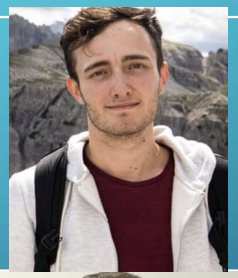
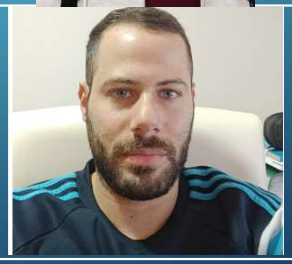
ELECTRONIC INSTRUMENTATION

LECTURER DR. SAMUEL KOSOLAPOV

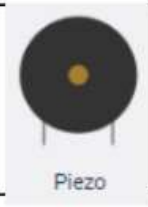
HOMEWORK 01 - ALARM SYSTEM HARDWARE & TINKERCAD SIMULATION

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

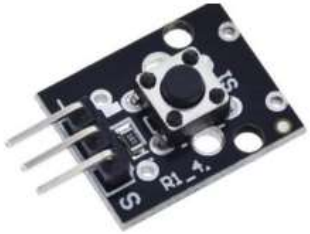
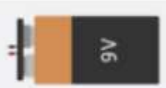
Link to TinkerCad:

https://www.tinkercad.com/things/hZZeO89awSf-HW1/editel?sharecode=_l0NxVFJaEX1Usf8wy-QkD1jkYUm4oNNnHBh2bHN1rs

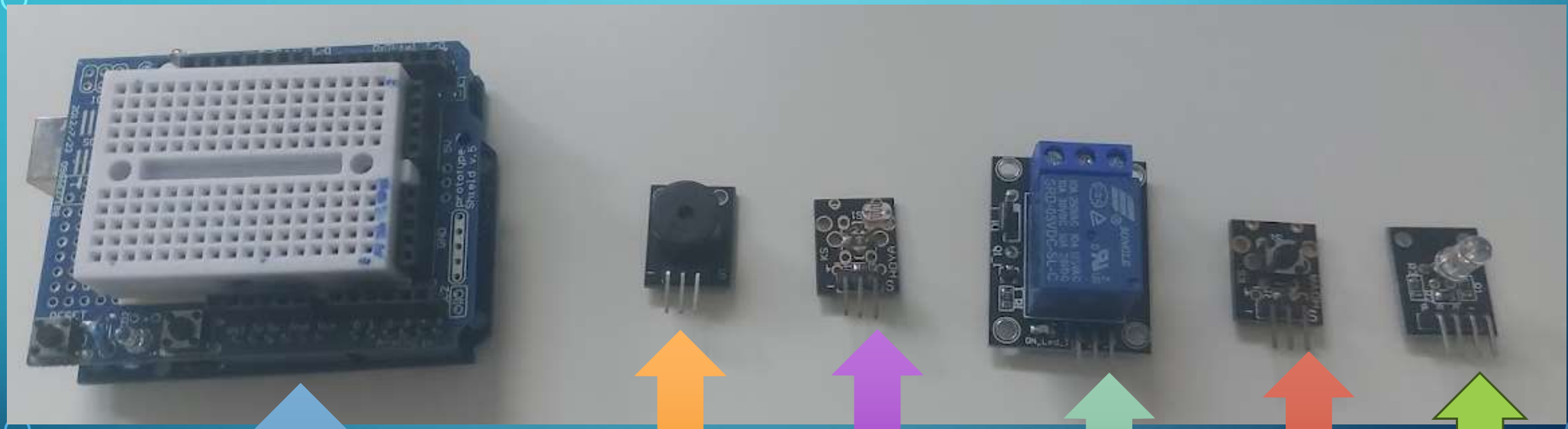
OUR PROJECT CONTAIN THE FOLLOWING COMPONENTS:

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

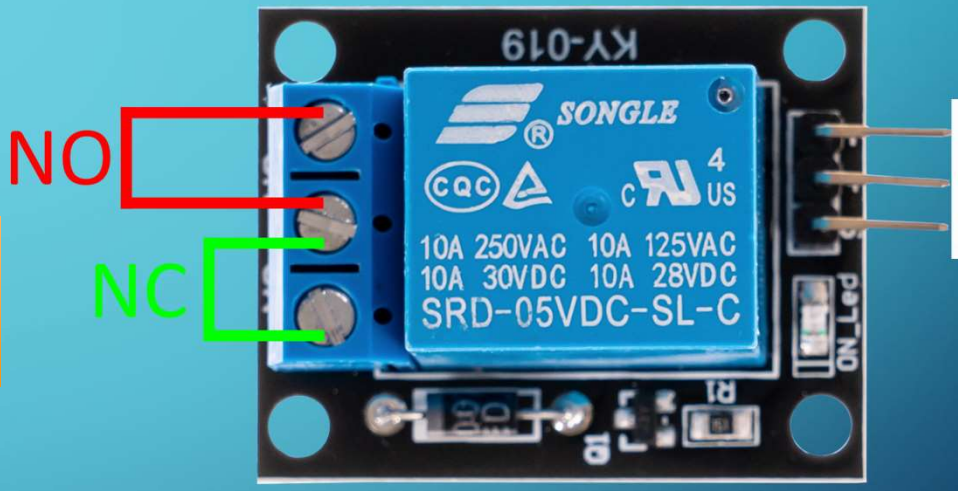
The relay module - KY-019

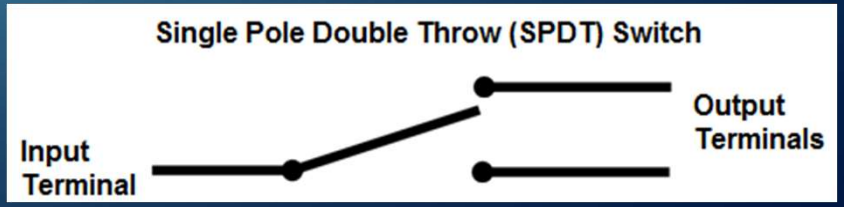
The KY-019 relay module has two contacts, normally open (NO) and normally closed (NC) that are operated by pulling the signal pin **HIGH**.

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.

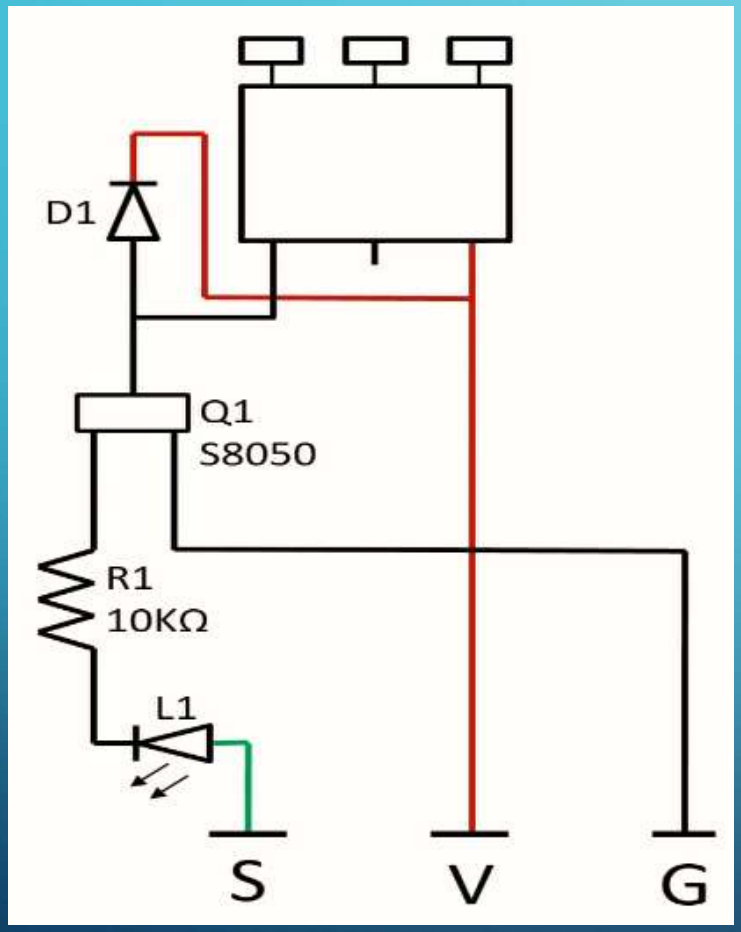
Normally closed will pass current until the signal line is activated. Normally Open will not pass current until the signal line is activated. The common terminal is shared between the two contacts.

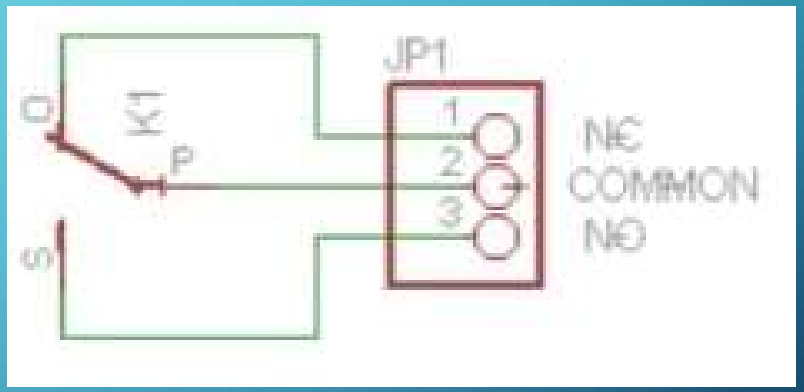
This relay is called a **SPDT** relay – it means "single pole double throw" – it means that the switch has only a single input and can connect to and switch between 2 outputs.





