Examples of Electrostatic Discharge Testing

Consumer electronics, professional electronic equipment, and in teaching electronics experiments are all sensitive to electrostatic discharge. Critical components like processors, microcontrollers, amplifiers etc. come in ESD protective packages and with warnings for usage related to ESD levels. A human assembling the prototype or using it is usually on different potential (typically this potential difference can be up to $\pm 15\,kV$ or more) if special safety measures are not preformed (e.g., grounding of the prototype and the human by using ESD anti-static wrist-strap band). The objective of this document is to present several non-destructive experiments in which effects of ESD are easily repeatable, while demonstrating the usage of developed open-hardware for ESD testing.

Each example comes with description, in-detail instructions and a link to a short video of the ESD experiment.

1. Controlling RGB LED with STM32NUCLEO-L432KC

The first example is simple RGB diode controlled by STM32NUCLEO-L432KC [1].

1.1. Objectives

- Turn on and alternate colors of RGB LED using STM32NUCLEO-L432KC microcontroller.
- Apply ESD pulse(s) to the pins to check immunity to ESD and observe the effects of ESD.

1.2. Assembly instructions

The following components are needed for assembly:

(1) STM32NUCLEO-L432KC microcontroller,

(2) KY-016 Full Color RGB LED module,

(3) 4 female-female jumpers (wires),

(4) USB cable for connecting the microcontroller to the PC and

(5) PC with installed Arduino IDE [2]. Note that this microcontroller can be programmed with Arduino IDE which has been chosen due to ease of use.

The schematic is presented in Fig. 1.1. Step-by-step instructions for system assembly for are as follows.

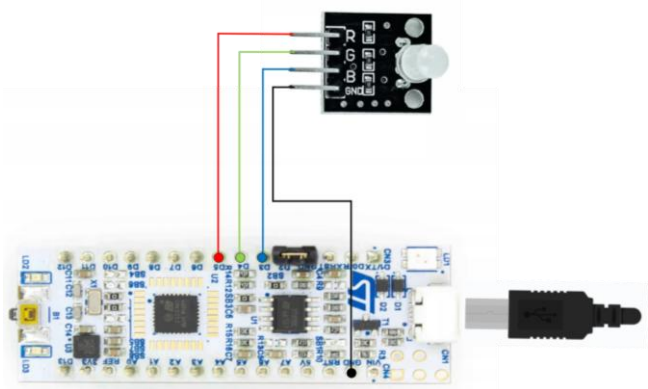

Fig. 1.1. Schematic for controlling RGB LED with STM32NUCLEO-L432KC.

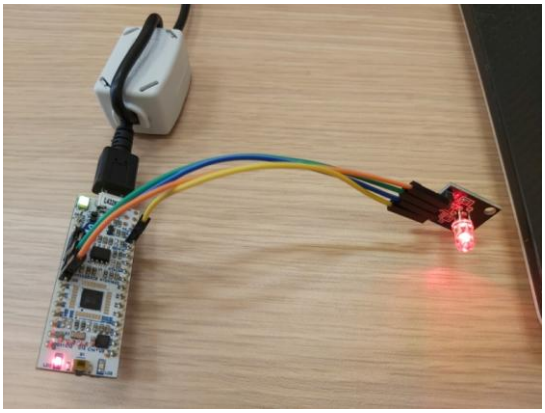

Fig. 1.2. Assembled prototype for controlling RGB LED with STM32NUCLEO-L432KC.

(1) Connect the ground pin from the LED module to the ground pin from the microcontroller, as it is shown in the schematic in Fig. 1.1.

(2) Connect the microcontroller to the PC via USB cable.

(3) Open Arduino IDE. Install STM32 MCU based boards library from the Boards Manager in Arduino IDE. In Tools>Board choose Nucleo-32, in Tools>Board part number choose Nucleo L432KC.

(4) Copy the code below and upload it to the microcontroller.

Full setup for the first example is shown in Fig. 1.2.

