

Установка USBpix (Pixel Module Testing Software)

Подробное описание (с комментариями) процесса установки программы для тестирования пиксельных детекторов USBpix на систему CentOS 7 и Ubuntu 16.04LTS.

Скачать исходники можно из закрытого репозитория CERN'a:

<https://gitlab.cern.ch/jgrosse/USBpix>

Требования к окружению

- **ROOT** (Data Analysis Framework).
- **Qt 5.5.1** (или выше) необходим для компилирования исходников. С более старыми версиями некоторые компоненты USBpix работать не будут, а с более новыми - могут, но не со всеми версиями тестировались. Онлайн установщик с возможностью выбора версии можно скачать на официальном сайте: www.qt.io.

установка последней версии qt5 через терминал:

```
# для CentOS 7
yum install qt5-qtbase-devel

# для Ubuntu
sudo apt-get install qt5-default
```

- **libusb-1.0** - для доступа к USB-устройствам. Установка:

```
# для CentOS 7
yum install libusb
# как узнать установленную версию:
rpm -qa | grep -i libusb

# для Ubuntu
sudo apt-get install libusb-1.0-0-dev
# как узнать установленную версию:
dpkg -l libusb-1.0*
# или
apt-cache policy libusb-1.0*
```

- GPIB драйвера (linux-gpib).

Установка USBpix

Настройка переменных окружения

- Находим расположение установленного фреймворка ROOT (скорей всего "/usr/bin")
- Находим, куда установлен Qt5. Нам нужна папка вроде win32-msvc (например win32-msvc2015):

```
locate qt5 | grep msvc
```

- Заходим в корневую папку с исходниками USBpix, открываем файл setup.sh, прописываем туда (где-нибудь сверху после строк #!/bin/sh и прочих комментариев, начинающихся с #) две переменные среды:

```
export "ROOTSYS=<root_install_dir>"
export "QT5DIR=<qt5_msvc_dir>"

# где
# <root_install_dir> - путь до фреймворка ROOT
# <qt5_msvc_dir> - путь до msvc компонентов фреймворка Qt5
```

Пример:

```
export "ROOTSYS=/usr/bin"
export "QT5DIR=/usr/lib64/qt5/mkspecs/win32-msvc"
```

Запускаем настройку сборки

```
source setup.sh
```

Эту команду необходимо выполнять при каждом перезапуске ОС или терминала!

Указывать параметр `-gpib yes` не надо, т.к. при этом сборка будет пытаться использовать драйверы NI GPIB, а мы установили linux-gpib, это приведет к ошибкам на стадии

компилирования. Если параметр не указан, то используется значение по умолчанию "auto", при котором сборка проверит, установлены ли драйверы NI GPIB, не найдет их, затем проверит, установлены ли драйверы linux-gpib, найдет их и будет использовать.

Запускаем компиляцию исходников:

```
make -j 3
```

где 3 - это количество параллельных процессов компилирования (оптимально для двоядерного процессора с двумя потоками на ядро).

Компилирование не устанавливает никакие файлы в систему и не изменяет защищенные файлы, поэтому "sudo" не нужно.

Обновление программы

Чтобы пересобрать (обновить) программу, нужно сначала удалить скомпилированные файлы:

```
# запускать из корневой директории USBpix  
make clean  
make distclean
```

Затем внести изменения в файлы (например скачать новую версию исходников), выполнить скрипт setup.sh, и запустить компиляцию.

Запуск USBpix

После каждой перезагрузки системы нужно запускать скрипт setup.sh из папки с исходниками:

```
source setup.sh
```

Запускаемые файлы находятся в папке **bin**, библиотеки в папке **lib**.

После перезапуска ОС (напоминалка)

Для правильного функционирования программы USBpix, после каждой перезагрузки

системы перед первым запуском программы необходимо выполнить следующие команды:

```
# Если не настроена автозагрузка модуля (драйвера) контроллера GPIB:
# для контроллера NI PCIe-GPIB:
sudo modprobe tnt4882
# или
sudo modprobe ni_usb_gpib
# для контроллера NI GPIB-USB-HS

# Загрузка настроек GPIB драйвера
sudo gpib_config --minor 0
# или
sudo /usr/local/sbin/gpib_config --minor 0
# путь до исполняемого файла может отличаться

# Настройка переменных окружения для USBpix
source <USBpix_path>/setup.sh
# где <USBpix_path> - путь до корневой директории USBpix
```

Шпаков Константин, август 2018