

# *ShloTiny5*

*Описание программного обеспечения*

2023

## Оглавление

1 Введение.....	5
2 Основные термины и определения.....	6
3 Интерфейс пользователя ShloTiny5.....	7
3.1 ShloTiny Info – страница информации.....	8
3.2 Configure – страница конфигурации.....	9
4 Редактор ShloTiny EDraw.....	11
4.1 Палитра узлов.....	12
4.2 Узлы.....	13
4.2.1 Узлы с меню выбора функции.....	14
4.2.2 Узлы с полем параметров узла.....	14
4.2.3 Узлы с привязкой к ножкам микроконтроллера – GPIO.....	14
4.2.4 Узлы с адресом устройства на шине.....	15
4.2.5 Особенности ввода и просмотра чисел в редакторе EDraw5 в различных системах счисления.....	15
4.3 Связи.....	17
4.3.1 Как проводить связи между узлами.....	17
4.3.2 Меню свойств связи.....	18
4.3.3 Изменение цвета связи или группы связей.....	18
4.3.4 Автоматическое перестроение связей.....	19
4.4 Кнопки управления, горячие клавиши.....	20
4.5 Сохранение схемы в файл.....	22
4.6 Загрузка схемы из файла.....	22
4.7 Загрузка схемы в устройство ShloTiny5 на выполнение.....	22
4.7.1 Ошибка связи с устройством.....	23
4.7.2 Ошибки конфигурации схемы-программы.....	24
4.7.2.1 Требование привязки входов.....	25
4.7.2.2 Требование настройки функций.....	25
4.7.2.3 Требование привязки GPIO.....	26
4.8 Считывание схемы из устройства ShloTiny5.....	26
4.9 Удаление схемы из устройства.....	28
4.10 Режим «Монитор».....	28
4.11 Выбор и перемещение элементов схемы.....	30
4.11.1 Выбор одного элемента схемы.....	30
4.11.2 Выбор нескольких элементов схемы.....	30
4.11.3 Выбор всех элементов схемы.....	30
4.11.4 Снятие выбора элементов схемы.....	30
4.11.5 Перемещение элементов схемы.....	30
5 Алгоритмы, Узлы, События.....	32
6 Описание узлов.....	34
6.1 Comment – комментарии и общесистемные параметры.....	34
6.1.1 Comment – комментарий.....	34
6.1.2 Device Settings – общие параметры.....	34
6.2 Math functions – математические функции.....	35
6.2.1 Const – числовая константа.....	35
6.2.2 eConst – предопределённые специальные константы.....	35
6.2.3 Int/Check – проверки и округления.....	35
6.2.3.1 Inf?(a).....	36

6.2.3.2 NaN?(a).....	36
6.2.3.3 Ceil(a).....	36
6.2.3.4 Round(a).....	36
6.2.3.5 Abs(a).....	36
6.2.4 A op B, A op Bc – математические операции над двумя операндами.....	36
6.2.5 A cmp B, A cmp Bc – сравнение двух чисел.....	37
6.2.6 kx+b – линейная функция.....	38
6.2.7 X0X1 → Y0Y1 – линейное преобразование.....	38
6.3 Basic logic – логические функции и сдвиги.....	38
6.3.1 Lops – базовые логические операции.....	38
6.3.2 ~1 – операция НЕ.....	39
6.3.3 RS – RS-триггер.....	39
6.3.4 RG – регистр хранения данных.....	40
6.3.5 Mux – мультиплексор.....	40
6.3.6 BitS – преобразование двоичного знака и инверсии.....	41
6.3.7 BitOps и BitOpc – узлы побитовых операций и сдвигов.....	41
6.3.8 Bits Split – разбиение числа на биты.....	42
6.3.9 Bits Join – объединение отдельных бит в число.....	43
6.4 One-bit IO – однобитный ввод-вывод.....	43
6.4.1 DInput – узел бинарного входа.....	43
6.4.2 DOutput – узел бинарного выхода.....	44
6.4.3 PWM – широтно-импульсный модулятор (ШИМ).....	44
6.4.4 PWM freq – задать частоту ШИМ.....	45
6.4.5 Encoder – энкодер (волкодер), датчик угла поворота.....	45
6.5 Analog-to-Digital – аналого-цифровой преобразователь.....	47
6.6 Data Transfer – обмен данными по сети.....	48
6.6.1 Mqtt Config – настройка соединения с MQTT-брокером.....	48
6.6.2 MqttPub – публикация данных на MQTT-брокере.....	49
6.6.3 MQTTdesc – получение данных от MQTT-брокера.....	50
6.7 Delay, timers – таймеры, задержки, счётчики, сумматоры.....	51
6.7.1 Timer и Timer(t) – таймеры с заданным периодом.....	51
6.7.2 TimerFunc и TimerFunc(t) – функции задержек, импульсов.....	52
6.7.3 Counter – счётчик с ограничением и сбросом.....	53
6.8 Barriers and filters – барьеры и фильтры событий.....	53
6.8.1 B(t) и B(tin) – временный барьер прохождения событий.....	53
6.8.2 EmitEv – формирование события.....	54
6.8.3 BD(cond) и BE(cond) – узлы условного прохождения событий.....	54
6.8.4 BE(cs) – узел запрета прохождения специальных событий.....	55
6.9 Net Services – состояние сети.....	55
6.10 Date/Time – работа с датой и временем.....	55
6.10.1 Get Time – получить системное время.....	56
6.10.2 Set Time – установить системное время.....	57
6.10.3 NTP Time – получить время с NTP-сервера.....	57
6.10.4 Split Time – извлечь из временной метки дату и время.....	57
6.10.5 Synth Time – составить временную метку из даты и времени.....	58
6.11 1-wire-bus chip – устройства с однопроводным интерфейсом.....	58
6.11.1 DHT – однопроводные датчики температуры и влажности.....	58
6.11.2 DS182x – датчик температуры.....	58
6.12 I2C bus chip – устройства на шине I2C.....	60

## ShloTiny5 – сделай сложности проще

6.12.1 I2C bus – конфигуратор шины I2C.....	60
6.12.2 BMP180, BMP280 – датчики температуры и давления окружающей среды.....	60
6.12.3 BME280 – датчики влажности, температуры и давления окружающей среды.....	61
6.12.4 BH1750 – датчик освещённости.....	61
6.12.5 DS1307 – часы реального времени.....	61
6.12.6 SCD30 – датчик содержания CO <sub>2</sub> , температуры и влажности.....	62
6.13 7-seg LEDS – управление семисегментными индикаторами, шкалы.....	63
6.13.1 TM1637 – контроллер 6-разрядного семисегментного индикатора.....	63
6.13.2 Conv7Seg – преобразователь чисел в коды 7-сегментного индикатора.....	64
6.13.3 Scale – шкала на базе 7-сегментного индикатора.....	66
6.13.3.1 Пример 1. Шкала из 13 светодиодов без сдвига.....	67
6.13.3.2 Пример 2. Шкала из 12 светодиодов со сдвигом на 5.....	68
6.13.3.3 Пример 3. Объединение шкал.....	69
7 Настройка ShloTiny5.....	71
7.1 Режимы работы ShloTiny5.....	71
7.1.1 Режим «Config mode».....	71
Как перейти в «Config mode»?.....	72
7.1.2 Режим «Station mode».....	72
7.1.3 Режим «AP mode».....	73
7.1.4 Режим «AP+Station mode».....	74
7.1.5 Режим «Single mode».....	74
7.2 Блокировка WEB-сервера.....	74
7.3 Блокировка TxD и вывода информационных сообщений в консоль.....	74
7.4 Безопасный режим «Safe Mode».....	75
Пример настройки безопасного режима.....	77
8 Установка программного обеспечения ShloTiny5 на ESP-модуль (прошивка).....	79
8.1 Где взять прошивку ShloTiny5?.....	79
8.2 Как подключить ESP-модуль к ПК?.....	79
8.3 Как соединить Shlotiny5 с терминалом?.....	81
8.4 Как прошивать Shlotiny5?.....	81
9 OTA – обновление программного обеспечения по сети.....	84

### 1 Введение

Программное обеспечение **ShloTiny5** предназначено для построения устройств и систем на базе микроконтроллеров, например модулей ESP.

Особенностью **ShloTiny5** является возможность использования стандартного web-браузера (Chrome, Firefox и т.п.) в качестве средства визуального программирования модулей на базе чипа **ESP8266** (модули ESP).

Подключение персонального компьютера (ПК, ноутбука) с установленным на нём web-браузером к модулю ESP осуществляется посредством сети WiFi.

Для установки ПО **ShloTiny5** на модуль ESP необходима однократная загрузка образа ПО в модуль ESP посредством интерфейса UART.

Более подробно об установке ПО и настройке модуля можно прочитать в разделе Установка программного обеспечения **ShloTiny5** на ESP-модуль (прошивка).

Онлайн-версия редактора схем **ShloTiny5** находится по [ссылке](#). Её можно пользоваться, не имея в руках готового устройства.

Образ ПО **ShloTiny5** можно получить по [ссылке](#) на раздел документации **ShloTiny5**. Прямой ссылки на образ нет, так как версия образа будет меняться.

#### Краткий путеводитель по разделам настоящего документа.

- Основные термины и определения – основные понятия и их расшифровка.
- Редактор **ShloTiny EIDraw** – самый главный раздел, посвящённый редактированию редактору схем и работе с ним.
- Алгоритмы, Узлы, События – принципы работы схемы-программы.
- Описание узлов – описание функций узлов (справочник).
- Настройка **ShloTiny5** – описание режимов работы устройства под управлением **ShloTiny5** на базе модуля ESP8266.
- Установка программного обеспечения **ShloTiny5** на ESP-модуль (прошивка) – первоначальная установка ПО **ShloTiny5** на ESP-модуль на базе ESP8266.
- OTA – обновление программного обеспечения по сети – обновление программного обеспечения ESP-модуля по сети (доступно на модулях с объёмом FLASH не менее 2Мбайт, начиная с версии ПО **v0.29**).

## 2 Основные термины и определения

Прежде, чем о чём-то говорить, необходимо определиться с терминами. Поэтому в начале – кратенький словарик терминов.

Редактор схем, редактор, EIDraw	– редактор визуальных схем-программ, работающий в браузере, входящий в состав ПО ShloTiny5
Схема-программа, схема, программа	– визуальное представление алгоритма работы модуля ESP под управлением ПО ShloTiny5 в редакторе схем. Схема состоит из узлов и связей между ними.
Узел	– визуальное представление на схеме устройства или действия.
Событие	– инициация процесса обработки данных узлом.
Связь, связь между узлами	– визуальное представление на схеме пути прохождения событий, т.е. пути передачи данных от одного узла к другому.
Палитра, палитра узлов	– список доступных узлов, находящийся в левой части окна редактора.
Рабочее поле редактора	– область рисования схемы-программы (см. рисунок 1 ниже).
Модуль ESP, ESP-модуль	– модуль на базе чипа ES8266 (NodeMCU, ShloTiny и т.п.).
Устройство ShloTiny5	Устройство на базе микроконтроллера (например модуль ESP), управляемое программным обеспечением ShloTiny5.
OTA	– технология обновления ПО ShloTiny5 по беспроводной сети

### 3 Интерфейс пользователя ShloTiny5

Интерфейс пользователя ShloTiny5 реализован как WEB-интерфейс и работает через любой современный интернет-браузер (Chrome, Mozilla и проч.).

Для доступа к интерфейсу пользователя устройства ShloTiny5 необходимо запустить браузер и в адресной строке набрать `http://<IP-адрес устройства ShloTiny>`.

Если устройство находится в режиме конфигурации, то IP-адрес фиксирован.

В режиме конфигурации IP-адрес устройства ShloTiny5 всегда фиксирован и для доступа к интерфейсу пользователя необходимо в адресной строке браузера набрать:

`http://192.168.4.1`

Если устройство подключено как WiFi станция к какой-либо точке доступа WiFi, то можно посмотреть в настройках точки доступа.

Также можно сконфигурировать устройство ShloTiny5 в режиме «AP+Station mode» (см. пп. 7.1.4 ), подключиться к нему как точке доступа и посмотреть на странице информации адрес, который устройство ShloTiny5 получило от точки доступа.

Кроме этого, возможна публикация информации о параметрах устройства ShloTiny5 на MQTT-брокере (см. пп. 6.6.2 ). В числе прочей информации, там публикуется также и IP-адрес устройства ShloTiny5.

Будем считать, что тем или иным способом, IP-адрес устройства стал известен.

После того, как в адресной строке браузера набрали `http://<IP-адрес устройства ShloTiny>`, в окне браузера появляется главная страница устройства.

Интерфейс пользователя состоит из трёх страниц-вкладок:

- **ShloTiny Info** – страница информации, она же главная страница устройства ShloTiny5.
- **Configure** – страница конфигурации. На этой странице настраиваются параметры доступа в сеть, указываются параметры режима работы устройства и параметры безопасного режима.

## ShloTiny5 – сделай сложности проще

– **ShloTiny EIDraw** – редактор визуальных программ, в котором и происходит основная работа по программированию устройства ShloTiny5. Редактору ShloTiny EIDraw посвящён раздел 4.

### 3.1 ShloTiny Info – страница информации

На этой странице находится общая информация о текущих параметрах функционирования устройства ShloTiny5. Пример страницы информации ShloTiny5 показан на рисунке ниже.

#### ShloTiny Info

Info

EIDraw

Name: esp\_8266\_155da4  
Ver: 28.10.2023 19:04:19 v0.25 +0 adec9c5 clean  
OTA: OTA enabled. Boot from rom1. Flash Info: mode:QIO, speed:26MHz, size:4MB (4194304 bytes).  
  
MQTT: mqtt.cloud.ru:1880  
WiFi: STA ip: 192.168.1.133 AP ip: 192.168.10.12 WPA/WPA2+PSK  
  
Now: Sun Nov 26 2023 00:12:52 GMT+0700 (Красноярск, стандартное время)  

Configure

Used GPIOs

GPIO 13: (DHT22): 6C 98%

GPIO 17: (DS1820): 6C

GPIO 2: (Input): 1

GPIO 4: (Input): 1

GPIO 5: (Input): 1

GPIO 16: (Relay): 0

GPIO ADC: 0.4735696228010826

GPIO I2C\_0: (BMP280): 0.291360039904055

Страница разбита на два раздела.

Раздел «Info» – общая информация о параметрах подключения к сети и наличии соединений и прочем.



## ShloTiny5 – сделай сложности проще

Раздел «Used GPIO» показывает информацию об использовании и состоянии линий ввода-вывода.

Кнопка «Configure» предназначена для перехода на страницу конфигурации.

Кнопка «EIDraw» предназначена для перехода в редактор визуальных программ EIDraw.

### 3.2 *Configure* – страница конфигурации

Страница конфигурации предназначена для настройки параметров доступа в сеть, режимов работы устройства и параметров безопасного режима.

Пример страницы конфигурации приведён на рисунке ниже.

Configure

ShloTiny Info

EIDraw

Network mode and access

Name (SSID)  
IOTap

Password  
\*\*\*\*\*

AP Name (SSID)  
esp\_8266\_dcf6f2

AP Password  
\*\*\*\*\*

AP IP address  
192.168.4.1

ShloTiny mode: Station mode ▼

Lock Web in Station mode ☐

Lock TxD (GPIO1) pin ☒

Scan WiFi

Safe Mode

Select safe mode: Switch to Config mode ▼

Time to switch to safe mode: 1 min ▼

Time to stay in safe mode: 5 min ▼

Save Cancel Reset to default

Страница разбита на два раздела.

## ShloTiny5 – сделай сложности проще

Раздел «Network mode and access» определяет режим работы, подключение к точке доступа WiFi и конфигурацию встроенной точки доступа. Подробное описание параметров данного раздела приведено в пп. 7.1 – 7.2 .

Раздел «Safe Mode» определяет параметры безопасного режима работы. Подробное описание безопасного режима работы и его настройки приведено в пп. 7.4 .

Кнопка «ShloTiny Info» предназначена для перехода на страницу информации.

Кнопка «ElDraw» предназначена для перехода в редактор визуальных программ ElDraw.

Если на вашем модуле достаточно памяти, то на странице конфигурации автоматически появляется кнопка «Upgrade», предназначенная для обновления ПО прямо через WEB-страницу, описанное в разделе OTA – обновление программного обеспечения по сети.

## 4 Редактор ShloTiny EIDraw

Редактор ShloTiny EIDraw (далее – просто EIDraw), входящий в состав ПО ShloTiny5, предназначен для ввода схем-программ в память модуля ESP.

При работе с устройством редактор загружается в браузер ПК посредством сети WiFi. Поэтому никаких иных средств программирования, кроме браузера, не нужно.

Более подробно об установке ПО и настройке модуля ESP можно прочитать в инструкции к предыдущей версии ShloTiny, находящейся по ссылке: [Caŭm ShloTiny](#) и [Ренозитаріў ShloTiny \(старая версия\)](#).

Образ ПО ShloTiny5 можно получить по [ссылке](#) на раздел документации ShloTiny5. Прямой ссылки на образ нет, так как версия образа будет меняться.

Онлайн-версия редактора схем ShloTiny5 находится по [ссылке](#). Её можно пользоваться, не имея в руках готового устройства.

Итак, считаем, что ПО ShloTiny5 установлено на модуль ESP и связь с ним настроена. Заходим на страницу модуля ESP браузером и кликаем на вкладку EIDraw.

Или, если модуля ESP нет под рукой, то можно просто перейти по [ссылке](#).

И мы попадаем в редактор EIDraw, вид которого показан на рисунке 1.

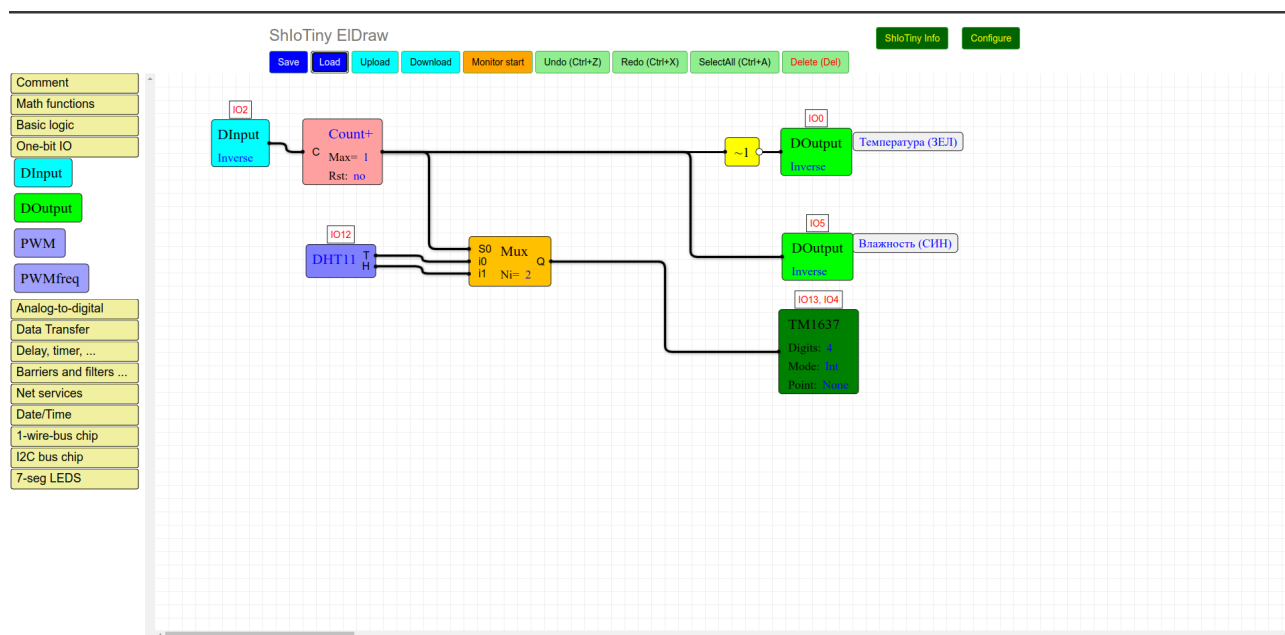


Рисунок 1 – Вид редактора схем с нарисованной схемой-программой NTP-часы

## ShloTiny5 – сделай сложности проще

Разумеется, если схемы в памяти модуля ESP нет, то на рабочем поле будет чистый лист.

Итак, что мы видим.

Слева находится «палитра узлов» или просто «палитра» – то есть те узлы, которые мы можем вставлять в схему.

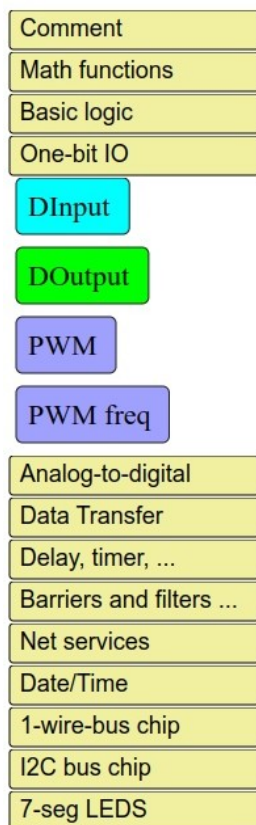
Сверху находятся кнопки управления редактором. Они позволяют сохранять схему в файл или загружать её на устройство. Но об этом ниже.

Рабочее поле редактора – это то самое место, где нарисована наша схема-программа, то есть алгоритм работы модуля ESP под управлением ПО ShloTiny5.

Кроме этого, справа вверху находятся ссылки для перехода к другим вкладкам управления модулем ESP – «ShloTiny Info» и «Configure». Эти вкладки к редактору отношения не имеют, поэтому на них останавливаться не будем. В онлайн-версии редактора они вообще отсутствуют.

Теперь пройдемся по всем частям редактора отдельно и более подробно. В этом разделе рассмотрим только функции редактирования, без привязки их к функционированию узлов.

### 4.1 Палитра узлов



Палитра узлов или просто палитра – это те узлы, которые мы можем вставлять в схему.

Она состоит из нескольких раскрывающихся вкладок, с помощью которых узлы сгруппированы по функциям.

Работать с палитрой довольно просто.

Одиночный клик на любой раскрывающейся вкладке приведёт к тому, что указанная вкладка раскроется, а остальные вкладки закроются. То есть в каждый момент времени видно только содержимое одной вкладки.

