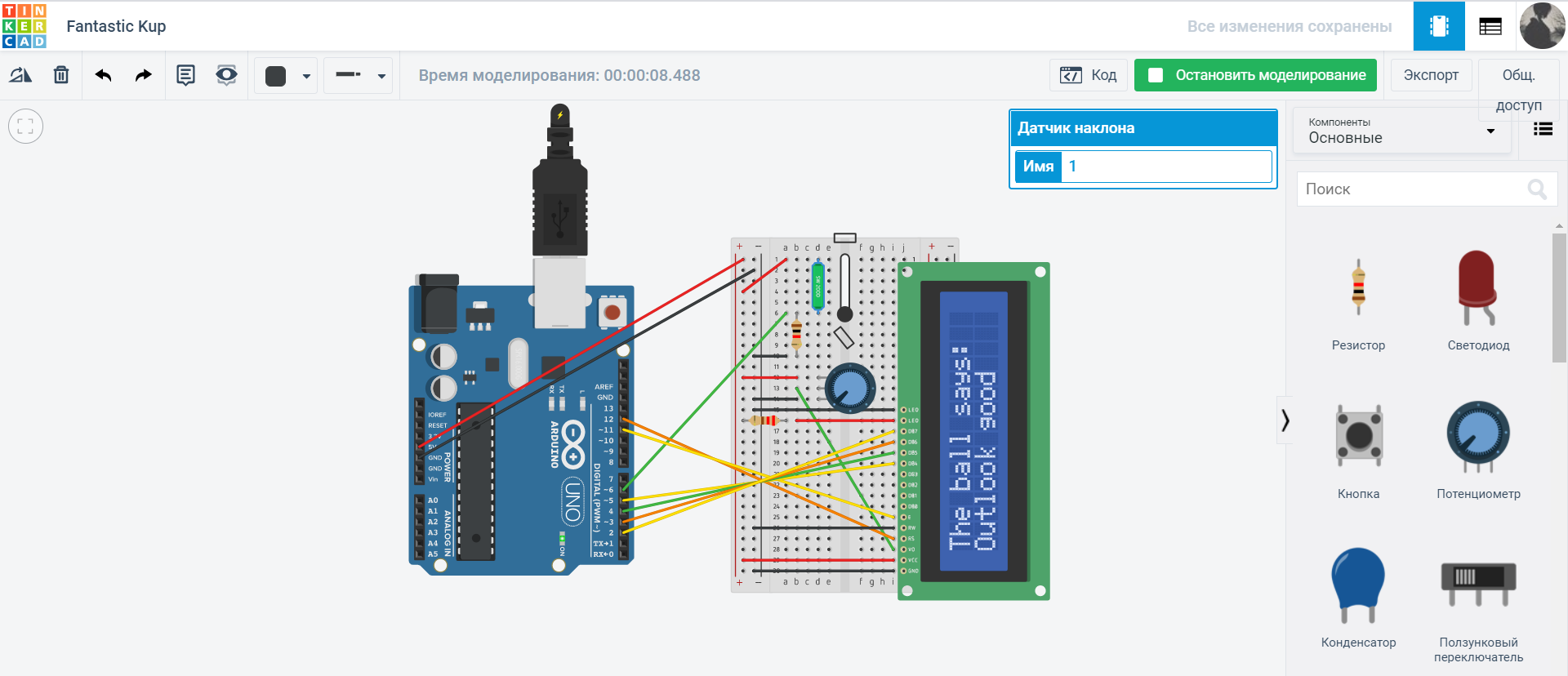
**Instructor:** Yerasyl Amanbek

**Student: Amir Yakubov**

**Date: 27.04.2021**

**Part 1. Project 11 from Arduino Projects Book.** **(30%)**  
*Build project 11 from Arduino Projects Book in Tinkerсad.*

1. Provide screenshot of the circuit.

  
  
**2.** Explain the purpose of each LCD pin.

1. GND for connecting breadboard to ground or other devices.
2. VCC is the inverse of ground. Connect with power.
3. V0 can give 0 to 5 volts, adjusts the difference on the display.
4. RS takes the value 0 or 1 and toggles them. 0 - data mode, 1 - command mode.
5. RW takes the value 0 or 1 and toggles them. 0 - write, 1 read.
6. E is connected to the microcontroller and is at a high level to perform the write / read process.
7. DB0-DB7 are used to send data to the display.
8. LED is connected to 5V.
9. LED is connected to GND.  
     
