## 1. Установка зависимостей и настройка системы

Сначала необходимо **проверить/выполнить обновление** **системы**:

sudo apt-get update

sudo apt-get -y upgrade

***Примечание***: флаг **“-y”** отвечает на все вопрос об установке “yes” (если у модуля не проверен сертификат и т.п.).

**Устанавливаем средства** для работы с I2C шиной:

sudo apt-get install i2c-tools

**Выполняем команду:**

sudo nano /etc/modules

И **добавляем модули,** приведенные ниже, в конец файла:

snd-bcm2835

i2c-bcm2835

i2c-dev

rtc-ds1307

**В результате** файл «/etc/modules» должен содержать приблизительно следующее:

**Листинг 1.** Файл “/etc/modules”

# /etc/modules: kernel modules to load at boot time.

#

# This file contains the names of kernel modules that should be loaded

# at boot time, one per line. Lines beginning with "#" are ignored.

**snd-bcm2835**

**i2c-bcm2835**

**i2c-dev**

**rtc-ds1307**

***Примечание***: драйвер **rtc-ds1307** полностью совместим с устройством DS3231.

Для активация программного доступа к I2C-шине в самой системе (используется Raspbian) используем команду:

sudo raspi-config

В открывшемся меню проходим по следующей цепочке:

“Intefacing Option” -> “I2C” -> “Enable? Yes”.

Затем выбираем «**Finish**» и перезагружаемся (sudo reboot). После выполнения указанных команд система будет иметь доступ к МК и RTC-модулю. Теперь необходимо произвести проверку подключения.

## 2. Проверка корректности настроек и подключения

### 2.1. Проверка подключения

После завершения загрузки системы проверяем подключение устройств на I2C-шине. Для этого выполняем команду:

sudo i2cdetect -y 1

***Примечание***: флаг -y указывает I2C шину. Обычно это 0, 1 или 2 шина.

В данном случае RPi предоставляет первую I2C-шину.

**Результат исполнения:**

pi@raspberrypi:~ $ **sudo i2cdetect -y 1**

0 1 2 3 4 5 6 7 8 9 a b c d e f

00: -- -- -- -- -- -- -- -- -- -- -- -- --

10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

20: -- -- -- -- -- -- **26** -- -- -- -- -- -- -- -- --

30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

50: -- -- -- -- -- -- -- **57** -- -- -- -- -- -- -- --

60: -- -- -- -- -- -- -- -- **68** -- -- -- -- -- -- --

70: -- -- -- -- -- -- -- --

**0x68** – это статический шестнадцатеричный адрес RTC таймера на i2c-шине. На используемом таймере ZS-042 возможно изменение i2c-адреса посредствам физического соединения (пайки) перемычек A0, A1 и/или A2. В таком случае адресе варьируются 3 младших бита.

**0x57** – это статический шестнадцатеричный i2c-адрес AT24C32-блока энергонезависимой памяти (EEPROM). Обязательная часть RTC-модуля, которая используется модулем DS3132.

0x26 – это адрес МК, установленный им п

***Примечание***: в данной работе рассматривается взаимодействие только с DS3132.

### 2.2. Проверка зависимостей и корректности настроек

На данный момент отображается адрес 0x68 – это не только указывает адрес устройства, но и говорит о том, что устройство не обслуживается никаким из драйверов, хотя необходимые драйверы уже подгружены в ядро. **Теперь необходимо указать системе, что новое устройство по адресу 0х68 – это RTC-часы, и для его обслуживания можно использовать модуль ядра «rtc-ds1307».**

Для этого выполним команду:

sudo /bin/sh -c "/bin/echo **ds1307** **0x68** > /sys/class/i2c-adapter/i2c-1/**new\_device**"

Теперь проверим как отображается устройство на шине:

pi@raspberrypi:~ $ **sudo i2cdetect -y 1**

0 1 2 3 4 5 6 7 8 9 a b c d e f

00: -- -- -- -- -- -- -- -- -- -- -- -- --

10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

20: -- -- -- -- -- -- **26** -- -- -- -- -- -- -- -- --

30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

50: -- -- -- -- -- -- -- **57** -- -- -- -- -- -- -- --

60: -- -- -- -- -- -- -- -- **UU** -- -- -- -- -- -- --

70: -- -- -- -- -- -- -- --

Теперь адрес устройство по адресу 0x68 (см.пересечении строки 60 и столбца 8) имеет значение UU – это означает, что RTC таймер DS3231 контролируется подгруженным модулем ядра (и это хорошо). Именной такой результат должен быть.

Теперь для окончательной проверки успешности всех подготовительных настроек, выполним команду получения системного времени и времени RTC-модуля

pi@raspberrypi:~ $ **sudo date; sudo hwclock -r**

Fri 26 May 14:55:58 UTC 2017

Fri 26 May 2017 14:55:58 UTC -0.027346 seconds

***Примечание***: время не обязательно должно совпадать, т.к. это только проверка связи с модулем (в данном случае на модуле уже установлено точное время).

### 3. Установка и запуск

Для регистрации и активации демонов выполнить:

**а) rtc**

sudo systemctl enable rtc # **добавить** rtc **в автозапуск**

sudo systemctl start rtc # **запустить** созданную службу

**б) btn\_sound**

sudo systemctl enable cerebro # **добавить** rtc **в автозапуск**

sudo systemctl start cerebro # **запустить** созданную службу

**в) cerebro**

sudo systemctl enable btn\_sound # **добавить** rtc **в автозапуск**

sudo systemctl start btn\_sound # **запустить** созданную службу