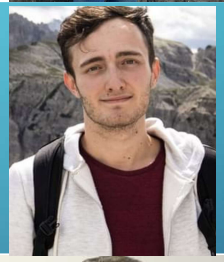
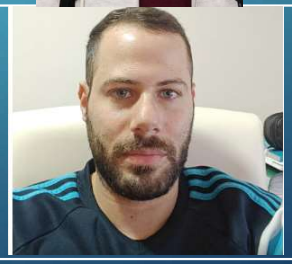
ELECTRONIC INSTRUMENTATION

LECTURER DR. SAMUEL KOSOLAPOV

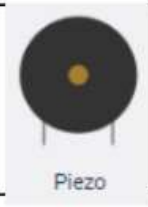
HOMEWORK 01 - ALARM SYSTEM HARDWARE & TINKERCAD SIMULATION

	ID	Name	Photo of the student
Student #1	**316352210**	**Omri pony**	
Student #2	**316146836**	**Steve Reznikov**	
Student #3	**313601866**	**Or Salganik**	

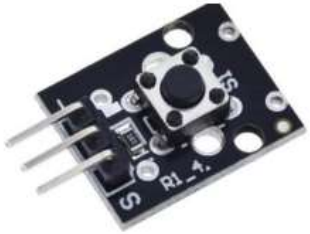
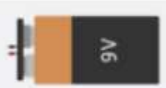
Link to TinkerCad:

https://www.tinkercad.com/things/1RQUUYt9YRy-hw0116protocoloftesttest6simulation

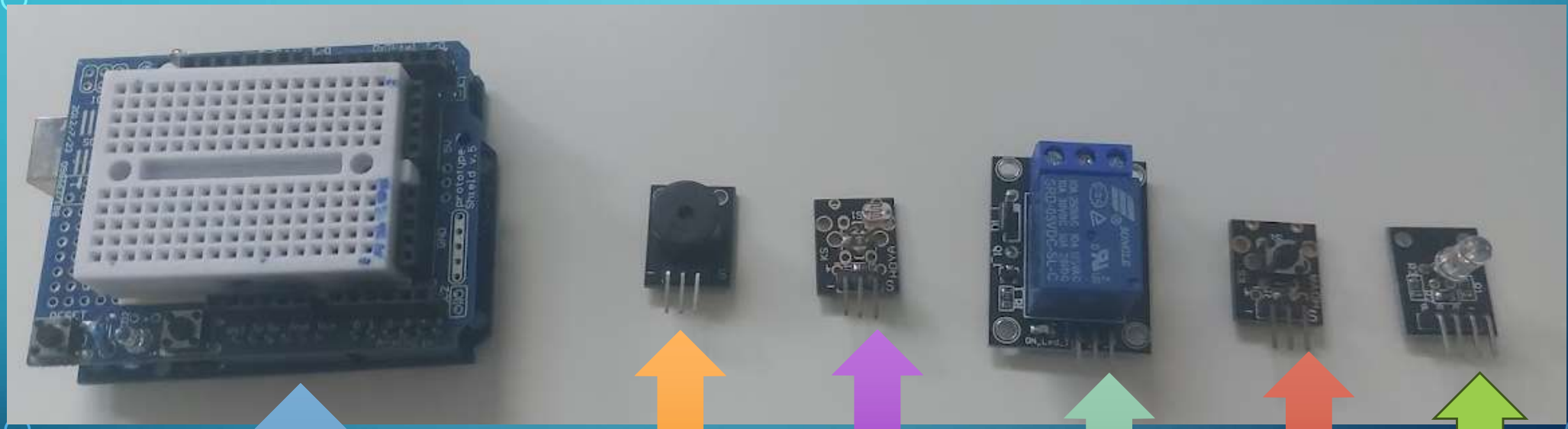
OUR PROJECT CONTAIN THE FOLLOWING COMPONENTS:

Table 1. Alarm System. Components from Arduino 37 in 1 Sensor Kit and equivalent TinkerCad components			
1	Arduino KY-018 Photo resistor module → TinkerCad : “Photoresistor”		
2	Internal LED 13 (Seen on the Arduino board under pin13) Hardware: Arduino KY-016 3-color LED module (Use RED LED) Simulation: “Light Bulb” (When powered ON seen as YELLOW)		
3	Arduino KY-019 5V relay module → “Relay SPDT” (without transistor and protections)		
4	Arduino KY-006 Passive buzzer module → Piezo (Passive Buzzer !!!!)		

OTHER COMPONENTS ARE:

Table 1. Alarm System. Components from Arduino 37 in 1 Sensor Kit and equivalent TinkerCad components		
5	KY-004 Button Key Switch Module → "PushButton" (resistor must be added in case of need)	  Pushbutton
6	Resistors (In case of need select number of resistors and their value in accordance to YOUR needs)	
7	5V for relay with hardware and External 9V Battery for the Light Bulb	 9V Battery
8	"**Breadboard Mini**" – it is a challenge to arrange ALL Pins of ALL the components 1-7 on the Mini Breadboard!!!	 Breadboard Mini
9	Arduino UNO – Pins of Arduino UNO are to be connected to the Breadboard Mini by using reasonably arranged color wires	 Arduino Uno R3

THESE ARE THE REAL WORLD COMPONENTS – BEFORE ASSEMBLING THEM ON THE MINI-BREADBOARD



Arduino uno R3 with prototype shield

Buzzer module – ky006

photoresistor module –ky018

relay module – ky019

button module – ky004

bulb module –ky016

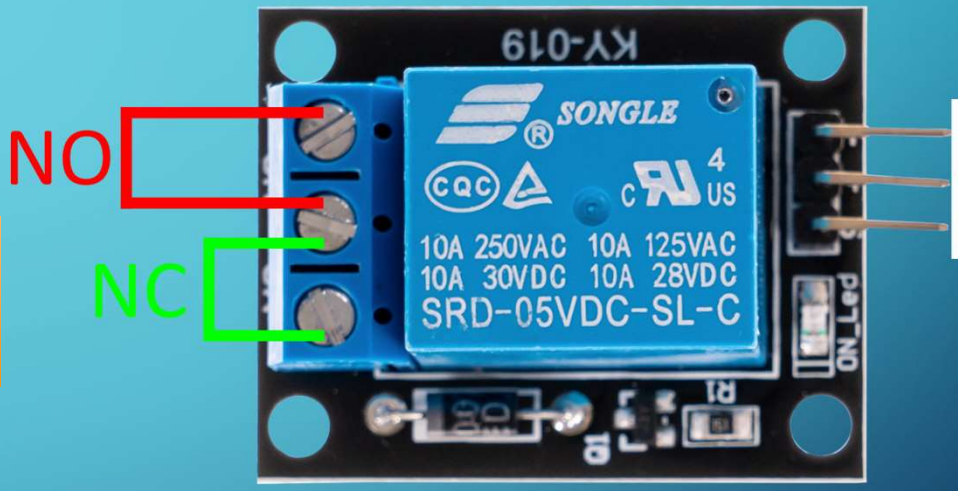
The relay module - KY-019

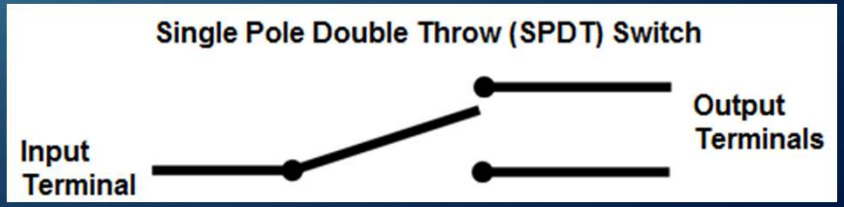
The KY-019 relay module has two contacts, normally open (NO) and normally closed (NC) that are operated by pulling the signal pin **HIGH**.

This allows low voltages devices such as the Arduino to safely control high voltage devices. The high voltage side of the relay can operate up to 250V AC or up to 30V DC.

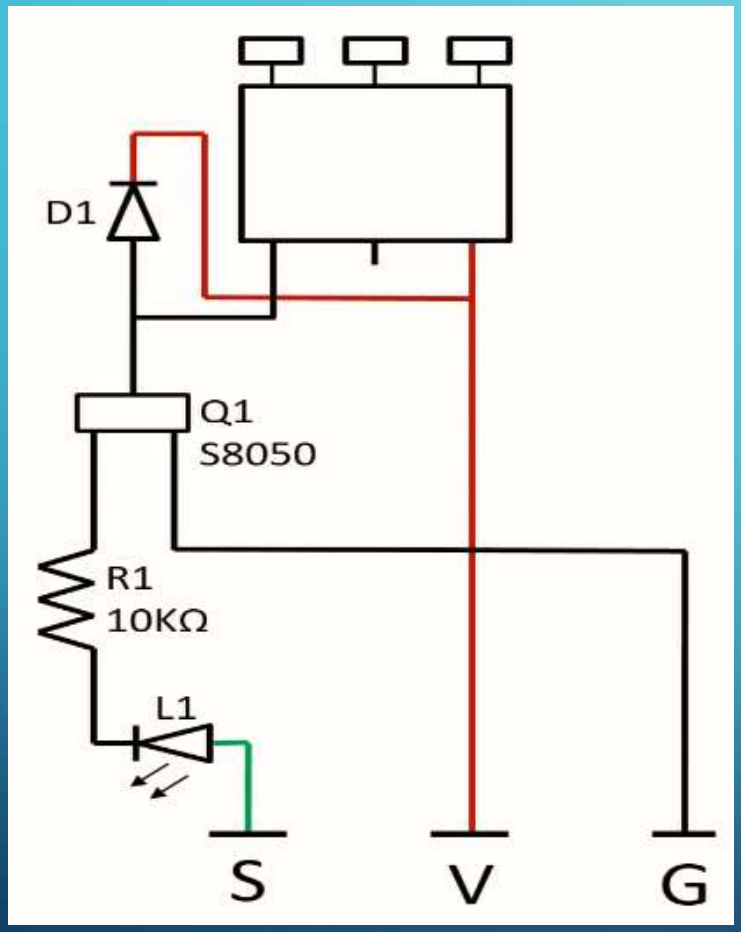
Normally closed will pass current until the signal line is activated. Normally Open will not pass current until the signal line is activated. The common terminal is shared between the two contacts.

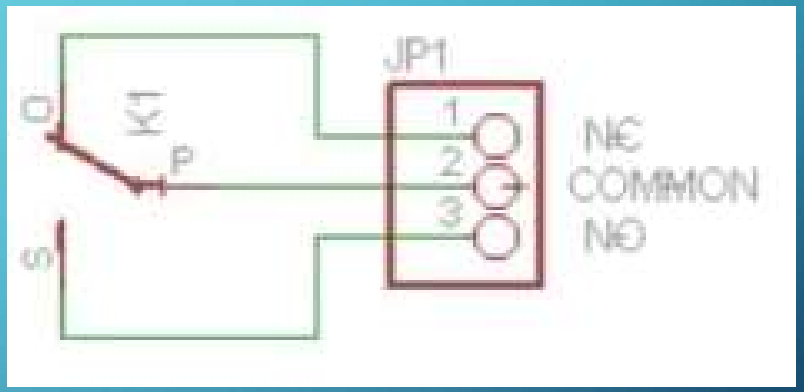
This relay is called a **SPDT** relay – it means "single pole double throw" – it means that the switch has only a single input and can connect to and switch between 2 outputs.





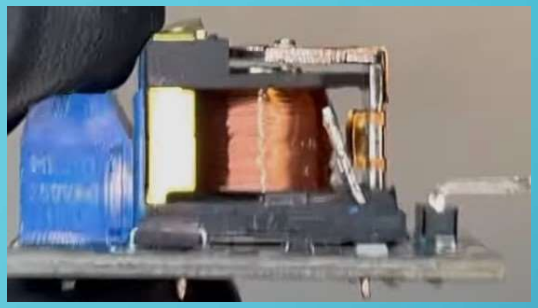
The relay module - KY-019 - internal structure



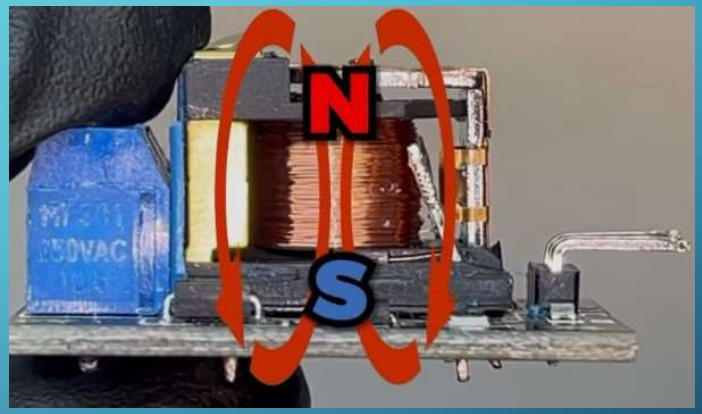


The relay module - KY-019– physical operation explanation

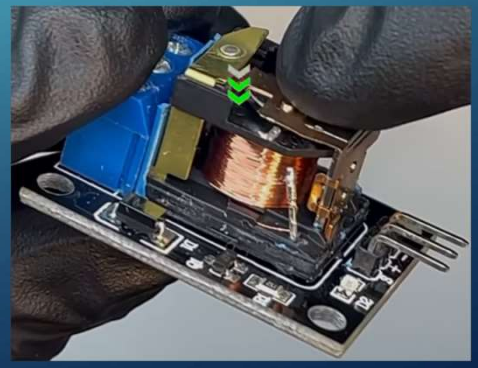
Inside the relay there's a coil



The coil creates a **magnetic field** when we move current through it



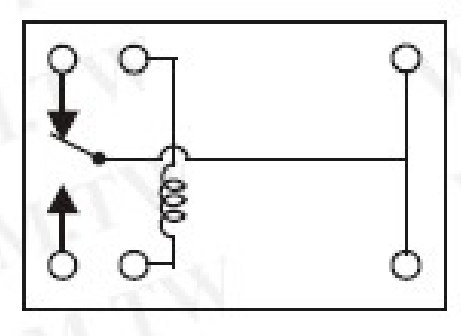
The **magnetic field toggles a pole which can connect or disconnect two circuits**



Tinkercad relay module - LU-5-R

In the Tinkercad we use a different relay module – the LU-5-R

available in various configurations, including SPDT (Single Pole Double Throw) and DPDT (Double Pole Double Throw), offering flexibility for different switching needs.



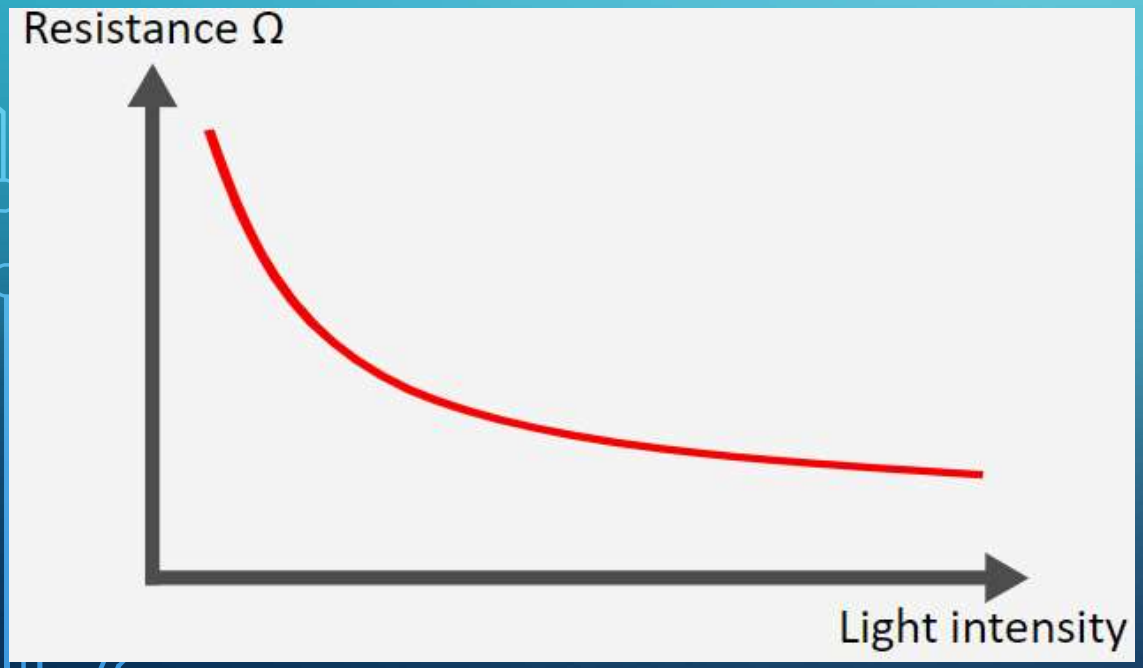
Supports various coil voltages, ranging from 5V to 24V DC, allowing compatibility with diverse power sources. In addition-Can handle currents up to 7A

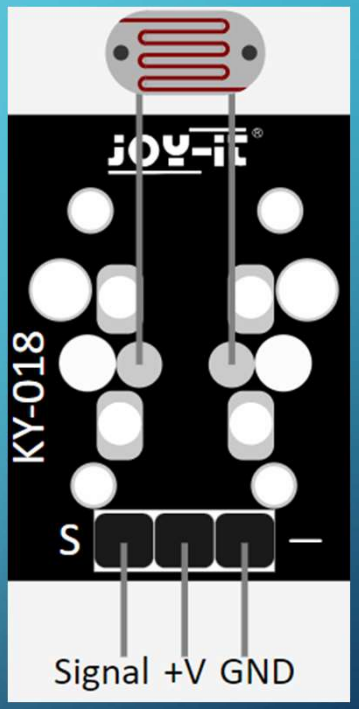
Features optocoupler isolation between the control input and the relay output, enhancing safety and preventing noise interference.

The photoresistor module - KY-018

This module contains an LDR resistor whose **resistance value decreases with brighter surroundings**.

LDR= light-dependent resistor

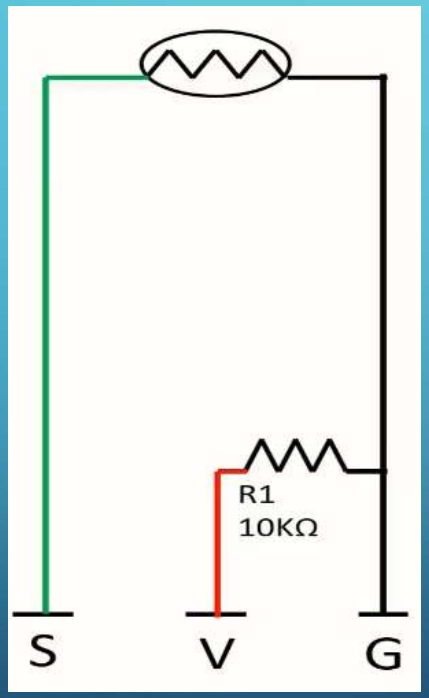


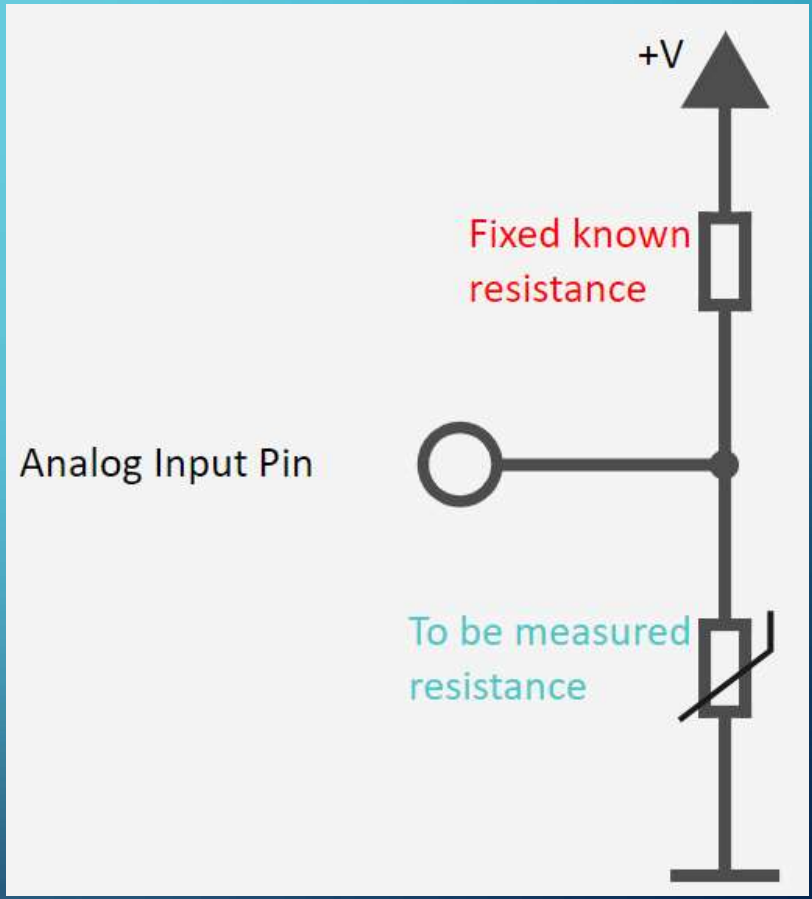


The **VCC** pin is used to provide power to the module.
The **GND** pin is connected to the ground of the Arduino.
The **signal** pin is the output pin that provides an analog voltage signal to the Arduino proportional to the intensity of light falling on the photoresistor.

The photoresistor module - KY-018 - working principle

This resistance can be determined using a voltage divider, where a known voltage is divided across a known (10 kΩ) and an unknown (variable) resistance. Using this measured voltage, the resistance can then be calculated.





The photoresistor module - KY-018 – physical operation explanation

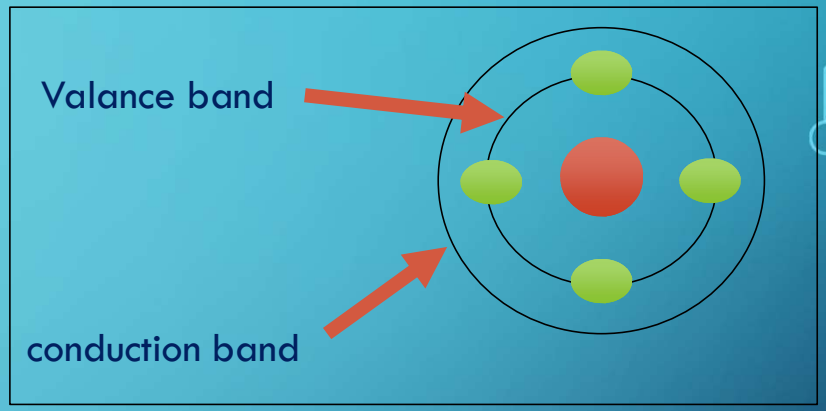
The semiconductor material on the resistor works on the principle of Photoconductivity:

In the atom there are electrons in a low energy state stored in a band called the **valance band** – the atoms are bound to the band and cannot move.

In the **conduction band** – electrons on him are free to move around.

Light is made of tiny energy packets called **photons** – then a light hit the semiconductor – the photons energizes the electrons located in the valance band – and forces them to jump to the conduction band – creating **current** in the circuit.

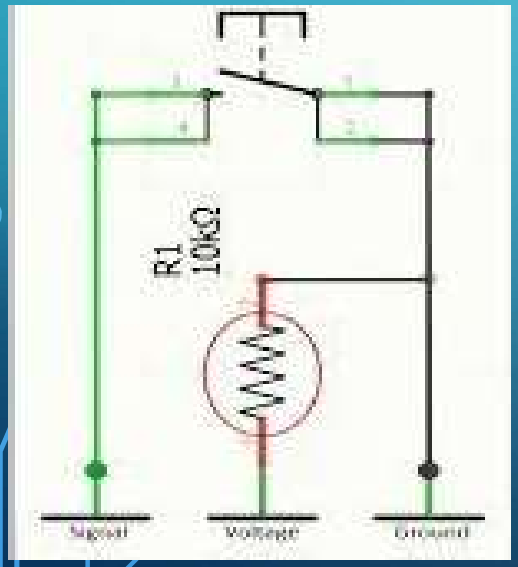
When there are no more photons-these electrons will fall back to the valance band – and current will stop.

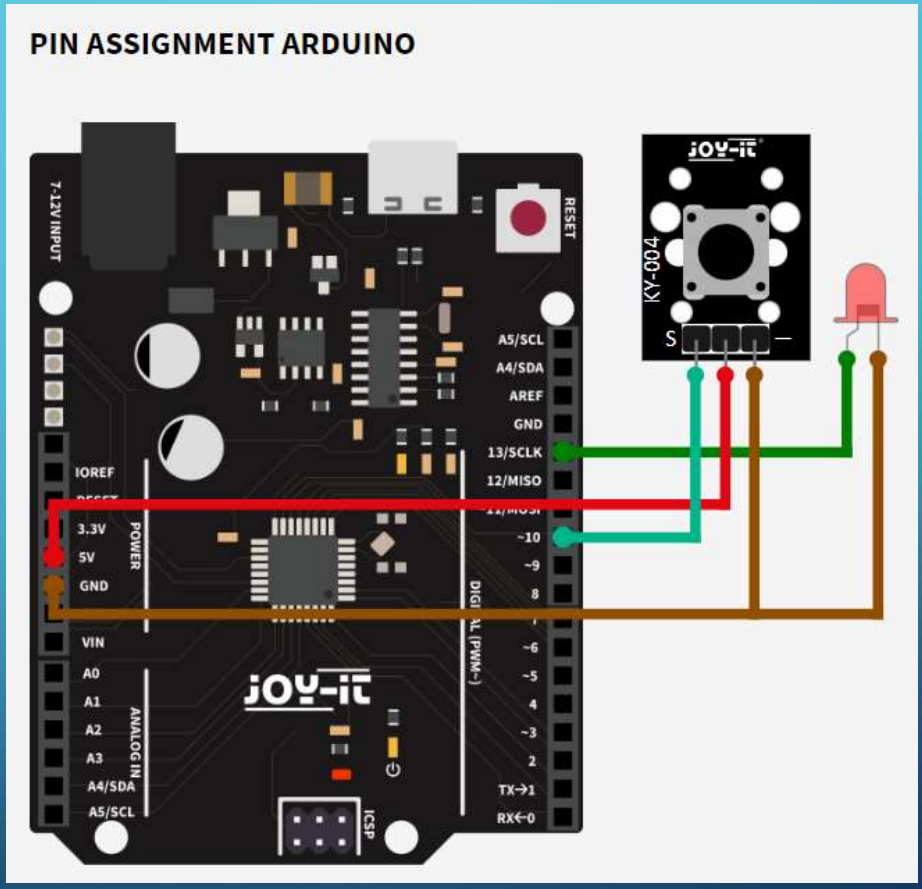


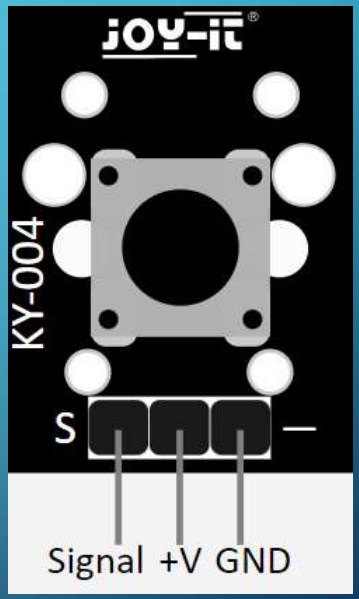
The button module - KY-004

The KY-004 Key Switch Module is a push button that will close the circuit when pressed, sending a high signal.

