

# ТЕХНОЛОГИЯ ETHERNET OVER USB КАК ОСНОВА ВЫЧИСЛИТЕЛЬНОГО КЛАСТЕРА НА RASPBERRY PI

Калоша С. О., Нестеренков С. Н.

Кафедра программного обеспечения информационных технологий, факультет компьютерных систем и сетей, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: swetofdark@gmail.com, nsn@bsuir.by

*Кластер из нескольких вычислительных устройств Raspberry PI является весьма популярным решением в последнее время. Зачастую его реализация подразумевает использование «стандартных» подходов с дополнительным сетевым оборудованием, беспроводными сетями и написанием низкоуровневых программных пакетов. В данной статье предлагается альтернативный подход по реализации небольших Master-Slave кластеров, основанный на использовании Ethernet over USB технологии и реализуемый простым конфигурированием операционной системы.*

## ВВЕДЕНИЕ

В рамках работы над магистерской диссертацией, целью которой является исследование моделей машинного обучения для классификации объектов реального мира, передо мной была поставлена задача проектирования устройства для сканирования твердых тел небольшого размера и массы. Данное устройство включило в себя несколько фотокамер. А полученным на них изображениям требовалась обработка. Для сохранения возможности использовать различные алгоритмы обработки изображений, а также высокой степени параллелизации таких вычислений было принято решение управлять каждой камерой и ее выводом отдельным вычислительным устройством. При этом, одно из таких вычислительных устройств также должно было координировать работу всех устройств между собой, предоставлять интерфейс взаимодействия с пользователем и обрабатывать данные с других датчиков (см. рис. 1).

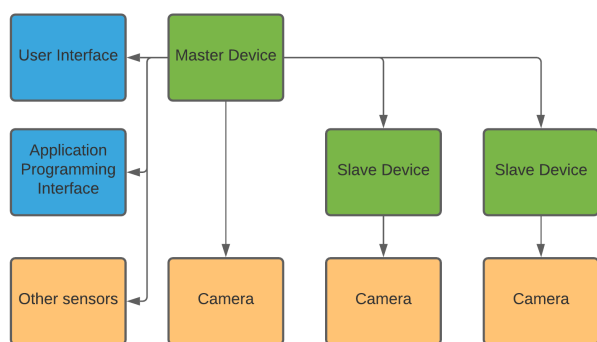


Рис. 1 – Схема устройства для сканирования объектов

Исходя из изложенных условий, было принято условие использовать микрокомпьютеры семейства Raspberry PI [1]. Эти микрокомпьютеры обладают достаточной вычислительной мощностью для обработки изображений, имеют возможность подключать камеры через отдельно выделенный интерфейс, а также позволяют подключать различные цифровые датчики. В каче-

стве координирующего устройства (Master) был выбран Raspberry PI 4, в качестве остальных устройств (Slave) – Raspberry PI Zero v1.2. После чего было необходимо решить каким образом настроить взаимодействие данных устройств между собой.

## I. ВАРИАНТЫ СОЕДИНЕНИЙ МЕЖДУ RASPBERRY PI

Существует несколько способов настроить обмен данными между несколькими Raspberry PI. Ниже приведены рассмотренные некоторые из таких способов:

- настройка последовательных соединений SPI, UART [2] через GPIO [3];
- проводное ethernet-соединение [4];
- беспроводное Wi-Fi соединение [4];
- Ethernet-соединение через USB.

Настройка GPIO портов для последовательных соединений требует низкоуровневого программирования, в следствие чего такие соединения будут более сложными в поддержке. Также, Raspberry обладают ограниченным набором GPIO-портов и не каждые из них могут быть использованы для SPI или UART соединений, более того, использование этих портов сильно уменьшает количество датчиков, подключаемых к микрокомпьютеру.

Соединение Ethernet-проводом для более чем двух выбранных устройств невозможно без дополнительного сетевого оборудования, в силу наличия только одного Ethernet-разъема, а также ограниченный в использовании Ethernet одновременно с модулем камеры на Raspberry Pi Zero.

Беспроводное Wi-Fi соединение позволяет объединить более двух устройств в одну сеть, однако это ограничивает возможности по подключению Master к домашним беспроводным сетям, а также делает возможным сетевое обнаружение всех вычислительных узлов конечного устройства, что нарушает принцип «черного ящика».

В силу вышеуказанных недостатков было принято решение настроить соединение, осно-

ванное на Ethernet over USB технологии. Такой подход позволит подключить к Raspberry PI 4 до четырех устройств, и настроить взаимодействие по принципу Master-Slave между ними по сетевым протоколам прикладного уровня модели ISO/OSI, таким как HTTP.

