

Инструкция по сборке и настройке модулей
источника смещения компании ЗАО "СКОНТЕЛ".
Префикс CU4.
Предназначено для внутреннего использования.
Rev. 0.2

Ожегов Роман Викторович

27 апреля 2018 г.

Оглавление

Предисловие.	2
Необходимое оборудование, инструменты и программное обеспечение	3
Оборудование	3
Программное обеспечение	3
Проверка и настройка модуля управления (CU4CU)	4
Проверка блока питания (CU4PS) и коммутационной платы (CU4CP)	8
Проверка и настройка источника смещения однофотонного детектора (CU4SD)	10
Проверка питания и первоначальная прошивка модуля CU4SD.	10
Проверка цепей постоянного смещения и калибровка CU4SD.	12
Проверка ВЧ цепей и счетчика импульсов CU4SD.	15
Проверка и настройка модуля измерения температуры и давления (CU4TD)	16
Проверка питания и первоначальная прошивка модуля CU4TD.	16
Калибровка модуля CU4TD.	17
Приложение 1. Подготовка рабочей станции для разработки ПО для RaspberryPi 3.	20
Приложение 2. Настройка устройства при помощи стандартного менеджера пакетов.	29
Приложение 3. Краткое описание debian пакетов.	30
Приложение 4. Работа с debian пакетами на Raspberry Pi.	32

Предисловие

Итак, перед Вами лежат разрозненные блоки устройства смещения SSPD. Что с ними делать и как их настраивать вопрос, который у меня возникает каждый раз, когда я вновь возвращаюсь к разработке данного устройства. В попытке структурировать шишки, набиваемые каждый раз при возвращении к разработке, и был создан данный документ.

Надеюсь, что документ будет претерпевать многочисленные исправления со временем, и самое главное, что этот документ будет читаться не только мной, но и другими людьми, участвующими в настоящем и будущем в сборке, настройке, обслуживании устройств. Начинаю писать этот документ со взглядом полным надеждой на долгую историю данного устройства.

На момент написания первой версии данного текста (май 2017) устройство представляет из себя набор модулей:

- CU4CU - модуль ControlUnit - модуль управления всеми остальными блоками устройства. Существующие модификации: CU4CUM0 (на момент написания описания в версии V1). На борту устройства находится микрокомпьютер RaspberryPi3, с помощью которого осуществляется связь с внешним миром, а также управление отдельными модулями устройства;
- CU4PS - блок питания. Существующие модификации: CU4PSM0 (на момент написания описания в версии V2);
- CU4CP - коммутационная плата, соединяющая все модули с модулем управления. Также присутствуют разъемы для подключения блока питания. Существуют модификации: CU4CPM0(для подключения 6 модулей одновременно: CU+TD+4SD на текущий момент в версии V1) и CU4CPM1(для подключения 4 модулей одновременно: TD+2SD+CU на текущий момент в версии V1R1);
- CU4SD - собственно сам модуль смещения однофотонного детектора. Существующие модификации CU4SDM0 (на текущий момент в версии V2R1);
- CU4TD - модуль измерителя температуры. Существующие модификации CU4TDM0 (на текущий момент в версии V1R1).

Настоящая инструкция предназначена для последовательной настройки, проверки работоспособности и калибровки всех модулей устройства. Повествование документа не подразумевает строгой последовательности действий. Отладку и настройку каждого из модулей можно проводить независимо.

Необходимое оборудование и программное обеспечение

Оборудование

Для работы нам потребуется следующее оборудование (список может быть уточняться и дополняться):

- Блок питания 5 В, 2 А. Предназначен для питания модуля управления CU4CU (Matrix годится).
- Блок питания +5 В, -5 В, +15 В. Предназначен для питания модуля смещения однофотонного детектора CU4SD (тот же Matrix годится).
- Мультиметр Agilent 34403A, предназначенный для калибровки модуля смещения однофотонного детектора CU4SD.
- GPIB-USB-HS или GPIB-USB-HS+ – адаптер GPIB интерфейса, предназначенный для подключения мультиметра Agilent 34403A к персональному компьютеру.
- Преобразователь интерфейсов USB-RS485 (UCON-485). Предназначен для подключения шины данных модулей к персональному компьютеру. **В будущем, после разработки калибровочного модуля, эта штука будет не нужна.**
- Программатор ST-Link V2 для программирования микроконтроллеров STMicroelectronics. Программатор я покупал на Aliexpress: [ссылка на Aliexpress](#)

Программное обеспечение

Для работы нам понадобится следующее программное обеспечение (список может быть уточняться и дополняться):

- ST-LINK Utility: набор утилит от STMicroelectronics для работы с микроконтроллерами, включая драйвера программатора и программы, позволяющие программировать микроконтроллеры. Ссылка: [STSW-LINK004](#). Для скачивания необходима регистрация на сайте.
- [Win32 Disk Imager](#) для клонирования образов диска с "малины" на ПК и обратно.
- Утилиты от меня. Названия нет, ссылок нет, населена роботами.

Проверка и настройка модуля управления (CU4CU)

Начать я хотел бы как ни странно не с блока питания, а с модуля управления (бортового компьютера), поскольку его можно разценивать как независимое устройство.

Подготовительные работы

1. Прежде всего необходимо подключить питание к плате, используя стандартный 96-пиновый DIN41612 разъем. Распиновка разъема показана на рисунке [1](#). Фактически необходимо "-" источника питания подключить в 6 линии контактов снизу разъема, а "+" к пятой. На разъем питания подается +5 В. При включении питания светодиод D2 должен загореться красным.
2. Далее подключаем программатор ST-Link V2 к разъему программирования. Распиновка разъема показана на рисунке [2](#).
3. При помощи ST-Link Utility прошиваем устройство актуальной прошивкой **CU4CUxx.bin**, где **xx** - номер модификации и версии соответствующий модулю управления. В случае успеха закорачивание 57 пина разъема DIN41612 (рисунок [1](#)) на землю (P_GND) приводит к характерному щелчку переключения твердотельного реле и загоранию зеленого светодиода D5.
4. Записываем актуальную версию образа диска с помощью программы **Win32 Disk Imager** (этот вариант работает в ОС Windows). Для этого выполняем следующие действия:
 - Жмем на кнопку с папкой и выбираем образ диска. В моем случае это был 2017-04-10-raspbian-jessie-lite.img.
 - Далее выбираем правильный диск, куда будем записывать образ. В моем случае – это G.
 - Жмем на кнопку Write и ожидаем 2-3 минуты (размер файла 1.3 ГБ).
 - Достаем флэшку и устанавливаем ее в Raspberry Pi.
5. Подключаем к USB клавиатуру к Raspberry Pi, кабель Ethernet, монитор через HDMI разъем и запускаем устройство. Изначальные настройки:
 - **Login:** pi
 - **Password:** raspberry

