

Impact Factor:

ISRA (India) = 4.971
ISI (Dubai, UAE) = 0.829
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
PIHII (Russia) = 0.126
ESJI (KZ) = 8.716
SJIF (Morocco) = 5.667

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

SOI: [1.1/TAS](#) DOI: [10.15863/TAS](#)

International Scientific Journal Theoretical & Applied Science

p-ISSN: 2308-4944 (print) e-ISSN: 2409-0085 (online)

Year: 2020 Issue: 04 Volume: 84

Published: 21.04.2020 <http://T-Science.org>

QR – Issue



QR – Article



Alexey Dmitrievich Shumsky

Yanka Kupala state university of Grodno
Student,
Grodno, Republic of Belarus

Mariya Igorevna Goreglyad

Yanka Kupala state university of Grodno
Student,
Grodno, Republic of Belarus

SOFTWARE MONITORING OF IP-CODER BASED ON RASPBERRY PI MICROCOMPUTER

Abstract: This article involves the IP-coder monitoring system based on the Raspberry Pi microcomputer. The possibilities of forming and receiving a response to certain messages of the Telegram bot to monitor the correct operation of the IP encoder are revealed.

Key words: IP-coder, microcomputer, Telegram-bot, Raspberry Pi, program code, Node.js.

Language: Russian

Citation: Shumsky, A. D., & Goreglyad, M. I. (2020). Software monitoring of IP-coder based on Raspberry Pi microcomputer. *ISJ Theoretical & Applied Science*, 04 (84), 301-306.

Soi: <http://s-o-i.org/1.1/TAS-04-84-52> **Doi:**  <https://dx.doi.org/10.15863/TAS.2020.04.84.52>

Scopus ASCC: 1709.

ПРОГРАММНЫЙ МОНИТОРИНГ IP-КОДЕРА НА ОСНОВЕ МИКРОКОМПЬЮТЕРА RASPBERRY PI

Аннотация: Данная статья рассматривается система мониторинга IP-кодера на основе микрокомпьютера Raspberry Pi. Выявляются возможности формирования и получения ответной реакции на определенные сообщения Telegram-бота для мониторинга корректной работы IP-кодера.

Ключевые слова: IP-кодер, микрокомпьютер, Telegram-бот, Raspberry Pi, программный код, Node.js.

Введение

В последние годы подавляющее большинство компаний и организаций имеют систему технической поддержки в виде различных программ-ботов. Такие программы-боты имеют огромное значение в выполнении однообразной и повторяемой работы с максимально возможной скоростью.

Именно чат-боты способны к решению самых разнообразных задач – от предоставления медицинской консультации до мониторинга специализированных прикладных решений, от распознавания эмоций до решения сложных консалтинговых задач в службах поддержки клиентоориентированных информационных

систем. Независимо от платформы, чат-бот – это прикладная программа, с помощью которой получая информацию от пользователя, получает корректные, логически обоснованные ответы.

Разновидностью чат-ботов являются Telegram-боты. Их суть заключается в возможности формирования и получения ответной реакции на определенные сообщения от пользователей. Эти возможности Telegram-бота, необходимые для мониторинга корректной работы IP-кодера на основе микрокомпьютера Raspberry Pi, являются предметом рассмотрения данной статьи.

Для реализации мониторинга был использован язык Node.js – программная

Impact Factor:

ISRA (India)	= 4.971	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 0.829	РИИЦ (Russia)	= 0.126	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 8.716	IBI (India)	= 4.260
JIF	= 1.500	SJIF (Morocco)	= 5.667	OAJI (USA)	= 0.350

платформа на движке V8, который транслирует JavaScript-код в машинный. В связки вместе с Telegram API Bot.

