

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Национальный исследовательский университет «МЭИ»

Институт: ИВТИ

Кафедра: ВТ

Направление подготовки:

09.03.01 Информатика и вычислительная техника

**ОТЧЕТ по практике**

**Наименование  
практики:**

Производственная практика: технологическая  
(проектно-технологическая) практика

**СТУДЕНТ**

  
(подпись)

/ Архипов Д. Г. /  
(Фамилия и инициалы)

Группа

А-06-21

(номер учебной группы)

**ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ  
ПО ПРАКТИКЕ**

Отлично

(отлично, хорошо, удовлетворительно, неудовлетворительно,  
зачтено, не зачтено)

  
(подпись)

/ Никуйко С.А. /  
(Фамилия и инициалы члена комиссии)

**Москва  
2024**

## Оглавление

1.	Введение .....	3
2.	Постановка задачи .....	3
3.	Описание объекта задания.....	3
4.	Описание метода решения задачи .....	3
5.	Экспериментальная часть .....	9
6.	Выводы .....	12
7.	Список используемой литературы.....	12
8.	Заключение.....	13

## **1. Введение**

Проверка проводов вручную может занимать продолжительное время, к тому же велики шансы допустить ошибки при долгой и монотонной работе. Чтобы ускорить процесс проверки проводов и уменьшить риски ошибок, следует разработать стенд, выполняющий проверку множества проводов в автоматическом режиме.

## **2. Постановка задачи**

Разработка алгоритма проверки на наличие неправильного соединения проводов в кабеле и разработка пользовательского интерфейса. Кабель может состоять из множества других проводов. В текущей задаче 40 проводов – это максимум, поскольку это соотносится с готовой схематической частью стенда.

## **3. Описание объекта задания**

Необходимо написание кода для микроконтроллера Arduino, который бы выводил поочередно состояние сигналов на каждом из проводов. Также важно создать пользовательский интерфейс, на котором наглядно был бы виден результат проверки.

## **4. Описание метода решения задачи**

Первая часть задания – написание кода для микроконтроллера Arduino. В работе алгоритма можно описать несколько важных составляющих:

1. Сброс счетчиков (осуществляется подачей высокого сигнала RESET).
2. Подача прямоугольного синхросигнала (CLK – по фронту каждого нового импульса происходит переключение счетчиков и подача сигнала на новый провод).
3. Чтение сигналов с определенных пинов (всего их 40 штук по максимально возможному числу проводов) и определение того, с какого провода пришел сигнал.

Уже этого кода на Arduino достаточно, чтобы вместе с запущенной Arduino IDE узнать результат проверки проводов, поскольку вся обработка и так осуществляется на Arduino, а обмен сведениями идет через монитор последовательного порта (COM3 в этом случае).

Код программы:

```
const int pins_input_size = 40;

int time_clk = 10,
    time_reset = 10;

const int pin_clk = A0,
    pin_reset = A1,
    max_number_of_wires = 40;

int cycles = 0;

int pins_input[pins_input_size] = {A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15, 22,
    23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37,
    38, 39, 40, 41, 42, 43, 44, 45, 46, 47};
int values[pins_input_size];
int wire_values[pins_input_size];

void setup() {
    Serial.begin(9600);
    pinMode(pin_clk, OUTPUT);
    pinMode(pin_reset, OUTPUT);
    for (int i = 0; i < pins_input_size; i++)
        pinMode(pins_input[i], INPUT);
    Serial.print("\nНачало работы:\n");
}

void tickCLK(int pin, int time) {
    digitalWrite(pin, HIGH);
    delay(time);
    digitalWrite(pin, LOW);
    delay(time);
}

void reset(int pin, int time) {
    digitalWrite(pin, HIGH);
    delay(time);
    digitalWrite(pin, LOW);
}

void printPinStates() {
    Serial.print("Синхросигнал - ");
    Serial.print(cycles);
    Serial.print("; Значения ПИНов - ");
    for (int i = 0; i < pins_input_size; i++) {
        values[i] = digitalRead(pins_input[i]);
        Serial.print(values[i]);
        Serial.print(";");
        if (values[i] == 1) {
            wire_values[i] = cycles+1;
        }
    }
}
```



Рисунок 1 – вывод значений пинов

```
56 0
57 0
58 0
59 0
60 0
61 0
62 0
63 0
64 0
65 0
66 0
67 0
68 0
69 0
22 0
23 0
24 0
25 0
```

Рисунок 2 – вывод пин-провод (0, если провод не подключен)

Вторая часть задания предусматривает написание интерфейса для более удобного взаимодействия пользователя со стендом. Не должно быть необходимости постоянно обращаться к Arduino IDE с целью узнать проверки проводов.

