

Task 1.

Design an inductor as shown in pictures and additional description below using any software (AutoCAD, Fusion360 etc.). Calculate inductance of created inductor. The result of realization must be a file with 3D-model and inductance value.

$$L = 5 \text{ мкГн}$$

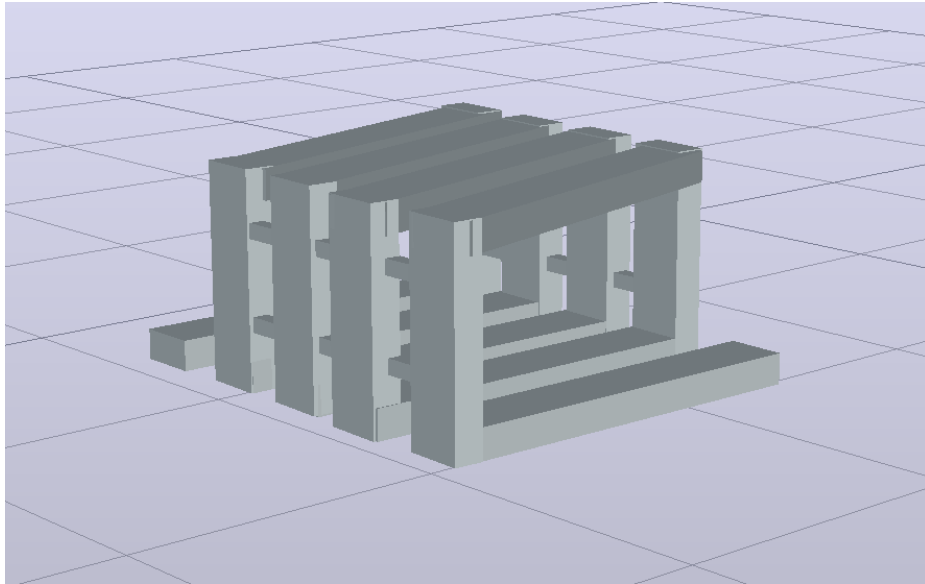


Рис. 1 3D вид

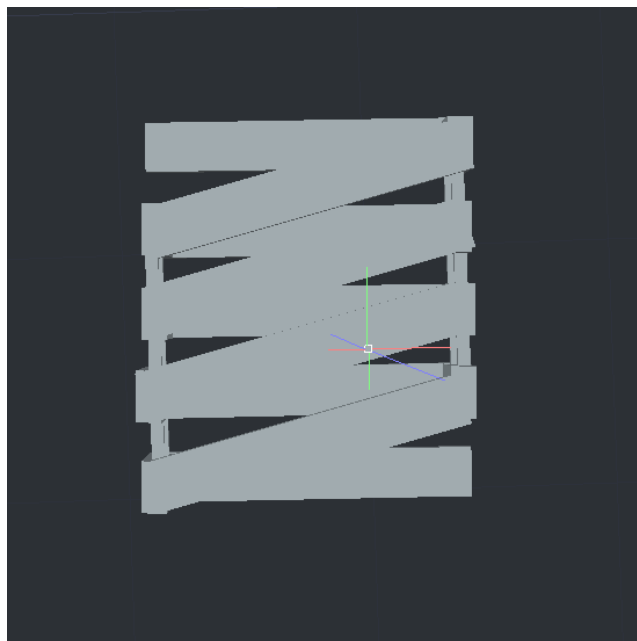


Рис. 2 Вид зверху

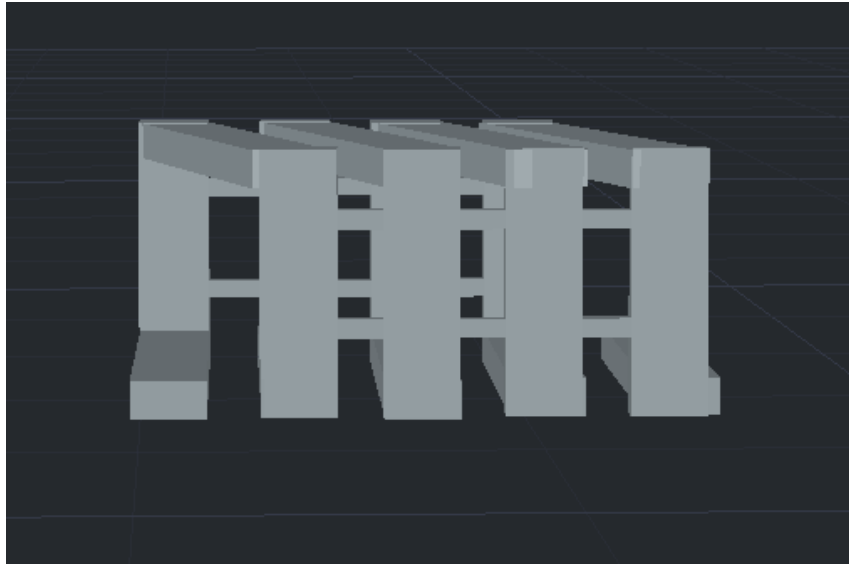


Рис. 3 Вид збоку

Task 2.

A simple pinout of one of the Dialog's IPS is illustrated on Figure 1. When ON voltage reaches ON_VIH level, the IPS will turn On and when ON voltage reaches ON_VIL level, the IPS will turn OFF. Depends on power supply voltage VDD and fixed VIN voltage, please develop a simple algorithm for automatic finding of ON_VIH and ON_VIL levels:

1. VDD = 2.5 V, 3.3 V, 5 V; 5.5 V; VIN = 2.5 V
2. Define minimum necessary equipment for measurements
3. Create a simple block diagram of the Test Setup

Необхідне обладнання для вимірювань:

- Мультиметр
- Потенціометр
- Макетна плата
- З'єднуючі кабелі

Схематичне зображення досліджуваного об'єкта:

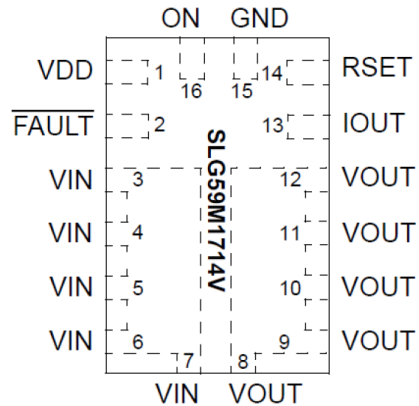


Рис. 4 Контролер живлення

Принцип роботи алгоритму:

Контролер постійно вимірює напругу на вході та проводить перевірку, при досягненні певного рівня вмикається або вимикається певний вихід.

Цей алгоритм підходить для мікрочіпу, так як завдання полягало у розробці алгоритму знаходження певного рівня напруги і включення певних входів та виходів.

Код програми:

```
#define PIN_LED 8 // Initialization of LED foam.
float input_volt = 0.0; // Variable for output
float data = 0.0; // Variable for converting voltage values
float formula = 0.0; // Variable for the voltage divider calculation formula
