



ОДНИМ ИЗ ВАРИАНТОВ ПРОГРАММИРОВАНИЯ В НАБОРЕ «ЗНАТОК ДЛЯ ARDUINO START» (В ДАЛЬНЕЙШЕМ «ARDUINO START») ЯВЛЯЕТСЯ ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ **PYTHON**.

В данном руководстве представлена инструкция для использования данного набора в сочетании с языком программирования Python.

## СОВМЕСТИМОСТЬ «ARDUINO START» И «PYTHON»

В отличие от использования «классического» подхода с Arduino IDE, где программа пишется на компьютере и потом загружается на плату и выполняется автономно, использование **Python** требует постоянного соединения модуля **111** с компьютером при помощи USB-кабеля. С одной стороны – нельзя будет унести собранную схему далеко от компьютера, но с другой – изменения в программах можно будет проводить очень быстро.

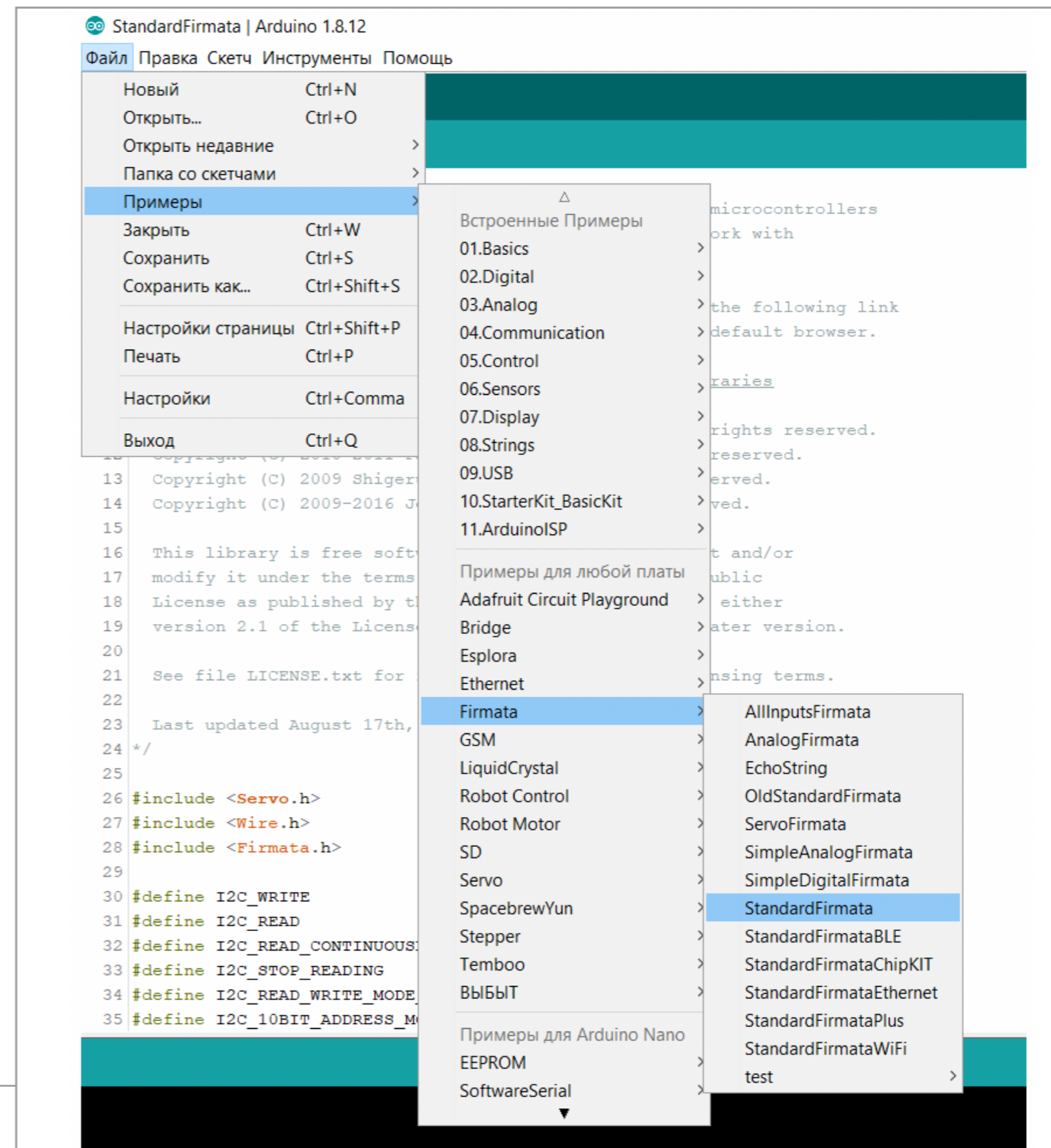
Из-за того, что язык Python изначально не предназначался для использования с микроконтроллерами, **не все проекты из данного набора могут быть запрограммированы** в данной среде, однако, вы всегда можете придумать что-то свое!