Двойной клик по на любой раскрывающейся вкладке приведёт поочередно к раскрытию-закрытию всех вкладок. Это если очень хочется посмотреть все узлы разом.

Чтобы вставить узел из палитры в схему, необходимо просто перетащить его мышью из палитры в рабочее поле редактора:

– навести курсор мыши на нужный узел в палитре;

- нажать левую клавишу мыши;
- не отпуская левой клавиши мыши перетащить узел из палитры в рабочее поле редактора.

Собственно, это всё, что надо знать о работе с палитрой.

Некоторые узлы доступны в ограниченном количестве экземпляров. То есть больше заданного количества на схему их добавить нельзя. В этом случае, узел в палитре помечается серым цветом и перетащить его в схему уже не получится. При удалении такого узла из схемы, он вновь становится доступен в палитре.

### 4.2 Узлы

Узел – это визуальное представление на схеме устройства или действия.

То есть узел является источником, приёмником или обработчиком информации.

Каждый узел может иметь следующие части:

- входы;
- выходы;
- меню функций узла;
- поле параметров узла;
- меню привязки к выводам GPIO;
- меню выбора адреса устройства.

Чтобы не запутаться в этом многообразии, рассмотрим примеры и увидим, что не всё так страшно.

Самый простой узел – комментарий. Этот узел ничего не делает.

**Комментарий - любой текст**

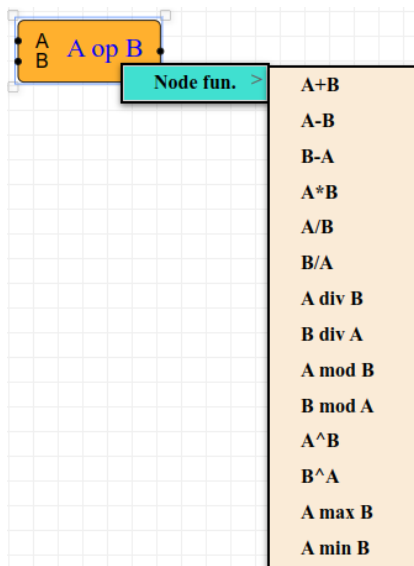
Он служит лишь для того, чтобы вставлять на схему пояснения. Но он содержит поле параметров узла, состоящее из одного элемента – текста комментария.

Заметим, что это поле – синего цвета. Все поля узла, которые можно редактировать, выделены синим цветом. Те, которые нельзя редактировать – показаны чёрным цветом.

То есть, если внутри узла вы видите синий текст, то кликнув на него мышкой, вы сможете что-то поменять. Это может быть или ручной ввод параметра (числа, строки) или контекстное меню – в зависимости от конкретного узла и параметра.

### 4.2.1 Узлы с меню выбора функции

Возьмём простой узел «А ор В». Это узел, осуществляющий математическую операции с двумя операндами.



Тут мы видим два входа, расположенных слева и один выход, расположенный справа.

Входы узлов (приёмники данных) всегда расположены слева. А выходы узлов (источники данных) – всегда расположены справа.

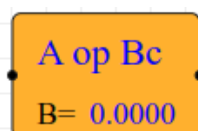
Входы узла обозначены А и В, чтобы было понятно, какой операнд на какой вход узла поступает.

Название узла «А ор В» выделено синим цветом, значит этот узел имеет меню функций узла. При клике мышкой на название узла появится контекстное меню выбора функции, как показано на рисунке.

После выбора функции в меню название узла сменится на название выбранной функции.

### 4.2.2 Узлы с полем параметров узла

Возьмём другой узел «А ор Вс». Это узел, осуществляющий математическую операции с двумя операндами, как и предыдущий узел. Но только операнд В в данном узле – константа. Поэтому узел имеет один вход А и один выход.

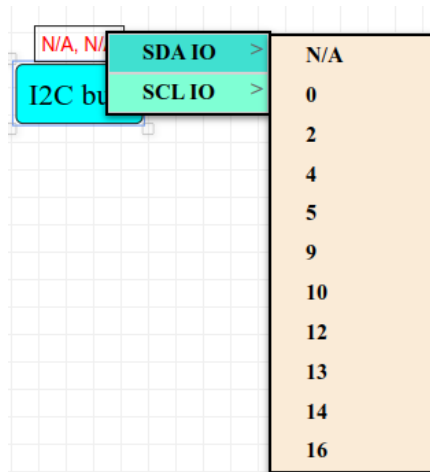


Операнд В задаётся прямо в поле параметров узла, выделенном синим цветом. Двойной клик мыши на выделенный синим цветом параметр «В=0.0000» вызовет появление строки редактирования, в которой можно задать своё число – операнд В.

Отметим, что помимо задания параметров узла с помощью строки ввода, в некоторых узлах используются контекстные меню. Конкретика находится в разделе описания узлов ниже.

### 4.2.3 Узлы с привязкой к ножкам микроконтроллера – GPIO

Теперь рассмотрим узел, который представляет собой аппаратное устройство – шину I2C. Аппаратное устройство подключено к модулю ESP при помощи каких-то выводов GPIO.



Номера привязанных к узлу выводов GPIO отображаются красным сверху узла. Если вывод не привязан – то выводится надпись «N/A».

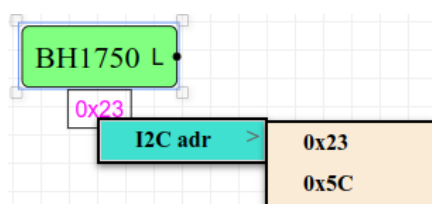
Для привязки выводов GPIO к узлу служит «контекстное меню привязки к выводам GPIO». Чтобы попасть в это меню, достаточно кликнуть мышью в область, где отображаются номера выводов GPIO, привязанных к узлу (сверху узла).

Затем можно выбрать назначение вывода GPIO и его номер. Если выбрать «N/A», то вывод GPIO будет отключён от узла.

Выводы GPIO, которые уже привязаны к какому-то узлу, исчезают из меню привязки выводов GPIO и не могут быть привязаны к другому узлу. Это позволяет избежать дублирования привязок.

### 4.2.4 Узлы с адресом устройства на шине

Ещё один узел аппаратного устройства – датчика освещённости BH1750, подключённый по шине I2C – продемонстрирует адресную часть узла.



К одной шине I2C может быть подключено несколько разных устройств. Чтобы отличить устройство одно от другого у них имеется уникальный адрес. Если устройство позволяет задать несколько адресов, то это можно сделать в адресной части узла. Она аналогична области привязки узла к выводам GPIO, но располагается не сверху, а снизу от узла.

Для задания адреса устройства, привязанного к узлу, служит «контекстное меню выбора адреса устройства». Чтобы попасть в это меню, достаточно кликнуть мышью в область, где отображается адрес устройства, привязанного к узлу (снизу узла). Затем можно выбрать адрес узла из возможных.

### 4.2.5 Особенности ввода и просмотра чисел в редакторе ElDraw5 в различных системах счисления

Все числа в редакторе ElDraw5 можно вводить в различных системах счисления: десятичной, двоичной, восьмеричной или шестнадцатеричной.

Так как аппаратура часто связана с обработкой отдельных бит и групп бит, то возможность ввода чисел в разных системах счисления очень удобна.

## ShloTiny5 – сделай сложности проще

При вводе в поле параметров узла чисел в любой системе счисления, отличной от 10-й, число автоматически переводится в 10-ю систему счисления и в поле параметров узла всегда отображается в 10-м виде.

Для ввода чисел, в системе отличной от 10-й, перед вводимым числом необходимо вставить префикс.

Десятичные (10-е) числа записываются без префикса.

Двоичные (2-е) числа записываются с префиксом **0b**.

Восьмеричные (8-е) числа записываются с префиксом **0o**.

Шестнадцатеричные (16-е) числа записываются с префиксом **0x**.

В таблице ниже показаны несколько примеров записи одного и того же числа в различных системах счисления

№	10-е число (пример)	2-е число BIN	8-е число OCT	16-е HEX
		<b>Префикс 0b</b>	<b>Префикс 0o</b>	<b>Префикс 0x</b>
1	12	0b1100	0o14	0x0C
2	100	0b1100100	0o144	0x64
3	1876	0b11101010100	0o3524	0x754

Так как все числа в поле параметров узлов представлены в 10-м виде, то для удобства к полям ввода чисел привязаны всплывающие подсказки, на которых отображается введенное в поле ввода число в различных системах счисления – OCT, HEX и BIN.

148

Oct: 224  
Hex: 94  
Bin: 10010100

Если в поле ввода находится целое число, то при наведении на него указателя мыши, всплывёт подсказка как на рисунке слева.

Если в поле ввода находится число с плавающей точкой, то при наведении на него указателя мыши, всплывёт подсказка как на рисунке справа. Обратите внимание, что число с плавающей точкой вначале будет округлено до целого и только потом представлено в различных системах счисления.

175.473

Floating point  
Rounded Dec: 175  
Oct: 257  
Hex: AF  
Bin: 10101111



## 4.3 Связи

### 4.3.1 Как проводить связи между узлами

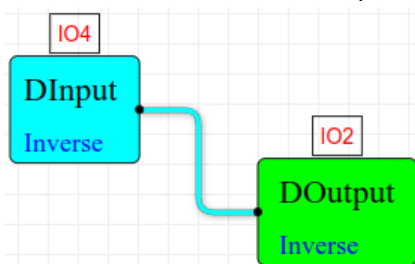
Чтобы программа заработала необходимо описать её алгоритм. В нашем случае это значит, что надо указать какие узлы используются и как они связаны между собой.

Связать можно только вход одного узла с выходом другого узла и не иначе.

Причём к выходу узла можно подключить любое число входов других узлов. Но ко входу узла можно подключить только один выход другого узла.

Входы узла нельзя подключить к выходу того же самого узла.

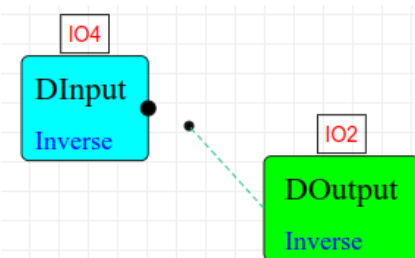
Рассмотрим простую схему, передающую состояние входа GPIO4 на выход GPIO2. Если ко входу GPIO4 подсоединить кнопку, то при её нажатии будет загораться светодиод, подключенный к выходу GPIO2.



Для того, чтобы создать эту простейшую схему, надо перетащить из палитры в рабочую область два узла – «Dinput» и «Doutout».

Затем нужно задать номера GPIO, привязанных к этим узлам, как было показано ранее.

И, наконец, надо соединить выход узла «Dinput» со входом узла «Doutout». После этого схему можно загружать в устройство на выполнение.



Чтобы соединить вход одного узла с выходом другого узла, необходимо навести мышкой на выбранный вход нужного узла и нажать левую клавишу мыши.

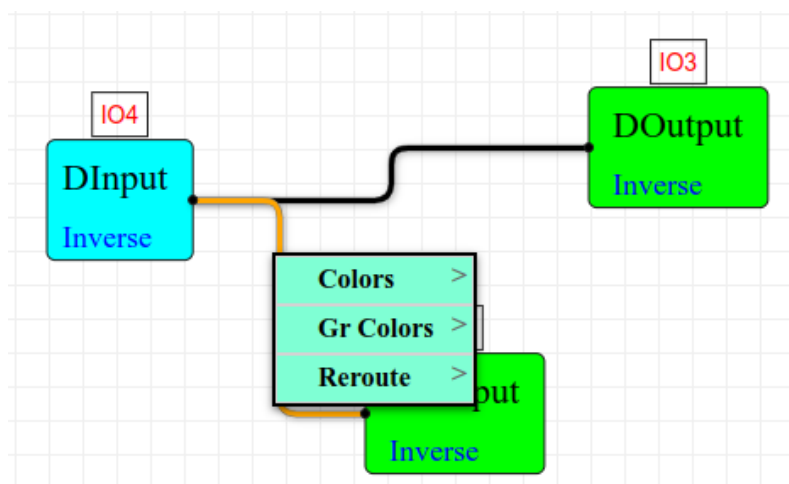
Затем, не отпуская левой клавиши мыши, надо перетащить выбранный вход к нужному выходу другого узла. После чего нужно отпустить левую клавишу мыши. Появится связь между узлами.

Если связь расположена неудобно (например перекрывается другими узлами), то любой сегмент связи можно выделить левой клавишей мыши и переместить его влево-вправо или вверх-вниз, удерживая нажатой левую клавишу мыши.

### 4.3.2 Меню свойств связи

Меню свойств связи служит для установки свойств связи и переразводки «запутанных» связей.

Для вызова меню свойств связей надо кликнуть правой клавишей мыши на связь, свойства которой вы хотите изменить.



На экране появится меню управления свойствами связи, как показано на рисунке слева.

Пункт **Colors** позволяет изменить цвет отдельной связи.

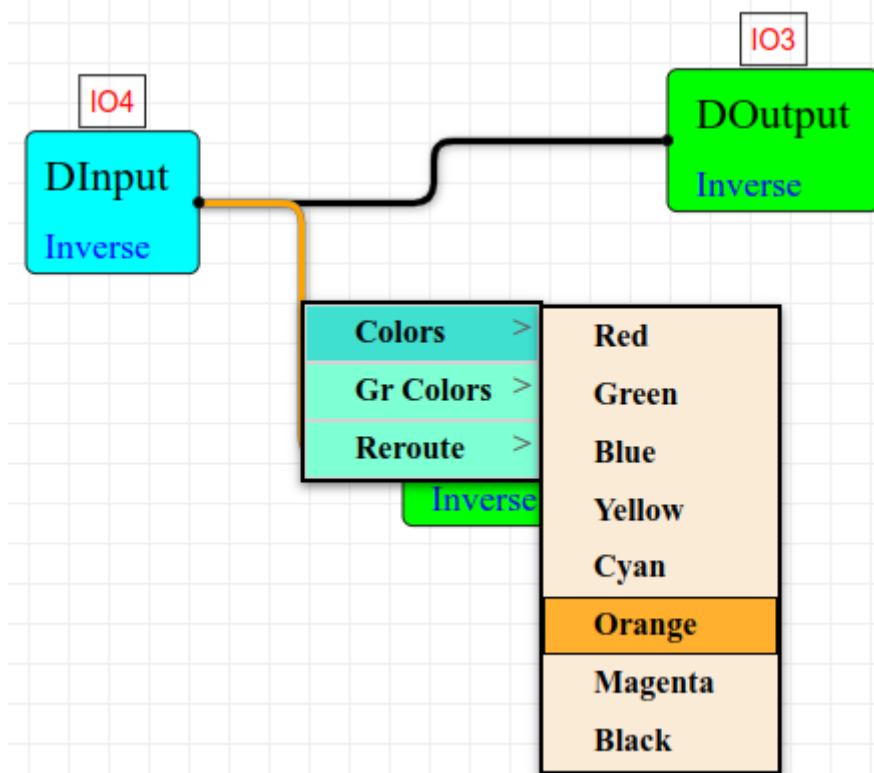
Пункт **GrColors** позволяет изменить цвет группы

связей, подключённых к одному выходу.

Пункт **Reroute** позволяет автоматически перестроить отдельную связь, группу связей или вообще все связи на экране – «переразвести» их.

### 4.3.3 Изменение цвета связи или группы связей

Для удобства можно пометить цветом связь или группу связей, подключённую к одному выходу. Для этого надо вызвать «Меню свойств связи» (см пп. 4.3.2).



За управление цветом отдельной связи отвечает подменю «**Colors**».

За управление цветом группы связей отвечает подменю «**GrColors**».

Для установки цвета связи (группы связей) следует выбрать меню «**Colors**» («**GrColors**»). Затем надо выбрать в указанном подменю

## ShloTiny5 – сделай сложности проще

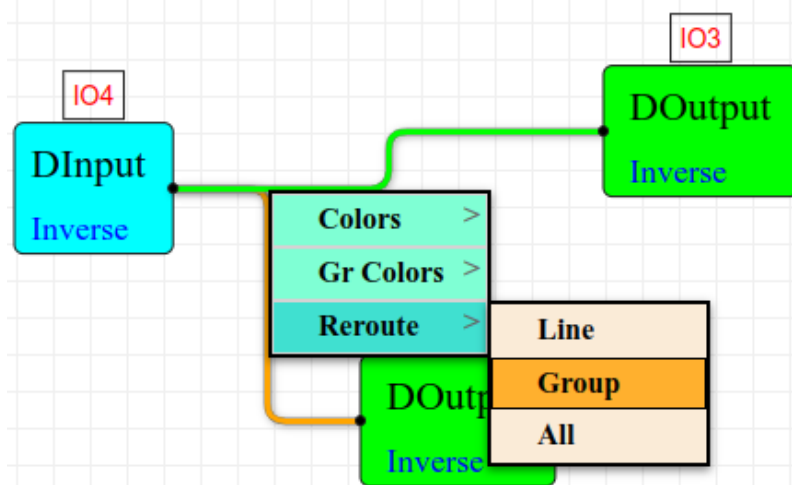
желаемый цвет для пометки связи. После этого связь (группа связей) окрасится в выбранный цвет. В таблице ниже приведено соответствие названий цветов меню и цветов связей.

№	Название цвета в меню	Цвет
1	Red	Красный
2	Green	Зелёный
3	Blue	Синий
4	Yellow	Желтый
5	Cyan	Голубой
6	Orange	Оранжевый
7	Magenta	Розовый
8	Black	Чёрный

Никакой функциональной нагрузки цвета связей не несут. Они предназначены для визуального удобства при составлении схем с большим количеством узлов и связей.

### 4.3.4 Автоматическое перестроение связей

В процессе построения схемы, когда узлы и связи перемещаются много раз вручную, иногда возникают узлы, петли и прочие визуально неприятные связи. Для устранения этого эффекта служит подменю «**Reroute**» – автоматическое перестроение связей. Это подменю, показанное на рисунке ниже, находится в «**Меню свойств связи**» (см пп. 4.3.2 ).



При выборе пункта «**Line**» перестраивается одна связь, которая выбрана.

При выборе пункта «**Group**» перестраивается группа связей, в которую входит выбранная связь (подключены к одному выходу).

При выборе пункта «**All**» перестраиваются все связи на схеме.

#### 4.4 Кнопки управления, горячие клавиши

Кнопки управления редактором (кнопки управления) находятся сверху.

**Save** – сохранить схему-программу из редактора на диск. Можно сохранять недорисованные схемы (с неподключенными входами, с незадаанными функциями и привязками GPIO). Схема сохраняется в формате JSON. Руками туда лучше не лазить. Будет больно.

**Load** – загрузить схему-программу с диска в редактор.

**Upload** – загрузить схему в устройство. Производится проверка правильности схемы (все входы подключены, функции узлов заданы, GPIO привязаны). Имеет смысл только при работе с модулем ESP. В онлайн-версии редактора отсутствует.

Если в устройство загрузить пустую схему, т.е. чистый лист, то это будет означать, что схема устройстве будет удалена!

**Download** – считать схему из устройства в редактор. Все изменения в редакторе удаляются, схема перечитывается из устройства. Имеет смысл только при работе с модулем ESP. В онлайн-версии редактора отсутствует.

**Monitor start** – запустить отладочный монитор, показывающий состояние входов и выходов узлов. Имеет смысл только при работе с модулем ESP. В онлайн-версии редактора отсутствует.

**Undo (Ctrl+Z)** – отменить последнее изменение. Горячая клавиша CTRL+Z. Обратите внимание, что не всё можно отменить. Например, изменение числа входов или выходов узла пока отменить нельзя.

**Redo (Ctrl+X)** – вернуть последнее отменённое изменение. Горячая клавиша CTRL+X. Обратите внимание, что не всё можно вернуть. Например, изменение числа входов или выходов узла пока вернуть нельзя.

**SelectAll (Ctrl+A)**

## ShloTiny5 – сделай сложности проще

– выделить все элементы схемы. Горячая клавиша CTRL+A.

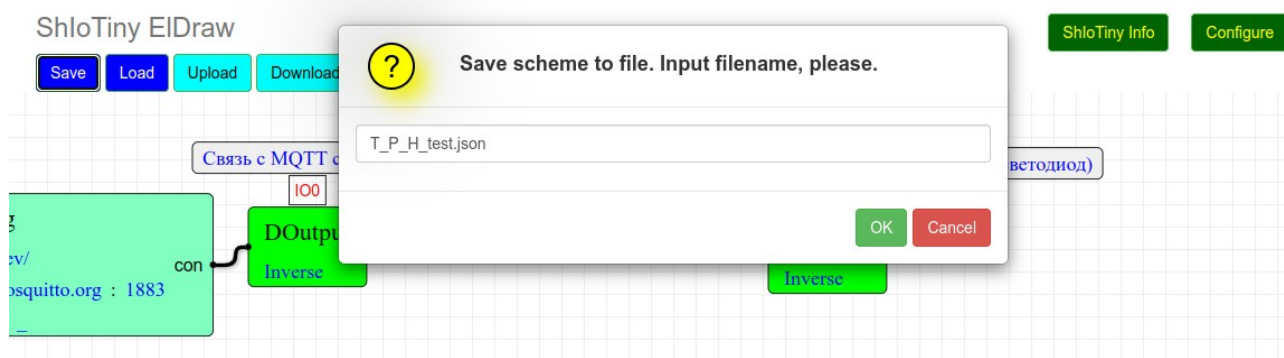
Delete (Del)

– удалить выделенные элементы схемы. Горячая клавиша DEL.

### 4.5 Сохранение схемы в файл

Для сохранения разработанной схемы-программы в файл возможно с помощью кнопки **Save**.

При нажатии этой кнопки появляется диалоговое окно ввода имени файла, как на приведённом рисунке.



При нажатии на кнопку **OK** произойдёт сохранение схемы в файл с указанным именем штатными средствами WEB-браузера.

При нажатии на кнопку **Cancel** сохранение схемы в файл не производится и произойдёт возврат к редактированию схемы-программы.

### 4.6 Загрузка схемы из файла

Для загрузки схемы-программы из файла возможно с помощью кнопки **Load**. При нажатии на эту кнопку произойдёт вывод штатного диалогового окна WEB-браузера, в котором необходимо выбрать загружаемый файл.