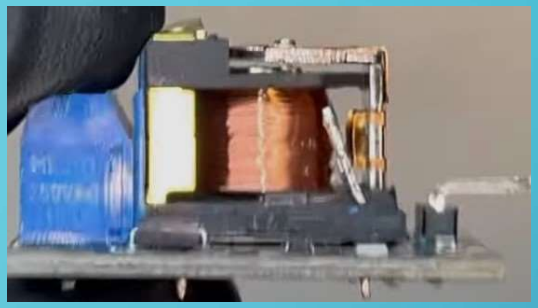
The relay module - KY-019 - internal structure



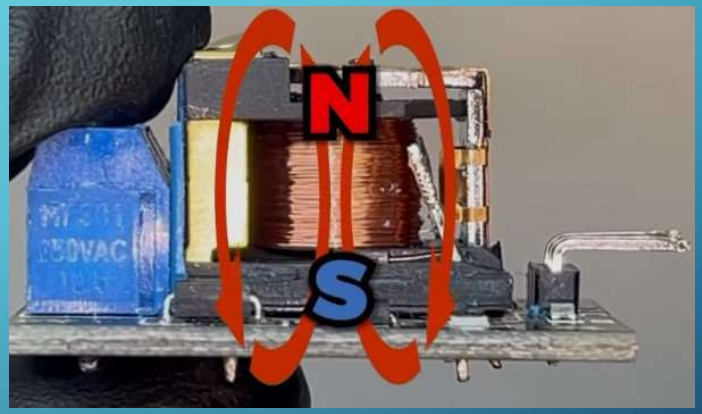


The relay module - KY-019– physical operation explanation

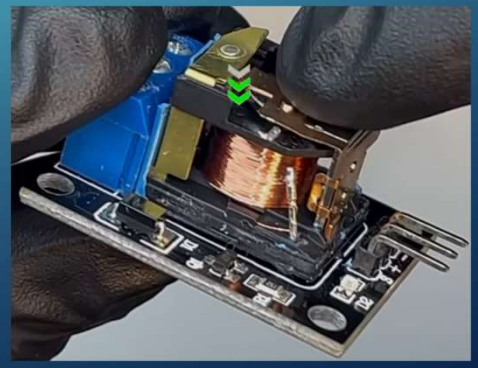
Inside the relay there's a coil



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



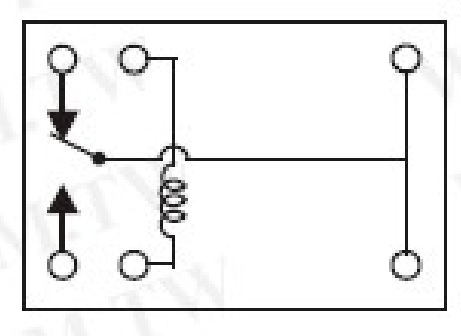
The **magnetic field toggles a pole which can connect or disconnect two circuits**



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.



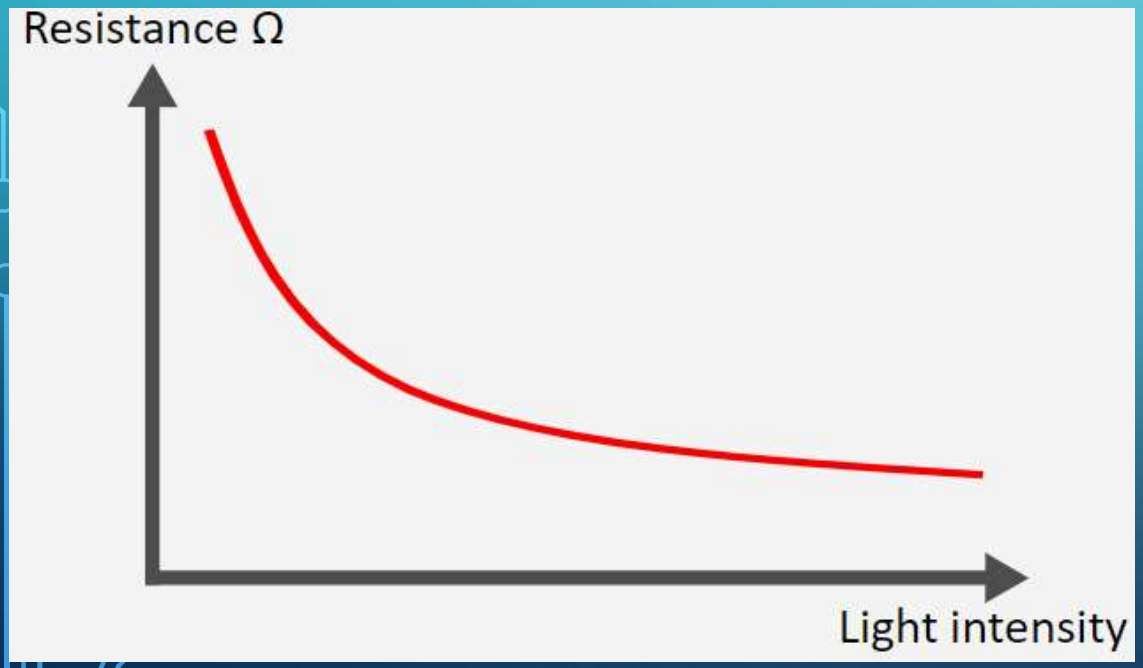
Supports various coil voltages, ranging from 5V to 24V DC, allowing compatibility with diverse power sources. In addition-Can handle currents up to 7A

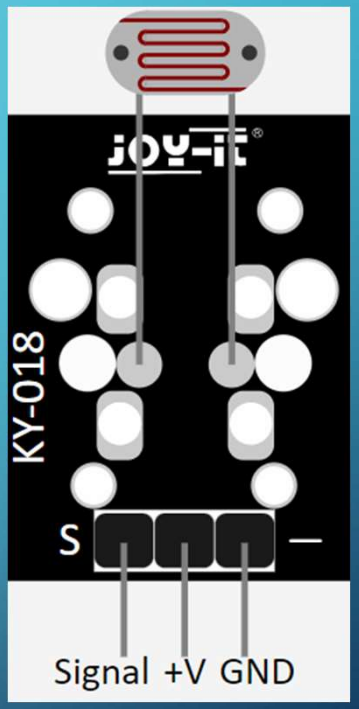
Features optocoupler isolation between the control input and the relay output, enhancing safety and preventing noise interference.

The photoresistor module - KY-018

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

LDR= light-dependent resistor

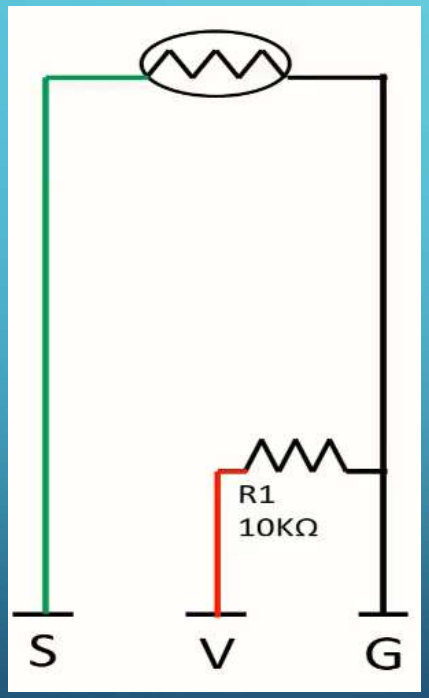


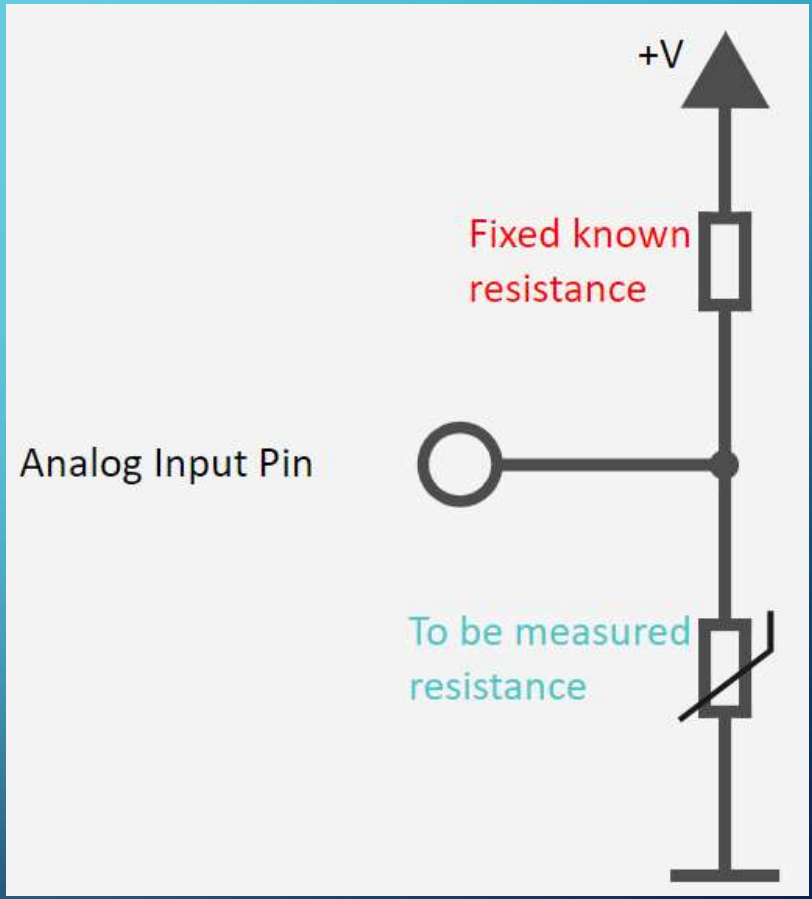


The **VCC** pin is used to provide power to the module.
The **GND** pin is connected to the ground of the Arduino.
The **signal** pin is the output pin that provides an analog voltage signal to the Arduino proportional to the intensity of light falling on the photoresistor.

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

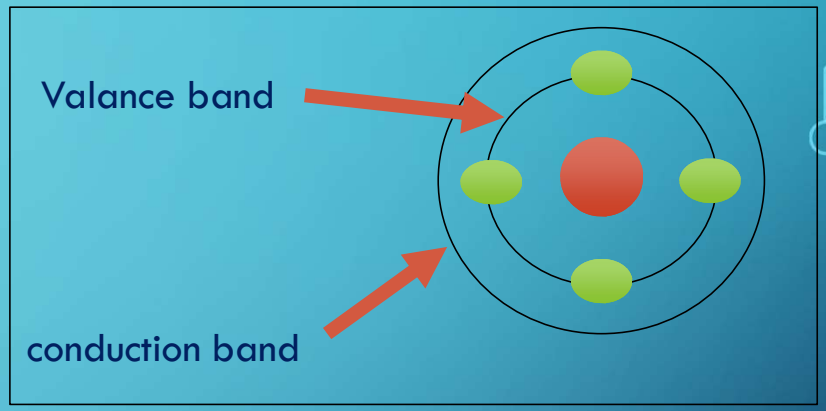
The semiconductor material on the resistor works on the principle of Photoconductivity:

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.