# УСТАНОВКА

1. Установите на компьютер Arduino IDE и драйвер для модуля Arduino (см. бумажную инструкцию стр. 20)
2. Скачайте с сайта последнюю версию Python: <https://www.python.org/downloads>
3. Установите при помощи утилиты **pip** библиотеку **pyfirmata** (*pip install pyfirmata*)
4. Установите Snap4Arduino, следуя инструкциям установщика
5. Откройте **Arduino IDE**
6. Откройте проект StandardFirmdata (Файл – Примеры – Firmdata – StandardFirmdata)
7. Произведите настройку для загрузки в модуль **111** (Arduino Nano):

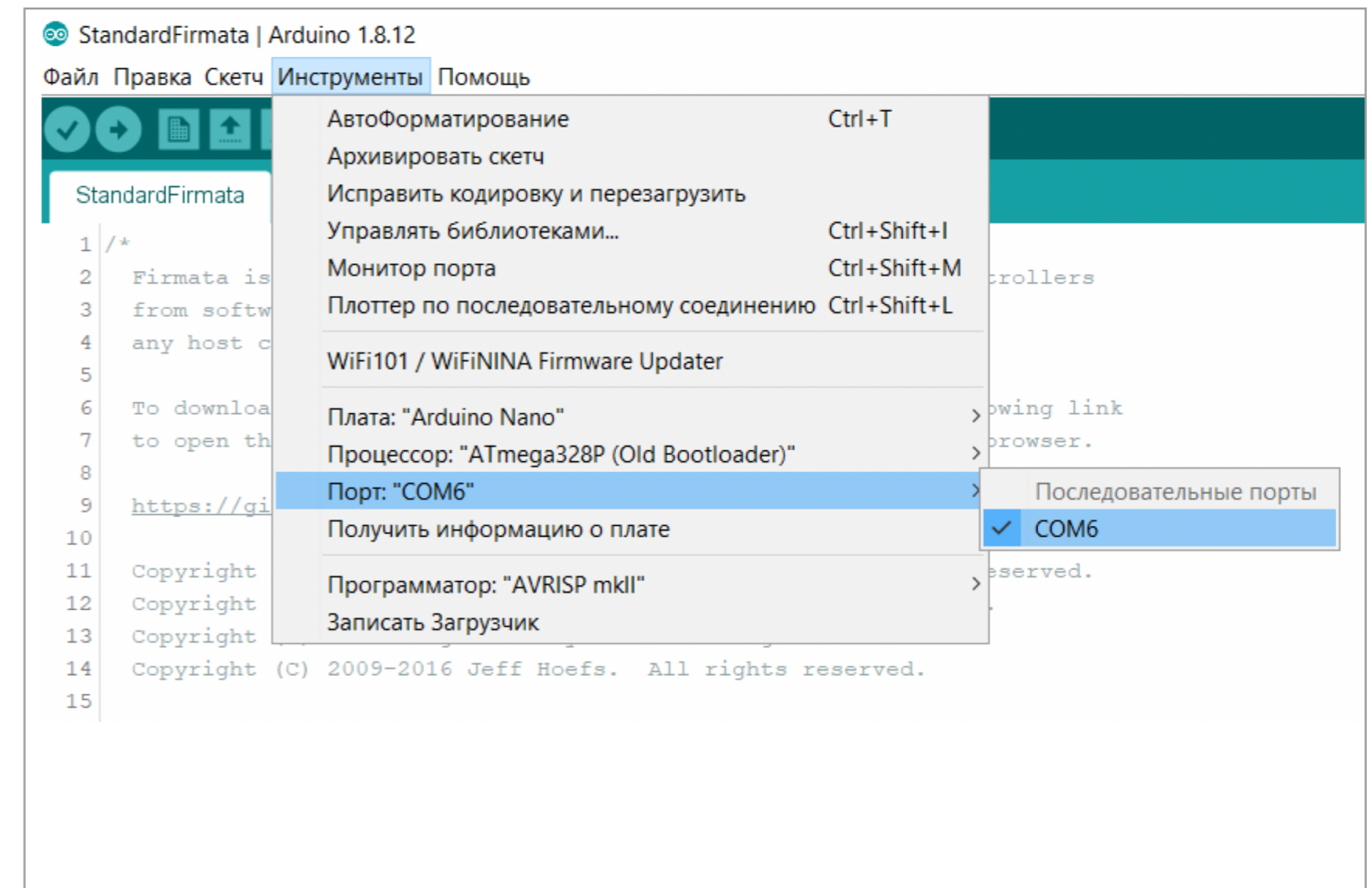
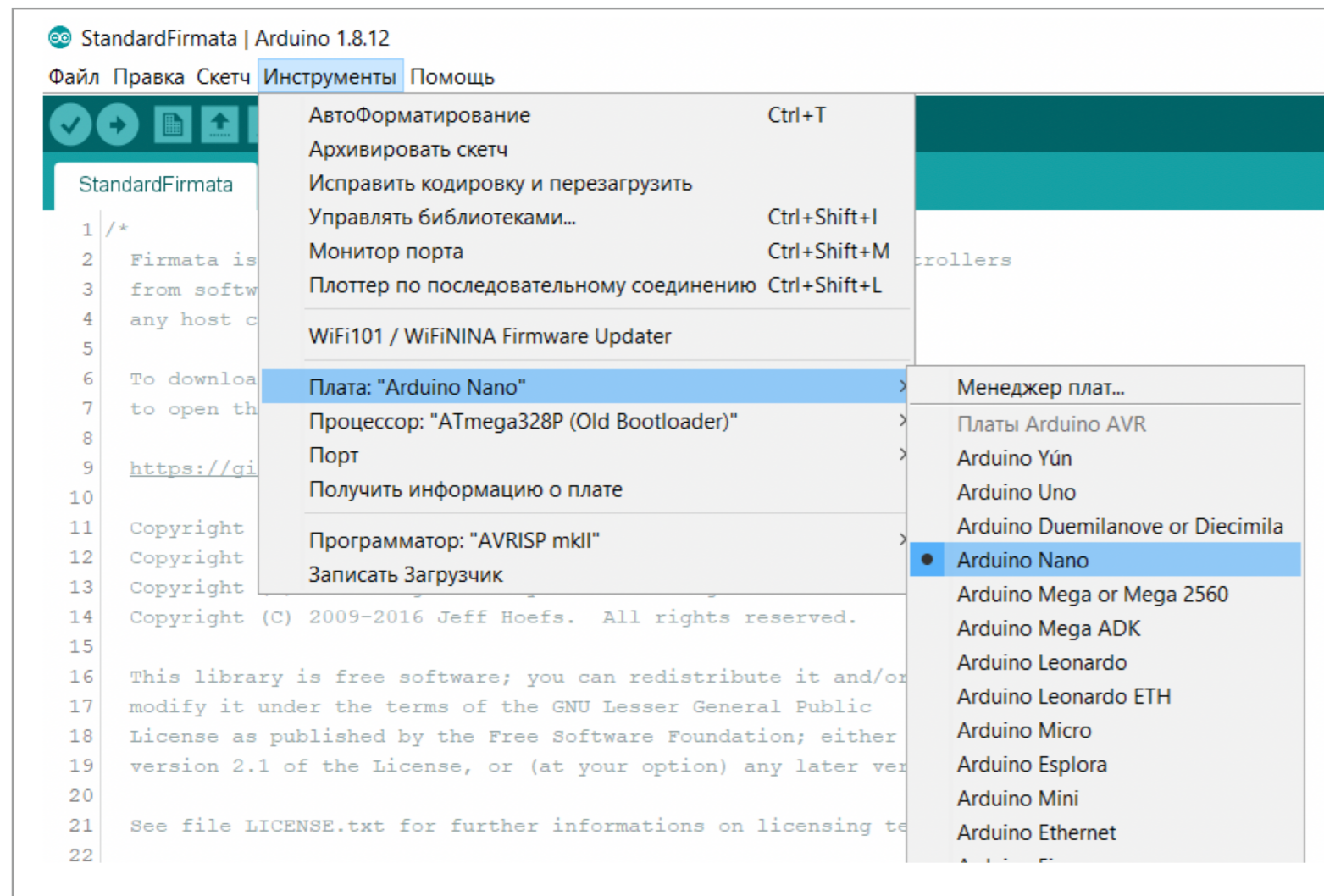
Плата: «Arduino Nano»

