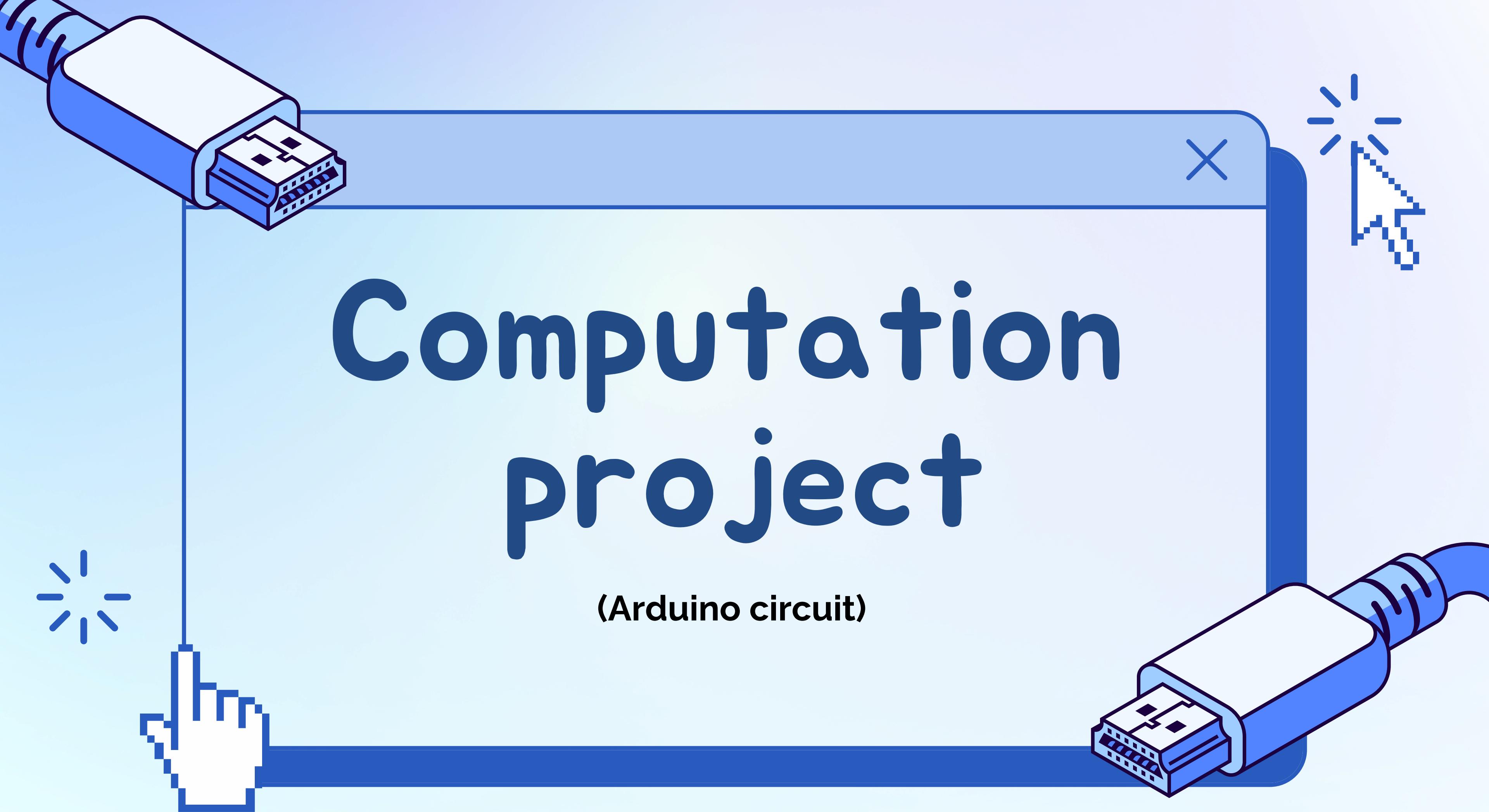


Computation project

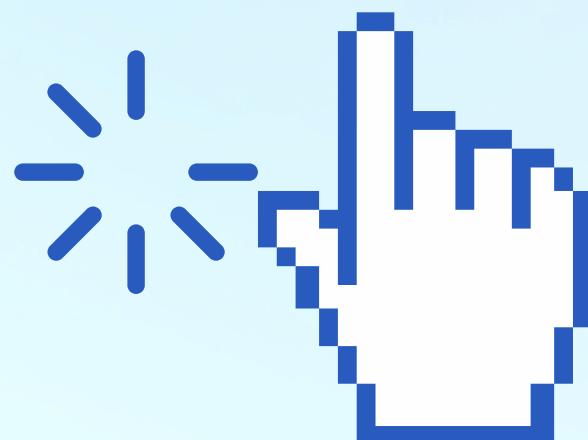
(Arduino circuit)





Introduction

Warm greetings to all present. As we gather here today, I am excited to introduce our project presentation for smart home light system, which aims to address key skills and capitalize on emerging opportunities.