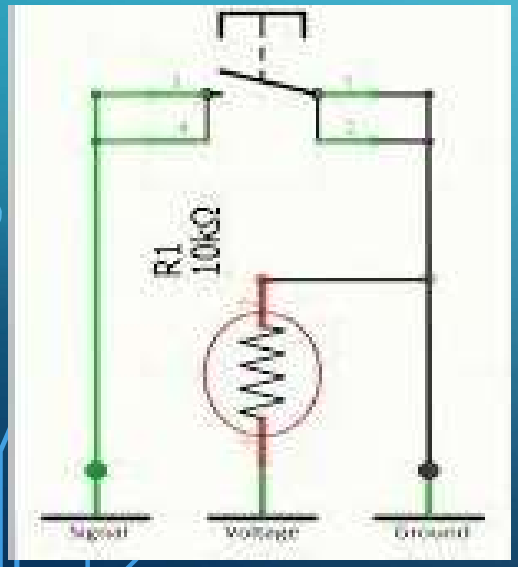
When there are no more photons-these electrons will fall back to the valance band – and current will stop.

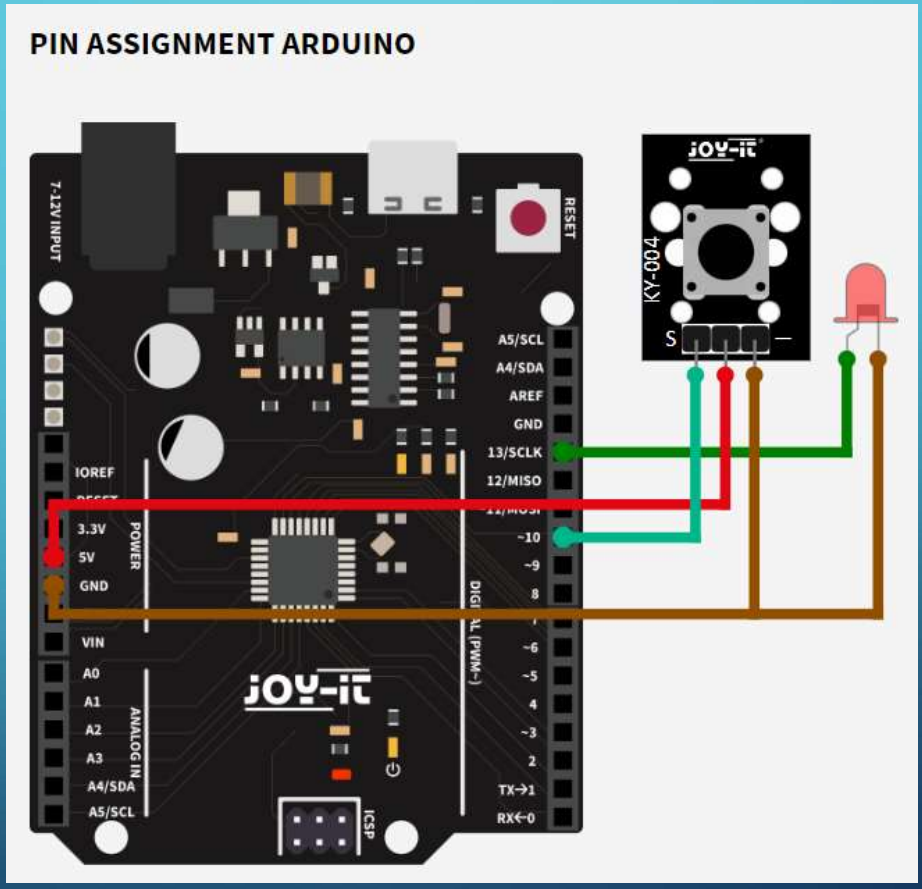


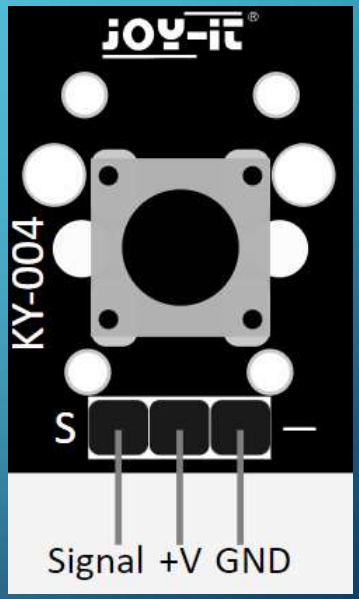
The button module - KY-004

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



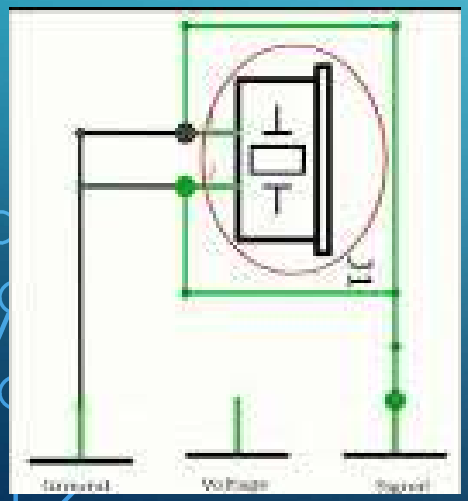


