



ОДНИМ ИЗ ВАРИАНТОВ ПРОГРАММИРОВАНИЯ В НАБОРЕ «ЗНАТОК ДЛЯ ARDUINO MOBILE» ЯВЛЯЕТСЯ ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON.

Примечание: компания «Знатор» не занимается поддержкой или разработкой указанных систем. Наша цель – предоставить примеры программ управления конструктором для ряда платформ и объяснить как можно с ними работать. Мы постарались учесть различные сценарии использования, но данное руководство может изменяться/дополняться со временем.

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

Данное руководство не является руководством по изучению языка Python!

СОВМЕСТИМОСТЬ «ARDUINO MOBILE» 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 почти «на лету»!

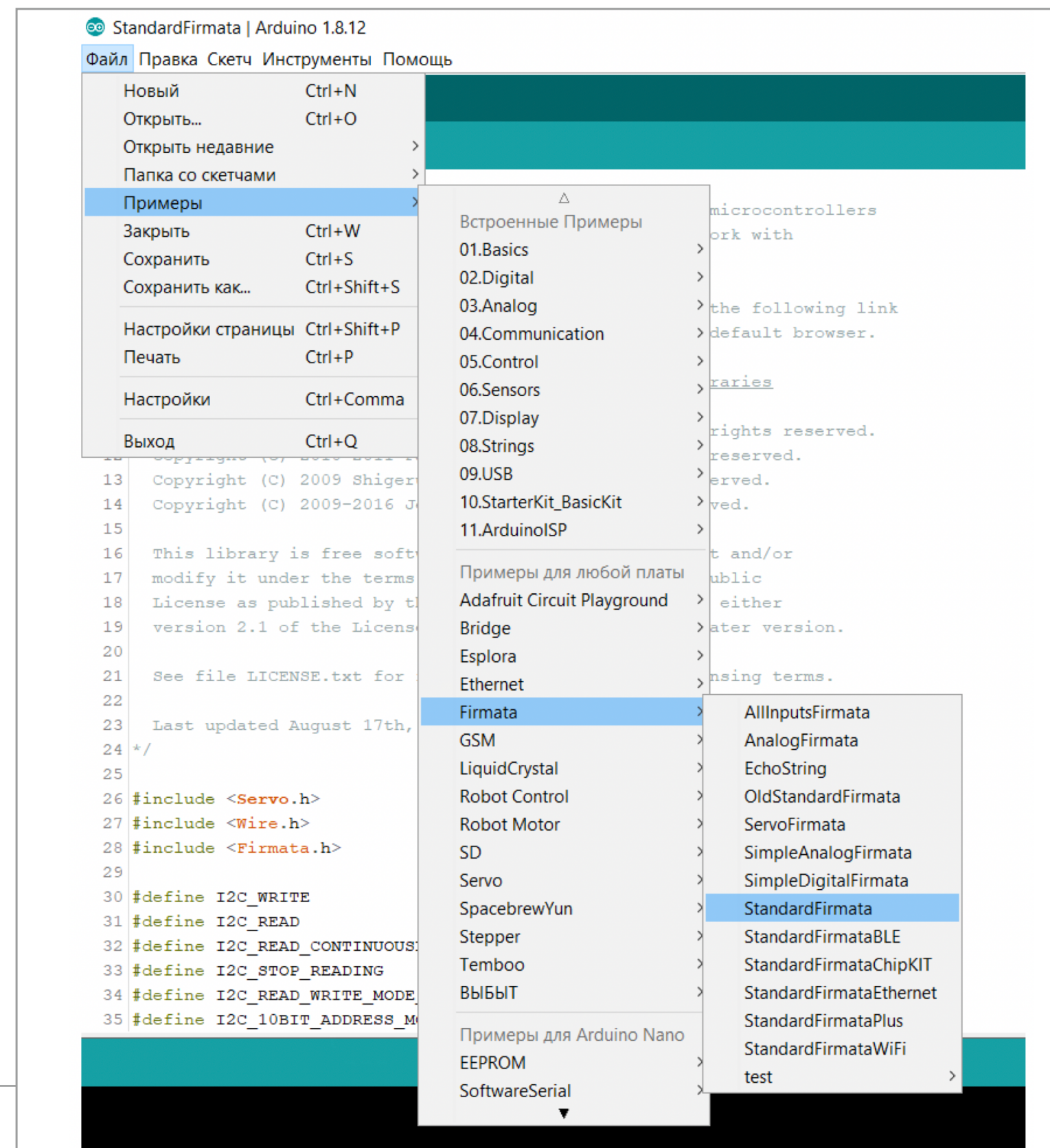
Из-за того, что язык Python изначально не предназначался для использования с микроконтроллерами, **не все проекты из данного набора могут быть запрограммированы** в данной среде, однако, вы всегда можете придумать что-то свое!