float r1 = 4650.0; // First resistor
float r2 = 2350.0; // Second resistor
float x = 1.56; // Basic power supply at input A0
const int delay_prog = 500; // Wait between processes
const int delay_end = 100; // Delay in the end of program

void setup() {
  Serial.begin(9600);
}

void loop() {
  measure(); // Performance of input voltage measurement
  delay(delay_prog);
  search(); // Search and comparison of input current
  delay(delay_prog);
}
```

```

void search() { // Determining the desired value of the input voltage
if (input_volt >= 0 && input_volt < 0.5) { // Device OFF condition
    Serial.println("0 ... 0.5");
    Serial.println("...");
    digitalWrite(PIN_LED, LOW);
    delay(delay_end);
}
else if (input_volt >= 0.5 && input_volt < 1.3) { // Device ON condition
    Serial.println("0.5 ... 1.3");
    Serial.println("...");
    digitalWrite(PIN_LED, HIGH);
    delay(delay_end);
}
else if (input_volt >= 1.3 && input_volt < 2.5) { // Device OFF condition
    Serial.println("1.3 ... 2.5");
    Serial.println("...");
    digitalWrite(PIN_LED, LOW);
    delay(delay_end);
}
else if (input_volt >= 2.5 && input_volt < 3.5) { // Device ON condition
    Serial.println("2.5 ... 3.5");
    Serial.println("...");
    digitalWrite(PIN_LED, HIGH);
    delay(delay_end);
}
}

void measure() { // Function of calculation and transfer of input data
int analogvalue = analogRead(A0); // Reading data from port A0
float formula = r2/(r1+r2); // Calculate the voltage divider
data = (analogvalue * x) / 1024.0; // Voltage conversion
input_volt = data / formula; // Calculate the input voltage

Serial.print("analogvalue = ");
Serial.println(analogvalue);
Serial.print("Voltage= ");
Serial.println(input_volt);
}