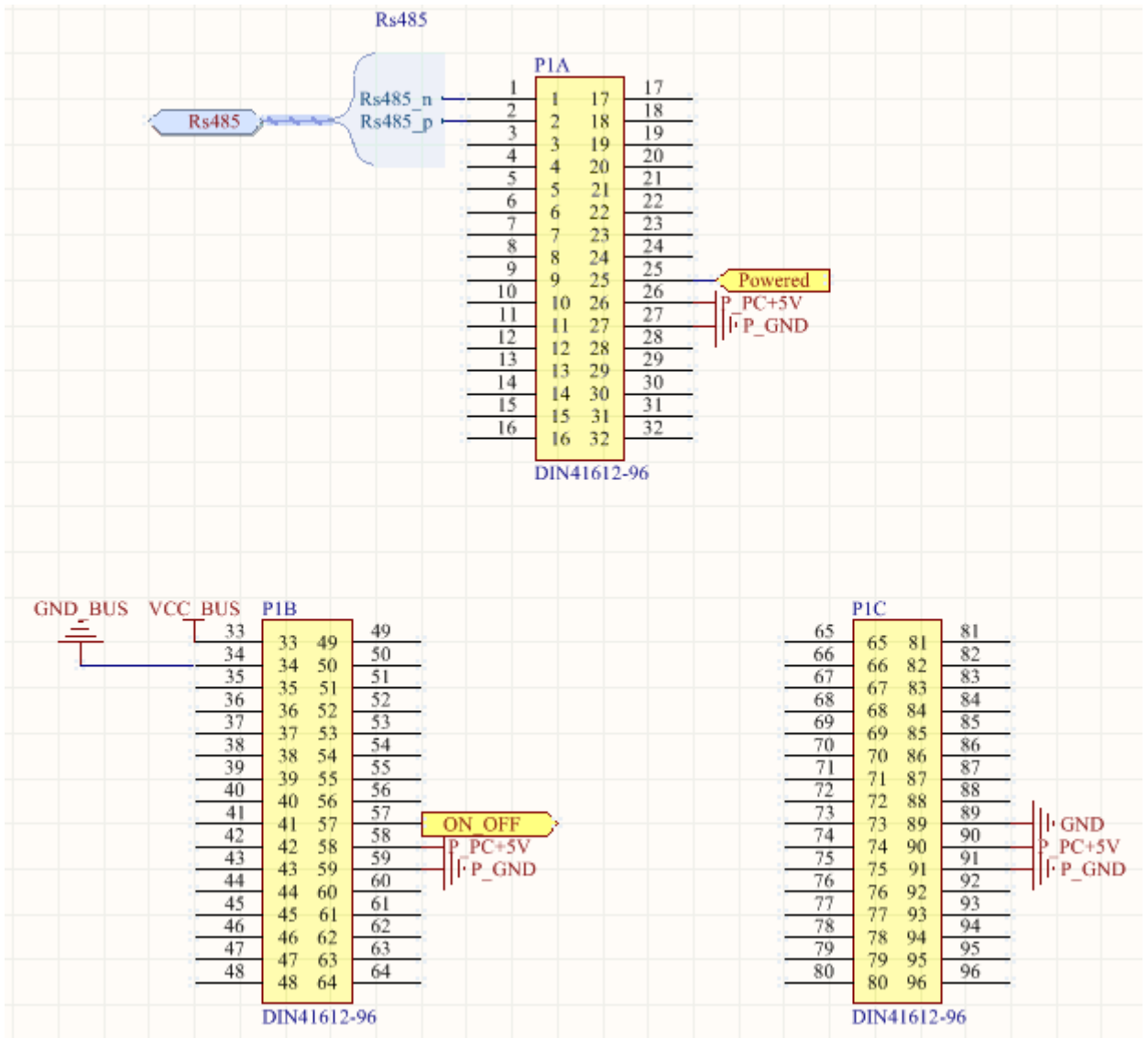


Рис. 1: Распиновка 96-пинового разъема DIN41612 модуля управления CU4CU

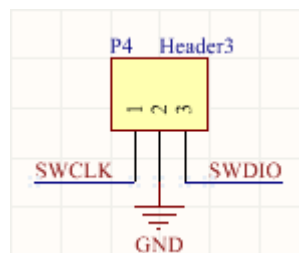


Рис. 2: Распиновка SWD разъема программирования модуля управления CU4CU

6. Настраиваем Raspberry Pi

```
> sudo raspi-config
```

Делаем следующие настройки:

- 1. Меняем пароль системы на scontel
 - 2. Меняем имя компьютера на ControlUnit4
 - 3. **Boot Options->B2** выставляем в No, т.е. не надо дожидаться пока установится интернет соединение
 - 5. **Interfacing Options->P4 SPI** Включаем SPI.
 - 5. **Interfacing Options->P6 Serial** Включаем Uart и подтверждаем. Последовательность ответов: Would you like a login shell to be accesible over serial (Консоль в Uart) -> No. Would you like the serial port hardware to be enabled? (Включение Uart) -> Yes.
 - 7. **Advanced Options ->A1** Расширяем файловую систему на весь объем диска.
 - 7. **Advanced Options ->A2** Включаем поддержку мониторов с Overscan.
 - 7. **Advanced Options ->A3** 256МБ для GPU.
7. Переустанавливаем openssh сервер для того, чтобы в дальнейшем можно было работать с raspberrry pi удаленно по ssh протоколу

```
> sudo apt-get purge openssh-server  
> sudo apt-get update  
> sudo apt-get install openssh-server
```

8. Выясняем TcpIp адрес, чтобы далее работать уже со стационарного компьютера при помощи утилиты **SmartTTY**:

```
> ifconfig
```

В моем случае адрес был 192.168.255.197

9. Производим настройку репозитория, откуда будем брать программы и прочее:

```
> sudo nano /etc/apt/sources.list
```

В конце файла добавляем следующие строки:

```
> deb http://rplab.ru/~ozhegov/.aptly/public/ jessie main
```

После этого выходим из редактора с сохранением файла.

10. Устанавливаем GPG ключ для правильной аутентификации пакетов, устанавливаемых с моего сервера

```
> wget http://rplab.ru/~ozhegov/gpg-key.key
> sudo apt-key add gpg-key.key
> rm gpg-key.key
```

11. Далее обновляем и апгрейдем устройство

```
> sudo apt-get update && sudo apt-get upgrade -y
```

Полный апгрейд системы может занять достаточно большое время, наберитесь терпения.

12. После данной процедуры можно начинать процесс установки программного обеспечения

```
> sudo apt-get install cu-software
```

13. В процессе установки будет вызван диалог конфигурации пакета, в котором настраивается включение при запуске различных программ пакета, а также делается правильный вызов программ настройки пакета: калибровка экрана, подключение-отключение новых устройств и т.д.

14. На этом настройку CU4CU модуля можно считать завершенной. Модуль закрывается и устанавливается в корпус прибора.

Проверка блока питания (CU4PS) и коммутационной платы (CU4CP)

1. Для проверки блока питания (CU4PS) и платы коммутации (CU4CP) необходимо, чтобы модуль управления (CU4CU) был протестирован, и установлен в корпус прибора.
2. Задняя крышка прибора снимается, подключается сетевое питание и нажимается кнопка питания на задней панели прибора. В случае исправности блока питания загорится красный светодиод модуля управления.
3. Далее необходимо нажать на кнопку питания на передней панели прибора. В случае правильного подключения кнопки включения на модуле управления загорится зеленый светодиод.
4. Спустя 10-20 секунд должен прозвучать короткий щелчок, означающий переключение реле на плате коммутации. После этого момента необходимо измерить контрольные напряжения на плате коммутации. Контрольные напряжения представлены на рисунке 3. Цветом показаны различные группы развязанных друг с другом напряжений. Расхождение напряжений питания допустимо в пределах 5-10%.
5. На этом тестирование блока питания и коммутационной платы можно считать завершенным. Прибор можно собрать, установить на место заднюю крышку.



Рис. 3: Контрольные измерительные точки платы коммутации CU4CP

Проверка и настройка источника смещения однофотонного детектора (CU4SD)

Проверка и настройка источника смещения включает в себя несколько этапов: проверка питания, прошивка, калибровка, проверка усилителей и счетчиков

Проверка питания и первоначальная прошивка модуля CU4SD.

1. Подключаем разъем питания: DIN-41612 96pin. Используем стандартный блок питания с биполярным источником питания (Matrix годится). Нам нужно +5/0/-5V. Подключить питание нужно к пинам разъема, как это показано на рисунке 4.
2. Включаем источник питания. Проверяем следующее:
 - Светодиод D7(Power) горит ярко красным.
 - Напряжение на предохранителе F1 – +5V относительно корпуса.
 - Напряжение на предохранителе F2 – -5V относительно корпуса.
 - Ток потребления положительного напряжения порядка 40–100 мА.
 - Ток потребления отрицательного напряжения порядка 10 мА.
3. Подключаем программатор ST-Link V2 к разъему P7. Пины разъема маркируются на плате, начиная с версии плат CU4SDM0V2R1. С помощью программы **STM32 ST_Link Utility** прошиваем модуль прошивкой SspdBlank.bin. Данная прошивка включает в себя Bootloader (автозагрузчик) версии 0.4, а также прошивку CU4SDM0V2.bin также версии 0.4. Данная связка позволяет в будущем обновить рабочую прошивку до актуальной версии.
4. Блокируем память устройства на чтение. Для этого в программе **STM32 ST_Link Utility** выбираем пункт меню **Target->Option Bytes**. Далее в выпадающем списке **Read Out Protection** выбрать **Enabled**. Выполнение данной операции приводит к полной блокировке на чтение устройства через программатор. После этого становится невозможным.