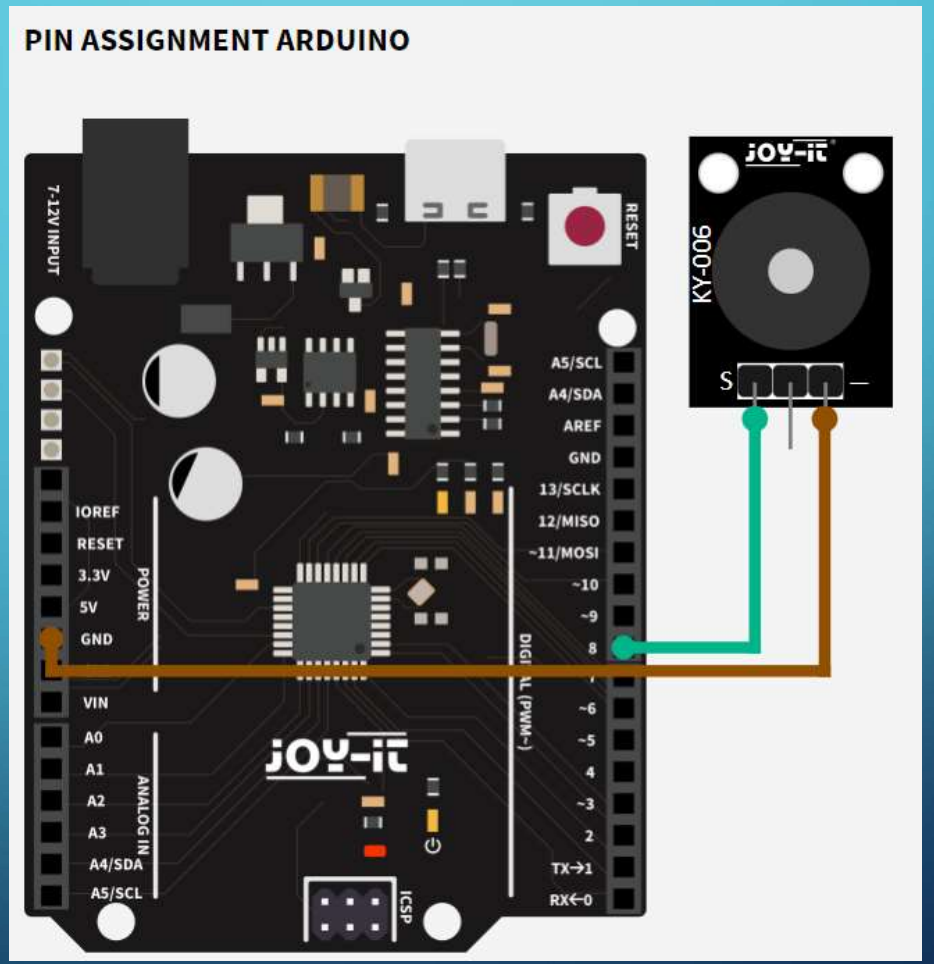


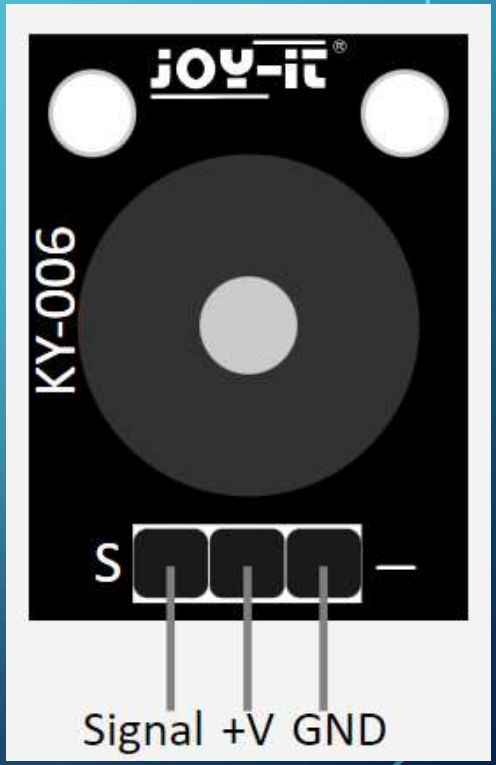
The buzzer module - KY-006

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





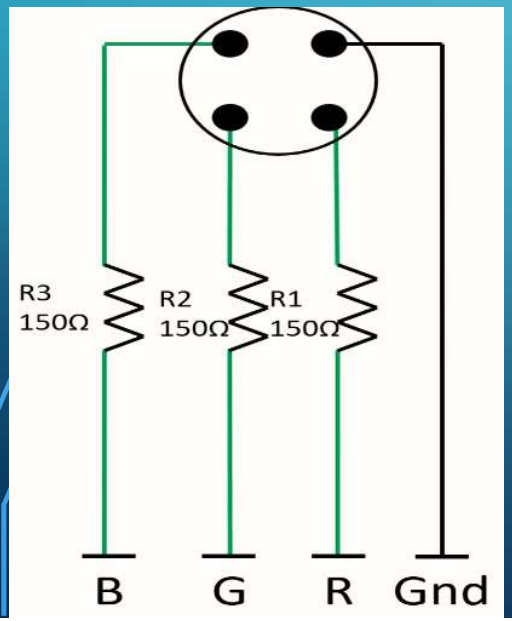


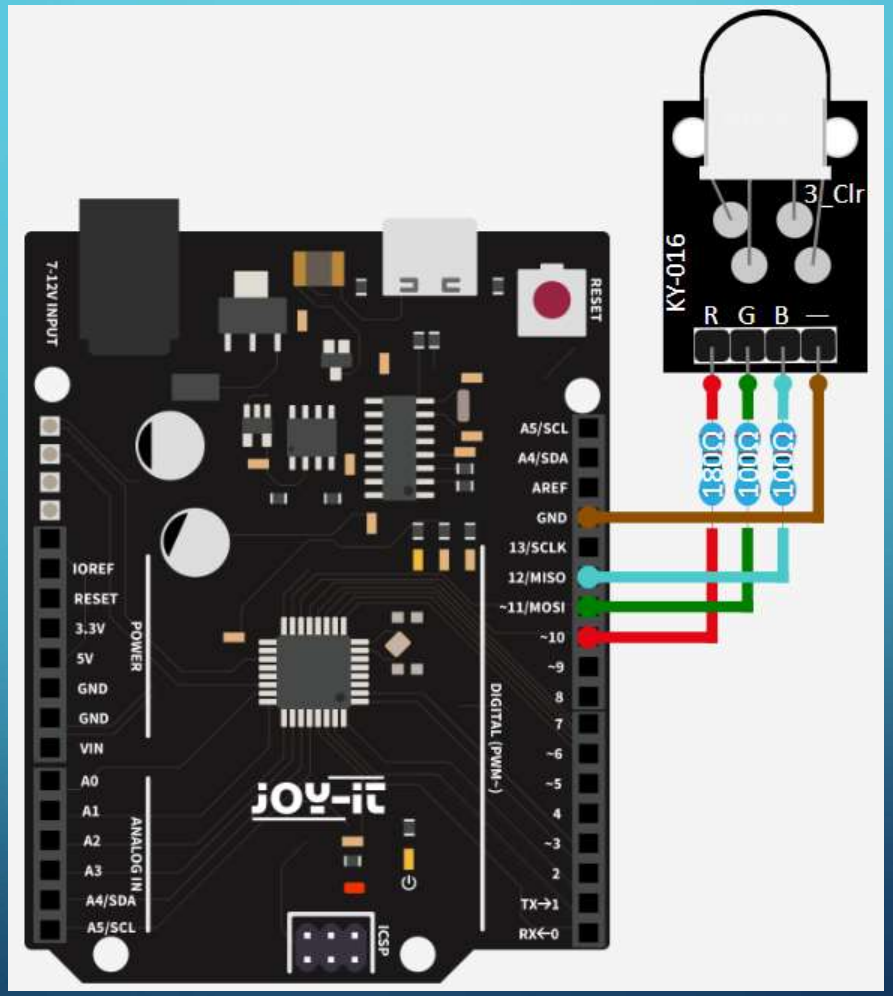
+V pin is not in use

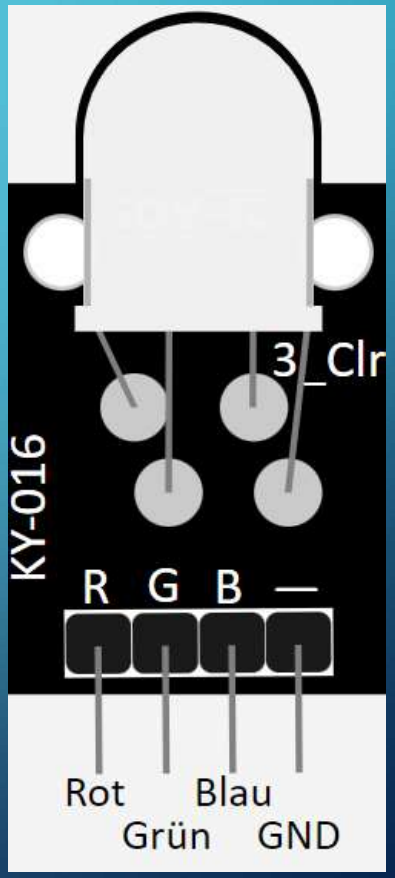
The RGB LED module - KY-016