Operating voltage 3.3 V - 5 V



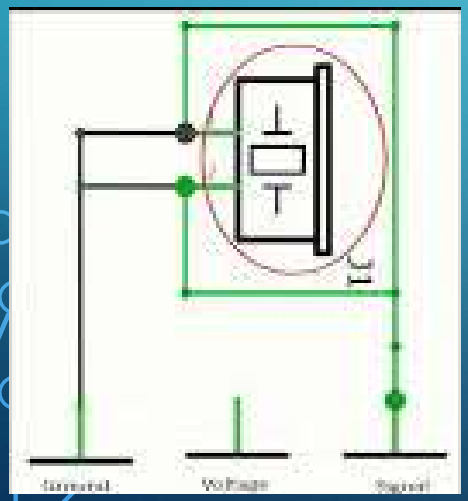


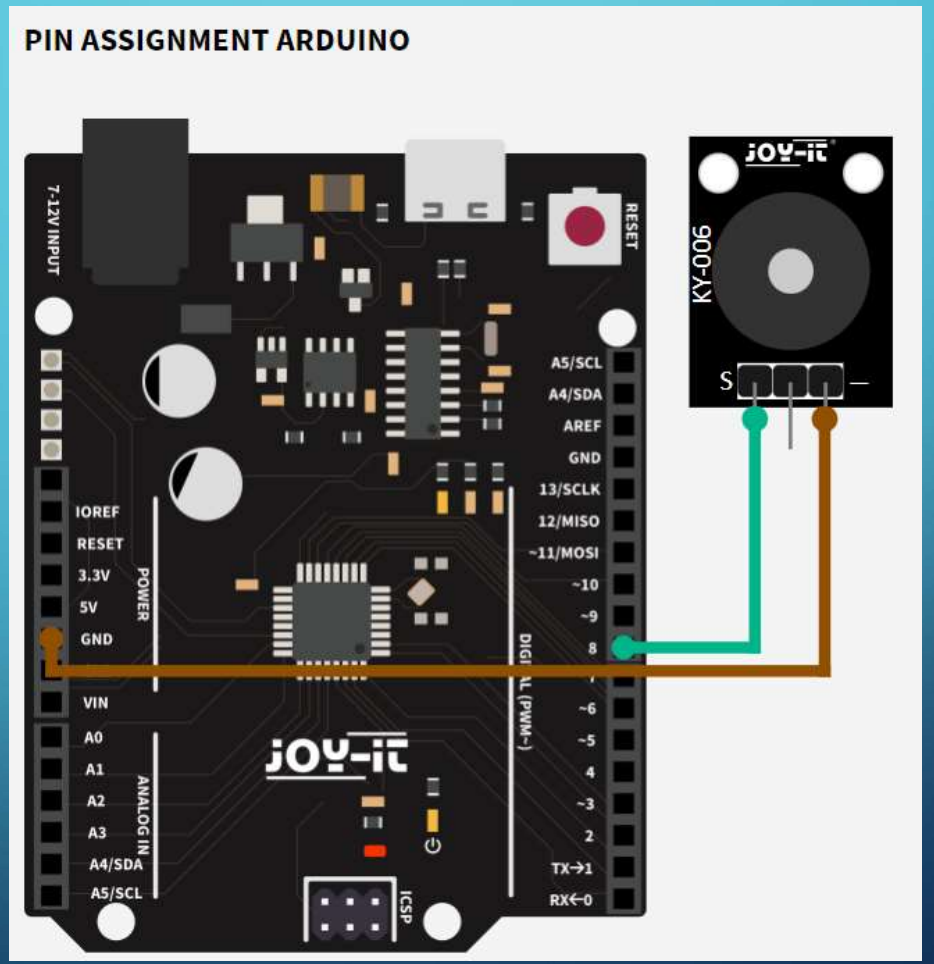


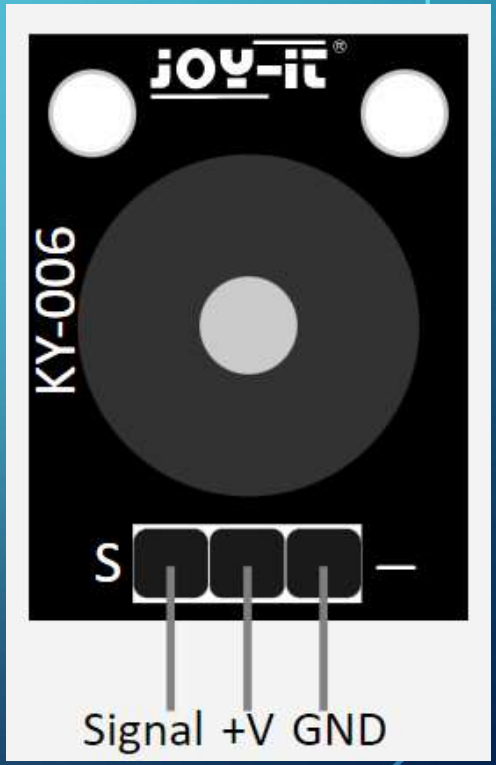
The buzzer module - KY-006

Controlled with PWM signals of different frequencies, the passive piezo buzzer can be used to generate different sounds.

Tone generation range=1,5 kHz - 2,5 kHz
Operating voltage=3.3 V - 5 V





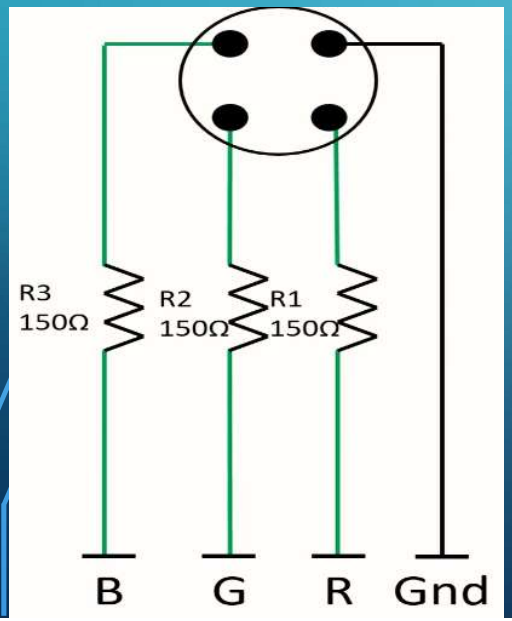


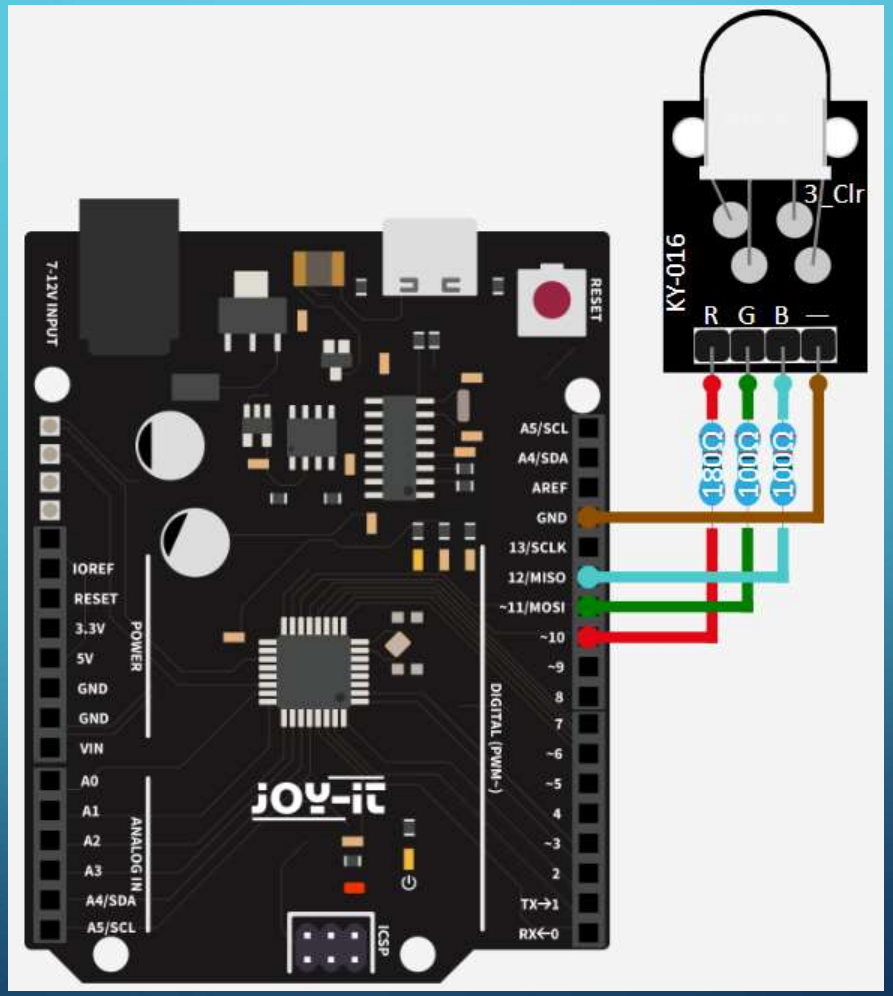
+V pin is not in use

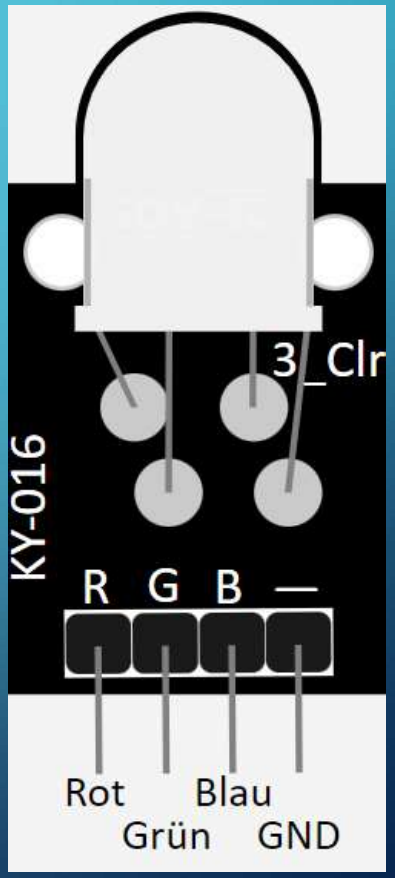
The RGB LED module - KY-016

LED module which contains a red, blue and green LED. These are connected to each other by means of a **common cathode**.

Forward voltage [Red]=1.8 V
Forward voltage [Green, Blue]=2.8 V
Forward current=20 mA



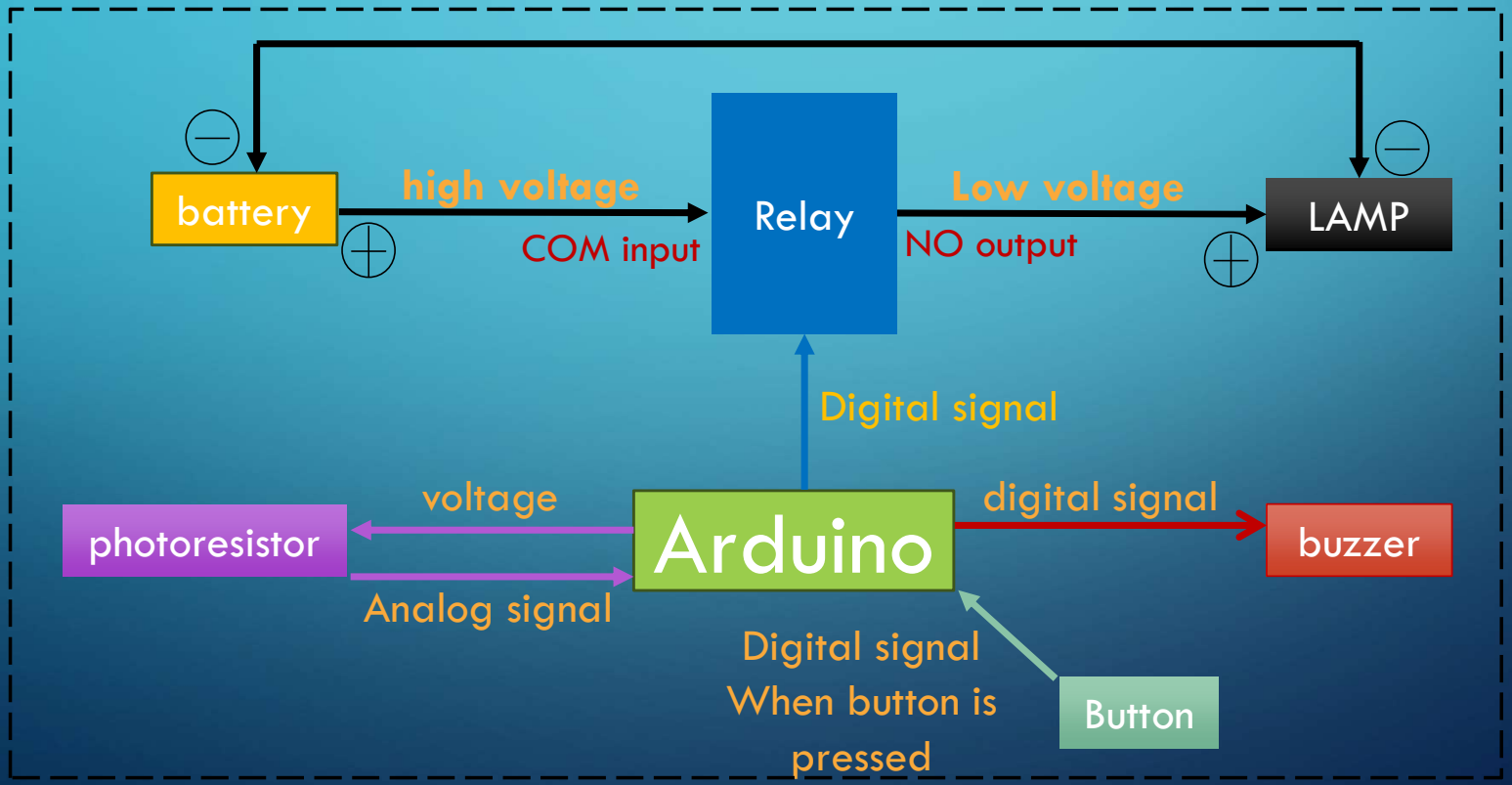




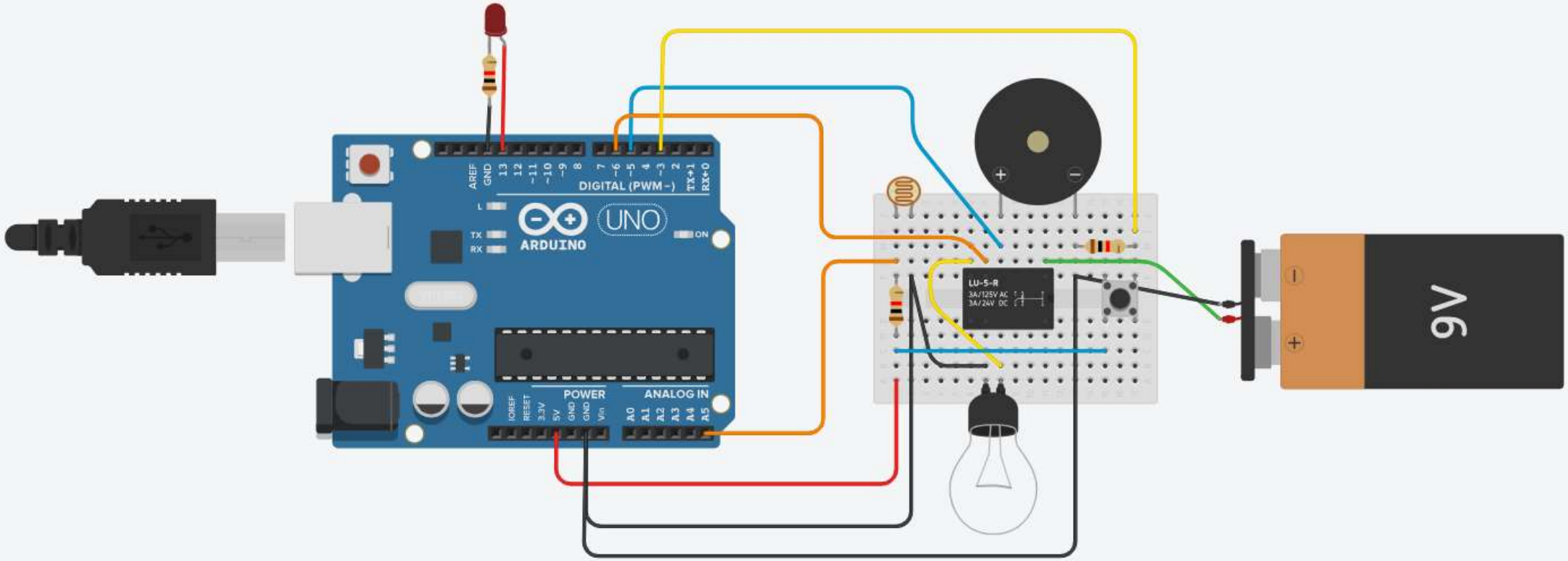
basic flow chart

Before starting to assemble the circuit on Tinkercad and on the real Arduino – we will create a basic flow chart

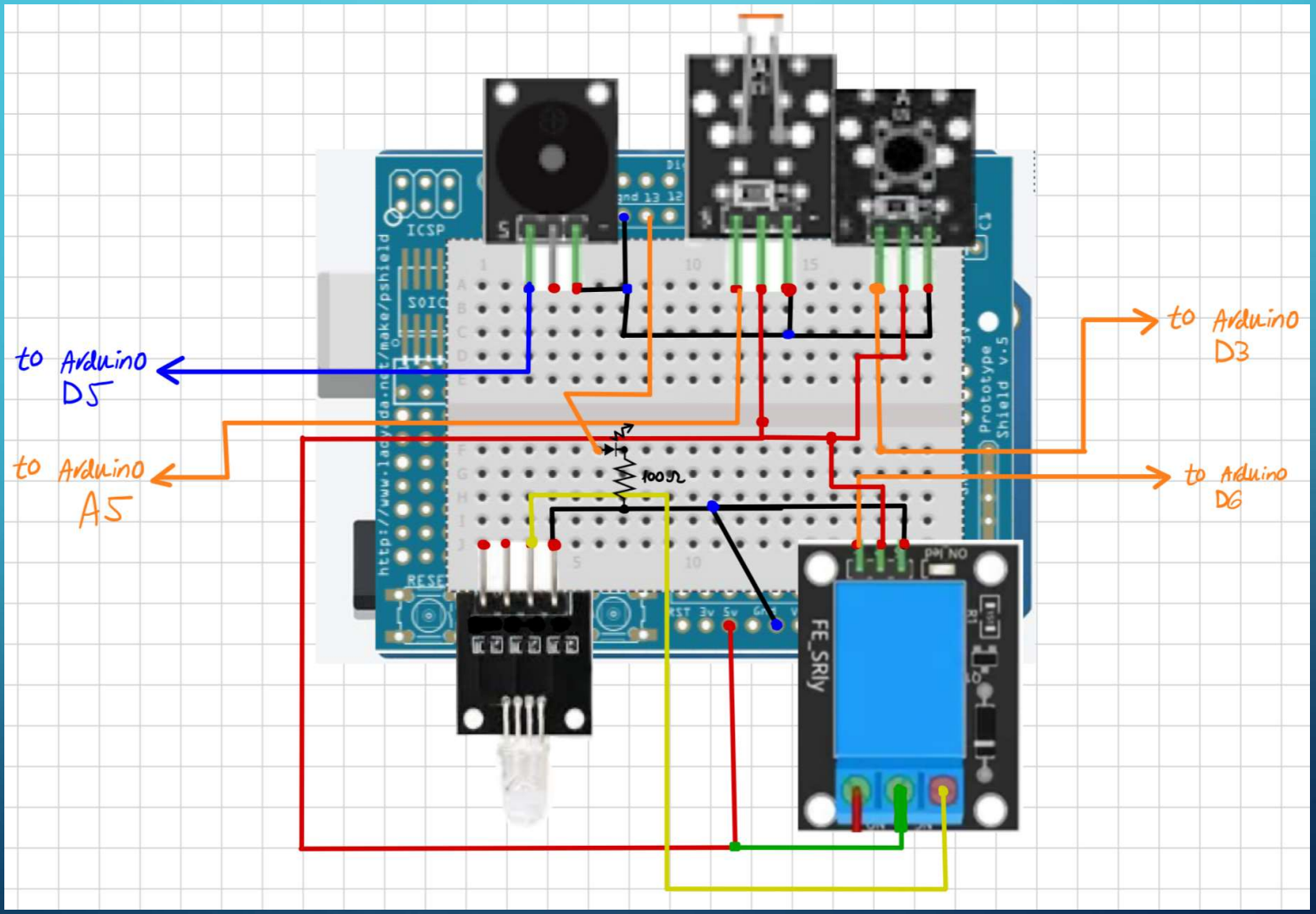
The basic flow chart will help us to control the assembling process and writing code later



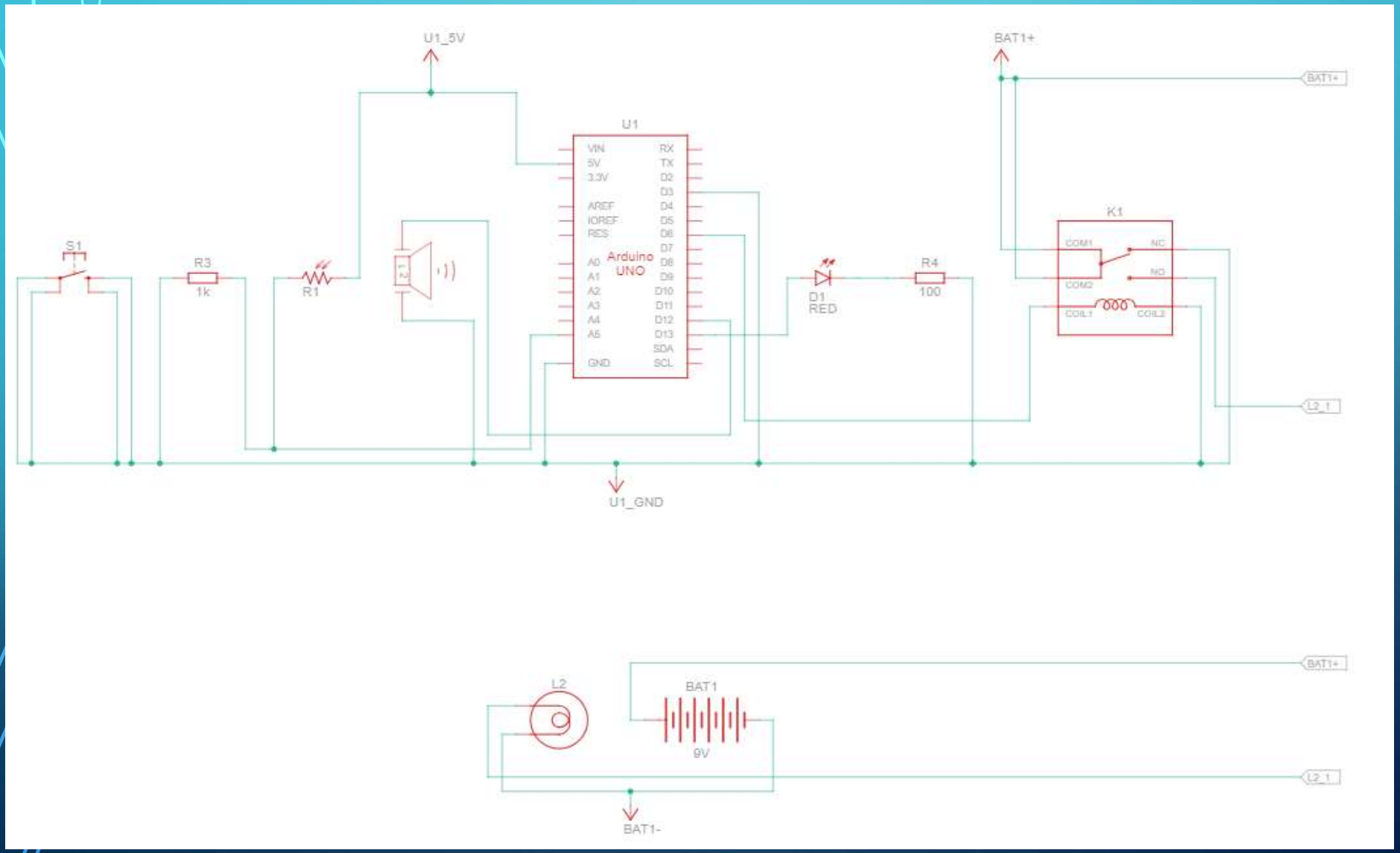
TinkerCad simulation of the circuit



Layout of the real circuit

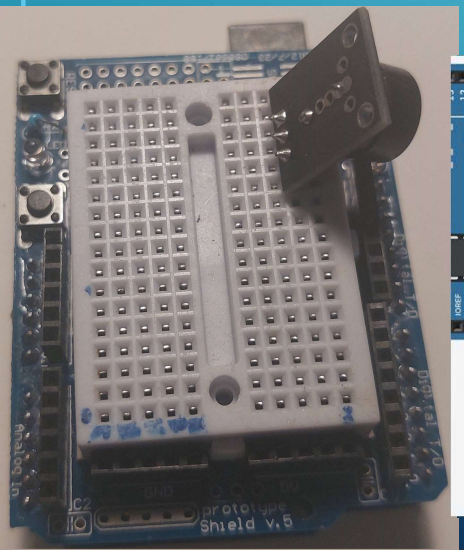


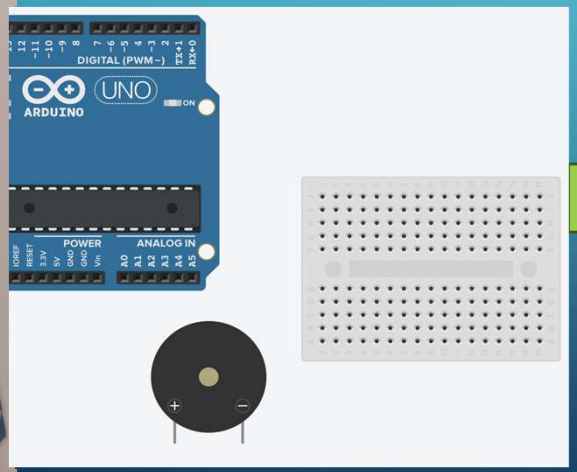
Real layout of the circuit



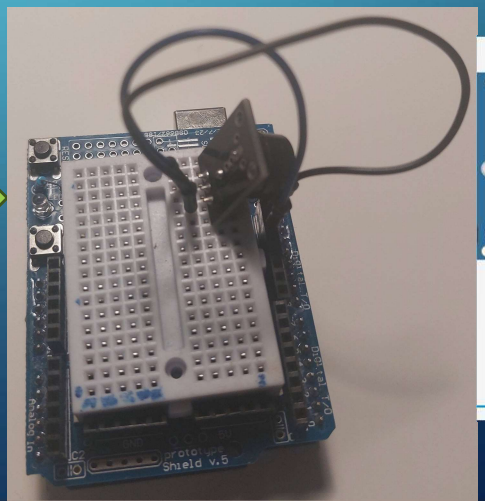
PLACEMENT OF buzzer module - KY-006 ON BREADBOAD

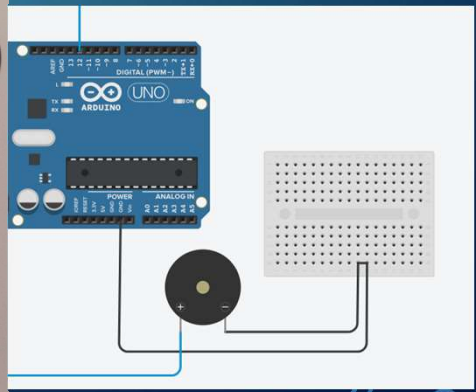
#	Component pin name	Breadboard hole name	comments
1	+V	A4	NOT CONNECTED!
2	SIGNAL	A3	Connected to digital pin 5 of Arduino
3	GND	A5	Connected to GND of Arduino (on the shield)





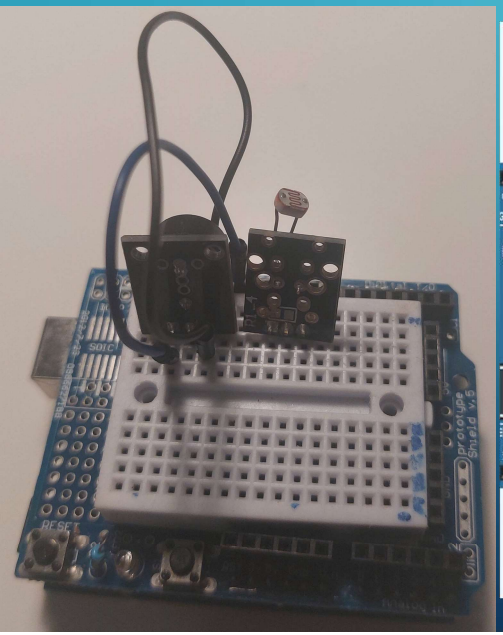
After wiring

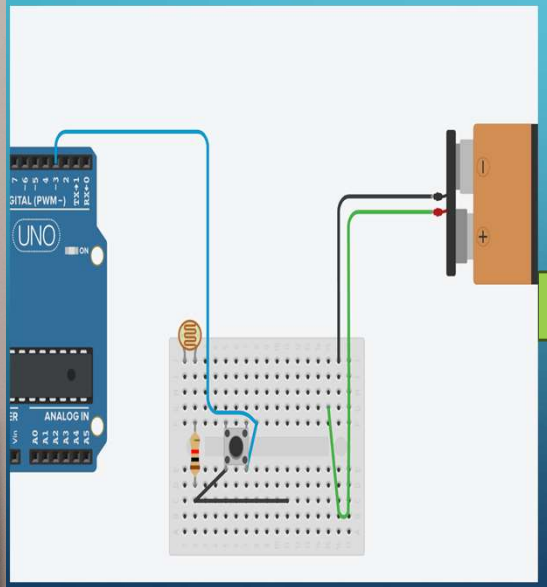




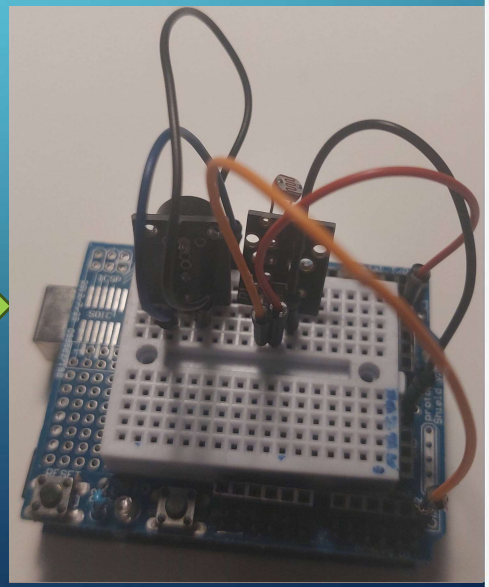
PLACEMENT OF photoresistor module - KY-018 ON BREADBOAD