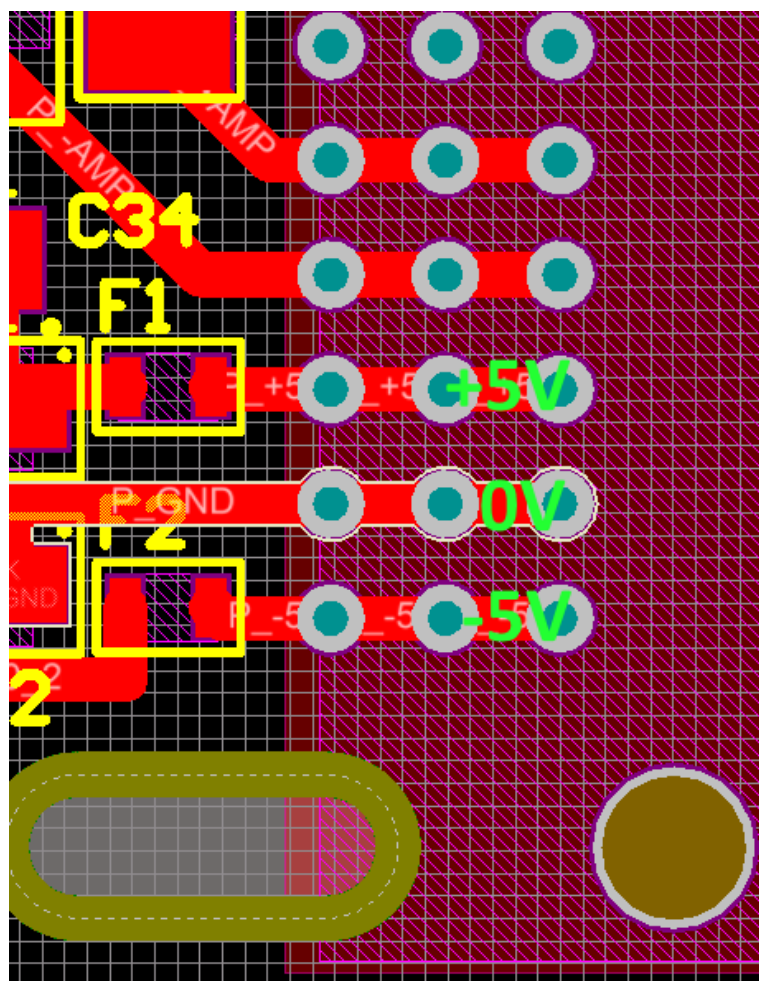


Рис. 4: Подключение питания к модулю CU4SD

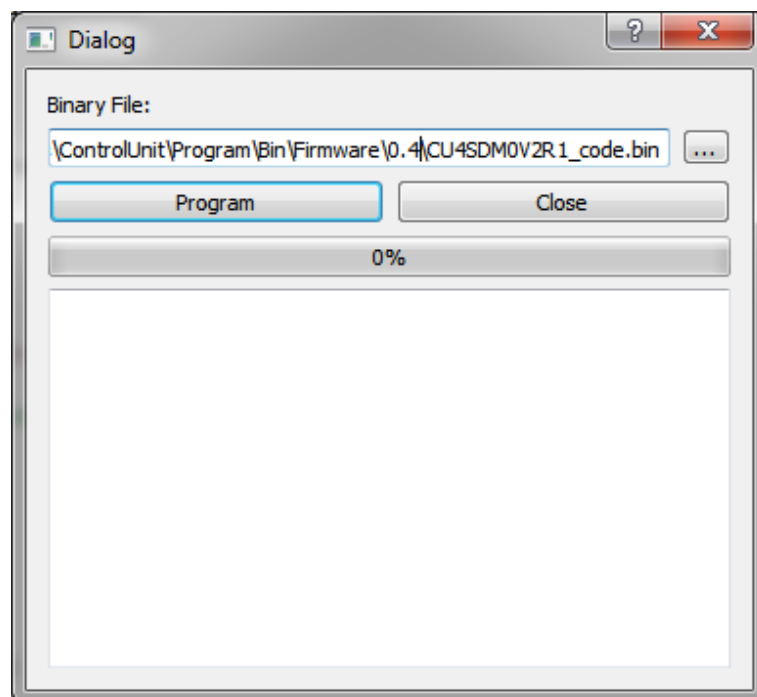


Рис. 5: Внешний интерфейс программы автозагрузчика.

5. При перезагрузке устройства первоначально включается Bootloader. При этом он моргает лампочкой D10 (Uart_Send на плате или CMD на передней панели) 5 раз в течении 5 секунд. Далее программа переключается в штатный режим.
6. Необходимо обновить прошивку модуля до последней версии. Делается это с помощью утилиты Bootloader из набора моих утилит. Внешний вид программы показан на рисунке 5. В поле редактирования указываем путь к закодированной актуальной версии нашего устройства. Также выбор нужного файла прошивки можно произвести нажав на кнопку "...", что приведет к вызову стандартного диалога выбора файла. Далее для старта автообновления прошивки необходимо нажать на кнопку "Program".

Проверка цепей постоянного смещения и калибровка CU4SD.

1. Вооружившись мультиметром Agilent 34401A, подключенным с помощью USB-GPIB кабеля к компьютеру, а также калибруемым модулем CU4SD, подключенным с помощью переходника USB-RS485 к компьютеру, запускаем программу CU4SD_Calibration, являющуюся частью набора утилит для работы с устройством.
2. При запуске программы будет вызван диалог, предлагающий выбор COM-порта, к которому подключена шина данных калибруемого модуля. Внешний вид программы представлен на рисунке 6.

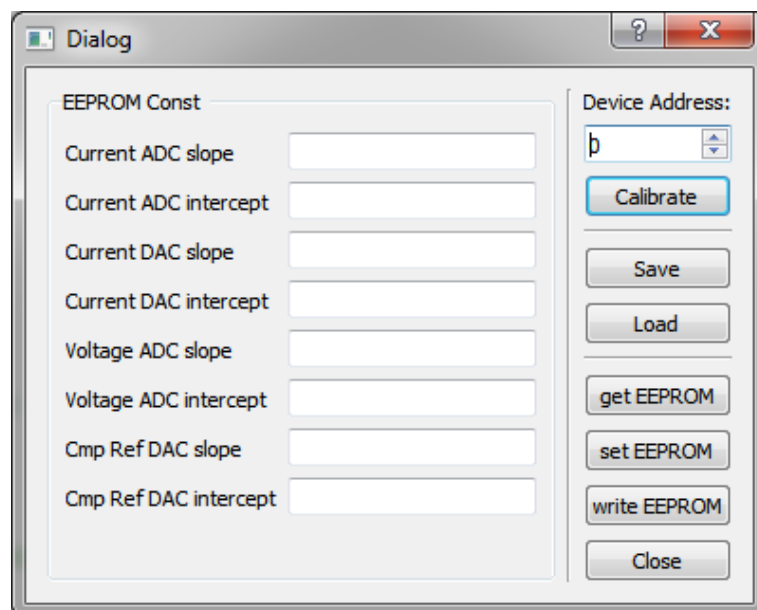


Рис. 6: Внешний вид программы калибровки модуля CU4SD

3. Устанавливаем адрес используемого устройства в 0 (0 адрес используется, когда шина адреса никуда не подключена). Жмем на кнопку **getEEPROM**. Для пустого устройства поля редактирования принимают значение "nan"
4. Далее необходимо нажать на кнопку **"Calibrate"**. Нажатие на эту кнопку приводит к открытию окна калибровки, показанному на рисунке 7.
5. Для калибровки измерителя напряжения подключите кабелем вход модуля **IN** к потенциальным входам мультиметра. Вход мультиметра необходимо зашунтировать через сопротивление порядка 10 кОм. После этого необходимо выбрать из выпадающего списка выбора режима калибровки **U mode**, установить правильный GPIB адрес и нажать на кнопку **Start**. Необходимо обеспечить, чтобы данные не насыщались на краях диапазона. Данные в статусной строке выводятся в следующем формате: "Status: I: x/y, U: u, Agilent: v", где x – значение АЦП канала измерения тока, y – устанавливаемое значение ЦАП, u – значение АЦП канала измерения напряжения, v - реальные показания мультиметра. На графике отображаются следующие кривые: красным показана зависимость устанавливаемого напряжения от значения ЦАП (нижняя горизонтальная шкала), синим показана зависимость измеряемого напряжения от показаний АЦП (верхняя горизонтальная шкала). Сам процесс калибровки включает в себя следующие фазы:
 - Инициализация калибровочных коэффициентов. Коэффициенты **"slope"** приравниваются 1, коэффициенты **"intercept"** – 0.
 - Установка минимального выводимого тока и ожидание в течении 2 секунд (реально ток не успевает установиться, но это устраняется на последующих этапах обработки данных)
 - Медленная развертка тока от минимального до максимального значения,

