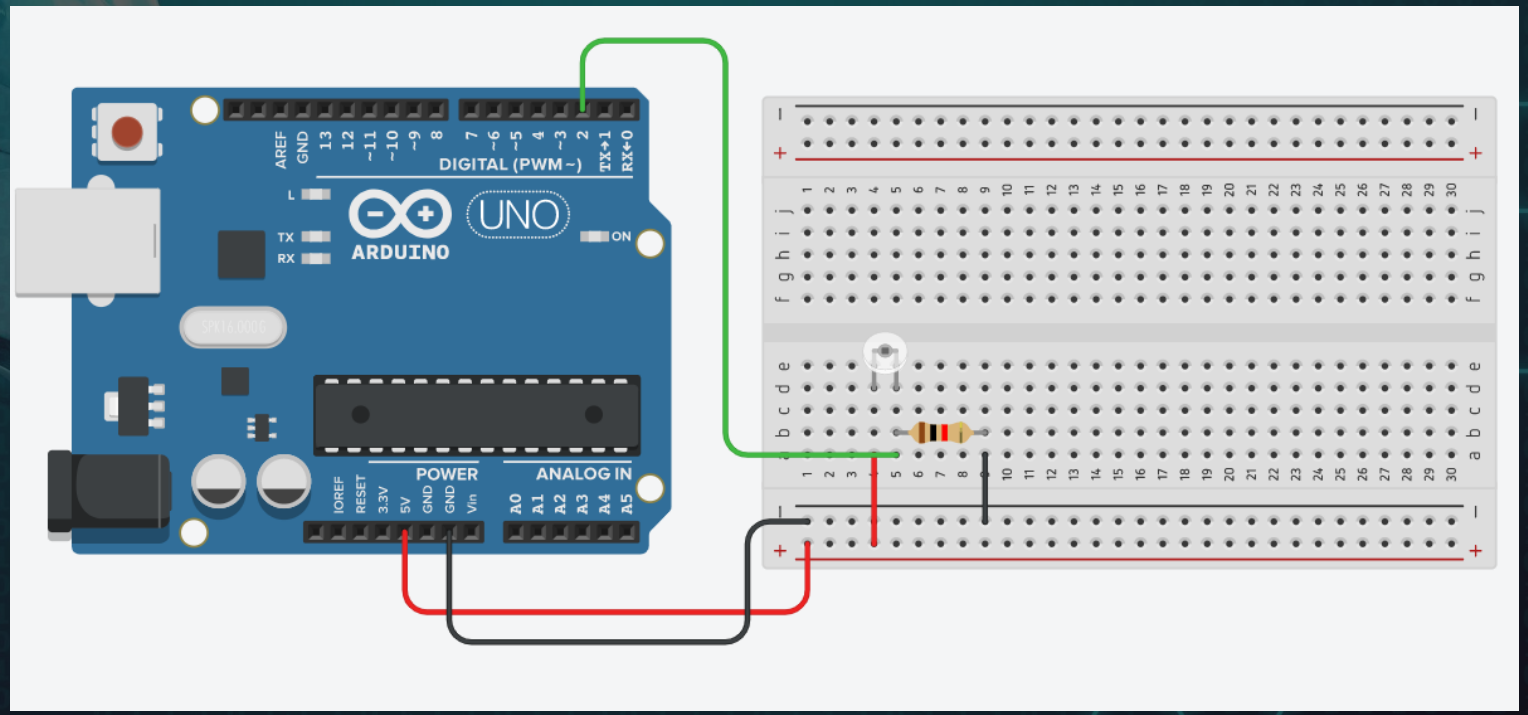
- Connection & Code

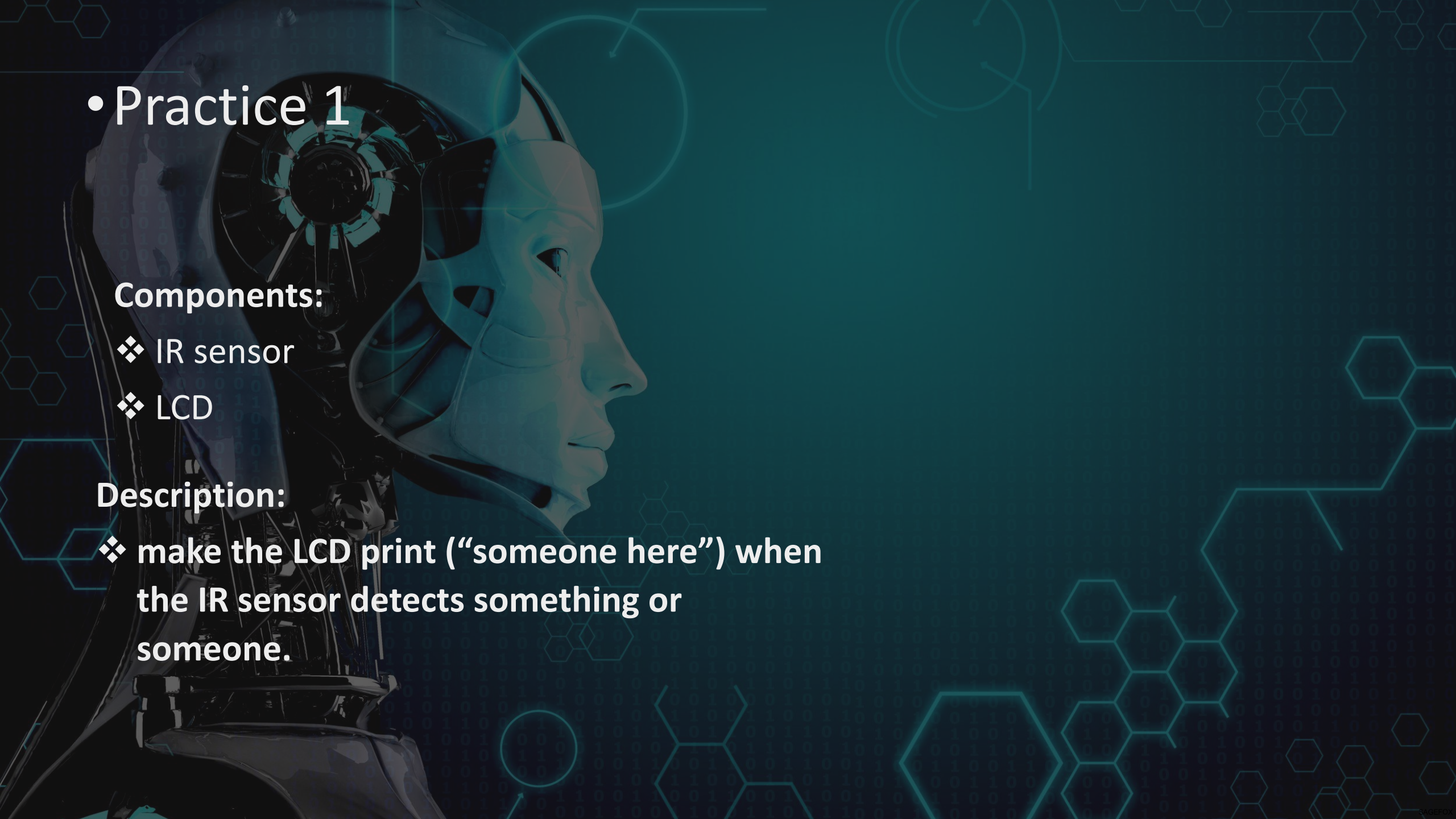



```
void setup()
{
  Serial.begin(9600);

  pinMode(2, INPUT);
}

void loop()
{
  int val=digitalRead(2);
  Serial.println(val);
}
```