   **3.** Provide codes with comments.

#include <LiquidCrystal.h> //library

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

const int switchPin = 6;

int switchState = 0;

int prevSwitchState = 0;

int reply;

void setup() {

lcd.begin(16, 2);

pinMode(switchPin,INPUT); //switch-input

lcd.print("Ask the");

lcd.setCursor(0, 1);

lcd.print("Crystal Ball!"); //new row

}

void loop() {

switchState = digitalRead(switchPin);

if (switchState != prevSwitchState) {

if (switchState == LOW) {

reply = random(8); //any numbers(gives digits from 0 to 7)

lcd.clear();

lcd.setCursor(0, 0); //index of row

lcd.print("The ball says:");

lcd.setCursor(0, 1);

switch(reply){

case 0:

lcd.print("Yes");

break;

case 1:

lcd.print("Most likely");

break;

case 2:

lcd.print("Certainly");

break;

case 3:

lcd.print("Outlook good");

break;

case 4:

lcd.print("Unsure");

break;

case 5:

lcd.print("Ask again");

break;

case 6:

lcd.print("Doubtful");

break;

case 7:

lcd.print("No");

break;

}

}

}

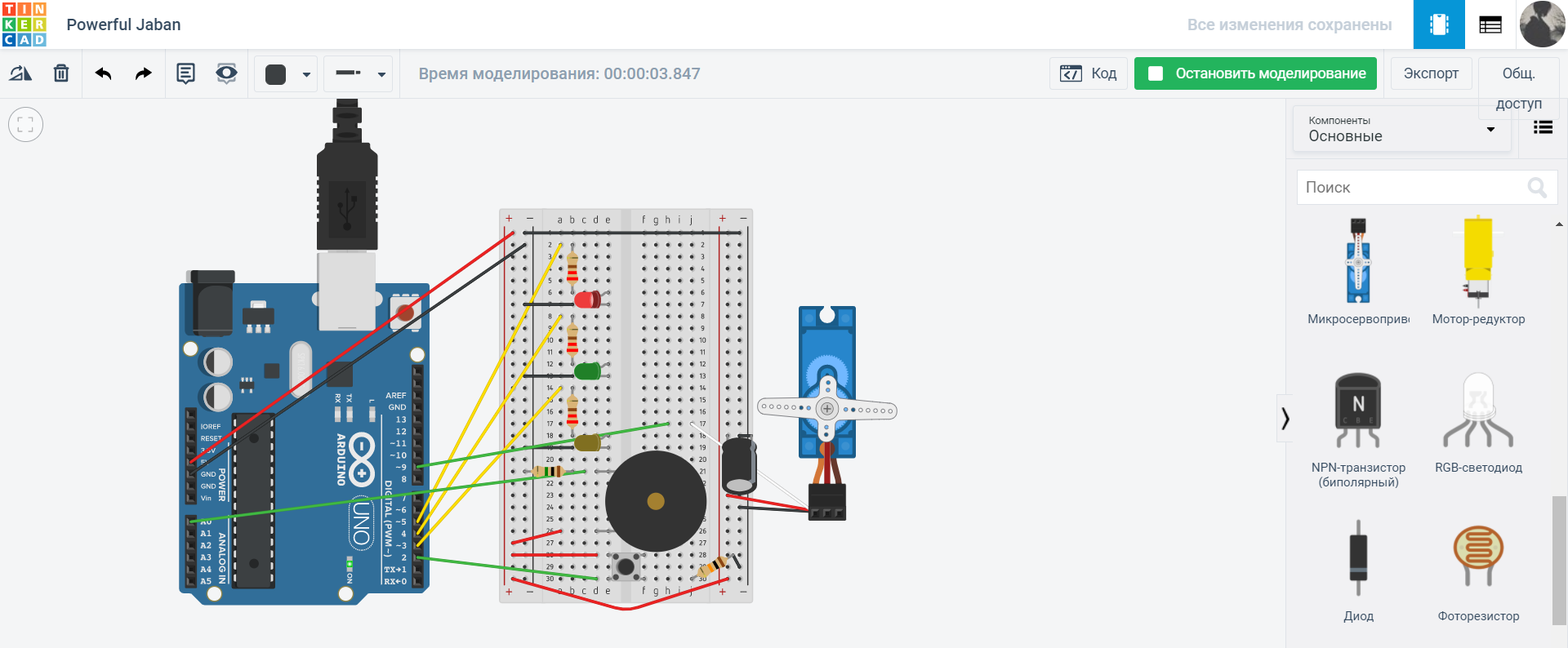
prevSwitchState = switchState;