При успешной загрузке, старая схема удалится, а новая появится в поле редактирования.

### 4.7 Загрузка схемы в устройство ShloTiny5 на выполнение

Для сохранения разработанной схемы-программы в устройство ShloTiny5 возможно с помощью кнопки **Upload**.

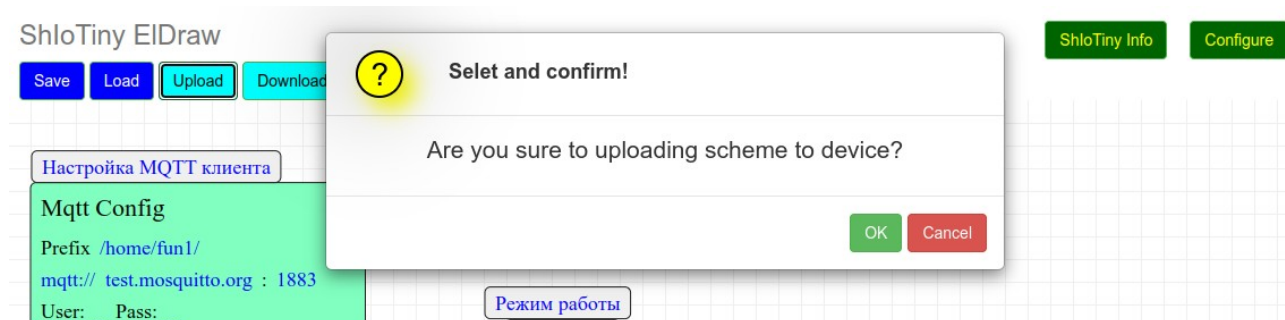
Чтобы схема могла быть загружена в устройство ShloTiny5 к ней предъявляются ряд требований (см. пп. 4.7.2).

При несоблюдении этих требований редактор выведет сообщение об ошибке и вернёт пользователя в режим редактирования схемы-программы. При этом, схема в устройство загружена не будет. Подробнее об этих требованиях – ниже.

## ShloTiny5 – сделай сложности проще

Итак, мы ввели схему, которую хотим выполнить на нашем устройстве ShloTiny5. Для этого нажимаем кнопку **Upload**.

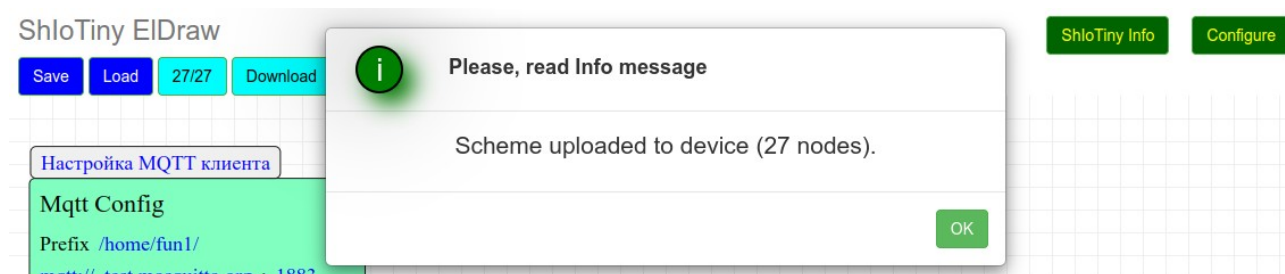
Если все необходимые требования к схеме соблюдены, то в окне браузера появится диалог подтверждения загрузки схемы в устройство, как на рисунке ниже.



При нажатии на кнопку **Cancel** загрузка схемы в устройство не производится и произойдёт возврат к редактированию схемы-программы.

При нажатии на кнопку **OK** произойдёт загрузка схемы в устройство.

В случае успешной загрузки схемы-программы в устройство ShloTiny5, в окне браузера появится сообщение об успешной загрузке схемы в устройство, как на рисунке ниже. Также в этом окне выводится количество загруженных в устройство узлов.



После успешной загрузки схемы-программы в устройство ShloTiny5, её выполнение автоматически начинается сразу же.

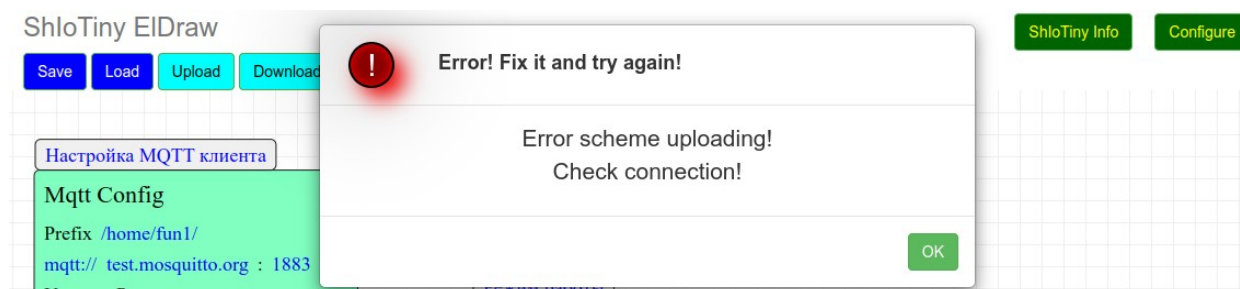
В процессе загрузки схемы-программы в устройство ShloTiny5 могут возникать различные ошибки. Рассмотрим эти ошибки и способы их устранения.

### 4.7.1 Ошибка связи с устройством

Пока вы рисовали свою программу в редакторе, связь с устройством могла быть потеряна. Редактор не поддерживает постоянного контроля связи с устройством ShloTiny5, поэтому о потере связи станет известно только при попытке передачи данных из или в устройство ShloTiny5. В

## ShloTiny5 – сделай сложности проще

случае потери связи, при попытке загрузки схемы-программы в устройство ShloTiny5, на экран выведется сообщение об ошибке, как на рисунке ниже.



Что делать в этом случае?

Во-первых, чтобы схема не пропала, надо сохранить её в файл. Вообще, это лучше делать почаще, чтобы не было мучительно больно за пропавшие труды.

Во-вторых, необходимо убедиться, что само устройство ShloTiny5 включено.

Во-третьих, необходимо проверить, что ваша точка доступа WIFI, к которой подключено устройство ShloTiny5, включена и доступна. Если, конечно, ShloTiny5 сконфигурировано в режимах «Station mode» или «AP+Station mode».

В-третьих, если устройство сконфигурировано в режимах «Config mode» или «AP mode» – нужно проверить, что оно видно как точка доступа WiFi.

Конечно, всех возможных ситуаций предусмотреть нельзя, но думаю, что пытливым читателем инструкции разберётся с возникшим у него казусом связи с устройством.

После того, как причина отсутствия связи найдена и устранена, можно попытаться загрузить схему-программу в устройство ShloTiny5 вновь.

### 4.7.2 Ошибки конфигурации схемы-программы

Другой тип ошибок, которые возникают при попытке загрузить схему-программу в устройство ShloTiny5 – это неверная конфигурация самой схемы-программы.

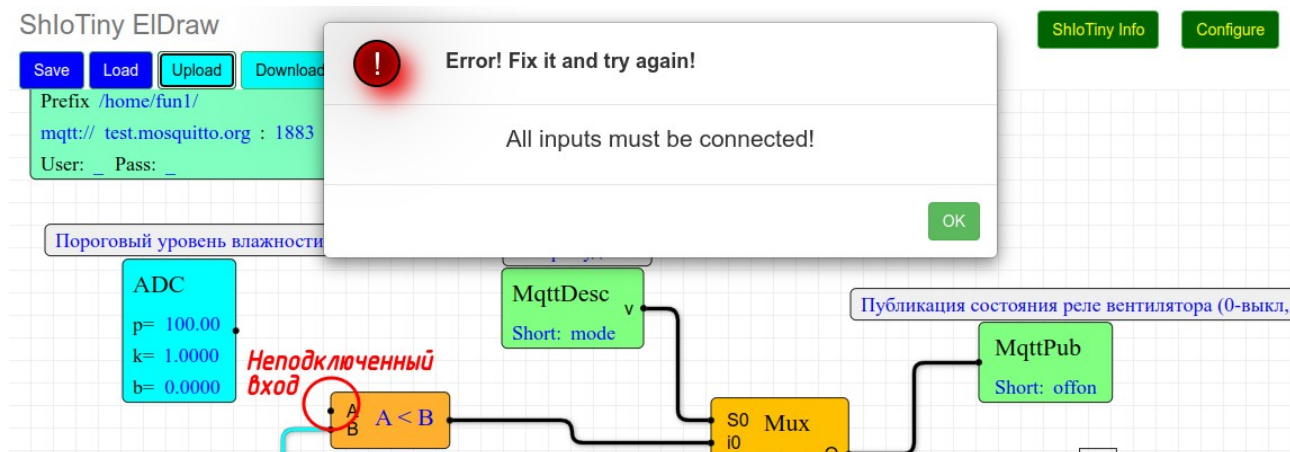
В отличие от сохранения схемы в файл, для того, чтобы схема-программа загрузилась и начала исполняться на устройстве ShloTiny5, она должна соответствовать ряду требований.

Рассмотрим эти требования.



#### 4.7.2.1 Требование привязки входов

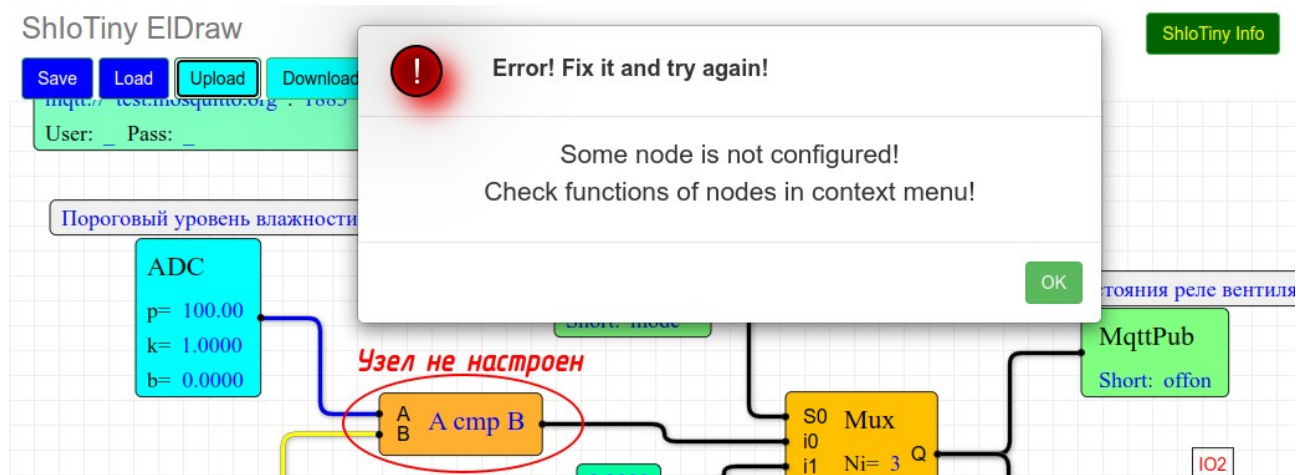
Все входы узлов должны быть соединены с выходами других узлов. «Висячие входы» не допускаются. К выходам узлов таких требований нет, они могут оставаться не подключёнными. Если в схеме есть неподключенные входы узлов, то при попытке загрузить схему-программу в устройство, будет выведено сообщение об ошибке, как на рисунке.



На рисунке показана причина ошибки - «висящий», то есть никуда не подключенный вход узла. Устранить причину этой ошибки просто - необходимо «висящий вход» узла подключить к выходу другого узла.

#### 4.7.2.2 Требование настройки функций

Все функции узлов должны быть настроены. То есть, если узел имеет несколько функций, выбираемых из контекстного меню, то функция должна быть выбрана. Если в схеме есть неконфигурированные узлы, то при попытке загрузить схему-программу в устройство, будет выведено сообщение об ошибке, как на рисунке.

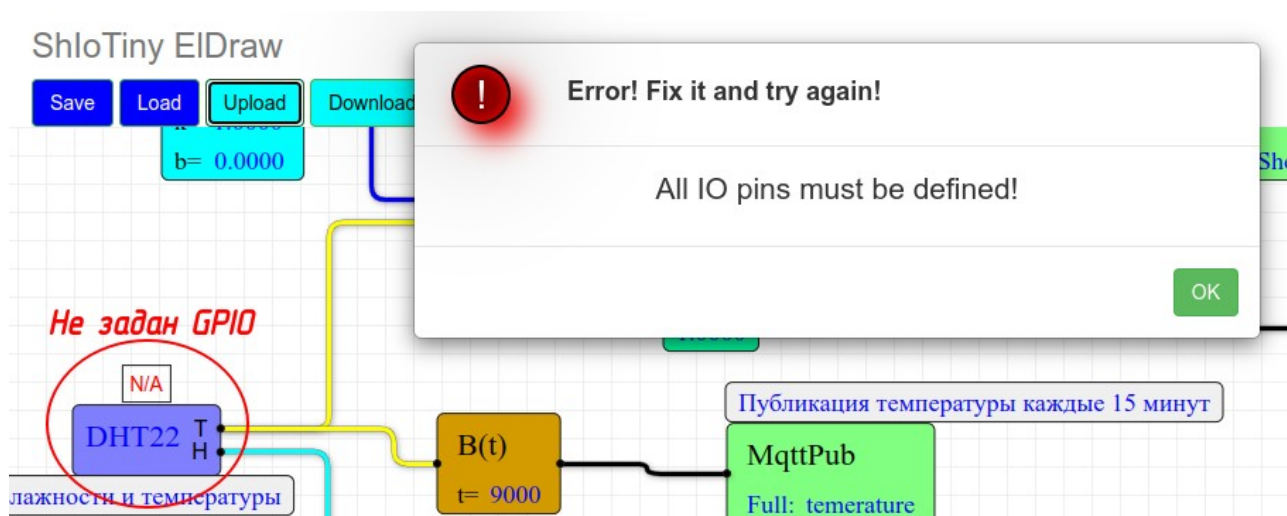


## ShloTiny5 – сделай сложности проще

В примере на рисунке – для устранения ошибки необходимо выбрать функцию узла «А стр В» из контекстного меню.

### 4.7.2.3 Требуемое привязки GPIO

Все GPIO узлов должны быть привязаны к ножкам микроконтроллера. То есть, если узел привязывается к GPIO, то номера GPIO должны быть указаны. Если в схеме есть не привязанные GPIO, то при попытке загрузить схему-программу в устройство, будет выведено сообщение об ошибке, как на рисунке.



Не привязанные GPIO узлов обозначаются «N/A» вместо номера GPIO. Устранить причину этой ошибки тоже несложно – надо выбрать из контекстного меню привязки GPIO нужный номер ножки микроконтроллера и привязать её к узлу.

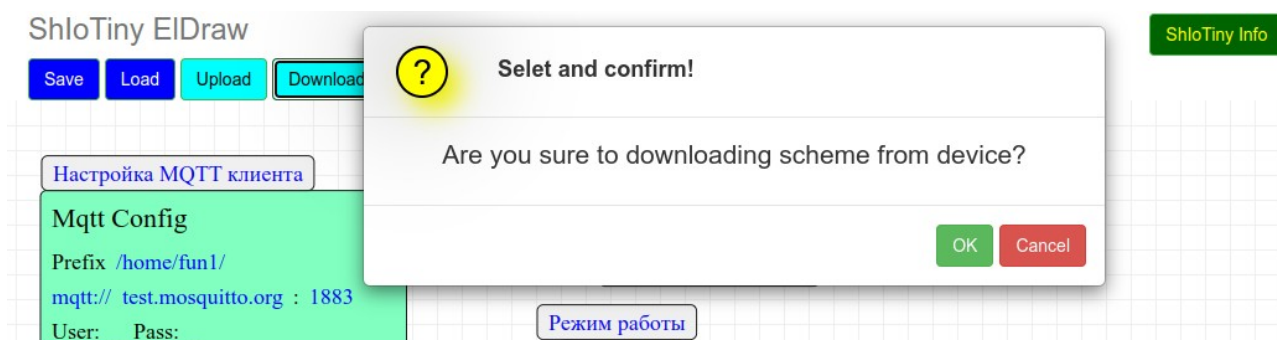
## 4.8 Считывание схемы из устройства ShloTiny5

Обычно в считывании схемы из устройства ShloTiny5 нет необходимости. Схема загружается в редактор EIDraw из устройства автоматически.

Но если вы решили отказаться от изменений, которые произвели в редакторе EIDraw, но не успели загрузить в устройство, то вам поможет кнопка принудительного считывания схемы из устройства – [Download](#).

При нажатии на эту кнопку, в окне браузера появится диалог подтверждения загрузки схемы в устройство, как на рисунке ниже.

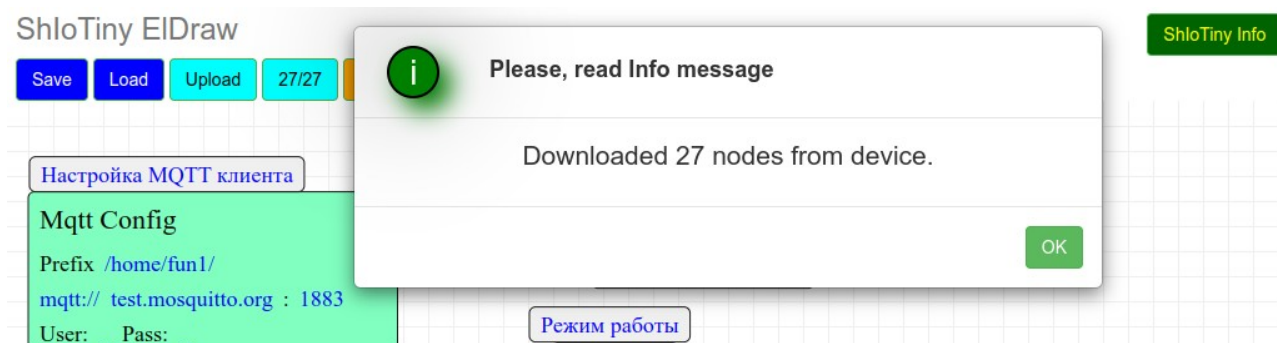
## ShloTiny5 - сделай сложности проще



При нажатии на кнопку **OK** произойдёт считывание схемы из устройства. При этом схема, которая было в окне редактора EIDraw5, удаляется и в окне редактора появляется схема, считанная из устройства.

При нажатии на кнопку **Cancel** считывание схемы из устройства не производится и произойдёт возврат к редактированию схемы-программы.

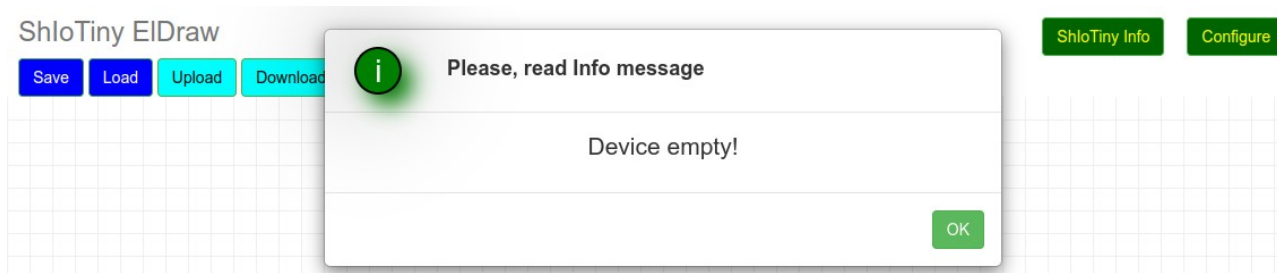
При успешном считывании схемы из устройства ShloTiny5, в окне браузера появляется сообщение с количеством считанных узлов, как на рисунке ниже.



Если возникают ошибки, то выводится окно сообщений об ошибке. Единственная ошибка, которая может возникнуть при считывании схемы – это ошибка потери связи, полностью аналогичная пп. 4.7.1.

В случае, если в устройстве ShloTiny5 нет загруженной схемы-программы, то оно называется «пустым».

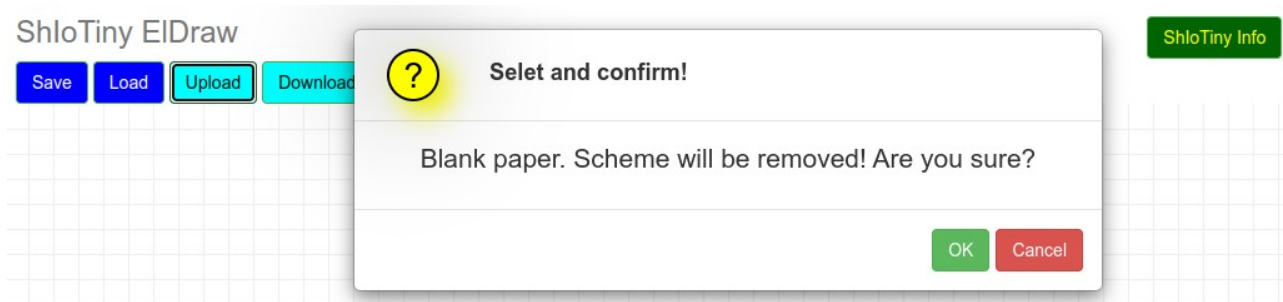
При попытке считывания схемы из такого «пустого» устройства, на экран выводится сообщение, как на рисунке ниже.



## 4.9 Удаление схемы из устройства

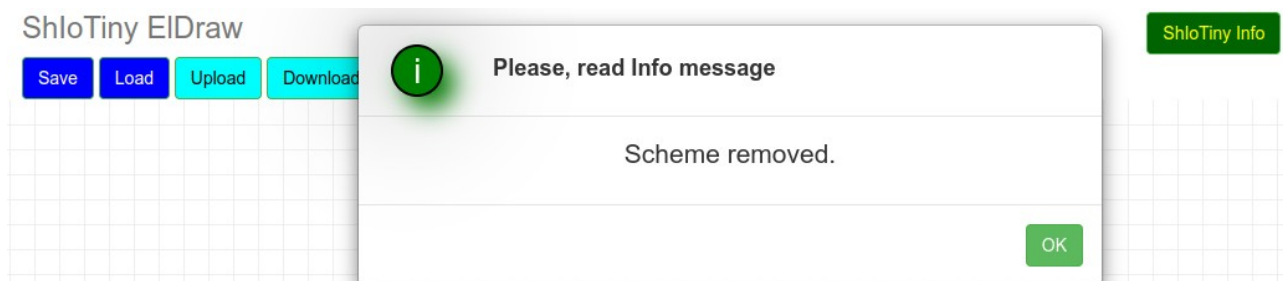
Если вы хотите полностью удалить схему из памяти устройства, то для этого надо загрузить в устройство «чистый лист».