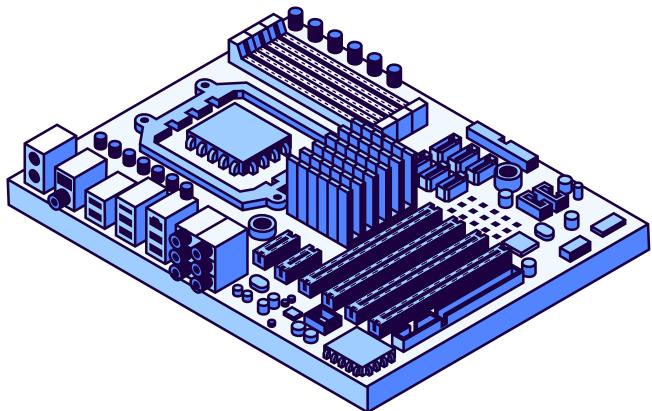


Project concept

our project represents a smart home light system with feedback based capabilities for automated control over a specific parameters including sun light presence and motion in specifc area.

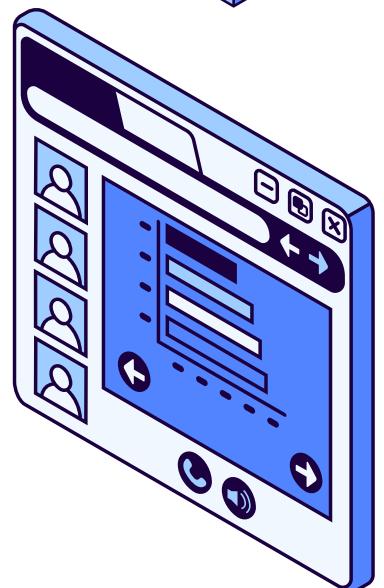


Main sections



Hardware

flexable hardware
components for a flexable
solution



Software

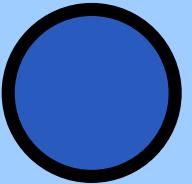
An efficient system
focusing on being user
friendly



Presentation

Targeting the main points in details
to ensure undrestanding without
needing further explanation

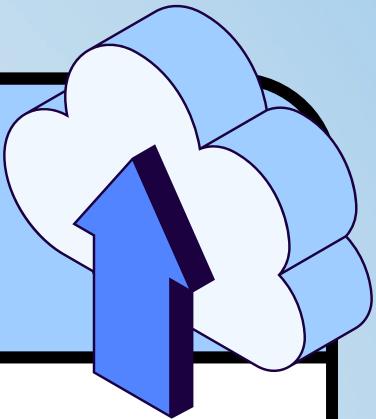




Presentation

- How it was planned?
- What's the Presentation process?
- Learning and choosing the skills needed for presentation





Joe

What It Is The Hardware component?

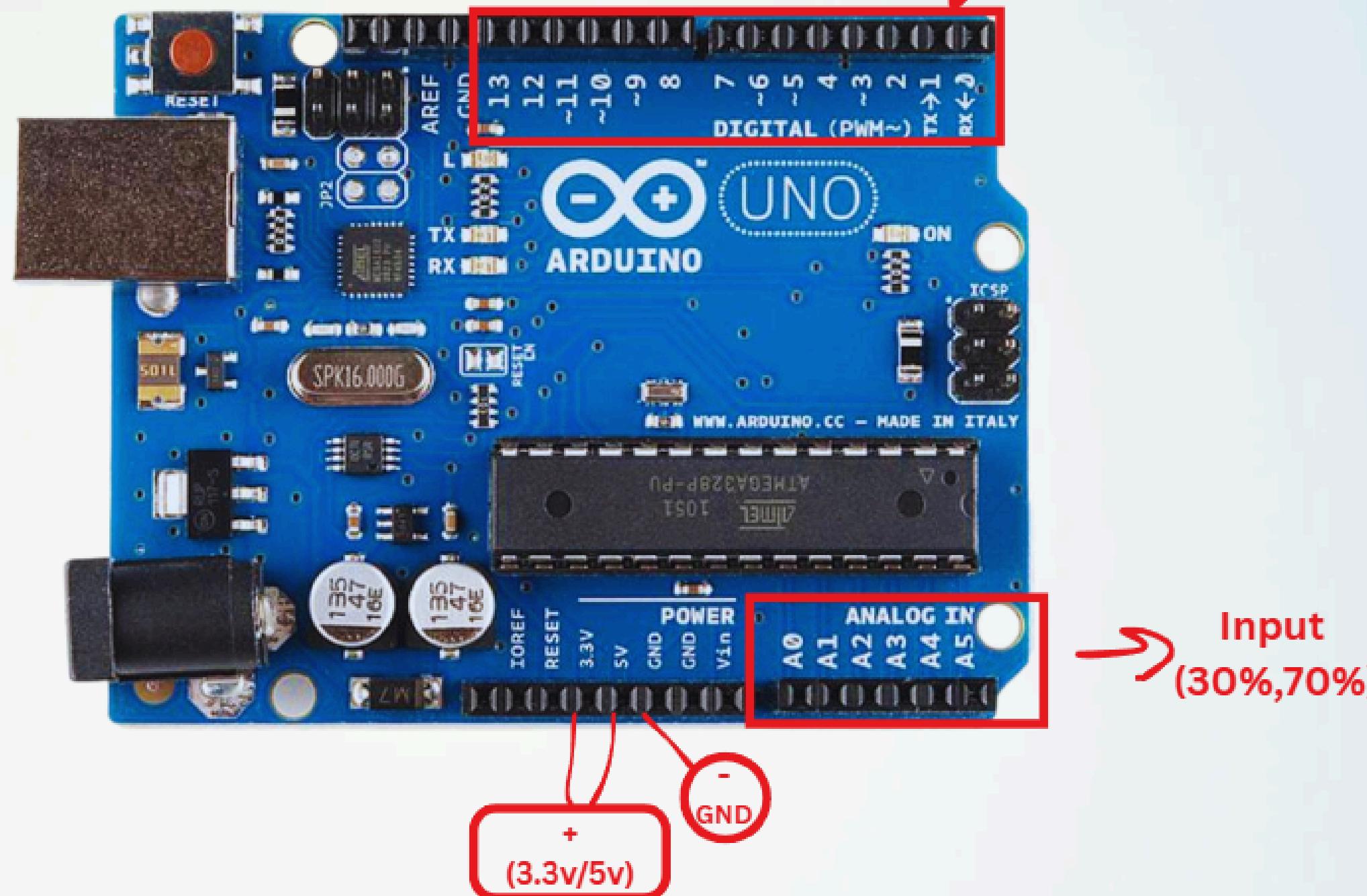
- Arduino UNO
- Bread board
- Light sensor (LDR)
- Ultra-Sonic sensor
- LED lights
- Resestors
- Wiring jumpers





