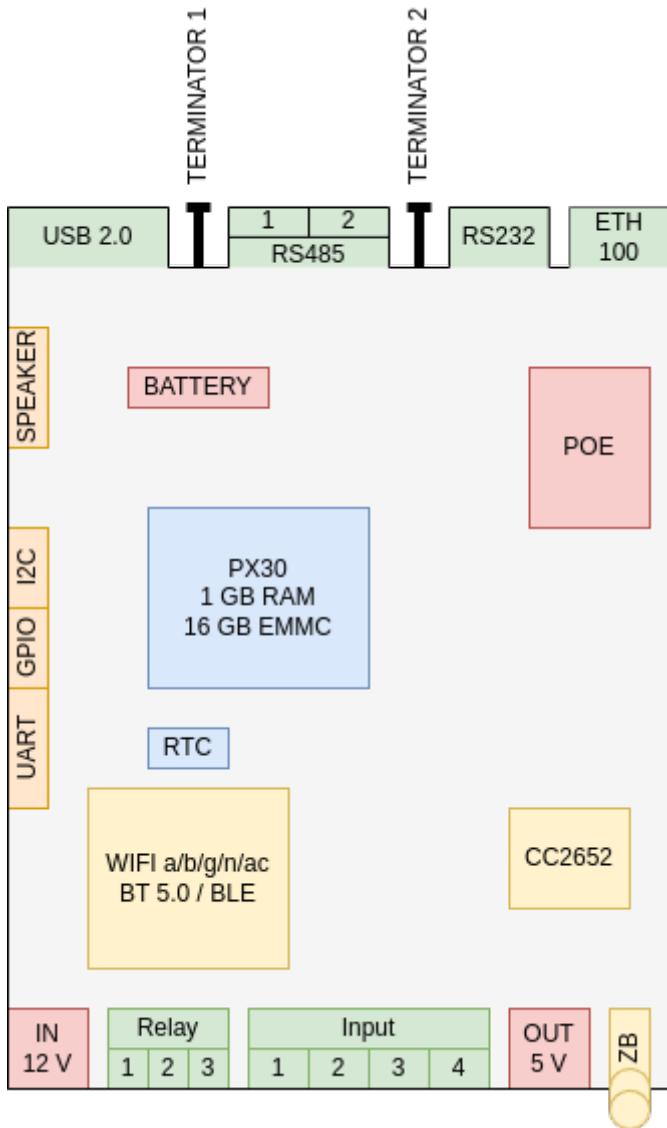


## Содержание

<b>Отладочная плата с модулем PX30</b>	3
<b>Техническое описание</b>	3
<b>Возможности монтажа</b>	4
<b>Быстрый старт</b>	4
Запуск линукса на основе buildroot	4
Запуск Diet Pi	4
<b>Подача питания</b>	4
<b>Запуск устройства</b>	5
<b>Работа с периферией</b>	5
WiFi	5
Zigbee	6
USB 2.0	6
PMIC	6
Последовательные интерфейсы	6
Сухие контакты	7
Реле и Светодиоды на передней панели	7
Кнопки на передней панели	7
I2C	7
RTC	7
<b>Запуск Zigbee2MQTT</b>	7



# Отладочная плата с модулем PX30



## Техническое описание

- Процессор Rockchip PX30
- WiFi ac
- Bluetooth 5.0
- 100Mbps Ethernet с возможностью PoE
- Zigbee на основе CC2652
- Интерфейс RS232
- Два интерфейса RS485
- 3 линии реле ЗА
- 4 входного контакта
- Выход 5 Вольт
- Встроенный аккумулятор
- USB 2.0

## Возможности монтажа

Устройство предназначено для установки DIN-рейку.

## Быстрый старт

Вы можете использовать уже собранный образ операционной системы, на данный момент, поддерживаются Buildroot и Diet Pi. Скачайте архив с образами по [этой ссылке](#) и распакуйте его. Инструкция предназначена для прошивки в линуксе, например Ubuntu.

### Запуск линукса на основе buildroot

Убедитесь что плата выключена.

1. Поставьте джампер X6.
2. Подключите кабель для прошивки устройства к компьютеру и к плате умного хаба.
3. Подайте питание на устройство и снимите джампер X6.
4. Перейдите в папку `burn_buildroot`
5. Выполните скрипт `burn_br.sh`

Если устройство успешно вошло в режим прошивки, на экране, вы увидите надпись «maskrom: OK». Если прошивка прошла успешно, вы увидите надпись «Success». В сборке предустановлен Docker.

### Запуск Diet Pi

Убедитесь что плата выключена.

1. Поставьте джампер X6.
2. Подключите кабель для прошивки устройства к компьютеру и к плате умного хаба.
3. Подайте питание на устройство и снимите джампер X6.
4. Перейдите в папку `burn_dietpi`
5. Выполните скрипт `burn_dietpi.sh`

Логин и пароль по умолчанию:

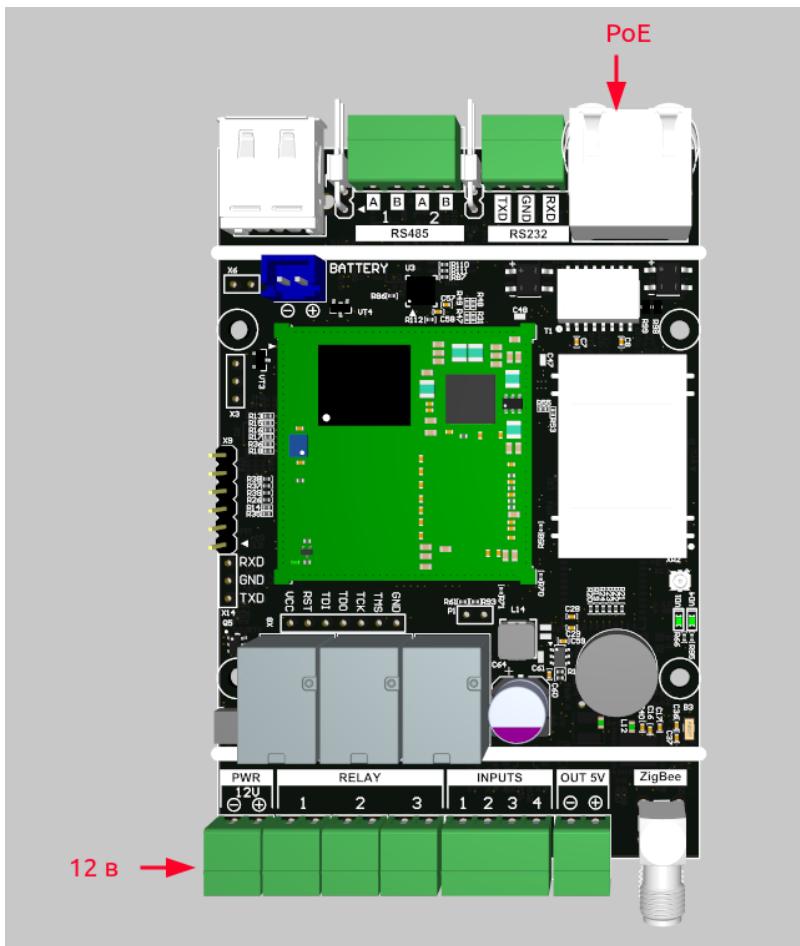
**Логин: root**

**Пароль: dietpi**

## Подача питания

Хаб можно запитать двумя способами: подав 12 вольт (3 ампера) на клеммы снизу (подписано PWR), либо через Ethernet разъем по технологии PoE.

На плате хаба предусмотрен разъем для подключение батарейки, однако обратите внимание **штекерография неправильная**.



## Запуск устройства

На плате хаба, на правой стороне устройства нанесены метки *RDX*, *GND*, *TXD*. Это разъем UART к которому можно подключить для взаимодействия с устройством. *БОД 152000*. Для открытия терминала, на своем компьютере воспользуйтесь утилитой *picocom*

```
picocom -b 115200 /dev/ttyUSB0
```

## Работа с периферией

### WiFi

Включить wifi можно командой

```
connmanctl enable wifi
```

Чтобы проверить что wifi сети обнаруживаются, выполните две команды

```
connmanctl scan wifi
connmanctl services
```

## Zigbee

Zigbee модуль CC2652 R74 по пути: /dev/ttyS5

Для проверки версии zigbee, воспользуйтесь следующей командой:

```
/mnt/store/cc2652/check.sh
```

## USB 2.0

Разъем USB на верхней части устройства, позволяет подключать флешки и прочую периферию к хабу умного дома. По умолчанию USB разъем активен. Для проверки разъема, вставьте флешку и выполните команду

```
cat  
/proc/partitions
```

. Должно появиться новое устройство.

## PMIC

Проверить состояние аккумулятора можно командой

```
cat /sys/class/power_supply/battery/uevent
```

Пример вывода:

```
POWER_SUPPLY_NAME=battery
POWER_SUPPLY_STATUS=Discharging
POWER_SUPPLY_CURRENT_NOW=1000
POWER_SUPPLY_VOLTAGE_NOW=4230000
POWER_SUPPLY_HEALTH=Good
POWER_SUPPLY_CAPACITY=96
POWER_SUPPLY_CAPACITY_LEVEL=Full
POWER_SUPPLY_TEMP=188
POWER_SUPPLY_CHARGE_COUNTER=0
POWER_SUPPLY_CHARGE_FULL=5000000
POWER_SUPPLY_CHARGE_FULL_DESIGN=5000000
POWER_SUPPLY_TIME_TO_FULL_NOW=0
```

## Последовательные интерфейсы

Интерфейс	Устройство /dev
RS232	/dev/ttyS2
RS485_1	/dev/ttyS0
RS485_2	/dev/ttyS4
Console	/dev/ttyS3
Zigbee	/dev/ttyS5

\* Dietpi по умолчанию запускает getty сервис на ttyS2. Если не работает RS232, надо сделать disable сервиса

```
systemctl | grep ttyS2
systemctl disable SERVICE_NAME
```

## Сухие контакты

Заведены как gpio. DIG\_IN1 ... DIG\_IN4

[Скрипт для управления gpio через sysfs](#)

gpio.sh

```
#read state
./gpio.sh DIG_IN1
```

## Реле и Светодиоды на передней панели

Заведены как светодиоды.

```
ls /sys/class/leds/*
cat /sys/class/leds/RELAY_1/brightness
echo 1 > /sys/class/leds/RELAY_1/brightness
```

## Кнопки на передней панели

Левая кнопка (питание) заведена на PMIC и при долго нажатии отключает устройство. В линуксе генерирует событие **KEY\_POWER**

```
evtest /dev/input/event0
```

Правая кнопка (PAIR) это GPIO процессора и в линуксе генерируется событие **KEY\_UP**

```
evtest /dev/input/event1
```

## I2C

## RTC

## Запуск Zigbee2MQTT