Чтобы добиться этого, надо было выбрать язык программирования, средства визуализации и разобраться, как работать с последовательным интерфейсом для получения данных с Arduino (на микроконтроллере постоянно работает код, приведенный выше, интерфейс должен принимать и обрабатывать полученные значения).

Программа интерфейса была написана на Python, в качестве графической библиотеки использован TKinter.

Код программы:

```
import serial
import time
from tkinter import *
from tkinter import scrolledtext

def read_data_from_file(file_path):
    data = []
    with open(file_path, 'r') as file:
        for line in file:
            numbers_str = line.split()
            numbers_int = [int(num) for num in numbers_str]
            data.append(numbers_int)
    return data

def read_text_file(file_path):
    try:
```

```

        with open(file_path, 'r', encoding='utf-8') as file:
            content = file.read()
            return content
    except FileNotFoundError:
        print(f"Файл с путем {file_path} не был найден.")
    except IOError:
        print(f"Произошла ошибка во время чтения {file_path}.")

def create_list_of_lists(data_string):
    numbers = [int(num) for num in data_string.split()]
    return [numbers[i:i + 2] for i in range(0, len(numbers), 2)]

def compare_lists_element_by_element(list_file, list_arduino):
    result_message = ""
    errors = 0
    if len(list_file) != len(list_arduino):
        result_message = "Несовпадение наборов данных по длине"
        return result_message
    for sublist_file, sublist_arduino in zip(list_file, list_arduino):
        if sublist_file != sublist_arduino:
            result_message += f"Соединение некорректно - {sublist_file}, {sublist_arduino}\n"
            errors += 1
    if errors == 0:
        result_message += 'Все соединения работают правильно'
    else:
        result_message += f'Число ошибок: {errors}\n'
    return result_message

def clear():
    txt.delete('1.0', END)
    txt_resume.delete('1.0', END)

def run_testing():
    try:
        ser = serial.Serial('COM3', 9600)
        lbl_port.configure(text = "Статус подключения - подключен (COM3)", background='green')
        time.sleep(2)
        cycles = 0
        border = 42
        data_string = ""
        data_from_arduino_output = ""
        while cycles < 83:
            data = ser.readline().decode('utf-8')
            if cycles > border:
                data_from_arduino_output += data
            else:
                data_string += data
            time.sleep(0.1)
            cycles += 1

        ser.close()
        txt.insert(INSERT, data_string)

        data_from_gui = create_list_of_lists(txt_pin_wire.get('1.0', END))
        data_from_arduino = create_list_of_lists(data_from_arduino_output)

        resume_message = compare_lists_element_by_element(data_from_gui, data_from_arduino)
        txt_resume.insert(INSERT, resume_message)

```

```

lbl_port.configure(text = "Статус подключения - проверка \nзавершена (порт отключен)",
background='gray')

except:
    lbl_port.configure(text="Статус подключения - порт недоступен", background='red')

window = Tk()
window.title("Стенд по проверке проводов")
window.geometry('1550x900')
file_path = 'pin-wire-compliance.txt'

lbl_pin_wire = Label(window, text="Соответствия ПИН-ПРОВОД:")
lbl_pin_wire.grid(column=0, row=2)

btn_start = Button(window, text="Подключиться и начать проверку", command=run_testing)
btn_start.grid(column=1, row=0)

btn_clear = Button(window, text="Очистить поля вывода - X", command=clear)
btn_clear.grid(column=2, row=2)

lbl_res = Label(window, text="Результат работы программы >>")
lbl_res.grid(column=1, row=2)

txt_pin_wire = scrolledtext.ScrolledText(window, width=24, height=46)
txt_pin_wire.grid(column=0, row=3)
data_string = read_text_file(file_path)
txt_pin_wire.insert(INSERT, data_string)

txt_resume = scrolledtext.ScrolledText(window, width=43, height=46)
txt_resume.grid(column=1, row=3)

lbl_port = Label(window, text="Статус подключения - не подключен", background='red')
lbl_port.grid(column=0, row=0)

txt = scrolledtext.ScrolledText(window, width=118, height=46)
txt.grid(column=2, row=3)

window.mainloop()

```

Ниже приведены скрины интерфейса.