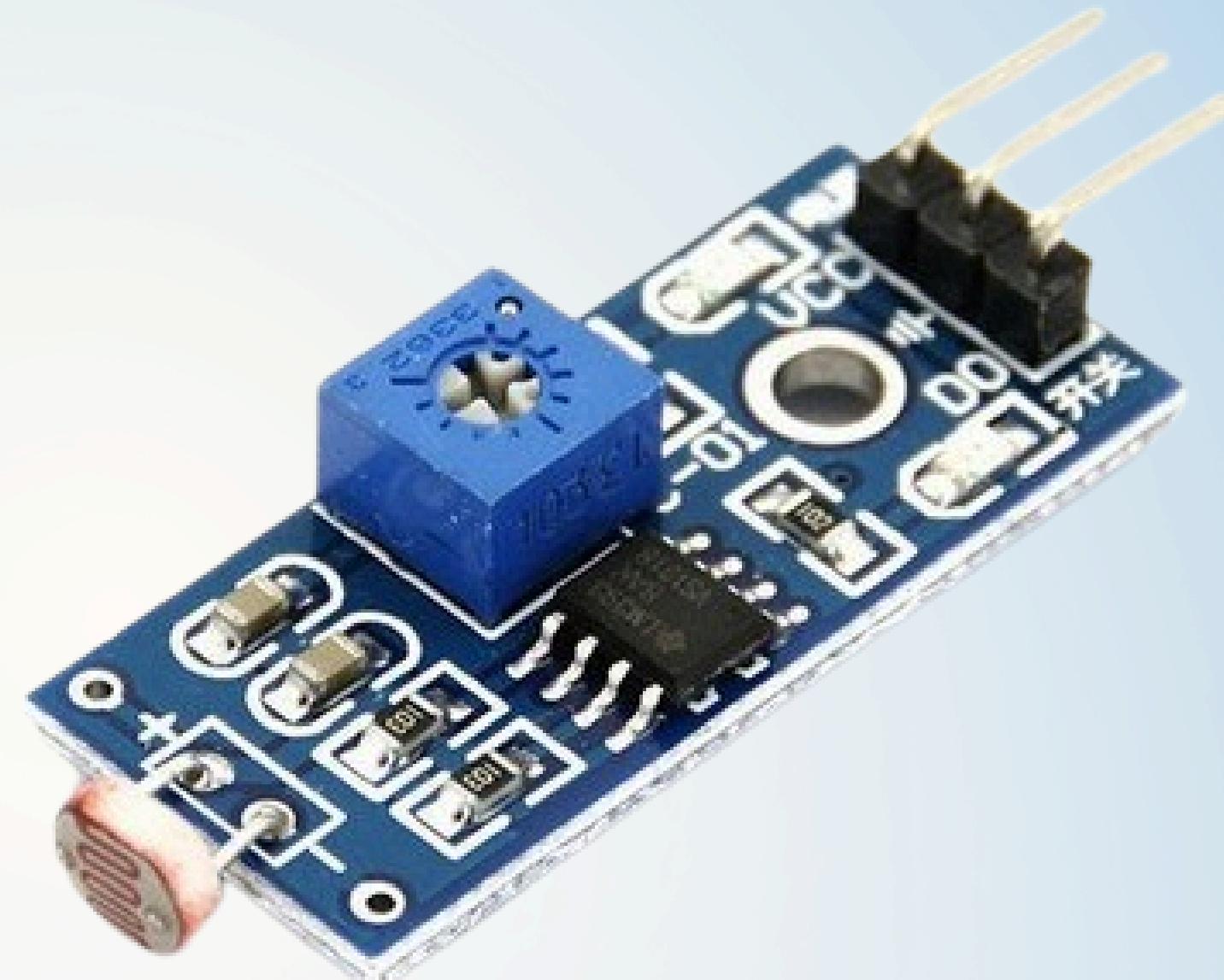
Input (on/off)