Порт: Может быть любой, как правило вида COMX, где X – число (см. инструкцию на стр. 20)




# УСТАНОВКА

8. Загрузите скетч в модуль **111** (Скетч – Загрузка). Данный скетч позволит связать модуль **111** с компьютером и программами на Python





# ОПИСАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ

 Данное руководство позволит управлять проектами с использованием модуля **111** при помощи скриптов на языке Python с использованием библиотеки `pyfirmata` – это один из множества способов связи Arduino и Python.



## ИСПОЛЬЗОВАНИЕ:

Для создания переменной, связанной с пином используется метод `get_pin`:

```
led = board.get_pin('d:5:p')
```

Создает переменную **led**, связанную с цифровым выводом D5, данный пин будет работать с использованием ШИМ (об этом говорит параметр **p**).

### «Параметры» в скобках A:B:C

Первый параметр – может быть либо **d** (digital-цифровой), либо **a** (analog-аналоговый).

Второй – указывает на номер.

Третий – указывает на режим работы, **i** – input(ввод), **o** – output (вывод), **p** – PWM (ШИМ)



## ПРИМЕР РАБОТЫ С ПРОЕКТОМ ИЗ НАБОРА

В архиве с работами ([ссылка Znatok for Arduino. Start. Python projects](#)) в соответствующей папке **Проекты** расположены файлы с проектами из набора Arduino START. Имена соответствуют именам проектов из соответствующего руководства. Рассмотрим подробно проект `01_Control_LED-1`.

Файл `01_Control_LED-1.py`

```
import pyfirmata
import time

board = pyfirmata.Arduino('COM7')

while True:
    board.digital[5].write(1)
    time.sleep(0.1)
    board.digital[5].write(0)
    time.sleep(0.9)
```

# ОПИСАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ



## ПРИМЕР РАБОТЫ С ПРОЕКТОМ ИЗ НАБОРА (ПРОДОЛЖЕНИЕ)

При запуске, данная программа выполняет функционал лабораторной работы 01\_Control\_LED-1, рассмотрим построчно код программы на Python.

```
import pyfirmata
import time
```

Данные строки импортируют 2 библиотеки: `pyfirmata` – для связи Arduino и компьютера и `time` – для работы со временем.

```
board = pyfirmata.Arduino('COM7')
```

Здесь объявляется переменная **board**, которая в дальнейшем будет связана с Arduino, подключенной к порту COM7. Узнать конкретно ваш порт, вы можете при загрузке прошивки (см. пункт «Установка») на модуль **111**.

```
while True:
```

Запускается бесконечный цикл (действия будут повторяться до выключения питания или загрузки новой программы)

```
board.digital[5].write(1)
```

Мы записываем «1» на цифровой вывод 5 нашего модуля **111**. Данная команда соответствует функции `digitalWrite(5, HIGH)`; на языке программирования C++ в среде Arduino IDE – мы включаем светодиод, подключенный к выводу D5 модуля **111**.

```
time.sleep(0.1)
```

Ждем 0.1 секунды (100 мс) ничего не делая. Данная команда соответствует функции `delay(100)` на языке программирования C++ в среде Arduino IDE.

```
board.digital[5].write(0)
time.sleep(0.9)
```

Аналогично предыдущим командам – выключаем светодиод и ждем 900мс.

# ДОСТУПНЫЕ ПРОЕКТЫ ИЗ НАБОРА ARDUINO START

Для программирования на языке Python в данном наборе доступны следующие программы – аналоги из бумажной инструкции:

01\_Control\_LED-1

01\_Control\_LED-7

06\_Lamp\_Control-1

01\_Control\_LED-2

01\_Control\_LED-8

06\_Lamp\_Control-4

01\_Control\_LED-3

03\_RGB\_Mix-1

06\_Lamp\_Control-5

01\_Control\_LED-4

03\_RGB\_Mix-2

06\_Lamp\_Control-6

01\_Control\_LED-5

05\_Motor\_Control-1

06\_Lamp\_Control-7

01\_Control\_LED-6

05\_Motor\_Control-2