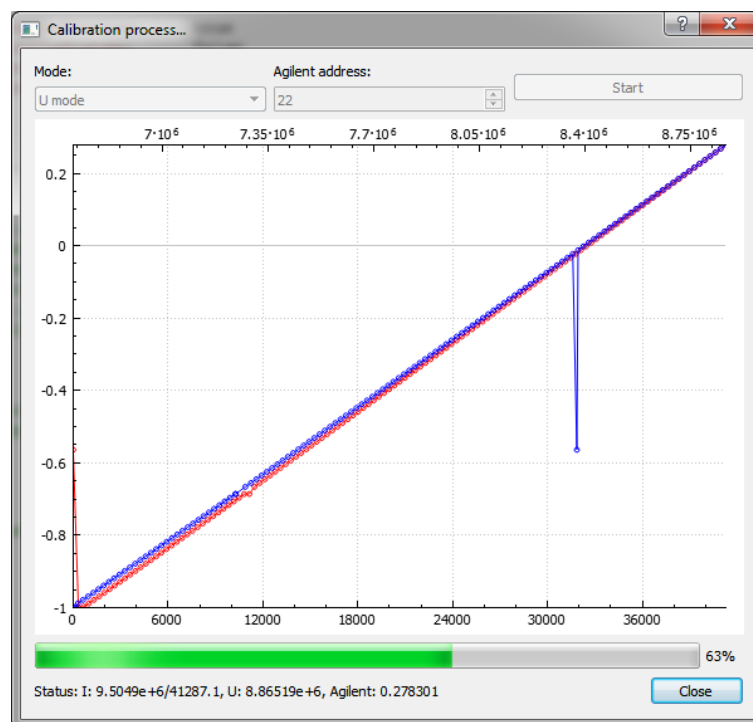


Рис. 7: Процесс калибровки модуля CU4SD

при этом происходят измерения напряжения на мультиметре, а также все напряжения, получаемые АЦП

- Определения коэффициентов линейной регрессии. Удаление 10% "плохих" точек (рекуррентное удаление точек наиболее далеко располагающихся от результата линейной регрессии).
 - Проверка нуля мультиметра. Устанавливается ноль смещения, вычисляются поправки к ЦАП и АЦП и вычисляются новые калибровочные коэффициенты. Данный процесс используется потому, что наиболее важный диапазон смещений для модуля CU4SD располагается вблизи нулевого смещения.
6. Для калибровки источника и измерителя тока подключите кабелем вход модуля **IN** к токовым входам мультиметра. После этого необходимо выбрать из выпадающего списка выбора режима калибровки **I mode**, установить правильный GPIB адрес и нажать на кнопку **Start**. Описание данных и сам процесс измерения соответствует описанному в предыдущем пункте.
 7. Калибровка напряжения уровня компарации не проводится. Устанавливаются следующие значения: $Cmp Ref DAC slope = 65536/2.5 = 26214.4$
 $Cmp Ref DAC intercept = 0$. Приблизительный результат калибровки показан на рисунке 8.
 8. Для сохранения результатов калибровки необходимо сначала нажать на кнопку **"set EEPROM"** после чего нажать на кнопку **"write EEPROM"**.
 9. Для сохранения результатов калибровки в базе данных необходимо нажать на

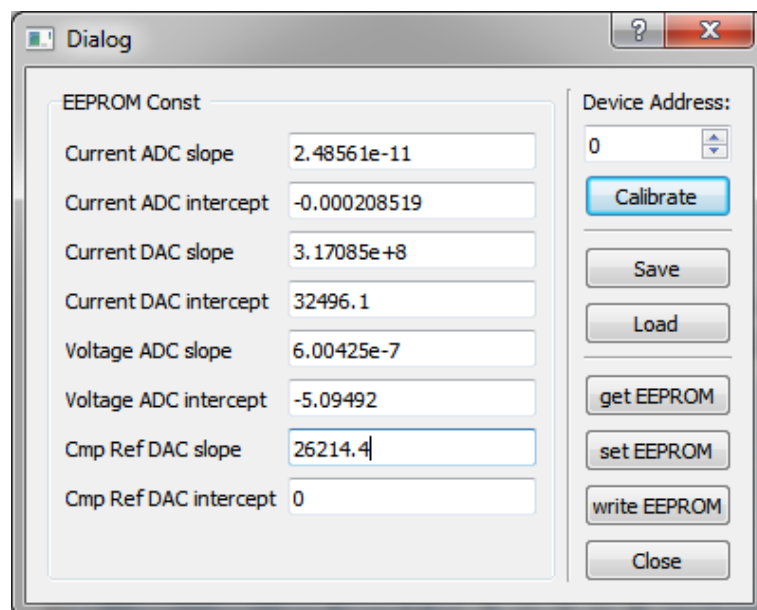


Рис. 8: Приблизительный результат калибровки модуля CU4SD.

кнопку "Save". В диалоговом окне выбора калибровок необходимо выбрать каталог со всеми калибровками. В результате калибровки будут сохранены в каталог, соответствующий типу устройства (например CU4SDM0V2). Сохраняется файл в формате json. Название файла соответствует уникальному идентификационному номеру устройства.

Проверка ВЧ цепей и счетчика импульсов CU4SD.

Пока не написал.

На этом настройку модуля CU4TD можно считать завершенной. Модуль закрывается и устанавливается в корпус прибора.

Проверка и настройка модуля измерения температуры и давления (CU4TD)

Проверка и настройка модуля измерения температуры и давления включает в себя несколько этапов: проверка питания, прошивка, калибровка.

Проверка питания и первоначальная прошивка модуля CU4TD.

1. Подключаем разъем питания: DIN-41612 96pin. Используем стандартный блок питания с биполярным источником питания (Matrix годится). Нам нужно +5/0/-5V. Питание модуля аналогичное модулю CU4SD. Подключить питание нужно к пинам разъема, как это показано на рисунке 4.
2. Включаем источник питания. Проверяем следующее:
 - Светодиод D7(Power) горит ярко красным.
 - Напряжение на предохранителе F1 – +5V относительно корпуса.
 - Напряжение на предохранителе F2 – -5V относительно корпуса.
 - Ток потребления положительного напряжения порядка 40 мА.
 - Ток потребления отрицательного напряжения менее 10 мА.
3. Подключаем программатор ST-Link V2 к разъему P6. Пины разъема маркируются на плате, начиная с версии плат CU4SDM0V2R1. С помощью программы **STM32 ST_Link Utility** прошиваем модуль прошивкой TempBlank.bin. Данная прошивка включает в себя Bootloader (автозагрузчик) версии 0.4, а также прошивку CU4TDM0V1.bin также версии 0.4. Данная связка позволяет в будущем обновить рабочую прошивку до актуальной версии.
4. Блокируем память устройства на чтение. Для этого в программе **STM32 ST_Link Utility** выбираем пункт меню **Target->Option Bytes**. Далее в выпадающем списке **Read Out Protection** выбрать **Enabled**. Выполнение данной операции приводит к полной блокировке на чтение устройства через программатор. После этого становится невозможным.

5. При перезагрузке устройства первоначально включается Bootloader. При этом он моргает лампочкой D3 (Uart Send на плате или CMD на передней панели) 5 раз в течении 5 секунд. Далее программа переключается в штатный режим.
6. Необходимо обновить прошивку модуля до последней версии. Делается это с помощью утилиты Bootloader из набора моих утилит. Внешний вид программы показан на рисунке 5. В поле редактирования указываем путь к закодированной актуальной версии нашего устройства. Также выбор нужного файла прошивки можно произвести нажав на кнопку "...", что приведет к вызову стандартного диалога выбора файла. Далее для старта автообновления прошивки необходимо нажать на кнопку "Program".