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



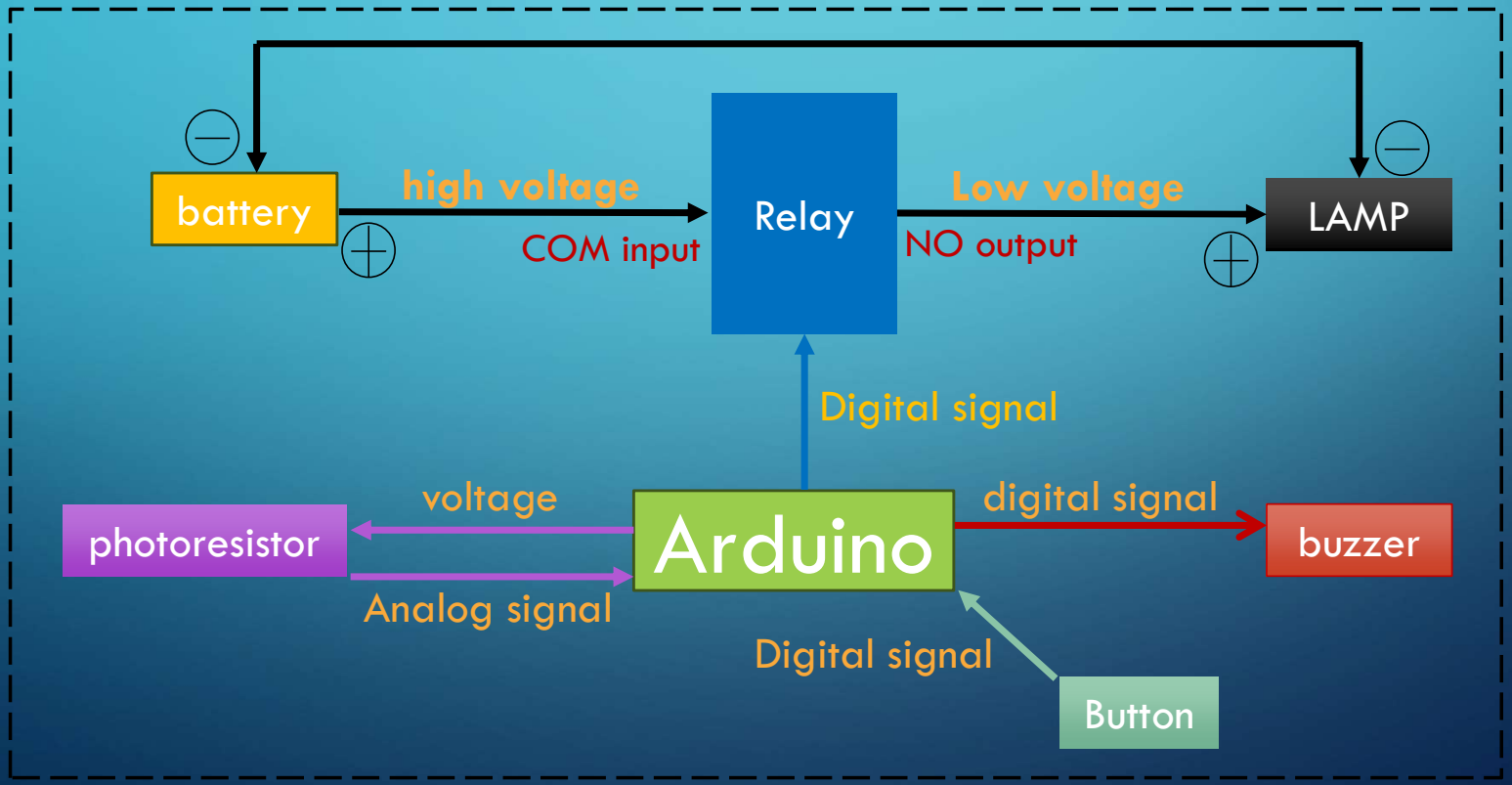




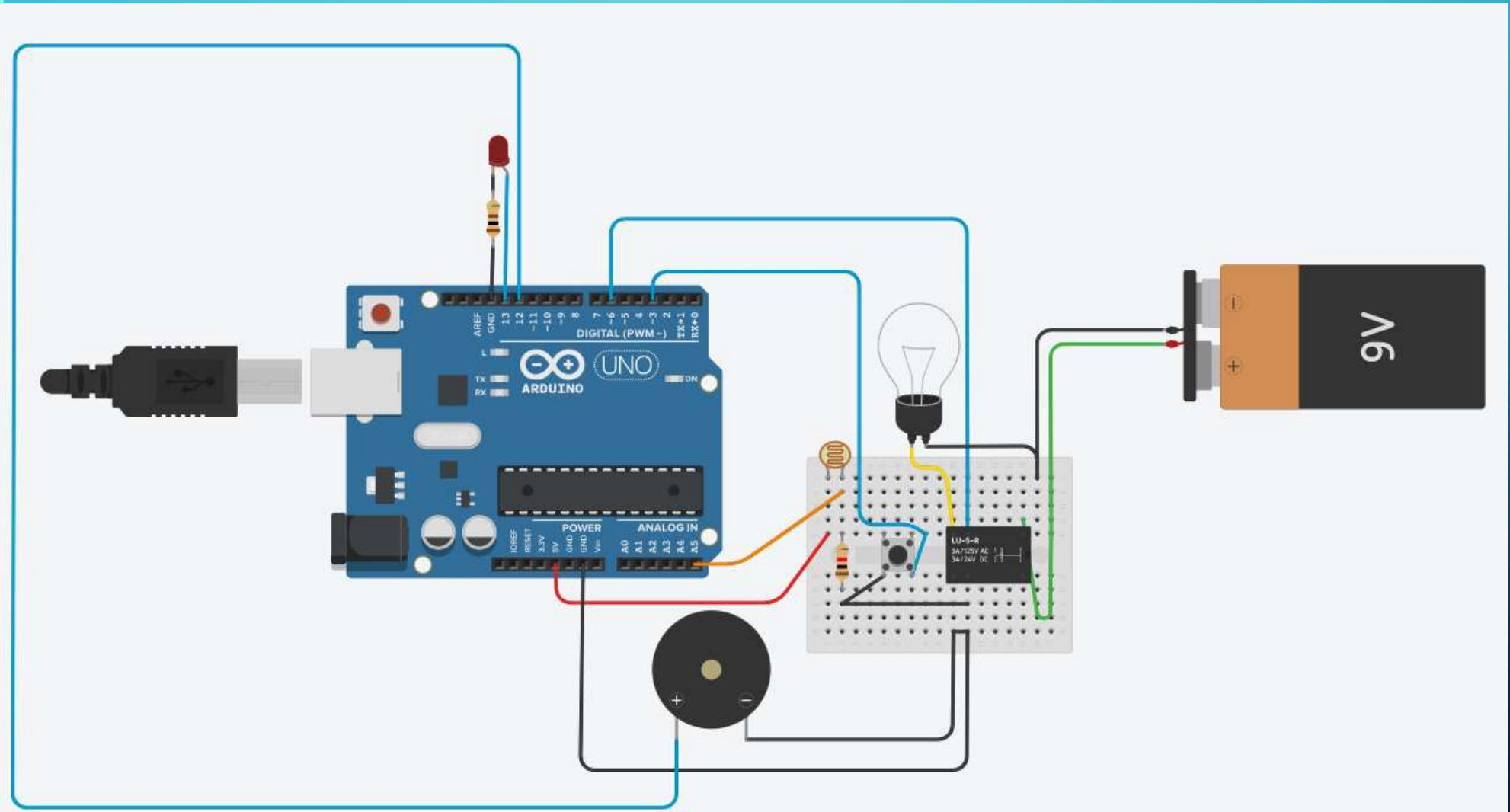
basic flow chart

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

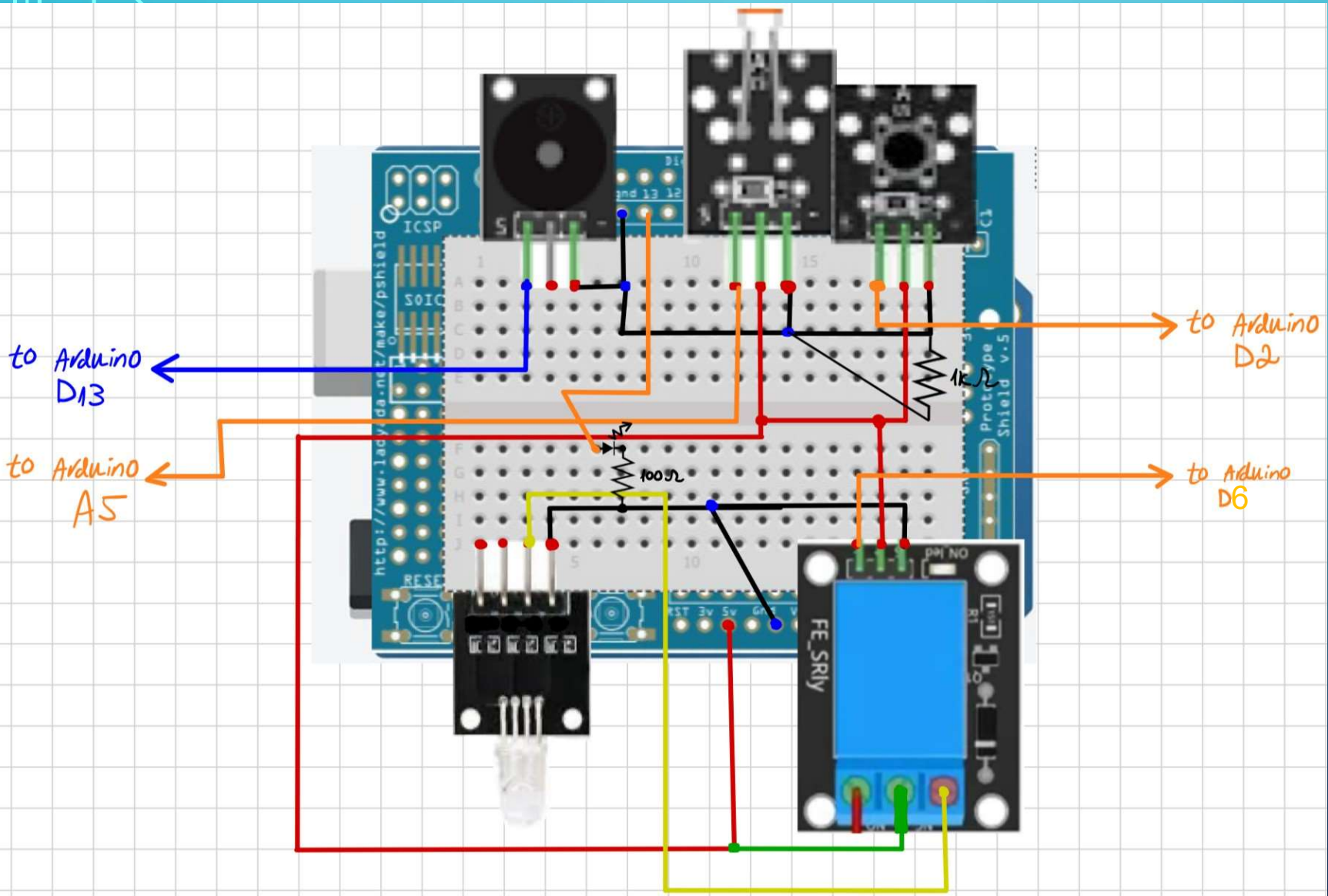
The basic flow chart will help us to control the assembling process and writing code later