- Con. Connection

In Tinkercad





- Practice 1

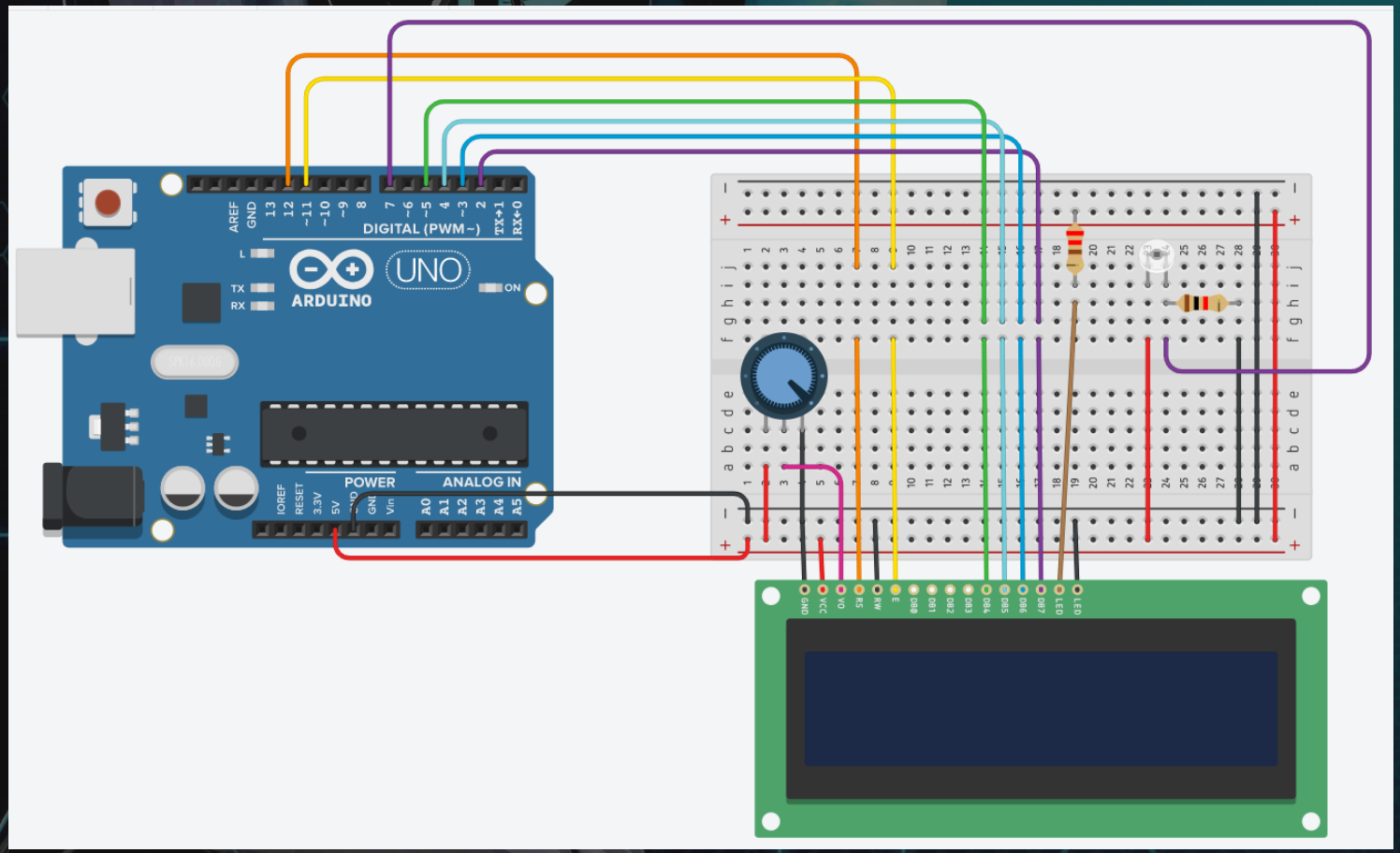
Components:

- ❖ IR sensor
- ❖ LCD

Description:

- ❖ **make the LCD print (“someone here”) when the IR sensor detects something or someone.**

• Practice 1 solution



sketch_aug01a | Arduino 1.8.8

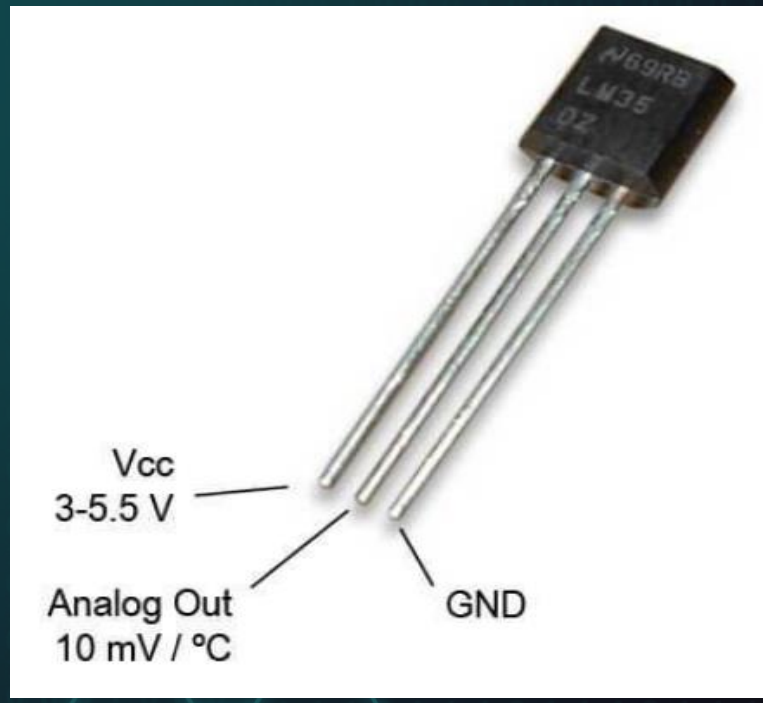
File Edit Sketch Tools Help

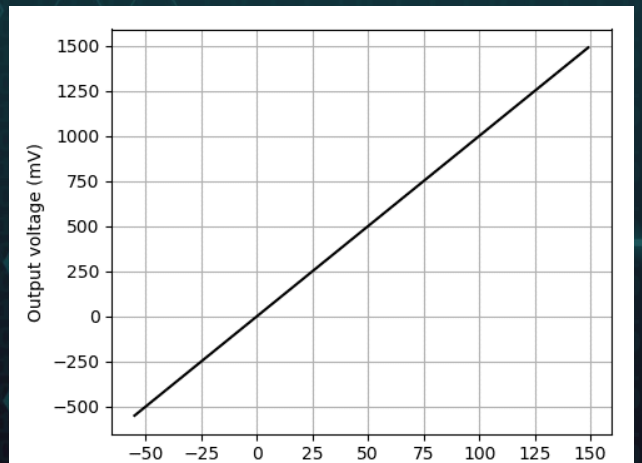
sketch_aug01a §

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int IR=7;
bool tmp = true;
void setup() {
  lcd.begin(16, 2);
  pinMode(IR,INPUT);
  lcd.setCursor(0,1);
}

void loop() {
  int value=digitalRead(IR);
  if(value==1 && tmp == true)
  {
  lcd.print("Someone here");
    tmp = false ;
  }
  else if (value == 0 && tmp == false)
  {
    lcd.clear();
    tmp = true;
  }
}
```

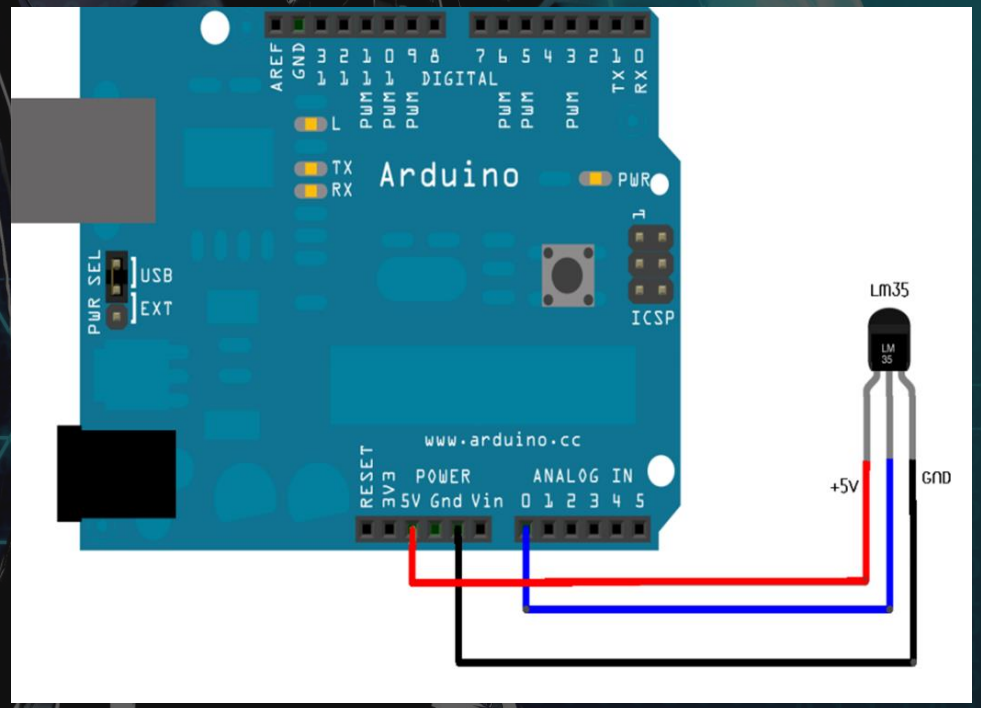
Temperature Sensor LM35

- It is measured by making relation between change in volt and change in temperature
- The voltage output of the LM35 increases 10mV per degree Celsius rise in temperature
- It can measure temperature from -55 degree Celsius to +150 degree Celsius.





• LM35 Connections & Code



sketch_jul17a §