Node.js часто используется как back-end язык программирования, выполняя роль сервера, но имеется возможность разработки оконных десктопных приложений при помощи специальных платформа, например, NW.js, AppJS или Electron. Из спецификации можно отметить возможность использовать язык Node.js для разработки приложений в сфере «интернета вещей» (Internet of Things, IoT). В последнее время мы все чаще встречаемся с смарт-браслетами, колонками и любой другой «умной» техникой, которая в теории может быть запрограммирована на Node.js. История Node.js началась с разработки

нового JavaScript-движка V8 в датском отделении компании Google. Ведущий разработчик Lars Bak выделил основные проблемы, которые должны были решены в движке, это производительность и масштабируемость. Первая лабораторная версия была готова 3 июля 2008 года, а позже был представлен первый браузер Chromium в публичный релиз которого входил новый движок.

Telegram API Bot – это программный интерфейс, позволяющий программировать собственного бота. API включает в себя объекты и команды, предназначенные для установки поведения бота Telegram. Используя интерфейс, вы можете создавать собственные программные коды, которые при запуске в Telegram начинают работать как боты.

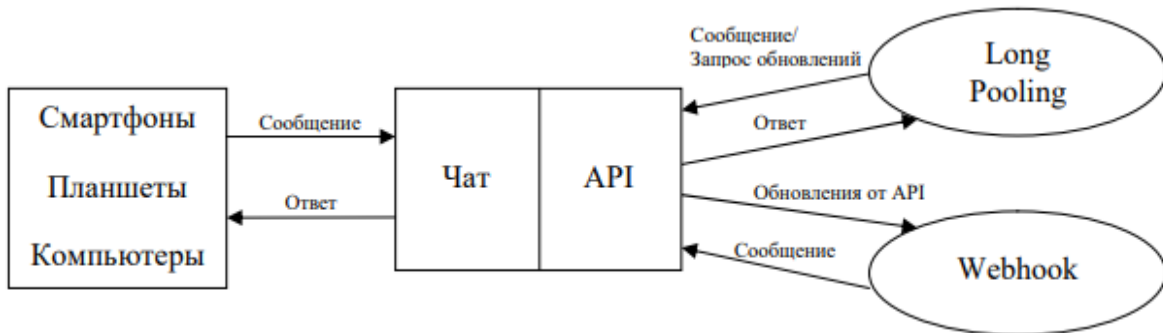


Рисунок 1 – Принцип работы чат-бота на платформе Telegram

Как видно на приведенном ниже рисунке 2, регистрация бота в Telegram происходит через пользователя BotFather, который также является ботом, позволяющим осуществлять учет и настройку пользовательских ботов. BotFather позволяет задать имя нашему боту, сделать его описание, изображение и список команд, доступных пользователям. Нами создаются и

реализуются 2 команды, которые требуют ответной реакции. При наборе команды /start мы видим возможности BotFather. С помощью этих команд можно создавать, управлять своим ботом.

При создании бота выдается специальный ключ, необходимый для связи пользователей с ботом и организации их взаимодействия. Этот ключ для каждого бота уникальный.

Impact Factor:

ISRA (India)	= 4.971	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 0.829	ПИИИ (Russia)	= 0.126	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 8.716	IBI (India)	= 4.260
JIF	= 1.500	SJIF (Morocco)	= 5.667	OAJI (USA)	= 0.350