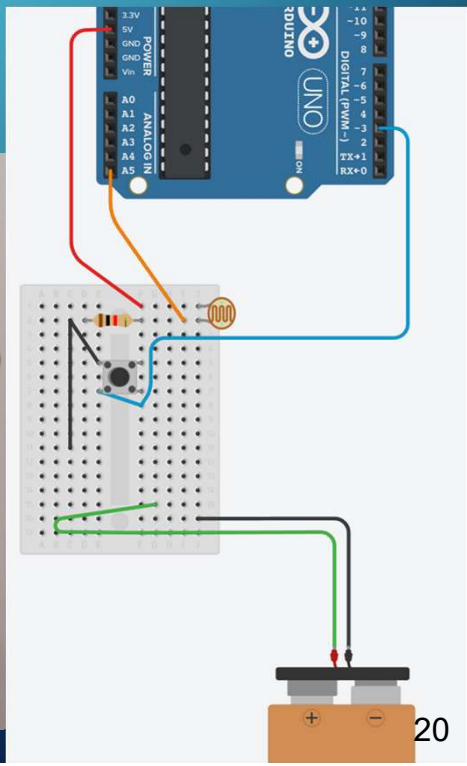
#	Component pin name	Breadboard hole name	Comments
1	+V	A13	Connected to 5V of Arduino (on the shield)
2	SIGNAL	A12	Connected to analog pin A5
3	GND	A14	Connected to GND of Arduino (on the shield)





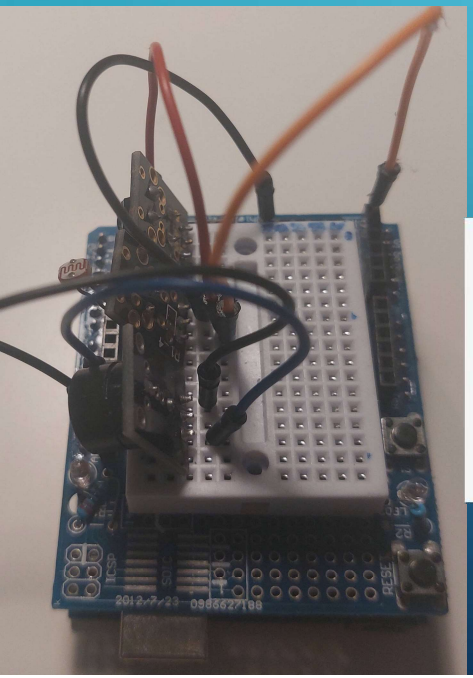
After wiring

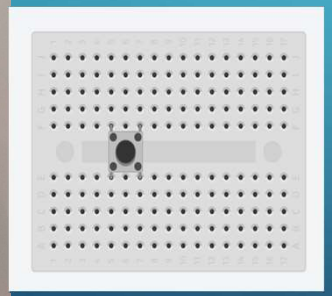




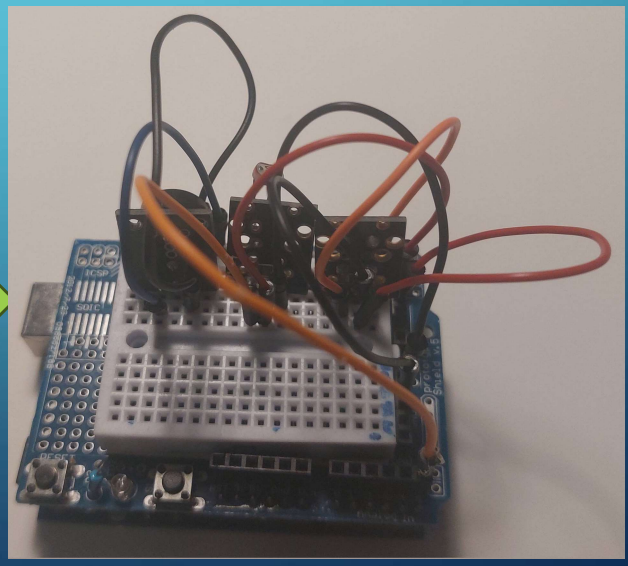
PLACEMENT OF button module - KY-004 ON BREADBOAD

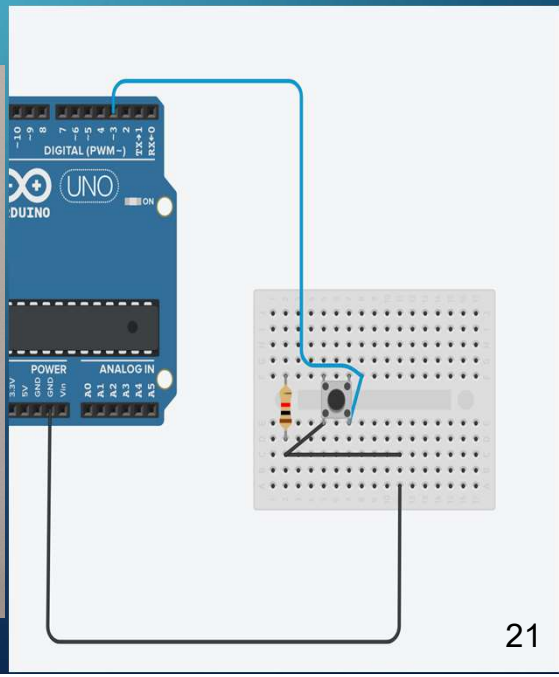
#	Component pin name	Breadboard hole name	comments
1	+V	A19	Connected to 5V of Arduino (on the shield)
2	SIGNAL	A18	Connected to digital pin 3 of Arduino
3	GND	A20	Connected to GND of Arduino (on the shield)





After wiring



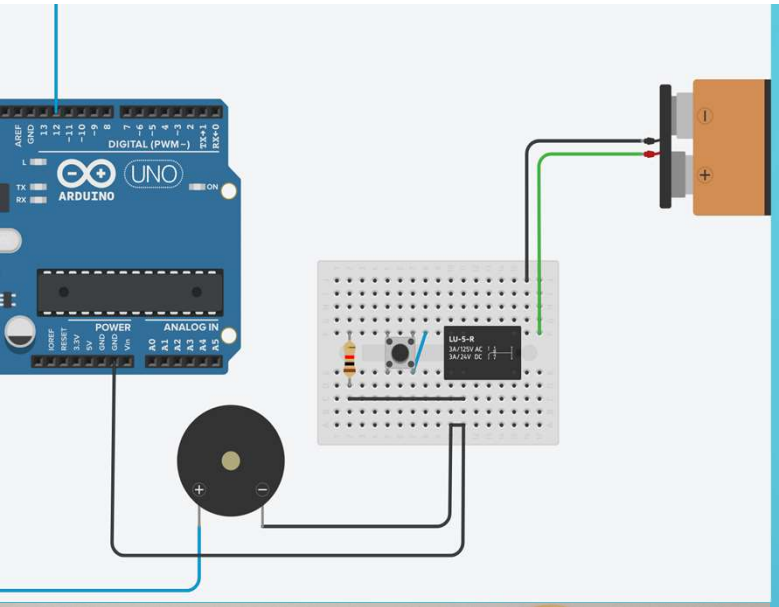


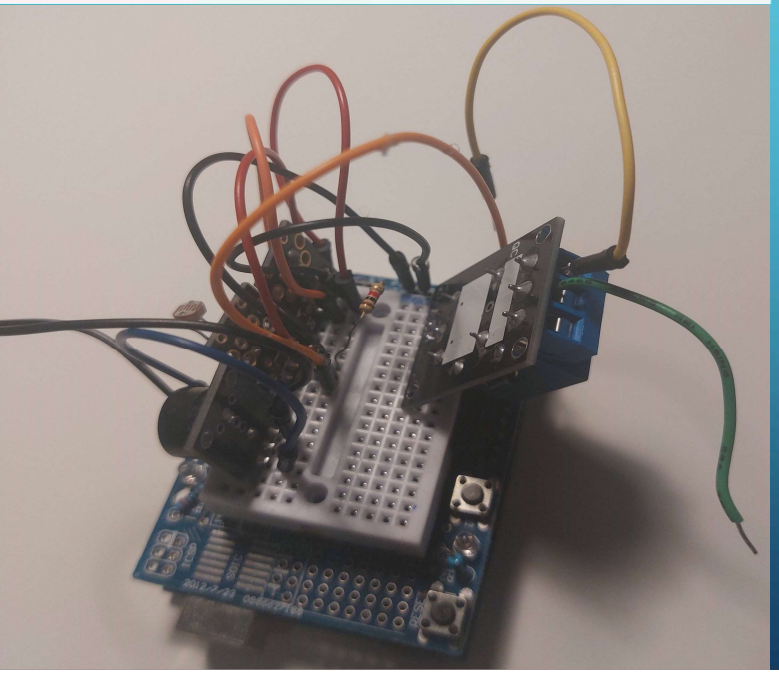
PLACEMENT OF relay module - KY-019 ON BREADBOAD

#	Component pin name	Breadboard hole name	comments
1	+V	J18	Connected to 5V of Arduino (on the shield)
2	SIGNAL	J17	Connected to digital pin 6 of Arduino
3	GND	J19	Connected to GND of Arduino (on the shield)
4	COM	None	Connected to 5V of Arduino (on original Arduino) (GREEN WIRE)
5	NO	None	To RED pin of LED module-J3
6	NC	None	Not connected

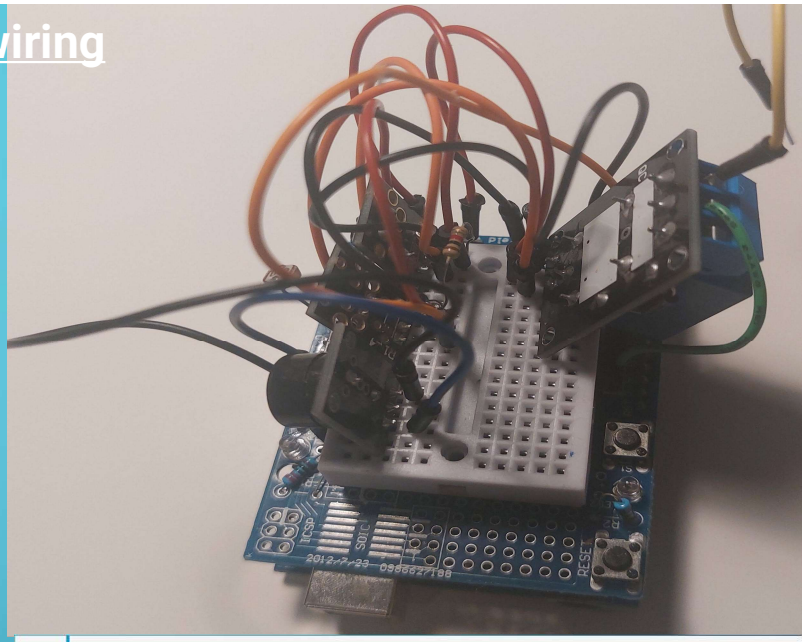
Photos of relay module before and after wiring in the next slide!

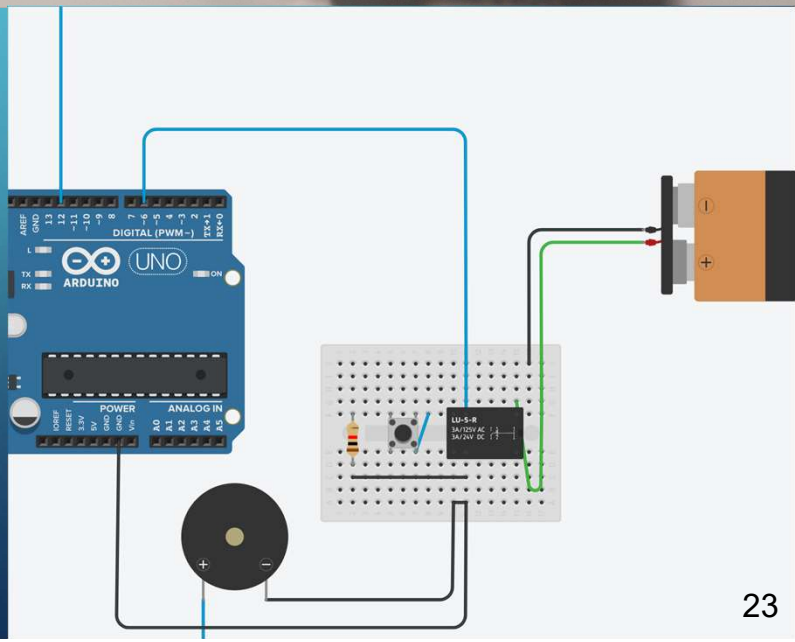
relay module before and after wiring





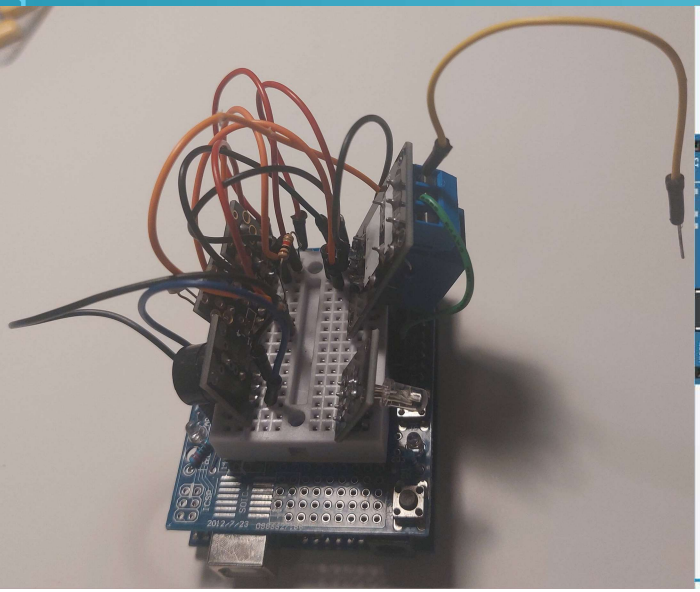
After wiring

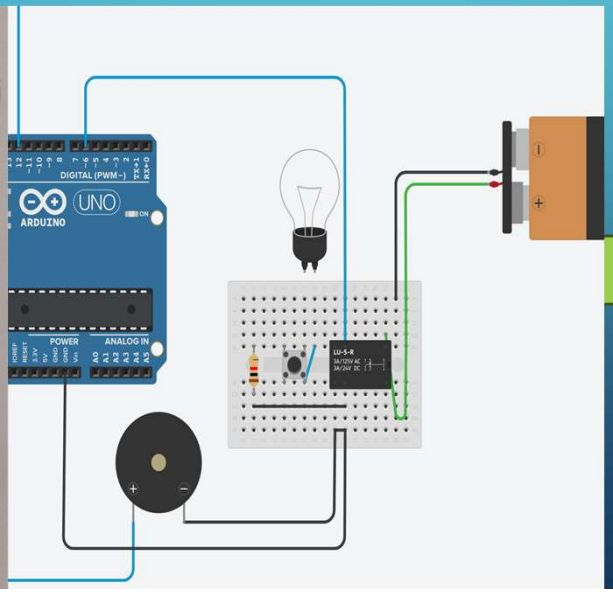




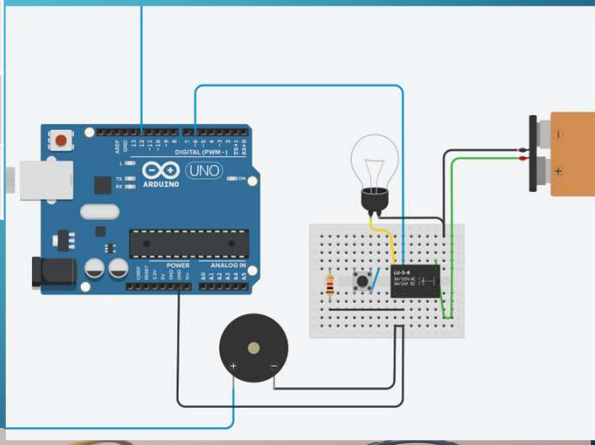
PLACEMENT OF LED module - KY-016 ON BREADBOAD

#	Component pin name	Breadboard hole name	comments
1	GND	J4	Connected to GND of Arduino (on the shield)
2	RED	J3	Connected to YELLOW wire from NO of relay
3	BLUE	J1	NOT CONNECTED
4	green	J2	NOT CONNECTED





After wiring



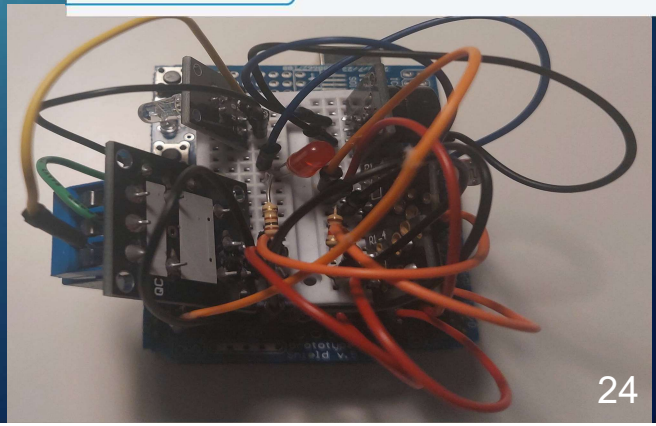
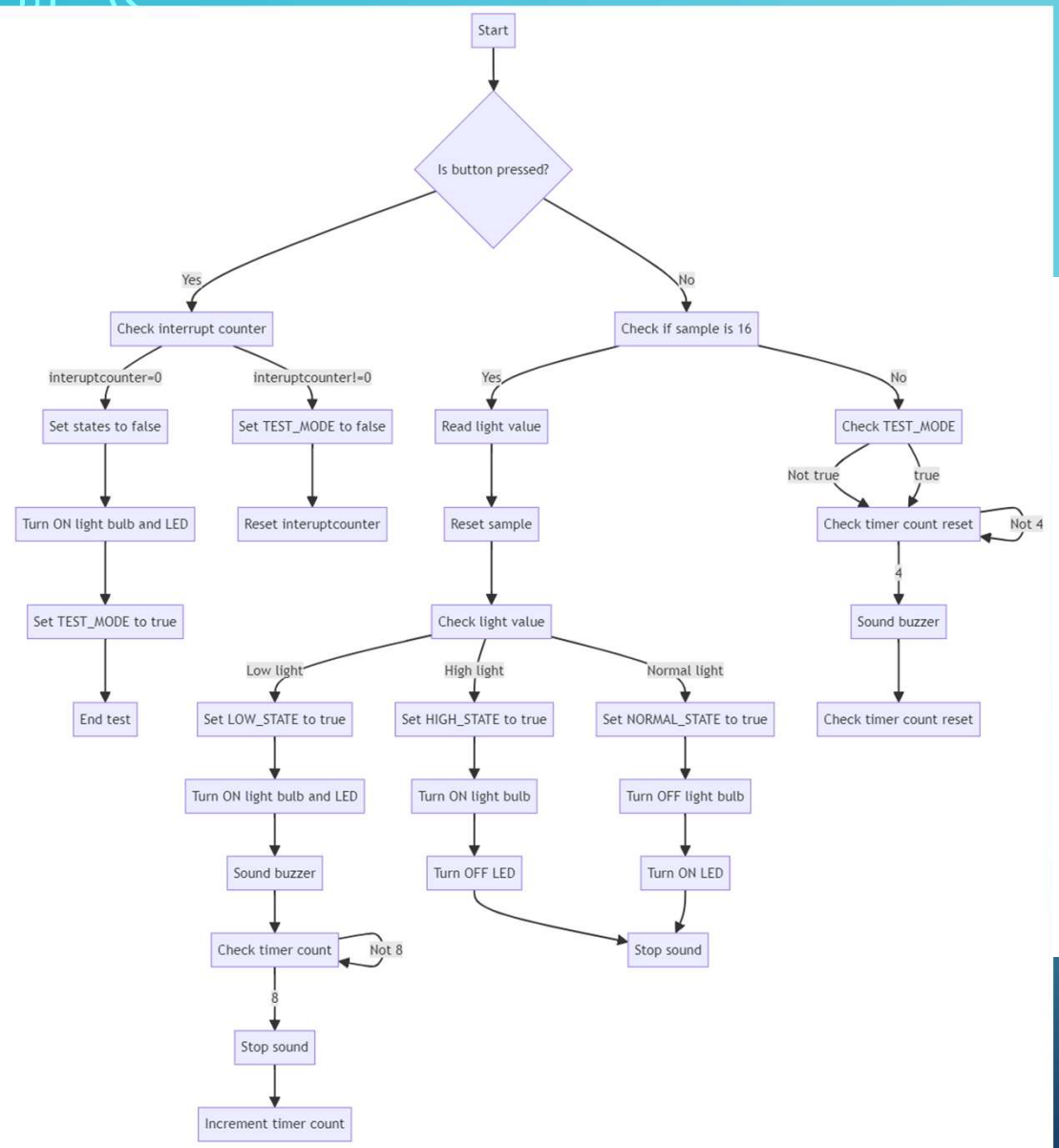
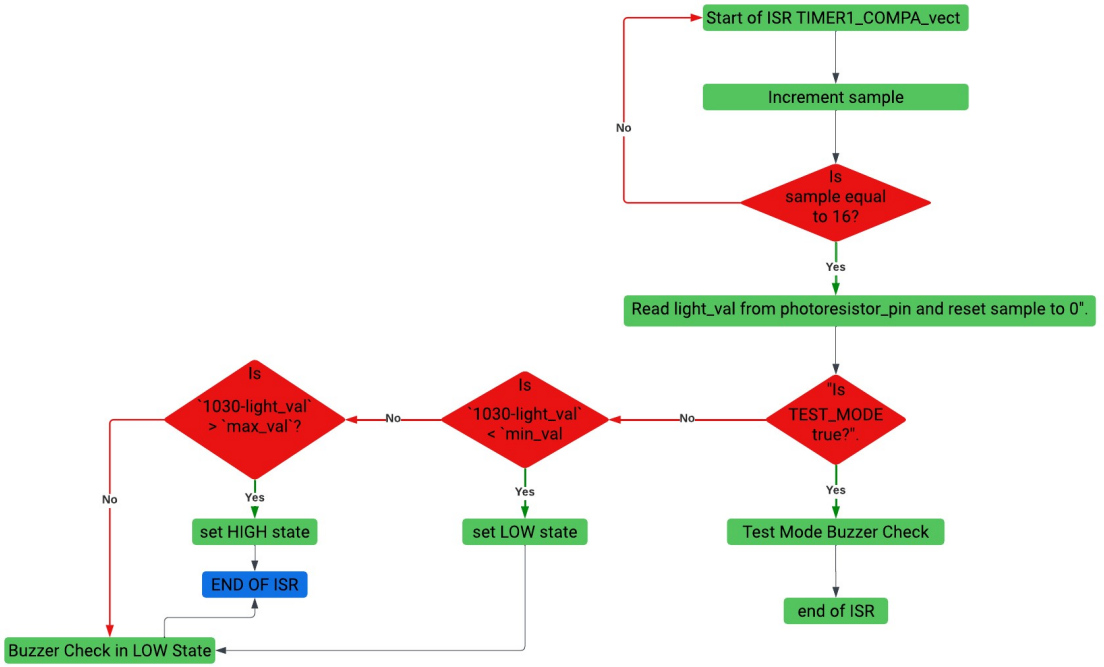


Table 3-wire' connections on the shield' breadboard

#	Wire color	Module pin name	Point A	Point B	comment
1	black	GND	D5	GND of the Arduino	Buzzer module
2	blue	Signal	D3	Arduino digital pin 5	Buzzer module
3	without	V+	Not connected	Not connected	Buzzer module
4	black	GND	D14	GND of the Arduino	Photoresistor module
5	orange	Signal	D12	Arduino analog pin 5	Photoresistor module
6	red	V+	D13	VCC of the Arduino	Photoresistor module
7	black	GND	D20	GND of the Arduino	Butten module
8	orange	Signal	D18	Arduino digital pin 3	Butten module
9	red	V+	D19	VCC of the Arduino	Butten module
10	black	GND	G19	GND of the Arduino	Relay module
11	orange	Signal	G17	Arduino digital pin 6	Relay module
12	red	V+	G18	VCC of the Arduino	Relay module
13	yellow	NO	none	To F20 (shortcut wire)	Relay module
14	green	COM	none	VCC of the Arduino	Relay module
15	yellow	RED	G3	F20	Bulb module
16	black	GND	G4	GND of the Arduino	Bulb module
17	blue	anode of LED	G7	Arduino digital pin 13	LED 13

Detailed Block chart of the system-part 1



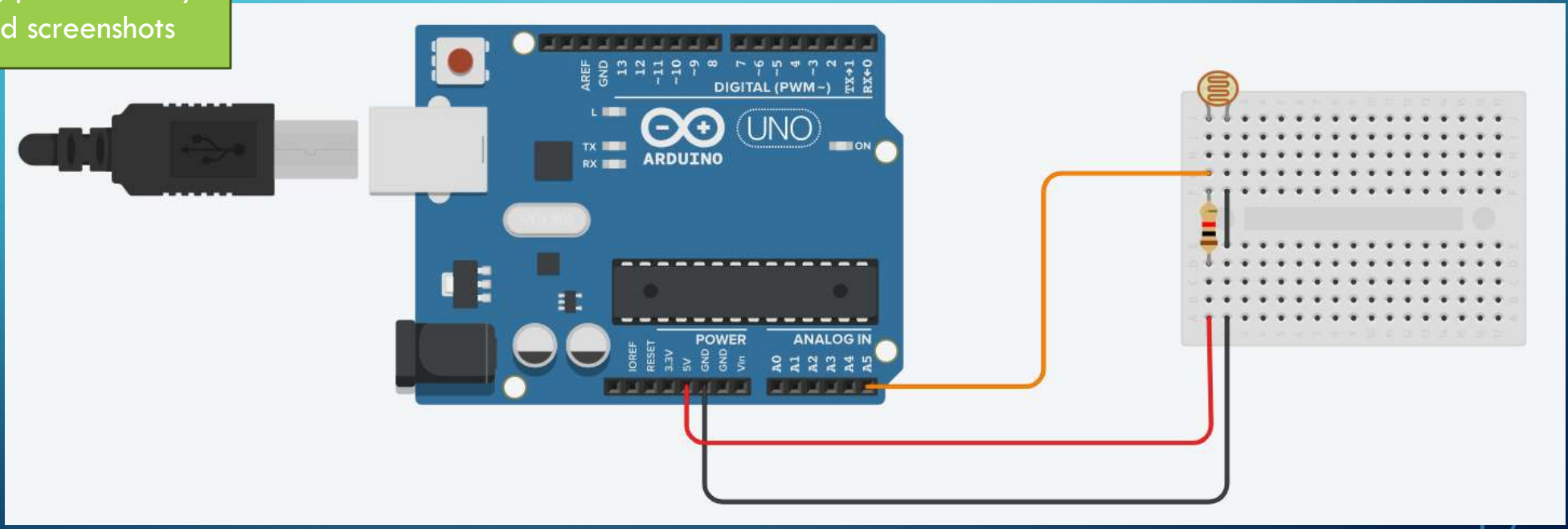


HW01.16. PROTOCOL OF TEST-TEST 1-SIMULATION-PART 1

Being done by:
ID : 2210
On : 9/3/2024
10:00 (Saturday morning)
By testing simultaneously the hardware and the simulation in my PC – taking pictures in my smartphone and screenshots

Test 1: check the range of photoresistor.
In this test we are going to measure the range of light of photoresistor in order to decide the value of L1 and L2.

Simulation:
we have a circuit with photoresistor connected to analog pin A5 and a resistor of 1KΩ connected to 5v pin in Arduino.