Complete microcontroller code for this example is given below.

```
// Defining variable and the GPIO pin
int redPin= 5;
int greenPin = 4;
int bluePin = 3;
void setup()
{
  // Defining the pins as OUTPUT
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}
void loop()
{
  setColor(127,127,127); // Light Blue
  delay(200);
  setColor(255, 0, 0); // Red
  delay(200);
  setColor(0, 255, 0); // Green
  delay(200);
  setColor(255, 255, 255); // White
  delay(200);
  setColor(255,153,204); // Pink
  delay(200);
  setColor(170, 0, 255); // Purple
  delay(200);
  setColor(0, 0, 255); // Blue
  delay(200);
}
void setColor(int redValue, int greenValue,  int blueValue)
{
  analogWrite(redPin, redValue);
  analogWrite(greenPin, greenValue);
  analogWrite(bluePin, blueValue);
}
```

1.3. Electrostatic Discharge Testing

Set the O-ESD to contact discharge mode. Select voltage level to +1 kV. Connect the ground strap of O-ESD to GND pin of STM32NUCLEO-L432KC. Apply 1 kV ESD pulse to the pin D13 of STM32NUCLEO-L432KC. This can be done either directly by pressing the (sharp) tip of O-ESD to pin D13, or by connecting D13 with a wire that is pressed against the tip of O-ESD at the time of the pulse generation. LED diode should stop changing colors immediately after applying 1 kV ESD pulse. Restart the microcontroller in order to start functioning again.

A short video for this example is available at:

https://o-esd.etf.bg.ac.rs/forgejo/dragan.olcan/O-ESD/src/branch/main/Demos/01_LED/O-ESD_demo_01_LED.mp4.

2. Water Sensor with Arduino Nano Every

The second example consists of a water sensor connected to Arduino Nano Every [3]. The purpose of the system is to detect the presence of water and to signalize it.

2.1. Objectives

- Connect the water sensor to Arduino Nano Every and detect the presence of water. Signalize the water presence by powering the on-board LED.

- Apply ESD pulse(s) to the pins to check immunity to ESD and observe the effects of ESD.

2.2. Assembly Instructions

List of components needed for assembly:

(1) Arduino Nano Every microcontroller,

(2) Water sensor compatible with Arduino,

(3) three female-female jumpers,

(4) USB cable for connecting the microcontroller to the PC,

(5) PC with installed Arduino IDE [2] and

(6) cup filled with water.

The schematic is given in Fig. 2.1. Step-by-step instructions for building this prototype are as follows.

(1) Connect the water sensor to the microcontroller, as shown in the schematic in Fig. 2.1.

(2) Connect the microcontroller to the PC via USB cable.

(3) Open Arduino IDE. In Tools>Board choose Arduino Nano Every.

(4) Copy the code below and upload it to the microcontroller.

Full setup for the second example is shown in Fig. 2.2.

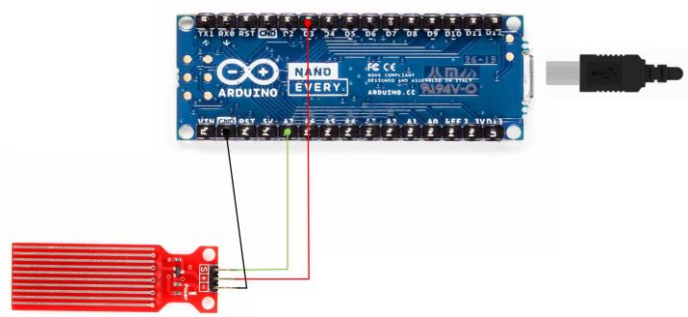

Fig. 2.1. Schematic for water sensor with Arduino Nano Every.

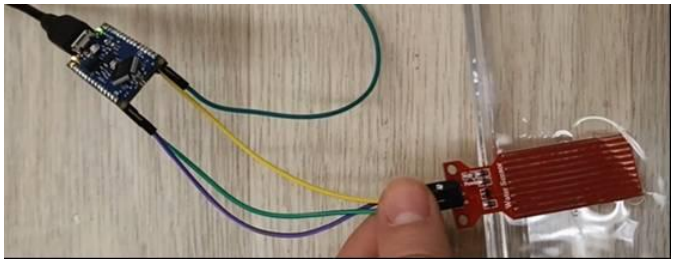

Fig. 2.2. Assembled prototype for water sensor with Arduino Nano Every. The water sensor is submerged into water.