То есть, необходимо удалить из редактора все узлы, а затем нажать кнопку **Upload**. При этом в окне браузера появится диалог подтверждения, как на рисунке ниже.



При нажатии на кнопку **OK** произойдет удаление схемы из памяти устройства.

При нажатии на кнопку **Cancel** удаление схемы из устройства не производится и произойдет возврат к редактированию схемы-программы. При успешном удалении схемы из памяти устройства, в окне браузера появится сообщение, как на рисунке ниже.



Если возникают ошибки, то выводится окно сообщений об ошибке. Единственная ошибка, которая может возникнуть при удалении схемы из устройства – это ошибка потери связи, полностью аналогичная пп. 4.7.1.

## 4.10 Режим «Монитор»

Для отладки статических состояний схемы существует режим «Монитор».

Суть режима заключается в том, что редактор EIDraw периодически считывает из устройства состояние выходов всех узлов и отображает их в окне браузера.

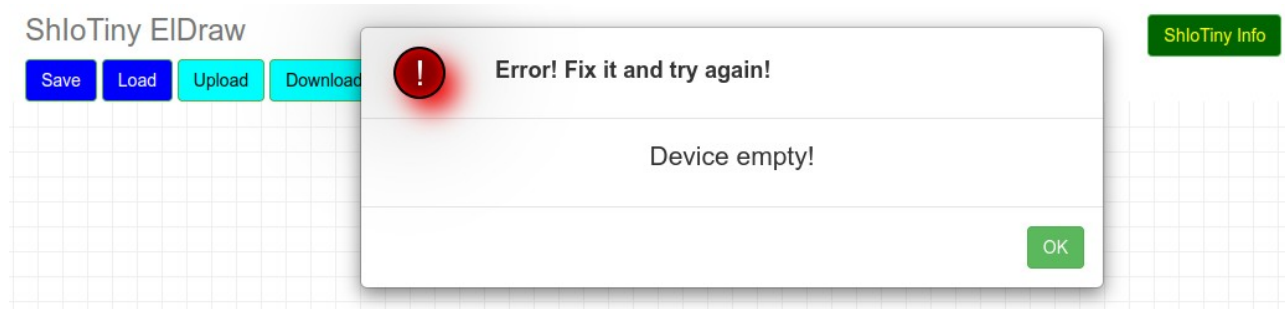
## ShloTiny5 – сделай сложности проще

В связи с тем, что запросы редактора к устройству идут относительно медленно, режим «Монитор» подходит только для медленных процессов. Увидеть быстро меняющиеся состояния выхода узла в нем не удастся.

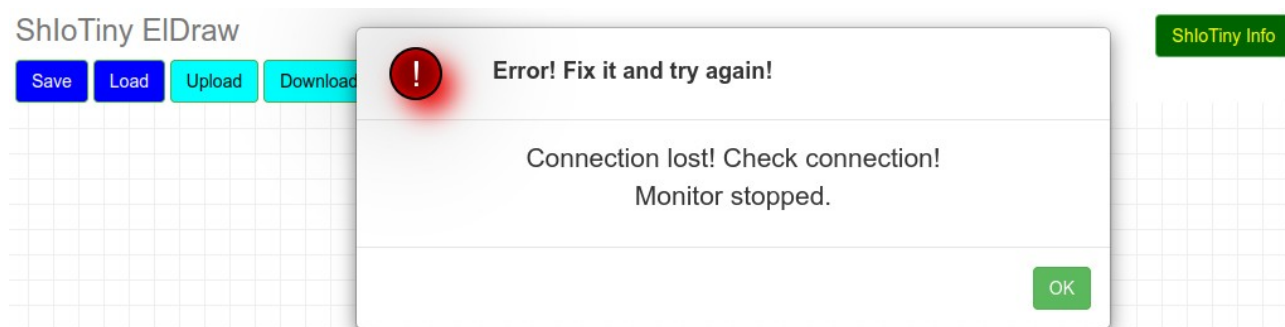
Включение режима «Монитор» производится нажатием кнопки **Monitor start**. После чего, надпись на этой кнопке меняется на **Monitor stop** и возле выходов и входов узлов появляются считанные из устройства текущие значения на этих выходах и входах.

Отключение режима «Монитор» производится нажатием кнопки **Monitor stop**. После этого надпись на этой кнопке меняется на **Monitor start** и данные о состояниях выходов и входов узлов удаляются с экрана, опрос состояния выходов узлов устройства прекращается.

Если у устройстве ShloTiny5 отсутствует схема или она не запущена на выполнение (например, при загрузке устройства в «Config mode»), то появляется сообщение об ошибке, как на рисунке ниже.



Если во время работы «монитора» прервалась связь с устройством, то спустя некоторое (довольно длительное) время в окне браузера будет выведено сообщение об ошибке, как на рисунке ниже.



При возникновении такой ошибки рекомендуется после восстановления связи с устройством обновить окно браузера.

#### ***4.11 Выбор и перемещение элементов схемы***

Выделение (выбор) элементов схемы осуществляется стандартным образом для многих программ образом – с помощью мыши или сочетания мыши и клавиши SHIFT.

##### ***4.11.1 Выбор одного элемента схемы***

Выбор одного элемента схемы (узла или связи) осуществляется с помощью мыши. Для выбора нужного элемента схемы необходимо кликнуть на него левой клавишей мыши. После этого на нем возникнет обозначение выделения. При этом выделение со всех ранее выбранных элементов снимается.

##### ***4.11.2 Выбор нескольких элементов схемы***

Выбор нескольких элементов схемы (узлов или связей) осуществляется с помощью мыши и клавиши SHIFT. Для выбора нужных элементов схемы необходимо нажать клавишу SHIFT и, удерживая её нажатой, кликнуть по очереди на нужные элементы левой клавишей мыши. После этого на них возникнет обозначение выделения.

##### ***4.11.3 Выбор всех элементов схемы***

Выбор всех элементов схемы осуществляется с помощью клика мыши на кнопку управления «Select All» или сочетанием клавиш CTRL+A.

##### ***4.11.4 Снятие выбора элементов схемы***

Снятие выбора (выделения) элементов схемы осуществляется кликом левой клавиши мыши в любом месте рабочей области редактора, свободной от элементов – узлов и связей.

##### ***4.11.5 Перемещение элементов схемы***

Перемещение выделенных элементов производится с помощью левой клавиши мыши.

Чтобы переместить выбранные элементы, необходимо их выделить, как было указано ранее.

Затем, наведя курсор мыши на любой из выделенных элементов, нажать левую клавишу мыши.

## ShloTiny5 – сделай сложности проще

После этого, не отпуская левую клавишу мыши, можно перемещать выделенные элементы с помощью перемещения мыши.

После того как выделенные элементы перемещены в желаемое место, левую клавишу мыши нужно отпустить.

## 5 Алгоритмы, Узлы, События

Программа – это некий алгоритм, описанный на некоем формальном языке.

Алгоритм – это чёткая пошаговая последовательность действий, указывающая откуда данные вводятся, как обрабатываются и куда выводятся.

Способ описания алгоритмов бывает разный: языки программирования, блок-схемы, графы и так далее.

Для программирования модуля ESP под управлением ПО ShloTiny5 используется графическое представление алгоритма обработки данных, сходное чем-то с электрическими схемами.

Весь алгоритм работы устройства под управлением ПО ShloTiny5 описывается тремя сущностями – узлами, связями между ними и событиями, передающимися от одного узла к другому по связям.

Узел – это графическое представление физического устройства или функции обработки данных.

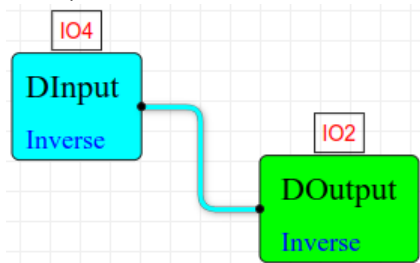
Связь – это описание того, каким путём пойдут данные и события с выхода одного узла на вход другого узла.

Событие – это какое-то явление, которое заставляет узел активизироваться и начать обработку данных. Данные, в зависимости от узла, могут поступать от других узлов, от аппаратуры или от таймеров.

Например, события возникающие на выходе узла вызывают активацию узлов, подключённых своими входами к выходам этого узла.

Следует различать данные на выходе узла и события, посылаемые узлом. Если узел изменил данные на своём выходе, но не послал на этот выход событий, то узлы, получающие эти данные не активируются и не обработают данных.

Рассмотрим простейшую схему, состоящую из бинарного входа (кнопки) и бинарного выхода (светодиода).



Пока состояние кнопки (узел Dinput GPIO4) не меняется, то никаких событий не генерируется и состояние светодиода (узел Doutput GPIO2) не меняется.

Как только кнопка будет нажата (или отпущена), то произойдёт несколько вещей:

- на уровне драйвера аппаратуры вызовется прерывание;



## ShloTiny5 – сделай сложности проще

- это прерывание вызовет появление события в узле Dinput GPIO4;
- на выходе узла установятся новые данные (0 или 1 – в зависимости от того нажата или отпущена кнопка);
- узел Dinput GPIO4 сгенерирует событие, которое передастся с выхода этого узла на вход узла Doutput GPIO2;
- узел Doutput GPIO2 примет это событие, считает данные с выхода узла Dinput GPIO4, затем передаст их в соответствующий драйвер аппаратуры, который в свою очередь установит состояние вывода GPIO2. И светодиод зажёгнется или погаснет в зависимости от того – придет в данных 0 или 1.

Передачей событий можно также управлять, как и передачей данных. Для управления передачей событий служит специальная группа узлов «Barriers and filters».

Эта группа узлов может пропускать или не пропускать данные и события дальше, в зависимости от указанных условий. Подробнее об этом см. описание группы узлов «Barriers and filters».

## 6 Описание узлов

### 6.1 Comment – комментарии и общесистемные параметры

**Comment** – Вкладка «Comment» содержит узел-комментарий и узлы общесистемной конфигурации.

#### 6.1.1 Comment – комментарий

**Comment** Узел Comment – позволяет делать комментарии на схеме. Никаких иных функций не несёт. Двойной клик мыши по узлу Comment вызывает строчный редактор. После редактирования комментария надо нажать Enter или кликнуть мышкой на пустое от элементов пространство рабочего поля редактора.

#### 6.1.2 Device Settings – общие параметры

**Device Settings** Узел Device Settings – это узел общесистемных параметров (будет дополняться).

Параметры, задаваемые в настройках узла:

Alias – название (псевдоним имени) устройства. Если узел не установлен в схему, то при обмене данными используется имя устройства по умолчанию (esp\_8266\_<id чипа>). Это не очень удобно. Поэтому можно переназначить имя узла на удобное, например «RoomWin0» или «Garden\_Light\_7». Под именем Alias устройство будет соединяться с MQTT-брокером или отправлять пакеты по протоколу UDP Rlink. «Родное» имя устройства (esp\_8266\_<id чипа>) всегда можно посмотреть на вкладке «ShloTiny Info».

Узел имеет три выхода, которые позволяют определить режим работы устройства (про режимы работы и их настройку см.пп. 7.1 ).

Выход «m» показывает номер нормального режима работы устройства, заданного на странице «Configure» в поле «ShloTiny mode».

Выход «s» показывает номер «безопасного» режима работы устройства, заданного на странице «Configure» в поле «Select safe mode».

Выход «?» имеет значение 0, если устройство находится в основном (нормальном) режиме работы и имеет значение 0, если устройство находится в указанном безопасном режиме. Про безопасный режим работы и его настройку см.пп. 7.4 ).

В таблице ниже показаны номера режимов работы устройства для выходов «m» и «s» данного узла.

№	Номер режима	Режим работы устройства	Примечание
	-1	Режим не выбран	Только для выхода «s». Безопасный режим отключён.
	0	Config mode	
	1	Station mode	Только для выхода «m».
	2	AP mode	
	3	AP+Station mode	Только для выхода «m».
	4	Single mode	

## 6.2 Math functions – математические функции

**Math functions** – Вкладка «Math functions» содержит узлы математических функций. Их немало.

### 6.2.1 Const – числовая константа

**Const**

Узел «Const» позволяет задать числовую константу. Допускаются положительные и отрицательные числа с плавающей точкой, а также нуль.

### 6.2.2 eConst – предопределённые специальные константы

**eConst**

Узел «eConst» позволяет задать предопределённые заранее заданные константы:  $\pi \approx 3.1415926$ ,  $e \approx 2.71828$ ,  $Inf = \infty$  (бесконечность) и NaN (не-число). Выбор осуществляется из контекстного меню функций узла.

### 6.2.3 Int/Check – проверки и округления

**Int/Check**

Узел «Int/Check» предоставляет специальные функции проверки числа на входе узла, а также функции округления. Выбор осуществляется из контекстного меню функций узла.

**6.2.3.1  $Inf?(a)$** 

$Inf?(a)$  – функция проверки входного значения на «бесконечность». На выходе узла устанавливается значение 1, если на входе узла установлено значение «бесконечность». Иначе на выходе узла устанавливается 0.

**6.2.3.2  $NaN?(a)$** 

$NaN?(a)$  – функция проверки входного значения на «не-число». На выходе узла устанавливается значение 1, если на входе узла установлено значение «не-число». Иначе на выходе узла устанавливается значение 0.

**6.2.3.3  $Ceil(a)$** 

$Ceil(a)$  – функция округления до меньшего целого (отброс значений после запятой).

**6.2.3.4  $Round(a)$** 

$Round(a)$  – функция округления до ближайшего целого.

**6.2.3.5  $Abs(a)$** 

$Abs(a)$  – функция вычисления абсолютного значения.

**6.2.4  $A \text{ op } B$ ,  $A \text{ op } Bc$  – математические операции над двумя операндами** **$A \text{ op } B$** 

Математические операции над двумя операндами.

 **$A \text{ op } Bc$** 

Узлы полностью аналогичны по функциям за исключением того, что узел « $A \text{ op } B$ » имеет два входа –  $A$  и  $B$ , а узел « $A \text{ op } Bc$ » имеет один вход  $A$ . Вторым операндом –  $B$  – узла « $A \text{ op } Bc$ » задаётся как константа в поле параметров узла. Выбор операции осуществляется из контекстного меню функций узла.

Доступны следующие операции.

N	Операция	Описание
1	$A+B$	Сложение чисел с плавающей точкой
2	$A-B$	Вычитание с плавающей точкой
3	$B-A$	Вычитание с плавающей точкой
4	$A*B$	Умножение с плавающей точкой
5	$A/B$	Деление с плавающей точкой
6	$B/A$	Деление с плавающей точкой

7	$A \div B$	Целочисленное деление
8	$B \div A$	Целочисленное деление
9	$A \bmod B$	Остаток от деления
10	$B \bmod A$	Остаток от деления
11	$A^B$	Возведение A в степень B
12	$B^A$	Возведение B в степень A
13	$A \max B$	Ограничение A не более B
14	$A \min B$	Ограничение A не менее B
15	$\text{NaN}(A)?B:A$	Если $A=\text{NaN}$ – то на выходе B, иначе на выходе A
16	$\text{Inf}(A)?B:A$	Если $A=\text{Inf}$ – то на выходе B, иначе на выходе A

### 6.2.5 A стр B, A стр Bc – сравнение двух чисел

**A стр B**

Узлы производят сравнение двух чисел, выставляя на выходе значение 1 (если условие сравнения выполняется) или 0 (если условие сравнения не выполняется).

**A стр Bc**

Узлы полностью аналогичны по функциям за исключением того, что узел «A стр B» имеет два входа – A и B, а узел «A стр Bc» имеет один вход A. Вторым операндом – B – узла «A стр Bc» задаётся как константа в поле параметров узла. Выбор условия сравнения осуществляется из контекстного меню функций узла.

Доступны следующие функции сравнения.

N	Условие	Описание
1	$A=B$	Если $A=B$ , на выходе 1, иначе на выходе 0
2	$A \neq B$	Если $A \neq B$ , на выходе 1, иначе на выходе 0
3	$A > B$	Если $A > B$ , на выходе 1, иначе на выходе 0
4	$A < B$	Если $A < B$ , на выходе 1, иначе на выходе 0
5	$A \geq B$	Если $A \geq B$ , на выходе 1, иначе на выходе 0
6	$A \leq B$	Если $A \leq B$ , на выходе 1, иначе на выходе 0

### 6.2.6 $kx+b$ – линейная функция

**$kx+b$**

Линейная функция осуществляет преобразование входного значения  $x$  согласно формуле  $y(x)=k \cdot x+b$ . На вход узла подаётся значение  $x$ , а на выходе узла устанавливается значение  $y(x)$ .

Значения коэффициентов  $k$  и  $b$  задаются в поле параметров узла.

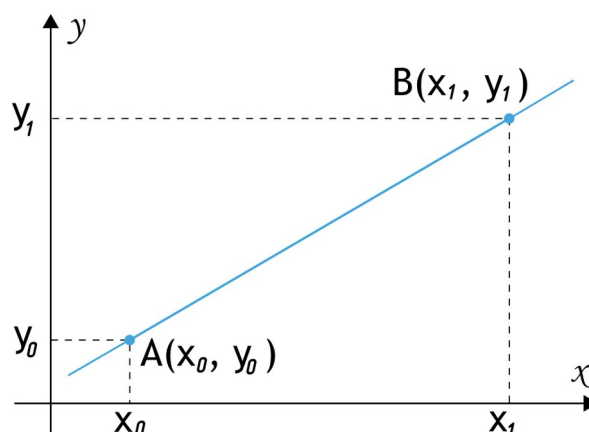
### 6.2.7 $X_0X_1 \rightarrow Y_0Y_1$ – линейное преобразование

**$x_0x_1 \rightarrow y_0y_1$**

Линейное преобразование – это преобразование входного значения узла в выходное на основе двух заданных точек с координатами  $(x_0, y_0)$  и  $(x_1, y_1)$ . По сути является линейной функцией, коэффициенты которой рассчитываются по координатам двух заданных точек. Значения  $x_0, y_0, x_1, y_1$  задаются в поле параметров узла.

На рисунке справа показано, как трактуются заданные координаты.

Коэффициенты  $k$  и  $b$  рассчитываются по формулам  $k = \frac{y_1 - y_0}{x_1 - x_0}$ ; и  $b = y_0 - k \cdot x_0$ .



## 6.3 Basic logic – логические функции и сдвиги

**Basic logic**

– Вкладка «Basic logic» содержит узлы логических функций, функций сдвига и выбора данных.

### 6.3.1 Lops – базовые логические операции

**Lops**

Логические операции над двумя операндами.

Выбор операции осуществляется из контекстного меню функций узла. В меню имеется два подменю: «Direct» и «Inverse». Если операция выбрана из меню «Direct», то выход узла обычный и операция производится без инверсии. Если операция выбрана из меню «Inverse», то выход узла помечается как инверсный и данные на выходе узла инвертируются.

Значения входов трактуются следующим образом. Если на входе число не равно нулю, то он считается лог.1. Если на входе нуль или «не-число», то он считается лог.0.

На выходе узла – всегда 0 или 1, в зависимости от результата операции.

Доступны логические функции.

N	Операция	Подменю	Описание
1	Лог. И (&)	Direct	Вых = Bx1 && Bx2
2	Лог. И-НЕ (&)	Inverse	Вых = !(Bx1 && Bx2)
3	Лог. ИЛИ (1)	Direct	Вых = Bx1     Bx2
4	Лог. ИЛИ-НЕ (1)	Inverse	Вых = !(Bx1     Bx2)
5	Лог. искл. ИЛИ (=1)	Direct	Вых = (!Bx1) ^ (!Bx2)
6	Лог. искл. ИЛИ-НЕ (=1)	Inverse	Вых = !((!Bx1) ^ (!Bx2))

### 6.3.2 ~1 – операция НЕ



Логическая операция НЕ – инверсия входа. Значения входа трактуется следующим образом. Если на входе число не равное нулю, то он считается лог.1. Если на входе нуль или «не-число», то он считается лог.0. На выходе узла – всегда 0 или 1, в зависимости от результата операции.

Значение выхода: Вых=!Вх.

### 6.3.3 RS – RS-триггер



Узел RS-триггер реализует функцию, аналогичную одноимённому электронному устройству – запоминание одного бита информации.

Узел имеет два входа. Вход R – установка 0; вход S – установка 1.

Состояние выхода зависит от состояния входов как показано в таблице ниже. Qt-1 – предыдущее состояние, то есть то состояние, которое было до изменения значений на входах узла.

Вх. S	Вх. R	Вых. Q	Примечание
0	0	Qt-1	Хранение

0	1	0	Установка 0
1	0	1	Установка 1
1	1	$Q_{t-1}$	Хранение

#### 6.3.4 RG – регистр хранения данных

**RG**

Узел RG – регистр хранения данных позволяет запоминать значение на выходе и хранить его в памяти узла. Узел RG реализует функцию, аналогичную одноимённому электронному устройству – регистру.

Узел RG имеет два входа – D и C. Вход D – это вход данных (вещественное число). Вход C – это входу управления. Выход Q у регистра один.

Если на входе управления C установлена лог.1, то состояние выхода Q равно состоянию входа D.

Если на входе управления C установлен лог.0, то на выходе Q установлены запомненные данные. Запоминание данных происходит при изменении входа C из состояния лог.1 в состояние лог.0.

Состояние выхода зависит от состояния входов как показано в таблице ниже.  $X_{t-1}$  – предыдущее состояние входа D, то есть то состояние, которое было до изменения значения входа C из состояния лог.1 в состояние лог.0.