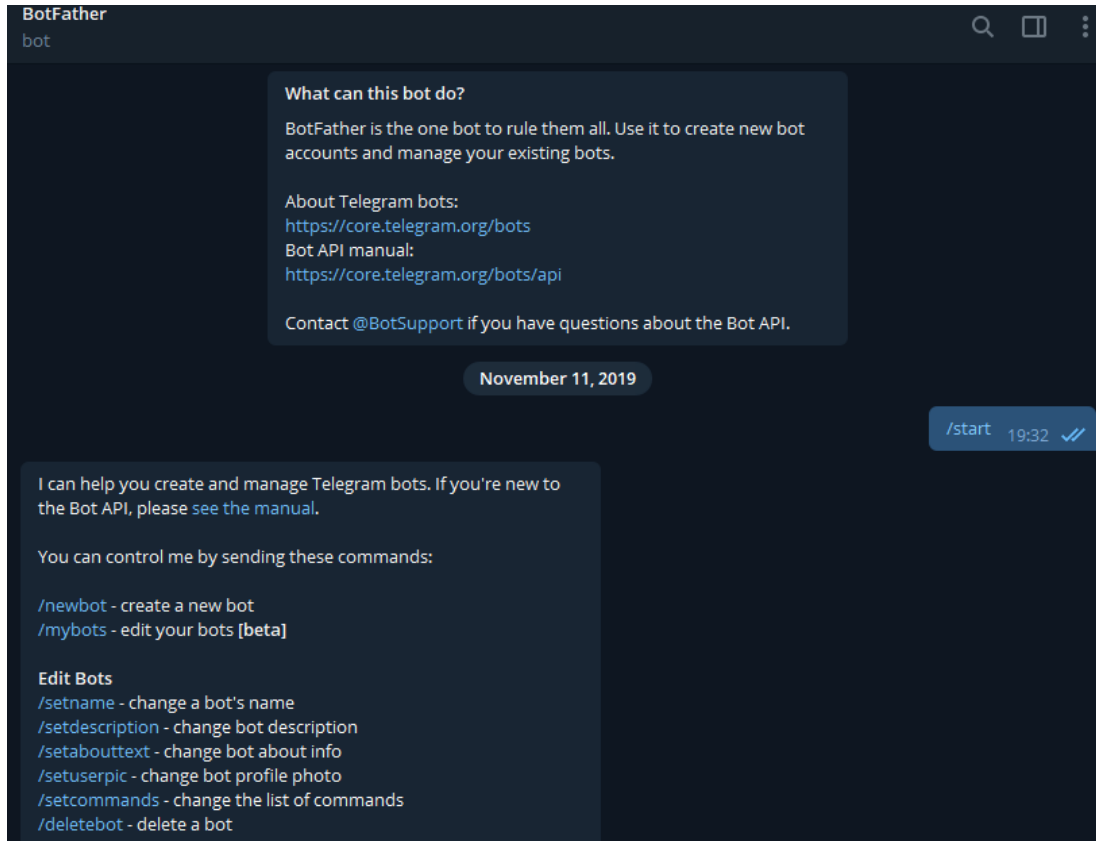


Рисунок 2 – Чат с пользователем BotFather в Telegram

На приведенном ниже рисунке. 3 можно увидеть, что, отправляя команду /newbot, нужно в ответ ввести имя бота и его никнейма, по которому он будет идентифицироваться. В дальнейшем все настройки, кроме никнейма,

можно будет поменять. При необходимости также можно получить новый уникальный ключ. По аналогии создается второй такой же бот, только с другим именем.

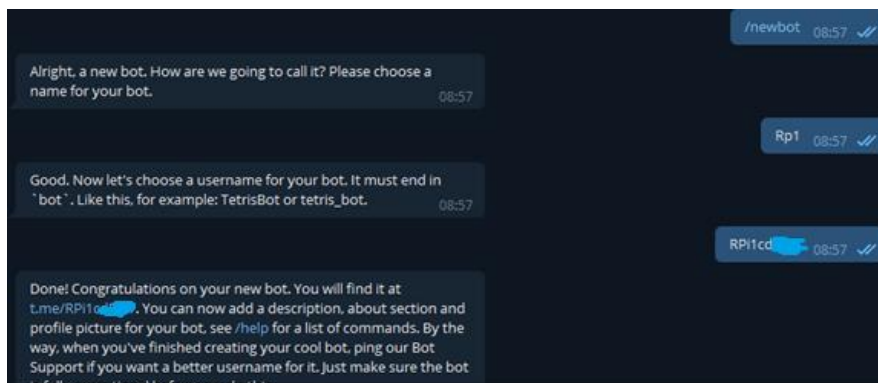


Рисунок 3. – Чат с пользователем BotFather для регистрации бота

В итоге, как видно на приведенном ниже рисунке 4, при вводе команды /mybots появляется 2 бота. При выборе каждого из них появляются

дополнительные настройки в виде описания, установки аватарки и настроек приватности.

