**Instructor:** Yerasyl Amanbek

**Student:** Amir Yakubov

**Date:** 08.04.2021

**Part 1. Introduction to Microcontrollers and Electronics (20%)**  
  
**1.** Briefly describe (1 sentence) at least 8 components in Arduino board

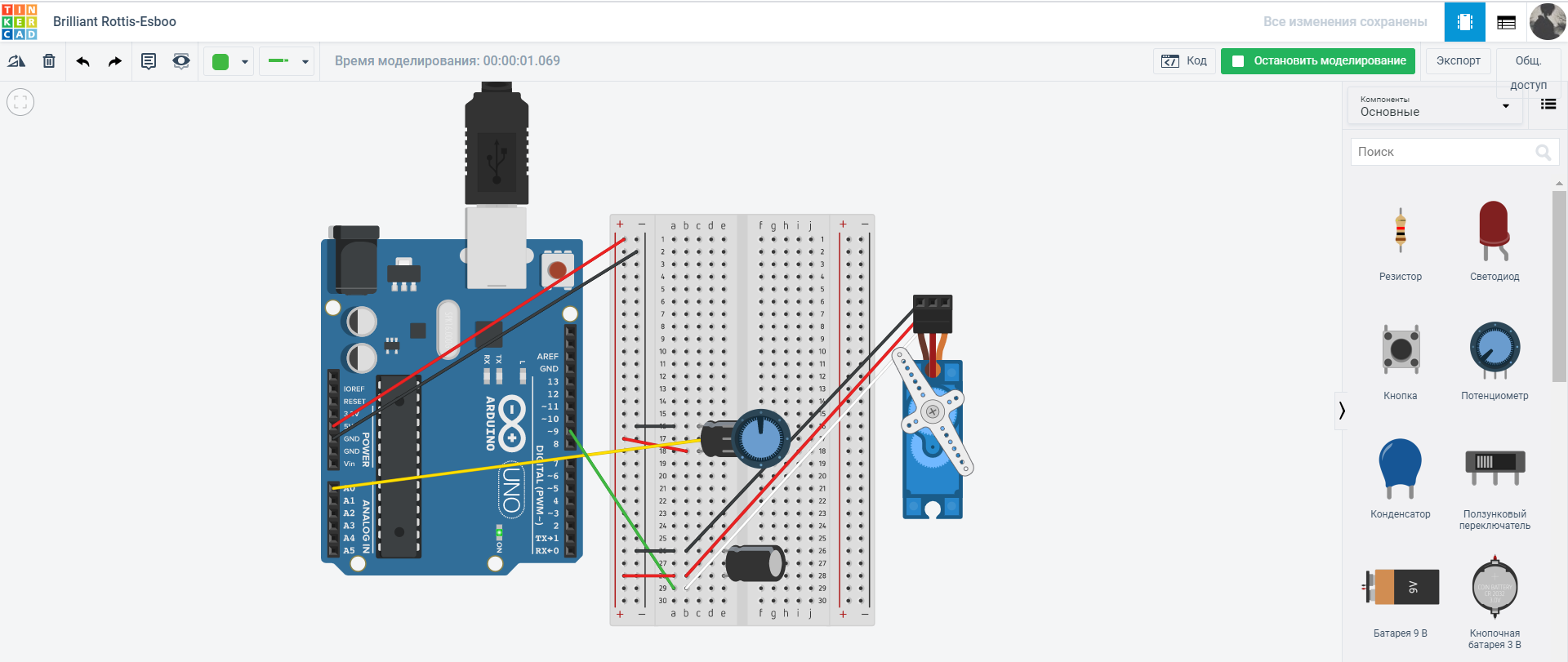
|  |  |
| --- | --- |
| **Name** | **Description** |
| Power connector | Used to power the Arduino, accepts voltage from 7-12 V. |
| USB port | Also, to charge, and a way to "communicate" with our Arduino. |
| ATmega microcontroller | In short, the heart of Arduino. |
| Reset Button | The button resets the ATmega microcontroller. |
| 5V pin | Used to supply current. |
| GND pin | The place where all current loops end and the current returns to the battery. |
| Analog | Reads the value from the specified analog pin, used with analogRead(). |
| Digital pins | Used to read and write some components to others. |
| Power LED | Shows that Arduino is receiving energy. |
| Pin 13 LED | The only built-in drive in the Arduino board, has an LED and a resistor. |
| TX and RX LEDs | Determines if there is a connection between the computer and Arduino |

**2.** Select 8 electronic components and describe them (1 sentence), also explain how they can be used in projects (1 sentence).