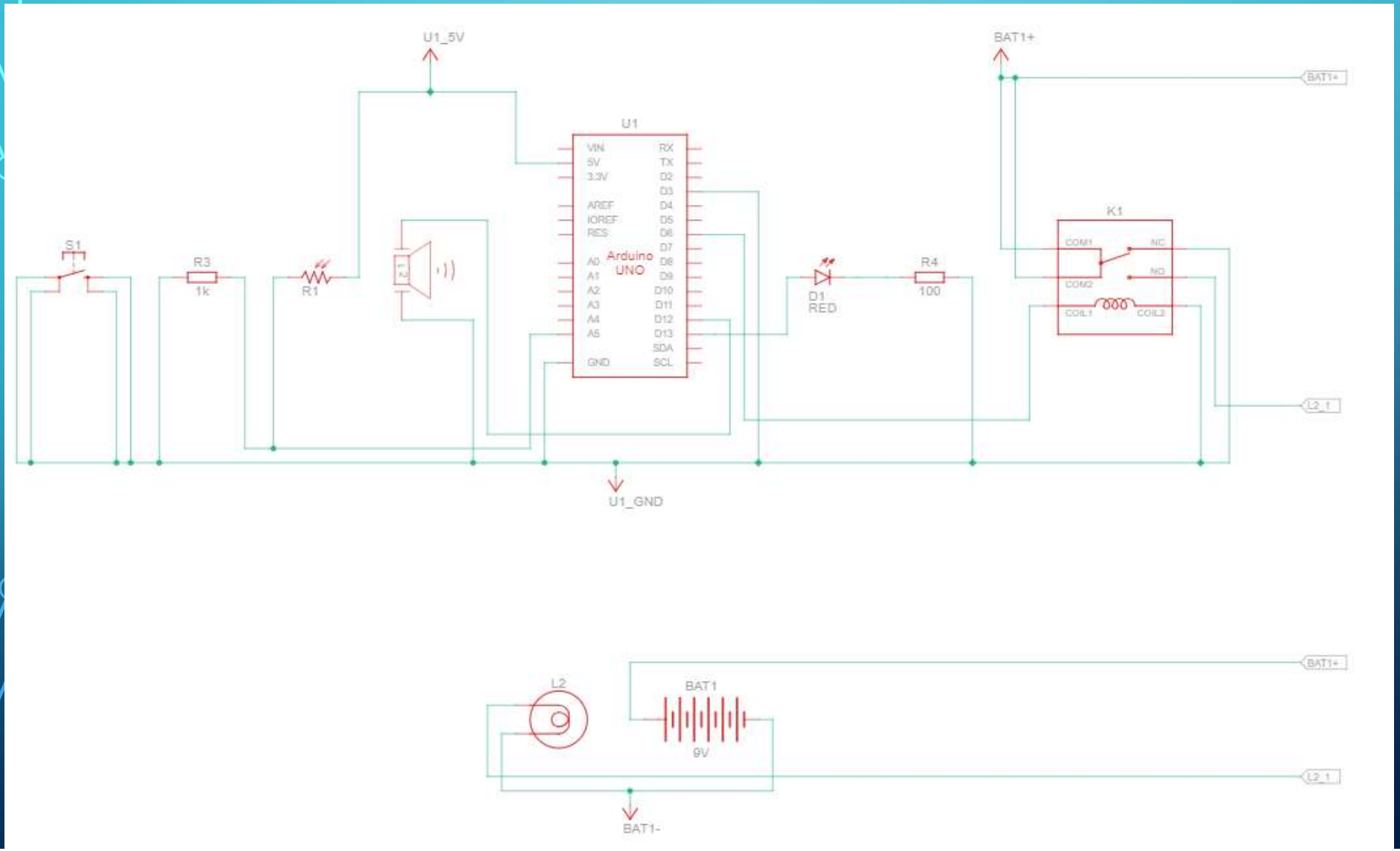
TinkerCad simulation of the circuit



Layout of the real circuit

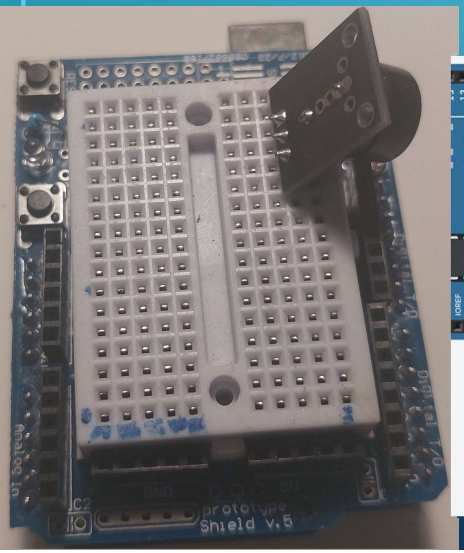


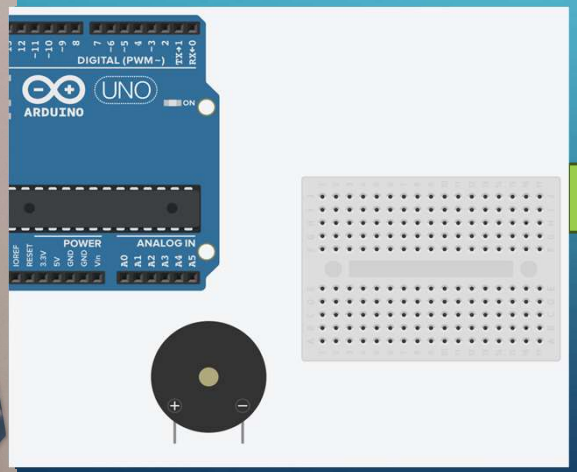
Real layout of the circuit



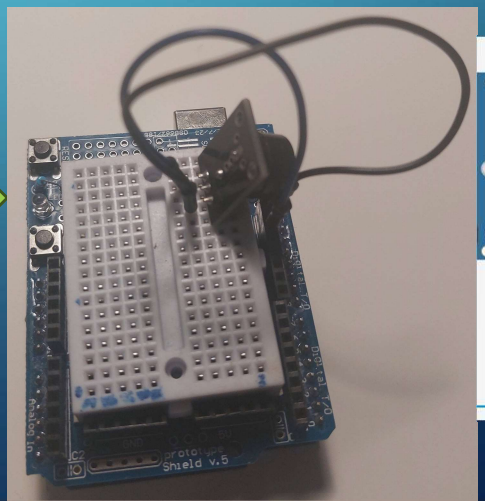
PLACEMENT OF buzzer module - KY-006 ON BREADBOAD

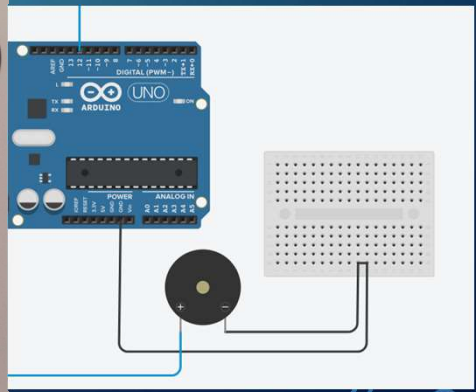
#	Component pin name	Breadboard hole name	comments
1	+V	A4	NOT CONNECTED!
2	SIGNAL	A3	Connected to digital pin 13 of Arduino
3	GND	A5	Connected to GND of Arduino (on the shield)





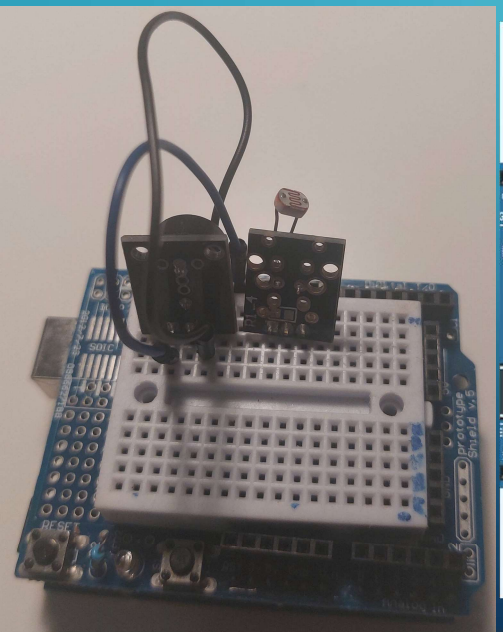
After wiring

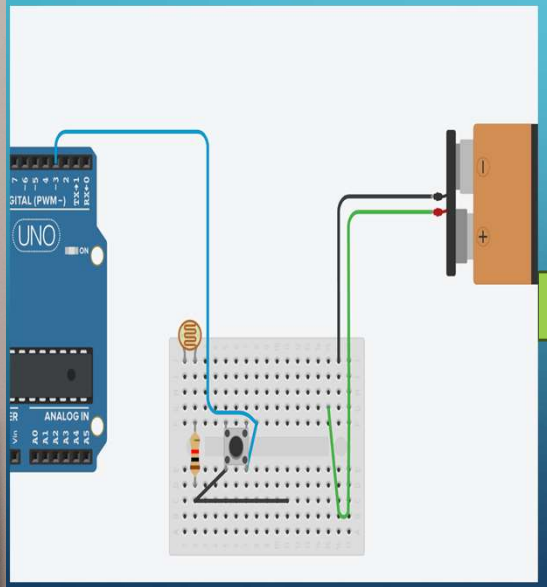




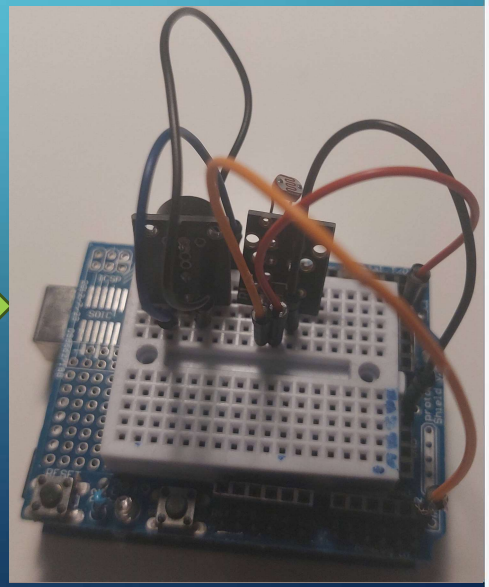
PLACEMENT OF photoresistor module - KY-018 ON BREADBOAD

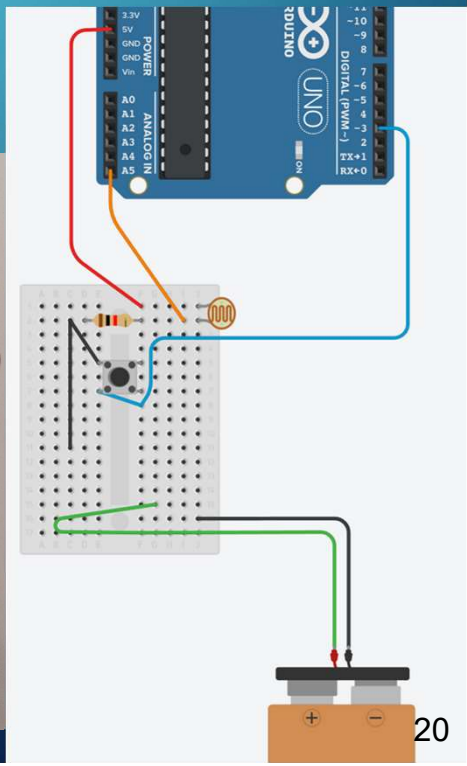
#	Component pin name	Breadboard hole name	Comments
1	+V	A13	Connected to 5V of Arduino (on the shield)
2	SIGNAL	A12	Connected to analog pin A5
3	GND	A14	Connected to GND of Arduino (on the shield)





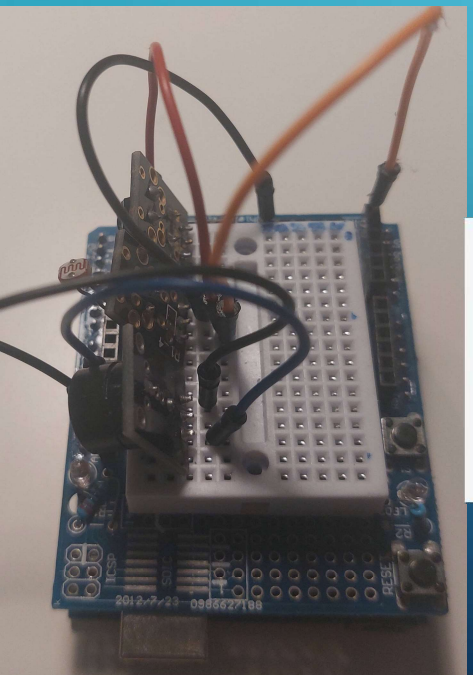
After wiring

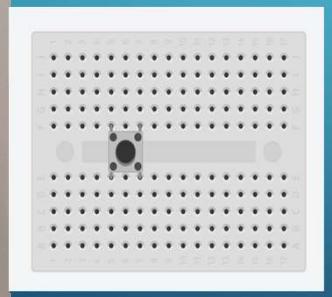




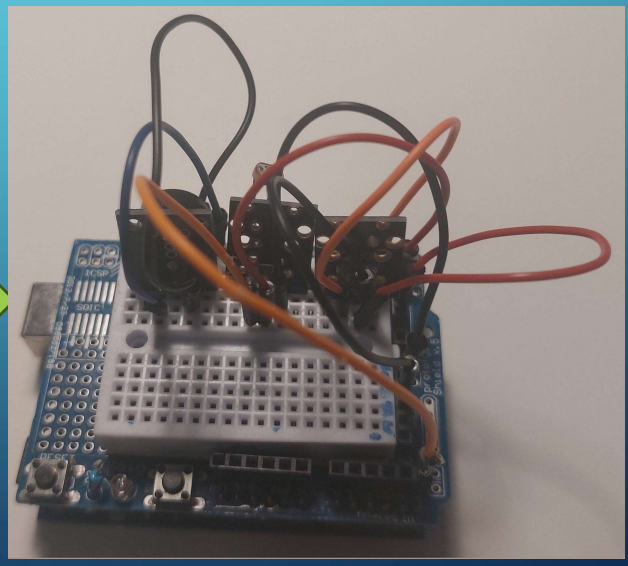
PLACEMENT OF button module - KY-004 ON BREADBOAD