## II. НАСТРОЙКА ETHERNET OVER USB

Для настройки Ethernet over USB в первую очередь необходимо записать оригинальный образ операционной системы на SD-карту, используемую в устройстве. Образ исходной операционной системы может быть записан с помощью утилиты Raspberry PI Imager. Далее устройство должно быть переведено в режим OTG-гаджета включением `dwc2` оверлея дерева устройств. Для этого в конец файла `config.txt` необходимо добавить строку `dtoverlay=dwc2`.

Следующим шагом является установка параметров запуска ОС в файле `cmdline.txt`. Здесь необходимо указать загружаемые модули `modules-load=dwc2,g_ether` сразу после параметра `rootwait`, которые переопределяют USB-порты как Ethernet-порты. [5]

Для простой проверки, что устройство функционирует в режиме Ethernet over USB, нужно создать `ssh` файл в образе операционной системы, что включит `ssh` модуль ОС. И после ее установки достаточно подключить Raspberry PI к компьютеру через USB, и установить SSH соединение по сетевому имени по умолчанию `raspberrypi.local`.

## III. НАСТРОЙКА IP-АДРЕСАЦИИ

После конфигурации Ethernet over USB для всех устройств, каждое подключение к Master через USB Slave ведет к созданию сетевого интерфейса `usb*`, например `usb0`. Однако получить IP-адрес подключенного к этому интерфейсу узла невозможно без использования специальных модулей сетевого обнаружения. В случае, когда топология получившейся сети заранее известна и проста, данная проблема решается установкой статических IP-адресов для каждого подключаемого устройства.

Большинство существующих в данный момент руководств предлагают решать данную задачу редактированием файла `/etc/network/interfaces`. Однако такие изменения в Raspberry PI OS приведут к нарушению работы других сетевых интерфейсов, например `wlan`.

Поэтому нужно изменить `dhcpcd` настройки ОС в файлах `/etc/dhpcd.conf` Master и Slave устройств. Так, например для Master `usb0` интерфейс может быть настроен следующим образом:

```
interface usb0
static ip_address=192.168.1.1
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

Для Slave-устройства, присоединенного через сконфигурированный интерфейс, должны использоваться те же настройки, за исключением `ip_address`.

## IV. ПРАВИЛА СОЗДАНИЯ СЕТЕВЫХ ИНТЕРФЕЙСОВ

В Raspberry PI OS `usb*` сетевые интерфейсы не привязаны к конкретному физическому разъему, поэтому выполненная ранее настройка IP-адресов может давать сбои в случае, когда к Master одновременно подключены несколько Slave-устройств. Решить данную проблему можно созданием привязки имени сетевого интерфейса к USB-разъему.

В философии UNIX-систем всё, в том числе и сетевые интерфейсы, представлено файлом [6]. Файлы, представляющие устройства, хранятся в директории `/dev`, и управляются утилитой `UDEV`. Данная утилита использует набор правил, хранящихся в директории `/etc/udev/rules.d`. Добавление нового правила в эту директорию позволит переименовать устройство или изменить права доступа к нему. Поэтому добавление файла со следующим правилом позволит привязать интерфейс `usb0` к конкретному USB-разъему Raspberry PI 4 B:

```
SUBSYSTEM=='net', SUBSYSTEMS=='usb',
KERNELS=='1-1.3:*', NAME='usb0'
```

## ЗАКЛЮЧЕНИЕ

Изложенный выше подход по настройке сетевых соединений между несколькими Raspberry PI через USB позволил быстро получить систему из нескольких взаимодействующих вычислительных узлов без дополнительных программных и аппаратных затрат. Аспекты конфигурирования, описанные в данной статье могут оказаться полезными для настройки подобных соединений. Стоит отметить, что Ethernet over USB не является единственно верным вариантом, поэтому для каждой конкретной задачи стоит начать со сравнения альтернатив, как, например, было сделано в первом разделе данной статьи.

1. Raspberry Pi Documentation [Электронный ресурс] / Raspberry Pi Foundation. – Режим доступа: <https://www.raspberrypi.org/documentation/>. – Дата доступа: 10.10.2020.
2. Дэвид М. Харрис, Сара Л. Харрис. Цифровая схемотехника и архитектура компьютера. / Imagination Technologies. – М.: ДМК Пресс, 2018. – 792 с.
3. Монк, Саймон. Raspberry Pi. Сборник рецептов: решение программных и аппаратных задач, 2-е изд. / СПб.: ООО «Диалектика», 2019. – 528 с.
4. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. / СПб.: Питер, 2017. – 960 с.
5. The GNU/Linux USB driver [Электронный ресурс] / Linux-USB Org. – Режим доступа: <http://www.linux-usb.org/usbnet/>. – Дата доступа: 10.10.2020.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. / СПб.: Питер, 2019. – 1120 с.