Impact Factor:

ISRA (India)	= 4.971	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 0.829	ПИИЦ (Russia)	= 0.126	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 8.716	IBI (India)	= 4.260
JIF	= 1.500	SJIF (Morocco)	= 5.667	OAJI (USA)	= 0.350

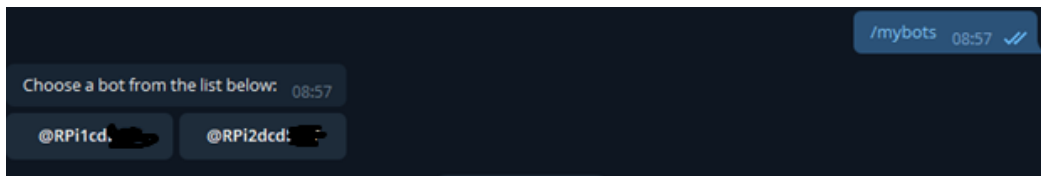


Рисунок 4 – чат в Telegram с выбором ботов

Следующим шагом, после создания telegram-бота, требуется установить библиотеки, необходимые для связи программного кода и API-функций telegram'a. Для работы устанавливается следующий ряд библиотек:

- types – это скриптовый язык, компилируемый в JavaScript;
- debug – модуль логгера для отладки и детектирования ошибок;
- esm – загрузчик модулей JavaScript'ов;
- ms – библиотека для преобразования строки (1y, 2.5hrs и т.д.) в миллисекунды;
- node-fetch – библиотека для работы с запросами и ответами HTTP;
- node-os-utils – модуль, собирающий статистику из ОС (диск, трафик, процессор);
- sandwich-stream – библиотека для прослушивания разных протоколов, HTTP, TCP;
- spawn-sync – библиотека для чтения запущенных процессов (IceCast, liquidsoap);
- telegraf – библиотека API-функций для связи с telegram'ом;
- telegram-typings – модуль для отправки данных в telegram.

После установки библиотек необходимо перейти к написанию кода, то есть обучению нашего бота каким-то командам, или изменениям рабочей системы в зависимости от корректных условий и получения на это моментальной реакции.

С этой целью используется код программирования Node.js. В Node.js существует возможность взаимодействовать с устройствами ввода-вывода через свой API, написанный на C++, использовать сторонние библиотеки, созданные при помощи разных языков программирования, вызывая их из JavaScript-кода.

На рисунке 5 изображена функциональная блок-схема для мониторинга IP-кодера на основе микрокомпьютера Raspberry. В начале запуска скрипта требуется настроить telegram-чат для работы ботов. После настройки необходимо ввести данные каждого бота, на каждом из компьютеров RPi. Также необходимо ввести token, который выдавался при регистрации и chatId. В этот token будут отправляться данные, а также с его помощью будет производиться связь между системой и Telegram.

Impact Factor:

ISRA (India) = 4.971	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИЦ (Russia) = 0.126	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 8.716	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 5.667	OAJI (USA) = 0.350