```
int val;
int tempPin = A0;

void setup()
{
  Serial.begin(9600);
}
void loop()
{
  val = analogRead(tempPin);
  float mv = ( val/1024.0)*5000;
  float cel = mv/10;

  Serial.print("TEMPRATURE = ");
  Serial.print(cel);
  Serial.print("*C");
  Serial.println();
  delay(1000);
```

•LDR

❖ Light Dependent Resistor is a light-controlled variable resistor

❖The resistance of a photoresistor decreases with
❖increasing incident light intensity.

❖Resistance increasing when the light remains very dark.

❖**Equation of LDR**

$$V_{out} = V_{in} * \frac{R_2}{R_1+R_2}$$

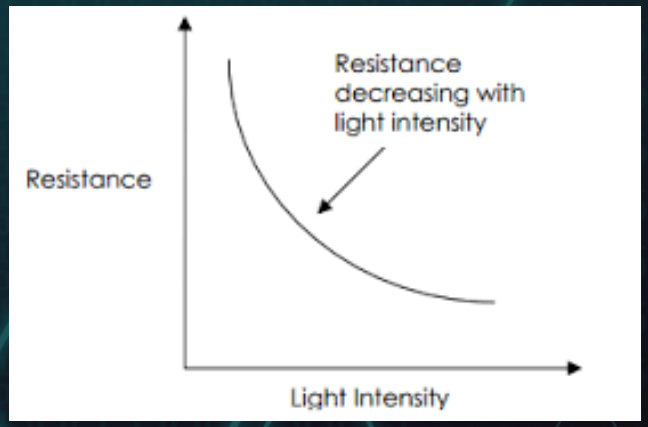
$$(R_1 + R_2) * V_{out} = V_{in} * R_2$$

$$V_{out} * R_1 + V_{out} * R_2 = V_{in} * R_2$$

$$V_{out} * R_1 = V_{in} * R_2 - V_{out} * R_2$$

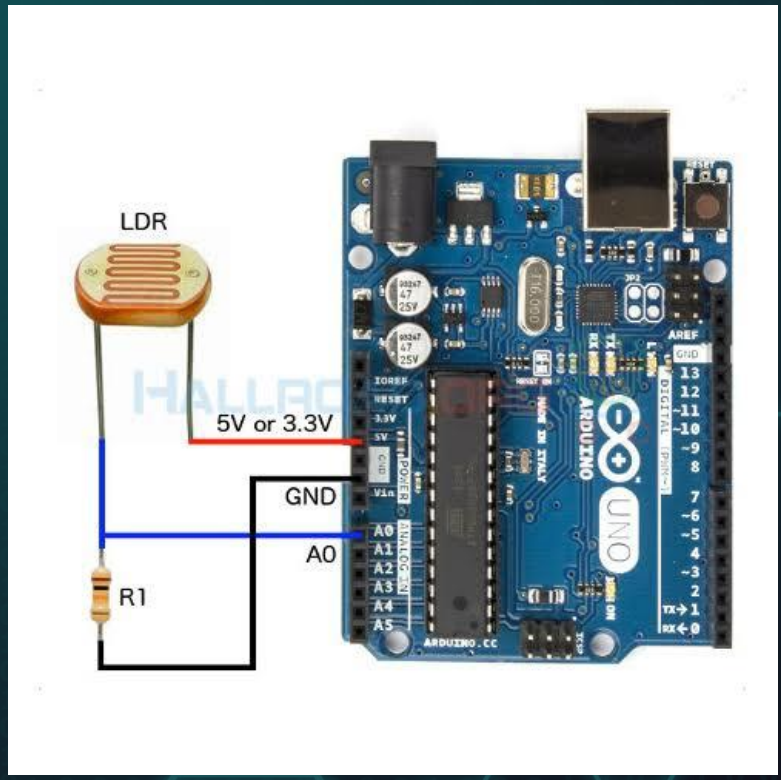
$$V_{out} * R_1 = (V_{in} - V_{out}) * R_2$$

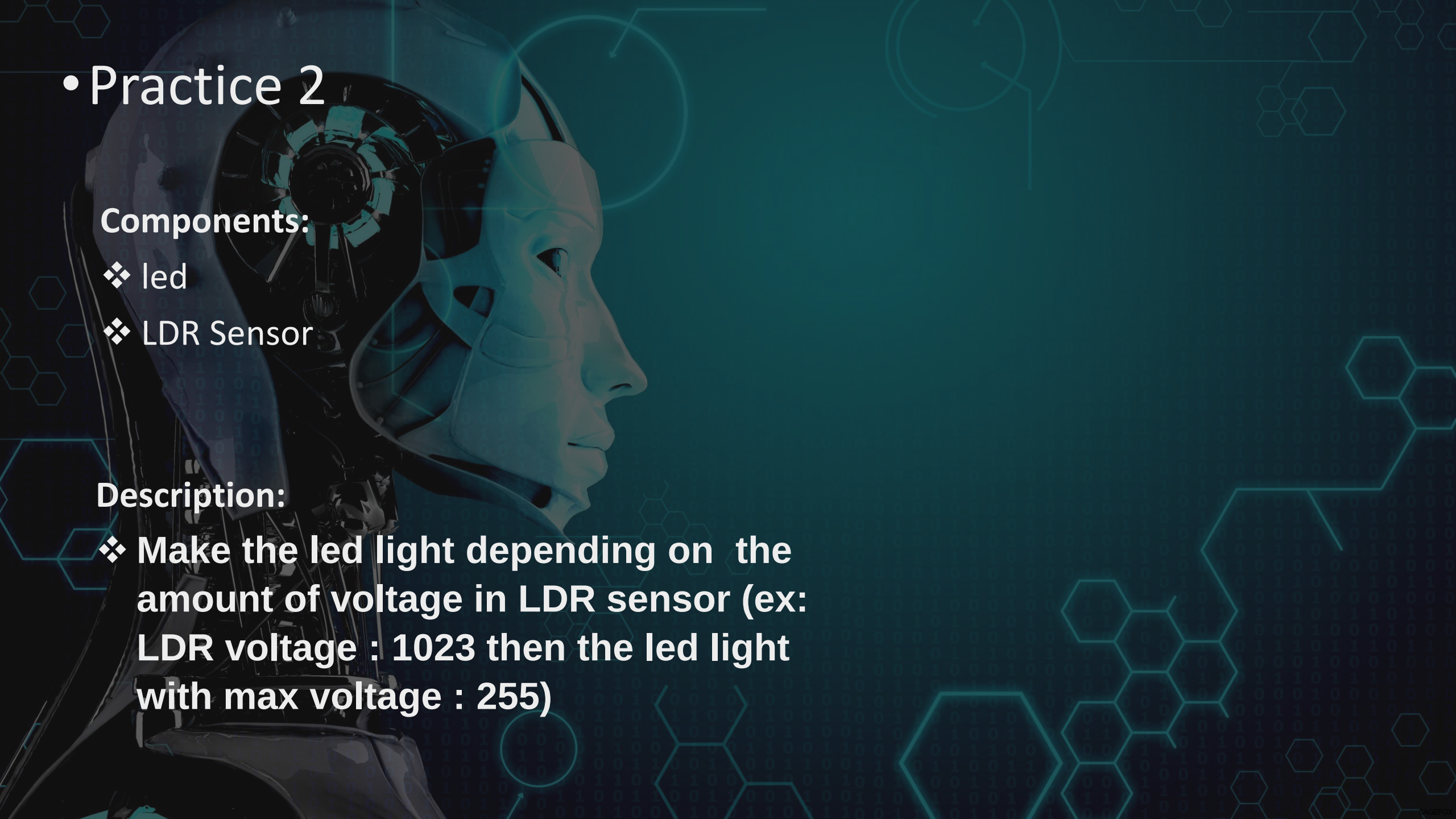
$$R_2 = \frac{V_{out} * R_1}{V_{in} - V_{out}}$$



• LDR Connections & Code