OutPut (on/off)
~ (30,40,70)





LDR
(Light Dependent Resistor)



LDR Sensor Module



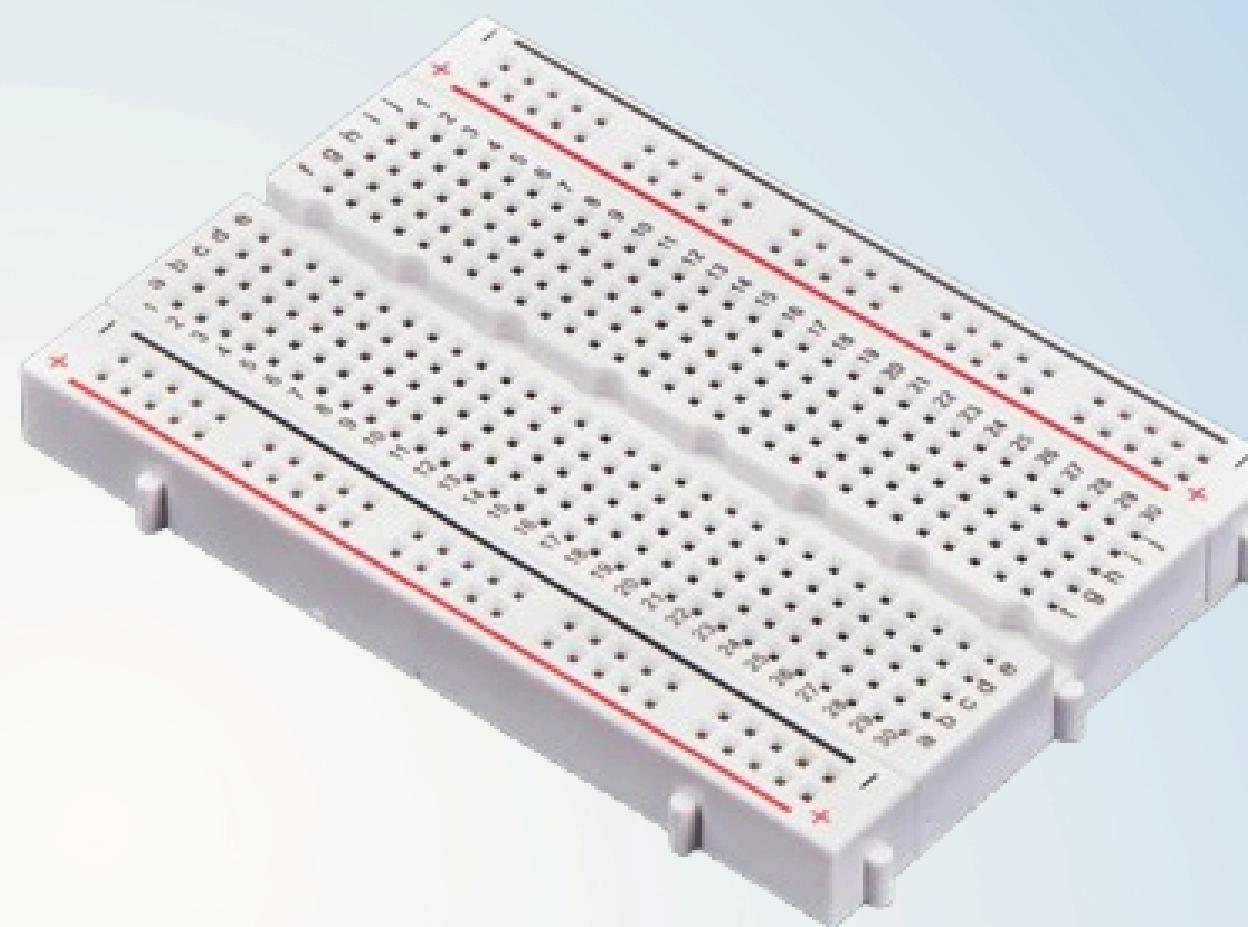
Ultra-Sonic sensor