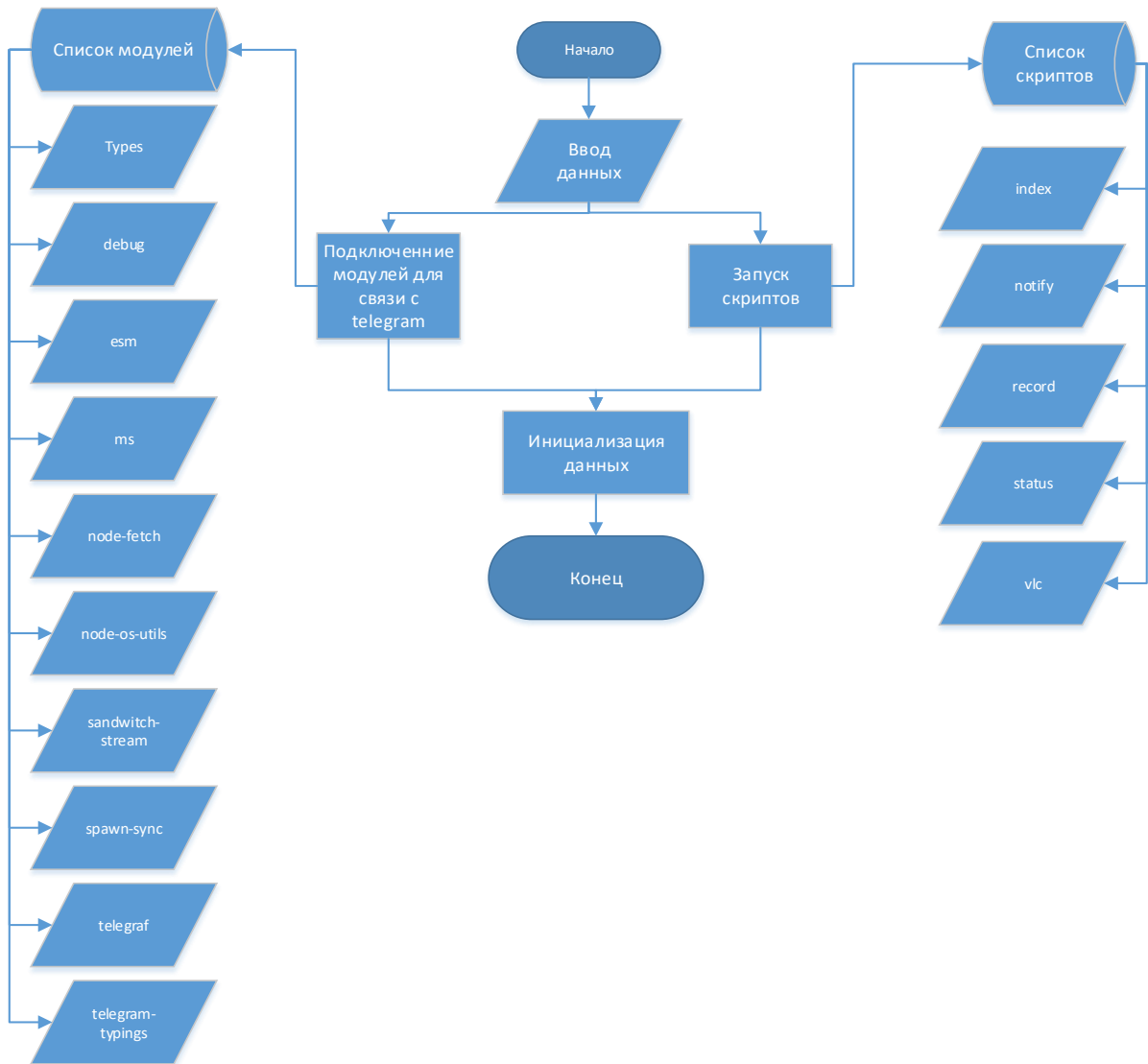


Рисунок 5 – Функциональная блок-схема устройства для мониторинга IP-кодера на основе микрокомпьютера Raspberry Pi.

Для обмена информацией необходимо подключить модули, с помощью которых можно получать уведомление о состоянии системы, захвате звука и контроле работы устройства.

После запуска модулей необходимо написать код, с помощью которого будут выполняться задачи, поставленные в нем. Программный блок будет состоять из 5 программных ячеек: *index*, *notify*, *record*, *status*, *vlc*.

Программный блок с названием *Index* собирает обобщенную информацию и является связующим звеном между блоками *notify*, *record*, *status*, *vlc* и *telegram*'ом.

Следующим блоком является *notify*. В данном блоке происходит контроль и отправка информации в случае отклонения от идеала. То есть, если происходит остановка программного сервиса, с помощью которого идет вещание в Icecast, *vlc*, *liquidsoap*, в Relegram придет

уведомление, об обнаружении проблемы на каком-то из блоков. Такое уведомление будет присылаться до тех пор, пока не будет остановлено командой в чат */stopalert*. Если ошибка будет исправлена, то в чат будет отправлена команда от бота - *подключение по telnet восстановлено*.