```
int sensorPin = A0;
int sensorValue = 0;
unsigned long res=0;
float value=0.0;
void setup()
{
  pinMode(sensorPin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  sensorValue = analogRead(sensorPin);
   value=(sensorValue/1023.0)*5.0;
   res=(1000*value)/(5-value);
  Serial.println(res);
  delay(10);

}
```





- Practice 2

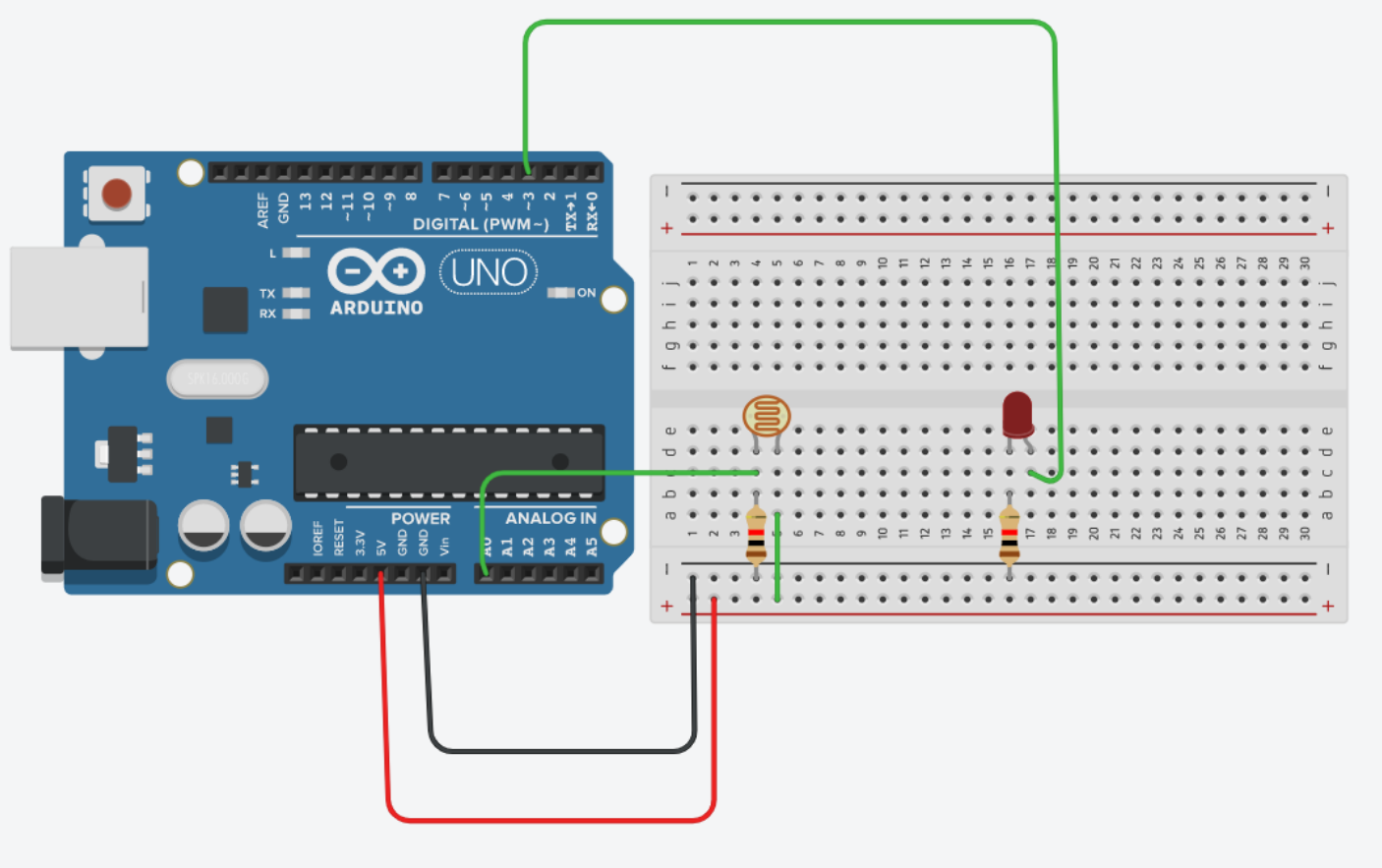
Components:

- led
- LDR Sensor

Description:

- **Make the led light depending on the amount of voltage in LDR sensor (ex: LDR voltage : 1023 then the led light with max voltage : 255)**

• Practice 2 solution



sketch_aug01a | Arduino 1.8.8

File Edit Sketch Tools Help

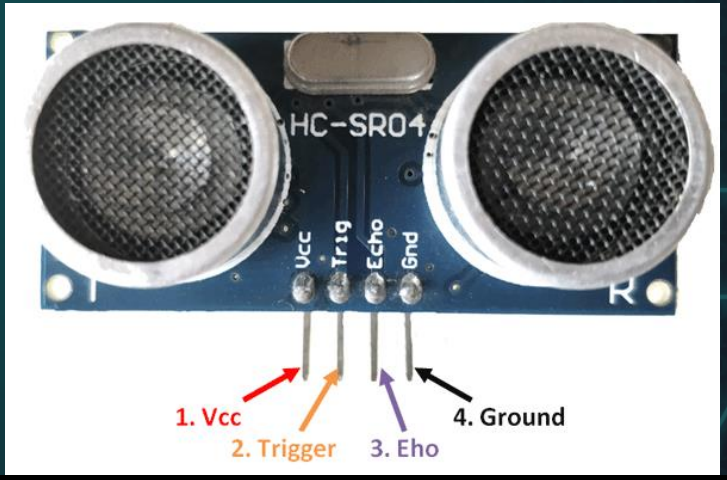
sketch_aug01a §

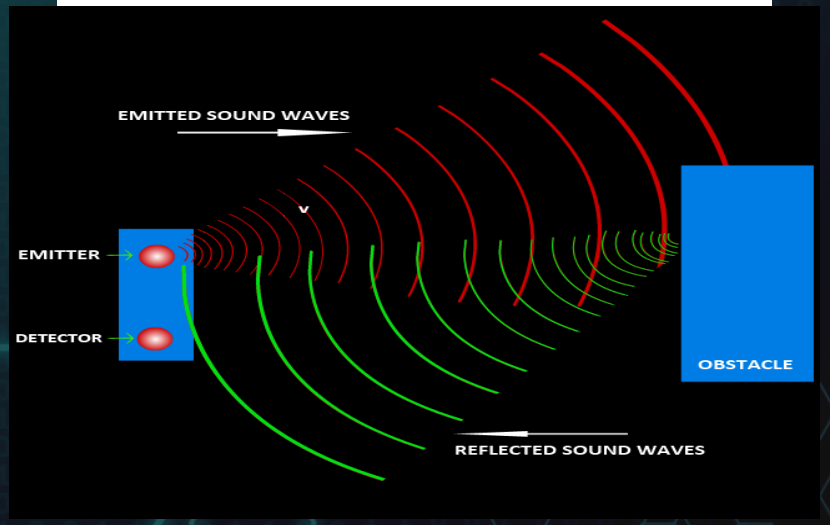
```
int led=3;
int ldr=A0;

void setup()
{
  pinMode(led, OUTPUT);
  pinMode(ldr, INPUT);

}

void loop()
{
  int value=analogRead(ldr);
 int mapvalue= map(value,0,1023,0,255);
 analogWrite(led,mapvalue);