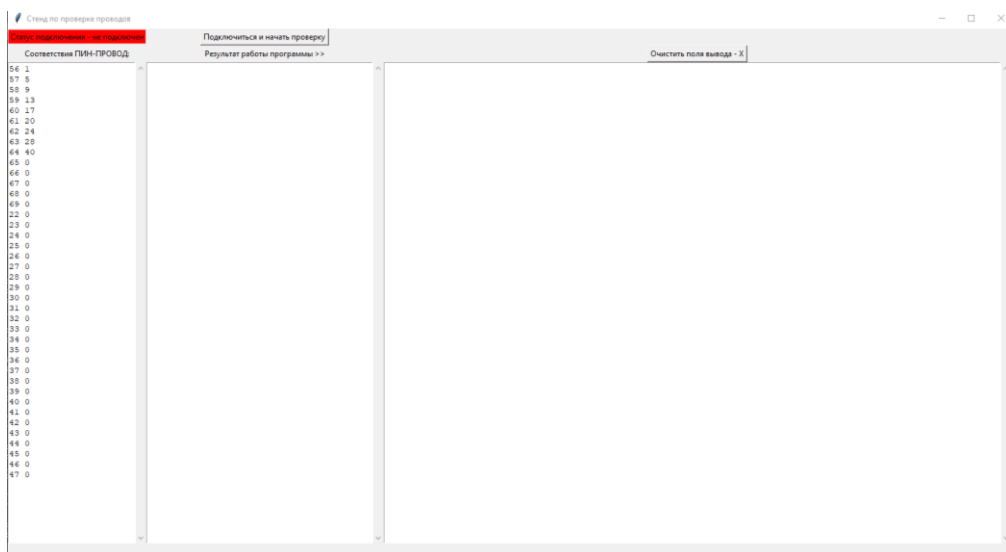




Рисунок 3 – интерфейс

Набор соответствий типа «пин-провод» описывается в текстовом файле, расположенном в одном каталоге с .ру-скриптом. Поле в пользовательском интерфейсе считывает эти значения только при загрузке, потом же их можно видоизменять и запускать программу уже с параметрами поля.

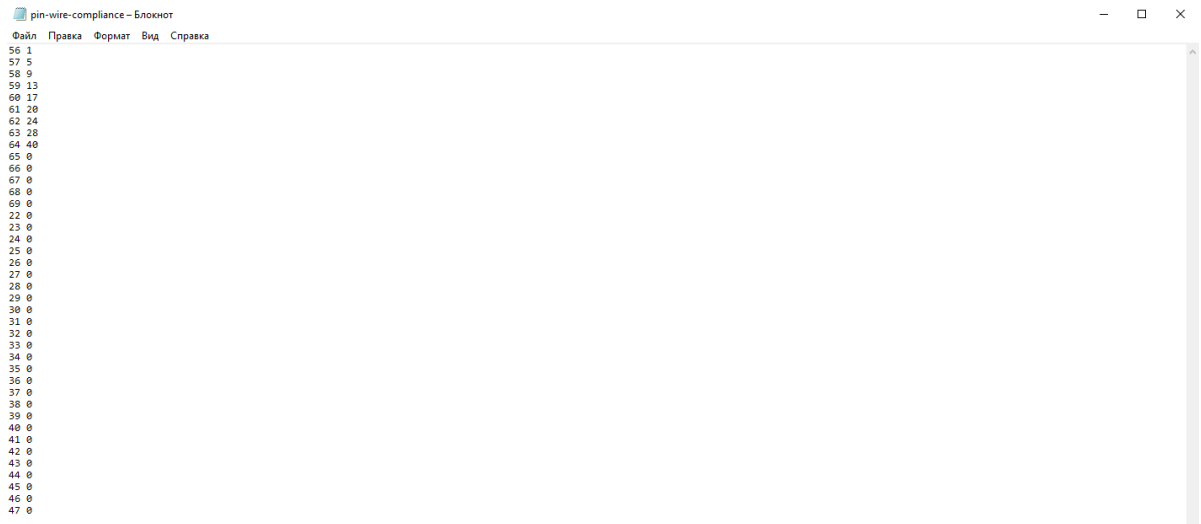


Рисунок 4 – файл с набором соответствий «пин-провод»

В случае, если установить связь по СОМ-порту не удалось, программа укажет, что порт недоступен. Далее нужно будет повторно нажать кнопку подключения и старта программы.