Вх. D	Вх. C	Вых. Q	Примечание
X	1	X	Установка Q
X	0	$X_{t-1}$	Хранение

#### 6.3.5 Mux – мультиплексор

**Mux**

Узел Mux – мультиплексор – реализует функцию, аналогичную одноимённому электронному устройству – выбор данных. Узел имеет изменяемое количество входов данных от 2 до 16, которые названы  $i0..i15$ , один вход управления – S0 и один выход Q.

Количество входов задаётся в поле параметров узла, параметр Ni.



Узел передаёт данные входа номер S0 на выход Q. Т.е., если на входе S0 установлено число 0, то на выходе будут данные с входа i0; если на входе S0 установлено число 1, то на выходе будут данные с входа i1 и так далее.

Если число, установленное на входе S0 отрицательное или оно больше или равно Ni – то на выходе установится значение NaN (не-число).

### 6.3.6 BitS – преобразование двоичного знака и инверсии

**BitS**

Узел BitS выполняет две функции – побитовое инвертирование числа и преобразование знака в дополнительном обратном коде.

Узел имеет один вход и один выход. Число, поступающее на вход трактуется как 32-битное целое число. Затем над ним выполняется выбранная функция и её результат поступает на выход узла.

Выбор функции осуществляется из контекстного меню функций узла. В таблице представлены функции узла BitS.

N	Функция	Описание
1	INV	Побитовая инверсия всех 32 битов. Пример: 11110000111100001111000011110000 → INV → 00001111000011110000111100001111
2	CPL	Дополнение до 2х (изменение знака), т. е. инверсия всех бит плюс 1. Пример: 11110000111100001111000011110000 → CPL → 00001111000011110000111100010000

### 6.3.7 BitOps и BitOpsC – узлы побитовых операций и сдвигов

**BitOps**

Узлы побитовых операций и сдвигов имеют два входных параметра – A и B – и один выход.

**BitOpsC**

Узлы полностью аналогичны по функциям за исключением того, что узел «BitOps» имеет два входа – A и B, а узел «BitOpsC» имеет один вход A. Второй операнд – B – узла «BitOpsC» задаётся как константа в поле параметров узла. Выбор операции осуществляется из контекстного меню функций узла.

Числа, установленные на входах A и B трактуются как 32-битные целые числа.

Доступны следующие операции узлов BitOps и BitOps.

N	Операция	Описание
1	A and B	Побитовое И. Пример: 11110000111101101111010011110010 and 11110110111100001111010011110010 → 11110000111100001111010011110010
2	A or B	Побитовое ИЛИ. Пример: 11110000111101101111010011110010 or 11110110111100001111010011110010 → 11110110111101101111010011110010
3	A xor B	Побитовое искл. ИЛИ. Пример: 11110000111101101111010011110010 or 11110110111100001111010011110010 → 00000110000001100000000000000000
4	A>>B	Побитовый сдвиг вправо со знаком. Пример: 11110110111100001111010011110010 >> 2 → 1111101101111000011110100111100
5	A>>>B	Побитовый сдвиг вправо без знака. Пример: 11110110111100001111010011110010 >> 2 → 00111101101111000011110100111100
6	A<<B	Побитовый сдвиг влево. Пример: 11110110111100001111010011110010 >> 2 → 11011011110000111101001111001000

### 6.3.8 Bits Split – разбиение числа на биты

#### Bits Split

Узел «Bits Split» предназначен для разбиения целого числа или его части на биты. На вход узла поступает число, которое трактуется как 32-битное целое беззнаковое число.

Узел извлекает из этого числа заданное количество бит и выводит их на выходы – по одному биту на выход. Выходные значения – всегда 0 или 1.

В поле параметров узла задаются два параметра – First (номер первого бита, извлекаемого из числа) и Last (номер последнего бита извлекаемого из числа). Обратите внимание, что First всегда меньше либо равен Last!

Например, если надо извлечь 3 бита, начиная с 5-го, то надо установить First=5 и Last=7.

Количество и названия выходов узла устанавливаются в соответствии со значениями First и Last. Например, если First=5 и Last=7, то будет создано 3 выхода b5, b6, b7.

Значения First и Last лежат в диапазоне от 0 до 31 (номер бита).

### 6.3.9 Bits Join – объединение отдельных бит в число

#### Bits Join

Узел «Bits Join» предназначен для объединения отдельных бит в число, которое трактуется как 32-битное целое беззнаковое число.

На входы узла поступают значения заданных битов. Узел генерирует 32-битное целое беззнаковое число, в котором биты, начиная с бита номер First по бит номер Last устанавливаются в соответствии со входами узла.

В поле параметров узла задаются два параметра – First (номер первого бита, устанавливаемого в числе на выходе b32) и Last (номер последнего бита, устанавливаемого в числе на выходе b32). Обратите внимание, что First всегда меньше либо равен Last!

Например, если надо установить значение 5 бита, начиная с 12-го, то надо установить First=12 и Last=16.

Количество и названия входов узла устанавливаются в соответствии со значениями First и Last. Например, если First=12 и Last=16, то будет создано 5 выходов b12, b13, b14, b15, b16.

Значения First и Last лежат в диапазоне от 0 до 31 (номер бита).

Биты вне диапазона First..Last в числе на выходе b32 имеют значение 0.

### 6.4 One-bit IO – однобитный ввод-вывод

#### One-bit IO

– Вкладка «One-bit IO» содержит узлы однобитового ввода и вывода данных, такие как вход GPIO, выход GPIO, ШИМ.

#### 6.4.1 DInput – узел бинарного входа

#### DInput

Узел бинарного входа – DInput – позволяет вводить данных о состоянии заданной физической ножки модуля ESP. Узел имеет один выход, который может устанавливаться в состояние 0 или 1 в зависимости от состояния заданной физической ножки модуля ESP.

## ShloTiny5 – сделай сложности проще

Узел имеет два режима работы, выбираемые в поле параметров узла – прямой (Direct) и инверсный (Inverse).

В прямом режиме узел DInput устанавливает на своём выходе 1, если на указанном выводе GPIO присутствует напряжение высокого уровня (Vcc) и устанавливает на своём выходе 0, если на указанном выводе GPIO присутствует напряжение низкого уровня (0V).

В инверсном режиме узел DInput устанавливает на своём выходе 0, если на указанном выводе GPIO присутствует напряжение высокого уровня (Vcc) и устанавливает на своём выходе 1, если на указанном выводе GPIO присутствует напряжение низкого уровня (0V).

Выбор GPIO, к которому привязан узел DInput задаётся в меню привязки к выводам GPIO.

### 6.4.2 DOutput – узел бинарного выхода

**DOutput**

Узел бинарного выхода – DOutput – позволяет устанавливать состояние заданной физической ножки модуля ESP. Узел имеет один вход, который трактуется как лог.0 или лог.1.

Узел имеет два режима работы, выбираемые в поле параметров узла – прямой (Direct) и инверсный (Inverse).

В прямом режиме узел DOutput устанавливает на указанном выводе GPIO напряжение высокого уровня (Vcc), если на входе узла установлен лог.1 и устанавливает на указанном выводе GPIO, напряжение низкого уровня (0V), если на входе узла установлен лог.0.

В инверсном режиме узел DOutput устанавливает на указанном выводе GPIO напряжение высокого уровня (Vcc), если на входе узла установлен лог.0 и устанавливает на указанном выводе GPIO, напряжение низкого уровня (0V), если на входе узла установлена лог.1.

Выбор GPIO, к которому привязан узел DOutput задаётся в меню привязки к выводам GPIO.

### 6.4.3 PWM – широтно-импульсный модулятор (ШИМ)

**PWM**

Узел ШИМ – PWM – позволяет генерировать ШИМ-сигнал на заданной физической ножке модуля ESP. Частота ШИМ по умолчанию – 1000Гц. Частота может быть изменена использованием узла PWM Freq (см. ниже).

На вход узла PWM подаётся значение – число с плавающей точкой в диапазоне от 0 до 1. Отрицательные числа трактуются как 0. Числа больше 1 трактуются как 1.

Узел имеет два режима работы, выбираемые в поле параметров узла – прямой (Direct) и инверсный (Inverse).

В прямом режиме число на входе узла устанавливает заполнение импульса напряжением высокого уровня (Vcc).

В инверсном режиме число на входе узла устанавливает заполнение импульса напряжением низкого уровня (0V).

### 6.4.4 PWM freq – задать частоту ШИМ

#### PWM freq

Узел PWM freq позволяет задать частоту ШИМ, которую используют узлы PWM. Узел не имеет ни входов ни выходов. Частота ШИМ задаётся в поле параметров узла.

Узел можно использовать на схеме только в одном экземпляре.

### 6.4.5 Encoder – энкодер (волкодер), датчик угла поворота

#### Encoder

Энкодер (волкодер), он же – датчик угла поворота (ДУП) – это устройство для преобразования угловых положений или линейных перемещений в цифровой сигнал, т.е. энкодер – это датчик угла или линейного перемещения, соответственно есть крутильные и линейные энкодеры.

Принцип работы энкодера заключается в преобразовании механического перемещения в электрические сигналы. У инкрементального энкодера, который мы будем рассматривать, эти сигналы представляют собой два прямоугольных сигнала, сдвинутых по фазе на 90 градусов.



Внешний вид энкодера для ручного управления показан на рисунке справа.

Энкодер имеет две сигнальные линии – **Sa** и **Sb**. Эти линии подключаются к выводам ESP8266, задаваемым в меню привязки к выводам GPIO.

Узел **Encoder** имеет один выход (значение на нём будем называть **Out**).

В зависимости от настройки в поле параметров узла **In**, узел **Encoder** может не иметь входов, или иметь вход **Init**, вход **Rst** или оба эти входа сразу.

В поле параметров узла задаются следующие параметры:

**Init** – начальное значение на выходе узла при старте схемы при отсутствии входа **Init** (см. поле параметров **In**). Начальное значение на

## ShloTiny5 – сделай сложности проще

выходе **Out** после запуска схемы определяется входом **Init**, если он есть или полем параметров узла **Init**, если вход **Init** отсутствует.

**Step** – шаг. При повороте вала энкодера на один шаг значение выхода **Out** увеличивается или уменьшается на значение **Step** в зависимости от направления вращения.

**Min** – минимальное значение на выходе узла (см. поле параметров **Mode**).

**Max** – максимальное значение на выходе узла (см. поле параметров **Mode**).

### Замечание!

Поля **Step**, **Min**, **Max**, **Init** могут иметь целое, дробное, положительное или отрицательное значения!

Это удобно для пересчёта количества импульсов в нужную величину.

Например, для изменения направления счета, достаточно изменить знак у параметра **Step**.

**Mode** – режим ограничений. Имеется 4 режима, как узел обрабатывает значение **Min** и **Max**. Их описание приведено в таблице ниже.

№	Значение поля Mode	Описание режима
1	Rotate	циклическое изменение. То есть если значение стало больше, чем <b>Max</b> , то оно устанавливается в <b>Min</b> , а если значение стало меньше, чем <b>Min</b> , то оно устанавливается в <b>Max</b> . (Прошли круг и начали сначала).
2	Fix	фиксация минимума и максимума. Если значение стало больше, чем <b>Max</b> , то оно устанавливается в <b>Max</b> , а если значение стало меньше, чем <b>Min</b> , то оно устанавливается в <b>Min</b> . (Упираемся в ограничители слева и справа).
3	FixMin	фиксация минимума. Если значение стало меньше, чем <b>Min</b> , то оно устанавливается в <b>Min</b> . Но если значение стало больше, чем <b>Max</b> , то оно устанавливается в <b>Min</b> . (Упираемся в ограничитель только слева).
4	FixMax	фиксация максимума. Если значение стало меньше, чем <b>Min</b> , то оно устанавливается в <b>Max</b> . Но если значение стало больше, чем <b>Max</b> , то оно устанавливается в <b>Max</b> . (Упираемся в ограничитель только справа).

## ShloTiny5 – сделай сложности проще

**In** – определяет наличие управляющих входов. Их описание приведено в таблице ниже.

№	Значение поля In	Описание режима
1	No	Управляющих входов нет. Начальное значение задаётся полем Init.
2	Init	Управляющий вход Init. Начальное значение задаётся входом Init. См. примечание ко входу Init.
3	Rst	Управляющий вход Rst. При лог.1 на этом входе – значение выхода Out устанавливается в начальное, определяемое полем Init. При лог.0 на этом входе идёт счёт импульсов энкодера.
4	Init+Rst	Управляющие вход Init и Rst одновременно. При лог.1 на входе Rst – значение выхода Out устанавливается в начальное, со входа Init. При лог.0 на входе Rst идёт счёт импульсов энкодера.

Ниже приведено важное примечание к тому, как ведёт себя вход Init.

### Вход Init.

Пока на входе Init узла установлено значение NaN (не число), на выходе узла также будет установлено значение NaN и счет не ведётся.

Как только на входе Init будет установлено числовое значение, то узел Encoder будет проинициализирован и в дальнейшем значение входа Init игнорируется до перезапуска схемы или до появления лог.1 на входе Rst.

Если в момент появления лог.1 на входе Rst на входе Init узла установлено значение NaN, то узел Encoder вновь переходит в неинициализированное состояние.

Это удобно для начальной установки, например, с помощью приёма начального значения по MQTT.

## 6.5 Analog-to-Digital – аналого-цифровой преобразователь

### Analog-to-digital

– Вкладка «Analog-to-digital» содержит содержит узел встроенного АЦП.



Узел ADC позволяет вводить значение напряжения на входе встроенного в модуль ESP аналого-цифрового преобразователя (АЦП). Допустимое входное напряжение встроенного АЦП лежит в диапазоне 0–1В.

Если на входе АЦП фиксируется напряжение больше 1В, то узел ADC выставляет на своём выходе значение NaN (не число).

Узел ADC имеет функции линейного преобразования входных данных и параметр «точность».

Параметр точность показывает на сколько процентов должны измениться данные на входе АЦП, чтобы на выходе узла произошло изменение данных. Это позволяет устанавливать необходимую точность. Параметр точности называется «р» и задаётся в диапазоне 0–100% в поле параметров узла.

Измеренный уровень напряжения преобразуется по линейной формуле  $out = k \cdot u_{ex} + b$ , где  $out$  – значение, устанавливаемое на выходе узла ADC;  $u_{ex}$  – значение входного напряжения в диапазоне 0–1В,  $k$  и  $b$  – произвольные коэффициенты, задаваемые в поле параметров узла.

### 6.6 Data Transfer – обмен данными по сети

#### Data Transfer

– Вкладка «Data Transfer» содержит узлы, отвечающие за обмен данными по сети.

#### 6.6.1 Mqtt Config – настройка соединения с MQTT-брокером

##### Mqtt Config

Узел «MQTT Config» предназначен для настройки соединения с MQTT-сервером (брокером). Этот узел не имеет входов и выходов. Другие узлы MQTT недоступны пока узел «Mqtt Config» не вставлен в схему.

Параметры, которые позволяет задать узел:

Prefix – префикс MQTT-параметров (см. ниже);

Тип протокола (сейчас только mqtt);

Адрес сервера (URL или IP);

Порт сервера;

User – имя пользователя для авторизации (\_ – не задано);

Pass – пароль для авторизации (\_ – не задан).

Узел имеет один выход «сop». Этот выход показывает наличие связи с MQTT-сервером. Если связь есть – то на выходе «сop» установлено значение 1. Если связь отсутствует – то на выходе «сop» установлено значение 0.

Авторизация на USER/PASSWORD работает без SSL.



Префикс MQTT-параметров в ShloTiny5 всегда начинается и заканчивается символами «слэш» (/).

Если в префиксе MQTT-параметров нет символа «слэш», то редактор добавляет его автоматически. Таким образом, не бывает «пустого» префикса. Он всегда содержит как минимум один символ «слэш».

### 6.6.2 *MqttPub* – публикация данных на MQTT-брокере

#### MqttPub

Узел «MqttPub» предназначен для публикации данных на MQTT-сервере (брокере). Настройка соединения с MQTT-сервером производится с помощью узла «MQTT Config».

Узел имеет один вход и трактуется как число с плавающей точкой. Также имеются специальные значения: NaN (не-число) и Inf (бесконечность).

Название публикуемого параметра без префикса (topic в терминологии MQTT) вводится в поле параметров узла.

Параметр публикуется под названием /<prefix>/<topic>.

<prefix> – строка, задаваемая при настройке узла «MQTT Config».

<topic> – задаётся в поле параметров узла.

Узел «MqttPub» имеет три режима публикации, см. таблицу ниже.

Режим выбирается в контекстном меню в поле параметров узла. В режиме «Short» публикуется только значение, поступающее на вход в виде текстовой строки. Это может быть целое число, число с плавающей точкой, или специальные значения – inf (бесконечность) и nan – (не-число).

В режимах «Stamp» и «Full» публикуется строка в формате JSON, содержащая несколько полей.

"v" – входное значение данного узла «MqttPub». Строка, содержащая число с плавающей точкой, или специальные значения – "inf" (бесконечность) или "nan" – (не-число).

"t" – временная метка. Если время узла не установлено (см. пп. 6.10.2 , «Set Time» ), то поле "t" имеет значение ноль. Временная метка устанавливается в момент последнего обновления данных на входе узла.

"name" – родное, заводское, название данного узла в формате esp\_8266\_<id чипа>.

"alias" – псевдоним имени узла, т.е. название узла, которое ему дал пользователь (см. 6.1.2 «Device settings»). Если псевдоним не задан – то поле "alias" содержит пустую строку.

№	Режим	Формат	Комментарий
1	Short	Текст	Публикуется только строка, содержащая число, inf или nan
2	Stamp	JSON	Публикуется JSON-строка формата: <pre>{   "v": "&lt;число/inf/nan&gt;",   "t": &lt;временная метка&gt; }</pre>
3	Full	JSON	Публикуется JSON-строка формата: <pre>{   "v": "&lt;число/inf/nan&gt;",   "t": &lt;временная метка&gt;,   "name": "&lt;название узла (заводское)&gt;"   "alias": "&lt;псевдоним узла&gt;" }</pre>

### 6.6.3 MQTTdesc – получение данных от MQTT-брокера

#### MqttDesc

Узел «MQTTdesc» предназначен для подписки на данные MQTT-сервера (брокера). Настройка соединения с MQTT-сервером производится с помощью узла «Mqtt Config». Узел имеет один выход «v» или два выхода – «v» и «t» в зависимости от режима получения данных, см. ниже. Название параметра без префикса, на который узел будет подписан (topic в терминологии MQTT) вводится в поле параметров узла.

Подписка оформляется на параметр под названием `/<prefix>/<topic>`.

`<prefix>` – строка, задаваемая при настройке узла «Mqtt Config».

`<topic>` – задаётся в поле параметров узла.

Узел «MQTTdesc» имеет два режима получения данных, см. таблицу ниже.

Режим выбирается в контекстном меню в поле параметров узла.

В режиме «Short» считается, что от MQTT брокер присылает только значение, в виде текстовой строки. Это может быть целое число, число с плавающей точкой, или специальные значения – inf (бесконечность) и nan – (не-число).

В режиме «Stamp» считается, что от MQTT брокер присылает строку в формате JSON, содержащая несколько полей.

## ShloTiny5 – сделай сложности проще

"v" – строка, содержащая число с плавающей точкой, или специальные значения – "inf" (бесконечность) или "nan" – (не-число). Это значение устанавливается на выходе "v" узла «MQTTdesc».

"t" – число-временная метка. Это значение устанавливается на выходе "t" узла «MQTTdesc».

№	Режим	Формат	Комментарий
1	Short	Текст	Узел имеет один выход «v». Брокер присылает строку, содержащую число, inf или nan
2	Stamp	JSON	Узел имеет два выхода «v» и «t». Брокер присылает JSON-строку формата: { "v": "<число/inf/nan>", "t": <временная метка> }

### Важное замечание!

На выходе «v» узла MQTTdesc по умолчанию установлено значение NaN (не-число). Это значение будет на выходе данного узла до тех пор, пока с сервера/брокера не считается опубликованное там значение темы.

Это позволяет определить, что данный узел MQTTdesc принял или не принял значение с сервера/брокера.

## 6.7 Delay, timers – таймеры, задержки, счётчики, сумматоры

### 6.7.1 Timer и Timer(t) – таймеры с заданным периодом

#### Timer

#### Timer(t)

Узлы Timer и Timer(t) являются таймерами, генерирующими последовательность 010101.. с заданным периодом. Период задаётся в децисекундах (0.1сек). То есть число 10 соответствует 1секунде. Функции узлов аналогичны, но в узле Timer период импульсов задаётся как константа в поле параметров узла, а для узла Timer(t) – период импульсов принимается со входа t. Поэтому период импульсов таймера Timer(t) можно изменять. Кроме того, если на вход t подать 0, то таймер остановится.

### 6.7.2 TimerFunc и TimerFunc(t) – функции задержек, импульсов

TimeFunc

TimeFunc(t)

Узлы TimerFunc и TimerFunc(t) содержат ряд функций задержек и формирования импульсов. Время задержки или длительность импульса задаётся в децисекундах (0.1сек).

Функции узлов аналогичны, но в узле TimerFunc время задержки или длительность импульса задаётся как константа в поле параметров узла, а для узла TimerFunc(t) – время задержки или длительность импульса принимается со входа t., а данные – со входа D.

Узел TimerFunc имеет только один вход данных.

Узел TimerFunc(t) имеет два входа – D (вход данных) и t (длительность задержки или импульса).

Вход данных этих узлов трактуется как логический (0 или 1).

Значения входа данных NaN и 0 трактуются как 0. Остальные значения входа данных трактуются как 1.

Выбор функции осуществляется в меню функций узла. В узлах TimerFunc и TimerFunc(t) доступны следующие функции.

N	Функция	Описание
1	Delay 0	Задержка 0. При изменении входа данных 0→1 без задержки на выходе устанавливается 1. При изменении входа данных 1→0 на выходе устанавливается 0 в случае, если состояние 0 входа продолжалось не менее времени t.
2	Delay 1	Задержка 0. При изменении входа данных 1→0 на выходе без задержки устанавливается 0. При изменении входа данных 0→1 на выходе устанавливается 1 в случае, если состояние 1 входа продолжалось не менее времени t.
3	Delay 0/1	При изменении входа данных 0→1 на выходе устанавливается 1 в случае, если состояние 1 входа продолжалось не менее времени t. При изменении входа данных 1→0 на выходе устанавливается 0 в случае, если состояние 0 входа продолжалось не менее времени t.
4	Pulse	Формирователь импульса. При изменении входа данных 0→1 на выходе устанавливается 1 и эта 1 удерживается на выходе время t, затем выход автоматически переходит в состояние 0 до следующего изменения входа данных

		0→1.
5	Actual	Проверка актуальности данных. При любом событии на входе данных, на выходе устанавливается 1 и удерживается там не менее времени $t$ . Если за время $t$ не происходило событий, то на выходе устанавливается 0.

