# Прошивка кубка

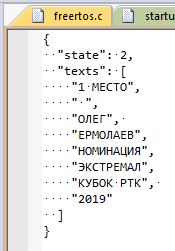
## Подготовка файлов для отправки в кубок

Перед тем как начать заливать информацию в кубок по uart-у, необходимо убедиться, что на компьютере установлен Python3. Версия питона ниже не подойдет. Далее качаем архив с проектом с GitHub, вот ссыль

<https://github.com/RTC-SCTB/CupRtc>

Если качали из мастер-ветки, то в корне есть две директории: keil и Python. Нас интересует второе. В директории Python нас могут заинтересовать два файла: audio.wav и conf.json.

*conf.json*

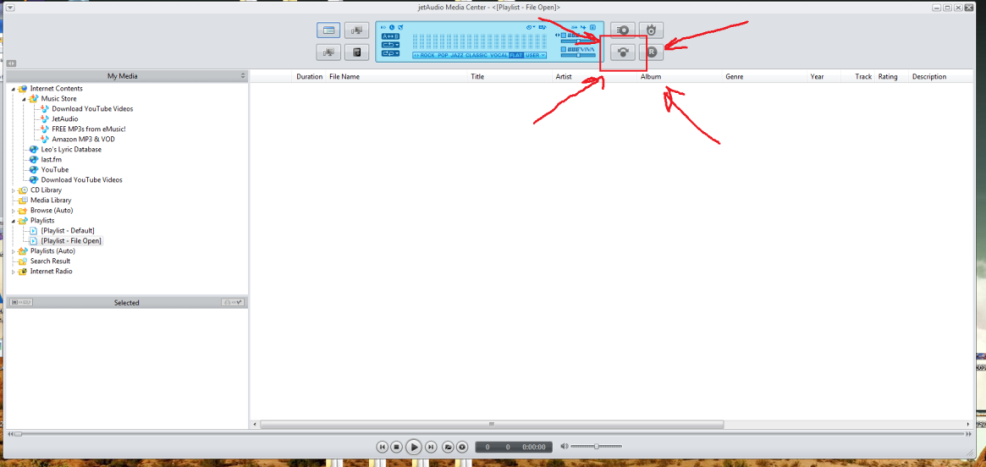
Открываем. Я открыл его в своем случае в keil-е. Советую открывать в каком-нибудь редакторе кода, т.к. можно будет указать кодировку файла, а она должна быть utf-8 (в кейле параметр encode in utf-8 without signature). В этом файле мы прописываем 8 текстов для дисплея и указываем место. Параметр state в данной конфигурации не учувствует, его можно не менять. Тексты записывать нужно из расчета на то, что в строку дисплея поместится 16 символов, могут быть написаны как в кириллице, так и в латинице. Тексты выводятся попарно, в первой и второй строке соответственно (1 и 2, 3 и 4 и т.д.). Заполняем примерно, как показано на скрине.

*audio.wav*

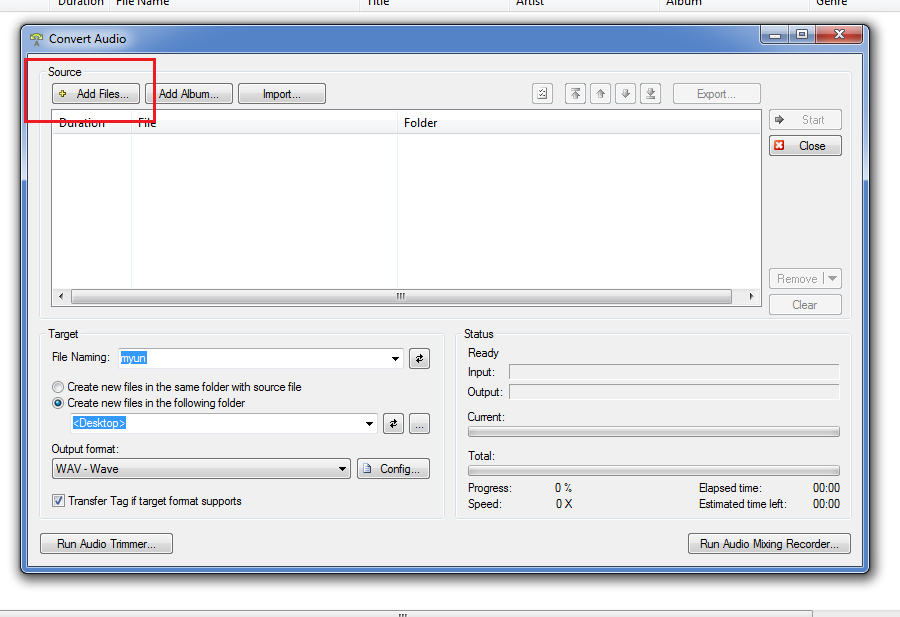
Это наш файл аудио, которое будет транслироваться на динамик.