Калибровка модуля CU4TD.

1. Вооружившись мультиметром Agilent 34401A, подключенным с помощью USB-GPIB кабеля к компьютеру, а также калибруемым модулем CU4TD, подключенным с помощью переходника USB-RS485 к компьютеру, запускаем программу CU4TD_Calibration, являющуюся частью набора утилит для работы с устройством.
2. При запуске программы будет вызван диалог, предлагающий выбор COM-порта, к которому подключена шина данных калибруемого модуля. После выбора COM-порта откроется основное окно программы, представленное на рисунке 6.
3. Устанавливаем адрес используемого устройства в 0 (0 адрес используется, когда шина адреса никуда не подключена). Жмем на кнопку **getEEPROM**. Для пустого устройства поля редактирования принимают значение "nan". Статус передачи данных – "Status: Ok".
4. Калибровка модуля выполняется в несколько этапов. **Для упрощения процесса калибровки необходима специальная плата для калибровки, на которой будут все необходимые элементы питания, переключатели, а также мультивибратор, с помощью которого будет тестироваться счетчик модуля SD.** Для начала калибровки необходимо нажать на кнопку **"Calibrate"**. Нажатие на эту кнопку приводит к открытию окна калибровки, аналогичному тому, которое было показано на рисунке 7. Отличие заключается лишь в выпадающем списке выбора режима калибровки (**"Mode"**). В данной программе выпадающий список имеет значения: "PressSensorVoltageP", "PressSensorVoltageN", "TempSensorVoltage", "TempSensorCurrent".
5. Для калибровки канала измерения положительного сигнала с датчика давления подсоединяем пины 3(P+) и 6(T+) к положительному потенциальному входу мультиметра Agilent34401A, пин 5 (T-) подключаем к общему контакту мультиметра. Вход мультиметра шунтируем сопротивлением порядка 60 кОм и нажимаем кнопку **"Start"**.
6. Для калибровки канала измерения отрицательного сигнала с датчика давления подсоединяем пины 4(P-) и 6(T+) к положительному потенциальному входу

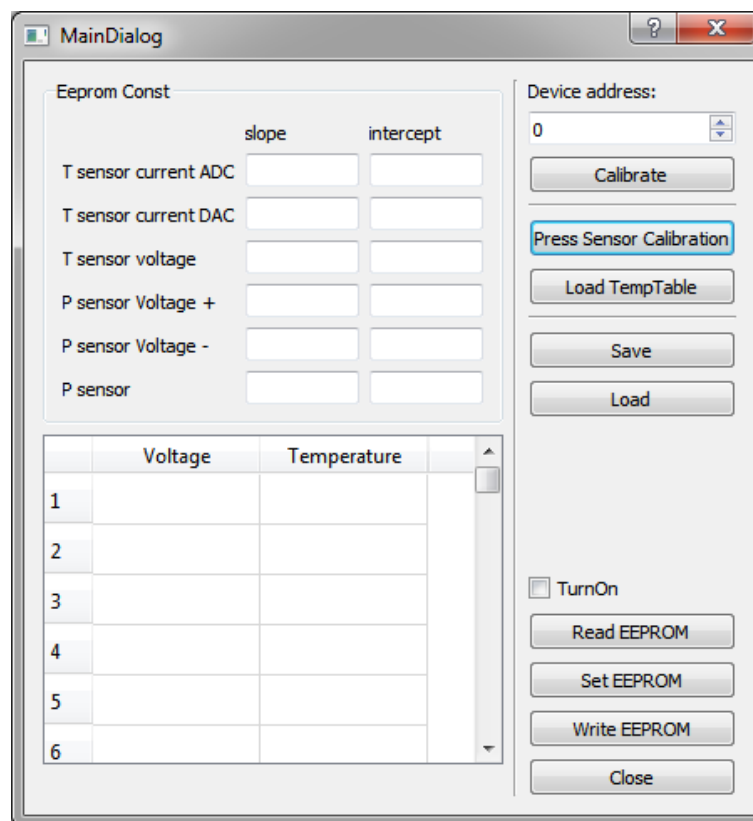


Рис. 9: Внешний вид программы калибровки модуля CU4TD.

мультиметра Agilent34401A, пин 5 (Т-) подключаем к общему контакту мультиметра. Вход мультиметра шунтируем сопротивлением порядка 60 кОм и нажимаем кнопку **"Start"**.

7. Для калибровки канала измерения напряжения с датчика температуры подсоединяем пин 6(Т+) к положительному потенциальному входу мультиметра Agilent34401A, пин 5 (Т-) подключаем к общему контакту мультиметра. Вход мультиметра шунтируем сопротивлением порядка 60 кОм и нажимаем кнопку **"Start"**.
8. Для калибровки канала измерения тока датчика температуры подсоединяем пин 6(Т+) к положительному токовому входу мультиметра Agilent34401A, пин 5 (Т-) подключаем к общему контакту мультиметра и нажимаем кнопку **"Start"**.
9. **Калибровка датчика давления.**
10. После окончания калибровки и закрытия диалогового окна калибровки данные измерения будут выглядеть так как показано на рисунке 10.
11. Для сохранения результатов калибровки необходимо сначала нажать на кнопку **"set EEPROM"** после чего нажать на кнопку **"write EEPROM"**.
12. Для сохранения результатов калибровки в базе данных необходимо нажать на кнопку **"Save"**. В диалоговом окне выбора калибровок необходимо выбрать каталог со всеми калибровками. В результате калибровки будут сохранены в ката-

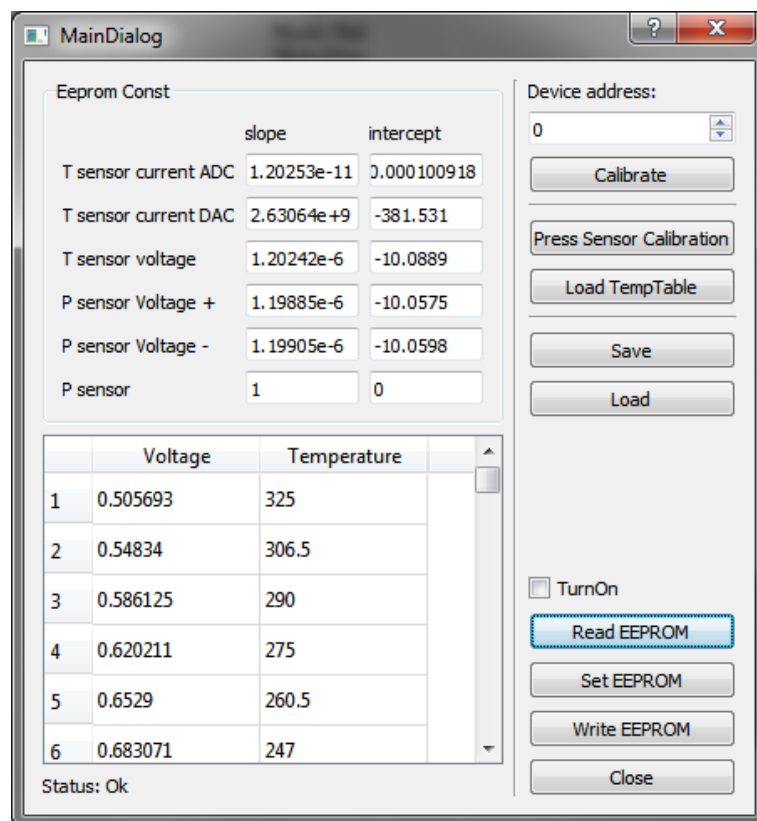


Рис. 10: Примерные результаты калибровки модуля CU4TD.

лог, соответствующий типу устройства (например CU4TDM0V1). Сохраняется файл в формате json. Название файла соответствует уникальному идентификационному номеру устройства.

Приложение 1. Подготовка рабочей станции для разработки ПО для RaspberryPi 3.

Процесс подготовки прошивки для Raspberry Pi Достаточно трудоемок, поэтому рекомендую использовать уже готовые прошивки. При необходимости также можно последовать действиям описанным ниже. Ниже описанные действия предназначены не только для подготовки полноценного образа для Raspberry Pi, но и для подготовки рабочего компьютера для разработки софта для "малины".