}
```

• Ultrasonic sensor

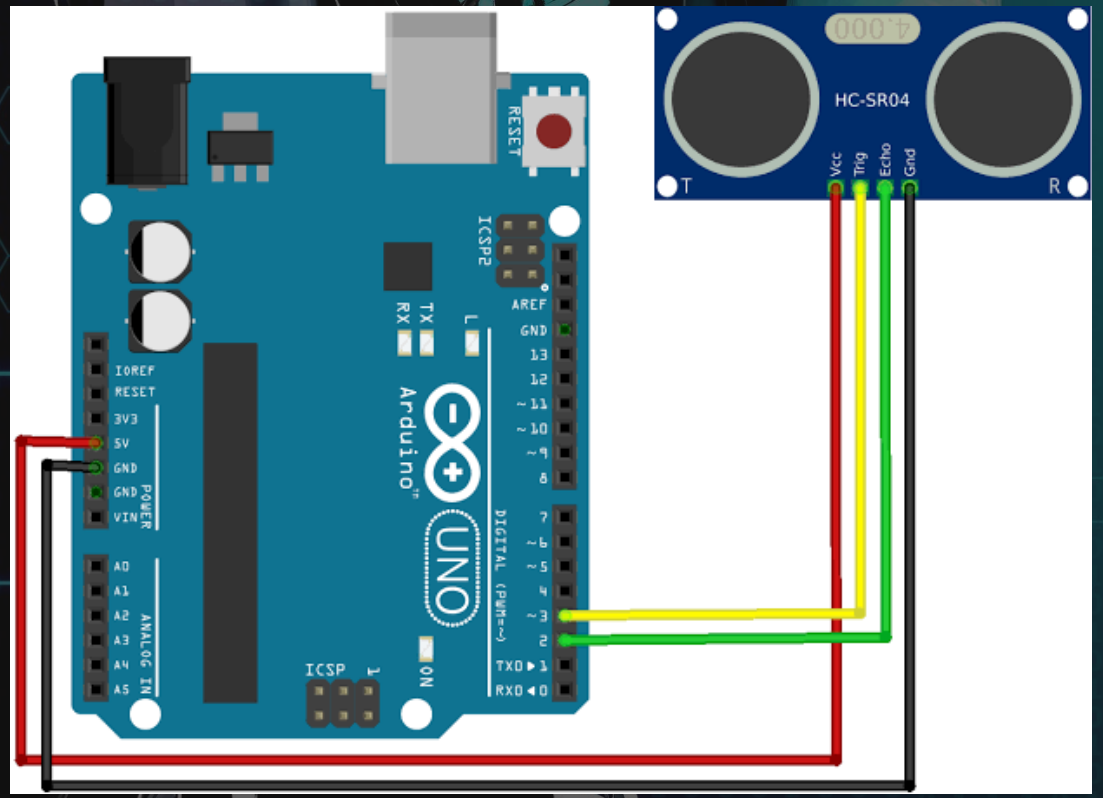
- Ultrasonic sensors work by sending sound wave at a frequency (40KHZ) above the range of human hearing(Range of human between 20KHZ and 40KHZ).
- The sensor send ultrasonic sound waves and convert the reflected sound waves to electrical signal
- The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse.
- Minimum distance =(2cm).
- Max effective range = 400 cm.
- Angle: 15 degrees

→ **D = 0.5 * Time * Speed**

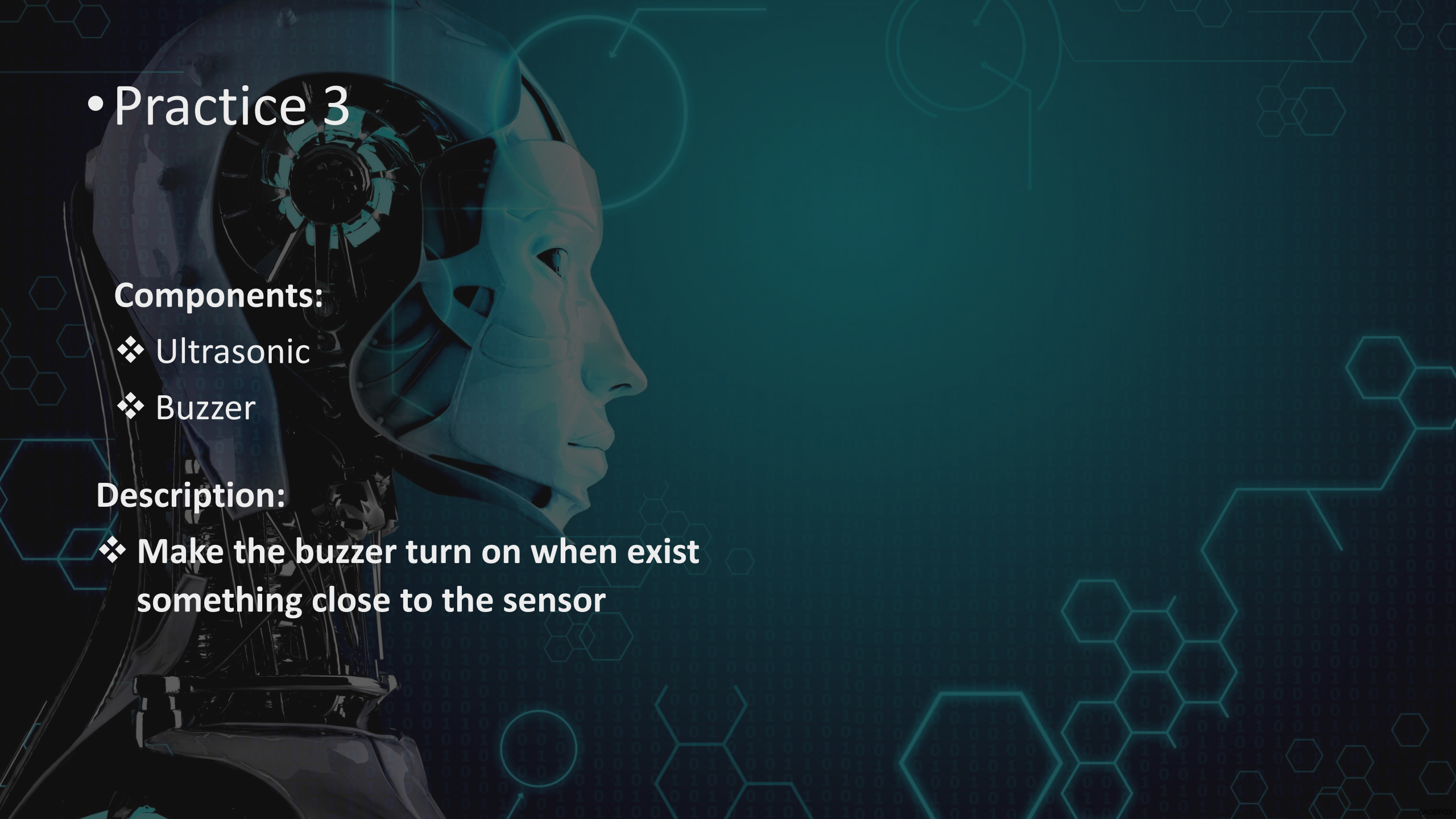




• Connection & Code




```
const int trig = 2;
const int echo = 3;

void setup() {
  Serial.begin(9600);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
}

void loop() {

  float duration, distance;
  digitalWrite(trig, LOW);
  delayMicroseconds(5);
  digitalWrite(trig, HIGH);
  delayMicroseconds(5);
  digitalWrite(trig, LOW);
  duration = pulseIn(echo, HIGH);
  distance = duration/29/2;