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Use in projects** |
| LEDs | Lights up when current passes through it. | For example, make a lighthouse, glowing clock or a picture, and other things. |
| Sound sensor | Analog models allow audio recognition and processing. | Turning on and off the light with a clap or some set sound. |
| Pushbuttons | Switches that close  a circuit when the button pressed. | It can be used as a switch, for example to turn on a light. |
| Piezo | Can create noise or a certain melody, also detect vibration. | Play the melody of the song. |
| Photoresistor | Resistor that changes resistance depending on the light incident on its surface. | Can be used, for example, in some supermarket to play music during the day or announcement. |
| Temperature sensor | It changes voltage depending on temperature. | Check the temperature in the room and set the desired one, for example, in the aquarium. |
| PIR | The sensor allows you to track the movement of objects that emit heat in a closed area. | Turn on alarm when motion is detected. Turn on the light at the entrance. |
| Current sensor | Measuring and monitoring flowing current. | For detecting non-working electrical appliances. |

**Part 2. Project 5 from Arduino Projects Book.** **(40%)**  
*Build project 5 from Arduino Projects Book in Tinkercad.*

**1.** Provide screenshot of the circuit



**2.** Record voltage across servo-motor

a) when potentiometer resistance is high

|  |  |
| --- | --- |
| servo-motor | Voltage |
| Legs (+ / -) (red / black) | 5.00 V |
| Legs (pulse / -) (white /black) | 700 mV |

b) when potentiometer resistance is low

|  |  |
| --- | --- |
| servo-motor | Voltage |
| Legs (+ / -) (red / black) | 5.00 V |
| Legs (pulse / -) (white /black) | 100 mV |

Your observation on behaviour of servo-motor when potentiometer value changes:

By changing the potentiometer values, the arrow starts to move counterclockwise. Also, having connected a multimeter, I noticed that by increasing the angle and value of the potentiometer, the pulse changes, voltage increases.  
  
**3.** Explain how current from 5V source flows through the circuit.   
The 5V current flows in 2 ways at once. I will describe this following the layout I built.

1) It flows and passes from plus to b18, further into the capacitor, then into the potentiometer and returns to ground.

2) The current flows from plus to a28, further into the capacitor, and the servo-motor, and returns to ground.  
  
**4.** Provide codes with comments.

#include <Servo.h> // import library to use servo

Servo myServo; //create an object to refer to the servo

int const potPin = A0; // constant for the output where the potentiometer is installed

int potVal; // variables to hold the analog input value

int angle; // angle we want the servo to move to

void setup() { // to tell the Arduino what pin our servo is attached to

myServo.attach(9);

Serial.begin(9600);

}

void loop() { // read the analog input and print out the value to the serial monitor

potVal = analogRead(potPin); // initialization

Serial.print("potVal: "); // output of this inscription

Serial.print(potVal); // output of this value

angle = map(potVal, 0, 1023, 0, 179); //change values ​​between 0-1023 to values ​​from 0 to 179