Необходимые инструменты

- Образ SD диска для Raspberry. На момент описания этого опуса использовалась сборка Rasbian Jessie Lite от 2017-04-10. [Образы дисков тут](#).
- Операционная система Linux. Использовал стабильную сборку [Xubuntu 16.04LTS](#). Необходим 32-битный вариант операционной системы. Выбрал Xubuntu, потому как на виртуальной машине он веселее чувствуется.
- Виртуальная машина, для работы с Linux из под операционной системы Windows. Использовал [Virtual Box](#) от Oracle.
- [SmartTTY](#) - удобная программа для установки ssh соединения между компьютером под Windows и Raspberry Pi.
- [Win32 Disk Imager](#) для клонирования образов диска с "малины" на ПК и обратно.
- Cross-compile tools for RaspberryPi: [ссылка](#)

Рабочие инструкции:

- <https://wiki.qt.io/RaspberryPi2EGLFS> - подготовка образа диска
- https://xdevs.com/guide/ni_gpib_rpi/ - настройка драйвера GPIB-USB
- [Уменьшение операций чтения-записи Raspberry Pi](#).

Устарело:

- http://wiki.qt.io/RaspberryPi_Beginners_Guide

- <http://www.ics.com/blog/building-qt-5-raspberry-pi>
- https://wiki.qt.io/Qt5_on_RaspberryPi
- http://wiki.qt.io/Building_Qt_5_from_Git

Для виндовс:

- <http://gntoolchains.com/raspberry/tutorial/>
- <http://visualgdb.com/tutorials/raspberry/qt/embedded/>

1. Подготовка рабочей станции (виртуальная машина Ubuntu).

1. Настройка виртуальной машины

Один из самых простых шагов. В виртуальной машине жмем кнопку "Создать" и создаем виртуальную машину с параметрами, указанных на рисунке 11.

После создания виртуальной машины, необходимо запустить ее, при этом появится диалоговое окно, в котором необходимо указать путь к образу LiveCD с операционной системой. Далее необходимо пройти сквозь стандартную процедуру установки Linux, которая, на мой взгляд, не требует пояснений. Весь процесс установки операционной системы занимает не более 15 минут.

После установки Ubuntu следует ее выключить и провести более тонкую настройку виртуальной машины:

- **Общие->Дополнительно->Общий буфер обмена:** Двухнаправленный
- **Система->Материнская плата->основная память:** 4096МБ (Возможный максимум и можно было выбрать еще на этапе установки системы)
- **Система->Материнская плата->Дополнительные возможности:** Включить I/O APIC
- **Система->Процессор->Процессор(ы):** 2
- **Дисплей->Экран->Видеопамять:** 128 МБ

Для полноценной работы необходимо установить дополнение для виртуальной машины. Для этого необходимо запустить виртуальную машину. В меню VirtualBox выбрать **Устройства->Подключить образ диска Дополнений гостевой ОС...** Далее необходимо выполнить команду:

```
> sudo <pathToDisk>\VBoxLinuxAdditions.run
```

После этого необходимо дождаться установки пакетов и перезапустить ОС Linux.

Заключительным этапом настройки является подключение внешней папки со всеми образами для Raspberry Pi. Для этого совершаем следующие действия: **Настройки->Общие папки->+** и указываем путь к папке с образами, а также

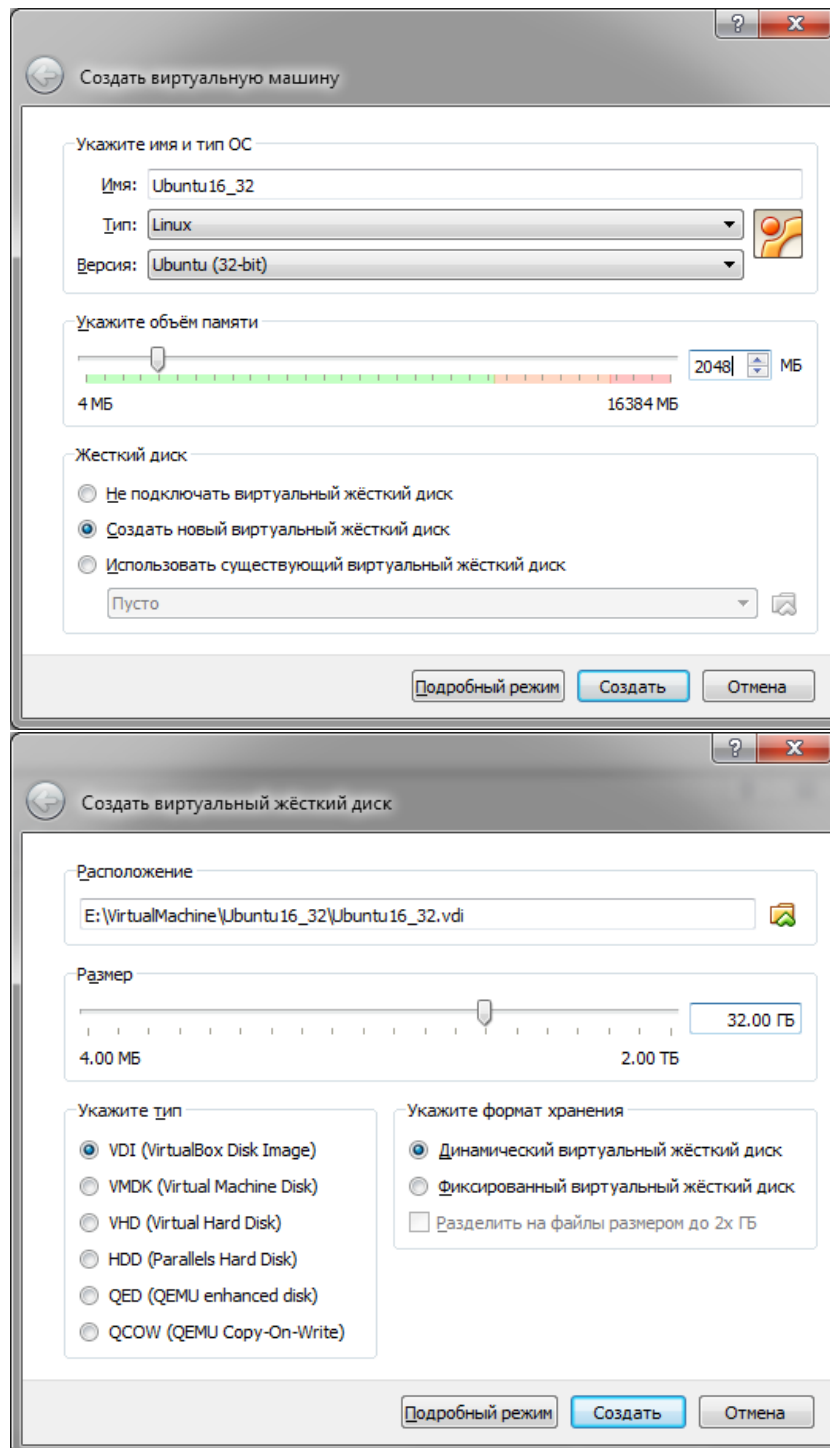


Рис. 11: Настройки виртуальной машины

псевдоним диска. В моем случае псевдоним "RaspberryPi" и это каталог, в котором лежат все необходимые материалы для работы с "малиной". Образы при этом лежат в подкаталоге **Images**. Также можно поставить галку **Авто-подключение** и **Создать постоянную папку**. Также необходимо добавить имя пользователя в число пользователей, которым доступны подмонтированные устройства командой:

```
> sudo usermod <userName> -a -G vboxsf
```

2. Копирование RSA ключей

1. Первоначально добавим путь к каталогу проекта в виртуальную машину. Для этого совершаем следующие действия: **Машина->Настроить....** Далее переходим во вкладку **Общие папки** и нажимаем на кнопку **Добавить**, после чего указываем путь к каталогу с программой проекта, а также псевдоним диска. В моем случае псевдоним "ControlUnitSoftware" и это каталог, в котором лежат все рабочие утилиты. Также можно поставить галку **Авто-подключение** и **Создать постоянную папку**. Необходимо добавить имя пользователя в число пользователей, которым доступны подмонтированные устройства командой:

```
> sudo usermod <userName> -a -G vboxsf
```

Для вступления в силу изменений необходимо перезагрузиться.