УСТАНОВКА

1. Установите на компьютер Arduino IDE и драйвер для модуля Arduino (см. бумажную инструкцию стр. 24)
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):

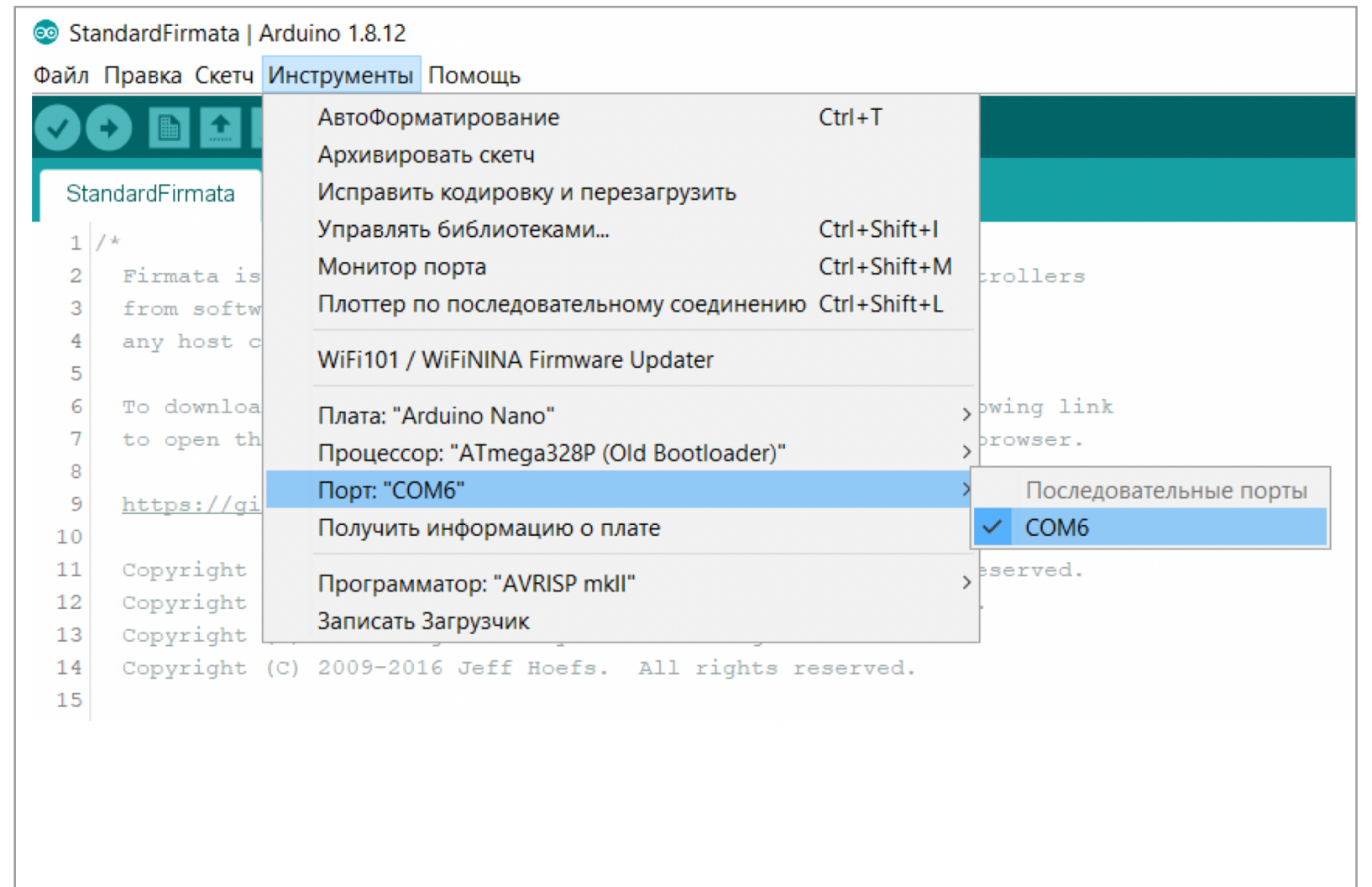
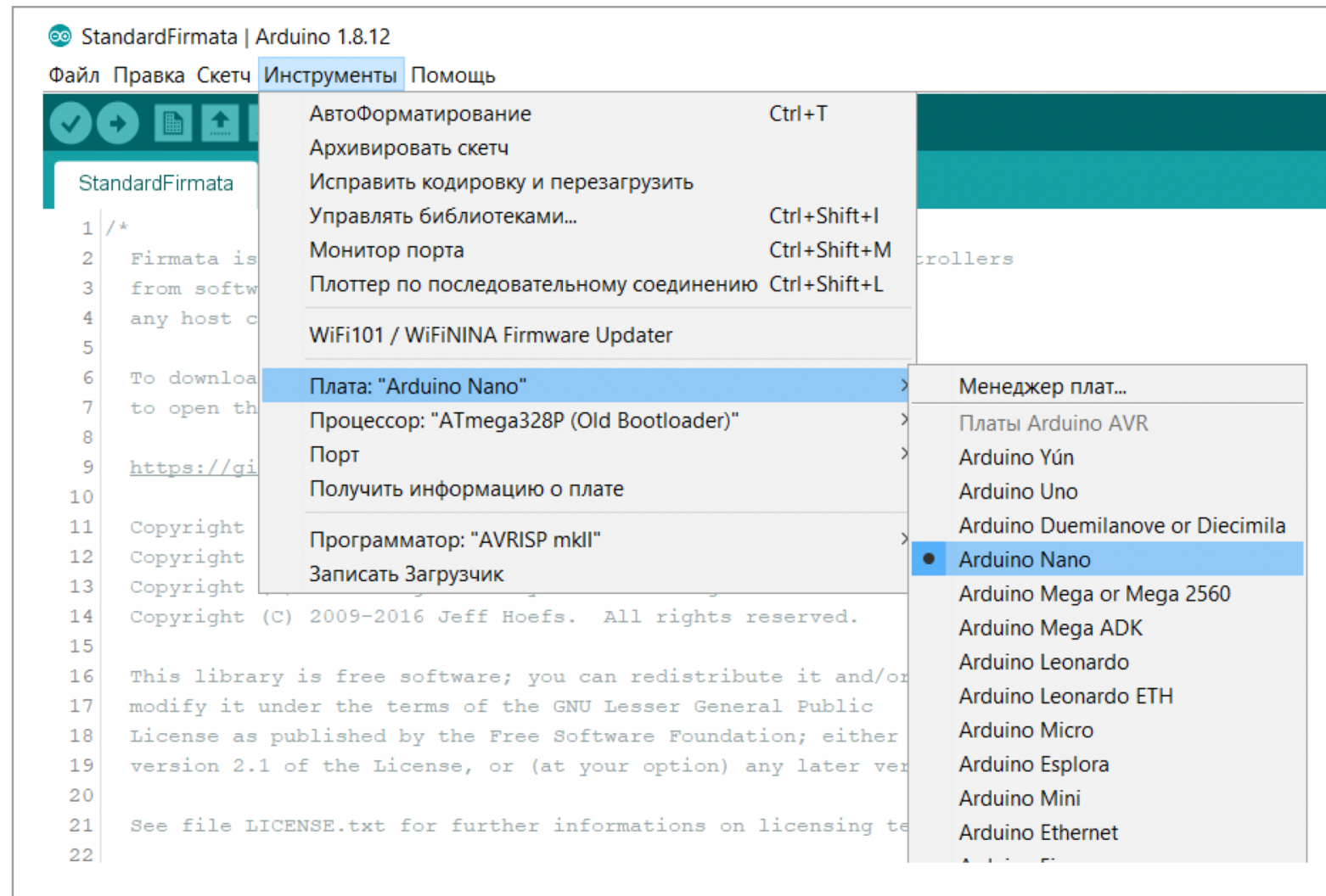
Плата: «Arduino Nano»