## 5. Экспериментальная часть

Пример работы при отсутствии подключения проводов был приведен выше. Тогда схема выглядела так.



Рисунок 5 – подключена только Arduino

Схема, позволяющая выдавать сигнал поочередно на разные провода выглядит следующим образом. Нижние 4 провода имеют назначения (слева направо): CLK, RESET, 5V, GND.

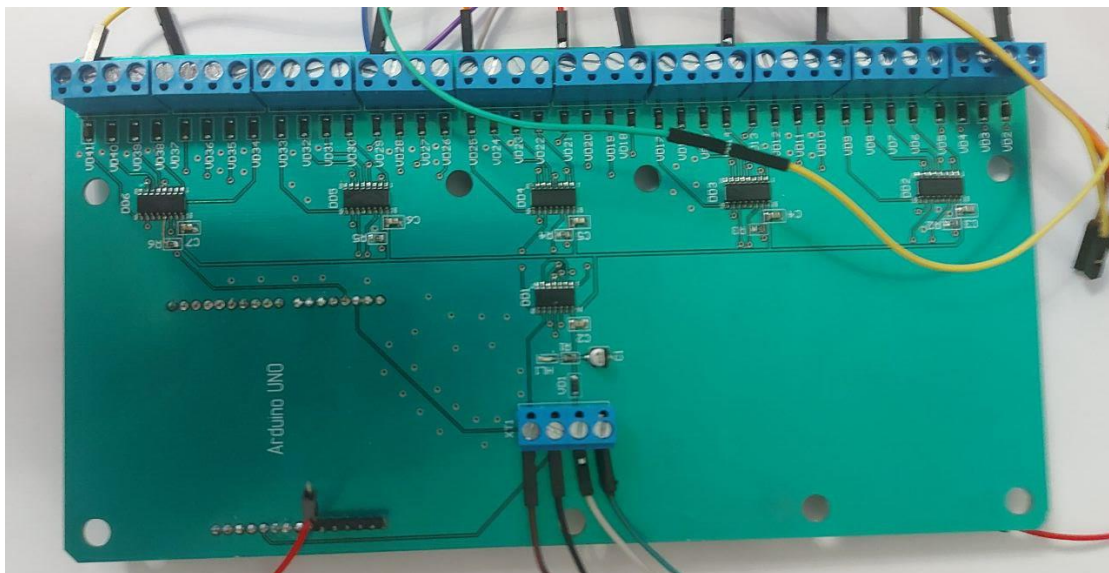


Рисунок 6 – схема для поочередной подачи сигналов

Собранный стенд для проверки проводов с соответствиями «пин-провод». Соответствия указаны на рисунке 4 (только 40й провод не был подключен).