}  
  
**4.** Provide link to the project.

<https://www.tinkercad.com/things/0uaJcixFpJn>

**Part 2. Project 12 from Arduino Projects Book.** **(30%)**  
*Build project 12 from Arduino Projects Book in Tinkercad.*

1. Provide screenshot of the circuit.



1. Explain what project does.

A project that can open something with the help of knocks. Set a specific command to open or close, use for this sound and the number of knocks.

1. List all components of the project and explain role of each.
2. The switch is needed to close.
3. The piezo is used to convert electrical sound into a signal.
4. There are 3 LEDs. Red, yellow and green. Red means blocked. Yellow should flash every time a knock is detected. And green means open.
5. Resistors to resist the flow of current.
6. A capacitor is a device for storing charge and energy of an electric field.
7. The servo motor in the project is like a lock or a lever for closing.
8. Provide codes with comments.

#include <Servo.h>

Servo myServo;

// Required hardware I/O connections

const int piezo = A0;

const int switchPin = 2;

const int yellowLed = 3;

const int greenLed = 4;

const int redLed = 5;

int knockVal;

int switchVal;

const int quietKnock = 10;

const int loudKnock = 100;

boolean locked = false;

int numberOfKnocks = 0;

void setup(){

myServo.attach(9);

pinMode(yellowLed, OUTPUT);

pinMode(redLed, OUTPUT);

pinMode(greenLed, OUTPUT);

pinMode(switchPin, INPUT);

Serial.begin(9600);

digitalWrite(greenLed, HIGH);

myServo.write(0);

Serial.println("The box is unlocked!");

}

void loop(){

// below conditions for open-lock

if(locked == false){

switchVal = digitalRead(switchPin);

if(switchVal == HIGH){

locked = true;

digitalWrite(greenLed,LOW);

digitalWrite(redLed,HIGH);

myServo.write(90); // rotate the servo 90 degrees to lock

Serial.println("The box is locked!");

delay (1000);

}

}

if(locked == true){

knockVal = analogRead(piezo);

if(numberOfKnocks < 3 && knockVal > 0){

if(checkForKnock(knockVal) == true){

numberOfKnocks++;

}

Serial.print(3-numberOfKnocks);

Serial.println(" more knocks to go");

}

if(numberOfKnocks >= 3){

locked = false;

myServo.write(0);

delay(20);

digitalWrite(greenLed,HIGH);

digitalWrite(redLed,LOW);

Serial.println("The box is unlocked!");

}

}

}

boolean checkForKnock(int value){

if(value > quietKnock && value < loudKnock){

digitalWrite(yellowLed, HIGH);

delay(50);

digitalWrite(yellowLed, LOW);

Serial.print("Valid knock of value ");

Serial.println(value);

return true;

}

else {

Serial.print("Bad knock value ");

Serial.println(value);

return false;