Complete microcontroller code for this example is given below.

```
#define POWER_PIN 3
#define SIGNAL_PIN A7
int value = 0; // variable to store the sensor value

void setup()
{
  Serial.begin(9600);
  pinMode(POWER_PIN, OUTPUT);   // configure D3 pin as an OUTPUT
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(POWER_PIN, LOW); // turn the sensor OFF
}

void loop()
{
  digitalWrite(POWER_PIN, HIGH);  // turn the sensor ON
  delay(10);                      // wait 10 milliseconds
  value = analogRead(SIGNAL_PIN); // read the analog value from sensor
  if (value>0) digitalWrite(LED_BUILTIN,HIGH);
  if (value==0) digitalWrite(LED_BUILTIN,LOW);
  digitalWrite(POWER_PIN, LOW);   // turn the sensor OFF
  delay(1000);
}
```

2.3. Electrostatic Discharge Testing

The O-ESD to contact discharge mode. Select voltage level to +4 kV. Connect the ground strap of O-ESD to GND pin of Arduino Nano Every. Apply 4 kV ESD pulse to the 5V pin of Arduino Nano Every. This can be done either directly by pressing the (sharp) tip of O-ESD to the 5V pin, or by connecting the 5V pin with a wire that is pressed against the tip of O-ESD at the time of the pulse generation. False water detection should occur after applying 4 kV ESD pulse.

A short video for this example is available at:

https://o-esd.etf.bg.ac.rs/forgejo/dragan.olcan/O-ESD/src/branch/main/Demos/02_water/O-ESD_demo_02_water.mp4.

3. Stopwatch on TI MSP-EXP430FR6989

The third example is based on stopwatch that is preinstalled on TI MSP-EXP430FR6989 microcontroller [4].

3.1. Goals

- Control stopwatch on TI MSP-EXP430FR6989 using on-board switches.
- Apply ESD pulse(s) to the pins to check immunity to ESD and observe the effects of ESD.

3.2. Assembly Instructions

List of components needed for assembly:

(1) TI MSP-EXP430FR6989 microcontroller,

(2) USB cable for connecting microcontroller to power source and

(3) phone charger.

Step-by-step instructions for building a system for this example are as follows.

(1) Connect the microcontroller to the phone charger via USB cable.

(2) Plug in the phone charger.

Full setup for the third example is shown in Fig. 3.1.

Fig. 3.1. Stopwatch on TI MSP-EXP430FR6989.

3.3. Electrostatic Discharge Testing

Set the O-ESD to contact discharge mode. Connect the ground strap of O-ESD to GND pin of TI MSP-EXP430FR6989. Then

(1) Select voltage level to +1 kV. Apply 1 kV ESD pulse to the GND pin of TI MSP-EXP430FR6989. This can be done either directly by pressing the (sharp) tip of O-ESD to GND pin, or by connecting the GND pin with a wire that is pressed against the tip of O-ESD at the time of the pulse generation. Red on-board LED turns on and stopwatch time resets or stops. Restart the microcontroller in order to start functioning again;

(2) Select voltage level to +3 kV. Apply 3 kV ESD pulse to the GND pin of TI MSP-EXP430FR6989. This can be done either directly by pressing the (sharp) tip of O-ESD to GND pin, or by connecting the GND pin with a wire that is pressed against the tip of O-ESD at the time of the pulse generation. Red on-board LED turns on and microcontroller resets (software reset);

(3) Select voltage level to +1 kV. Apply 1 kV ESD pulse to the pin 2.1 of TI MSP-EXP430FR6989. This can be done either directly by pressing the (sharp) tip of O-ESD to the pin 2.1, or by connecting the pin 2.1 with a wire that is pressed against the tip of O-ESD at the time of the pulse generation. Microcontroller resets (software reset).

A short video for this example is available at:

https://o-esd.etf.bg.ac.rs/forgejo/dragan.olcan/O-ESD/src/branch/main/Demos/03_stopwatch/O-ESD_demo_03_stopwatch.mp4.