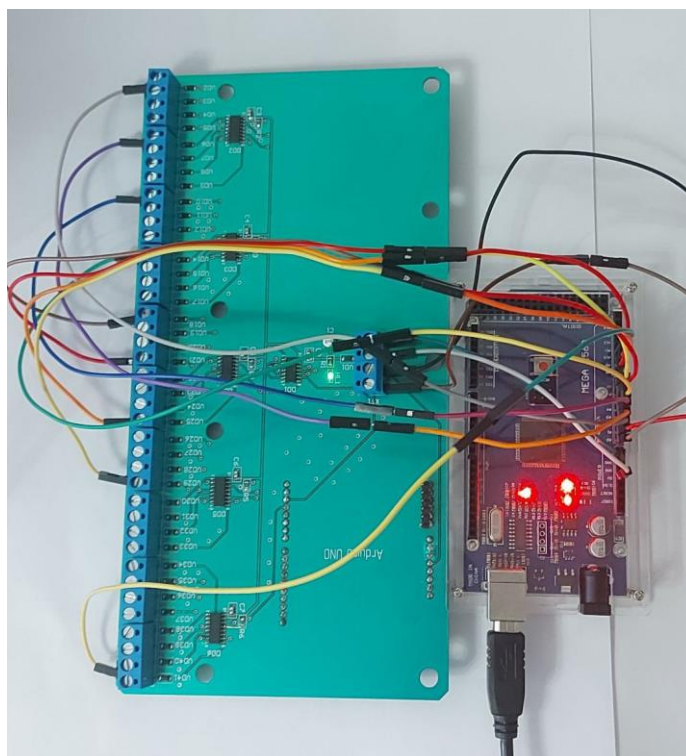


Рисунок 7 – собранный стенд для проверки проводов

Результат работы программы тестирования проводов.

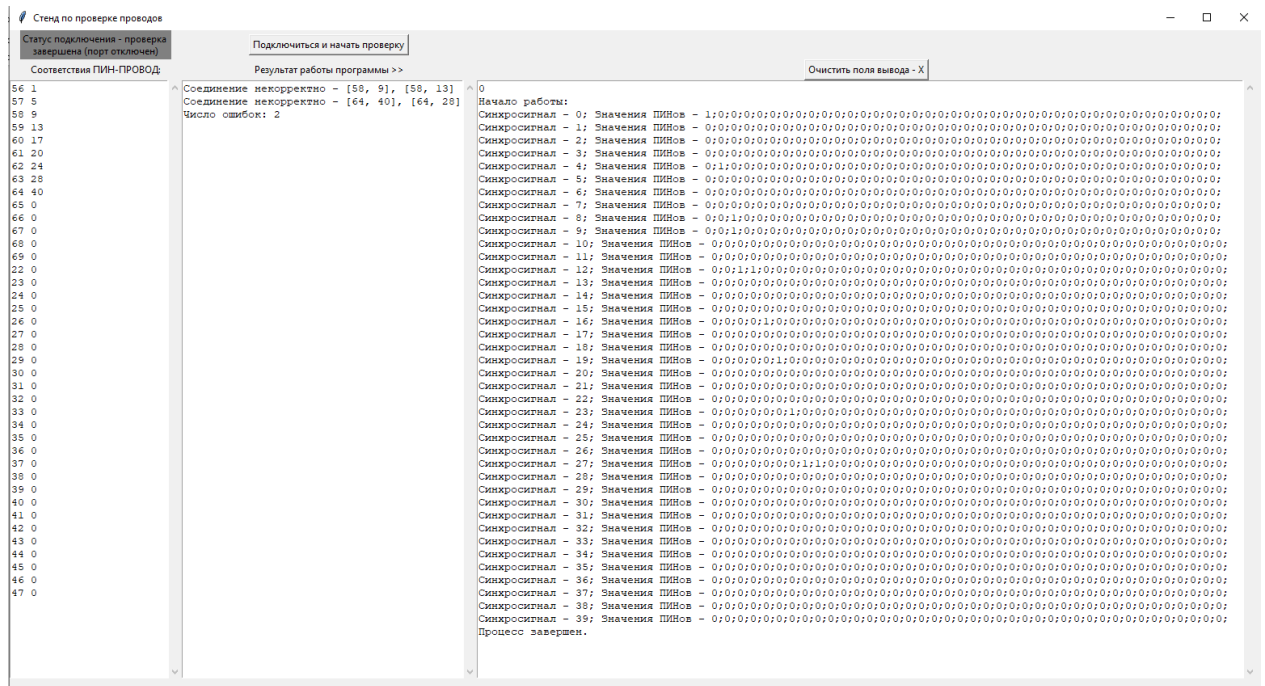


Рисунок 8 – результат проверки схемы (1)

Несмотря на то, что соединения были указаны верно, все равно проявились неточности в работе схемы. Они могли появиться из-за слабых креплений проводов, других помех.

Результаты, полученные в Arduino IDE:

Результат

ПИН: 56 - Провод: 1  
 ПИН: 57 - Провод: 5  
 ПИН: 58 - Провод: 13  
 ПИН: 59 - Провод: 13  
 ПИН: 60 - Провод: 17  
 ПИН: 61 - Провод: 20  
 ПИН: 62 - Провод: 24  
 ПИН: 63 - Провод: 28  
 ПИН: 64 - Провод: 28  
 ПИН: 65 - Провод: разрыв  
 ПИН: 66 - Провод: разрыв  
 ...