```

Task 3. Devise high accuracy voltmeter

Devise an electrical circuit for device that can measure input voltage in range from 0 V to 30 V with automatic range switching and with max 0.02 % accuracy. Once voltage is measured, the result should be transferred to PC through COM port

or USB and saved on HDD (you can use any convenient file format for data storage). Also please describe the device operation.

Для створення вольтметра потрібно:

- Мікроконтроллер Arduino
- 3 резистора (2*10кОм та 1*100кОм)
- Макетна плата

На макетній платі розмістити Arduino. До аналогового порта A0 мікроконтролера підключити дільник напруги, через який подаватиметься напруга для вимірювання (рис. 0 схема дільника). Дільник напруги складається з двох резисторів: резистор з більшим номіналом підключається одним кінцем до вимірюваного джерела напруги та другим кінцем до резистора з меншим номіналом, який підключений до «землі». В розрив між двома резисторами підключається порт A0. Формула для розрахунку напруги на виході дільника напруги:

$$V_{out} = V_{in} * \left(\frac{R_2}{R_1 + R_2} \right) \quad (1)$$

Дільник напруги використовується для вимірювання напруги яка є більшою за 5V, так як контролер не розрахований на більшу напругу. Для вимірювання в діапазоні 0...30V, потрібно підібрати потрібні номінали резисторів, які можна вирахувати з формули. А саме: $R_1 = 50\text{кОм}$, $R_2 = 10\text{кОм}$. Джерело напруги яке вимірюється, обов'язково потрібно підключити «мінусовим» контактом до «землі» макету, а іншим контактом до дільника напруги.

Вхідний сигнал потрібно перетворити у цифрове значення. АЦП Arduino має розрядність 10 біт, тому значення може коливатись від 0 до 1023. Зчитування вхідного значення виконується функцією `analogRead()`.

Опорну напруга на АЦП = 1.56V, розрахунок напруги на вході A0:

$$data = \frac{A0 * 1.56}{1024} \quad (2)$$

Де A0 – напруга на порті A0, 1024 – розрядність АЦП.

$$IV = \frac{data}{V_{out}} \quad (3)$$

Де IV – вхідна напруга, V_{out} = опір дільника.

Виведення даних на ПК відбувається за допомогою COM порта (Arduino IDE або Putty). Для збереження даних можна використати програму Putty.

Електрична схема пристрою:

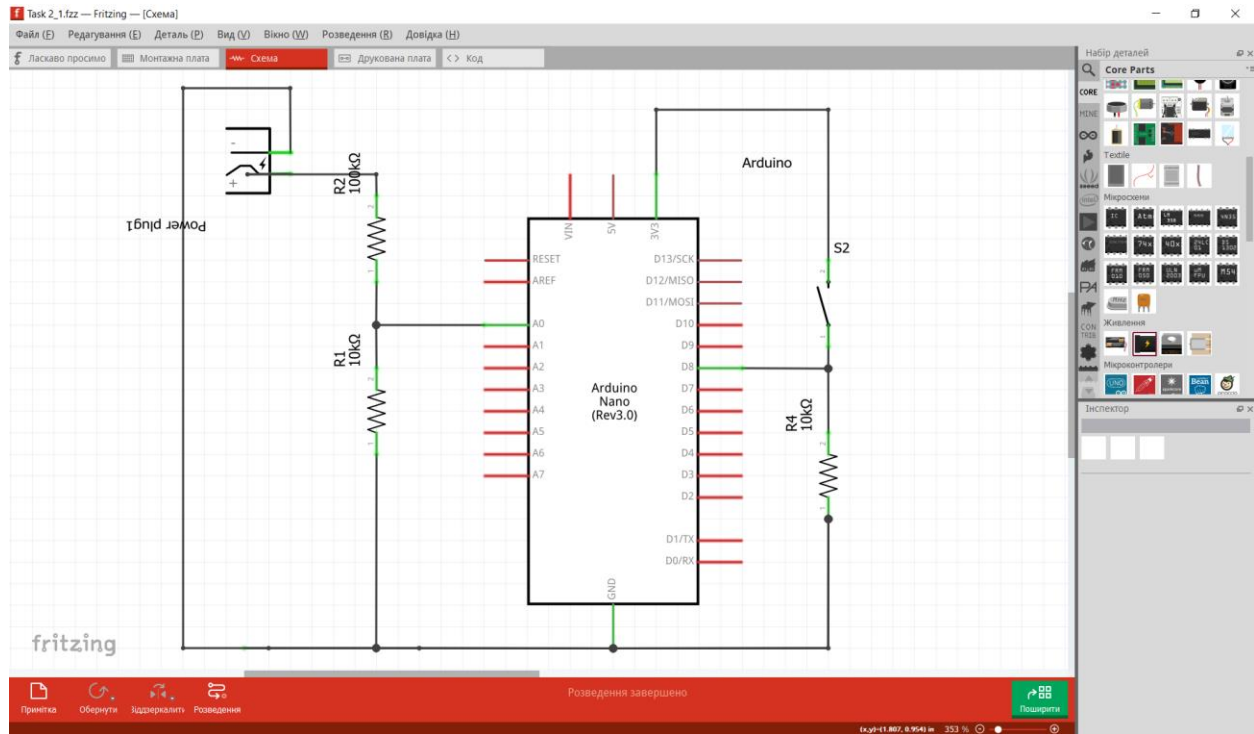


Рис. 5 Електрична схема вимірювального пристрою

Для реалізації вольтметра я використав:

- Мікроконтролер Arduino Nano
- 3 резистора (2*4.65кОм та 1*2.35кОм)
- Макетна плата
- Потенціометр 10кОм
- З'єднуючі кабелі
- Кнопка

На макетній платі розмістив Arduino. До аналогового порта A0 мікроконтролера підключив дільник напруги, через який подається напруга для вимірювання (рис. 0 схема дільника). Дільник напруги складається з двох резисторів: резистор з більшим номіналом підключається одним кінцем до вимірюваного джерела напруги та другим кінцем до резистора з меншим номіналом, який піклучений до «землі». В розрив між двома резисторами

підключається порт A0. Формула для розрахунку напруги на виході дільника напруги:

$$V_{out} = V_{in} * \left(\frac{R_2}{R_1 + R_2}\right) \quad (1)$$

Дільник напруги використовується для вимірювання напруги яка є більшою за 5V, так як контролер не розрахований на більшу напругу. З наявними в мене резисторами, а саме: $R_1 = 4.75\text{кОм}$, $R_2 = 2.35\text{кОм}$, максимально можлива напруга для виміру складає ~14.8V. Джерело напруги яке вимірюється, обов'язково потрібно підключити «мінусовим» контактом до «землі» макету, а іншим контактом до дільника напруги.

Вхідний сигнал потрібно перетворити у цифрове значення. АЦП Arduino має розрядність 10 біт, тому значення може коливатись від 0 до 1023. Зчитування вхідного значення виконується функцією `analogRead()`.

Опорну напруга на АЦП = 1.56В, розрахунок напруги на вході A0:

$$data = \frac{A0 * 1.56}{1024} \quad (2)$$

Де $A0$ – напруга на порті A0, 1024 – розрядність АЦП.

$$IV = \frac{data}{V_{out}} \quad (3)$$

Де IV – вхідна напруга, V_{out} = опір дільника.

Схема пристрою зображена на рисунку Н.

При підключенні вимірюваного джерела напруги, потрібно натиснути кнопку для виміру в потрібний момент. Після цього, програма обраховує отримані данні та виводить їх в програмі Putty через COM порт. Отримані результати зберігаються у файлі на ПК.

Потенціометр потрібен для перевірки правильності роботи приладу, оскільки зовнішнього джерела напруги в мене нема. Він підключений до виходу +5V на платі Arduino.