Serial.print(", angle: "); // output of this inscription

Serial.println(angle); // output of this value

myServo.write(angle); //moves the motor to the specified angle

delay(15); // the servo has time to change position

}

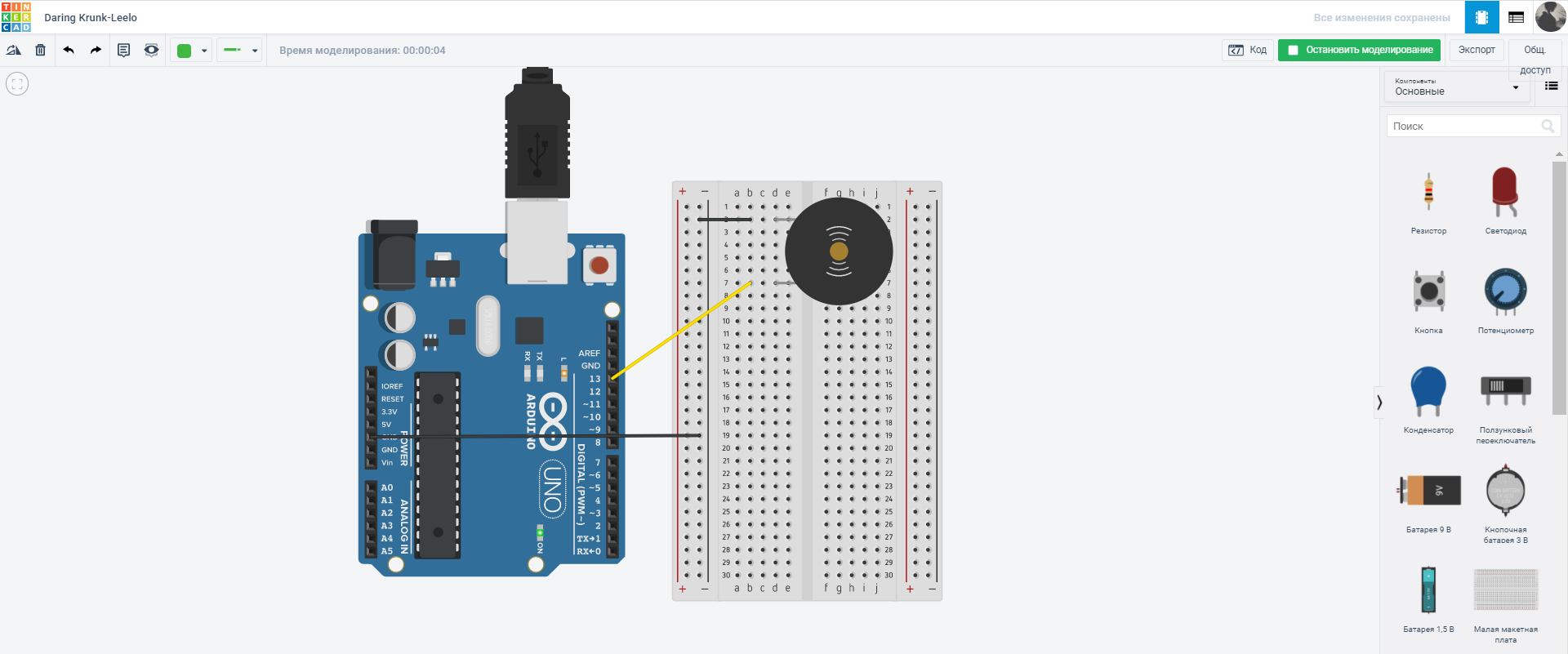
**5.** Provide link to the project  
<https://www.tinkercad.com/things/cIIdQOmMcqC>

**Part 3. Project 6 (modified) from Arduino Projects Book. (40%)**

*Build project 6 from Arduino Projects Book in Tinkercad.*

***Modify the project:*** *remove photoresistor and clear the code. Write new code, in which* ***PIEZO*** *will play a song (melody) of your choice. E.g., Star Wars theme, happy birthday, Game of thrones theme, etc.*

1. Provide screenshot of the modified circuit.



1. Explain what is PWM?

PWM is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts. PWM rapidly turns the output pin high and low over a fixed period. The change happens faster than the human eye can see.

1. Explain **tone()** function

Tone() is used to play a specific sound. You can specify the duration, if you do not do this, it will continue until the battery runs out or the noTone () function is called, Connects to a piezo or other speaker. It cannot reproduce several sounds at the same time, at the moment there is only one.

1. Explain **map()** function

The map function is intended to change one range of values into another range of values and a common use is to read an analogue input and change the output to a byte so the output would be from 0 to 255.