### 6.7.3 Counter – счётчик с ограничением и сбросом

#### Counter

Узел Counter представляет собой универсальный счётчик. Он имеет два режима – увеличивающийся (Counter+) счётчик и уменьшающийся (Counter-) счётчик. Выбор режима счётчика осуществляется в меню функций узла.

Счётные импульсы поступают на счётный вход «C» счётчика. Счётный вход данных этих узлов трактуется как логический (0 или 1). Значения входа «C» NaN и 0 трактуются как 0. Остальные значения входа «C» трактуются как 1. Счётный импульс – это изменение входа «C» 0→1.

Поле параметров узла содержит два поля: «Max» и «Rst». Параметр «Max» задаёт максимальное значение счётчика. Параметр «Rst» определяет наличие входа сброса счётчика.

В режиме «Counter+» каждый импульс вызывает увеличение значения на выходе счётчика на 1. При превышении максимального значения «Max» на выходе счётчика устанавливается 0 и счёт начинается снова.

В режиме «Counter-» каждый импульс вызывает уменьшение значения на выходе счётчика на 1. При достижении нуля на выходе счётчика, следующий импульс вызывает установку счётчика в значение «Max» и счёт начинается снова.

Если разрешён вход сброса «Rst», то этот вход управляет установкой счётчика в 0. Пока на входе сброса «Rst» установлена лог.1, на выходе счётчика установлено значение 0 и счётчик не считает. Если на входе сброса «Rst» установлен лог.0, то счётчик считает, начиная с 0.

## 6.8 Barriers and filters – барьеры и фильтры событий

#### Barriers and filters ...

– Вкладка «Barriers and filters» содержит содержит узлы, отвечающие за прохождение событий.

### 6.8.1 B(t) и B(tin) – временный барьер прохождения событий

#### B(t)

Узлы B(t) и B(tin) запрещают прохождение событий на заданное время после приёма очередного события на вход данных. Функции

#### B(tin)

узлов одинаковы, но в узле  $B(t)$  время запрета прохождения событий задаётся как константа в поле  $t$  параметров узла, а для узла  $TimerFunc(t)$  – время запрета прохождения событий принимается со входа  $T$ , а данные – со входа  $D$ .

При приёме очередного события на вход данных начинается отсчёт времени  $T$ . В течении этого времени события со входа данных не передаются на выход. По истечении времени запрета прохождения событий узел вновь передаёт очередное событие со входа данных на выход.

### 6.8.2 *EmitEv* – формирование события

**EmitEv**

Узел *EmitEv* предназначен для формирования событий. Он имеет два входа: вход данных  $D$  и вход разрешения  $C$ . Если на входе разрешения  $C$  установлен лог.0, то события со входа  $D$  не передаются на выход. Если на входе разрешения  $C$  установлена лог.1, то события со входа  $D$  свободно передаются на выход.

При изменении входа  $C$   $0 \rightarrow 1$  происходит формирование события на выходе вне зависимости от наличия события на входе, при этом данные берутся со входа  $D$ .

### 6.8.3 *BD(cond)* и *BE(cond)* – узлы условного прохождения событий

**BD(cond)**

**BE(cond)**

Узлы *BD(cond)* и *BE(cond)* предназначены для разрешения и запрета прохождения событий в зависимости от выполнения условия. Условие выбирается в меню функций узла. Узлы имеют по три входа. Входы  $A$  и  $B$  предназначены для проверки условия. Вход данных  $D$  предназначен для приёма событий. Вход данных узла *BD(cond)* обозначен как  $Dd$ . Вход данных узла *BE(cond)* обозначен как  $De$ .

Узел *BD(cond)* пропускает события, если выполняется заданное условие.

Узел *BE(cond)* запрещает события, если выполняется заданное условие.

Разрешение или запрет прохождения событий в зависимости от узла и выполнения выбранного условия приведены в таблице.

N	Условие	BD(cond)	BD(cond)	BE(cond)	BE(cond)
		выполняется	не выполняется	выполняется	не выполняется
1	$A=B$	разрешение	запрет	запрет	разрешение

2	$A \neq B$	разрешение	запрет	запрет	разрешение
3	$A > B$	разрешение	запрет	запрет	разрешение
4	$A < B$	разрешение	запрет	запрет	разрешение
5	$A \geq B$	разрешение	запрет	запрет	разрешение
6	$A \leq B$	разрешение	запрет	запрет	разрешение

#### 6.8.4 BE(cs) – узел запрета прохождения специальных событий

**BE(cs)** Узел запрета прохождения специальных событий BE(cs) предназначен для запрета прохождения событий со входа узла на его выход, если событие имеет заданное значение данных – «не-число» или «бесконечность».

Тип специального события выбирается в меню функций узла.

Если выбрана функция Be(NaN), то запрещается прохождение событий имеющих в данных значение NaN (не-число).

Если выбрана функция Be(Inf), то запрещается прохождение событий имеющих в данных значение Inf (бесконечность).

### 6.9 Net Services – состояние сети

#### Net services

– Вкладка «Net Services» содержит узлы, отвечающие за контроль состояния сети.

#### AP connect?

Узел проверки «AP connect?» предназначен для проверки наличия соединения модуля ESP с точкой доступа по сети WiFi.

Узел имеет один выход, который устанавливается в 0, если нет связи с точкой доступа и в 1, если связь с точкой доступа есть.

### 6.10 Date/Time – работа с датой и временем

#### Date/Time

– Вкладка «Date/Time» содержит узлы для работы с датой и временем.

Время в системе ShloTiny5 представлено в виде «временной метки» – timestamp, представляющей собой целое число секунд, отсчитываемое от 00 часов 00 минут 1 января 1970 года.

Время в системе ShloTiny5 считается всегда как «всемирное координированное время» – UTC+00.

Ниже приведён список часовых поясов различных регионов России.

UTC+2 (МСК-1) – Калининградская область

UTC+3 (МСК, московское время) – большая часть европейской территории России и вся Российская железная дорога

UTC+4 (МСК+1) – Удмуртская Республика, Астраханская область, Самарская область, Саратовская область и Ульяновская область

UTC+5 (МСК+2) – Республика Башкортостан, Пермский край, Курганская область, Оренбургская область, Свердловская область, Тюменская область, Челябинская область, Ханты-Мансийский автономный округ – Югра и Ямало-Ненецкий автономный округ

UTC+6 (МСК+3) – Омская область

UTC+7 (МСК+4) – Республика Алтай, Республика Тыва, Республика Хакасия, Алтайский край, Красноярский край, Кемеровская область – Кузбасс, Новосибирская область и Томская область

UTC+8 (МСК+5) – Республика Бурятия и Иркутская область

UTC+9 (МСК+6) – Республика Саха (Якутия) (западные и центральные районы), Забайкальский край и Амурская область

UTC+10 (МСК+7) – Республика Саха (Якутия) (ряд районов), Приморский край, Хабаровский край и Еврейская автономная область

UTC+11 (МСК+8) – Республика Саха (Якутия) (северо-восточные районы), Магаданская область и Сахалинская область

UTC+12 (МСК+9) – Камчатский край и Чукотский автономный округ

### 6.10.1 Get Time – получить системное время

#### Get Time

Узел Get Time предназначен для получения временной метки с заданным в часах смещением.

Значение на выходе узла равно UTC+<смещение в часах>.

Если с выхода узла получен 0, то системное время не установлено!

Смещение задаётся в поле параметров узла в диапазоне -12...+12. Это позволяет получать временную метку местного времени.

Например, чтобы получить московское время – надо задать смещение 3. Чтобы получить новосибирское время – надо задать смещение 7. А чтобы получить время Лос-Анджелеса – надо задать смещение -7.



### 6.10.2 *Set Time – установить системное время*

#### Set Time

Узел Set Time позволяет установить системное время. Узел имеет один вход и трактует его значение как временную метку UTC+0. При получении на вход узла данных устанавливается системное время в соответствии с полученной временной меткой.

### 6.10.3 *NTP Time – получить время с NTP-сервера*

#### NTP Time

Узел NTP Time позволяет получить время с заданного NTP-сервера. Узел имеет один выход, на котором устанавливается временная метка полученного с сервера времени. До получения времени с сервера на выходе узла установлен 0.

Адрес NTP-сервера устанавливается в поле параметров узла в формате <url узла>,[период опроса в минутах].

Например, если в поле параметров задано pool.ntp.org,60 – это означает, что время будет считываться с NTP-сервера pool.ntp.org каждые 60 минут.

Период опроса может быть не задан, тогда он устанавливается равным 60 минут. Т.е. вместо «pool.ntp.org,60» можно задать «pool.ntp.org».

### 6.10.4 *Split Time – извлечь из временной метки дату и время*

#### Split Time

Узел Split Time «разбивает» временную метку на составляющие её дату и время. Узел имеет один вход – временной метки и семь выходов:

day – число месяца,

month – номер месяца,

year – год,

hour – часы,

min – минуты,

sec – секунды,

wday – номер день недели.

При изменении данных на входе (получении события) – данные на выходах пересчитываются в соответствии с ней.

### 6.10.5 *Synth Time – составить временную метку из даты и времени*

#### **Synth Time**

Узел Synth Time позволяет получить временную метку по заданной дате и времени. Узел имеет один выход – временную метку и шесть входов:

day – число месяца,

month – номер месяца,

year – год,

hour – часы,

min – минуты,

sec – секунды.

При изменении данных на любом входе (получении события) – временная метка на выходе изменяется в соответствии с полученными данными.

### 6.11 *1-wire-bus chip – устройства с однопроводным интерфейсом*

#### **1-wire-bus chip**

– Вкладка «1-wire-bus chip» содержит узлы для работы с различными датчиками, имеющими однопроводные интерфейсы.

#### 6.11.1 *DHT – однопроводные датчики температуры и влажности.*

#### **DHT**

Узел DHT позволяет считывать данные с нескольких типов датчиков: DHT11, DHT22, AM2302, RHT03. Эти датчики измеряют два параметра – температуру и влажность воздуха.

Тип датчика задаётся в поле параметров узла. Если задать тип DHT(auto), то узел попытается определить тип подключённого датчика автоматически.

Ножка модуля ESP, к которой подключён датчик задаётся в меню привязки к выводам GPIO.

Входов у узла нет.

Узел имеет два выхода Т (температура) и Н (влажность воздуха).

#### 6.11.2 *DS182x – датчик температуры*

#### **DS182x**

Узел DS182x позволяет считывать данные с трёх типов датчиков – DS18B20, DS18S20 и DS1822. Тип датчика определяется автоматически.

## ShloTiny5 – сделай сложности проще

Обмен данными с этими датчиками ведётся по однопроводной шине OneWire.

Ножка модуля ESP, к которой подключён датчик задаётся в меню привязки к выводам GPIO.

Входов у узла нет. Узел имеет один выход T (температура).

К одной шине можно подключать от 1 до 16 датчиков.

Датчики могут быть разных типов вперемешку или одного типа.

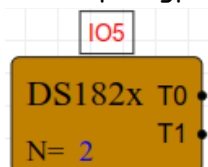
Количество датчиков (и выходов узла DS182x), подключённых к шине задаётся параметром N в поле параметров узла.

Количество датчиков, подключённых к шине должно быть равно тому, которое задано в поле параметров узла. Если этих датчиков меньше, то все выходы узла DS182x будут установлены в значение NaN (не число).

Порядок привязки датчиков к выходам узла  $T_0 - T_N$  определяется автоматически по уникальному 64-битному идентификатору датчика.

При замене одного датчика на другой – порядок привязки меняется непредсказуемым образом.

Идентификаторы и привязки датчиков можно посмотреть на странице «ShloTiny Info» в разделе «Used GPIOs». Например мы подключили два датчика температуры к линии GPIO5.



Слева показан узел DS182x в редакторе EIDraw, на котором, в поле параметров узла, указано количество датчиков **N=2**. В поле привязки линий ввода-вывода, разумеется, указана линия **IO5** (она же GPIO5).

### GPIO 5: (DS182x):

- 0: 26.56C Id: 96000802ab287110
- 1: 27.37C Id: f80300a279931f28

На странице «ShloTiny Info» в разделе «Used GPIOs» будет отображаться (см. рис. Слева): номер выхода, к которому привязан данный датчик, значение считанной с этого

датчика температуры и его уникальный 64-битный идентификатор.

Помните, что линия данных шины OneWire должна быть подтянута к линии питания Vcc через резистор сопротивлением 4.7-5.1 кОм. Если этого не сделать, то обмен по шине будет нестабильным или вообще невозможным.

## 6.12 I2C bus chip – устройства на шине I2C

**I2C bus chip** – Вкладка «I2C bus chip» содержит узлы для работы с различными устройствами, подключаемыми к двупроводной шине I2C.

Особенностью данной шины является то, что устройства имеют уникальный адрес, позволяющий на одну шину подключить несколько устройств одного или разных типов. Чтобы подключить устройства на I2C-шину, необходимо сначала сконфигурировать эту шину с помощью узла «I2C bus».

### 6.12.1 I2C bus – конфигуратор шины I2C

**I2C bus** Узел I2C bus служит для конфигурации шины I2C. Он не имеет входов и выходов. Ножки модуля ESP, к которым подключены сигналы SDA и SCL шины I2C, задаются в меню привязки к выводам GPIO.

Пока данный узел не установлен на схему – другие узлы вкладки I2C bus невозможно использовать.

Если данный узел удалить со схемы, то другие узлы вкладки I2C bus, ранее установленные в схему также удаляться.

### 6.12.2 BMP180, BMP280 – датчики температуры и давления окружающей среды

**BMP180** Узлы BMP180 и BMP280 служат для получения значений температуры и давления окружающей среды с одноимённых датчиков. Датчики подключаются по шине I2C. Функции узлов идентичны.

**BMP280** Узлы BMP180 и BMP280 имеют по два выхода – выход «Т» – значение температуры и выход «Р» – значение давления. При отсутствии связи с датчиком на всех его выходах устанавливается значение NAN – не число.

Датчик BMP180 имеет один фиксированный адрес на шине I2C – 0x77. Поэтому к шине I2C может быть подключено одновременно не более одного датчика BMP180.

Датчик BMP280 имеет возможность выбора адреса на шине I2C – 0x76 или 0x77. Поэтому может быть подключено к шине I2C одновременно не более двух датчиков BMP280 с разными адресами.

### **6.12.3 BME280 – датчики влажности, температуры и давления окружающей среды**

#### **BME280**

Узел BME280 служит для получения значений влажности, температуры и давления окружающей среды с одноимённого датчика.

Узел BME280 имеет три выхода – выход «Т» – значение температуры, выход «Р» – значение давления и выход «Н» – значение влажности. При отсутствии связи с датчиком на всех его выходах устанавливается значение NAN – не число.

Датчик BME280 имеет возможность выбора адреса на шине I2C – 0x76 или 0x77. Поэтому может быть подключено к шине I2C одновременно не более двух датчиков BME280 с разными адресами.

### **6.12.4 BH1750 – датчик освещённости**

#### **BH1750**

Узел BH1750 служит для получения значения освещённости с датчика BH1750. Датчик подключается по шине I2C. Датчик имеет один из адресов на выбор – 0x23 или 0x5C. Поэтому может быть подключено одновременно не более двух датчиков с разными адресами.

Узел имеет один выход – уровень освещённости.

Адрес датчика задаётся в меню выбора адреса устройства.

### **6.12.5 DS1307 – часы реального времени**

#### **DS1307**

Узел DS1307 служит для работы с энергонезависимой микросхемой-часами реального времени DS1307 и совместимыми с ней. Микросхема DS1307 предназначена для непрерывного отсчёта реального времени с учётом даты.

Батарейное питание микросхемы DS1307 позволяет отсчитывать время при отключённом основном питании ESP-модуля.

Узел DS1307 имеет один вход установки времени –  $t_i$  и один выход считывания временной метки –  $t_o$ .

В поле параметров узла задаётся параметр Т – время опроса часов реального времени DS1307. Это время задаётся в минутах в диапазоне от 1 мин до 1440мин. По умолчанию часы реального времени опрашиваются один раз в 60мин.

## ShloTiny5 – сделай сложности проще

Для установки времени в микросхеме DS1307, на вход **ti** узла поступает временная метка в формате UNIX-time. Если метка равна нулю, то она игнорируется!

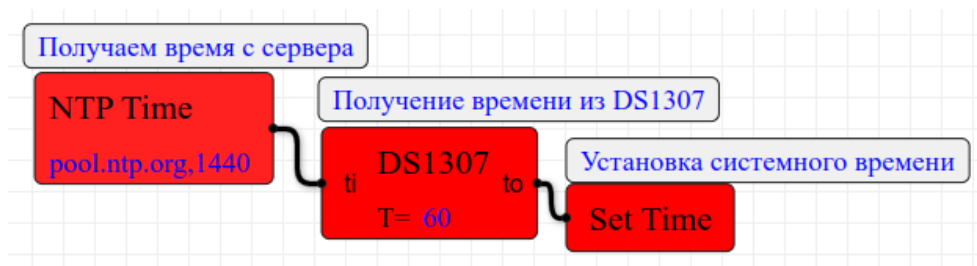
При успешной установке времени, новое значение времени в формате UNIX-time выставляется на выходе **to**.

Далее узел опрашивает микросхему с периодом, заданным параметром **T**.

Если связь с микросхемой по интерфейсу I2C не удалось установить, то на выходе **to** выставляется значение 0.

Для того, чтобы синхронизировать время в системе с NTP-сервера и отсчитывать его вне зависимости от наличия связи с NTP-сервером и наличием основного питания ESP-модуля, рекомендуется использовать схему, показанную ниже.

Разумеется, NTP-сервер и периоды опроса сервера и микросхемы DS1307 можно задать другие, какие нужно.



### 6.12.6 SCD30 – датчик содержания CO2, температуры и влажности

#### SCD30

Узел SCD30 служит для получения значений концентрации CO2, влажности и температуры с одноимённого датчика состояния окружающей среды.

Узел SCD30 имеет три выхода – выход «CO2» – значения концентрации CO2, выход «Т» – значение температуры и выход «Н» – значение относительной влажности. При отсутствии связи с датчиком на всех его выходах устанавливается значение NAN – не число.

Датчик SCD30 имеет фиксированный адрес на шине I2C – 0x61.

Значение концентрации CO2 измеряется в PPM (миллионных долях), температуры – в градусах Цельсия и относительной влажности – в процентах.

### 6.13 7-seg LEDS – управление семисегментными индикаторами, шкалы

**7-seg LEDS** – Вкладка «7-seg LEDS» содержит узлы для работы с контроллерами 7-сегментных индикаторов и узлы формирования данных для этих индикаторов.

Существует возможность работы с индикаторами как с набором отдельных светодиодов, как с набором индикаторов для вывода цифр или как с линейной шкалой.

#### 6.13.1 TM1637 – контроллер 6-разрядного семисегментного индикатора

**TM1637** Узел TM1637 предназначен для управления контроллером 6-разрядного 7-сегментного индикатора типа TM1637. Данный контроллер подключается по двум проводам – DIO и CLK. Ножки модуля ESP, к которым подключены сигналы DIO и CLK, задаются в меню привязки к выводам GPIO.

У узла есть входы данных **d0...d5** – их количество соответствует количеству разрядов индикатора и задаётся параметром **Digits** – от 1 до 6.

Также имеется вход **br** – управление яркостью. Этот вход появляется только в том случае, если параметр **Bright** установлен в значение «Input».

Узел TM1637 представляет каждый индикатор как набор 8-битов. Соответствие битов и сегментов индикатора представлено в таблице ниже.

	№ бита	Сегмент	BIN	HEX	DEC
	0	A	0b00000001	0x01	1
	1	B	0b00000010	0x02	2
	2	C	0b00000100	0x04	4
	3	D	0b00001000	0x08	8
	4	E	0b00010000	0x10	16
	5	F	0b00100000	0x20	32
	6	G	0b01000000	0x40	64
	7	H	0b10000000	0x80	128

Таким образом, задавая нужную комбинацию бит на любом входе – d0..d5 – узла TM1637 можно получить любой заданный символ – цифру, букву или спецсимвол.

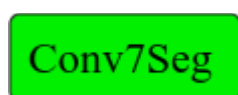
В меню параметров узла задаются несколько режимов работы узла.

**Digits** – задаёт количество используемых разрядов (символов) индикатора от 1 до 6.

**Dot** – режим вывода точки (сегмент H). Если Dot=«Input» – данные о состоянии точки берутся с соответствующего входа (бит 7). В режимах «1» – «6» положение точки жёстко фиксировано и она выводится всегда в составе разряда с соответствующим номером.

**Bright** – режим яркости. Числа от 0 до 7 задают яркость (0-индикатор отключён, 7 – яркость индикатора на максимуме). Если Bright=«Input», то появляется отдельный вход для управления яркостью. Значение на этом входе – число с плавающей точкой в диапазоне 0...1. (0-индикатор отключён, 1 – яркость индикатора на максимуме).

### 6.13.2 Conv7Seg – преобразователь чисел в коды 7-сегментного индикатора



Узел Conv7Seg служит для преобразования произвольного числа в код 7-сегментного индикатора с заданным количеством разрядов.

У узла один вход. Количество выходов определяется заданным количеством разрядов 7-сегментного индикатора – от 1 до 16.

В поле параметров узла задаются следующие параметры:

«Digits» – количество разрядов индикатора, в которые нужно преобразовать входное значение от 1 до 16. Этим параметром определяется количество выходов узла.

«Number» – тип значения на выходе: «Int», «Hex», «Float» – соответственно целые 10-е числа, целые 16-е числа или число с плавающей точкой. В режиме «Hex» отрицательные числа не выводятся! Выводится значение «меньше минимального» (см. ниже).

Например, если вывод осуществляется на 4-разрядный индикатор, то число 12.42 будет выведено при разных значениях поля «Number» так.