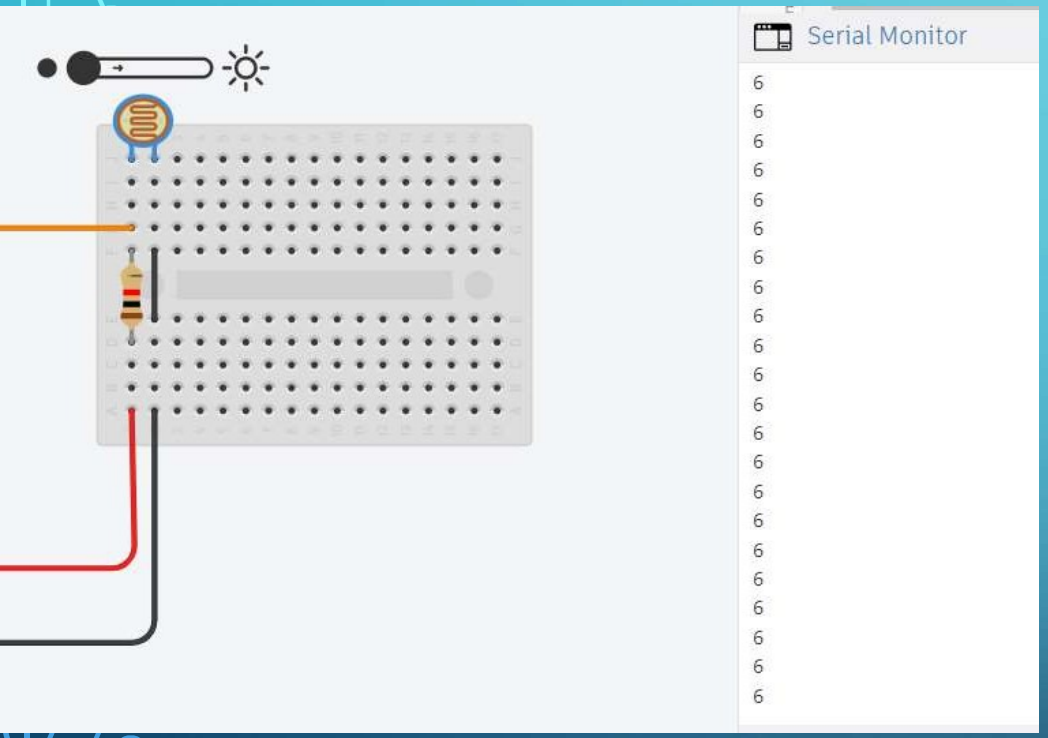
Link to TinkerCad of Test 1:
https://www.tinkercad.com/things/kGP9DIgnXHj-hw0116-protocol-of-test-test-1-simulation

Being done by:
ID : 2210
On : 9/3/2024
10:00 (Saturday morning)

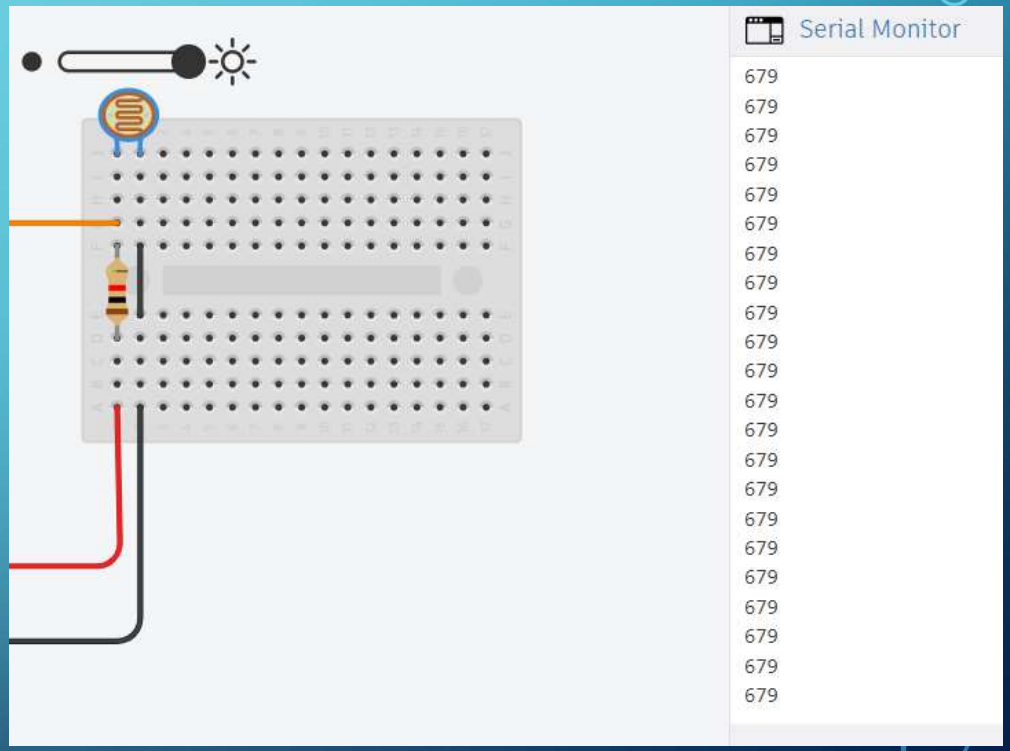
HW01.16. PROTOCOL OF TEST-TEST 1-SIMULATION-PART 2

The results of the simulation:

check lowest intensity of light:



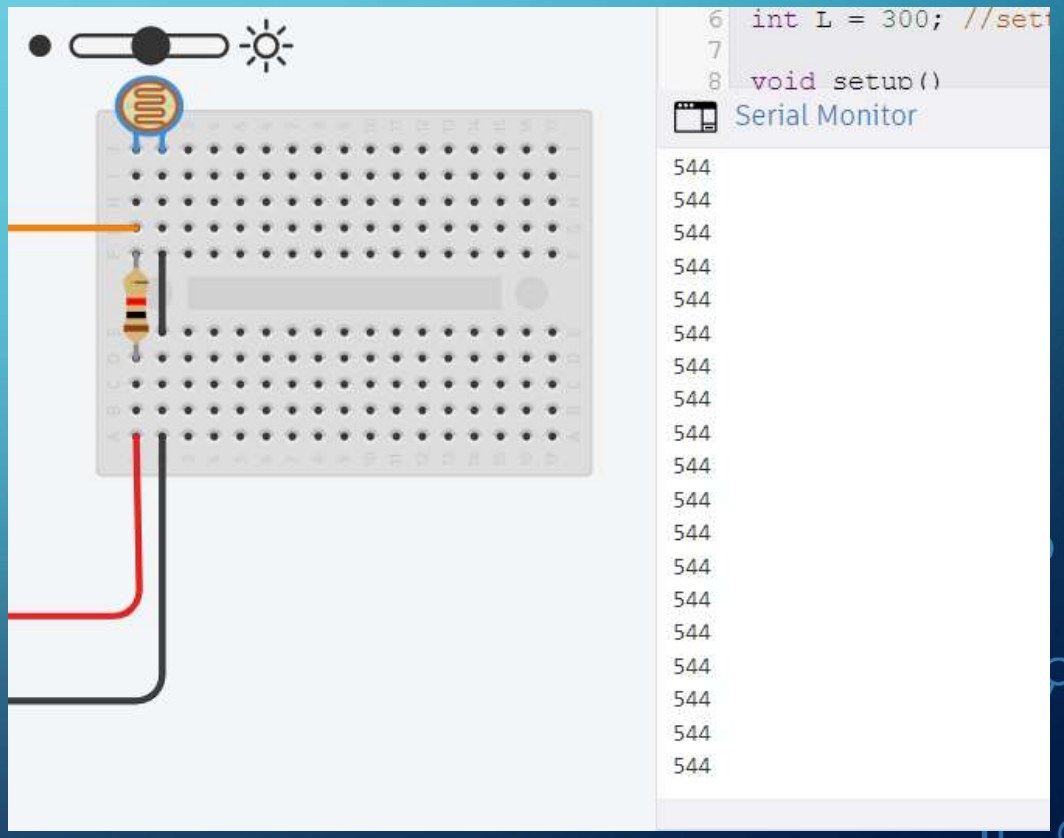
check highest intensity of light:



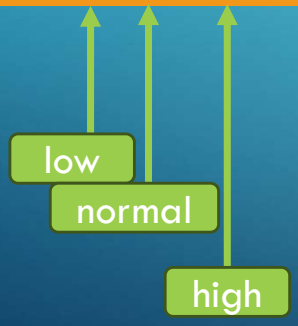
HW01.16. PROTOCOL OF TEST-TEST 1-SIMULATION-PART 2

The results of the simulation-PART 2

check MEDIUM intensity of light:



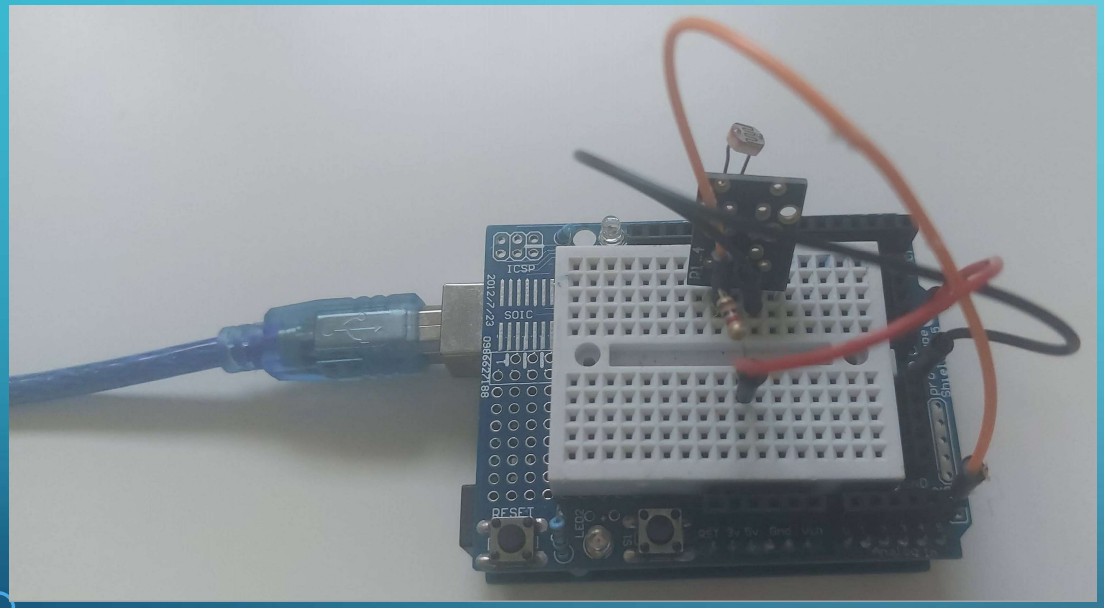
The lowest intensity of light is 6 and highest intensity of light is 679, and the medidum intensity is 544.
So the range is [6,544,679].

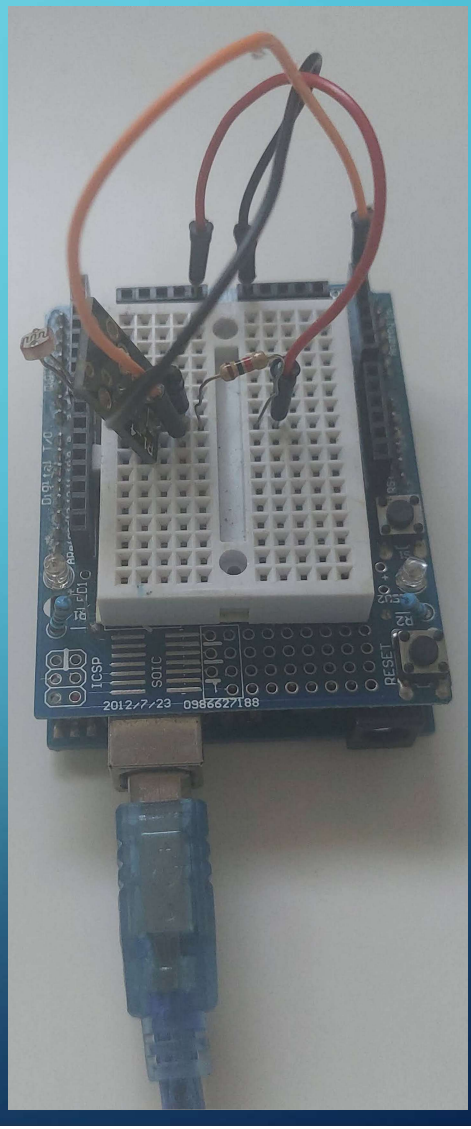


HW01.16. PROTOCOL OF TEST-TEST 1-HAWRDWARE-PART 1

Being done by:
ID : 2210
On : 9/3/2024
10:30 (Saturday morning)

Circuit: (from two angles of view)





HW01.16. PROTOCOL OF TEST-TEST 1-HAWRDWARE-PART 2

Being done by:
ID : 2210
On : 9/3/2024
10:30 (Saturday morning)

HW01_16_PROTOCOL_OF_TEST_TEST_1_HAWRDWARE.ino

```
1   /* flags */
2   bool NORMAL_STATE = false;
3   bool HIGH_STATE   = false;
4   bool LOW_STATE    = false;
5   bool TEST_MODE    = false;
6
7   int intensity_of_Light=A5;
8   int min_val = 100; //light
9   int max_val = 600; //no light
10  int light_val = 300; //setting-up initial value
11
12  /* time counter */
13  int timer_cnt = 0;
14  int timer_cnt_rst = 0;
15  int interuptcounter =0;
16  int sample=0;
```

Define of Pins and constants

Pay attention that the L1 &L2 defined in this code are just for test and not the final L1 & L2 !

At first we named them min_val and max_val

HW01.16. PROTOCOL OF TEST-TEST 1-HAWRDWARE-PART 3

Being done by:
ID : 2210
On : 9/3/2024
10:30 (Saturday morning)

```
18  void setup()
19  {
20    Serial.begin(9600);
21    // TIMER 1 for interrupt frequency 16 Hz:
22    cli(); // stop interrupts
23    TCCR1A = 0; // set entire TCCR1A register to 0
24    TCCR1B = 0; // same for TCCR1B
25    TCNT1  = 0; // initialize counter value to 0
26    // set compare match register for 16 Hz increments
27    OCR1A = 15624; // = 16000000 / (64 * 16) - 1 (must be <65536)
28    // turn on CTC mode
29    TCCR1B |= (1 << WGM12);
30    // Set CS12, CS11 and CS10 bits for 64 prescaler
31    TCCR1B |= (0 << CS12) | (1 << CS11) | (1 << CS10);
32    // enable timer compare interrupt
33    TIMSK1 |= (1 << OCIE1A);
34    sei(); // allow interrupts
35    }
```

Define the timer – to sample the intensity of the light every **1** second. Using online timer calculator: http://www.8bit-era.cz/arduino-timer-interrupts-calculator.html

HW01.16. PROTOCOL OF TEST-TEST 1-HAWRDWARE-PART 4

ISR (interrupt service routine) function to analog read the intensity of light

```
37  void loop()
38  {
39
40  }
41  ISR(TIMER1_COMPA_vect)
42  {
43    if(sample==16)
44    {
45      light_val = analogRead(intensity_of_Light);
46      sample=0;
47    }
48
49    else
50    {
51      sample++;
52    }
53      if(1)
54      {
55        if((1030-light_val) < min_val)
56        {
57            NORMAL_STATE = false;
58            HIGH_STATE   = false;
59            LOW_STATE    = true;
60            TEST_MODE    = false;
61        }
62        else if((1030-light_val) > max_val)
63        {
64            NORMAL_STATE = false;
65            HIGH_STATE   = true;
66            LOW_STATE    = false;
67            TEST_MODE    = false;
68        }
69        else
70        {
71            NORMAL_STATE = true;
72            HIGH_STATE   = false;
73            LOW_STATE    = false;
74            TEST_MODE    = false;
75        }
```

HW01.16. PROTOCOL OF TEST-TEST 1-HAWRDWARE-PART 5

```
77      if(LOW_STATE)
78      {
79        Serial.print("dark:\n");
80        Serial.println(1030-light_val);
81
82        if(timer_cnt == 8)
83        {
84          timer_cnt=0;
85        }
86        else
87        {
88            timer_cnt++;
89         }
90
91      }
92
93      else if(HIGH_STATE)
94      {
95        Serial.print("super high light:\n");
96        Serial.println(1030-light_val);
97
98      }
```

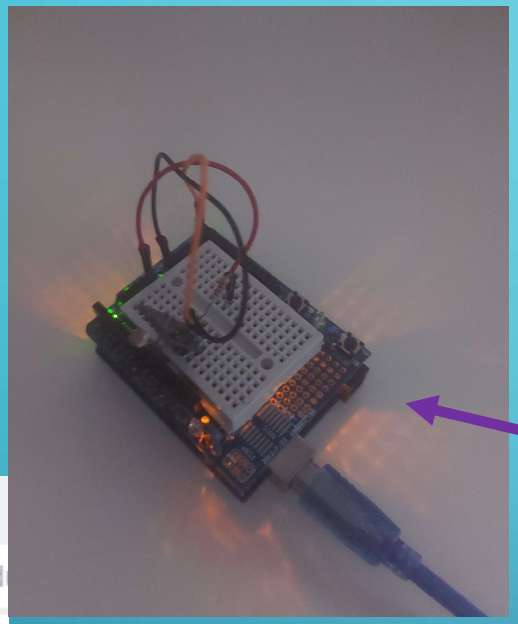
Print the intensity of light with a fix → 1023-L because otherwise it will print in reversed order

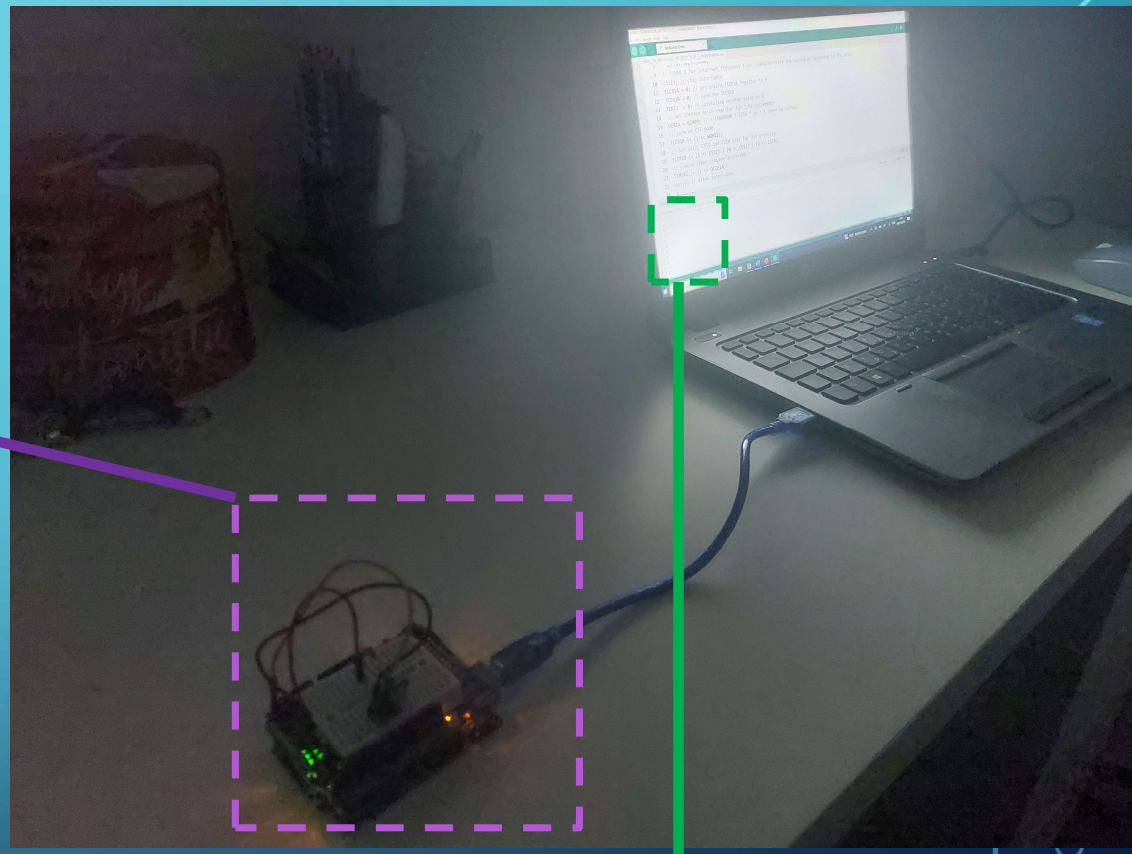
```
100       else if(NORMAL_STATE)
101       {
102         Serial.print("normal light:\n");
103         Serial.println(1030-light_val);
104       }
105     }
106
107
108   }
```

HW01.16. PROTOCOL OF TEST-TEST 1-HAWRDWARE-PART 6

Result Hardware: (when there is in dark in the room = LOW LIGHT STATE)

Being done by:
ID : 2210
On : 9/3/2024
11:10 (Saturday morning)





Output Serial Monitor ×

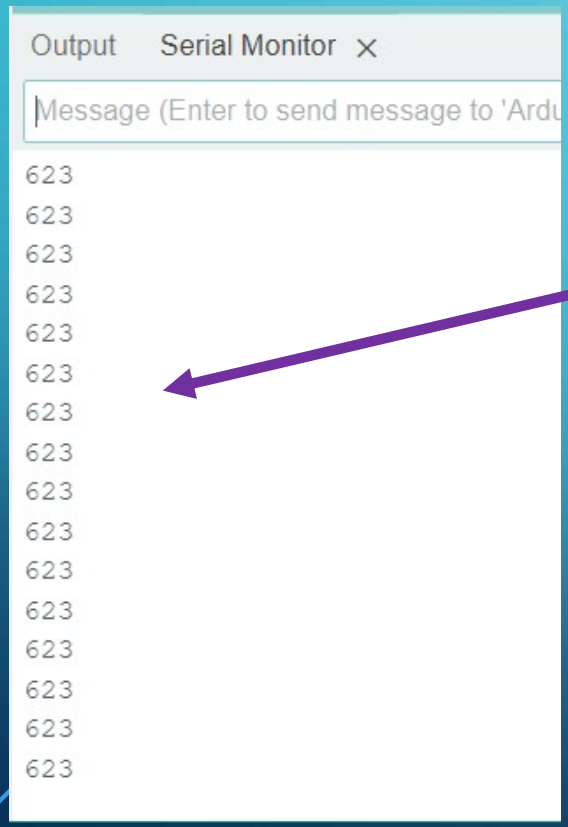
Message (Enter to send message to 'Ard

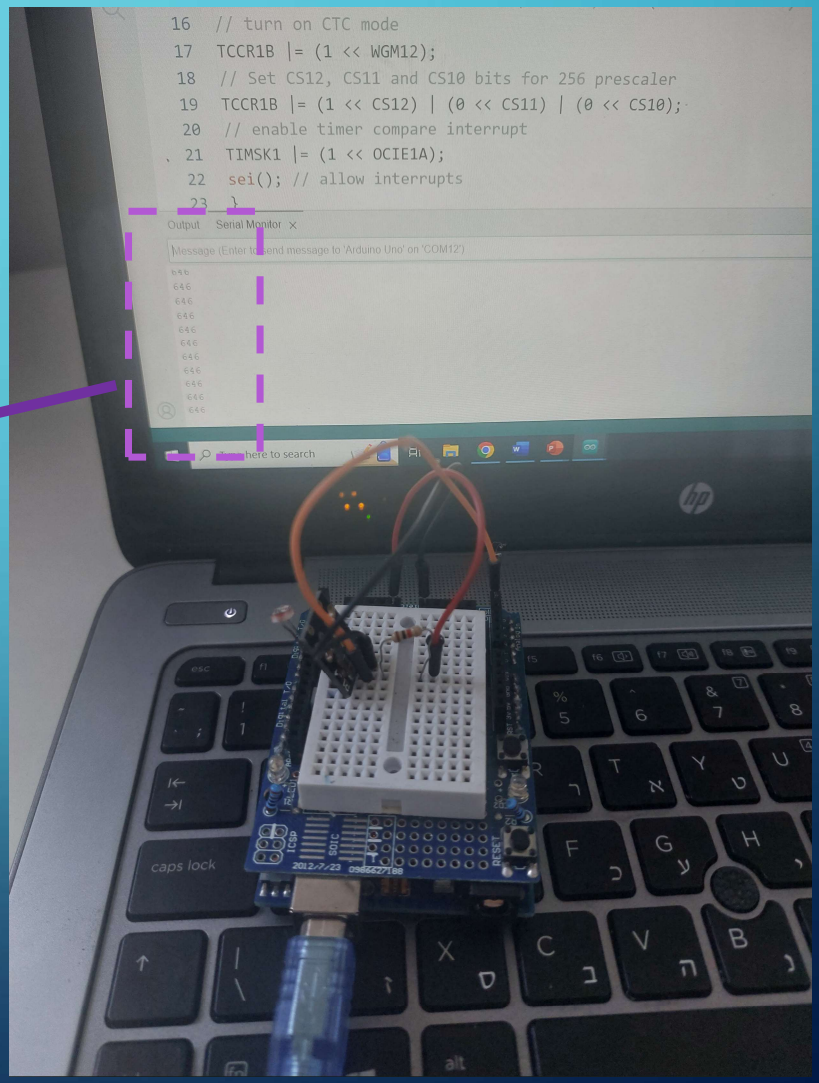
31
31
31
31
31
31
31
31
31
31
31
31

HW01.16. PROTOCOL OF TEST-TEST 1-HAWRDWARE-PART 7

Result Hardware: (when there is normal light in the room = NORMAL LIGHT STATE)

Being done by:
ID : 2210
On : 9/3/2024
10:50 (Saturday morning)

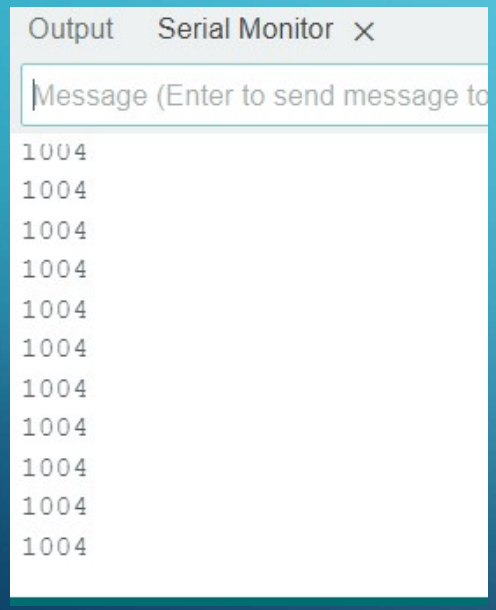




HW01.16. PROTOCOL OF TEST-TEST 1-HAWRDWARE-PART 8

Result Hardware: (when there is super high light in the room = HIGH LIGHT STATE)

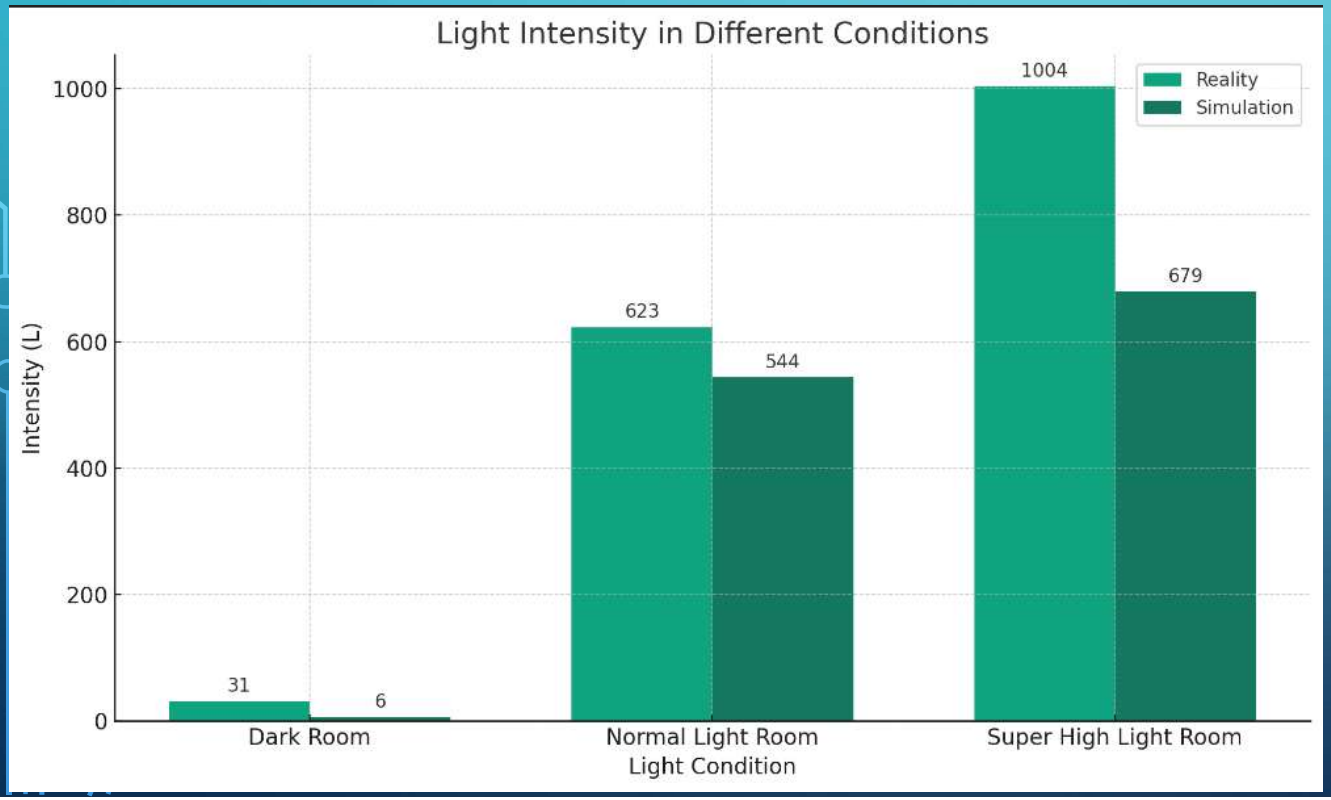
Being done by:
ID : 2210
On : 9/3/2024
11:20 (Saturday morning)



HW01.16. PROTOCOL OF TEST-TEST 1-HAWRDWARE-PART 9

Data Organization into Tables

To analyze our data effectively, we'll first arrange it into tables to have a clear overview. Then, we'll plot the data on graphs to visually compare the intensity of light in different conditions between reality and the TinkerCad simulation.
This comparison is for helping us identify the optimized L1 and L2 values for both scenarios.



Reality Measures

Intensity (L)	Light Condition
31	Dark Room (L < L2)
623	Normal Light Room (L2 < L < L1)
1004	Super High Light Room (L > L1)

Simulation Measures

Intensity (L)	Light Condition
6	Dark Room (L < L2)
544	Normal Light Room (L2 < L < L1)
679	Super High Light Room (L > L1)

HW01.16. PROTOCOL OF TEST-TEST 1-PART 6

Conclusions from the tests-decide values of L1 & L2

L2 Threshold: This value should be identified by the transition from "Dark Room" to "Normal Light Room". The reality measure jumps from 31 to 623, and the simulation from 6 to 544.