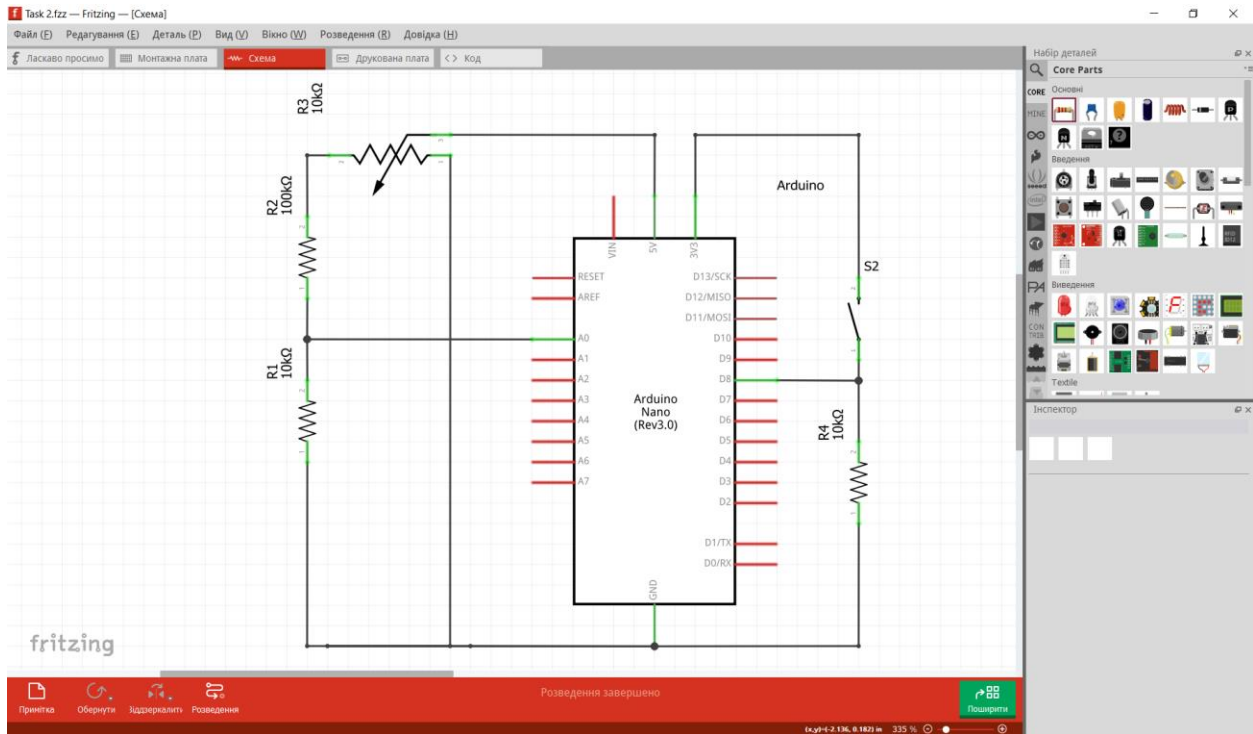


Рис. 6 Електрична схема макету

Код програми:

```
float input_volt = 0.0; // Variable for output
float data = 0.0; // Variable for converting voltage values
float formula = 0.0; // Variable for the voltage divider calculation formula
float r1 = 4650.0; // First resistor
float r2 = 2350.0; // Second resistor
float x = 1.56; // Basic power supply at input A0
const int buttonPin = 8; // Button connect pin
const int delay_button = 20; // Delay to eliminate noise
const int delay_prog = 150; // Wait between processes
const int delay_end = 2000; // Delay in the end of program

void setup() {
  pinMode(buttonPin, INPUT); // Button initialization
  Serial.begin(9600);
}

void loop() {
  int buttonState = digitalRead(buttonPin); // Assign read data from the port
  if (buttonState == HIGH) { // If the button is pressed - perform measurements
    delay(delay_button);
    measure(); // Performance of input voltage measurement
    buttonState = 0; // Resetting the value of the input variable
  }
}
```



```
    delay(delay_prog);
    Serial.println("Please, measure it ");
    Serial.println(".");
} else { // Otherwise, wait for 150 ms
    delay(delay_prog);
}
}

void measure() { // Function of calculation and transfer of input data
    int analog_value = analogRead(A0); // Reading data from port A0
    float formula = r2/(r1+r2); // Calculate the voltage divider
    data = (analog_value * x) / 1024.0; // Voltage conversion
    input_volt = data / formula; // Calculate the input voltage

    Serial.print("Analog Value = ");
    Serial.println(analog_value);
    Serial.print("Voltage = ");
    Serial.println(input_volt);
    delay(delay_end);
}
```