Digits: 4	Zeros: Off	Zeros: On
Number: Int	12	0012
Number: Hex	C	000C
Number: Float	12.42	12.42



## ShloTiny5 – сделай сложности проще

«Zeros» – вывод ведущих нулей. При значении «Off» – ведущие нули не выводятся. При значении «On» – ведущие нули выводятся в режиме «Int» и «Hex».

Например, если вывод осуществляется на 6-разрядный индикатор, то число 12.42 будет выведено в разных режимах так.

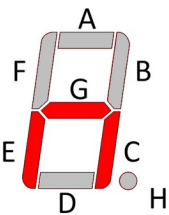
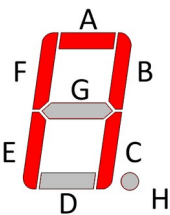
Digits: 4	Zeros: Off	Zeros: On
Number: Int	12	000012
Number: Hex	C	00000C
Number: Float	12.42	12.4200

Если на вход поступает значение NaN (не-число), то во всех разрядах индикатора выводится стилизованная буква «n».

Если на вход поступает значение «больше максимального», возможного для вывода в заданном режиме (например, Inf – бесконечность или число 1234567 при выводе на 6-значный индикатор), то во всех разрядах индикатора выводится стилизованная буква «P».

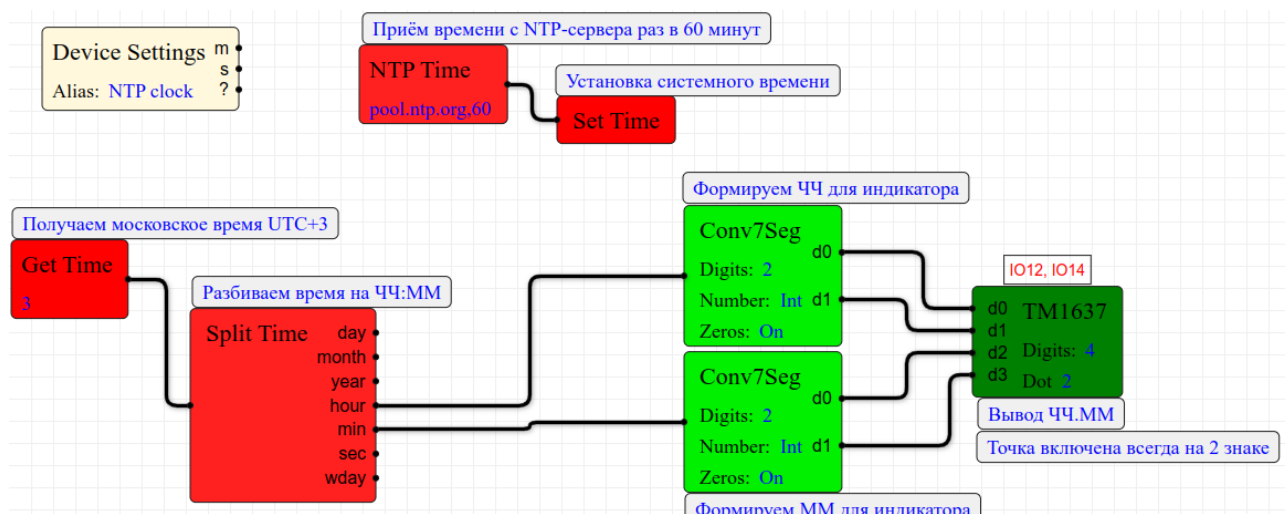
Если на вход поступает значение «меньше минимального», возможного для вывода в заданном режиме (например, число -654321 при выводе на 6-значный индикатор), то во всех разрядах индикатора выводится стилизованная буква «U».

Наглядно эти особые значения представлены в таблице ниже.

На входе узла значение NaN (не-число)	На входе узла значение «меньше минимального» или отрицательное значение в режиме «Hex»	На входе узла значение «больше максимального» или бесконечность (Inf)
		

Индикатор можно использовать в качестве нескольких независимых символов или в качестве отдельных светодиодов.

Например, простейшие NTP-часы на 4-сегментном модуле TM1637 выглядят как показано на рисунке.



### 6.13.3 Scale – шкала на базе 7-сегментного индикатора

**Scale** Узел «Scale» предназначен для преобразования входного значения в диапазоне 0..1 в представление последовательности бит для линейной шкалы.

Имеются опциональные функции индикации уровня «меньше нуля» и «больше единицы».

Предполагается, что шкала состоит из  $N$  светодиодов. Единица в  $i$ -м бите – светодиод  $i$  включён, нуль в  $i$ -м бите – светодиод  $i$  отключён.

Входное значение в диапазоне  $x \in [0..1]$  преобразуется в последовательность из  $N$  бит, в которой биты с номерами  $i = 0..(N-1) \cdot x$  – равны 1, а биты с номерами  $i = ((N-1) \cdot x + 1) .. (N-1)$  – равны 0.

Далее, сформированная последовательность бит преобразуется в несколько байт. Число байт равно  $B = (Leds + Shift + 7) \div 8$ , где параметры **Leds** и **Shift** задаются в поле параметров узла. Деление в формуле – целочисленное.

Эти сформированные байты можно затем подать на входы узла управления выходами или светодиодами. Например, на входы узла TM1637. В этом случае микросхема TM1637 будет рассматриваться просто как драйвер, управляющий отдельными 48 светодиодами, которые конструктивно объединены в одну или несколько шкал.

Параметр **Leds** задаёт количество светодиодов шкалы в диапазоне от 1 до 64.

Параметр **Shift** задаёт сдвиг 0-го бита шкалы относительно 0-го бита (выход d0).

Параметр **Flash min** задаёт режим «меньше минимального». То есть режим мерцания младшего, 0-го бита шкалы при значении на входе узла меньше нуля. Если параметр **Flash min** установлен в значение **yes**, то при появлении на входе шкалы значения меньше 0, младший, 0-й, бит шкалы начинает периодически переключаться с 0 на 1 и наоборот.

Параметр **Flash max** задаёт режим «больше максимального». То есть режим мерцания старшего, (N-1)-го бита шкалы при значении на входе узла больше 1. Если параметр **Flash max** установлен в значение **yes**, то при появлении на входе шкалы значения больше 1, старший, (N-1)-й, бит шкалы начинает периодически переключаться с 0 на 1 и наоборот.

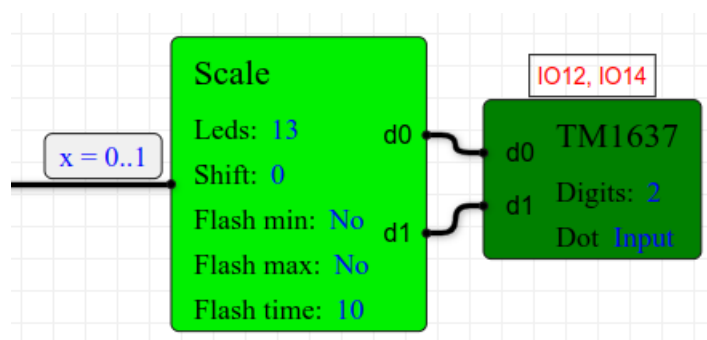
Параметр **Flash time** задаёт период мерцания светодиодов в режиме «меньше минимального» и «больше максимального». Период задаётся в децисекундах (0.1сек) в диапазоне от 0.1 до 10 сек. По умолчанию, период мерцания равен 1сек (10 децисекунд).

Рассмотрим примеры. Примеры выполнены на базе узла TM1637, поскольку модули на базе этой микросхемы широко доступны.

### 6.13.3.1 Пример 1. Шкала из 13 светодиодов без сдвига.

Простейший пример - формирование шкалы из 13 светодиодов. Будем считать, что шкала начинается с 0го бита 0го разряда индикатора.

Схема формирования такой шкалы показана на рисунке ниже.



Как видим, шкала занимает два разряда в соответствии с формулой  $B = (Leds + Shift + 7) \div 8 = (13 + 0 + 7) \div 8 = 2$ .

Приведём какие биты будут включены и какие выключены при различных значениях на входе x узла Scale. Цветом в таблице выделены те биты, которые используются шкалой.

Вход x=0.1	Выход d0	Выход d1
------------	----------	----------

## ShloTiny5 – сделай сложности проще

№дита	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Значение дита	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Вход x=0.48	Выход d0								Выход d1							
№дита	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Значение дита	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Вход x=0.73	Выход d0								Выход d1							
№дита	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Значение дита	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0

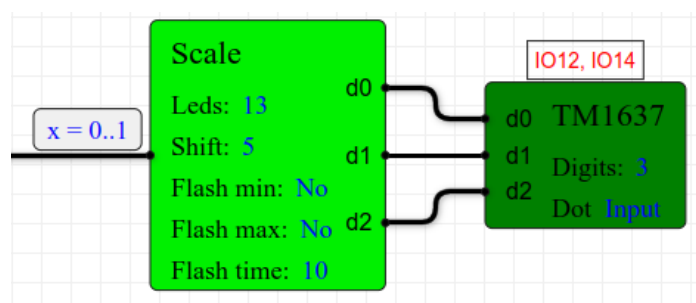
При  $x \leq 0$  все диты шкалы (выделены цветом) выключены (0).

При  $x \geq 1$  все диты шкалы (выделены цветом) включены (1).

### 6.13.3.2 Пример 2. Шкала из 12 светодиодов со сдвигом на 5

Следующий пример – формирование шкалы из 13 светодиодов со сдвигом на 5 дита относительно 0го дита шкалы. То есть шкала начинается с 5го дита 0го разряда индикатора.

Схема формирования такой шкалы показана на рисунке ниже.



Как видим, шкала занимает уже разряда в соответствии с формулой  $B = (Leds + Shift + 7) \div 8 = (13 + 5 + 7) \div 8 = 3$ . Это происходит из-за сдвига шкалы на 5 дит. Приведём какие диты будут включены и какие выключены при различных значениях на входе x узла Scale. Цветом в таблице выделены те диты, которые используются шкалой.

Вход x=0.1	Выход d0								Выход d1								Выход d2							
№ дита	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Знач. дита	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Вход x=0.48	Выход d0								Выход d1								Выход d2							

## ShloTiny5 – сделай сложности проще

№ дума	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Знач. дума	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Вход x=0.73	Выход d0								Выход d1								Выход d2							
№ дума	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Знач. дума	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
Вход x=0.87	Выход d0								Выход d1								Выход d2							
№ дума	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Знач. дума	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0

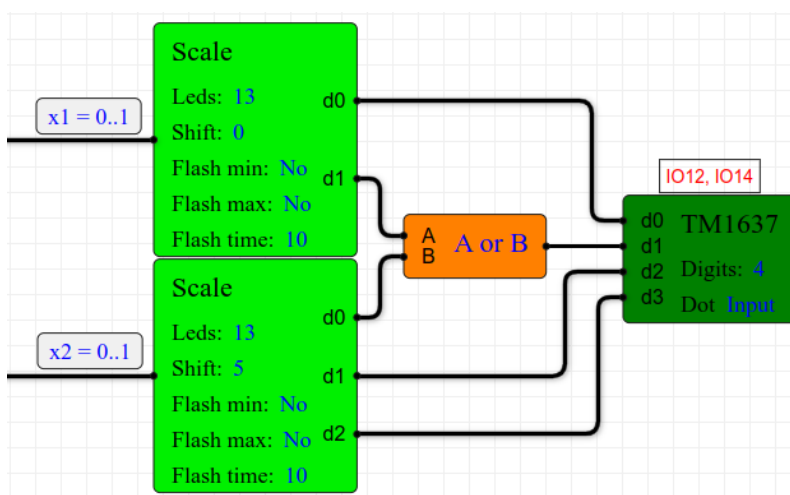
При  $x \leq 0$  все думы шкалы (выделены цветом) выключены (0).

При  $x \geq 1$  все думы шкалы (выделены цветом) включены (1).

### 6.13.3.3 Пример 3. Объединение шкал

Так как мы можем создавать шкалы со сдвигом, то шкалы можно безболезненно объединять, чтобы использовать светодиоды индикаторов без разрыва.

Объединение шкал из двух предыдущих примеров показано на рисунке ниже.



В таблице ниже показано соответствие бит входов узла TM1637 каждой из шкал.

Соответствие объединённых шкал номерам битов и входам TM1637																															
Шкала 1 (параметр x1)														Шкала 1 (параметр x2)														Не использ.			
Вход d0 TM1637								Вход d1 TM1637								Вход d2 TM1637								Вход d3 TM1637							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7

## ShloTiny5 – сделай сложности проще

Как видим, у нас осталось ещё 5 бит, которые не используются. На них можно «повесить», например, отдельные светодиоды для индикации режимов работы устройства.

## 7 Настройка ShloTiny5

Настройка устройства производится на странице конфигурации устройства – «Configure».

На вкладке имеются раздел настроек сети и доступа «Network mode and access» и раздел управления безопасным режимом «Safe Mode».

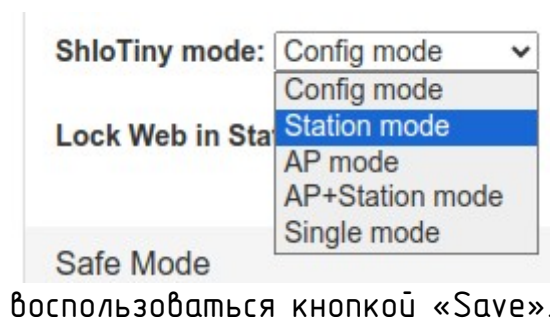
Для того, чтобы сохранить все изменения в памяти устройства, служит кнопка «Save». После сохранения изменений происходит перезапуск устройства.

Кнопка «Cancel» служит для перечитывания параметров из памяти устройства.

Кнопка «Reset to default» служит для возврата к заводским настройкам.

### 7.1 Режимы работы ShloTiny5

ShloTiny5 имеет пять режимов работы. Эти режимы отличаются по назначению, способу доступа к устройству и предоставляемым сетевым возможностям.



Выбор режима работы производится на странице конфигурации в поле «ShloTiny mode», как показано на рисунке.

Для перезапуска устройства в выбранном режиме, следует

воспользоваться кнопкой «Save».

#### 7.1.1 Режим «Config mode»

После того, как устройство первоначально «прошито» прошивкой ShloTiny5, после включения оно переходит в режим конфигурации – «Config mode».

Этот режим работы предназначен для того, чтобы провести начальную настройку устройства и затем переключить его в один из рабочих режимов.

В режиме «Config mode» устройство представляет собой открытую точку доступа сети WiFi. К ней можно подключиться с любого смартфона или ноутбука и затем зайти на устройство с помощью браузера по фиксированному адресу 192.168.4.1.

Особенностью режима «Config mode» является то, что схема, находящаяся в памяти не запускается автоматически на выполнение.

Кроме этого, в режиме «Config mode» всегда запускается WEB-сервер, вне зависимости от настройки его блокировки.

### *Как перейти в «Config mode»?*

Часто есть необходимость перейти в режим конфигурации, чтобы ввести новую схему или исправить существующую.

Есть два способа это сделать.

В модуле ESP предусмотрен вывод GPIO15, предназначенный для перевода модуля под управлением ПО ShloTiny5 в режим «Config mode».

**Способ первый.** Модуль работает, не «завис». В этом случае для перевода модуля в режим конфигурации достаточно замкнуть вывод GPIO15 на VCC на время более 5сек. После этого модуль должен перезагрузиться в режиме «Config mode».

**Способ второй.** Если модуль совсем «завис», то подойдёт второй способ. Он требует быстроты. Сразу после «сброса» модуля подачей сигнала низкого уровня на вывод RESET (например на модулях NodeMCU для этого есть кнопка), необходимо в течении времени не более 0.5сек от отпускания кнопки RESET, замкнуть вывод GPIO15 на VCC. После этого модуль должен перезагрузиться в режиме «Config mode».

Лучше всего делать это при помощи двух кнопок. Первая кнопка при нажатии замыкает RESET на GND. Вторая кнопка при нажатии замыкает GPIO15 на VCC.

Выглядит это так:

- жмем кнопку RESET.

- отпускаем кнопку RESET и тут же, как можно скорее, замыкаем GPIO15 на VCC с помощью второй кнопки и держим её нажатой не менее 1сек.

Кроме этого, для режимов «Station mode» и «AP+Station mode» предусмотрены так называемые «безопасные режимы» или «Safe mode» (см.ниже).

### **7.1.2 Режим «Station mode»**

Это рабочий режим, в котором устройство предоставляет собой станцию WiFi, подключаемую к какой-либо точке доступа.



## ShloTiny5 – сделай сложности проще

Параметры доступа к сети WiFi задаются в полях «Name (SSID)» и «Password» на странице конфигурации устройства, показанных на рисунке ниже.

Чтобы посмотреть какие точки доступа включены в данный момент, можно воспользоваться кнопкой «Scan WiFi».



The screenshot shows a configuration interface with two input fields. The first field is labeled 'Name (SSID)' and contains the text 'ssid'. The second field is labeled 'Password' and contains six dots, indicating a masked password.

После того, как выбран режим «Station mode» и устройство перезагружено, оно будет пытаться соединиться с указанной в поле «Name (SSID)» точкой доступа.

При успешном соединении, IP-адрес устройства будет получен от точки доступа.


Если не включена блокировка WEB-сервера (см. ниже), то он будет запущен и устройство ShloTiny5 будет доступно в локальной сети по этому адресу.

### 7.1.3 Режим «AP mode»

Это рабочий режим, в котором устройство предоставляет собой точку доступа WiFi.

В отличие от режима «Config mode», в «AP mode» должны быть заданы (см. рисунок ниже) название точки доступа в поле «AP Name (SSID)» и пароль для доступа к ней в поле «AP Password». Учтите, что пароль должен быть НЕ МЕНЬШЕ 8 символов длиной!

Кроме того, можно задать IP-адрес точки доступа «AP IP address» (по умолчанию 192.168.4.1).



The screenshot shows a configuration interface with three input fields. The first field is labeled 'AP Name (SSID)' and contains the text 'apssid'. The second field is labeled 'AP Password' and contains eight dots, indicating a masked password. The third field is labeled 'AP IP address' and contains the text '192.168.4.1'.

### 7.1.4 Режим «AP+Station mode»

Режим «AP+Station mode» это одновременное включение режимов «Station mode» и «AP mode».

То есть устройство сразу доступно и как точка доступа и пытается подключиться к указанной другой точке доступа.

Этот режим рекомендуется для целей отладки и экспериментов, но не желателен для рабочего, «боевого» применения.

### 7.1.5 Режим «Single mode»

В этом режиме сеть WiFi отключена вообще. Устройство работает как отдельный микроконтроллер.

## 7.2 Блокировка WEB-сервера

Во всех режимах, кроме «Config mode» и «Safe Mode» в целях безопасности можно заблокировать запуск WEB-сервера.

В этом случае, вы не сможете зайти на WEB-страницу устройства и что-либо там изменить. Также не сможет сделать это и злоумышленник.

Чтобы заблокировать запуск WEB-сервера, необходимо установить галочку «Lock Web in Station mode», как показано на рисунке ниже.

**Lock Web in Station mode** ☒

Отметим, что в безопасном режиме – WEB-сервер запускается и WEB-страница доступна. Поэтому, если вы хотите обеспечить безопасность подключения в «Safe Mode» – то выбирайте в качестве «безопасного» режима – «AP mode», поскольку он защищён паролем.

## 7.3 Блокировка TxD и вывода информационных сообщений в консоль

Системный UART выведен на GPIO1 (TxD) и GPIO3 (RxD). Так как системная консоль не используется, то GPIO3 всегда доступен в схеме как обычный вывод GPIO.

Что касается GPIO1 – то может возникать необходимость посмотреть в консоли отладочный вывод.

## ShloTiny5 – сделай сложности проще

С этой целью введена галочка «Lock TxD (GPIO1) pin» на странице «Configure».

Установка этой галочки означает блокировку вывода на линию TxD (GPIO1), то есть вывода в системную консоль. По умолчанию эта галочка **сброшена** – вывод сообщений в системную консоль **разрешён**. Это значит, что мы можем прочитать системные сообщения в консоли: режим работы, имя устройства, IP-адреса и т.д.

Если галочка «Lock TxD (GPIO1) pin» снята – то использовать GPIO1 в схеме нельзя. Результат будет непредсказуем.

Если вы хотите использовать вывод TxD (GPIO1) в схеме, то надо установить эту галочку как показано на рисунке ниже и сохранить конфигурацию устройства кнопкой «Save».

Lock TxD (GPIO1) pin ☒

После чего вывод в консоль ведётся только при работе загрузчика и можно использовать вывод TxD (GPIO1) в своих схемах.

*Помните, что короткое время (около 0.2 сек) после включения питания, на выводе TxD (GPIO1) появляется информация загрузчика и это не зависит от состояния галочки «Lock TxD (GPIO1) pin».*

*Поэтому вывод TxD нельзя использовать для критичных применений (например для управления силовым реле).*

### 7.4 Безопасный режим «Safe Mode»

Безопасный режим имеет смысл только для режимов работы «Station mode» и «AP+Station mode».

Суть «безопасного режима» заключается в том, что если пропадает связь с точкой доступа, то устройство ShloTiny5 через некоторое время автоматически переходит в другой режим работы – «AP mode» или «Config mode».

Из сказанного ясно, что для режимов работы «AP mode» и «Single mode» безопасный режим смысла не имеет и не включается.

По умолчанию безопасный режим отключён.

Настройка безопасного режима – «Safe Mode» – находится там же, на странице конфигурации, см. рисунок ниже.

Safe Mode	
Select safe mode:	No safe ▼
Time to switch to safe mode:	1 min ▼
Time to stay in safe mode:	Always (to reset) ▼

Тип переключения в безопасный режим определяется полем «Select safe mode». Первое значение «No safe» – означает, что режим отключён.

Значение «Switch to Config mode» означает, то при исчезновении связи с точкой доступа, по прошествии заданного времени, произойдёт автоматическое переключение в режим «Config mode».

И, наконец, значение «Switch to AP mode» означает, то при исчезновении связи с точкой доступа, по прошествии заданного времени, произойдёт автоматическое переключение в режим «AP mode».

Время, которое должно пройти без связи с точкой доступа для перехода

Time to switch to safe mode:	1 min ▼
Time to stay in safe mode:	Always (to reset) ▼

в безопасный режим задаётся полем «Time to switch to safe mode». На выбор несколько значений от 1 минуты до 1 часа.

