

Ярошенко А.И.

*Научный руководитель: к.т.н., доцент А. А. Колпаков, аспирант С.В. Савинов
Муромский институт (филиал) федерального государственного образовательного
учреждения высшего образования «Владимирский государственный университет имени
Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
e-mail: artawower@protonmail.com*

Автоматизированный органайзер на Raspberry PI3

Raspberry PI3 (рис. 1) представляет собой бюджетный одноплатный компьютер небольших размеров, оснащенный процессором BCM2837, 1 Гб оперативной памяти и обладающий такими интерфейсными разъемами как: 3.5 мм Jack, HDMI, Ethernet, Wi-fi, USB 2.0 x4, GPIO, Micro USB, Bluetooth 4.1 (рис. 2). Данный микрокомпьютер был выбран в качестве основного устройства благодаря своей относительно невысокой стоимости, низким энергопотреблением, поддержкой по умолчанию необходимых для разработки языков высокого уровня, UNIX подобную операционную систему – Raspbian.



Рис. 1. Raspberry PI3

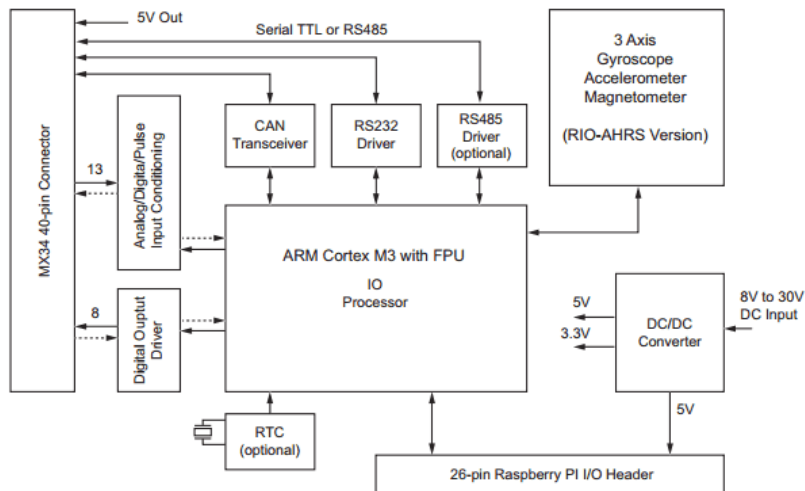


Рис. 2. Схема Raspberry PI3

Разработка программного обеспечения производилась через командную строку с помощью среды разработки VIM. В качестве языка программирования был использован Python3.6 и Golang. Для запуска задач с периодичным выполнением используется встроенная в операционную систему служба Cron, в качестве службы ведения логов используется пакет Mutt, все уведомления о внутренних ошибках поступают через программный интерфейс на почту. Для отправки email писем используется Gmail API.

Актуальность данной работы заключается в автоматизации управления рабочего процесса и личного времени, а также в решении фоновых задач с длительным периодом времени. Важным преимуществом перед другими аналогами является круглосуточная работа Raspberry PI. В отличие от работы ПК это возможно благодаря крайне низкому энергопотреблению.

Для разработки умного будильника, определяющего выходные и праздничные дни используется публичное API сервиса isdayoff.ru, модуль requests на языке Python3.6. Скрипт запускается каждый день в определенное пользователем время, перед запуском будильника идет сбор информации о том выходной день или нет. В качестве отказоустойчивости присутствует скрипт собирающий данные заранее, в случае отсутствия доступа в интернет сигнал будильника срабатывает из кэшируемых данных. При наличии подключения к сети интернет в качестве звукового сигнала для будильника выступает случайное звуковое сопровождение загруженное через Spotify API, при его отсутствии случайный звуковой трек, удовлетворяющий условие регулярного выражения берется из файловой системы. При срабатывании сигнала подается напряжение на 16 pin GPIO, приводящий к срабатыванию реле и включающий лампу дневного света. В необязательной конфигурации скрипта указывается количество повторений сигналов будильника, а также интервал через который он будет срабатывать.

Организованная рассылка рабочей почты достигается с помощью планировщика задач Cron, почтового клиента Mutt, программного обеспечения написанного на Bash, конвертора html файлов в PDF – Wkhtmltopdf. Данные для формирования отчетов берутся с приватного репозитория BitBucket, после чего формируется html файл, текстовые данные берутся из commit сообщений на удаленном сервере, графические изображения создаются с помощью Python библиотеки Matplotlib. Отчеты отправляются с указанной в скрипте периодичностью.

Для автоматизации сбора информации с помощью парсинга данных с публичных сайтов используется язык высокого уровня Golang. Для изменения ip адреса используется скрипт написанный на данном языке, переключение осуществляется запущенным в фоне процессом через openvpn, данные о доступных VPN серверах собираются при каждом 10ом запуске, что позволяет избежать простоев. Данные собираются с помощью библиотеки goquery, позволяющему искать по вложенным тегам внутри DOM дерева загруженной страницы.

Литература

1. Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
2. Хахаев И.А. Практикум по алгоритмизации и программированию на Python. – М.: Альт Линукс, 2010. — 126 с.
3. Марк Саммерфильд. Программирование на языке GO. МСК.: ДМК Пресс, 2016. – 556 с