Блок *record*, используется для контроля и мониторинга отправляемого и получаемого сигнала. Принцип работы заключается в «захвате» транслируемого сигнала, передаваемого через звуковые карты. Запись сигнала происходит в течение 5 секунд и отправляется в telegram в виде голосового сообщения. С помощью данный команды можно отслеживать, на каком этапе происходит проблема с передачей звука и отслеживать его наличие.

Блок *status* представляет собой сбор обобщенной информации о работе RPi. При

Impact Factor:

ISRA (India)	= 4.971	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 0.829	ПИИЦ (Russia)	= 0.126	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 8.716	IBI (India)	= 4.260
JIF	= 1.500	SJIF (Morocco)	= 5.667	OAJI (USA)	= 0.350

введении команды /status в чат бот отправляет команду о текущем состоянии системы, температуре процессора, продолжительности работы хоста, отправке пакетов интернет-трафика.

Последний блок представляет собой скрипт vlc, относящийся только к RPi2. Так как вещание происходит через проигрыватель VLC, требуется его контроль на наличие проигрывания контента. Для этого создается код, который отслеживает по telnet протоколу ошибки и отправляет в Telegram через блок notify. В случае обнаружения проблем в блоках, в чате Telegram появится соответствующее уведомление.

Если все блоки функционируют корректно, то функциональная блок-схема программного обеспечения для мониторинга IP-кодера на основе микрокомпьютера Raspberry Pi может считаться законченным вариантом работы скрипта.

На основании вышеизложенного можно сделать следующие выводы:

– мониторинг корректной работы IP-кодера на основе микрокомпьютера Raspberry Pi осуществляется с помощью Telegram-бота через введение пользователя BotFather; команды /mybots, выбора дополнительных настроек описания, установки аватарки и настроек приватности, установки библиотек, необходимых для связи программного кода и API-функций telegram'a;

– для запуска скрипта требуется настроить telegram-чат, ввести на каждом из компьютеров RPi данные бота, token и chatId; затем создать блок скриптов, состоящий из программных ячеек, обеспечивающих сбор, контроль, отправку и мониторинг отправляемого и получаемого сигнала, текущего состояния системы, температуры процессора, продолжительности работы хоста, пакетов интернет-трафика, отслеживание ошибок и уведомление о них; корректное функционирование всех блоков свидетельствует о завершении работы скрипта.

Научное исследование проведено под руководством Гачко Геннадия Алексеевича, проректора по учебной работе, Гродненского Государственного университета имени Янки Купалы.

References:

1. Petin, V.A. (2015). *Raspberry Pi Microcomputers: A Practical Guide*. (p.240). SPb.: BVH-Petersburg.
2. (2016). *Telegram Bot API*. *Telegram Documents*. (Date of access: 18.04.20). Retrieved from: <https://core.telegram.org/bots/api>
3. Avanesyan, N. L. (2017). *Telegram, as an example of a messenger: opportunities and development prospects*. (Date of access: 18.04.2020). Retrieved from: https://elibrary.ru/download/elibrary_29653726_34734017.pdf
4. Arkhakov, D. (2016). *NodeJS: Making buttons in the Telegram API (inlinekeyboards)*. (Date of access: 18.04.2020). Retrieved from <https://archakov.im/post/nodejs-make-buttons-on-telegramapi.html>
5. (2010). *Official site Node.JS* (Date of access: 18.04.2020). Retrieved from: <https://www.nodejs.org/>
6. Harron, D. (2014). *Node.js. Web server application development on Javascript*. (p. 144). Moscow: DMK Press.
7. Messina, C. (2016). *will be the year of conversational commerce*. (Date of access: 18.04.20). Retrieved from: <https://medium.com/chris-messina/2016-will-be-the-year-of-conversationalcommerce-1586e85e3991>
8. (n.d.). *Bot Code Examples – Telegram APIs*. (Date of access: 18.04.20) Retrieved from: <https://core.Telegram.org/bots/>
9. Warren, G. (2018). *Advanced Raspberry Pi*. (p.521). Ontario, Canada, apress.
10. Upton, U., & Halfacree, G. (2012). *Raspberry Pi User Guide*. (p.250). United Kingdom, A John Wiley and Sons.