}

}

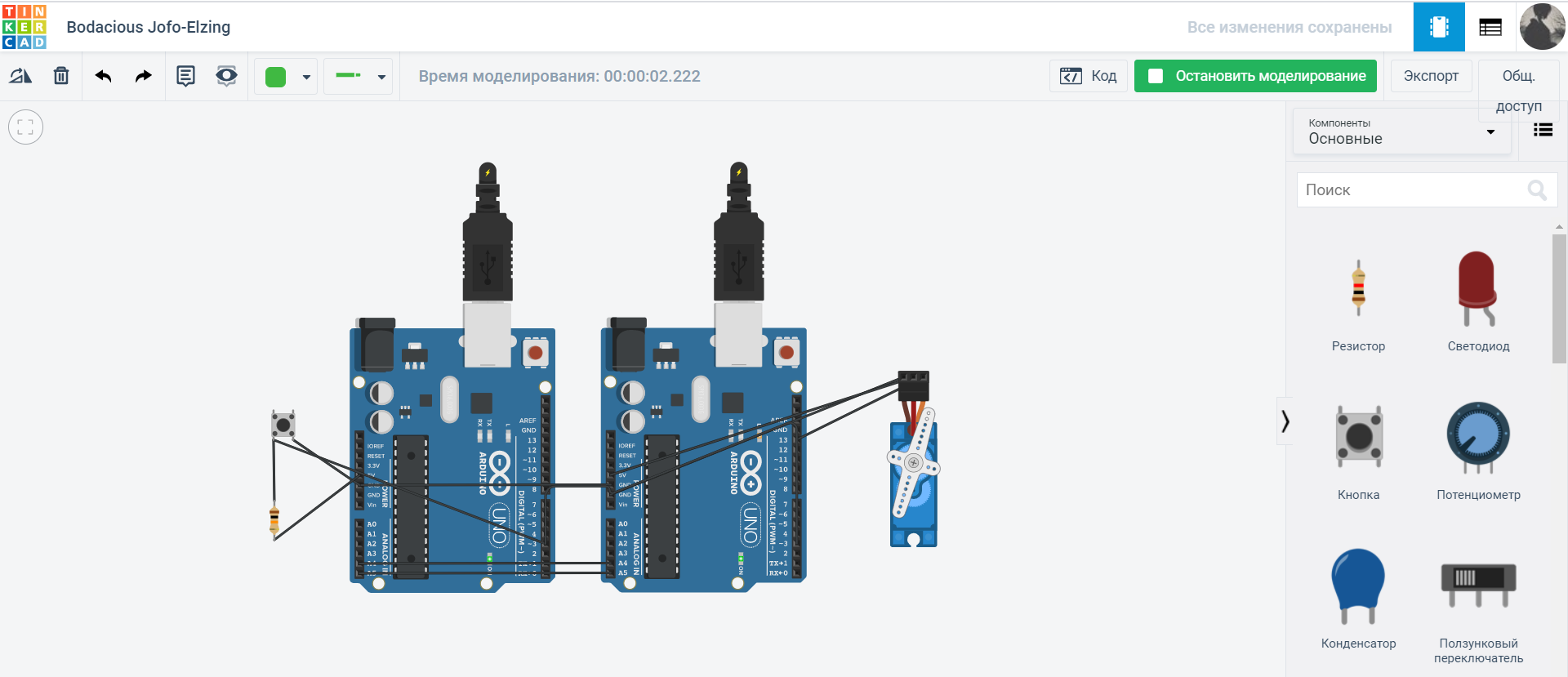
1. Provide link to the project.

<https://www.tinkercad.com/things/hb21kVviQdD>

**Part 3. Communication (I2C/SPI) (40%)**

*Connect 2 Arduinos using one of the interfaces (I2C/SPI), where 1 act as a master and another act as a slave. Master has 1 pushbutton; slave has 1 servo motor. When pushbutton is pressed the servo motor must rotate.*

1. Provide screenshot of the circuit.



1. Explain in detail the protocol that you have used.

The IIC serial communication protocol, also called I2C - Inter-Integrated Circuits, uses two bidirectional communication lines to transfer data, called the SDA (Serial Data) bus and the SCL (Serial Clock) bus. There are also two power lines. The SDA and SCL buses are pulled up to the power bus through resistors.

There is at least one Master in the network that initiates data transmission and generates synchronization signals. The network also has Slaves that transmit data at the request of the master. Each slave device has a unique address at which the master addresses it. The device address is indicated in the passport (datasheet). Up to 127 devices can be connected to one I2C bus, including several masters. Devices can be connected to the bus during operation, i.e., it supports hot plugging.

1. Provide codes with comments.

Arduino 1:

#include "Wire.h" //I2C library

void setup() { //set

Wire.begin();

pinMode(3, INPUT);

}

void loop() { //Transmission

Wire.beginTransmission(1);

Wire.endTransmission();

delay(10);

}

Arduino 2:

#include "Wire.h" //I2C library

#include<Servo.h> //library of servo

Servo Myservo;

void setup() { //set

Wire.begin(1);

Myservo.attach(13);

}

void loop() {

if(digitalRead(3)==LOW){ //if button is pushed

Myservo.write(180); //rotate to 180

}

else Myservo.write(0); // if not, then no actions

}

1. Provide link to the project.

<https://www.tinkercad.com/things/e5XDArWSd3Q>