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.print("cm");
  Serial.println();

  delay(100);
}
```



- Practice 3

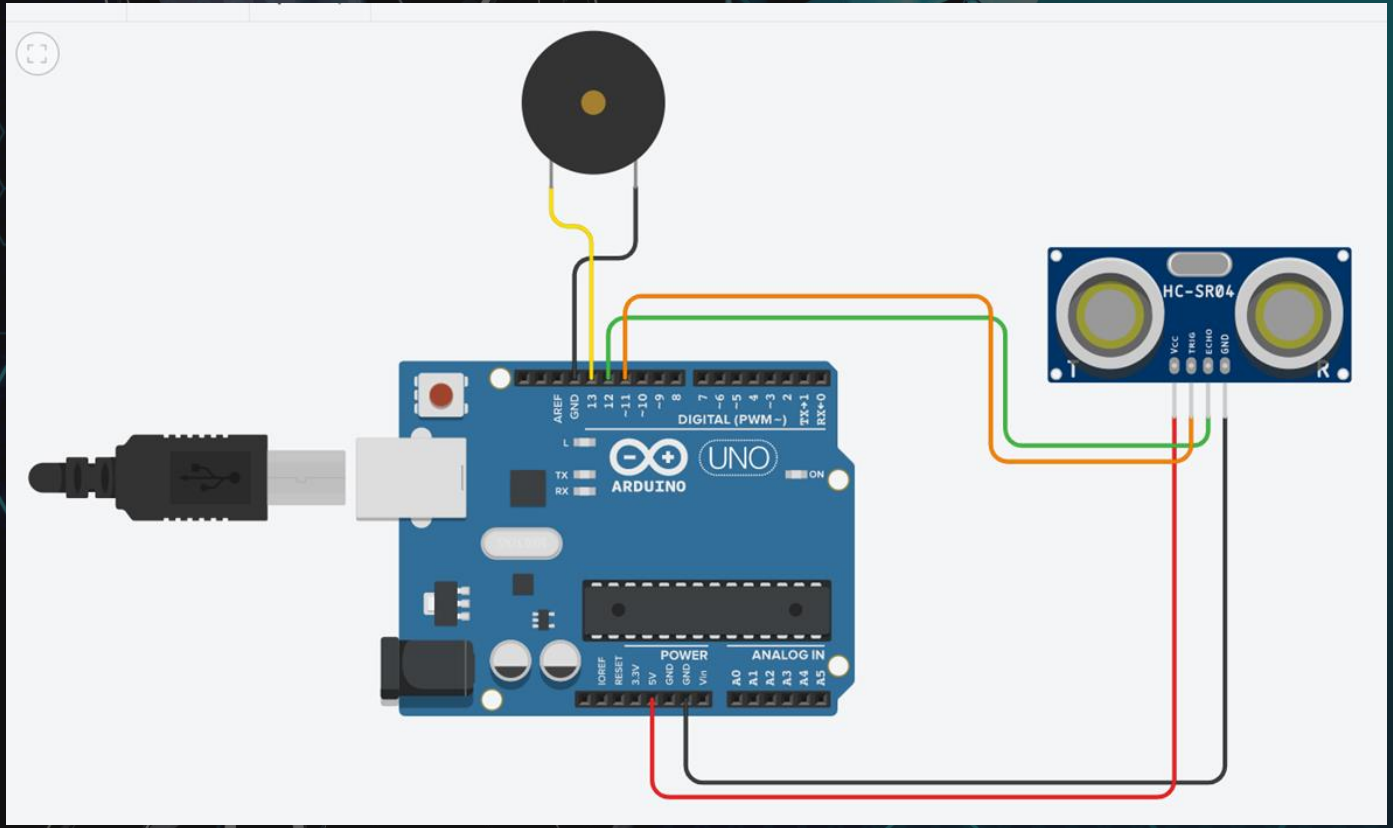
Components:

- ❖ Ultrasonic
- ❖ Buzzer

Description:

- ❖ **Make the buzzer turn on when exist something close to the sensor**

• Practice 3 solution

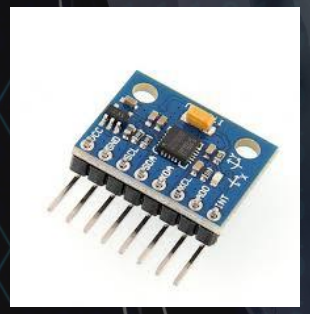


File Edit Sketch Tools Help

sketch_jan06a §

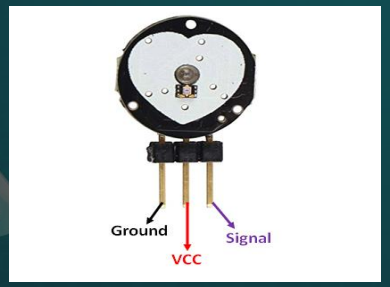
```
int trigPin = 11;
int echoPin = 12;
int buzzer = 13;

void setup() {
  pinMode (buzzer,OUTPUT);
  pinMode (trigPin,OUTPUT);
  pinMode (echoPin,INPUT);
}

