

# Собираем прошивку zigbee из исходников

Оригинальная статья от [sigma7i](#)

## Собираем прошивку zigbee из ИСХОДНИКОВ.

В этой статье я постараюсь описать процесс сборки прошивки под zigbee с использованием среды IAR Embedded Workbench® for 8051.

### Предисловие:

Сразу надо ответить на вопрос: а почему бы сразу не дать прошивку? На это есть множество причин:

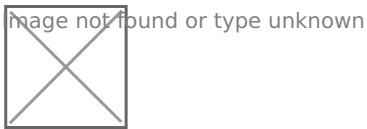
1. Это противоречит философии open source: если Вы захотите что-то подправить то у вас это не получится.
2. Если вы найдете ошибку/доработку/оптимизацию но автор уже давно закинул проект, то не имея исходного кода проект просто на просто “умрет”.
3. Если Вам понадобится поменять, что то очень простое, под себя, даже не умея программировать всегда можно разобраться, какие цифры за что отвечают.
4. Когда есть открытый код, и информация как все собрать вместе, написанная доступным языком, есть всегда возможность поэкспериментировать с настройками, разобраться что и как работает.

И много других плюсов.

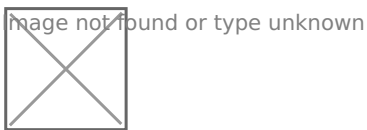
Для демонстрации установки установил чистую систему, чтобы показать как это делается с чистого листа.

### Установка среды разработки

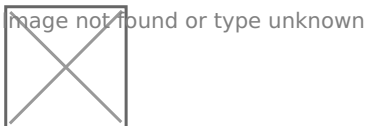
В качестве среды разработки будет использоваться IAR Embedded Workbench, где цифра 8051 это архитектура чипа, а не версия системы, как можно подумать. Для начала необходимо ее [скачать по ссылке](#)



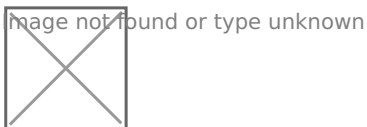
Запускаем, необходимо выбрать пункт "Install IAR Embedded Workbench® for 8051"



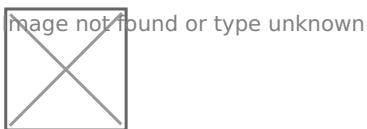
При установке iar надо выбрать режим custom:



и убрать галочку с установки Dongle drivers



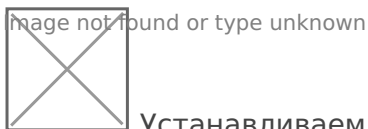
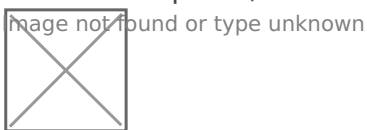
При попытке запуска проекта может быть такая ошибка:



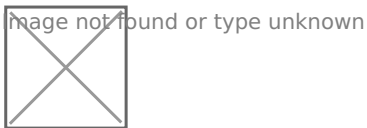
Поэтому перед запуском проекта необходимо убедиться что python установлен. Проверить это можно в командной строке:

```
python --version
```

Если все хорошо, то покажет текущую установленную версию python, как на примере:

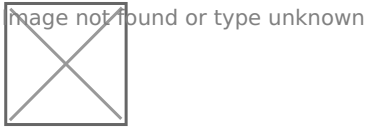


Если же нет, то [идем устанавливать python](#) Устанавливаем его. Обязательно необходимо проставить галочку "add python \*\*\*\* to path"

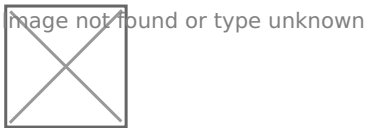


# Устанавливаем git и простой клиент для него.

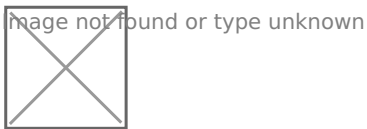
заходим и [скачиваем git](#) Устанавливаем, все галочки можно оставить по умолчанию.



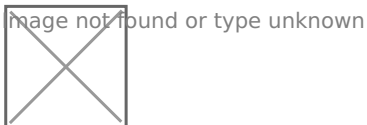
заходим на сайт <https://tortoisegit.org/> Скачиваем программу в зависимости от разрядности системы(ну или просто выбирайте 32 bit, никакой существенной роли не играет) По необходимости можно скачать русификацию. В этой инструкции делать это не буду.