1. Provide codes with comments

//below is a library of notes

#define NOTE\_B0 31

#define NOTE\_C1 33

#define NOTE\_CS1 35

#define NOTE\_D1 37

#define NOTE\_DS1 39

#define NOTE\_E1 41

#define NOTE\_F1 44

#define NOTE\_FS1 46

#define NOTE\_G1 49

#define NOTE\_GS1 52

#define NOTE\_A1 55

#define NOTE\_AS1 58

#define NOTE\_B1 62

#define NOTE\_C2 65

#define NOTE\_CS2 69

#define NOTE\_D2 73

#define NOTE\_DS2 78

#define NOTE\_E2 82

#define NOTE\_F2 87

#define NOTE\_FS2 93

#define NOTE\_G2 98

#define NOTE\_GS2 104

#define NOTE\_A2 110

#define NOTE\_AS2 117

#define NOTE\_B2 123

#define NOTE\_C3 131

#define NOTE\_CS3 139

#define NOTE\_D3 147

#define NOTE\_DS3 156

#define NOTE\_E3 165

#define NOTE\_F3 175

#define NOTE\_FS3 185

#define NOTE\_G3 196

#define NOTE\_GS3 208

#define NOTE\_A3 220

#define NOTE\_AS3 233

#define NOTE\_B3 247

#define NOTE\_C4 262

#define NOTE\_CS4 277

#define NOTE\_D4 294

#define NOTE\_DS4 311

#define NOTE\_E4 330

#define NOTE\_F4 349

#define NOTE\_FS4 370

#define NOTE\_G4 392

#define NOTE\_GS4 415

#define NOTE\_A4 440

#define NOTE\_AS4 466

#define NOTE\_B4 494

#define NOTE\_C5 523

#define NOTE\_CS5 554

#define NOTE\_D5 587

#define NOTE\_DS5 622

#define NOTE\_E5 659

#define NOTE\_F5 698

#define NOTE\_FS5 740

#define NOTE\_G5 784

#define NOTE\_GS5 831

#define NOTE\_A5 880

#define NOTE\_AS5 932

#define NOTE\_B5 988

#define NOTE\_C6 1047

#define NOTE\_CS6 1109

#define NOTE\_D6 1175

#define NOTE\_DS6 1245

#define NOTE\_E6 1319

#define NOTE\_F6 1397

#define NOTE\_FS6 1480

#define NOTE\_G6 1568

#define NOTE\_GS6 1661

#define NOTE\_A6 1760

#define NOTE\_AS6 1865

#define NOTE\_B6 1976

#define NOTE\_C7 2093

#define NOTE\_CS7 2217

#define NOTE\_D7 2349

#define NOTE\_DS7 2489

#define NOTE\_E7 2637

#define NOTE\_F7 2794

#define NOTE\_FS7 2960

#define NOTE\_G7 3136

#define NOTE\_GS7 3322

#define NOTE\_A7 3520

#define NOTE\_AS7 3729

#define NOTE\_B7 3951

#define NOTE\_C8 4186

#define NOTE\_CS8 4435

#define NOTE\_D8 4699

#define NOTE\_DS8 4978

int a = 13; //buzzer

void setup() { // to tell the Arduino what pin our servo is attached to

pinMode(a, OUTPUT);

}

void loop() { //repeat the function

GameOfThrones();

}

void GameOfThrones() { //song play function by notes

tone(a, NOTE\_G4);

delay(500);

noTone(a);

tone(a, NOTE\_C4);

delay(500);

noTone(a);

tone(a, NOTE\_DS4);

delay(250);

noTone(a);

tone(a, NOTE\_F4);

delay(250);

noTone(a);

tone(a, NOTE\_D4);

delay(500);

noTone(a);

tone(a, NOTE\_G3);

delay(500);

noTone(a);

tone(a, NOTE\_AS3);

delay(250);

noTone(a);

tone(a, NOTE\_C4);

delay(250);

noTone(a);

tone(a, NOTE\_D4);

delay(1500);

noTone(a);

tone(a, NOTE\_F4);

delay(1500);

noTone(a);

tone(a, NOTE\_AS3);

delay(1000);

noTone(a);

tone(a, NOTE\_DS4);

delay(250);

noTone(a);

tone(a, NOTE\_D4);

delay(250);

noTone(a);

tone(a, NOTE\_F4);

delay(1000);

noTone(a);

tone(a, NOTE\_AS3);

delay(1000);

noTone(a);

tone(a, NOTE\_DS4);

delay(250);

noTone(a);

tone(a, NOTE\_D4);

delay(250);

noTone(a);

tone(a, NOTE\_C4);

delay(500);

noTone(a);

tone(a, NOTE\_GS3);

delay(250);

noTone(a);

tone(a, NOTE\_AS3);

delay(250);

noTone(a);

tone(a, NOTE\_C4);

delay(500);

noTone(a);

tone(a, NOTE\_F3);

delay(500);

noTone(a);

tone(a, NOTE\_G4);

delay(1000);

noTone(a);

tone(a, NOTE\_C4);

delay(1000);

noTone(a);

tone(a, NOTE\_DS4);

delay(250);

noTone(a);

tone(a, NOTE\_F4);

delay(250);

noTone(a);

tone(a, NOTE\_G4);

delay(1000);

noTone(a);

tone(a, NOTE\_C4);

delay(1000);

noTone(a);

tone(a, NOTE\_DS4);

delay(250);

noTone(a);

tone(a, NOTE\_F4);

delay(250);

noTone(a);

tone(a, NOTE\_D4);

delay(500);

noTone(a);

tone(a, NOTE\_G3);

delay(500);

noTone(a);

tone(a, NOTE\_AS3);

delay(250);

noTone(a);

tone(a, NOTE\_C4);

delay(250);

noTone(a);

tone(a, NOTE\_D4);

delay(500);

noTone(a);

}

**6.** Provide link to the project

<https://www.tinkercad.com/things/klKq47QBL2b>