4. Sound Sensor with Arduino MKR ZERO

The fourth example is based on sound sensor with Arduino MKR Zero [5].

4.1. Objectives

- Control ON/OFF status of on-board LED by detecting loud sounds (e.g. clapping or snapping fingers) using sound sensor connected to Arduino MKR ZERO.
- Apply ESD pulse(s) to the pins to check immunity to ESD and observe the effects of ESD.

4.2. Assembly Instructions

List of components needed for assembly:

(1) Arduino MKR Zero microcontroller,

(2) USB cable for connecting microcontroller to PC,

(3) three female-female jumpers,

(4) PC with installed Arduino IDE [2] and

(5) HW-484 sound sensor module.

The schematic is given in Fig. 4.1. Step-by-step instructions for building a system for this example are as follows.

(1) Connect the microcontroller to the PC via USB cable.

(2) Connect the sound sensor module as shown in the schematic in Fig. 4.1.

(3) Open Arduino IDE. In Tools>Board choose Arduino MKR ZERO (programming port).

(4) Copy the code below and upload it to the microcontroller.

Full setup for the fourth example is shown in Fig. 4.2.

```
int sensor=9;
int led=13;
bool is_on=false;
void setup()
{
  pinMode(sensor, INPUT);
  pinMode(led,OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop()
{
  int data = digitalRead(sensor);
  if (data==1)
  {
    if (is_on==true)
    {
      digitalWrite(LED_BUILTIN,LOW);
      digitalWrite(led,LOW);
      is_on=false;
    }
    else
    {
      digitalWrite(LED_BUILTIN,HIGH);
      digitalWrite(led,HIGH);
      is_on=true;
    }
  }
}
```

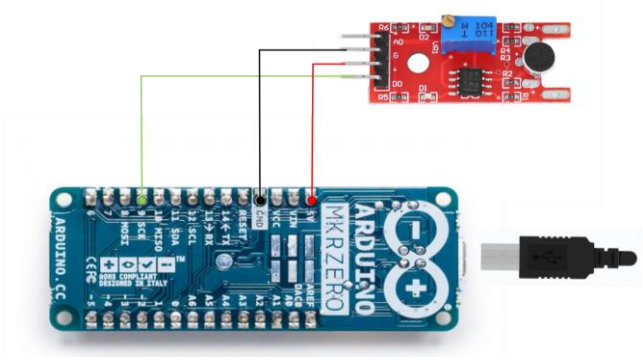

Fig. 4.1. Schematic for sound sensor with Arduino MKR ZERO.

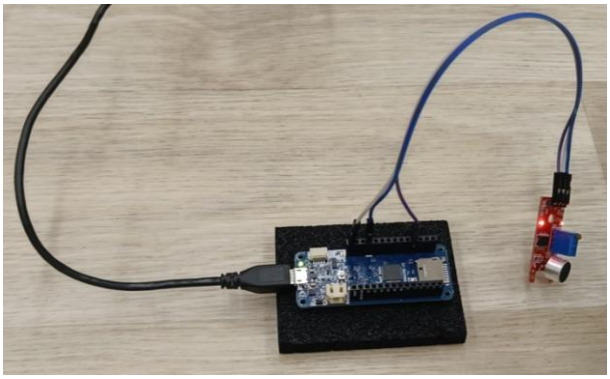

Fig. 4.2. Assembled sound sensor with Arduino MKR Zero.

4.3. Electrostatic Discharge Testing

Set the O-ESD to contact discharge mode. Select voltage level to +1 kV. Connect the ground strap of O-ESD to GND pin of Arduino MKR ZERO. Apply 1 kV ESD pulse to the pin D10 of Arduino MKR ZERO. This can be done either directly by pressing the (sharp) tip of O-ESD to the pin D10, or by connecting the pin D10 with a wire that is pressed against the tip of O-ESD at the time of the pulse generation. False sound detection should occur after applying 1 kV ESD pulse.

A short video for this example is available at:

https://o-esd.etf.bg.ac.rs/forgejo/dragan.olcan/O-ESD/src/branch/main/Demos/04_sound/O-ESD_demo_04_sound.mp4.