2. Копируем RSA ключи

```
> sudo cp /media/sf_ControlUnitSoftware/LinuxScripts/cu4_rsa* /root/.ssh/
&& chown 600 /root/.ssh/cu4_rsa
```

Далее необходимо открыть файл с ключем cu4_rsa.pub и установить правильное имя компьютера в конце файла.

2. Настройка Raspberry Pi.

1. Подготовка флешки. (Вариант для Linux)

1. Делаем копию образа, с которой в дальнейшем и будет происходить вся работа:

```
> cd /media/sf_RaspberryPi/
> cp Images/<ImageName>.img Images/raspberry-working-image.img
```

2. Подключаем флешку (MicroSD 8GB) к виртуальной машине: для чего в меню VirtualBox выбираем **Устройства->USB->Generic USB Storage**. После этой операции диск должен автоматически подмонтироваться.
3. Записываем образ диска на microSD карту:

```
> sudo dd bs=4M if=Images/raspberry-working-image.img of=/dev/sdb
309+1 records in
```



```
309+1 records out
1297862656 bytes (1,3 GB, 1,2 GiB) copied, 816,618 s, 1,6 MB/s
```

Процесс не быстрый занимает более 10 минут.

1. Подготовка флэшки. (Вариант для Windows)

1. Программа очень простая. Жмем на кнопку с папкой и выбираем образ диска. В моем случае это был 2017-04-10-raspbian-jessie-lite.img.
2. Далее выбираем правильный диск, куда будем записывать образ. В моем случае – это G.
3. Жмем на кнопку Write и ожидаем 2-3 минуты (размер файла 1.3 ГБ).
4. Достаем флэшку и устанавливаем ее в Raspberry Pi.

2. Первый запуск Raspberry Pi

1. Вставляем MicroSD флэш карту. Подключаем к USB клавиатуру, кабель Ethernet, монитор через HDMI разъем и запускаем. Изначальные настройки:

- **Login:** pi
- **Password:** raspberry

2. Настраиваем Raspberry Pi

```
> sudo raspi-config
```

Делаем следующие настройки:

- 1. Меняем пароль системы на scontel
- 2. Меняем имя компьютера на ControlUnit4
- 3. **Boot Options->B2** выставляем в No, т.е. не надо дожидаться пока установится интернет соединение
- 5. **Interfacing Options->P4 SPI** Включаем SPI.
- 5. **Interfacing Options->P6 Serial** Включаем Uart и подтверждаем. Последовательность ответов: Would you like a login shell to be accesible over serial (Консоль в Uart) -> No. Would you like the serial port hardware to be enabled? (Включение Uart) -> Yes.
- 7. **Advanced Options ->A1** Расширяем файловую систему на весь объем диска.
- 7. **Advanced Options ->A2** Включаем поддержку мониторов с Overscan.
- 7. **Advanced Options ->A3** 256МБ для GPU.

После всего этого перезагружаемся.

3. Переустанавливаем openssh сервер

```
> sudo apt-get purge openssh-server  
> sudo apt-get install openssh-server
```

4. Выясняем ТсrПr адрес, чтобы далее работать уже со стационарного компьютера:

```
> ifconfig
```

В моем случае адрес был 192.168.255.197

3. Настройка рабочей станции и Raspberry Pi с использованием скрипта установки

Переходим на рабочую станцию и запускаем скрипт raspberryPi_install_step1.sh. После этого ждем. Ждать придется долго. Лог процесса записывается в файл install.log.txt. Скрипт запускается под суперюзером и требует в качестве параметра ТсrПr адрес малины, полученный на предыдущем шаге

```
> sudo ./raspberryPi_install_step1.sh <TCPIP raspberry> $USER
```

При работе требует пароль Raspberry Pi(при первом заходе): scontel и пароль локальной машины

Маленькие добавления не вошедшие в скрипт. Внести ранее

```
$ sudo apt-get install git-core  
If you get any errors here, make sure your Pi is up to date with the latest  
versions of Raspbian: (this is a good idea to do regularly, anyway)  
  
$ sudo apt-get update  
$ sudo apt-get upgrade  
To obtain WiringPi using GIT:  
  
$ cd  
$ git clone git://git.drogon.net/wiringPi  
If you have already used the clone operation for the first time, then  
  
$ cd ~/wiringPi  
$ git pull origin  
Will fetch an updated version then you can re-run the build script below.  
  
To build/install there is a new simplified script:
```

```
$ cd ~/wiringPi
$ ./build
```

Настраиваем QtCreator на рабочем компьютере

Первоначально нужно установить Qt и QtCreator, используя стандартный установщик с сайта производителя. Странно: для 32-битного Linux доступен только Qt5.5, в то время как для Windows уже доступен Qt5.9. Запускаем QtCreator и используем следующие настройки

```
Go to Options -> Devices
  Add
    Generic Linux Device
    Enter IP address, user & password
    Finish
Go to Options -> Build & Run -> Compilers
  Add
    GCC
    Compiler path: ~/raspi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabi-hf-
      raspbian/bin/arm-linux-gnueabi-hf-g++
Go to Options -> Build & Run -> Debuggers
  Add
    Debugger Path: ~/raspi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabi-hf-
      raspbian-x64/bin/arm-linux-gnueabi-hf-gdb
Go to Options -> Build & Run -> Qt Versions
  Check if an entry with ~/raspi/qt5/bin/qmake shows up. If not, add it.

Go to Options -> Build & Run
  Kits
    Add
      Generic Linux Device
      Device: the one we just created
      Sysroot: ~/raspi/sysroot
      Compiler: the one we just created
      Debugger: the one we just created
      Qt version: the one we saw under Qt Versions
      Qt mkspec: leave empty
```

Настройка драйвера GPIB-USB-HS

Начинаем с того, что качаем заголовки ядра для Raspberry Pi.

```
> sudo wget https://raw.githubusercontent.com/notro/rpi-source/master/rpi-source
  -O /usr/bin/rpi-source
> sudo chmod +x /usr/bin/rpi-source
> /usr/bin/rpi-source -q --tag-update
```

Далее выполняем `rpi-source`, для сборки хидеров системного ядра. Данная операция занимает до 5 минут.

```
> sudo rpi-source
```

Устанавливаем все пакеты, которые нам в дальнейшем понадобятся

```
> sudo apt-get install bc
> sudo apt-get install libncurses5-dev
> sudo apt-get install tk-dev
> sudo apt-get install build-essential texinfo texi2html libcwidget-dev
    libncurses5-dev libx11-dev binutils-dev bison flex libusb-1.0-0 libusb-dev
    libmpfr-dev libexpat1-dev tofrodos subversion autoconf automake libtool
> sudo apt-get autoremove
> sudo apt-get upgrade
```

Далее создаем папку для драйверов `gpi-usb`, забираем последнюю версию драйверов из удаленного репозитория, конфигурируем и собираем.

```
> mkdir ~/linux-gpi
> cd ~/linux-gpi
> svn checkout svn://svn.code.sf.net/p/linux-gpi/code/trunk linux-gpi-code
> cd linux-gpi-code/linux-gpi/
> ./bootstrap
> ./configure
> make
> sudo make install
```

Для того, чтобы все заработало необходимо сначала запустить утилиту

```
> sudo ldconfig
```

Далее необходимо поправить файл `/etc/gpi.conf`. Данный файл должен содержать следующий текст:

```
interface {
    minor      = 0          /* board index, minor = 0 uses /dev/gpi0, minor =
        1 uses /dev/gpi1 */
    board_type = "ni_usb_b" /* type of interface board being used */
    name       = "violet"   /* optional name, allows you to get a board
        descriptor using ibfind() */
    pad        = 0          /* primary address of interface      */
    sad        = 0          /* secondary address of interface    */
    timeout    = T30s      /* timeout for commands */
    eos        = 0x0a      /* EOS Byte, 0xa is newline and 0xd is carriage
        return */
    set-reos   = yes       /* Terminate read if EOS */
    set-bin    = no        /* Compare EOS 8-bit */
    set-xeos   = no        /* Assert EOI whenever EOS byte is sent */
    set-eot    = yes       /* Assert EOI with last byte on writes */
    master     = yes       /* interface board is system controller */
}
```

```
}
```

После этого можно запустить

```
> sudo gpib_config
```

Команда должна завершаться без вывода ошибок. Можно поставить ее в автозагрузку, однако, если нет необходимости использовать gpib, то и команда эта не нужна.

