

# Отчет-конспект по созданию лабораторной установки с возможностью удаленного управления

10 апреля 2021 г.

## 1 Идейная часть

Пользователь взаимодействует с интерфейсом сайта. Сервер же принимает и обрабатывает запросы пользователя, формируя запросы для передачи их на лабораторный компьютер (далее LAB PC). LAB PC обрабатывает запросы, управляя устройствами. На приборы наведена камера.

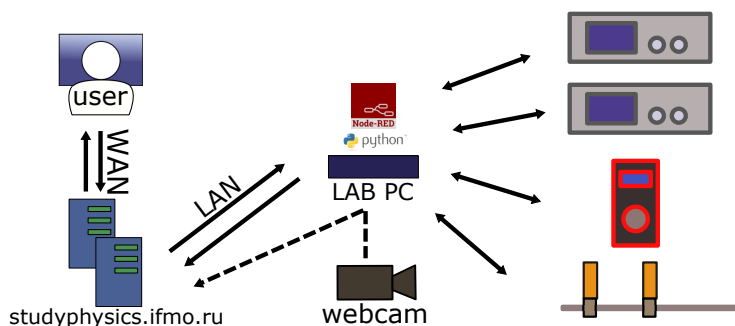


Рис. 1: Схема

На LAB PC установлены Python, скрипты которого декодируют сигналы оборудования, и Node-red, позволяющий запускать эти скрипты, а так же имеющий веб-интерфейс, который позволяет настроить лабораторную работу любой сложности. Простота важна для упрощения настройки будущих лабораторных.

Основное оборудование, используемое в лабораторных: осциллограф GDS 71102B, генератор АКИП-3409/2, а так же мультиметр UT61B, позволяющий заменить осциллограф, если это необходимо. Подключение именно этого оборудования будет описано далее. Установка же может управляться при помощи ардуино: провода могут быть заменены группой реле, подвижные части могут управляться шаговыми двигателями.

## 2 Генератор АКИП-3409/2

Существующий даташит на подобные генераторы описывает способ связи, требующий драйвер [NI-VISA](#). Команды представлены в [статье](#) и в [даташите](#) к одному из подобных приборов. Чтобы использовать python для управления, необходимо установить модуль PyVISA, позволяющую работать с приборами NI-VISA:

```
$ python -3 -m pip install PyVISA
```

Скрипт scripts/generator.py берет на вход два параметра: путь к генератору и команду.

```
import pyvisa
import sys
resurse = sys.argv[1]
cmd = sys.argv[2]
rm = pyvisa.ResourceManager()
inst = rm.open_resource(resurse)
try:
    inst.query(cmd)
except:
    pass
```

Данный скрипт не подразумевает ответ. Скрипт с ответом написан для осциллографа. Чтобы определить путь к генератору, необходимо подключить генератор к компьютеру по USB, открыть терминал python и выполнить несколько команд:

```
$ python -3
> import pyvisa
> pyvisa.ResourceManager().list_resources()
('USB0::0xF4ED::0xEE3A::NDG10GAQ3R0226::INSTR')
```

Соответственно устройство USB0::0xF4ED::0xEE3A::NDG10GAQ3R0226::INSTR и есть наш генератор.

Теперь, чтобы послать команду, например '\*IDN?' на генератор, можно воспользоваться командой:

```
$ python -3 generator.py 'USB0::0xF4ED::0xEE3A::NDG10GAQ3R0226::INSTR'
'*IDN?'
```

**PS:** возможно имеет смысл попробовать установить конкретный драйвер с сайта [www.ni.com](http://www.ni.com) -> support -> Software Downloads -> Драйверы от сторонних разработчиков (IDNet) -> Signal Generator -> найти SDG модели 805, 810. Но этот способ требует регистрации и нуждается в проверке.

#### Порядок подключения:

1. Подключить генератор к компьютеру
2. Скачать и установить драйвер NI-VISA
3. Установить модуль PyVISA
4. Через python терминал узнать путь к устройству

### 3 Осциллограф GDS 71102B

Как и генератор, осциллограф управляется при помощи SCPI команд. Команды, а так же принцип подключения описаны в [даташите](#). Необходимо установить драйвер 'USB driver': его можно скачать с [официального сайта](#).

Как и сказано даташите, связь осуществляется через UART с битрейтом 9600. Чтобы использовать python, необходимо установить модуль PySerial. Скрипт `scripts/oscilloscope.py` принимает два значения: порт, к которому подключен осциллограф и команду. Если команда подразумевает ответ, то возвращается ответ на запрос:

```
import serial
import sys
port = sys.argv[1]
cmd = sys.argv[2]
try:
    ser = serial.Serial(port, baudrate=9600, bytesize=8, parity='N',
stopbits=1, xonxoff=False, dsrtr=False, timeout = 2)
    if '?' in cmd:
        ser.write(str.encode(cmd + '\n'))
        print(ser.readline())
    else:
        ser.write(str.encode(cmd + '\n'))
        ser.close()
except:
    print('error')
```

Чтобы получить изображение с экрана осциллографа применяется уже написанный скрипт [gdstools](#). Данный скрипт требует установку дополнительного модуля Pillow (в описании на github ошибка: необходимо установить не Image, а Pillow, который включает Image). Пример использования указан на github.