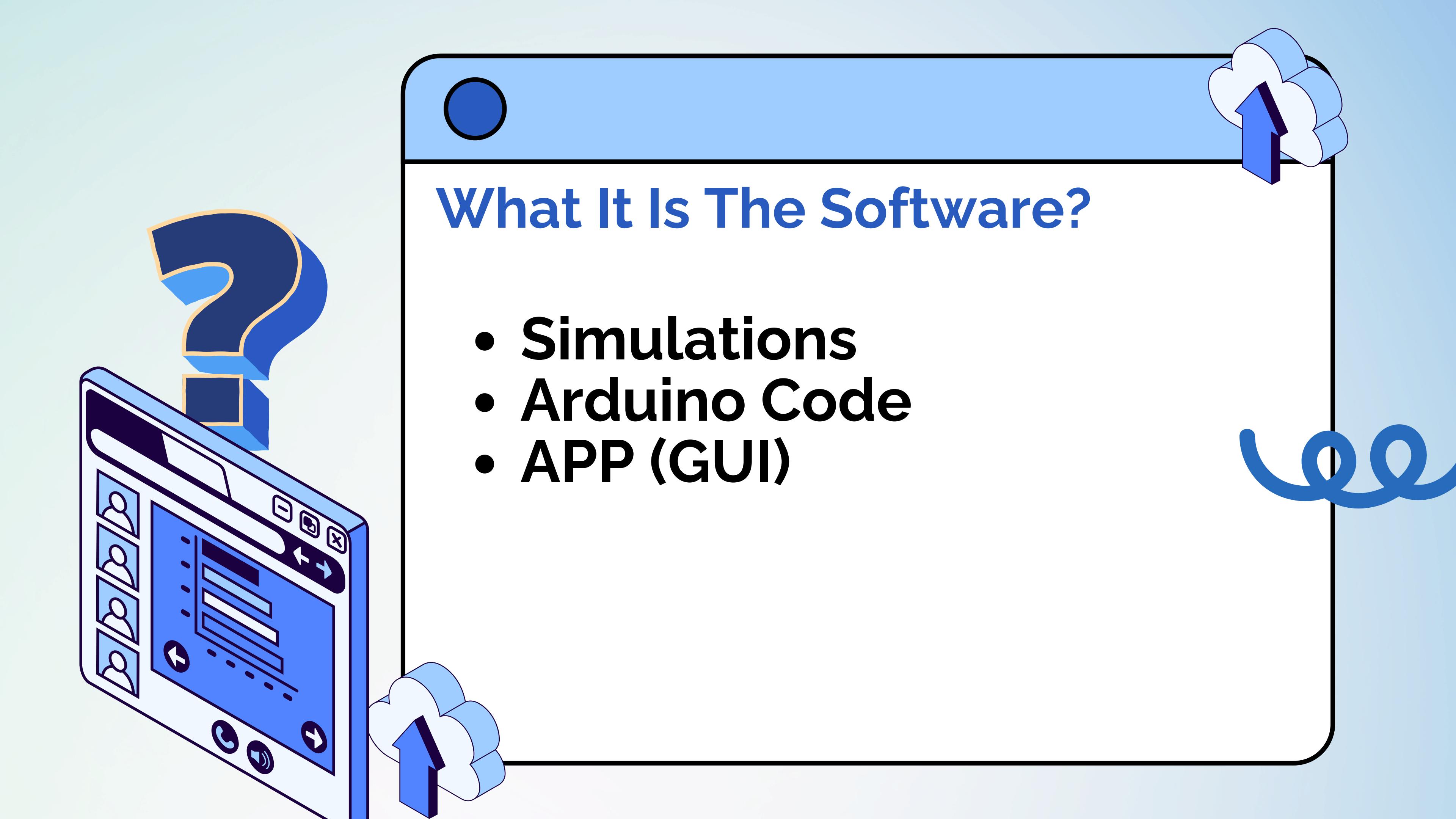
Ultrasonic Sensor HC-SR04



$$\text{Distance (cm)} = \frac{\text{Time (ms)} \times 0.0343}{2}$$