Если в течении этого времени связь с точкой доступа восстановится, то перехода в заданный «безопасный режим» не произойдёт.

И, наконец, последний параметр определяющий работу «безопасного режима» – время, которое устройство будет оставаться в заданном режиме до новой попытки соединения с точкой доступа. Оно задаётся в поле «Time to stay in safe mode». Тут также есть значения на выбор – от 5 мин до 1ч.

Time to stay in safe mode:	Always (to reset) ▼
----------------------------	---------------------

Особый случай – значение «Always (to reset)».

Если выставлено это значение – то устройство, перейдя в заданный «безопасный режим» будет находиться в нём до выключения питания или получения сигнала «RESET». В остальных случаях, по прошествии заданного

## ShloTiny5 – сделай сложности проще

времени, устройство вновь вернётся к попыткам соединиться с точкой доступа.

### *Пример настройки безопасного режима.*

Рассмотрим пример. Предположим, наше устройство сконфигурировано так, как показано на рисунке ниже.

Разумеется, те параметры режима «AP mode» (см. пп. 7.1.3 ) тоже заданы и известны.

The screenshot shows the ShloTiny5 configuration interface. At the top, there is a dropdown menu for 'ShloTiny mode' set to 'Station mode' and a green 'Scan WiFi' button. Below this is a checkbox for 'Lock Web in Station mode' which is unchecked. A section titled 'Safe Mode' contains three dropdown menus: 'Select safe mode' set to 'Switch to AP mode', 'Time to switch to safe mode' set to '5 min', and 'Time to stay in safe mode' set to '30 min'. At the bottom of the 'Safe Mode' section are three buttons: a green 'Save' button, a blue 'Cancel' button, and a blue 'Reset to default' button.

Устройство работает нормально в режиме «Station mode». Это самый популярный режим работы контроллеров ShloTiny5.

Пока есть связь устройства с точкой доступа, то безопасный режим не включается. Даже если связь устройства с точкой доступа кратковременно – на время не более 5 минут – прерывается, то ничего не происходит.

Но если связь прервётся на время более 5 минут (см. настройки), то устройство автоматически перейдёт в режим «AP mode», как это указано в настройках. Далее устройство в течении 30 минут (см. настройки) будет находиться в режиме «AP mode». В течении этих 30 минут, зная пароль можно подключиться к этому устройству как точке доступа и изменить его параметры нужным образом.

По прошествии указанных 30 минут, устройство вновь перейдёт в режим «Station mode» на 5 минут и попытается восстановить связь с точкой доступа. Если это удастся, то работа устройства продолжится в «Station mode». Если за эти 5 минут связь с точкой доступа не появится, то устройство вновь автоматически перейдёт в режим «AP mode» на 30 минут. И так далее.

## ShloTiny5 – сделай сложности проще

Указанные настройки очень удобны на практике. Предположим, у вас есть несколько устройств ShloTiny5, которые работают в сети WiFi, которая создаётся вашей точкой доступа.

И в один прекрасный момент вы захотели поменять параметры какого-то из устройств, но не знаете его IP-адрес.

Самый простой способ – это просто выключить питание вашего WiFi-роутера, к которому подключены устройства ShloTiny5.

Через 5 минут, как в нашем примере, все эти устройства станут точками доступа. Подключайтесь к ним с ноутбука или смартфона – и меняйте параметры.

Для удобства указываете устройствам понятные псевдонимы (см.пп. 6.2 ).

## **8 Установка программного обеспечения ShloTiny5 на ESP-модуль (прошивка)**

ПО ShloTiny5 было опробовано на различных модулях – ESP07, ESP12, ESP14 и даже на ESP-M1.

Главное условие – чтобы на модуле было установлено не менее 1Мбайта FLASH. Иначе программное обеспечение туда просто не влезет.

Сразу оговорюсь. Если на вашем модуле только 1Мб FLASH – то функция обновления ПО по сети – OTA – работать не будет и вам придётся каждый раз для смены версии ПО возиться с проводами и переконфигурацией модуля.

### **8.1 Где взять прошивку ShloTiny5?**

На сайте проекта, в разделе посвящённом ShloTiny5 – <https://shiotiny.ru/?mod=shiotiny-soft5&top=1>.

Там есть ссылка на последнюю актуальную прошивку, которая так и называется «СКАЧАТЬ ПРОШИВКУ».

Кликните на неё – и прошивка ваша. Кстати, там же, рядом лежит ссылка «СКАЧАТЬ ИНСТРУКЦИЮ». Но на неё почему-то мало обращают внимания.

### **8.2 Как подключить ESP-модуль к ПК?**

Однозначный ответ – по USB. Остальное зависит от того, какой у вас модуль.



Если это NodeMCU, WemosD1 или ещё какой-то модуль, в который уже встроен USB-UART преобразователь и на модуле имеется USB-разъём – то всё просто.

Достаточно взять кабель и одним концом воткнуть его в модуль, а другим – в ваш компьютер. И всё. Питание модуля обычно подаётся по тому же USB-кабелю. И этот же кабель служит для передачи данных на консоль терминала, если вам нужна отладочная информация.

Если в вашем модуле нет USB-UART преобразователя, то вам нужен другой кабель, с USB-UART преобразователем, вроде того, что на рисунке справа.



## ShloTiny5 – сделай сложности проще

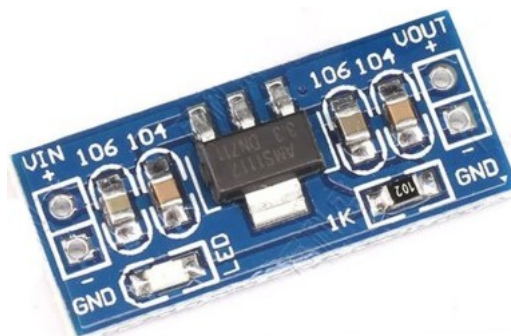
Как видим, с одной стороны у него USB-разъём, а с другой четыре провода. Как правило, красный и чёрный – это **+5В** и «земля», а зелёный и белый – линии **TxD** и **RxD** UART.

Но, чтобы не спалить свой модуль – лучше проверьте какого цвета какая линия на вашем кабеле по его описанию.

Если вы хотите питать ваш модуль от **USB-UART** преобразователя, то для этого надо подать +5В преобразователя на вход +5В модуля, а «землю» преобразователя соединить с «землёй» модуля.

Помните, что многие модули питаются напряжением +3В! Если модуль не имеет преобразователя напряжения, то его можно «спалить», подав +5В вместо +3В! Поэтому убедитесь, что ваш модуль имеет вход +5В. Если такого входа нет, то можно, например, использовать стабилизатор напряжения, преобразующий +5В в +3В.

Например, такие стабилизаторы выпускаются на базе чипа AMS1117 в виде готовых модулей, вроде того, что показан на рисунке справа.



Линии **TxD** и **RxD** надо соединять крест-накрест. То есть линию **TxD** преобразователя надо присоединить к **RxD** модуля, а линию **RxD** преобразователя – к **TxD** модуля.

Кроме того, если вы используете USB-UART преобразователь, то вам придётся вручную переводить модуль в режим программирования. Это не сложно. Достаточно замкнуть **GPIO0** на землю и нажать кнопку **RESET** модуля. Или просто выключить-включить питание модуля, если такой кнопки нет.

**Если модуль не подаёт признаков жизни**, то убедитесь, что его ножки правильно подтянуты через резисторы сопротивлением 1К – 5,1К к питанию и «земле» как показано в таблице.

Режим	Линия			
	<u>Reset</u>	<u>CH_PD</u>	GPIO0	GPIO15
Рабочий режим	+Vcc	+Vcc	+Vcc	0В
Программирование	+Vcc	+Vcc	0В	0В

Такие напряжения должны быть на указанных ножках модуля, чтобы он правильно запускался.



## ShloTiny5 – сделай сложности проще

Когда вы подключите свой модуль к компьютеру, то у вас появится новое устройство – последовательный порт.

Под Linux это будет что-то вроде `/dev/ttyUSB0`, `/dev/ttyUSB1`.

А под Windows – COM1, COM2 или что-то подобное.

### 8.3 Как соединить Shlotiny5 с терминалом?

Терминал – очень нужная вещь для начального этапа работы с любым микроконтроллером.

Он позволяет проверить «жив» ли микроконтроллер, посмотреть параметры и так далее.

В качестве терминала можно использовать кроссплатформенный универсальный терминал PuTTY.

В настройках терминала PuTTY выбираем тип сессии **«Serial» «RAW»**. Указываем ваш последовательный порт в строке **«Serial Line»**.

Во вкладке **«Serial»** указываем скорость **115200**, бит данных **8**, количество стоп-бит **1**.

И ещё надо обязательно установить поля **«Parity»** и **«Flow Control»** в значение **«None»**.

Для того, чтобы вывод был красивым, во вкладке **«Terminal»** устанавливаем галочки **«Implicit CR in every LF»** и **«Implicit LF in every CR»**.

После этого можно жать кнопку **«Open»** и любоваться окном терминала.

Помните, что терминал при «прошивке» устройства надо закрывать!

Если всё подключено правильно, то при нажатии кнопки **«RESET»** вашего не прошитого модуля, в терминал будет валиться «мусор».

Также можете настроить в терминале PuTTY шрифты, которые вам нравятся.

### 8.4 Как прошивать Shlotiny5?

«Прошивка» – образ FLASH – ShloTiny5, который вы скачали с сайта – это файл размером 1Мбайт.

Будем считать, что он называется **esp-shiotiny.bin**.

Будем считать, что порт, через который прошивается модуль называется **`/dev/ttyUSB0`**.

## ShloTiny5 - сделай сложности проще

Если у вас другой порт или файл – то соответственно измените их названия в командах.

Прошивка выполняется с помощью утилиты **esptool.py**. О том, как её установить – полно статей в интернете. Например тут: <http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D1%8B:esp8266:esptool>.

Будем считать, что с установкой утилиты **esptool.py** пользователь справится самостоятельно.

Если у вас модуль, который имеет разъём USB на борту, то всё довольно просто – надо выполнить две команды, показанные ниже. Этот способ прошивки проверялся на модулях **NodeMCU** и **Wemos D1 mini**.

```
# Стираем FLASH
esptool.py --port /dev/ttyUSB0 --baud 115200 erase_flash

# Записываем образ ПО ShloTiny5 во FLASH
esptool.py -p /dev/ttyUSB0 -b 115200 write_flash -fs detect -ff 26m
0x000000 esp-shiotiny.bin

# Запускаем терминал, нажимаем RESET на модуле и любуемся результатом на
терминале
```

Если у вас модуль, который подключается к терминалу по двум линиям – **RxD** и **TxD** через USB-UART преобразователь, то переводить его в режим программирования и жать RESET нужно вручную, как написано ниже.

Если отсутствует кнопочка RESET, то вместо него выключаем и включаем питание модуля. Этот способ прошивки проверялся на модулях ESP-07, ESP-12, ESP-14.

```
# Переводим модуль в режим программирования,
# для чего замыкаем GPIO0 на землю на всё время,
# пока модуль будет программироваться.

# Жмём RESET (или выкл-вкл питание)

# Стираем FLASH
esptool.py --port /dev/ttyUSB0 --baud 115200 erase_flash
```

## ShloTiny5 - сделай сложности проще

```
# Жмём RESET (или выкл-вкл питание)
# Записываем образ ПО ShloTiny5 во FLASH
  esptool.py -p /dev/ttyUSB0 -b 115200 write_flash -fs detect -ff 26m
0x000000 esp-shiotiny.bin

# Переводим модуль в рабочий режим,
# для чего отключаем GPIO0 от земли.
# В рабочем режиме GPIO0 должен быть подтянут к Vcc через резистор.

# Запускаем терминал, нажимаем RESET на модуле (или выкл-вкл питание) и
любимся результатом на терминале
# Любуемся результатом на терминале
```

Если всё прошло успешно, то после прошивки, модуль можно перезагрузить, нажав на кнопку RESET, если таковая имеется или просто выключив-включив его питание.

При этом, если запустить терминал, то после нажатия кнопки RESET модуля, на нём появится примерно такая информация.

```
l\m1
--- Init esp8266 node: esp_8266_dcf6f2 ---
--- Software ver: [25.11.2023 23:48:54 v0.29 +0 37454bf clean] ---

Init OTA system
Cur boot: rom1 [magic= E1 version= 1 currom= 1 count= 2 rom0= 2000 rom1= 102000]
Next boot: rom0
OTA enabled. Boot from rom1.
Flash Info: mode:QIO, speed:26MHz, size:4MB (4194304 bytes).

Spiff mount [spiffs1]... Found... Mounted

Starting in Config mode
mode : softAP(ae:0b:fb:dc:f6:f2)
add if1
dhcp server start:(ip:192.168.4.1,mask:255.255.255.0,gw:192.168.4.1)
bcn 100
█
```

Это значит, что прошивка **Shlotiny5** запущена в режиме «**Config mode**» и среди ваших WiFi сетей появилась открытая сеть с именем «esp\_8266\_dcf6f2».

Подключитесь к ней со своего ноутбука или смартфона и в браузере наберите адрес «<http://192.168.4.1>».

Если всё сделано правильно - то вы попадёте на главную страницу устройства Shlotiny5 - «ShloTiny Info».

Для настройки параметров подключения к сети WiFi и других параметров нажмите на кнопку «Configure» - и попадёте на страницу конфигурации, описанную в разделе Настройка ShloTiny5.

## 9 OTA – обновление программного обеспечения по сети

Технология OTA (Over The Air) – «программа по воздуху» – это технология обновления программного обеспечения прямо на работающем устройстве без какого-либо дополнительного аппаратного подключения к нему.

В ShloTiny5, начиная с версии **v0.29** предусмотрена возможность обновить программное обеспечение через WEB-страницу. Это не совсем OTA в классическом понимании, так как нет единого хранилища образов программного обеспечения. Так же пока нет автоматического обновления ПО.

На функцию OTA в ShloTiny5 есть ограничения!  
Если в используемом модуле менее 2Мбайт FLASH, то OTA работать не будет!

Если в вашем модуле не менее 2Мбайт FLASH, то на странице конфигурации появится кнопка **«Upgrade»**, как показано на рисунке ниже.

Configure

ShloTiny Info

EIDraw

Network mode and access

Name (SSID)

ssid

Password

\*\*\*\*\*

AP Name (SSID)

apssid

AP Password

\*\*\*\*\*

AP IP address

192.168.4.1

ShloTiny mode:

Config mode

▼

Scan WiFi

Lock Web In Station mode

☐

Lock TxD (GPIO1) pin

☐

Safe Mode

Select safe mode:

No safe

▼

Time to switch to safe mode:

60 min

▼

Time to stay in safe mode:

30 min

▼

Save

Cancel

Upgrade

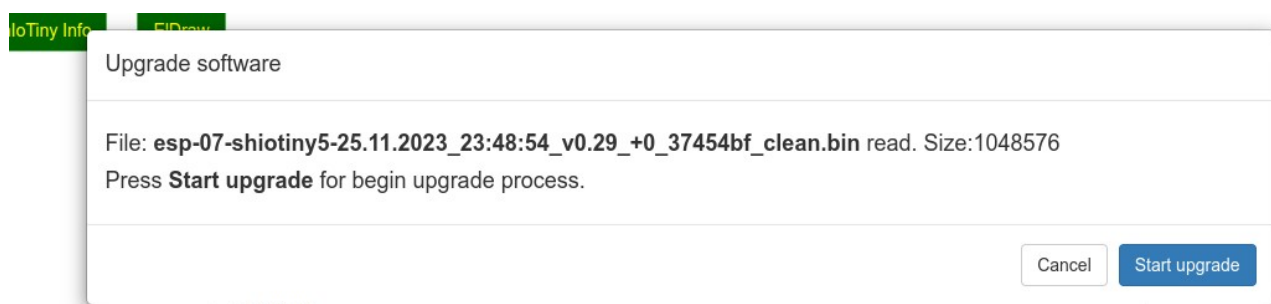
Reset to default

## ShloTiny5 - сделай сложности проще

Если в вашем модуле менее 2Мбайт FLASH, то на странице конфигурации кнопка **«Upgrade»** не появляется (функция обновления не доступна).

Для обновления ПО, необходимо нажать кнопку **«Upgrade»**. После этого появится диалог выбора файлов, в котором необходимо выбрать файл с образом программного обеспечения, версии не менее **v0.29**.

После того, как файл выбран, он загружается и проверяется на корректность. Если файл признан корректным, то на экран выводится окно с сообщением, подобным тому, что показано на рисунке.



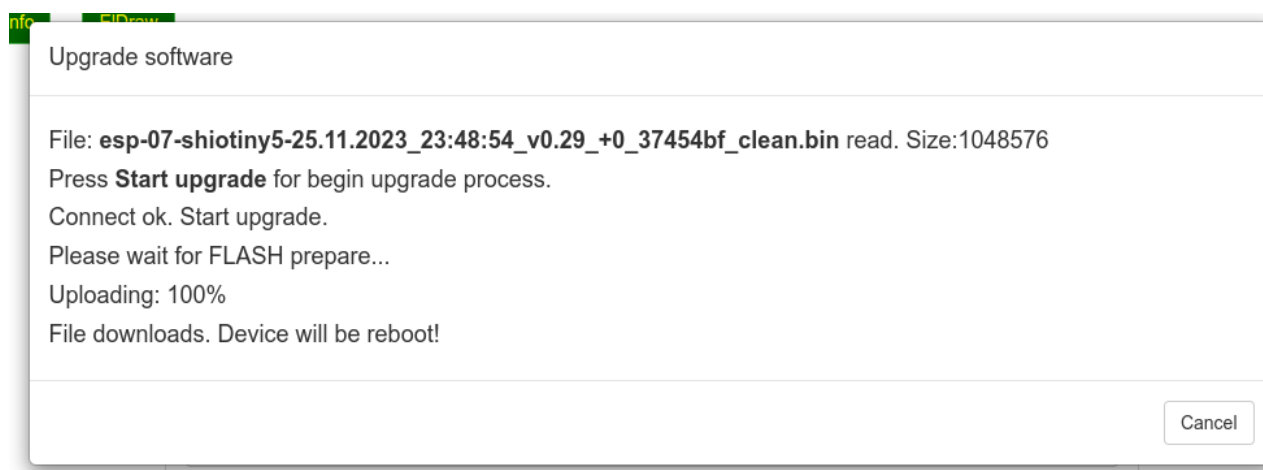
Если вы хотите начать процесс обновления программного обеспечения, то нажмите кнопку **«Start Upgrade»**. Иначе нажмите кнопку **«Cancel»**.

При нажатии на кнопку **«Start Upgrade»**, начнётся процесс загрузки программного обеспечения из файла в устройство. В окне также отображается информация о том, сколько процентов загружено.



После того, как программное обеспечение загружено, производится его настройка внутри устройства и, затем, устройство перезагружается. При этом, в окне появляется сообщение, подобное тому, как на рисунке ниже.

## ShloTiny5 – сделай сложности проще

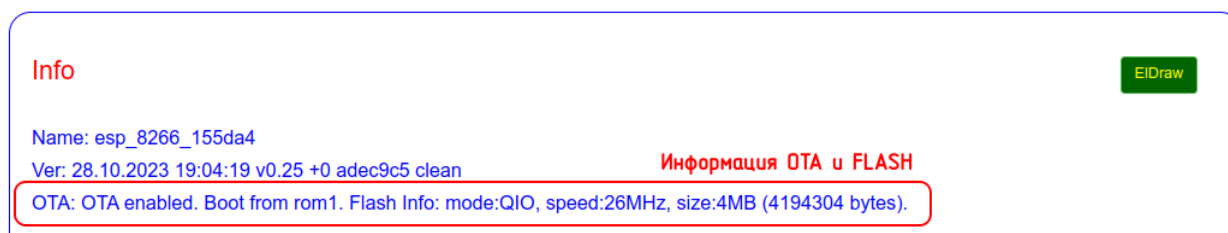


После загрузки образа ПО на устройство, процесс настройки и перезагрузки устройства длится около 5–7 секунд. Затем оно загружается обычным образом.

После перезагрузки устройства страница автоматически не обновляется!  
Обновите её вручную!  
Помните, что IP-адрес устройства, получаемый по dhcp может измениться после перезагрузки!

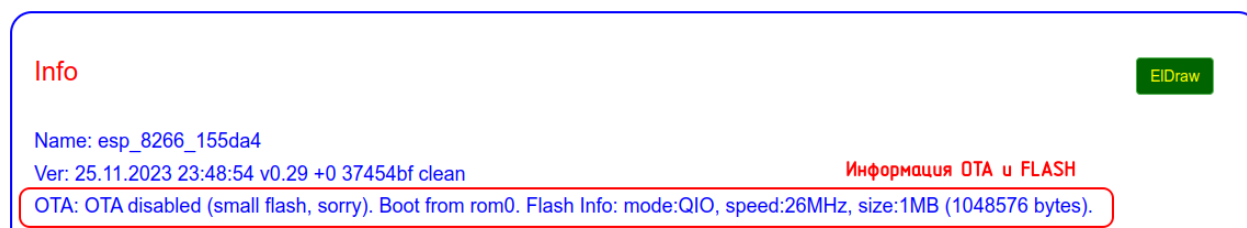
На странице информации «ShloTiny Info» есть подробная информация по состоянию подсистемы OTA и параметрам FLASH-памяти устройства, как показано на рисунке ниже. В этом примере OTA разрешено, памяти хватает.

### ShloTiny Info



А в следующем примере OTA запрещено, памяти мало.

### ShloTiny Info



## ShloTiny5 – сделай сложности проще

Рассмотрим подробнее информацию, выводимую на странице «ShloTiny Info» в строке OTA.

**OTA enabled / OTA disabled** – разрешена / запрещена функция обновления ПО через сеть. Если запрещена – кнопка «**Upgrade**» на странице «**Configure**» отсутствует. Если разрешена – эта кнопка есть.

**Boot from rom0/rom1** – показывает из какого раздела FLASH загружено ПО. Раздела 2 – rom0 и rom1. После каждого обновления эти разделы меняются местами.

**Flash Info** – информация о режиме работы FLASH-памяти, её частоте и объёме.