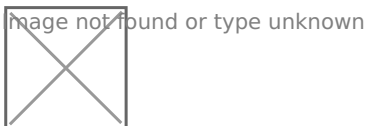
Устанавливаем tortoisegit, после установки ставим галочку "Run first start wizard"(обычно она по умолчанию стоит) По умолчанию пропишется путь к git.exe, либо нужно самому указать путь.



Сразу можно сгенерировать ssh ключи "Generate PuTTY key pair":



Нажимаем generate, активно двигаем мышкой/нажимаем на клавиши(человек выступает в роли ГСЧ) После генерации сохраняем приватный ключ "Save private key", публичный ключ(на рисунке он начинается "ssh-rsa AA..."), копируем в буфер обмена (можно так же сохранить его сохранить по необходимости "Save public key")

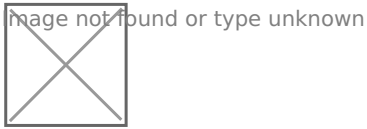


[Заходим в свой аккаунт git в раздел SSH keys](#)

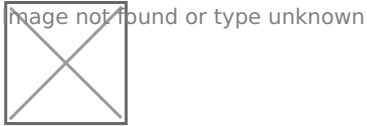


Нажимаем "Add new" вставляем скопированный ранее публичный ключ.

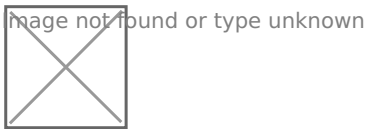
Устанавливаем прошивку [Z-Stack 3.0.2](#) — это SDK для разработки прошивок.



Отвечаем на каверзные вопросы, скачиваем.



Устанавливаем, можно немного уменьшить путь до zstack: C:\Z-Stack 3.0.2. Здесь вопросов быть не должно.



## Патч SDK

После установки IAR вы должны применить патч для правильной компиляции исходников.

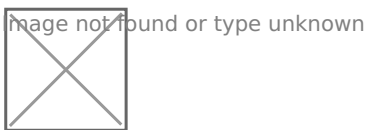
Вы можете найти патч [здесь](#)

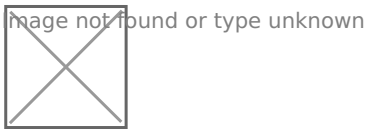
Как установить:

1. Сохраните 0001-Fixes.patch в рабочем каталоге Z-Stack (C:\Z-Stack 3.0.2\)
2. Откройте консоль в рабочем каталоге Z-Stack
3. Выполните `git apply 0001-Fixes.patch`

Скачиваем исходный код из репозитория Примером будет  
служить [Датчик CO2 Zigbee](#)

и выбираем Code, SSH и нажимаем на значок копировать в буфер обмена:

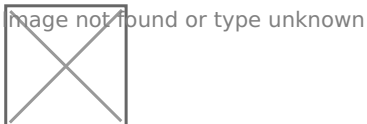




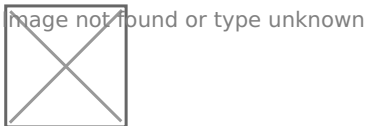
Заходим на компьютере в zstack: C:\Z-Stack 3.0.2\Projects\zstack\HomeAutomation (если вы не меняли путь то вначеле будет папка "Texas Instruments")

правая кнопка мыши, "Git clone..."

Адрес URL проставиться из буфера обмена, директория текущая, обязательно галочка recursive эта поможет стянуть зависимости BME280\_driver и zstack-lib



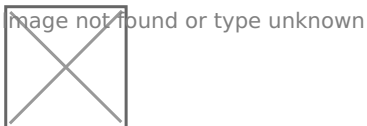
Экран успеха:



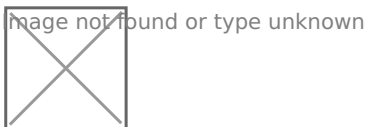
Теперь можно запускать IAR Embedded Workbench. Открываем проект пунктом меню Open

workspace 

Находим нашу папку со скаченным проектом, далее в папке CC2530DB находим файл проекта



Выбираем конфигурацию RouterEB и выполняем Rebuild All.



---

Версия #1

[xyzroe](#) создал Tue, Jan 26, 2021 12:33 PM

[xyzroe](#) обновил Tue, Jan 26, 2021 12:42 PM