#### Порядок подключения:

1. Подключить осциллограф к компьютеру

2. Скачать и установить драйвер
3. Установить модули PySerial, Pillow
4. скачать скрипт gdstools

## 4 Мультиметр UT61B

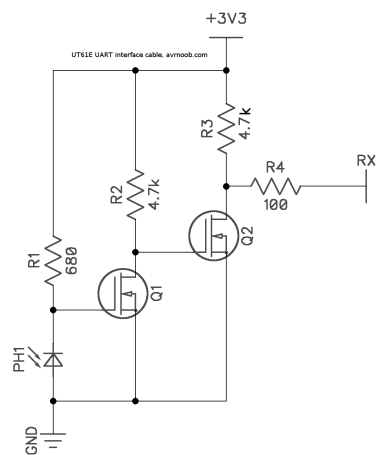
В комплекте с мультиметром идет кабель с rs232 разъемом. Чтобы использовать разъем usb был куплен rs232-usb переходник. Однако, из-за дешевизны последнего, встроенный приемник ИК сигнала с мультиметра не работал: для его работы необходимо 12 вольт питания, тогда как дешевый переходник дает 5. Возможный вариант решения: покупка оригинального ИК-usb кабеля – найти подходящий можно на [сайте](#). Но они дорогие, а в магазинах они практически не встречаются. На [другом сайте](#), часть которого заблокирована в России, из-за чего фото не отображаются, представлена [схема](#), по которой был собран приемник. В качестве USB-TTL конвертера использовался дешевый конвертер на чипе СН340. Резистор 680 Ом был заменен на подстроечный. По неизвестным причинам один из купленных мосфетов был мертв, поэтому вместо второго в каскаде мосфета был вставлен bc547. В наличии были только массивные, мощные резисторы, поэтому плата вышла чуть больше по размерам, чем могла бы быть. При проектировании корпуса была допущена ошибка, из-за которой отверстие для фотодиода и ИК светодиода на мультиметре не совпадают на 2мм. Однако оказалось, что и с таким несовпадением сигнал доходит, поэтому корпус не переделывался. Конвертер не закрыт термоусадкой полностью, чтобы были видны светодиоды: по ним можно подстраивать резистор. Показателем верной настройки является вспыхивание RX светодиода при наведении любого ИК пульта на фотодиод.

Встречалось описание, в котором битрейт мультиметра указан как 19200. Возможно, 19200 указана для другой модели мультиметра серии 61, но наш приемник выдавал значения с битрейтом 2400. Для получения данных необходим драйвер для микросхемы СН340 и модуль PySerial, который уже использовался для работы с осциллографом. Получаемые с мультиметра сырые данные имеют вид "+0914 21..." . Знак, четыре цифры - цифры, отраженные на экране, затем еще две цифры. Первая из последних обозначает данный предел измерения:

1: 2.2 В
2: 22 В
3: 220 В
4: 220 мВ
Другие значения изучены не были, информация есть на <a href="#">гит</a> .

Продолжая пример: мультиметр показывает 9,14 В.

Скрипт `scripts/multimeter.py` для чтения данных с мультиметра парсит значения:



(a) Схема



(b) USB-TTL конвертер



(c) Плата, два отверстия: для фотодиода и доступа к подстроечному резистору



(d) В сборе, без крышки

Рис. 2

```

import serial
import sys
port = sys.argv[1]
unit = {'4': 10**-4, '1': 10**-3, '2': 10**-2, '3': 10**-1}
try:
    ser = serial.Serial(port, baudrate=2400, bytesize=8, parity
='N', stopbits=1, xonxoff=False, dsrdtr=False, timeout = 2)
    msg = ser.read_until(b'\r\n')
    ser.close()
    print(float(msg[0:5]) * unit[msg[6:7].decode()])
except:
    print('error')

```

#### Порядок подключения:

1. Подключить мультиметр к компьютеру
2. Скачать и установить драйвер для CH340
3. Установить модуль PySerial

## 5 Пример установки

\*Ардуино + шаговый двигатель + концевик + две катушки\*

## 6 Node-red

Для начала работы требуется установить Node.js с [официального сайта](#). Затем выполнить команду в командной строке:

```
npm install -g -unsafe-perm node-red
```

Если возникнут ошибки (на windows), необходимо попробовать запустить командную строку от имени администратора и выполнить команду:

```
npm install -global -production windows-build-tools
```

А затем повторить установку node-red. Запустить node-red можно при помощи команды "node-red" **от имени пользователя**. После чего панель управления будет доступна по адресу <http://localhost:1880>. Для работы используется два модуля: node-red-dashboard и node-red-node-serialport, которые можно установить через Меню -> Manage palette -> Install. Затем следует импортировать уже созданный flow через Меню -> Import. После этого необходимо выставить порты USB, соответствующие подключенным устройствам, а так же глобальные переменные: адрес генератор, например.

Логины и пароли, а так же https настраиваются в конфиге. Настройка описана [на официальном сайте node-red](#).

## 7 Масштабируемость

Теоретически данная реализация готова к применению в самых различных лабораторных. Возможные доработки: отдельные ноды для отдельного оборудования, чтобы понизить порог вхождения, отвязка интерфейса node-red, создание репозитория. Необходимо: реализация вебкамеры.