Equation for Ultra-Sonic

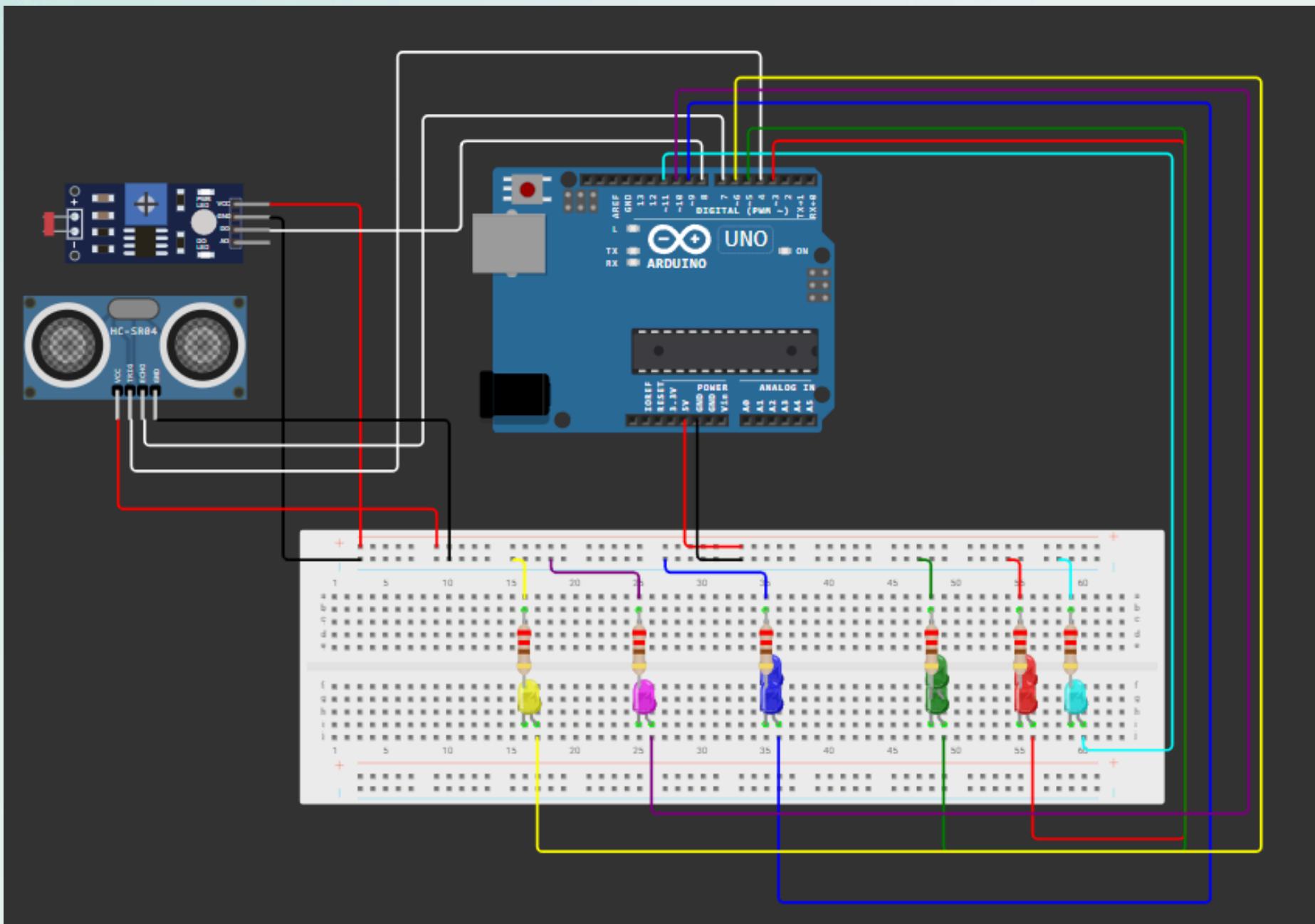




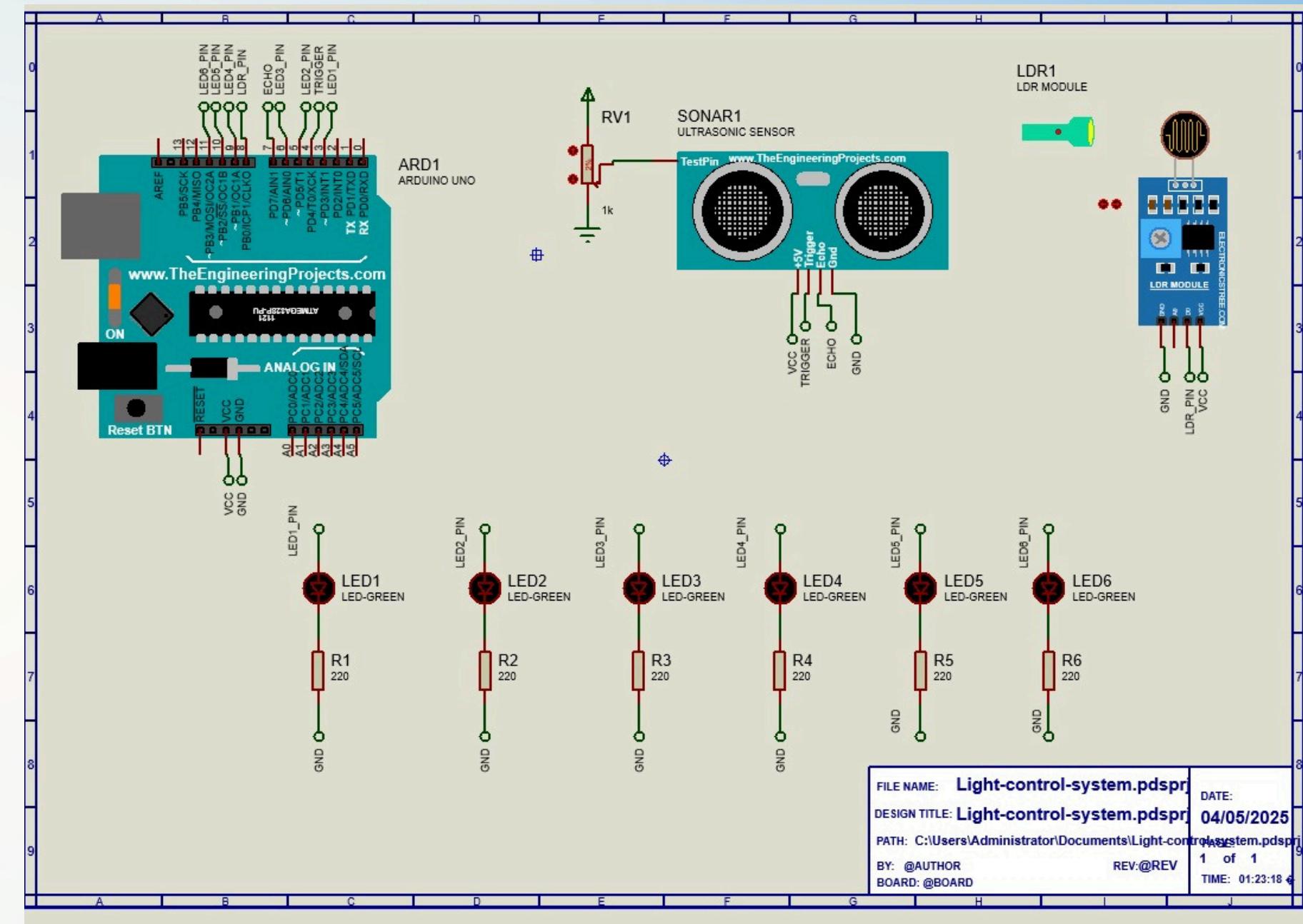
What It Is The Software?