Для начала нужно установить speex-кодировщик. Откуда я качал, не помню, поэтому в корне проекта лежит архив SpeexACM\_1.0.1.1.zip. Его открываем, и запускаем exe-шник. (Маленькая пометка – делалось все на винде, как на линуксе будет, не известно).

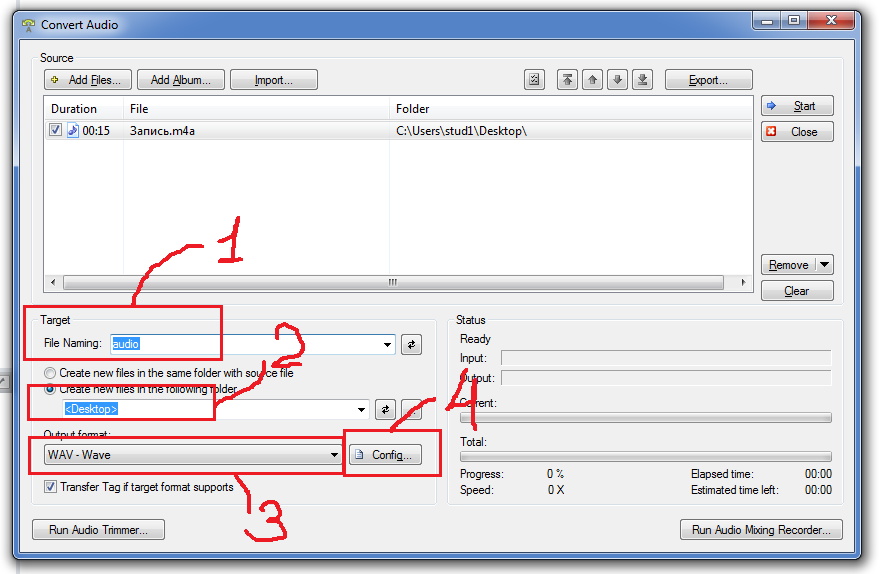
После этого нам нужен какой-нибудь аудио-редактор, я использую jetAudio, он бесплатный, вот ссыль <http://www.jetaudio.com/download/>. Устанавливаем. Открываем, и переходим в окно конвертации.



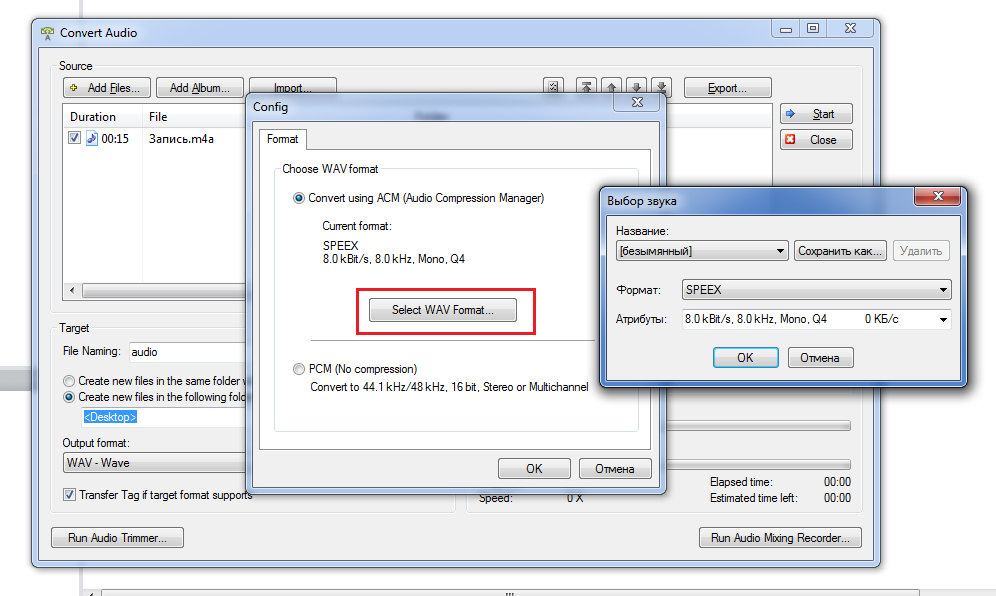
После этого добавляем новый файл в конвертер:



Далее даем название файлу “audio” (1), указываем куда сохранить (2), ставим формат конвертации WAV (3) и заходим в окно конфигурации формата (4).



В окне конфигурации нажимаем select wav format, и выставляем следующие настройки: *SPEEX, 8.0 kBit/s. 8.0kHz, Mono, Q4 0 КБ/с.*



Далее жмем окей и окей и нажимаем Start. Все, файл готов. Закидываем его в папку Python нашего проекта.

## Отправка данных в кубок

Подсоединяем кубок к компьютеру через преобразователь usb-uart. Выглядит это как то так:



Открываем консоль. В консоли переходим в директорию проекта и папку Python. Например командой: *cd Desktop\CupRtc\Python.* Далее прописываем следующую команду, запускающую наш файл loader.py:

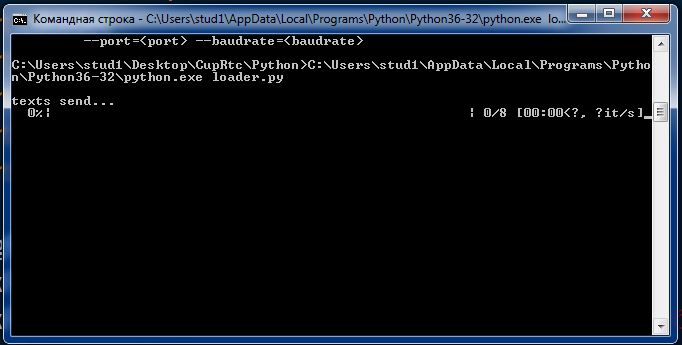
*C:\Users\stud1\AppData\Local\Programs\Python\Python36-32\python.exe loader.py*

Первая часть команды это путь до файла python.exe*.*

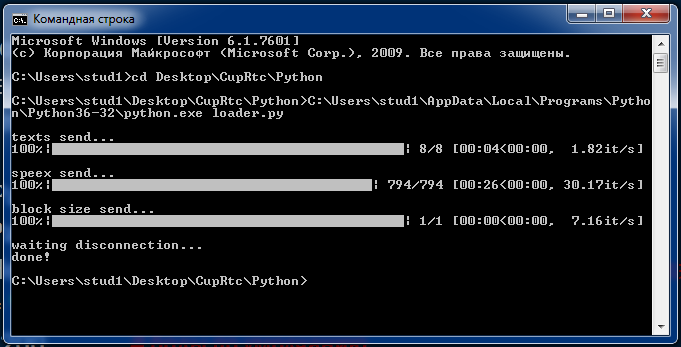
Есть следующие параметры запуска:

* --help – даст справку о параметрах;
* -с – имя файла json c текстами. По умолчанию conf.json. Можно править просто этот файл.
* -s – имя файла speex. По умолчанию audio.wav.
* -p – com-порт, к которому подключен преобразователь. По умолчанию COM3.
* -b – скорость передачи, по умолчанию она 19200, менять ее не надо, т.к. в кубке она тоже 19200.

Исходя из выше сказанного команда может преобразиться в:

*C:\Users\stud1\AppData\Local\Programs\Python\Python36-32\python.exe loader.py –с tezts.json –s voice.wav –p COM5*

После того, как ввели команду, и появилось ожидание отправки текста, зажимаем кнопку на плате кубка и включаем его. Как пошла прошивка, кнопку можно отпустить.

Окончание прошивки выглядит следующим образом. После отсоединяем преобразователь и перезапускаем кубок, проверяем, что все работает.

Все, кубки прошиты, спасибо за внимание ;-)