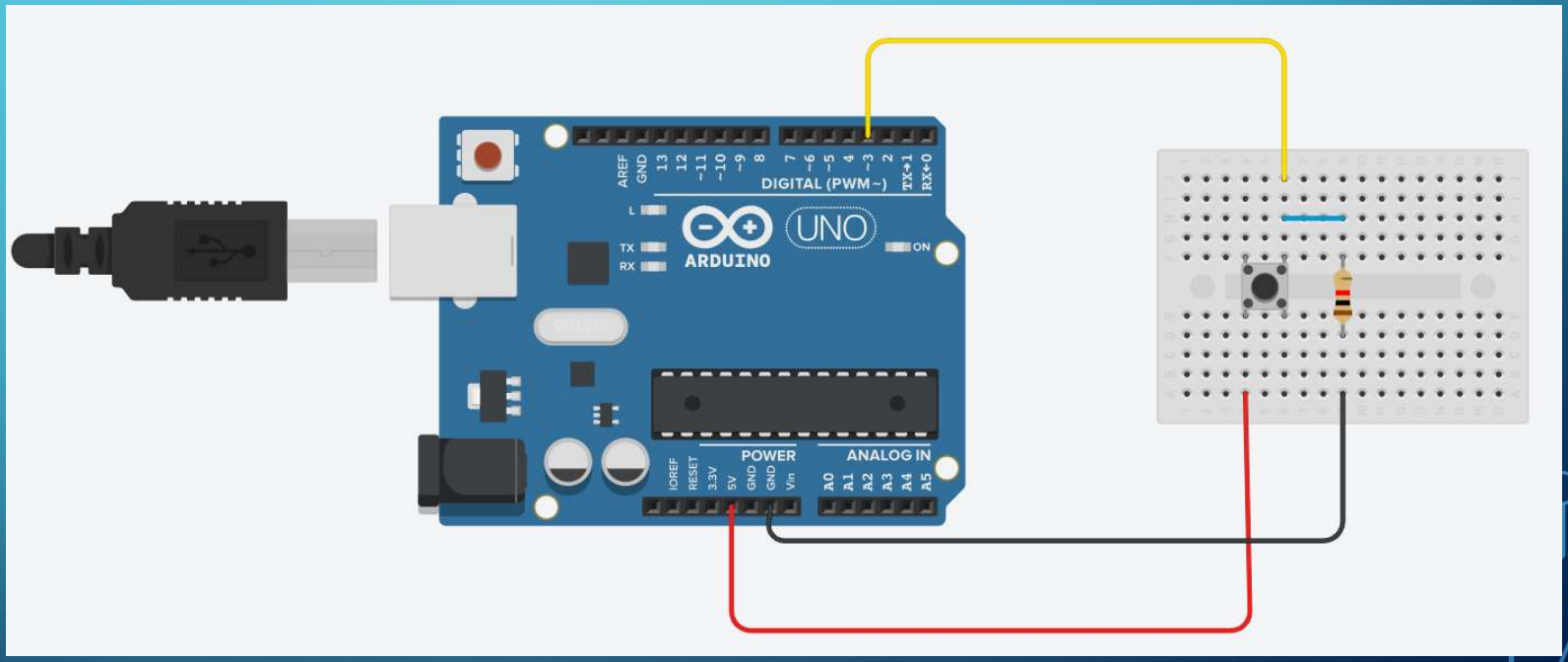
L1 Threshold: This is the transition point from "Normal Light Room" to "Super High Light Room".. In reality, the transition is from 623 to 1004, and in the simulation, it's from 544 to 679

we'll pick L2=100 & we'll pick L1=600 → these values will ensure that the system will work correctly both in reality and in simulation!

HW01.16. PROTOCOL OF TEST-TEST 2-SIMULATION-PART 1

Being done by:
ID : 2210
On : 9/3/2024
12:20 (Saturday morning)
By testing simultaneously, the hardware and the simulation in my PC – taking pictures in my smartphone and screenshots

Test 2: check the interrupt function of Button.
In this test we are going to check that when we press the Button, the interrupt is called, and message is printed to the serial monitor. The massage is: "Interrupt is working successfully!".



Link to TinkerCad of Test 2:
https://www.tinkercad.com/things/bALpTTHuNym-hw0116-protocol-of-test-test-2-simulation

HW01.16. PROTOCOL OF TEST-TEST 2-SIMULATION-PART 2

The results of the simulation:

```
Interrupt is working successfully!
Interrupt is working successfully!
Interrupt is working successfully!
Interrupt is working successfully!
Interrupt is working successfully!
Interrupt is working successfully!
```

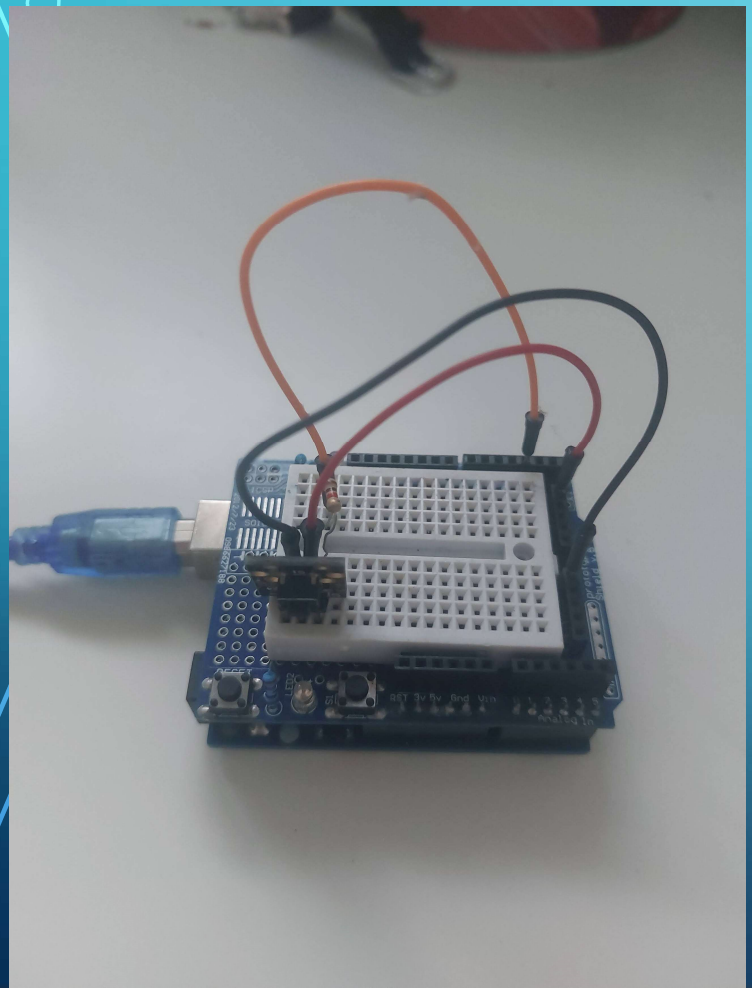
```
1  int Button = 3;
2
3
4  void setup()
5  {
6    Serial.begin(9600);
7    pinMode(Button, INPUT);
8    attachInterrupt(digitalPinToInterrupt(Button),TestInterrupt,RISING);
9  }
10
11 void TestInterrupt() //when button is pressed
12 {
13     Serial.println("Interrupt is working successfully!");
14 }
15
16 void loop()
17 {
18
19 }
```

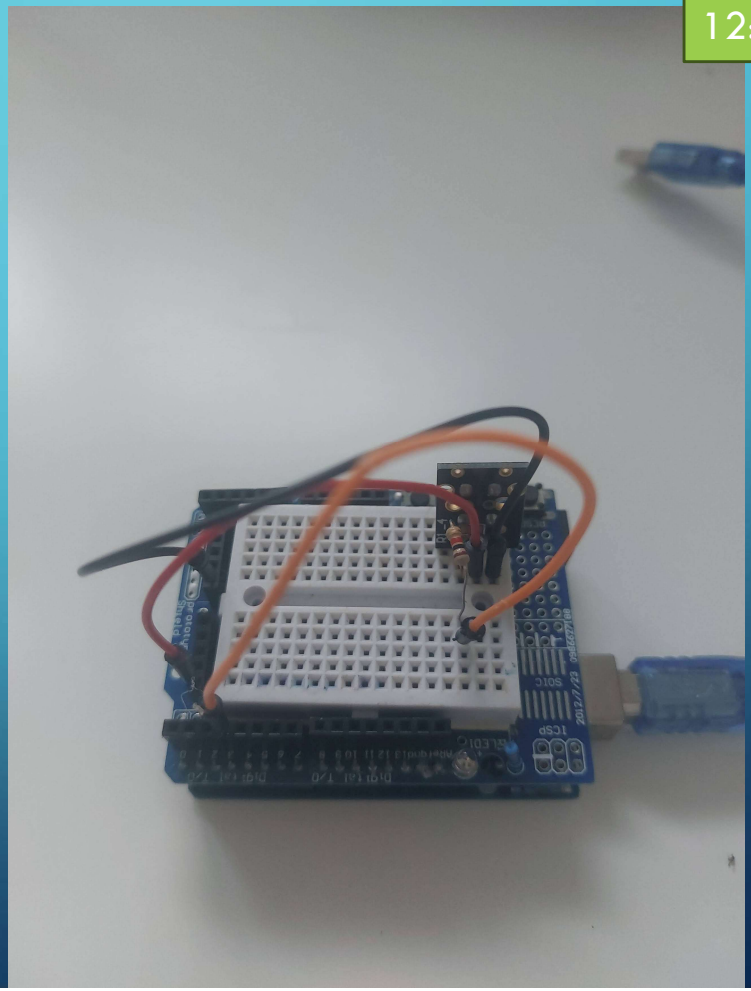
We pressed the Button 6 times, and the interrupt was called for each one of them , and our massage was printed to serial monitor-it means that It works correctly!

HW01.16. PROTOCOL OF TEST-TEST 2-HAWRDWARE-PART 1

Circuit: (from two angles of view)

Being done by:
ID : 2210
On : 9/3/2024
12:50 (Saturday morning)





HW01.16. PROTOCOL OF TEST-TEST 2-HAWRDWARE-PART 2

Being done by:
ID : 2210
On : 9/3/2024
12:50 (Saturday morning)

HW01_16_PROTOCOL_OF_TEST_TEST_2_HAWRDWARE.ino

```
1  int Button = 3;
2
3  void setup()
4  {
5    Serial.begin(9600);
6    pinMode(Button, INPUT);
7    attachInterrupt(digitalPinToInterrupt(Button),TestInterrupt,RISING);
8  }
9
10 void TestInterrupt() //when button is pressed
11 {
12   Serial.println("Interrupt is working successfully!");
13 }
14
15 void loop()
16 {
17
18 }
```

The order that creats the interrupt “attackInterrupt” – gets several parameters including the “rising” (in our case) parameter

The function “TestInterrupt” is being called every time that interrupt is being recorded – in order to print the message

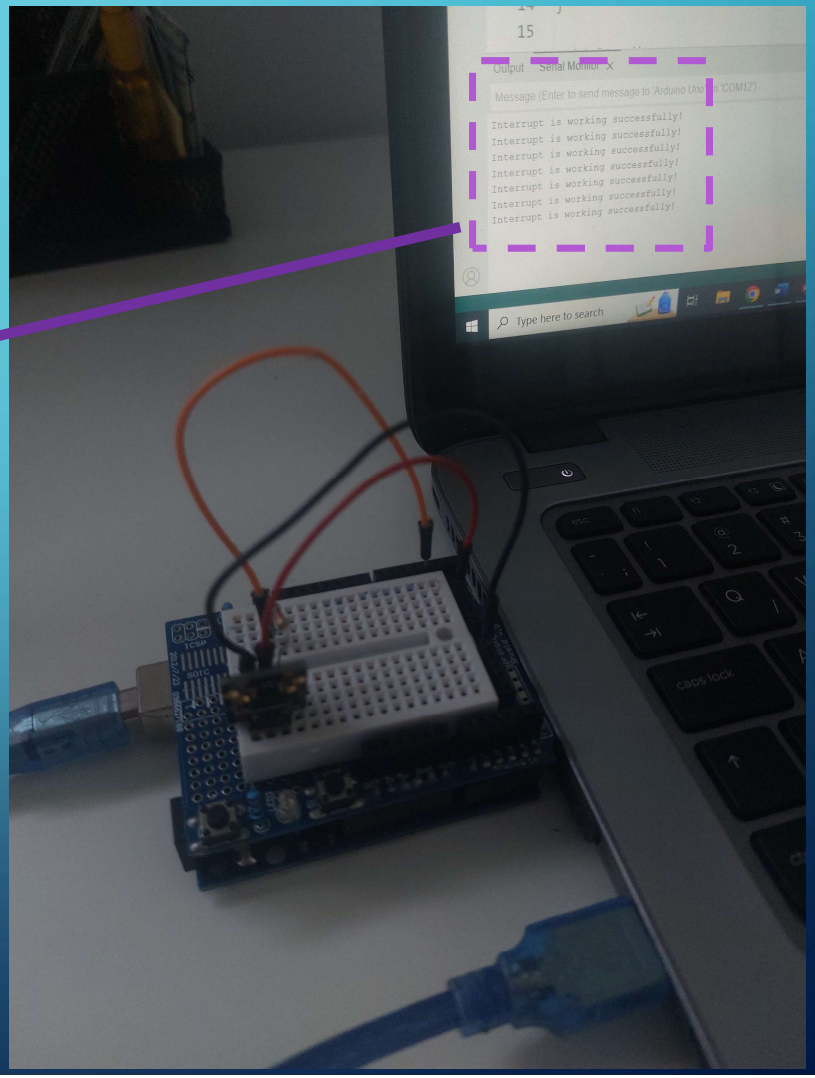
There is no need in the void loop function hence it is empty

HW01.16. PROTOCOL OF TEST-TEST 2-HAWRDWARE-PART 2

Result Hardware: (when pressing the button – the message arise in the Arduino serial monitor)

Being done by:
ID : 2210
On : 9/3/2024
13:00 (Saturday morning)

```
Interrupt is working successfully!
Interrupt is working successfully!
Interrupt is working successfully!
Interrupt is working successfully!
Interrupt is working successfully!
Interrupt is working successfully!
Interrupt is working successfully!
```



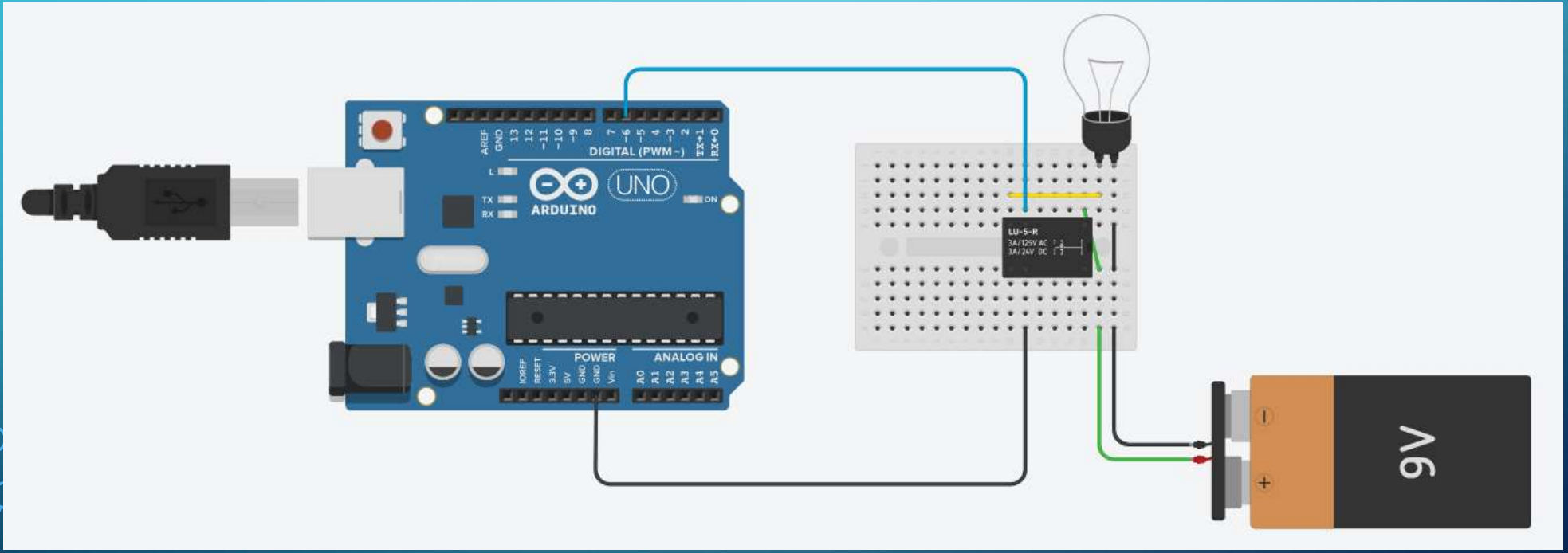
Conclusion : the button works good and ready for the whole project!

HW01.16. PROTOCOL OF TEST-TEST 3-SIMULATION-PART 1

Test 3: check the relay and light bulb

In this test we are going to control the relay. In order to see if the relay works, we connect a light bulb.

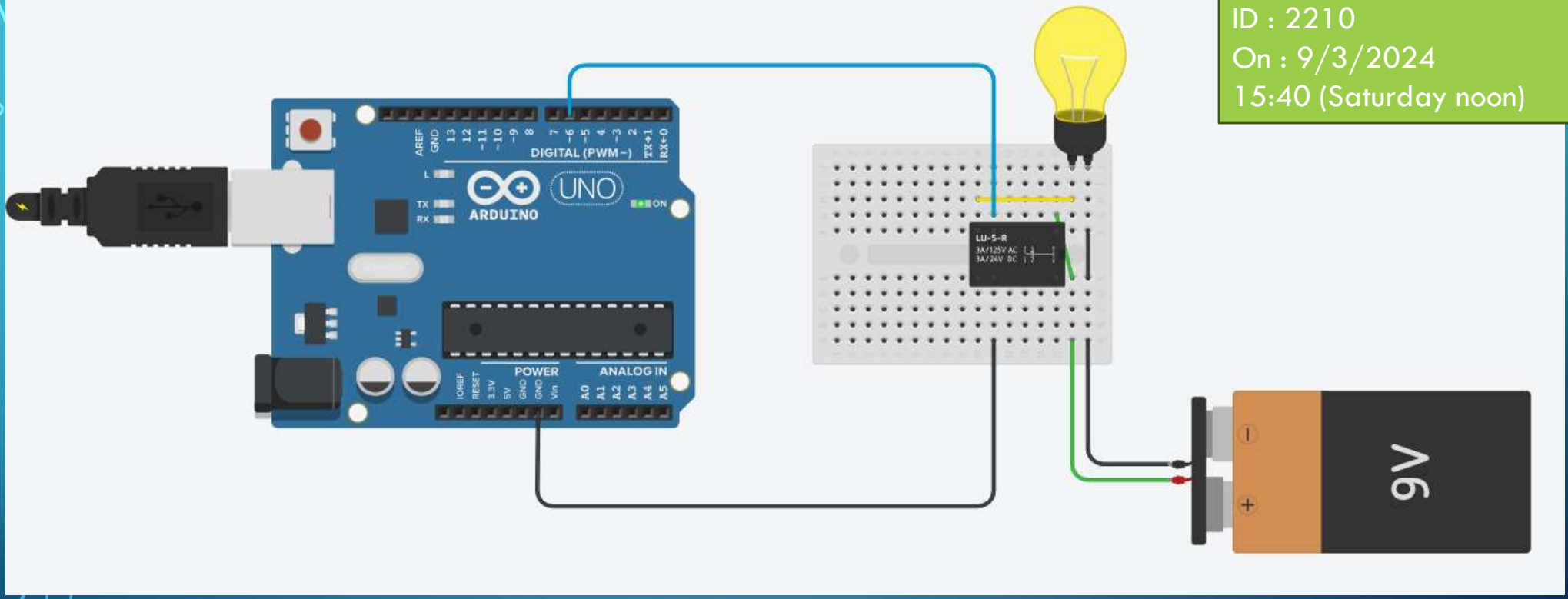
Being done by:
ID : 2210
On : 9/3/2024
15:40 (Saturday noon)



Link to TinkerCad to Test 3:
https://www.tinkercad.com/things/hhVTtMHuotx-hw0116-protocol-of-test-test-3-simulation

HW01.16. PROTOCOL OF TEST-TEST 3-SIMULATION-PART 2

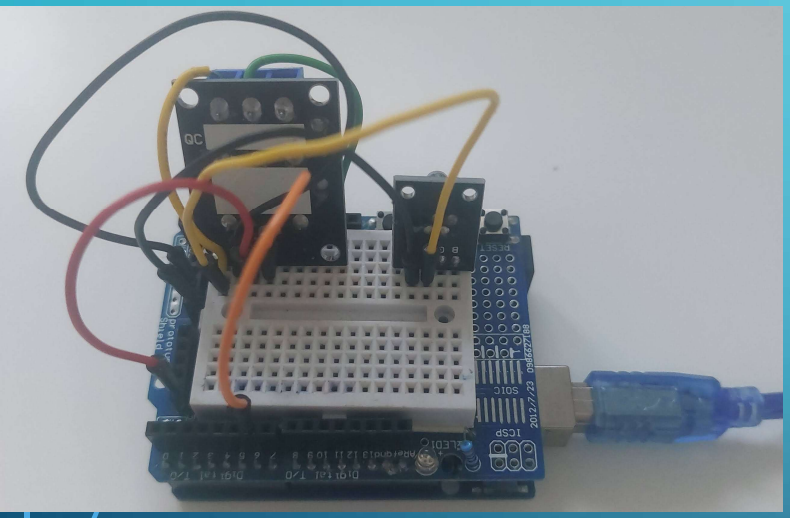
Being done by:
ID : 2210
On : 9/3/2024
15:40 (Saturday noon)

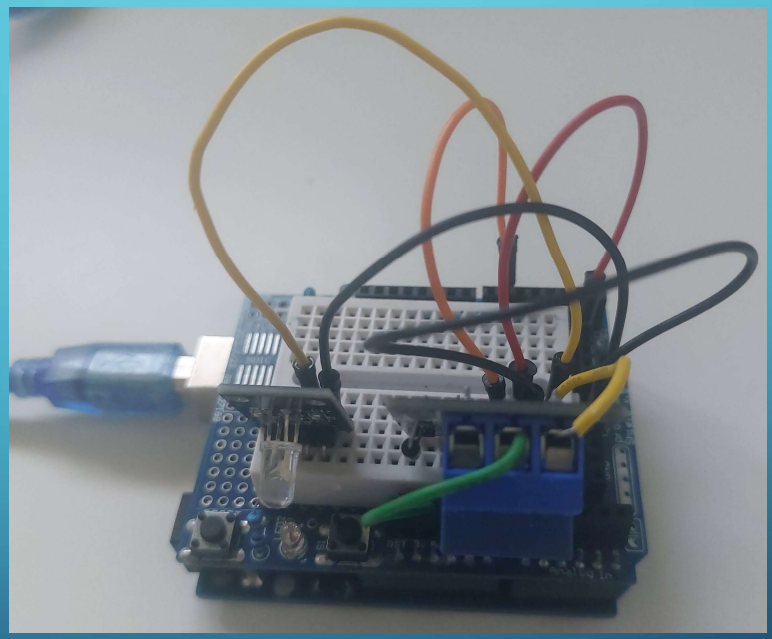


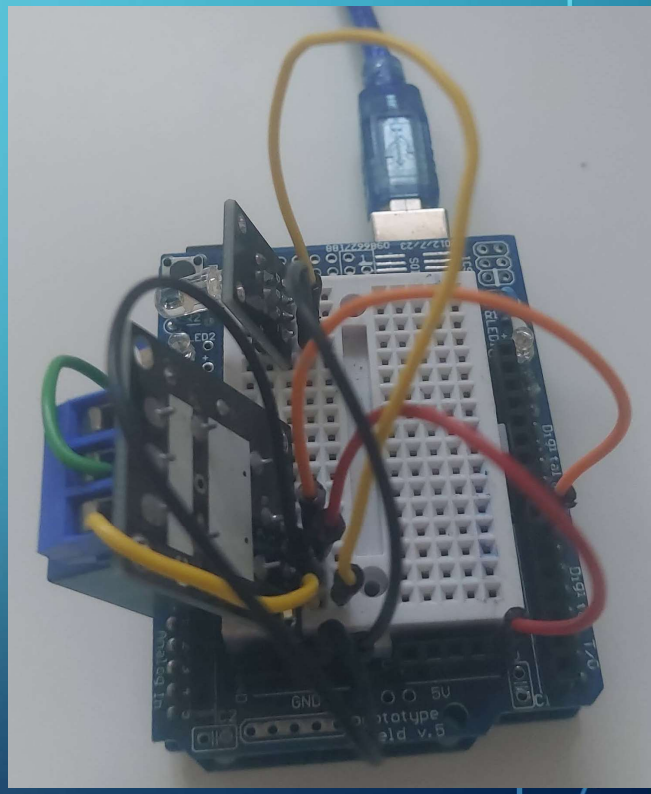
We basically wrote a simple code to turn on the bulb.
As we can see the light is turned on. Which means that the relay works properly.

HW01.16. PROTOCOL OF TEST-TEST 3-HARDWARE-PART 1

Circuit: (from three angles of view)





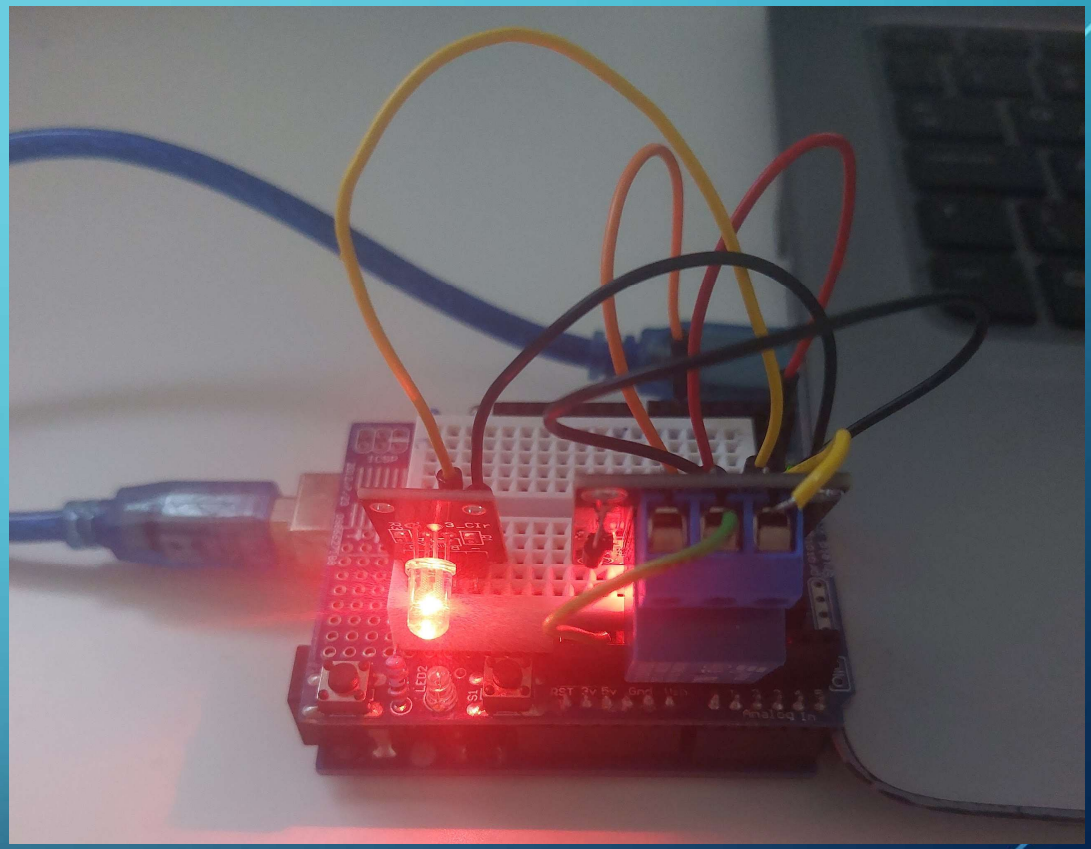


Being done by:
ID : 2210
On : 9/3/2024
15:50 (Saturday morning)

HW01.16. PROTOCOL OF TEST-TEST 3-HARDWARE-PART 2

HW01_16_PROTOCOL_OF_TEST_TEST_3_SIMULATION.ino

```
1   int RelayPin = 6;
2
3   void setup()
4   {
5     Serial.begin(9600);
6     pinMode(RelayPin, OUTPUT);
7     digitalWrite(RelayPin,HIGH);
8   }
9
10  void loop()
11  {
12  }
```



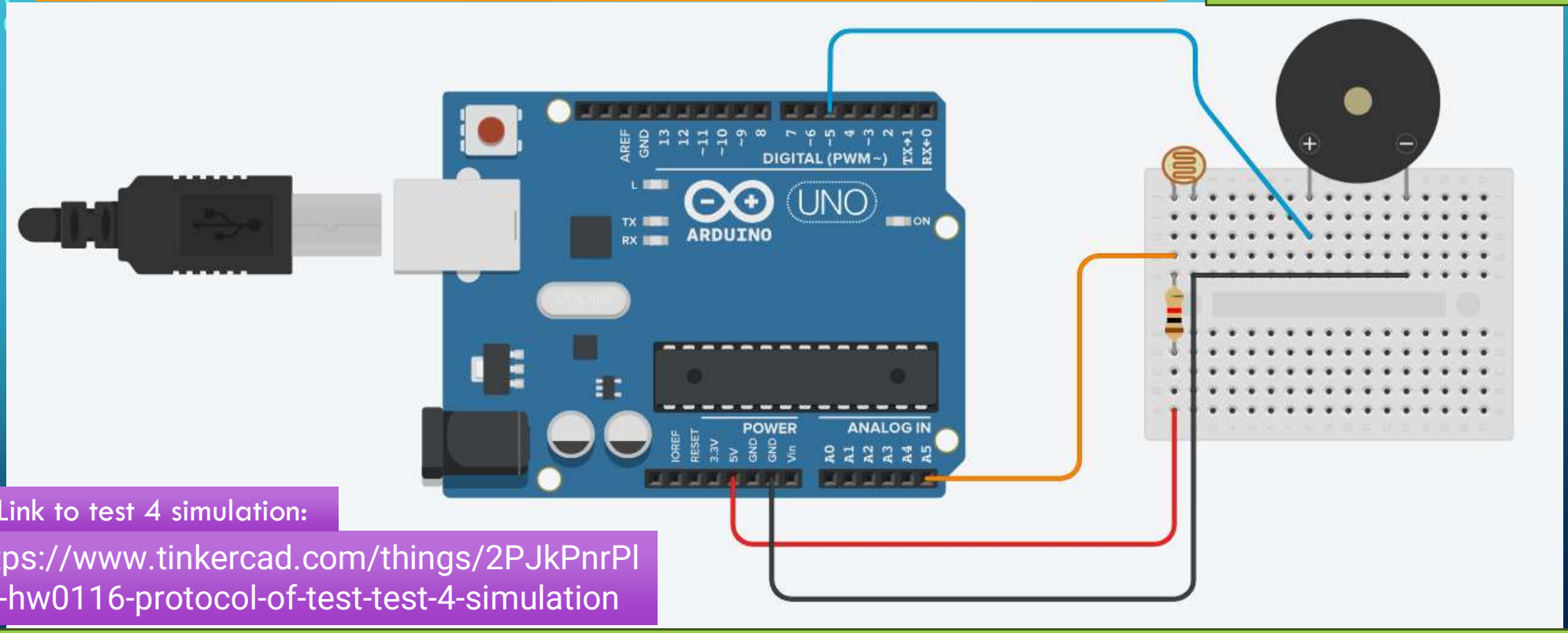
As we can see, the light bulb was turned on in red color so Test 3 works as expected!