void loop() {
  digitalWrite (trigPin,LOW);
  delayMicroseconds (5);
  digitalWrite (trigPin,HIGH);
  delayMicroseconds (10);
  digitalWrite (trigPin,LOW);
  int duration = pulseIn (echoPin,HIGH);
  int distance = duration/2/29;
  if (distance < 50)
  {
    digitalWrite (buzzer, HIGH);
  }
  else
  {
    digitalWrite (buzzer, LOW);
  }
}
```

• Another Sensors



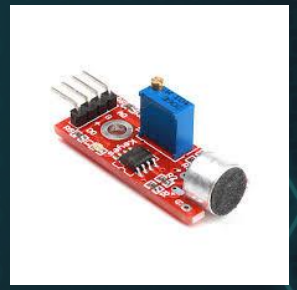
MPU

Flame



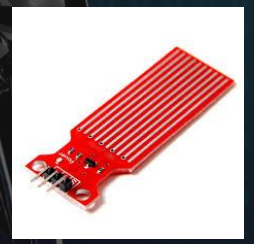
Heart rate

Gas



Sound

Humidity soil



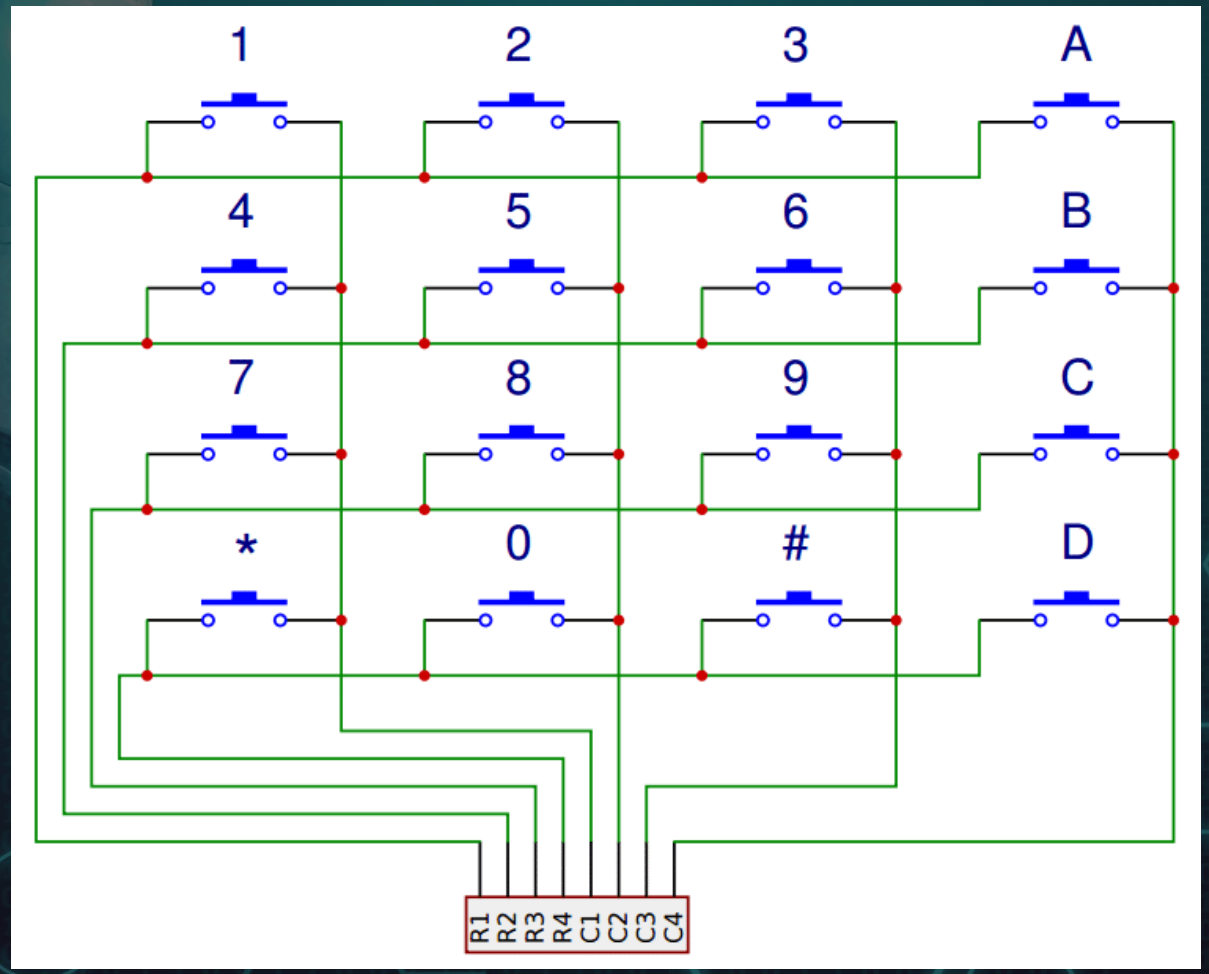
Water

RFID

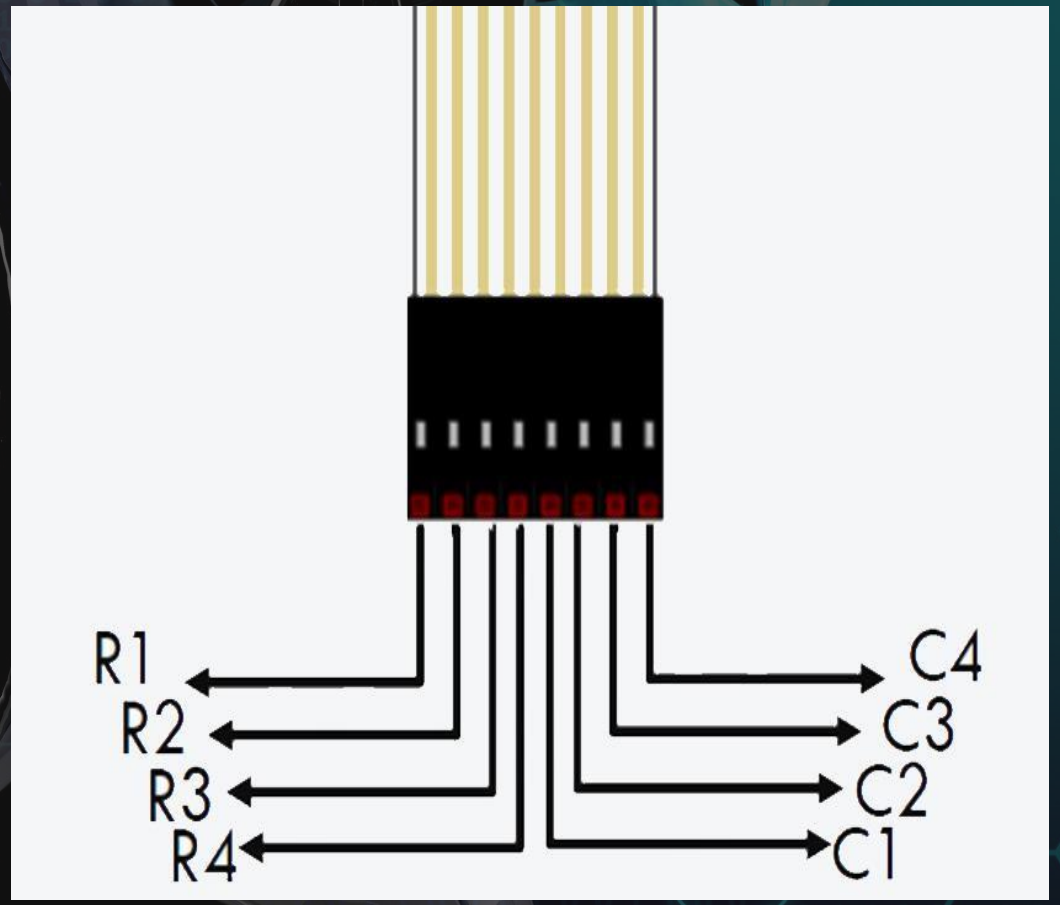
Color

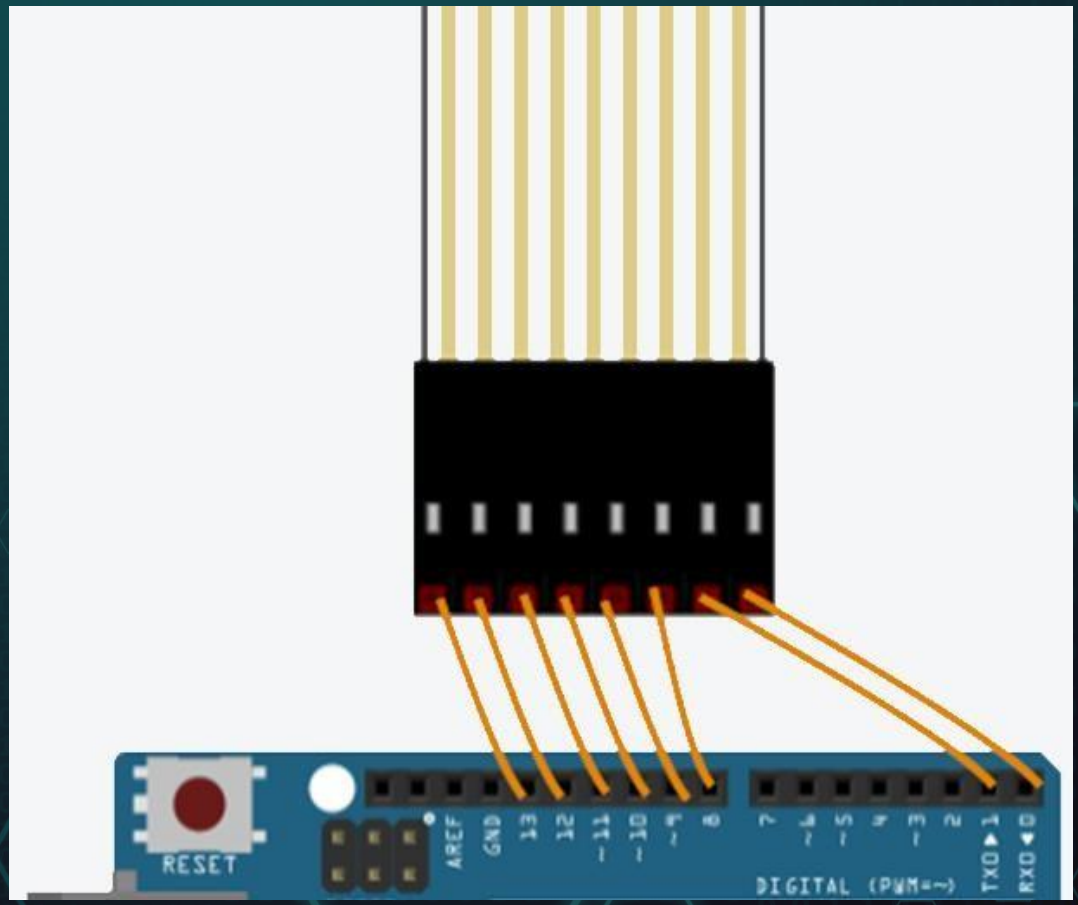
PIR

- Keypad



- Keypad connection



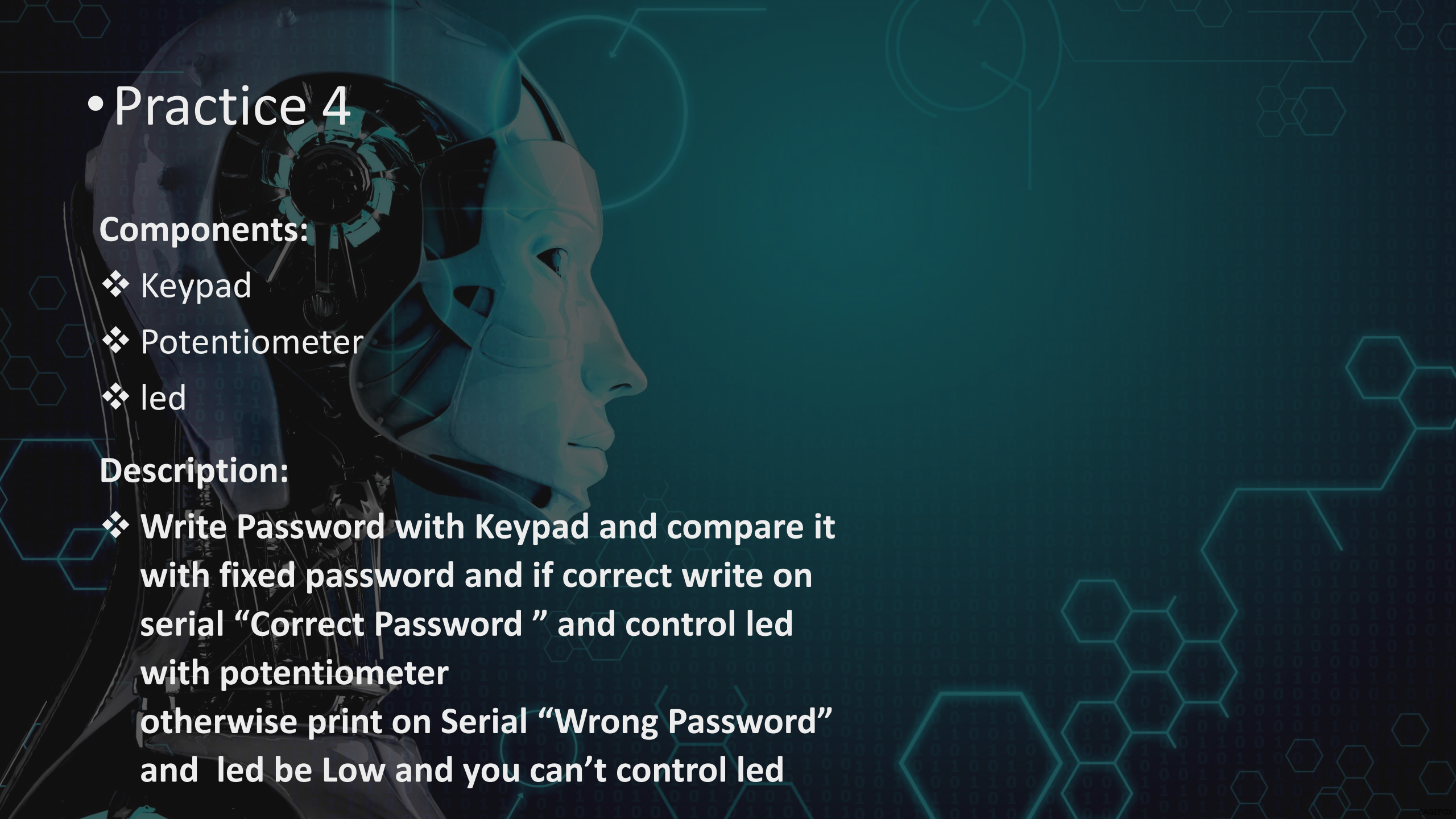


• Keypad functions

- Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
- Char getKey();
  - ret

-Returns the key that is pressed, if any.
-This Function is non-blocking.

- char waitForKey();



- Practice 4

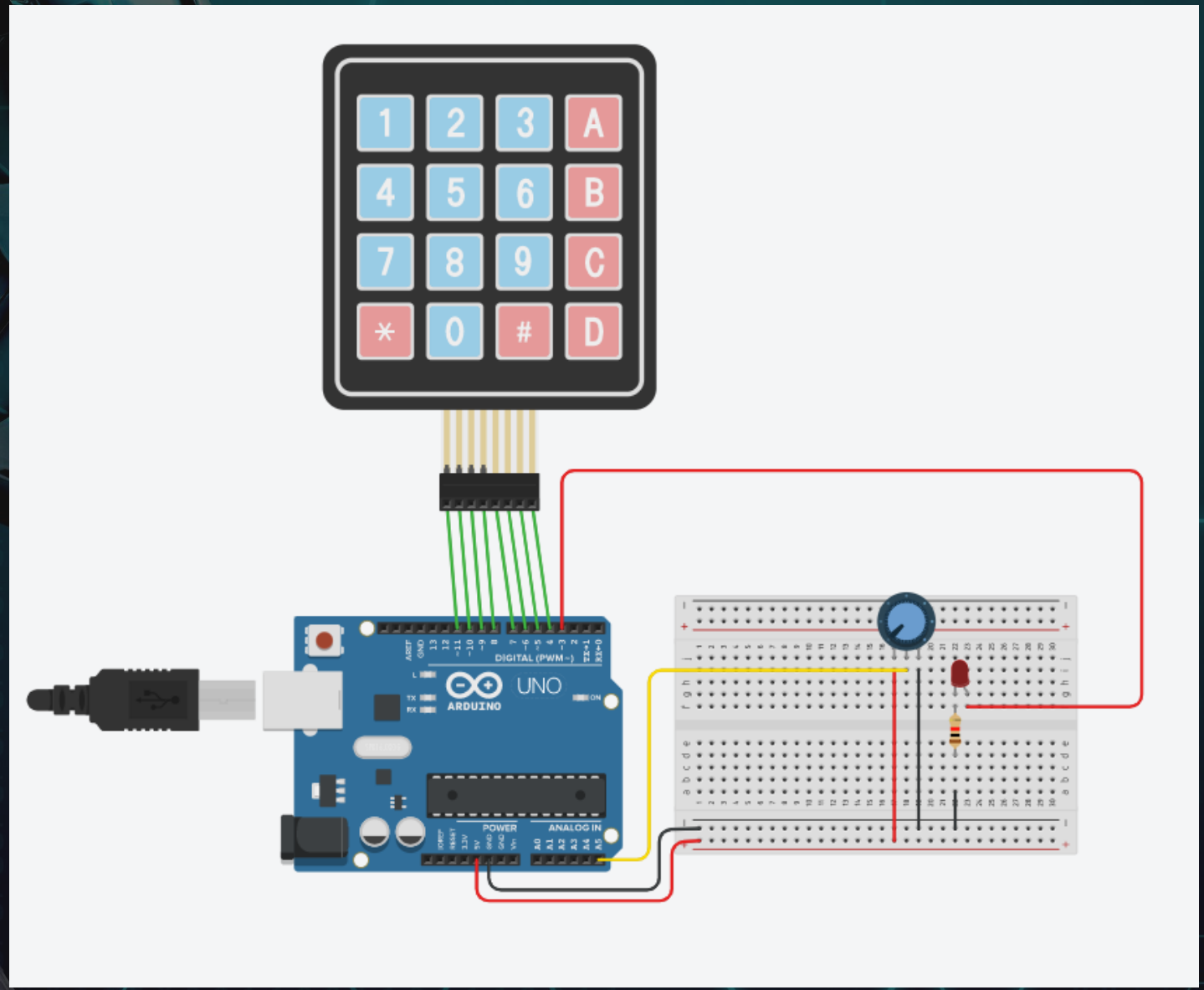
Components:

- ❖ Keypad
- ❖ Potentiometer
- ❖ led

Description:

- ❖ **Write Password with Keypad and compare it with fixed password and if correct write on serial “Correct Password ” and control led with potentiometer**