- Simulations
- Arduino Code
- APP (GUI)

Wokwi



Proteus



Setup code

```
1 // LED Pins
2 const int led1 = 3;
3 const int led2 = 5;
4 const int led3 = 6;
5 const int led4 = 9;
6 const int led5 = 10;
7 const int motionLed = 11;
8
9 // Sensor Pins
10 const int ldr = 8;
11 const int trigPin = 4;
12 const int echoPin = 7;
13
14 // Variables
15 unsigned long motionTime = 0;
16 bool motionDetected = false;
17 int ledBrightness[6] = {255, 255, 255, 255, 255, 255};
18 unsigned long motionLedDuration = 5000;
19 bool ledManualOverride[6] = {false, false, false, false, false, false};
20 const int ledPins[6] = {led1, led2, led3, led4, led5, motionLed};
21
22 void setup() {
23 //pin mode and resetting leds to off
24 for (int i = 0; i < 6; i++) {
25 pinMode(ledPins[i], OUTPUT);
26 digitalWrite(ledPins[i], 0);
27 }
28 pinMode(ldr, INPUT);
29 pinMode(trigPin, OUTPUT);
30 pinMode(echoPin, INPUT);
31
32 // Start serial communication
33 Serial.begin(9600);
34 }
35
```

```
35
36 void loop() {
37 int light = digitalRead(ldr);
38 float distance = measureDistance();
39
40 // auto led control
41 for (int i = 0; i < 5; i++) {
42 if (!ledManualOverride[i]) {
43 analogWrite(ledPins[i], (light == HIGH) ? ledBrightness[i] : 0);
44 }
45 }
46
47 // motion detection
48 if (distance > 0 && distance < 20) {
49 motionDetected = true;
50 motionTime = millis();
51 }
52
53 // Motion led
54 if (!ledManualOverride[5]) {
55 if (motionDetected && (millis() - motionTime < motionLedDuration)) {
56 analogWrite(motionLed, ledBrightness[5]);
57 } else {
58 analogWrite(motionLed, 0);
59 motionDetected = false;
60 }
61 }
62
63 handleSerialCommands();
64
65 delay(200);
66 }
67 // End of loop
```

Ultra-Sonic code

```
// ultra sonic code
float measureDistance() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    long duration = pulseIn(echoPin, HIGH, 20000);
    return duration * 0.034 / 2; // equation
}
```

The Function

```
117
118
119 // manual led on function
120 void forceLedOn(int num) {
121     if (num < 1 || num > 6) return;
122     ledManualOverride[num - 1] = true;
123     analogWrite(ledPins[num - 1], ledBrightness[num - 1]); // turn on at current brightness
124     Serial.print("LED "); Serial.print(num); Serial.println(" ON (Manual)");
125 }
126 // manual led off function
127 void forceLedOff(int num) {
128     if (num < 1 || num > 6) return;
129     ledManualOverride[num - 1] = true;
130     analogWrite(ledPins[num - 1], 0); // turn off
131     Serial.print("LED "); Serial.print(num); Serial.println(" OFF (Manual)");
132 }
133 // setting led to auto again
134 void setLedAuto(int num) {
135     if (num < 1 || num > 6) return;
136     ledManualOverride[num - 1] = false;
137     Serial.print("LED "); Serial.print(num); Serial.println(" AUTO Mode");
138 }
139 // changing brightness for led
140 void changeLedBrightness(int num, int b) {
141     if (num < 1 || num > 6) return;
142     b = constrain(b, 0, 255);
143     ledBrightness[num - 1] = b;
144
145     if (ledManualOverride[num - 1]) {
146         analogWrite(ledPins[num - 1], b);
147     }
148
149     Serial.print("LED"); Serial.print(num);
150     Serial.print(" brightness set to "); Serial.println(b);
151 }
152 }
```