После окончания почистим за собой:

```
> rm ~/linux-ef3b440e0e4d9ca70060483aa33d5b1201ceceb8.tar.gz
> rm -r ~/linux-ef3b440e0e4d9ca70060483aa33d5b1201ceceb8
> rm -r ~/linux
```

Сборка проекта и его Deploy

Это еще не все

написать чего-то, чуть позднее. потому что это пока не сделал

Отключаем Raspberry Pi

```
> sudo poweroff
```

Создание копии образа содержимого флэш-карты

Переходим на локальный компьютер (виртуальную машину), подключаем microSD карту и копируем ее содержимое:

```
> cd /media/sf_RaspberryPi
> sudo dd if=/dev/sdb of=Images/raspberry-working-image.img
```

В последний раз данная операция не прошла, выпадало множество ошибок, поэтому я воспользовался [Win32 Disk Imager](#) для создания клона диска. Время копирования достаточно велико. В моем случае составило порядка часа (1:12:20). Придется ждать.

Приложение 2. Настройка устройства при помощи стандартного менеджера пакетов.

1. Первое, что нужно сделать это добавить в список репозиториев наш репозиторий. Для этого в файл с ресурсами

```
>sudo nano /etc/apt/sources.list
```

Добавляем текст

```
#Scontel Soft  
deb http://rplab.ru/~ozhegov/.aptly/public/ jessie main
```

2. Теперь необходимо обновить информацию о репозиториях:

```
> sudo apt-get update
```

3. После этого можно установить наш пакет

```
> sudo apt-get install cu-software
```

4. Для обновления пакетов, в том числе и связанных с cu-software необходимо выполнить команды

```
> sudo apt-get update && sudo apt-get upgrade -y
```

5. Для удаления пакетов, в том числе и связанных с cu-software необходимо выполнить команды

```
> sudo apt-get remove cu-software && sudo apt-get update && sudo apt-get  
autoremove
```

6. Заново настроить пакет, провести перереконфигурацию пакета можно с помощью пакета dpkg

```
> sudo dpkg-reconfigure cu-software
```

Приложение 3. Краткое описание debian пакетов.

cu-software

cu-software - основной пакет, включающий все остальные пакеты. Функции выполняемые пакетом:

- Конфигурация устройства
- Запуск сервисов cu-tcpsrserver, cu-embeddeddisplay, cu-simpleapp

cu-powercontrol

cu-powercontrol - пакет, обеспечивающий управление питанием устройством. Функции выполняемые пакетом:

- Установка публичного ключа cu4_rsa.pub в каталог /root/.ssh и добавление его в авторизированные ключи.
- Отключение Bluetooth и подключение за место его Uart.
- Отключение логотипа с экрана
- Отключение иконки предупреждающих сообщений с экрана монитора
- Отключение логирования
- Отключение автозасыпания
- Включение сервиса, слушающего нажатие кнопки питания, по которой происходит автоотключение устройства

cu-tcpsrserver

cu-tcpsrserver - пакет, устанавливающий одноименную утилиту, предназначенную для работы. Запускает сервер, который слушает команды по порту 9876 и позволяет управлять каждым модулем по отдельности.

cu-displaycalibration

cu-displaycalibration - пакет, устанавливающий одноименную утилиту, предназначенную для калибровки экрана, параметры калибровки далее используются пакетом cu-embeddeddisplay.

cu-embeddeddisplay

cu-embeddeddisplay - пакет, устанавливающий одноименную утилиту, предназначенную для работы с устройством через встроенный экран. Приложение может работать как по TspIp протоколу, так и через последовательный порт. Выбор протокола происходит через конфигурацию пакета cu-software или путем редактирования файла /etc/default/cu-software.

cu-simpleapp

cu-simpleapp - пакет, устанавливающий одноименную утилиту, предназначенную для работы с устройством при помощи стандартных устройств ввода-вывода (клавиатура, мышь, стандартный монитор, подключенный к HDMI порту устройства). Приложение может работать как по TspIp протоколу, так и через последовательный порт. Выбор протокола происходит через конфигурацию пакета cu-software или путем редактирования файла /etc/default/cu-software.

Приложение 4. Работа с debian пакетами на Raspberry Pi.

Для удобства работы, установки программного обеспечения и дальнейшего его обновления на Raspberry Pi поддерживается работа через стандартную для Linux утилиту apt-get. Для работы этой утилиты должен быть создан специальный репозиторий, находящийся по адресу [ссылка](#). В этом разделе приведено описание как создать репозиторий, как поместить его на сервер, как обновить версию пакета на сервере, как подготовить устройство для получения данных с сервера, как установить и обновить пакет на устройстве.

По-хорошему, вся эта информация должна учитываться в предыдущем разделе

Создание своего пакета

Ниже приведено очень короткое описание, позволяющее создать простой пакет, который при проверке будет показывать ошибки, но позволяет обновлять файлы с сервера, что мне и нужно. Для более детального описания смотри [тут](#). Далее необходимо пакеты сделать более качественными и устранить все ошибки.

1. Итак создаем пакет. **Ubuntu-forum**. Название пакета устанавливаем следующим:

```
<project>_<major version>.<minor version>-<package revision>
```

Для примера название тестового пакета будет:

```
helloworld_1.0-1
```

2. Создаем директорию, в которой будем складывать все файлы пакета

```
> mkdir helloworld_1.0-1
```

3. Все файлы помещаются по папкам внутри данного каталога. Для примера:

```
> mkdir helloworld_1.0-1/usr
> mkdir helloworld_1.0-1/usr/local
> mkdir helloworld_1.0-1/usr/local/bin
> cp "~/Projects/Hello World/helloworld" helloworld_1.0-1/usr/local/bin
```

4. Далее создаем файл метаданных, необходимый для компиляции пакета.

```
> mkdir helloworld_1.0-1/DEBIAN
> gedit helloworld_1.0-1/DEBIAN/control
```

В него помещаем примерно следующий текст

```
Package: helloworld
Version: 1.0-1
Section: base
Priority: optional
Architecture: i386
Depends: libsomethingorrather (>= 1.2.13), anotherDependency (>= 1.2.6)
Maintainer: Your Name <you@email.com>
Description: Hello World
  When you need some sunshine, just run this.
  small program!.
.
(the space before each line and dot after each line in the description is
  important).
```

5. После этого собираем пакет

```
> dpkg-deb --build helloworld_1.0-1
```

Размещение пакета на ftp сервере

Тут все просто, просто копируем собранный пакет на сервер

```
scp helloworld_1.0-1.deb ozhegov@rplab.ru:
```

Добавление пакета в репозиторий

На данном этапе я пока опускаю то, как настраивать сервер, как настраивать репозиторий. В моем случае репозиторий был создан при помощи утилиты aptly. Более подробно по настройке и всем его функциям смотри документацию. Далее описаны конкретные шаги по добавлению пакета в репозиторий. все операции проводятся на сервере.

1. Добавляем пакет в репозиторий. Общий вид команды:

```
> aptly repo add <repo_name> <packet_name>
```

В моем случае это выглядит как:

```
> aptly repo add scontel-soft helloworld_1.0-1.deb
```

2. Далее необходимо обновить публичную часть репозитория командой.

```
> aptly publish update jessie
```

Установка пакета на устройство

1. Первое, что нужно сделать это добавить в список репозитория наш репозиторий. Для этого в файл с ресурсами

```
>sudo nano /etc/apt/sources.list
```

Добавляем текст

```
#Scontel Soft  
deb http://rplab.ru/~ozhegov/.aptly/public/ jessie main
```

2. Теперь необходимо обновить информацию о репозиториях:

```
> sudo apt-get update
```

3. После этого можно установить наш пакет

```
> sudo apt-get install helloworld
```

4. Далее нужный пакет можно обновлять (upgrade), удалять (remove) и так далее.