HW01.16. PROTOCOL OF TEST-TEST 4-SIMULATION-PART 1

Test 4: check the buzzer together with the photoresistor

In this test we are going to check the behavior of the buzzer module - together with the photoresistor (that has been checked in test 1).

Being done by:
ID : 2210
On : 9/3/2024
17:30 (Saturday afternoon)

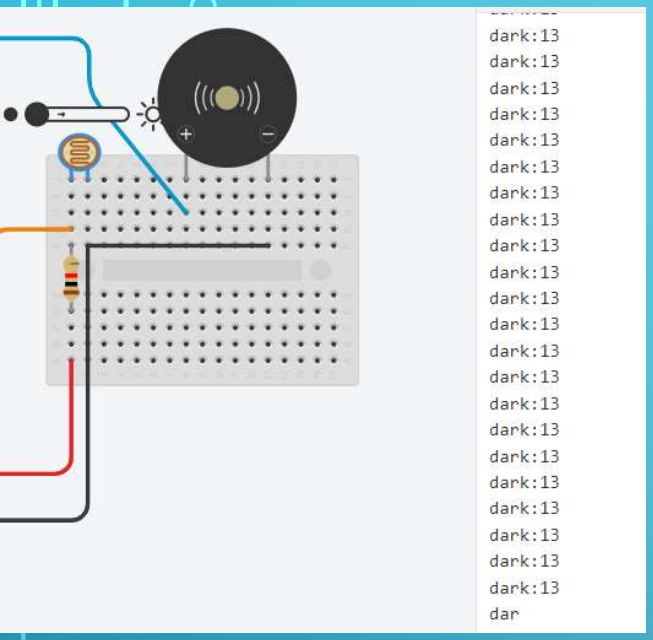


Link to test 4 simulation:

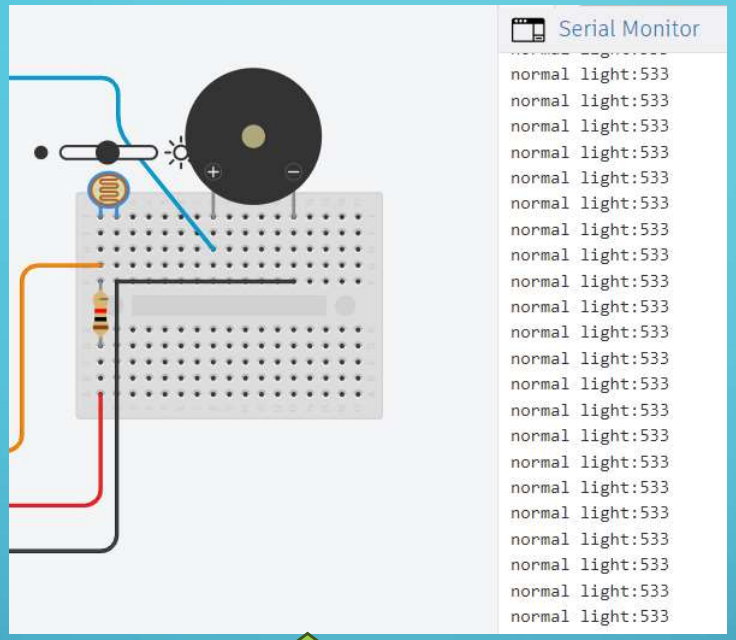
https://www.tinkercad.com/things/2PJkPnrPl3u-hw0116-protocol-of-test-test-4-simulation

the light intensity is measured and updated every 1000 milliseconds which is once every second.
In addition – in LOW light mode (L<L2) the buzzer "buzz" turned on and then off every 500 milliseconds

HW01.16. PROTOCOL OF TEST-TEST 4-SIMULATION-PART 2

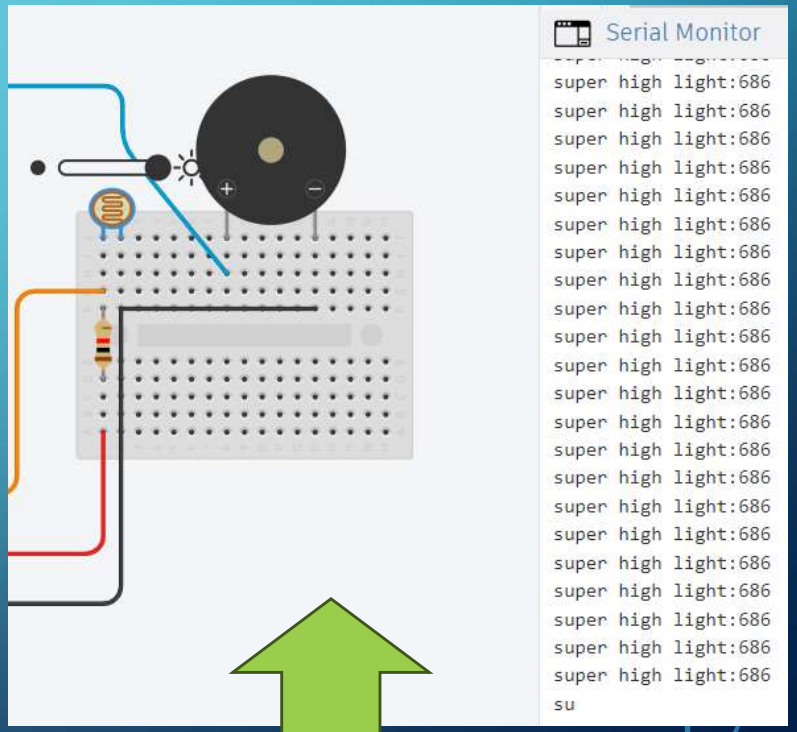


LOW LIGHT (L<L2) → THE BUZZER WORKS



NORMAL LIGHT (L2<L<L1) → THE BUZZER DOESN'T WORKS

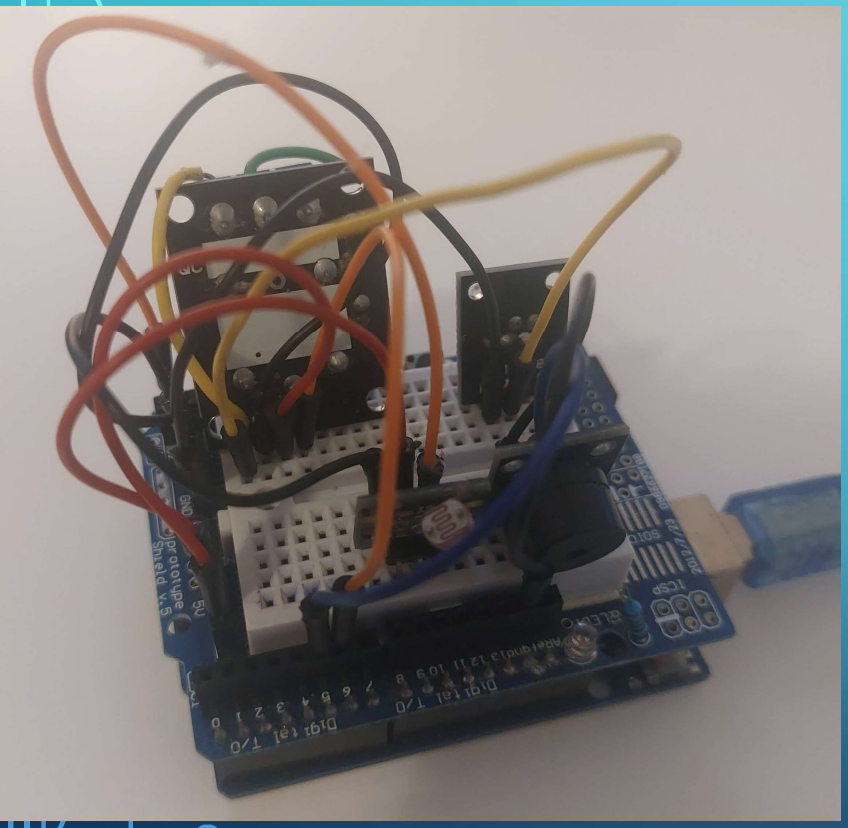
Being done by:
ID : 2210
On : 9/3/2024
17:30 (Saturday afternoon)

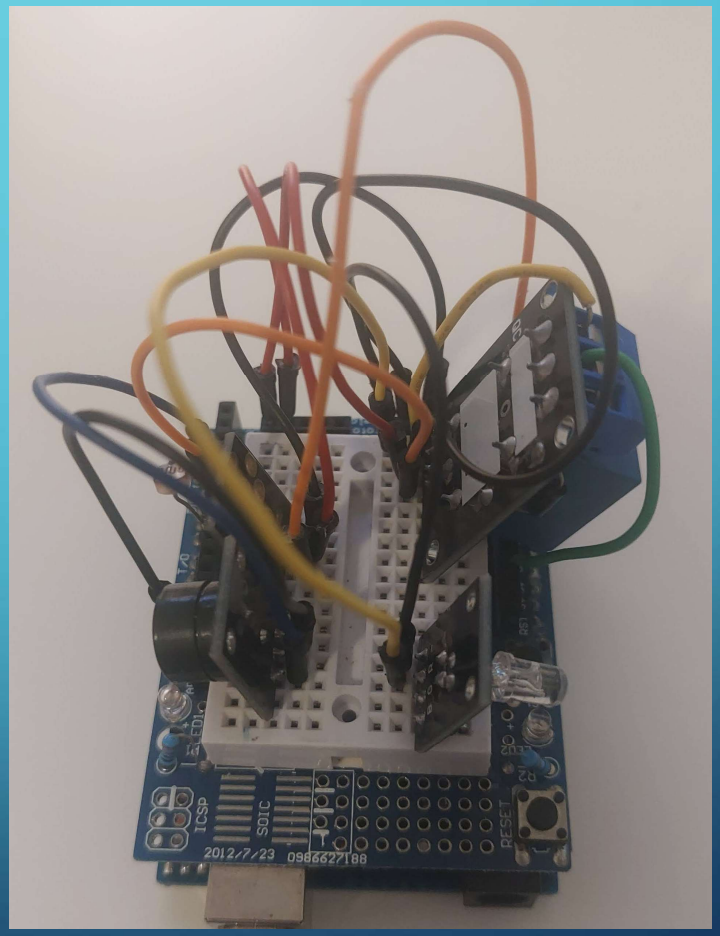


HIGH LIGHT (L>L1) → THE BUZZER DOESN'T WORKS

HW01.16. PROTOCOL OF TEST-TEST 4-HARDWARE-PART 1

Circuit: (from three angles of view)





Being done by:
ID : 2210
On : 9/3/2024
17:40 (Saturday afternoon)

Pay attention that in this test we only test the integration of the photoresistor and the buzzer. The relay and the bulb on the breadboard are preparation for future integration tests (ignore them for this step)

HW01.16. PROTOCOL OF TEST-TEST 4-HARDWARE-PART 2

```
/* flags */
bool NORMAL_STATE = false;
bool HIGH_STATE   = false;
bool LOW_STATE    = false;
bool TEST_MODE    = false;

/* time counter */
int timer_cnt = 0;
int timer_cnt_rst = 0;
int interuptcounter =0;
int sample=0;

/* photoresistor values */
int light_val = 0;
const int max_val = 600;
const int min_val = 100;

/* pins setup */
const int photoresistor_pin = A5;    //photoresistor to arduino
const int buzzer_pin = 5;        //buzzer to arduino pin
```

```
void setup()
{
  pinMode(buzzer_pin, OUTPUT); // Set buzzer - pin 11 as an output
  Serial.begin(9600);


  // TIMER 1 for interrupt frequency 16 Hz:
  cli(); // stop interrupts
  TCCR1A = 0; // set entire TCCR1A register to 0
  TCCR1B = 0; // same for TCCR1B
  TCNT1  = 0; // initialize counter value to 0
  // set compare match register for 16 Hz increments
  OCR1A = 15624; // = 16000000 / (64 * 16) - 1 (must be <65536)
  // turn on CTC mode
  TCCR1B |= (1 << WGM12);
  // Set CS12, CS11 and CS10 bits for 64 prescaler
  TCCR1B |= (0 << CS12) | (1 << CS11) | (1 << CS10);
  // enable timer compare interrupt
  TIMSK1 |= (1 << OCIE1A);
  sei(); // allow interrupts


}
```

HW01.16. PROTOCOL OF TEST-TEST 4-HARDWARE-PART 3

```
48  void loop()
49  {
50
51  }
52
53
54  ISR(TIMER1_COMPA_vect){
55    if(sample==16)
56    {
57      light_val = analogRead(photoresistor_pin);
58      sample=0;
59    }
60
61    else
62    {
63      sample++;
64    }
65      if(!TEST_MODE)
66      {
67        if((1030-light_val) < min_val)
68        {
69              NORMAL_STATE = false;
70              HIGH_STATE   = false;
71              LOW_STATE    = true;
72              TEST_MODE    = false;
73      }
74      else if((1030-light_val) > max_val)
75      {
76              NORMAL_STATE = false;
77              HIGH_STATE   = true;
78              LOW_STATE    = false;
79              TEST_MODE    = false;
80      }
81      else
82      {
83              NORMAL_STATE = true;
84              HIGH_STATE   = false;
85              LOW_STATE    = false;
86              TEST_MODE    = false;
87      }
88
89
90
```

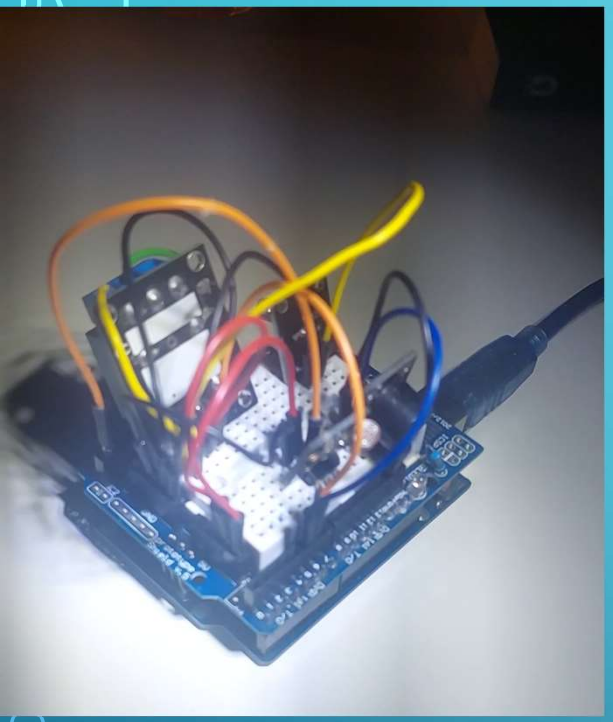
HW01.16. PROTOCOL OF TEST-TEST 4-HARDWARE-PART 4

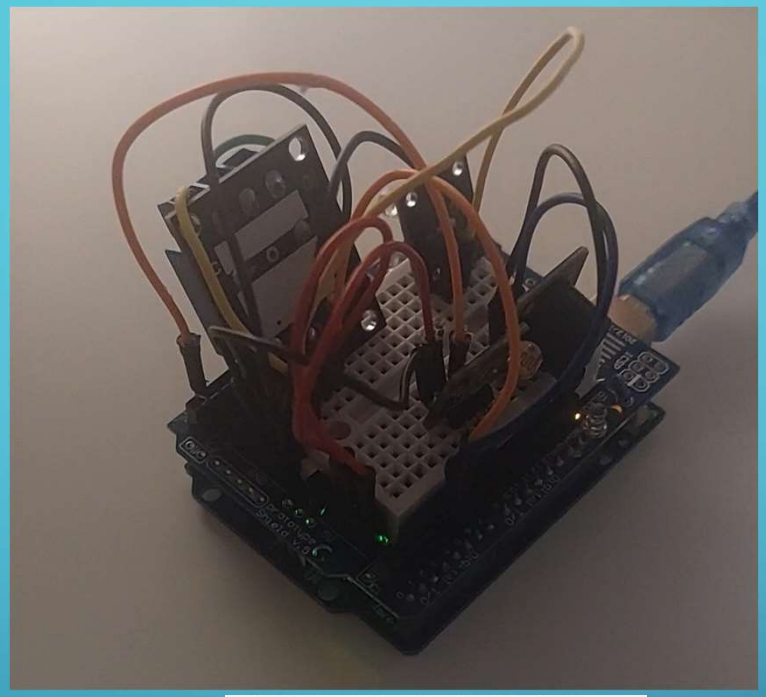
```
91      if(LOW_STATE)
92      {
93        Serial.print("dark:\n");
94        Serial.println(1030-light_val);
95
96        if(timer_cnt == 8)
97        {
98          tone(buzzer_pin, 1000); // Send 1 KHz sound signal
99          timer_cnt=0;
100       }
101       else
102       {
103         noTone(buzzer_pin);     // Stop sound
104         timer_cnt++;
105       }
106
107     }
108
109     else if(HIGH_STATE)
110     {
111       Serial.print("super high light:\n");
112       Serial.println(1030-light_val);
113       noTone(buzzer_pin);
```

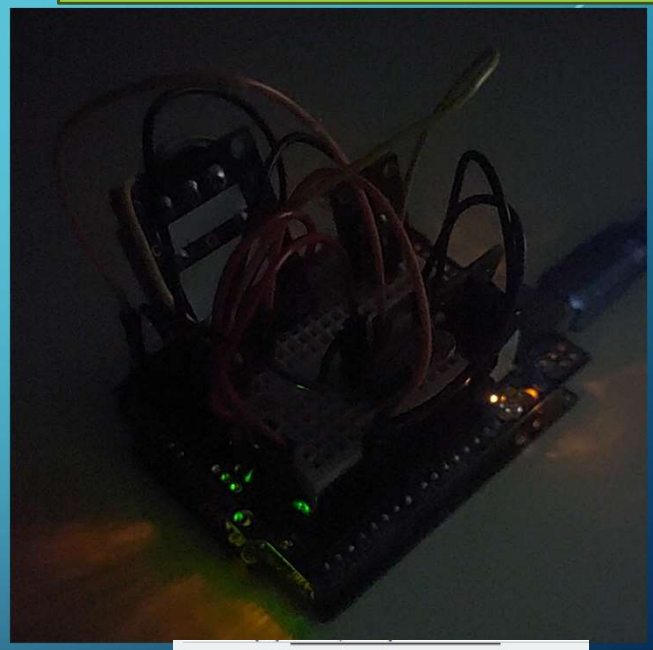
```
115       }
116
117       else if(NORMAL_STATE)
118       {
119         Serial.print("normal light:\n");
120         Serial.println(1030-light_val);
121         noTone(buzzer_pin);
122       }
123     }
124
125       else if(timer_cnt_rst==4)
126         {
127           tone(buzzer_pin, 1500); // Send 1.5KHz sound signal
128           timer_cnt_rst=0;
129         }
130         else
131         {
132           noTone(buzzer_pin);     // Stop sound
133           timer_cnt_rst++;
134         }
135
136
137   }
```

HW01.16. PROTOCOL OF TEST-TEST 4-HARDWARE-PART 5

Being done by:
ID : 2210
On : 9/3/2024
17:45 (Saturday afternoon)







Output Serial Monitor ×

Message (Enter to send message t

```
super high light:843
super high light:843
super high light:843
super high light:843
super high light:843
super high light:843
super high light:843
super high lig
```

IN THIS DEMONSTRATION PLEASE TURN ON YOUR HEADPHONES TO HEAR THE "BUZZING" OF THE BUZZER

Output Serial Monitor ×

Message (Enter to send messag

```
normal light:150
normal light:150
normal light:150
normal light:150
normal light:150
normal light:150
normal light:150
normal lig
```

ONLY IN THE DARK MODE (L2<L) the buzzer works

Output Serial Monitor ×

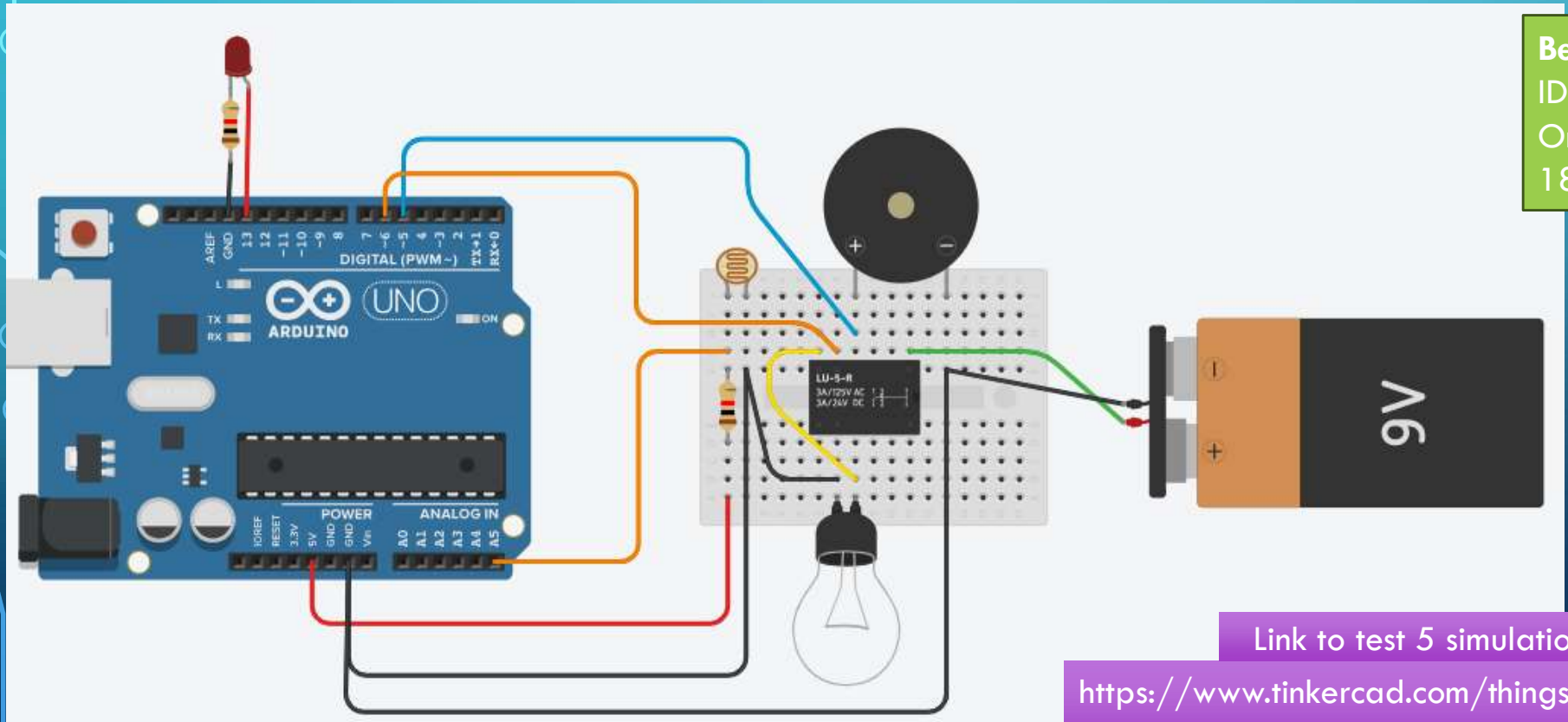
Message (Enter to send message to

```
dark:20
dark:20
dark:20
dark:20
dark:20
dark:20
dark:20
dark:
```

HW01.16. PROTOCOL OF TEST-TEST 5-SIMULATION-PART 1

Test 5: check the buzzer together with the photoresistor in addition to the relay , the bulb and the LED

In this test we are going to check the behavior of the buzzer module - together with the photoresistor integrated with the relay-bulb subsystem and the LED



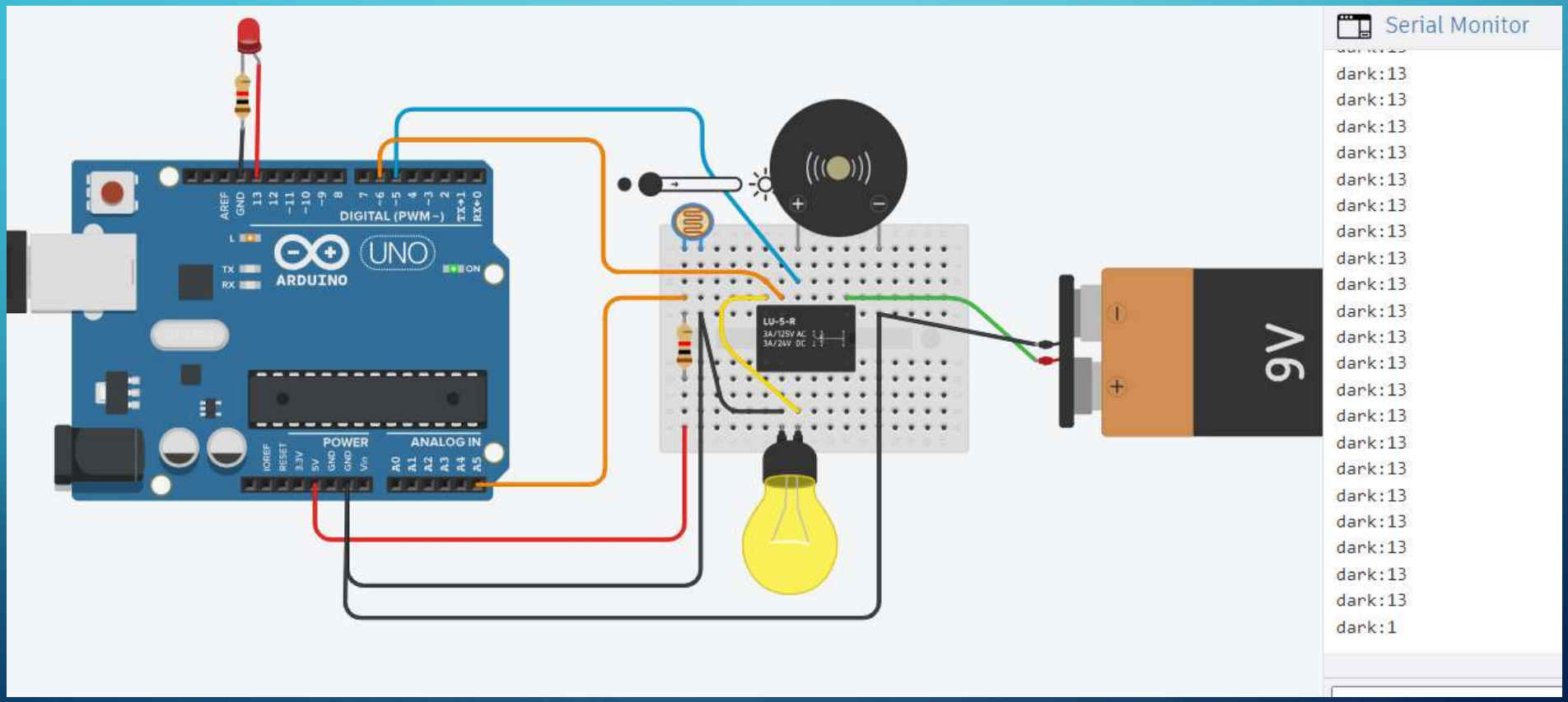
Being done by:
ID : 2210
On : 9/3/2024
18:30 (Saturday evening)

Link to test 5 simulation:
https://www.tinkercad.com/things/iMiCZcgZToQ-hw0116-protocol-of-test-test-5-simulation

HW01.16. PROTOCOL OF TEST-TEST 5-SIMULATION-PART 1

Being done by:
ID : 2210
On : 9/3/2024
18:30 (Saturday evening)