5. Personal Computer

The fifth example is based on a personal computer.

5.1. Goals

- Turn on a personal computer.
- Apply ESD pulse(s) to a USB port pin to check immunity to ESD and observe the effects of ESD.

5.2. Assembly Instructions

List of components needed for assembly:

(1) PC,

(2) old USB cable, cut in half.

Step-by-step instructions for building a system for this example are as follows.

(1) Turn on a PC.

(2) Connect an old USB cable (with accessible wires) to a PC USB port.

Full setup for the fifth example is shown in Fig. 5.1 and USB cable wiring is shown in Fig. 5.2.

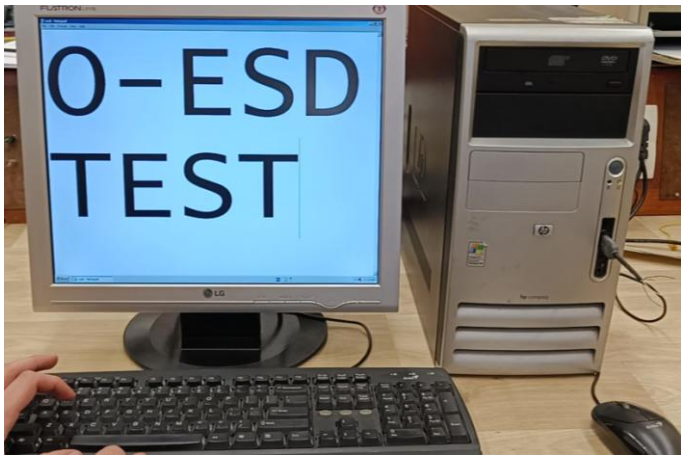

Fig. 5.1. PC with a USB cable connected to a USB port.

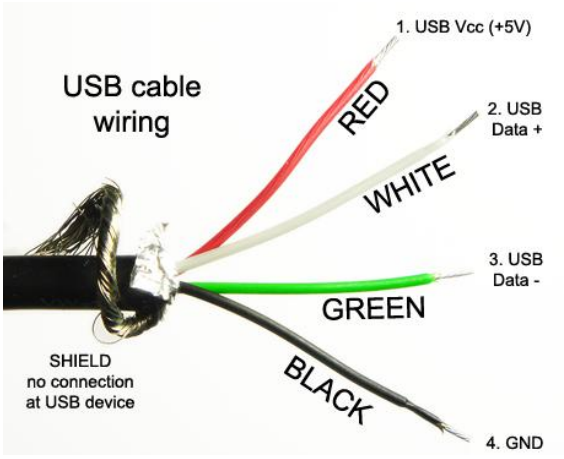

Fig. 5.2. USB cable wiring.

5.3. Electrostatic Discharge Testing

Set the O-ESD to contact discharge mode. Connect the ground strap of O-ESD to a PC ground. Then, select voltage level to +4 kV. Apply 4 kV ESD pulse to a data wire (white or green) of a USB cable connected to a USB port of the PC. The PC turns off.

A short video for this example is available at:

https://o-esd.etf.bg.ac.rs/forgejo/dragan.olcan/O-ESD/src/branch/main/Demos/05_PC/O-ESD_demo_05_PC.mp4.

References

[1] https://www.st.com/en/evaluation-tools/nucleo-l432kc.html, website to STM32NUCLEO-L432KC microcontroller, last accessed: 20/03/2025.
[2] https://www.arduino.cc/en/software, website to Arduino IDE, last accessed: 20/03/2025.
[3] https://docs.arduino.cc/hardware/nano-every, website to Arduino Nano Every microcontroller, last accessed: 20/03/2025.
[4] https://www.ti.com/tool/MSP-EXP430FR6989, website to TI MSP-EXP430FR6989 microcontroller, last accessed: 20/03/2025.
[5] https://docs.arduino.cc/hardware/mkr-zero, website to Arduino MKR ZERO microcontroller, last accessed: 20/03/2025.

Revision history

Version	Date	Author(s)	Description
1.0	23/03/2025	A.K., D.N., D.O.	Initial release.