Еще один пример работы программы. На этот раз специально указаны 3 ошибочных соответствия «пин-провод».

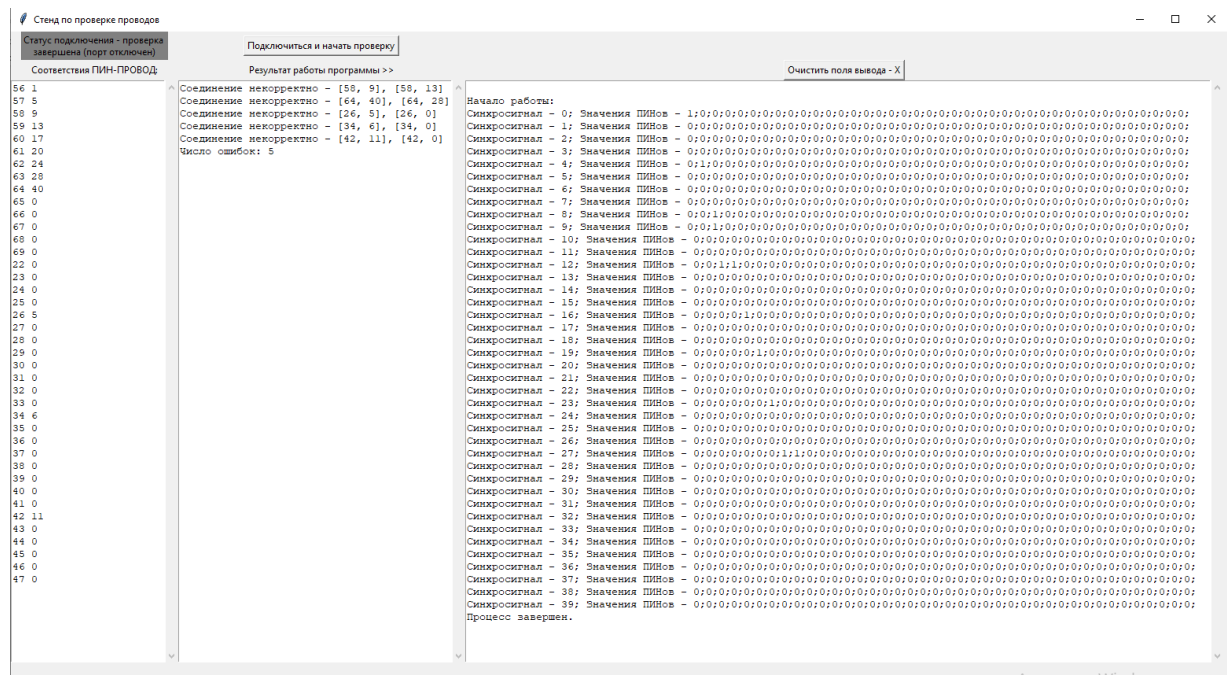


Рисунок 9 – результат проверки схемы (2)

## 6. Выводы

Получилось написать код для рассылки генерируемых Arduino сигналов по проводам для автоматизации процесса проверки распиновки и проверки работоспособности проводов в жгуте.

## 7. Список используемой литературы

1. Обучение Python GUI (уроки по Tkinter) – URL: <https://pythonru.com/uroki/obuchenie-python-gui-uroki-po-tkinter> (дата обращения 03.07.2024)
2. Arduino и Python: Связь через Serial порт – URL: <https://amperkot.ru/blog/serial-py/?ysclid=ly8wgnyiwa97199536> (дата обращения 03.07.2024)
3. Управляем Arduino через Python 3 – URL: <https://arduinoplus.ru/upravlyaem-arduino-cherez-python-3/?ysclid=ly8wg1piol226474728> (дата обращения 03.07.2024)
4. Mega 2560 Rev3 – URL: <https://docs.arduino.cc/hardware/mega-2560/> (дата обращения 03.07.2024)

## **8. Заключение**

В ходе выполнения практики научился писать код для микроконтроллера Arduino, понял как взаимодействовать с Arduino через последовательный порт и выводить нужные значения в пользовательский интерфейс, написанный на Python.