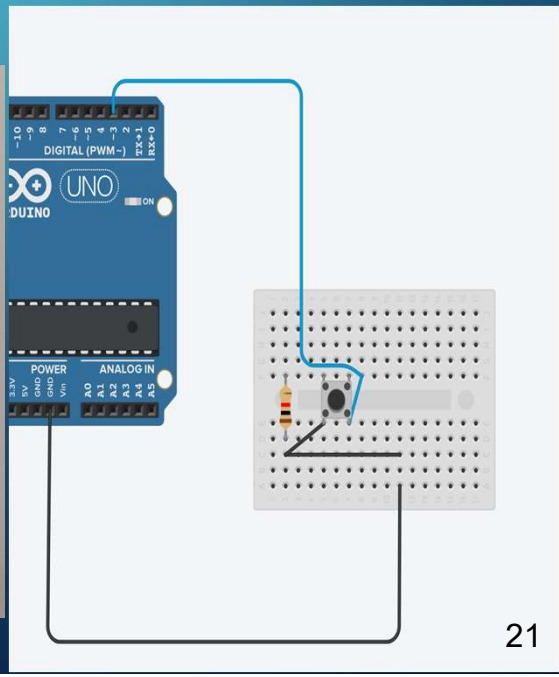
#	Component pin name	Breadboard hole name	comments
1	+V	A19	Connected to 5V of Arduino (on the shield)
2	SIGNAL	A18	Connected to digital pin 3 of Arduino
3	GND	A20	Connected to GND of Arduino (on the shield) In addition – a 1k[ohm] resistor is connected from E20 to GND





After wiring



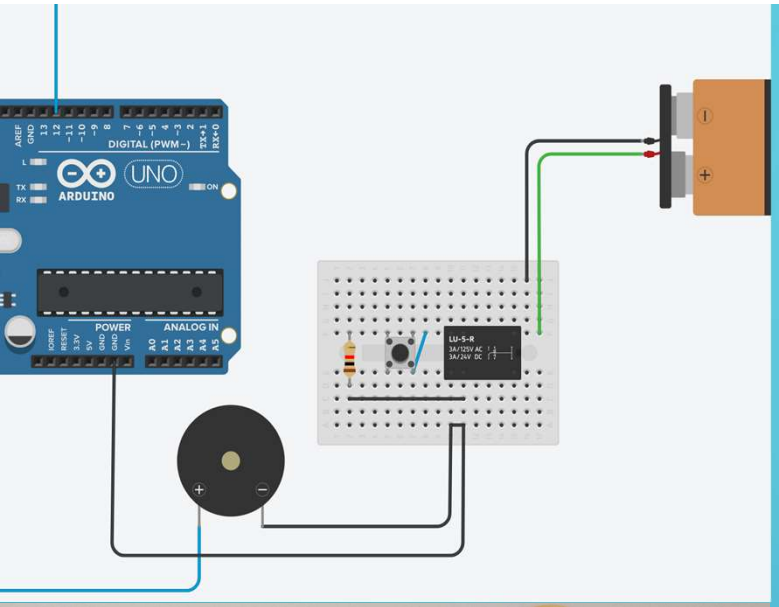


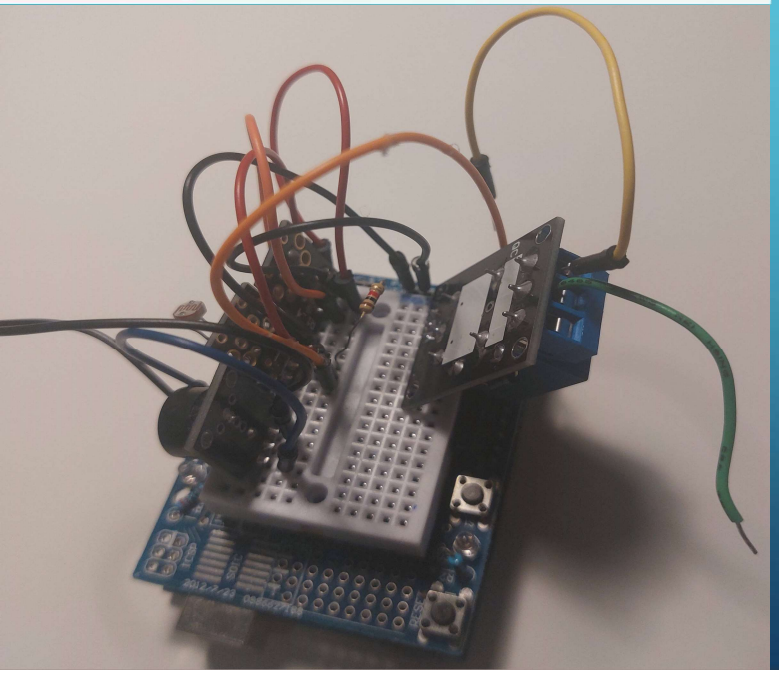
PLACEMENT OF relay module - KY-019 ON BREADBOAD

#	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

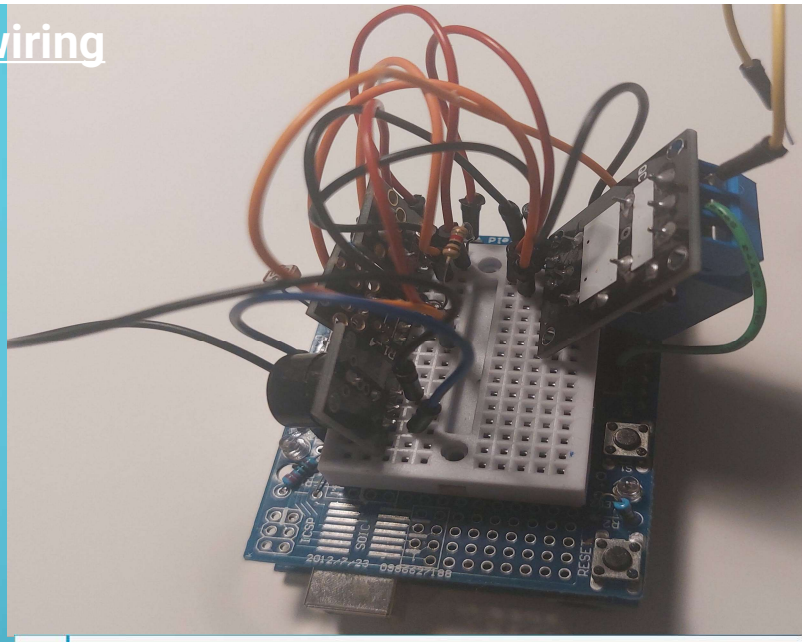
Photos of relay module before and after wiring in the next slide!