First mode-LOW LIGHT MODE- L<L2

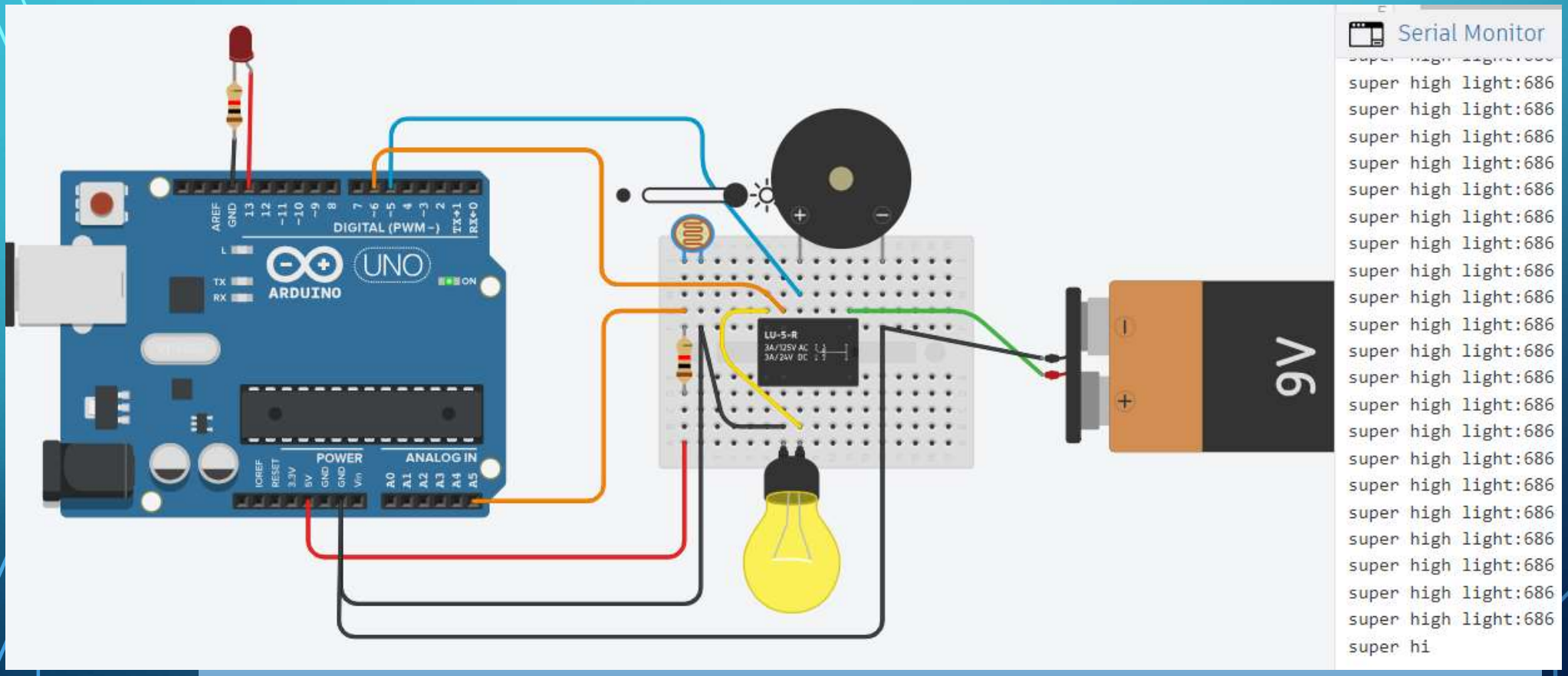


ALL THE NECCECERY PARAMETERS WORKS GOOD – LED 13 , BULB , BUZZER AND LIGHT MEASURE

HW01.16. PROTOCOL OF TEST-TEST 5-SIMULATION-PART 2

Being done by:
ID : 2210
On : 9/3/2024
18:35 (Saturday evening)

SECOND MODE-SUPER HIGH LIGHT MODE- L>L1

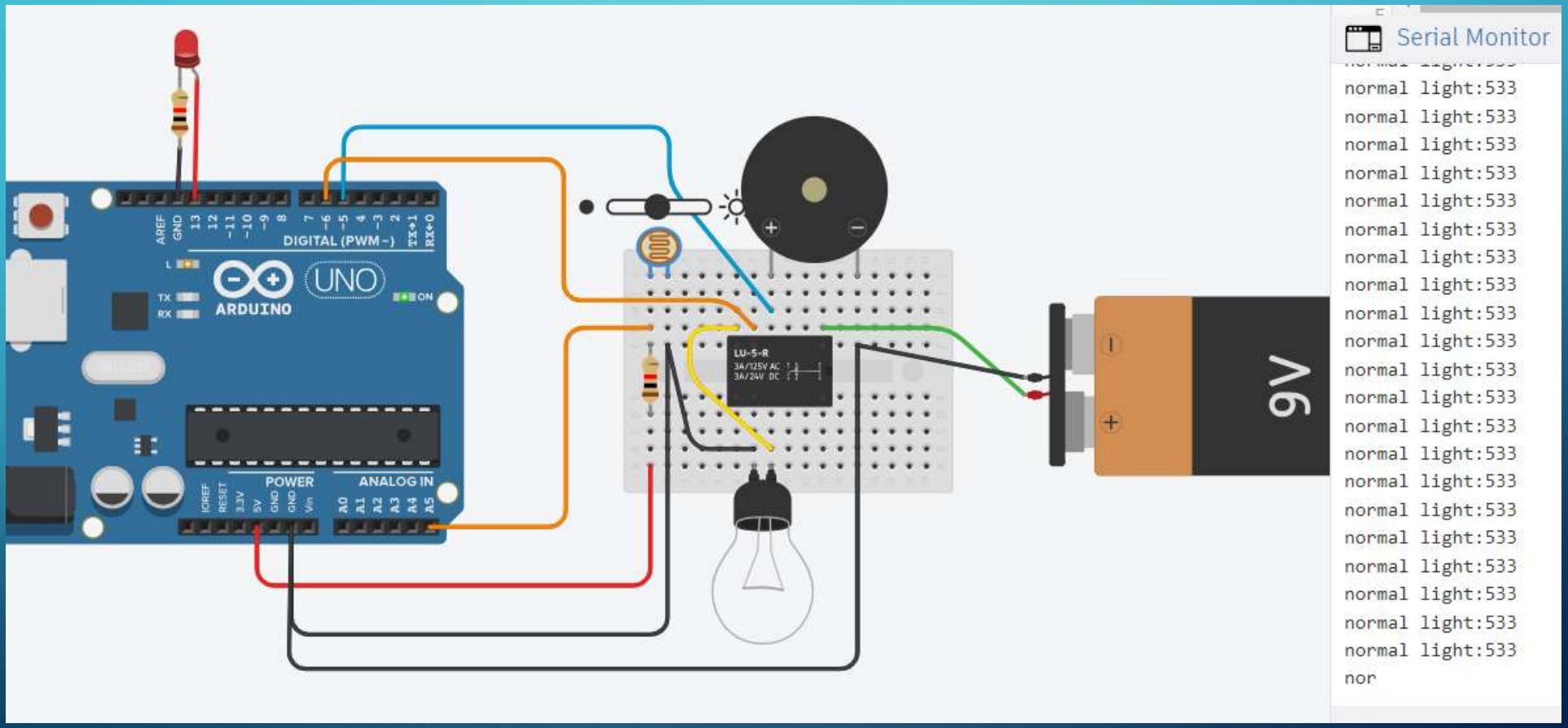


ALL THE NECCECERY PARAMETERS WORKS GOOD – LED 13 , BULB , BUZZER AND LIGHT MEASURE

HW01.16. PROTOCOL OF TEST-TEST 5-SIMULATION-PART 3

Being done by:
ID : 2210
On : 9/3/2024
18:40 (Saturday evening)

THIRD MODE- NORMAL LIGHT MODE- L2<L<L1

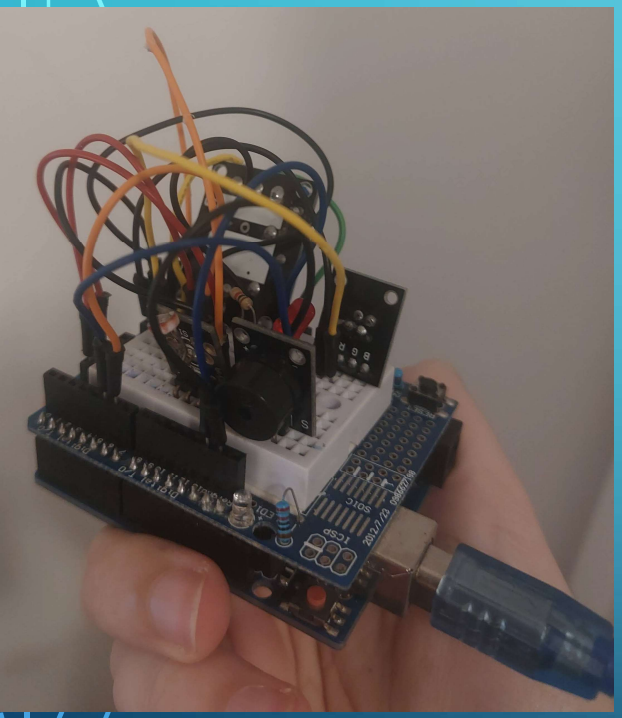


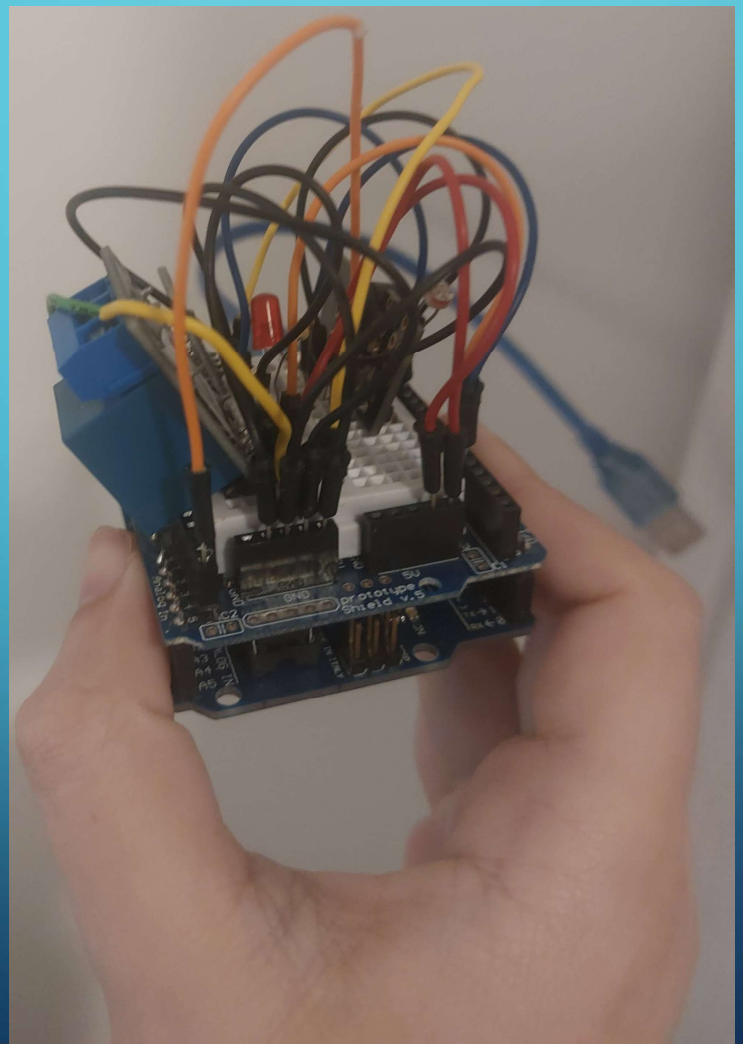
ALL THE NECCECERY PARAMETERS WORKS GOOD – LED 13 , BULB , BUZZER AND LIGHT MEASURE

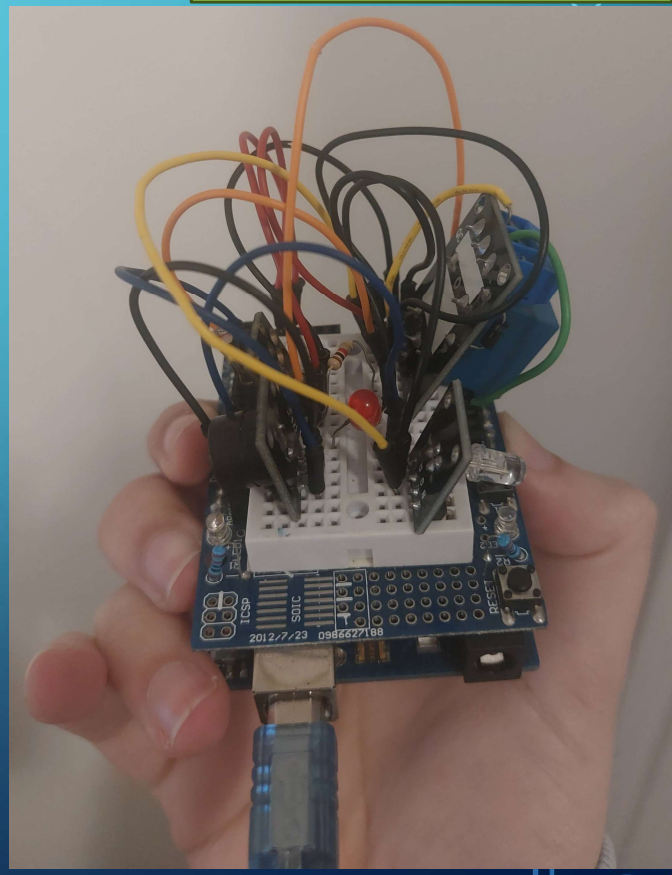
HW01.16. PROTOCOL OF TEST-TEST 5-HARDWARE-PART 1

Being done by:
ID : 2210
On : 9/3/2024
18:45 (Saturday evening)

Circuit: (from three angles of view)







HW01.16. PROTOCOL OF TEST-TEST 5-HARDWARE-PART 2

```
/* flags */
bool NORMAL_STATE = false;
bool HIGH_STATE   = false;
bool LOW_STATE    = false;
bool TEST_MODE    = false;

/* time counter */
int timer_cnt = 0;
int timer_cnt_rst = 0;
int interuptcounter =0;
int sample=0;

/* photoresistor values */
int light_val = 0;
const int max_val = 600;
const int min_val = 100;

/* pins setup */
const int photoresistor_pin = A5;    //photoresistor to arduino
const int buzzer_pin = 5;        //buzzer to arduino pin
const int relay_pin  = 6;
const int led_pin    = 13;
```

```
void setup()
{
  pinMode(buzzer_pin, OUTPUT); // Set buzzer - pin 11 as an output
  pinMode(relay_pin, OUTPUT);  // Set relay  - pin 12 as an output
  pinMode(led_pin, OUTPUT);    // Set led    - pin 13 as an output
  Serial.begin(9600);


  // TIMER 1 for interrupt frequency 16 Hz:
  cli(); // stop interrupts
  TCCR1A = 0; // set entire TCCR1A register to 0
  TCCR1B = 0; // same for TCCR1B
  TCNT1  = 0; // initialize counter value to 0
  // set compare match register for 16 Hz increments
  OCR1A = 15624; // = 16000000 / (64 * 16) - 1 (must be <65536)
  // turn on CTC mode
  TCCR1B |= (1 << WGM12);
  // Set CS12, CS11 and CS10 bits for 64 prescaler
  TCCR1B |= (0 << CS12) | (1 << CS11) | (1 << CS10);
  // enable timer compare interrupt
  TIMSK1 |= (1 << OCIE1A);
  sei(); // allow interrupts
}
```

HW01.16. PROTOCOL OF TEST-TEST 5-HARDWARE-PART 3

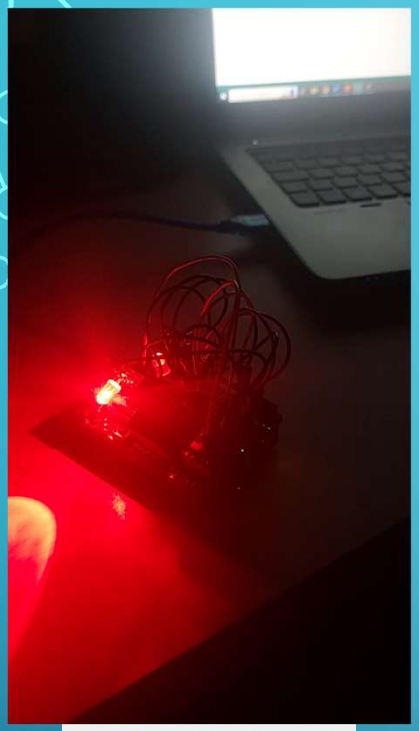
```
49  void loop()
50  {
51
52  }
53
54
55  ISR(TIMER1_COMPA_vect){
56    if(sample==16)
57    {
58      light_val = analogRead(photoresistor_pin);
59      sample=0;
60    }
61
62    else
63    {
64      sample++;
65    }
66      if(!TEST_MODE)
67      {
68        if((1030-light_val) < min_val)
69        {
70            NORMAL_STATE = false;
71            HIGH_STATE   = false;
72            LOW_STATE    = true;
73            TEST_MODE    = false;
74        }
75        else if((1030-light_val) > max_val)
76        {
77            NORMAL_STATE = false;
78            HIGH_STATE   = true;
79            LOW_STATE    = false;
80            TEST_MODE    = false;
81        }
82        else
83        {
84            NORMAL_STATE = true;
85            HIGH_STATE   = false;
86            LOW_STATE    = false;
87            TEST_MODE    = false;
88        }
```

HW01.16. PROTOCOL OF TEST-TEST 5-HARDWARE-PART 4


```
92      if(LOW_STATE)
93      {
94        Serial.print("dark:\n");
95        Serial.println(1030-light_val);
96        digitalWrite(relay_pin, HIGH); //turns high light bulb
97        digitalWrite(led_pin, HIGH);    //turns high led
98
99        if(timer_cnt == 8)
100       {
101         tone(buzzer_pin, 1000); // Send 1 KHz sound signal
102         timer_cnt=0;
103       }
104       else
105       {
106           noTone(buzzer_pin);     // Stop sound
107           timer_cnt++;
108        }
109
110     }
```

```
112     else if(HIGH_STATE)
113     {
114       Serial.print("super high light:\n");
115       Serial.println(1030-light_val);
116       noTone(buzzer_pin);
117       digitalWrite(relay_pin, HIGH); //turns ON light bulb
118       digitalWrite(led_pin, LOW);    //turns off led
119
120     }
121
122     else if(NORMAL_STATE)
123     {
124       Serial.print("normal light:\n");
125       Serial.println(1030-light_val);
126       noTone(buzzer_pin);
127       digitalWrite(relay_pin, LOW); //turns OFF light bulb
128       digitalWrite(led_pin, HIGH);  //turns ON led
129     }
130   }
132     else if(timer_cnt_rst==4)
133       {
134         tone(buzzer_pin, 1500); // Send 1.5KHz sound signal
135         timer_cnt_rst=0;
136       }
137       else
138       {
139         noTone(buzzer_pin);     // Stop sound
140         timer_cnt_rst++;
141       }
142
143
144 }
```

HW01.16. PROTOCOL OF TEST-TEST 5-HARDWARE-PART 6



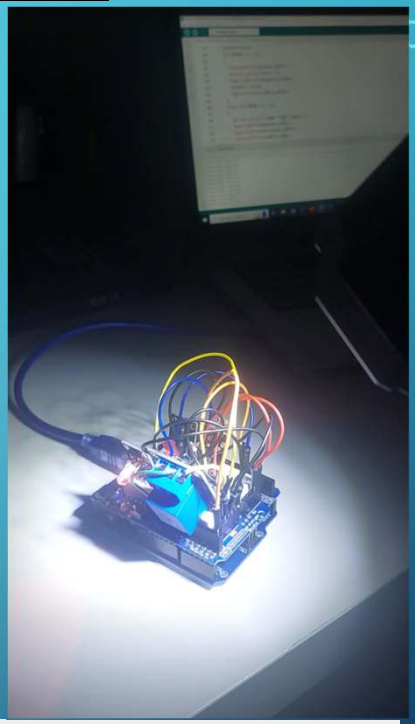
Output Serial Monitor ×

Message (Enter to send m

```
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
d
```

THESE ARE SHORT VIDEOS TO DEMONSTRATE THE TEST-PLEASE TURN ON YOUR AUDIO IN THE PC

```
normal light:177
normal light:177
normal light:177
normal light:177
normal light:177
normal light:177
normal light:177
normal light:177
normal light:177
normal light:177
```

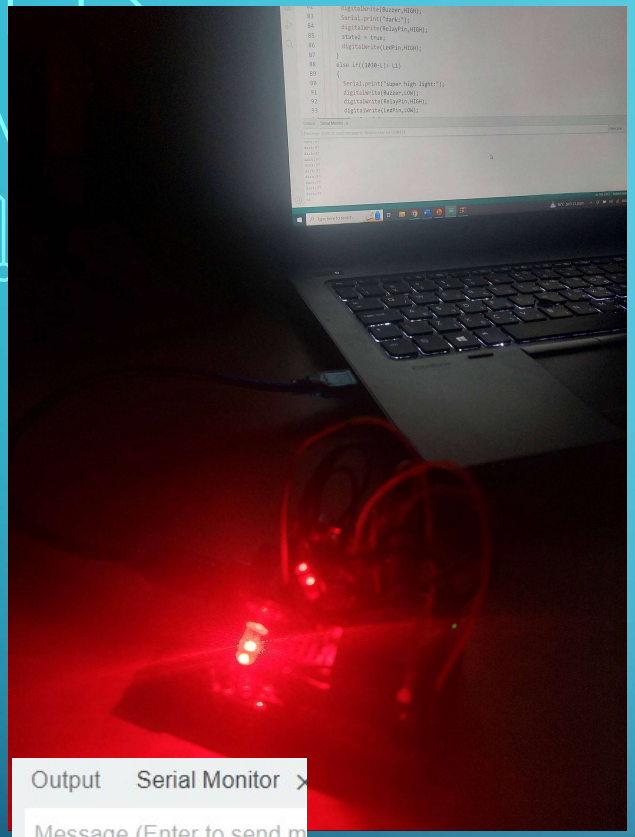


Output Serial Monitor ×

Message (Enter to send message to

```
super high light:986
super high light:986
super high light:986
super high light:986
super high light:986
super high light:986
super high light:986
super high light:986
super high light:986
super high light:986
su
```

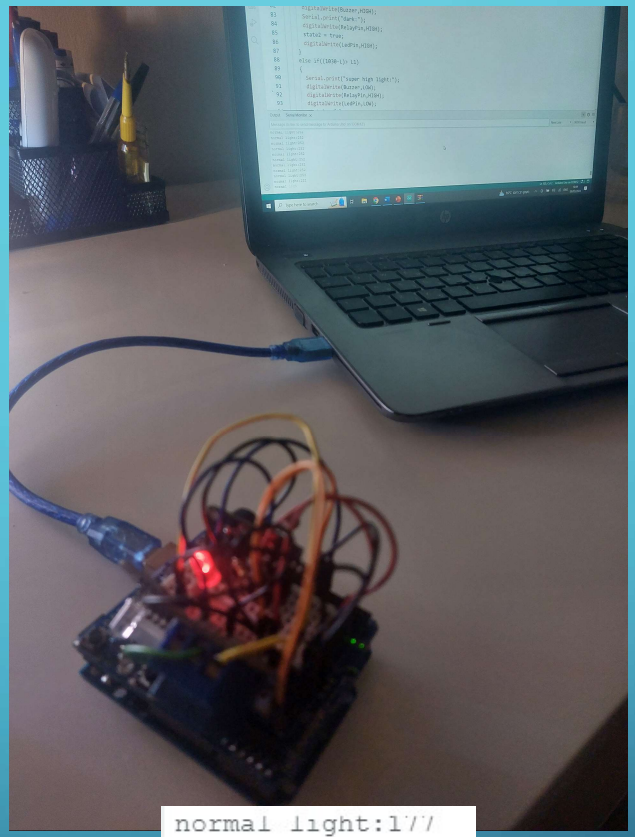
HW01.16. PROTOCOL OF TEST-TEST 5-HARDWARE-PART 7



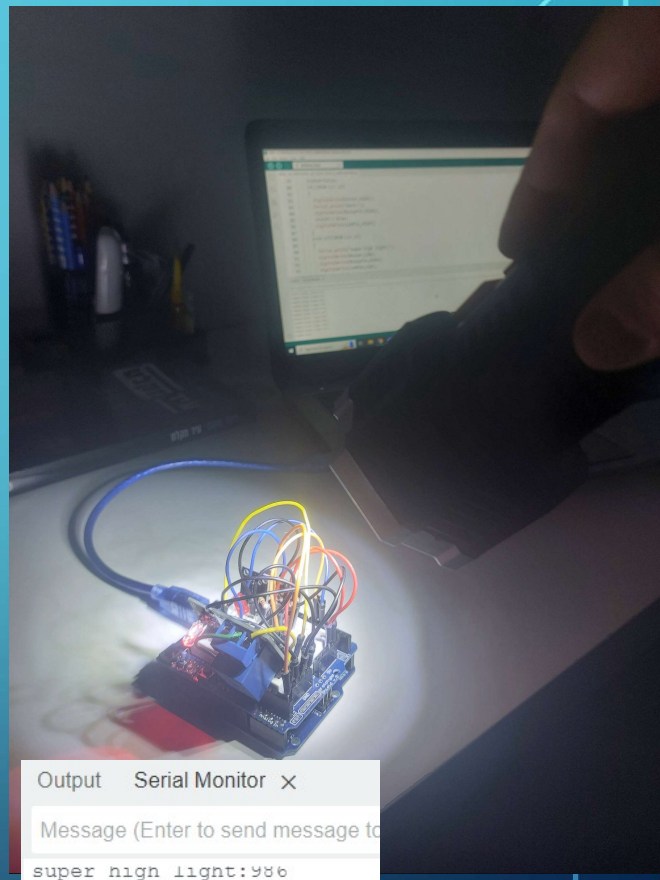
Output Serial Monitor ×

Message (Enter to send m

```
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
```



```
normal light:177
normal light:177
normal light:177
normal light:177
normal light:177
normal light:177
normal light:177
normal light:177
normal light:177
normal light:177
```



Output Serial Monitor ×

Message (Enter to send message to

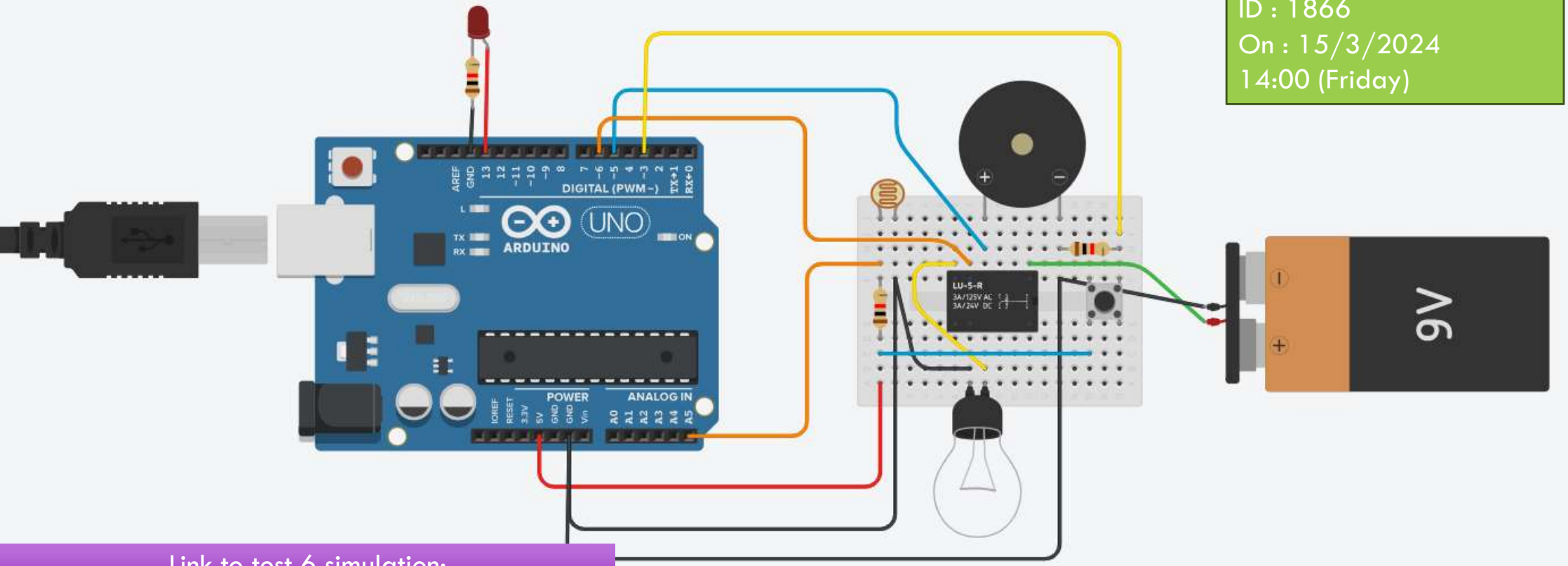
```
super high light:986
super high light:986
super high light:986
super high light:986
super high light:986
super high light:986
super high light:986
super high light:986
super high light:986
super high light:986
su
```

HW01.16. PROTOCOL OF TEST-TEST 6-FINAL TEST-SIMULATION-PART 1

IN THE Final Test We integrated all the modules to test them altogether and check that everything works as expected.