Linking code

```
79 // linking arduino with vb
80 void handleSerialCommands() {
81     if (!Serial.available()) return;
82
83     String cmd = Serial.readStringUntil('\n');
84     cmd.trim();
85
86     if (cmd == "Hello") {
87         Serial.println("Hello");
88     }
89     else if (cmd.startsWith("O")) {
90         int n = cmd.substring(1).toInt(); forceLedOn(n);
91     }
92     else if (cmd.startsWith("F")) {
93         int n = cmd.substring(1).toInt(); forceLedOff(n);
94     }
95     else if (cmd.startsWith("A")) {
96         int n = cmd.substring(1).toInt(); setLedAuto(n);
97     }
98     else if (cmd.startsWith("B")) {
99         int sp = cmd.indexOf(' ');
100        if (sp != -1) {
101            int n = cmd.substring(1, sp).toInt();
102            int b = cmd.substring(sp + 1).toInt();
103            changeLedBrightness(n, b);
104        }
105    }
106    else if (cmd == "GET_LDR_DATA") {
107        Serial.println(digitalRead(ldr) == HIGH ? "HIGH" : "LOW");
108    }
109    else if (cmd == "GET_MOTION_DATA") {
110        Serial.println(motionDetected ? "Motion Detected" : "No Motion");
111    }
112    else if (cmd.startsWith("T")) {
113        motionLedDuration = max(1000, cmd.substring(1).toInt() * 1000UL);
114        Serial.print("Duration:"); Serial.println(motionLedDuration / 1000);
115    }
116}
117
```

Control APP

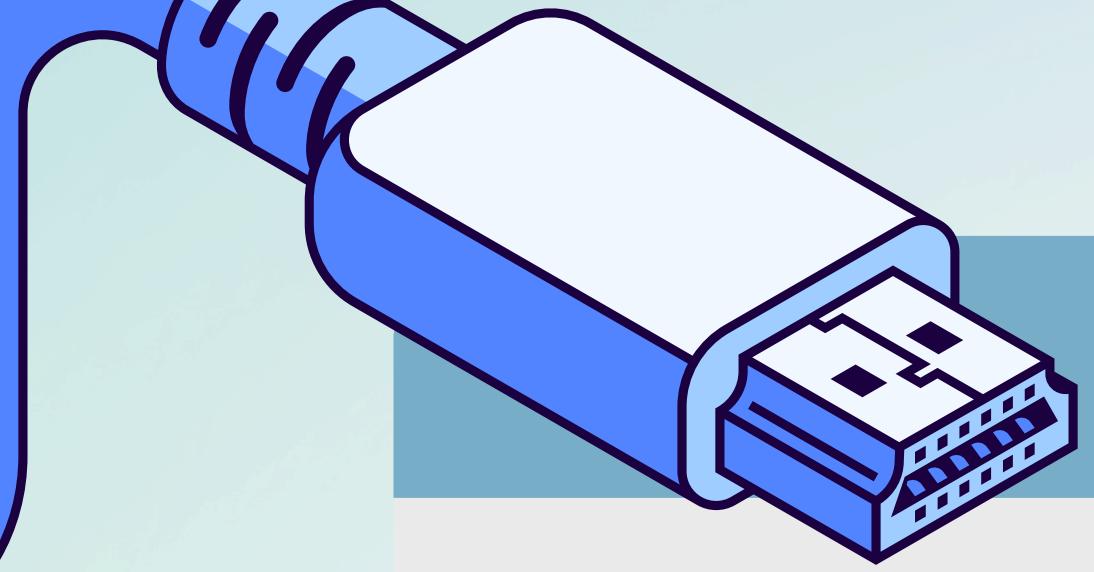
The image shows a mobile application interface for controlling smart lights across various rooms. On the left, a sidebar lists rooms and a statistics section. The main area displays seven light controls, each with a bulb icon, a room name, an 'Auto On - Off' switch, and a vertical line with a circular endpoint.

- Home:** Light off, Auto On - Off switch greyed out, vertical line ends open.
- Living room:** Light off, Auto On - Off switch greyed out, vertical line ends open.
- Bed room:** Light off, Auto On - Off switch greyed out, vertical line ends open.
- Bed room:** Light off, Auto On - Off switch greyed out, vertical line ends open.
- Bathroom:** Light off, Auto On - Off switch greyed out, vertical line ends open.
- Kitchen:** Light off, Auto On - Off switch greyed out, vertical line ends open.
- stairs:** Light off, Auto On - Off switch greyed out, vertical line ends open.

Statistics:

- Current: 0.03 A
- Power: 0.16 W

C



Thank You