  otherwise print on Serial “Wrong Password” and led be Low and you can’t control led

- Practice 4



• Practice 4

```
#include <Keypad.h>

#define pot A5
#define led 3


const String fixed_pass="1234";
String entered_pass="";
int i=0;
bool flag=0;

byte rowpins[4]={11,10,9,8};
byte colpins[4]={7,6,5,4};

const char keys[4][4]=
{
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};


Keypad kypd=Keypad(makeKeymap(keys),rowpins,colpins,4,4);

void setup()
{
Serial.begin(9600);
pinMode(led,OUTPUT);
Serial.print("Enter Your Password :");
}
```

```
void loop()
{

  while(i<4)
  {
    char pressed=kypd.waitForKey();
    entered_pass+=pressed;
    Serial.print(pressed);
    i++;
  }


  if(entered_pass==fixed_pass)
  {
    if(flag==0)
    {
    Serial.println("\n Correct Password");
    Serial.println("*YOU CAN CONTROL THE LED*");
    }

    flag=1;
    int pot_reading=analogRead(pot);
    int brightness=map(pot_reading,0,1023,0,255);
    analogWrite(led,brightness);

  }
  else
  {
    Serial.println("\n Worng Password");
    entered_pass="";
    i=0;
    Serial.print("Enter Your Password again :");
  }
```

- The End

THANK YOU