Порт: Может быть любой, как правило вида COMX, где X – число (см. инструкцию на стр. 24)



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 (<http://znatok.ru/link/?mobile-python-projects>) в соответствующей папке **Проекты** расположены файлы с проектами из набора Znatok for Arduino.MOBILE. Имена соответствуют именам проектов из соответствующего руководства. Рассмотрим подробно проект 01_LightAndSound-03 и как его можно создать/запустить. (Обратите внимание, в примере разбирается работа с одним светодиодом, скрипт полностью вы можете найти в папке с проектами).

Файл **01_LightAndSound-03-example.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_LightAndSound-03, 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 MOBILE

В наборе доступны следующие проекты:

Для программирования на языке Python в данном наборе доступны следующие программы – аналоги из инструкции (в дальнейшем возможно расширение данного списка):

01_LightAndSound_01	01_LightAndSound_11 (частично)	01_LightAndSound_25
01_LightAndSound_02	01_LightAndSound_12 (частично)	01_LightAndSound_40
01_LightAndSound_03	01_LightAndSound_13 (частично)	01_LightAndSound_41
01_LightAndSound_04	01_LightAndSound_14 (частично)	01_LightAndSound_42
01_LightAndSound_05	01_LightAndSound_15 (частично)	02_Motors_05 (частично)
01_LightAndSound_07 (частично)	01_LightAndSound_16 (частично)	02_Motors_06 (частично)
01_LightAndSound_08 (частично)	01_LightAndSound_22	02_Motors_07 (частично)
01_LightAndSound_09 (частично)	01_LightAndSound_23	
01_LightAndSound_10 (частично)	01_LightAndSound_24	