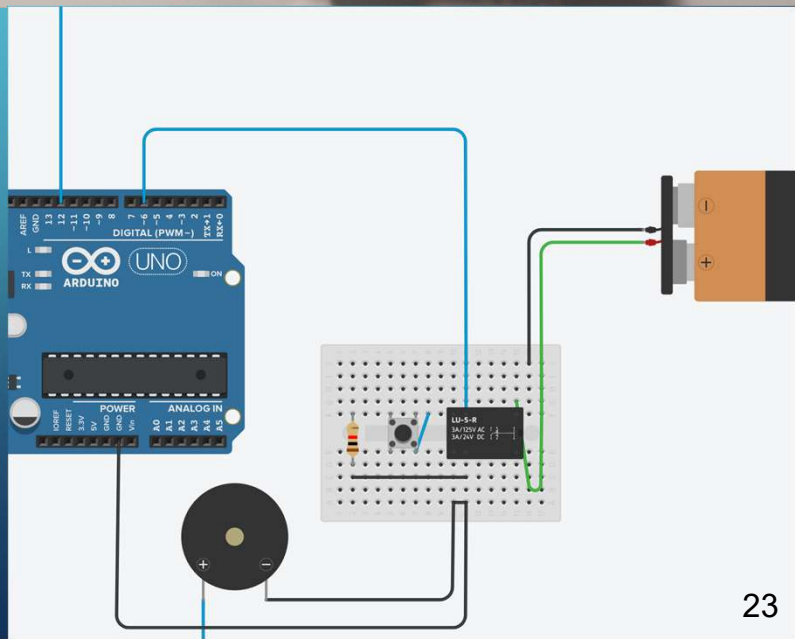
relay module before and after wiring





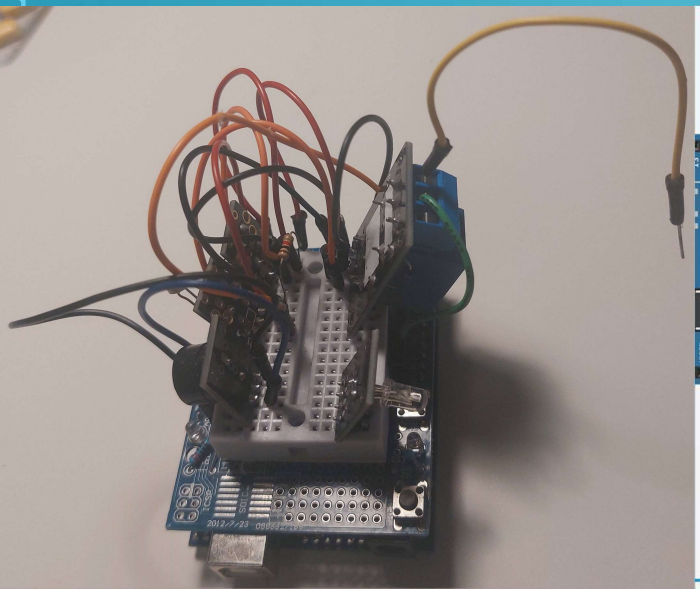
After wiring

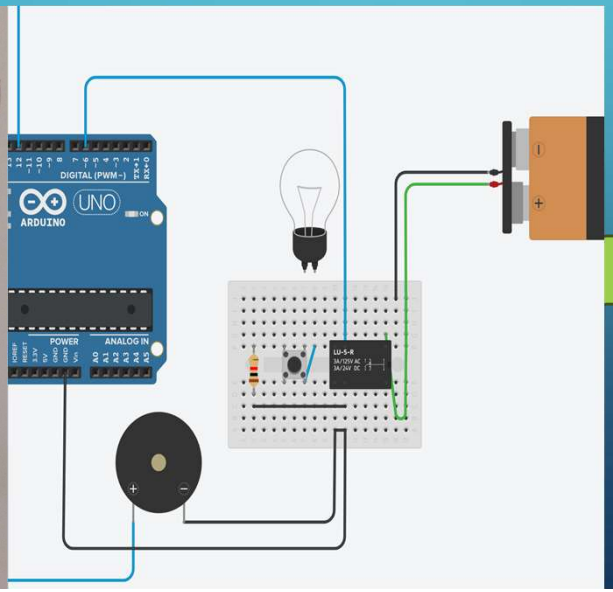




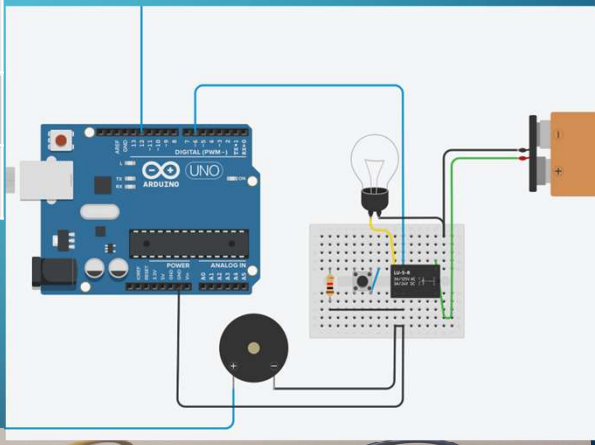
PLACEMENT OF LED module - KY-016 ON BREADBOAD

#	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



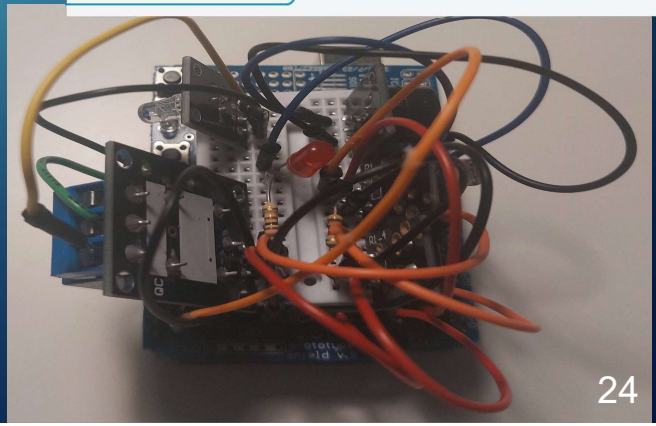


Table 3-wire' connections on the shield' breadboard

#	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 12	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	G4	F20	Bulb module
16	black	GND	G3	GND of the Arduino	Bulb module
17	blue	anode of LED	G7	Arduino digital pin 13	LED 13

The code we used-both in the simulation and in reality

Part 1 of the code-declerations of constants, flags and pin setup

```
/* 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;

/* photoresistor values */
int light_val = 100;
const int max_val = 600;
const int min_val = 400;

/* pins setup */
const int photoresistor_pin = A5;    //photoresistor to arduino
const int button_pin = 3;            //button_pin
const int buzzer_pin = 12;           //buzzer to arduino pin
const int relay_pin  = 6;
const int led_pin    = 13;
```

Part 2 of the code- setup function and timer declaration (16 kHZ)

```
25  void setup()
26  {
27    pinMode(button_pin, INPUT);  // Set BUTTON - pin 3 as an input
28    pinMode(buzzer_pin, OUTPUT); // Set buzzer - pin 11 as an output
29    pinMode(relay_pin, OUTPUT);  // Set relay  - pin 12 as an output
30    pinMode(led_pin, OUTPUT);    // Set led    - pin 13 as an output
31    pinMode(button_pin, INPUT_PULLUP);  //set Button pin as pullup input
32    //set Button as External Interrupt with change_mode function
33    attachInterrupt(digitalPinToInterrupt(button_pin), test,FALLING);
34    //FALLING for when the pin goes from high to low.
35    //Serial.begin(9600);
36
37
38    // TIMER 1 for interrupt frequency 16 Hz:
39    cli(); // stop interrupts
40    TCCR1A = 0; // set entire TCCR1A register to 0
41    TCCR1B = 0; // same for TCCR1B
42    TCNT1  = 0; // initialize counter value to 0
43    // set compare match register for 16 Hz increments
44    OCR1A = 15624; // = 16000000 / (64 * 16) - 1 (must be <65536)
45    // turn on CTC mode
46    TCCR1B |= (1 << WGM12);
47    // Set CS12, CS11 and CS10 bits for 64 prescaler
48    TCCR1B |= (0 << CS12) | (1 << CS11) | (1 << CS10);
49    // enable timer compare interrupt
50    TIMSK1 |= (1 << OCIE1A);
51    sei(); // allow interrupts
52
53
54  }
```

Part 3 of the code- define test function

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

```
75  void loop()
76  {
77
78  }
79
80
81  ISR(TIMER1_COMPA_vect){
82
83      light_val = analogRead(photoresistor_pin);
84      if(!TEST_MODE)
85      {
86        if(light_val < min_val)
87        {
88            NORMAL_STATE = false;
89            HIGH_STATE   = false;
90            LOW_STATE    = true;
91            TEST_MODE    = false;
92        }
93        else if(light_val > max_val)
94        {
95            NORMAL_STATE = false;
96            HIGH_STATE   = true;
97            LOW_STATE    = false;
98            TEST_MODE    = false;
99        }
```

Part 4 of the code- ISR (interrupt system) – flags status according to real status

Part 5 of the code- continuation of the previews slide (ISR)

```
100       else
101       {
102           NORMAL_STATE = true;
103           HIGH_STATE   = false;
104           LOW_STATE    = false;
105           TEST_MODE    = false;
106       }
107
108     if(LOW_STATE)
109     {
110       digitalWrite(relay_pin, HIGH); //turns high light bulb
111       digitalWrite(led_pin, HIGH);    //turns high led
112
113       if(timer_cnt == 8)
114       {
115         tone(buzzer_pin, 1000); // Send 1 KHz sound signal
116         timer_cnt=0;
117       }
118       else
119       {
120           noTone(buzzer_pin);     // Stop sound
121           timer_cnt++;
122        }
```

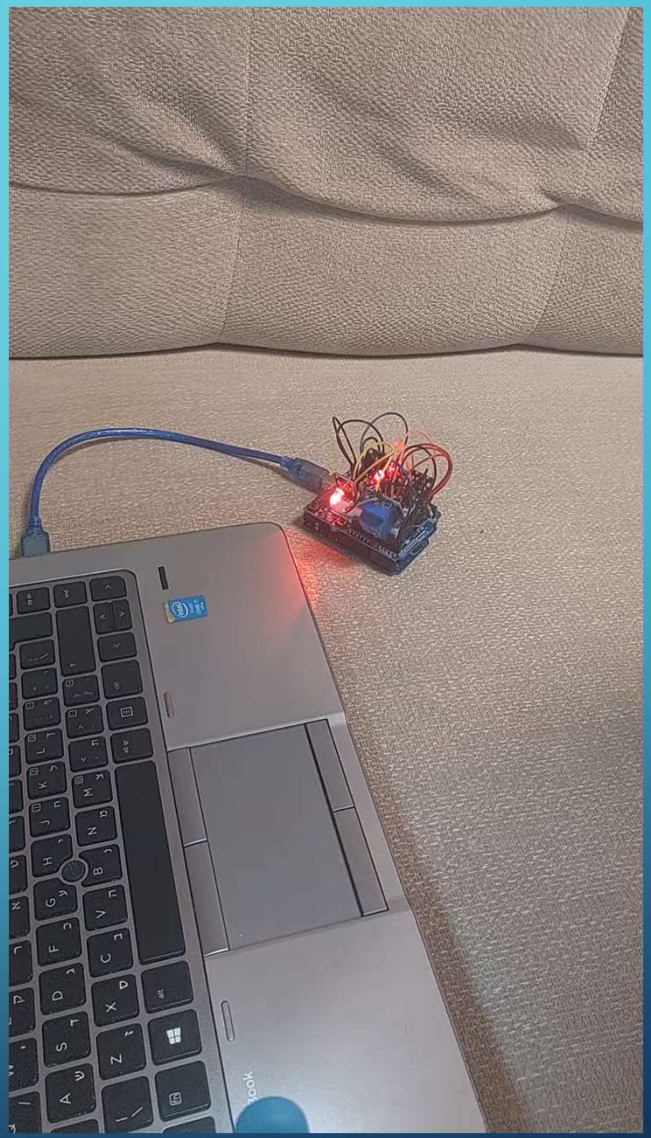
Part 6 of the code- continuation of the previews slide (ISR)

```
124     }
125     else if(HIGH_STATE)
126     {
127       noTone(buzzer_pin);
128       digitalWrite(relay_pin, HIGH); //turns ON light bulb
129       digitalWrite(led_pin, LOW);    //turns off led
130
131     }
132
133     else if(NORMAL_STATE)
134     {
135       noTone(buzzer_pin);
136       digitalWrite(relay_pin, LOW); //turns OFF light bulb
137       digitalWrite(led_pin, HIGH);  //turns ON led
138     }
139   }
140
141     else if(timer_cnt_rst==4)
142       {
143         tone(buzzer_pin, 1500); // Send 1.5KHz sound signal
144         timer_cnt_rst=0;
145       }
```

Part 7 of the code- continuation of the previews slide (ISR)

```
146         else
147         {
148             noTone(buzzer_pin);      // Stop sound
149             timer_cnt_rst++;
150         }
151     }
```

Bonus-a short video for demonstrating of the system



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