The addition is the bottom module

Being done by:
ID : 1866
On : 15/3/2024
14:00 (Friday)

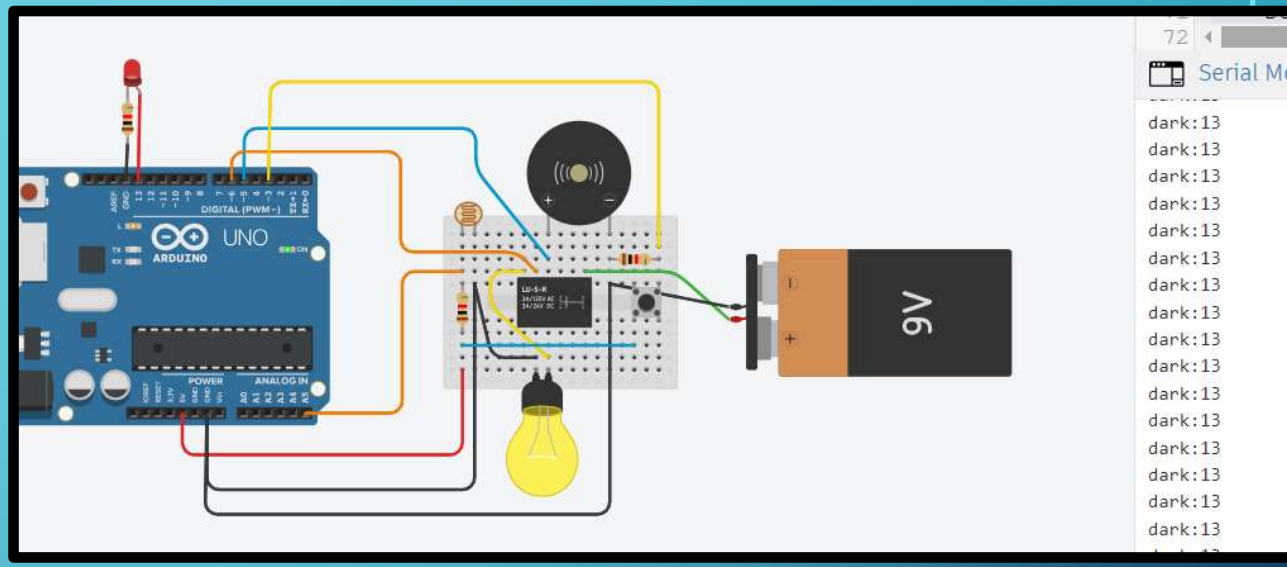


Link to test 6 simulation:

https://www.tinkercad.com/things/1RQUUYt9YRy-hw0116protocoloftesttest6simulation

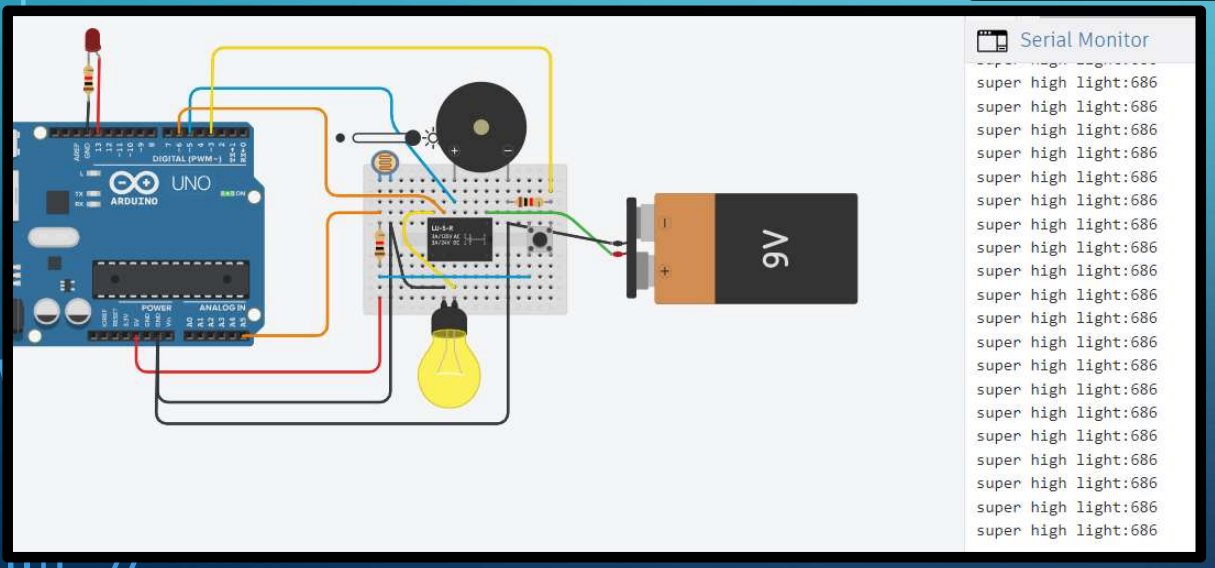
HW01.16. PROTOCOL OF TEST-TEST 6-FINAL TEST-SIMULATION-PART 2

Being done by:
ID : 1866
On : 15/3/2024
14:00 (Friday)

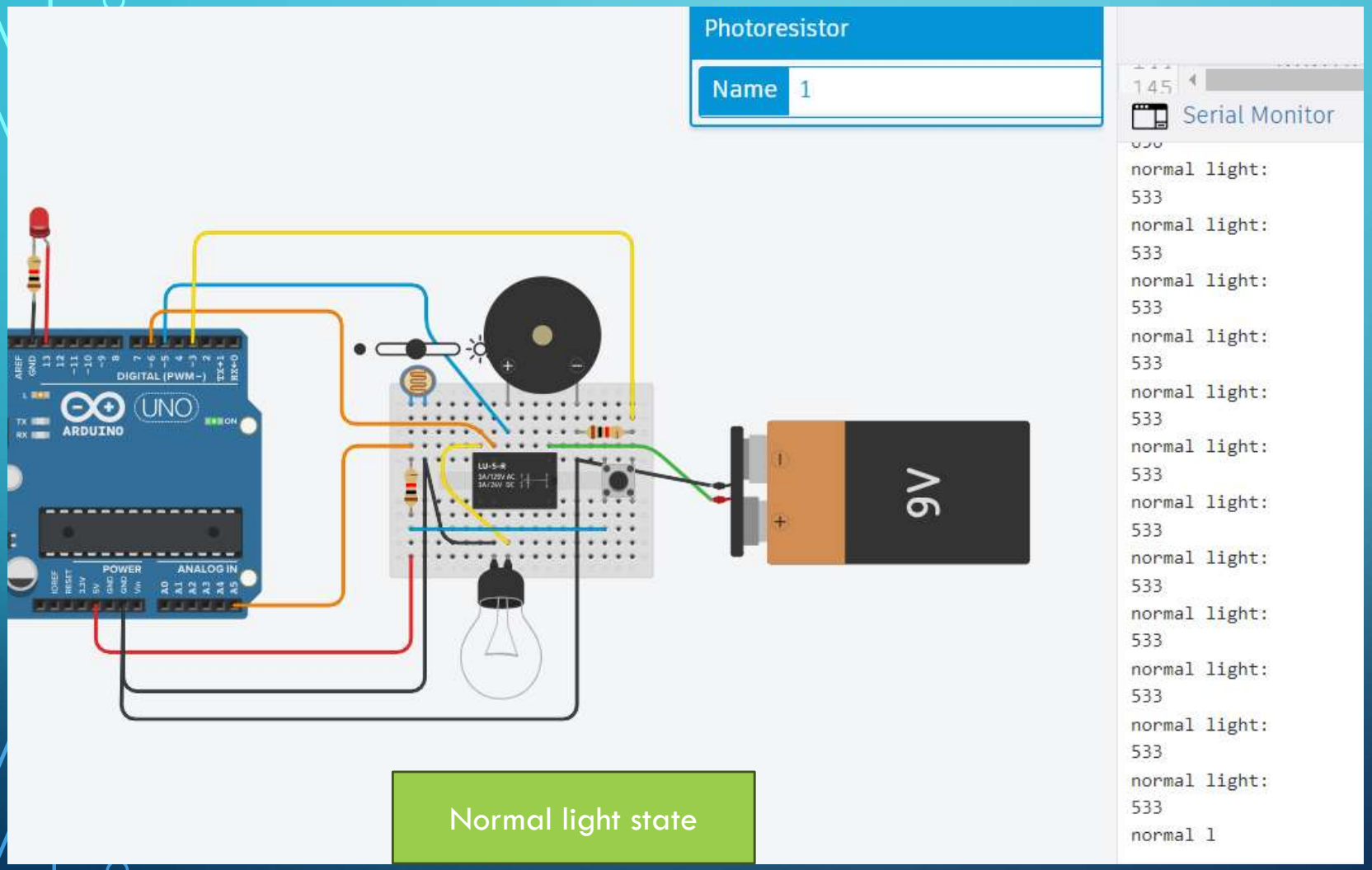


Low light (dark) state

Super high light state



HW01.16. PROTOCOL OF TEST-TEST 6-FINAL TEST-SIMULATION-PART 3

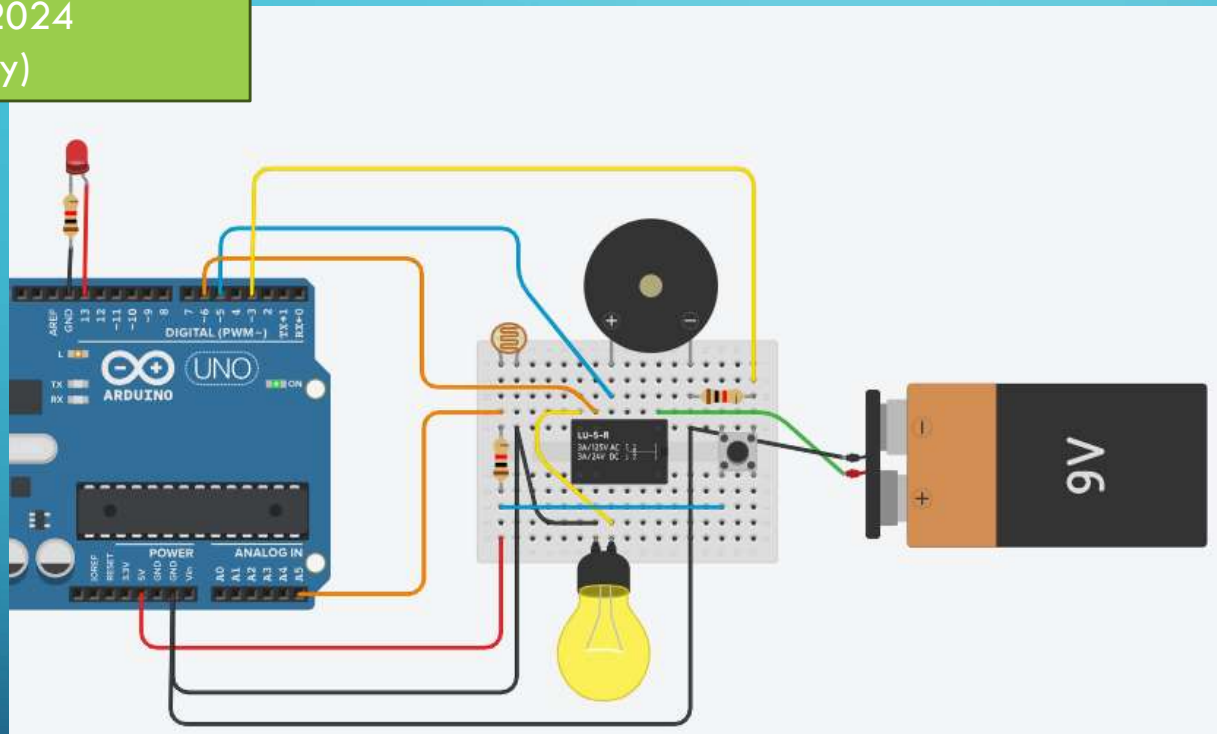


Normal light state

Being done by:
ID : 1866
On : 15/3/2024
14:00 (Friday)

HW01.16. PROTOCOL OF TEST-TEST 6-FINAL TEST-SIMULATION-PART 4

Being done by:
ID : 1866
On : 15/3/2024
14:00 (Friday)



Serial Monitor

```
dark:
13
dark:
13
dark:
13
dark:
13
dark:
13
dark:
13
dark:
13
dark:
13
dark:
13
dark:
13
dark:
13
dark:
13
dark:
13
dark:
13
button has been pressed
```

Stop measuring light and acting in test mode

Test mode

HW01.16. PROTOCOL OF TEST-TEST 6-FINAL TEST-code both for simulation and reality – part 1

```
/* flags */
bool NORMAL_STATE = false;
bool HIGH_STATE   = false;
bool LOW_STATE    = false;
bool TEST_MODE    = false;

/* time counter */
int timer_cnt = 0;
int timer_cnt_rst = 0;
int interuptcounter =0;
int sample=0;

/* photoresistor values */
int light_val = 0;
const int max_val = 600;
const int min_val = 100;

/* pins setup */
const int photoresistor_pin = A5;   //photoresistor to arduino
const int button_pin = 3;     //button_pin
const int buzzer_pin = 5;       //buzzer to arduino pin
const int relay_pin  = 6;
const int led_pin    = 13;
```

Being done by:
ID : 2210
On : 16/3/2024
16:00 (Saturday)

HW01.16. PROTOCOL OF TEST-TEST 6-FINAL TEST-code both for simulation and reality – part 2

```
26  void setup()
27  {
28    pinMode(button_pin, INPUT);  // Set BUTTON - pin 3 as an input
29    pinMode(buzzer_pin, OUTPUT); // Set buzzer - pin 11 as an output
30    pinMode(relay_pin, OUTPUT);  // Set relay  - pin 12 as an output
31    pinMode(led_pin, OUTPUT);    // Set led    - pin 13 as an output
32    pinMode(button_pin, INPUT_PULLUP);  //set Button pin as pullup input
33    //set Button as External Interrupt with change_mode function
34    attachInterrupt(digitalPinToInterrupt(button_pin), test,FALLING);
35    //FALLING for when the pin goes from high to low.
36    Serial.begin(9600);
37
38
39    // TIMER 1 for interrupt frequency 16 Hz:
40    cli(); // stop interrupts
41    TCCR1A = 0; // set entire TCCR1A register to 0
42    TCCR1B = 0; // same for TCCR1B
43    TCNT1  = 0; // initialize counter value to 0
44    // set compare match register for 16 Hz increments
45    OCR1A = 15624; // = 16000000 / (64 * 16) - 1 (must be <65536)
46    // turn on CTC mode
47    TCCR1B |= (1 << WGM12);
48    // Set CS12, CS11 and CS10 bits for 64 prescaler
49    TCCR1B |= (0 << CS12) | (1 << CS11) | (1 << CS10);
50    // enable timer compare interrupt
51    TIMSK1 |= (1 << OCIE1A);
52    sei(); // allow interrupts
```

Being done by:
ID : 2210
On : 16/3/2024
16:00 (Saturday)

HW01.16. PROTOCOL OF TEST-TEST 6-FINAL TEST-code both for simulation and reality – part 3

```
57  void test()
58  {
59    Serial.print("button has been pressed \n");
60    if (interuptcounter==0)//counter to check how many times buttton is pressed
61    {
62        interuptcounter+=1;
63        NORMAL_STATE  = false;
64        HIGH_STATE    = false;
65        LOW_STATE     = false;
66        digitalWrite(relay_pin, HIGH);    //turns ON light bulb
67        digitalWrite(led_pin, HIGH);    //turns ON led
68        TEST_MODE = true;
69      }
70    else
71      {
72        TEST_MODE = false;
73          interuptcounter=0;
74      }
75  }
```

Being done by:
ID : 2210
On : 16/3/2024
16:00 (Saturday)

HW01.16. PROTOCOL OF TEST-TEST 6-FINAL TEST-code both for simulation and reality – part 4

```
77  void loop()
78  {
79
80  }
81
82
83  ISR(TIMER1_COMPA_vect){
84    if(sample==16)
85    {
86      light_val = analogRead(photoresistor_pin);
87      sample=0;
88    }
89
90    else
91    {
92      sample++;
93    }
94      if(!TEST_MODE)
95      {
96        if((1030-light_val) < min_val)
97        {
98            NORMAL_STATE = false;
99            HIGH_STATE   = false;
100           LOW_STATE    = true;
101           TEST_MODE    = false;
102     }
103     else if((1030-light_val) > max_val)
104     {
105         NORMAL_STATE = false;
106         HIGH_STATE   = true;
107         LOW_STATE    = false;
108         TEST_MODE    = false;
109     }
110     else
111     {
112         NORMAL_STATE = true;
113         HIGH_STATE   = false;
114         LOW_STATE    = false;
115         TEST_MODE    = false;
116     }
```

Being done by:
ID : 2210
On : 16/3/2024
16:00 (Saturday)

HW01.16. PROTOCOL OF TEST-TEST 6-FINAL TEST-code both for simulation and reality – part 5

```
120     if(LOW_STATE)
121     {
122       Serial.print("dark:\n");
123       Serial.println(1030-light_val);
124       digitalWrite(relay_pin, HIGH); //turns high light bulb
125       digitalWrite(led_pin, HIGH);    //turns high led
126
127       if(timer_cnt == 8)
128       {
129         tone(buzzer_pin, 1000); // Send 1 KHz sound signal
130         timer_cnt=0;
131       }
132       else
133       {
134           noTone(buzzer_pin);     // Stop sound
135           timer_cnt++;
136        }
137
138     }
```

```
140       else if(HIGH_STATE)
141       {
142         Serial.print("super high light:\n");
143         Serial.println(1030-light_val);
144         noTone(buzzer_pin);
145         digitalWrite(relay_pin, HIGH); //turns ON light bulb
146         digitalWrite(led_pin, LOW);    //turns off led
147
148       }
149
150       else if(NORMAL_STATE)
151       {
152         Serial.print("normal light:\n");
153         Serial.println(1030-light_val);
154         noTone(buzzer_pin);
155         digitalWrite(relay_pin, LOW); //turns OFF light bulb
156         digitalWrite(led_pin, HIGH);  //turns ON led
157       }
158     }
160     else if(timer_cnt_rst==4)
161       {
162         tone(buzzer_pin, 1500); // Send 1.5KHz sound signal
163         timer_cnt_rst=0;
164       }
165       else
166       {
167         noTone(buzzer_pin);     // Stop sound
168         timer_cnt_rst++;
169       }
170
171
172   }
```

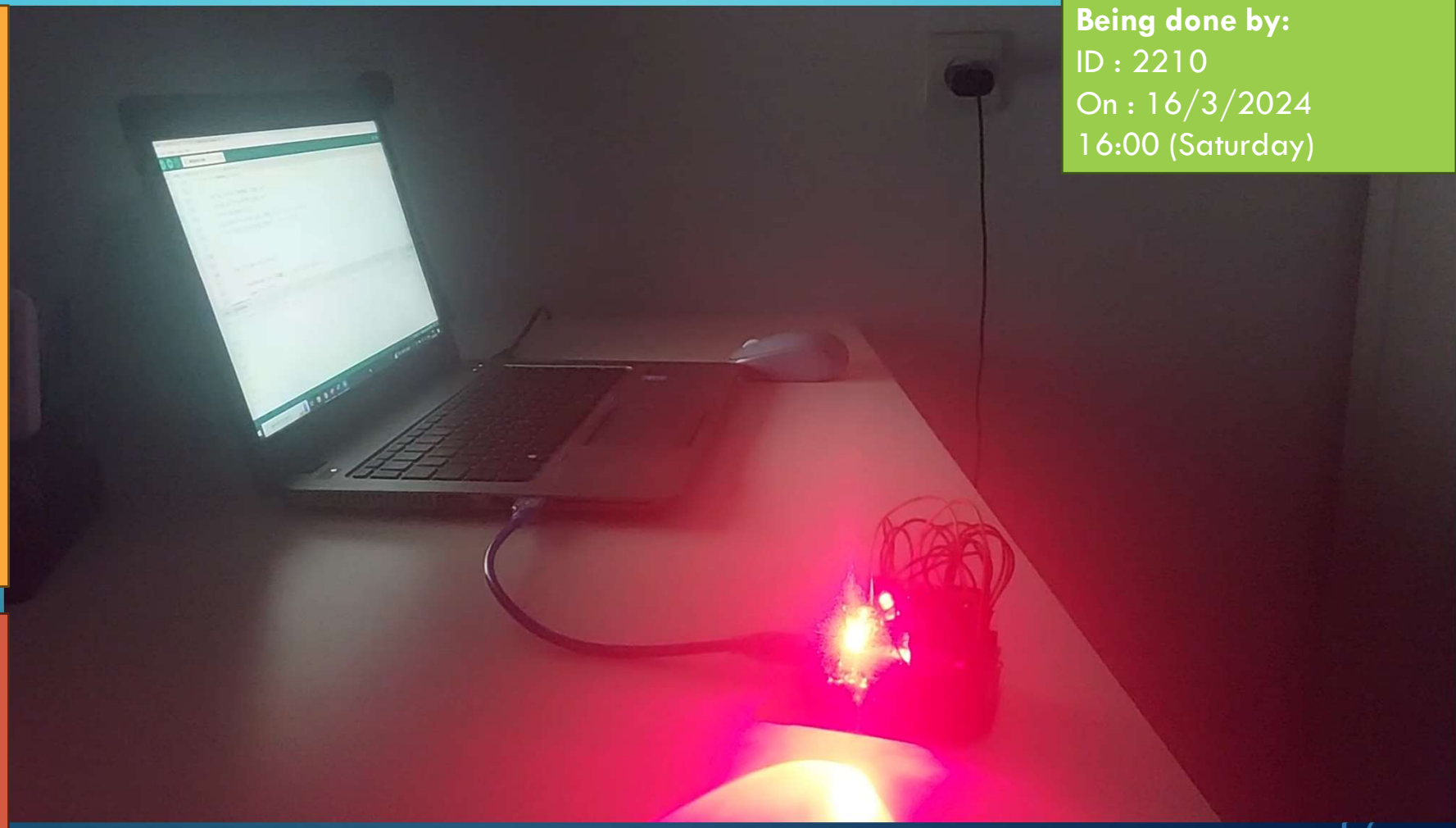
Being done by:
ID : 2210
On : 16/3/2024
16:00 (Saturday)

HW01.16. PROTOCOL OF TEST-TEST 6-FINAL TEST-HARDWARE-PART 1

The main test is presented in the video – please turn up the volume of the computer.

In the test we start at low light state (dark) , turn up a regular light to get into normal light state and then turn up a strong flashlight to examine super high light state

The test is successful and the data received in the monitor is as expected and like in the preparatory (1…5) protocols of test

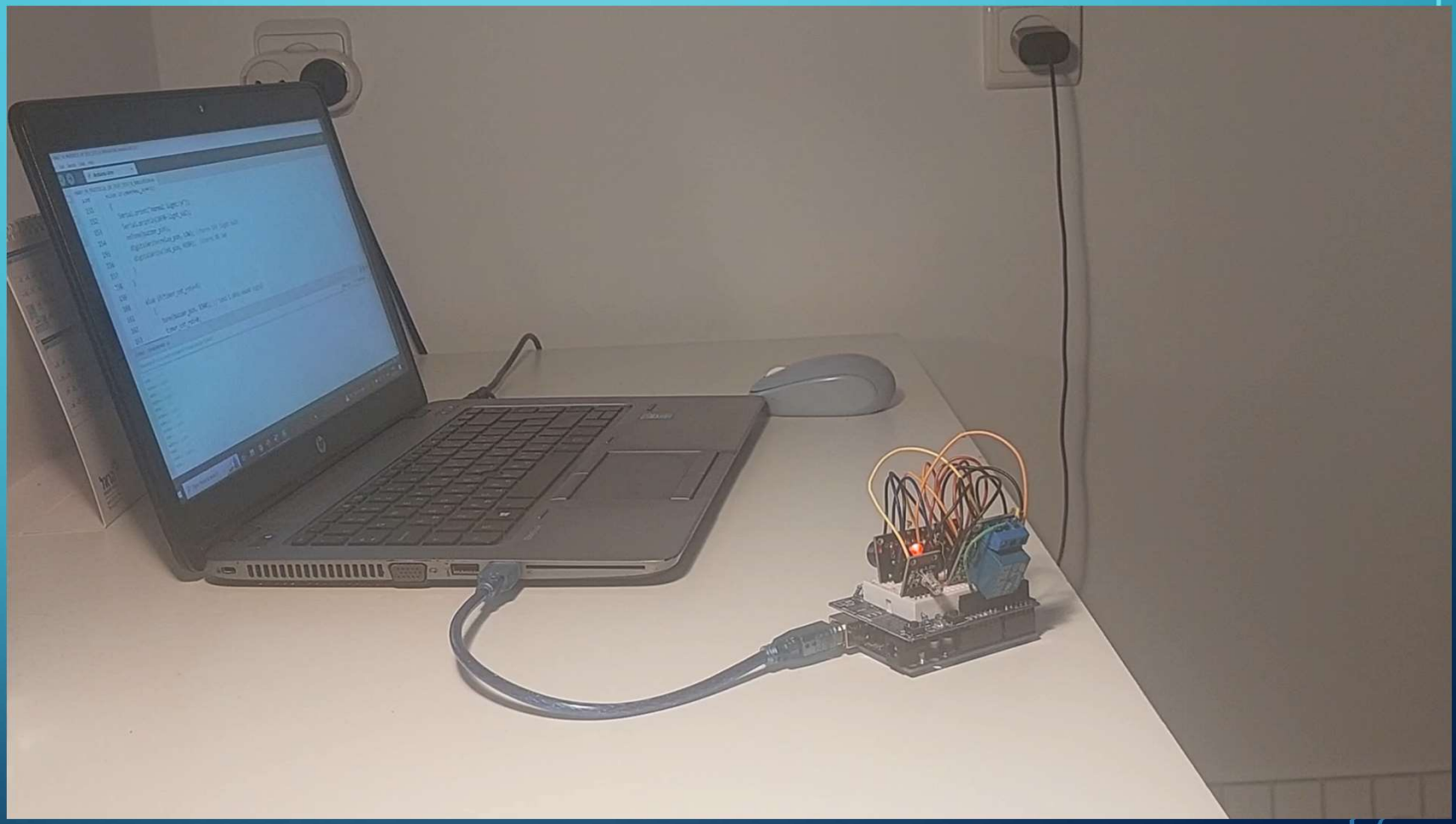


Being done by:
ID : 2210
On : 16/3/2024
16:00 (Saturday)

HW01.16. PROTOCOL OF TEST-TEST 6-FINAL TEST-HARDWARE-PART 2

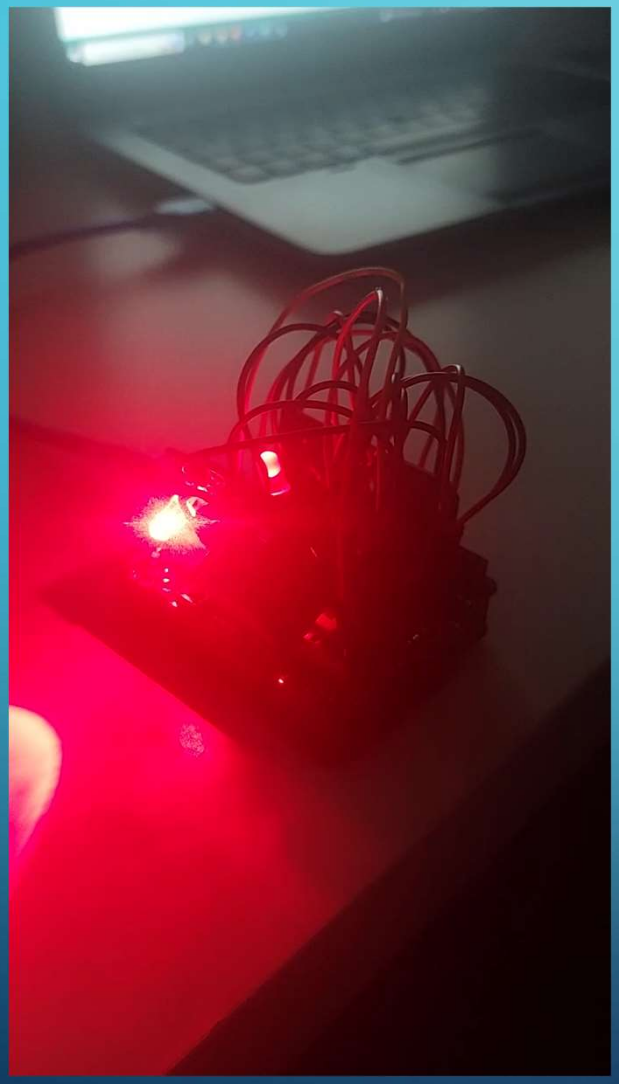
It can be seen in the laptop screen in the back that when I press the button – the measuring of the lights stops and the program prints that the button has been pressed.

This test is to check the entrance to test mode from normal light mode



Being done by:
ID : 2210
On : 16/3/2024
17:00 (Saturday)

HW01.16. PROTOCOL OF TEST-TEST 6-FINAL TEST-HARDWARE-PART 3



LOW LIGHT (DARK) STATE

Output Serial Monitor

Message (Enter to send m

```
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
dark:64
d
```

Being done by:
ID : 2210
On : 16/3/2024
17:15 (Saturday)

First part-devices: sources of information

Bibliography of that presentation

https://www.thegeekpub.com/wiki/sensor-wiki-ky-/module-relay-019

https://www.learningaboutelectronics.com/Articles/What-is-a-single-pole-double-throw-switch-SPDT

https://sensorkit.joy-it.net/en/sensors/ky-019

https://sensorkit.joy-it.net/en/sensors/ky-018

https://www.devobox.com/en/photosensors/html.018-ky-module-photoresistor-479

https://sensorkit.joy-it.net/en/sensors/ky-016

https://sensorkit.joy-it.net/en/sensors/ky-006

https://www.thegeekpub.com/wiki/sensor-wiki-ky-/module-buzzer-piezo-passive-006

https://sensorkit.joy-it.net/en/sensors/ky-004

https://www.faranux.com/product/ky-/module-button-push-004

https://www.youtube.com/watch?app=desktop&v=pF_9gBgx4pg&ab_channel=ITcomAcademy

https://youtu.be/1lLomjlwb8w?si=KqQZsDbuf-s3jQz5

https://youtu.be/ChLvsoYRZq8si=pahLsvp?0tWqFV7F8