Министерство образования и науки РФ  
  
Московский авиационный институт

(национальный исследовательский университет)

Факультет №3 «Системы управления, информатика и электроэнергетика»

Кафедра 304 «Вычислительные машины, системы и сети»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ

по дисциплине: «Операционные системы»

на тему: «Изучение ОС TI RTOS в составе программно-аппаратного комплекса на базе микроконтроллера MSP432»

Выполнил:

студент группы 3о-311Б

Волков А.К.

Проверил:

Ассистент каф. 304

Аляутдин Р.Т.

Задание

Необходимо изучить принципы работы с операционной системой (ОС) реального времени (RTOS) от компании Texas Instruments (TI). Для этого требуется разработать устройство, осуществляющее замеры некоторой физической величины. Основным элементом устройства является микроконтроллер MSP432P401R под управлением TI RTOS. Так же устройство должно взаимодействовать со смартфоном посредством Bluetooth Low Energy.

Содержание

[Задание 2](#_Toc483356558)

[Содержание 3](#_Toc483356559)

[1 Введение 5](#_Toc483356560)

[2 Теоретическая часть 6](#_Toc483356561)

[2.1 Используемые технологии 6](#_Toc483356562)

[2.2 MSP432P401R 6](#_Toc483356563)

[2.3 TI RTOS 7](#_Toc483356564)

[2.3.1 Работа с SD картой 7](#_Toc483356565)

[2.3.2 Управление производительностью 8](#_Toc483356566)

[2.3.3 Работа с BLE 9](#_Toc483356567)

[2.4 Bluetooth Low Energy 9](#_Toc483356568)

[2.4.1 CC2650 MODA 10](#_Toc483356569)

[3 Практическая часть 11](#_Toc483356570)

[3.1 Устройство ПАК 11](#_Toc483356571)

[3.2 Разработка программного обеспечения микроконтроллера 12](#_Toc483356572)

[3.2.1 Среда разработки 12](#_Toc483356573)

[3.2.2 Реализация программного обеспечения 12](#_Toc483356574)

[3.2.2.1 Принцип работы 13](#_Toc483356575)

[3.2.2.2 Data\_service 14](#_Toc483356576)

[3.2.2.3 Manager\_service 15](#_Toc483356577)

[3.2.2.4 Stream\_service 15](#_Toc483356578)

[3.3 Приложение для смартфона 16](#_Toc483356579)

[3.3.1 Среда разработки 16](#_Toc483356580)

[3.3.2 Разработка приложения 17](#_Toc483356581)

[3.3.3 Возможности приложения 19](#_Toc483356582)

[3.4 Замеры энергопотребления 20](#_Toc483356583)

[3.4.1 Оптимизация 22](#_Toc483356584)

[4 Заключение 23](#_Toc483356585)

[Список использованной литературы 24](#_Toc483356586)

[Приложение 25](#_Toc483356587)

# Введение

Благодаря распространению стандартов беспроводной связи, унифицированных средств разработки и методов сопряжения становится возможным создавать устройства, способные взаимодействовать, несмотря на то, что они могли быть разработаны разными производителями.

В данной курсовой работе был разработан программно-аппаратный комплекс (ПАК), в состав которого входит микроконтроллер – MSP432P401R, накопитель информации – SD карта, измеритель – ультразвуковой датчик расстояния HC-RS04, и модуль беспроводной связи – CC2650-MODA, с помощью которого пользователь со смартфоном и установленным приложением сможет управлять комплексом.

# Теоретическая часть

## Используемые технологии

При реализации ПАК был использован набор различных технологий.

Программа для микроконтроллера (МК) была написана на языке программирования С в среде разработки Code Composer Studio 7.1.0. Запись программы на МК производилась с помощью отладочной платы XDS110-ET, которая входит в состав MSP‑EXP432P401R (SimpleLink™ Microcontroller LaunchPad™ Development Kit).

Прошивка для модуля Bluetooth Low Energy (BLE) CC2650-MODA была взята у компании-производителя TI. Запись прошивки производилась с помощью все той же отладочной платы XDS110-ET, но через отдельный вывод JTAG с помощью программы Flash Programmer 2.

Приложение для смартфона под управлением Android 4.3 было написано с помощью библиотек Qt 5.8.0.

## MSP432P401R

MSP432P401R – микроконтроллер от TI.

Характеристики MSP432P401R:

* Процессор ARM Cortex-M4F разрядностью 32 бита;
* Тактовая частота 48 МГц;
* Flash-память объемом 256 КБ;
* Оперативная память объемом 64 КБ;
* АЦП разрядностью 14 бит;
* 40 внешних портов ввода/вывода;
* Поддержка интерфейсов SPI, I2C, UART, JTAG.

## TI RTOS

Программа на МК использует для своей работы TI RTOS. Это позволяет использовать уже реализованные возможности, такие как удобное управление энергопотреблением, работа с таймерами, работа с периферийными устройствами, в том числе SD картой, а так же работа с BLE модулем. Благодаря тому, что основа уже протестирована и интегрирована, TI-RTOS позволяет сфокусироваться на разработке конечного приложения. Тем не менее, при необходимости доступ к аппаратной части остается.

### Работа с SD картой

Для работы с ней в составе TI RTOS есть драйвер SDSPI. Он позволяет взаимодействовать с картой через интерфейс SPI. Файловой системой SD карты должен быть FatFs.

Драйвер SDSPI предоставляет высокоуровневый интерфейс для работы с картой памяти. Ниже представлен необходимый набор функций.

* void **SDSPI\_init**();

Инициализация драйвера. Перед вызовом этой функции структура SDSPI\_config, в которой хранится информация о конфигурации драйвера, должна быть проинициализирована.

* void **SDSPI\_Params\_init**(SDSPI\_Params \*params);

Инициализирует структуру SDSPI\_Params для использования в следующей функции.

* SDSPI\_Handle **SDSPI\_open**(uint\_least8\_t index, uint\_least8\_t drv, SDSPI\_Params \*params);

Открывает для записи устройства с номером, указанным в drv, параметрами params. Фактически монтирует файловую систему.

* void **SDSPI\_close**(SDSPI\_Handle handle);

Размонтирование файловой системы.

Непосредственно запись производится с помощью стандартных функций библиотеки «stdio.h». Основное отличие в том, что имени файла должен предшествовать префикс «fat:0:», указывающий на нужную ФС и номер устройства.

### Управление производительностью

Для управления производительностью МК в TI RTOS существует библиотека «power.h». Возможно изменение таковых частот МК и перевод МК в различные режимы сна. Для этого могут быть использоваться следующие функции:

* int\_fast16\_t **Power\_init**(void);
* void **Power\_disablePolicy**(void);
* void **Power\_enablePolicy**(void);
* void **Power\_setPolicy**(Power\_PolicyFxn policy);
* void **Power\_idleFunc**(void);
* int\_fast16\_t **Power\_setPerformanceLevel**(uint\_fast16\_t level);

В библиотеке предусмотрены 4 режима производительности:

Level MCLK (MHz) HSMCLK (MHz) SMCLK (MHz) ACLK (Hz)

----- ---------- ------------ ----------- ---------

0 12 3 3 32768

1 24 6 6 32768

2 48 24 12 32768

3 48 48 24 32768

MCLK отвечает за основную тактовую частоту МК.

### Работа с BLE

Для работы с BLE в TI RTOS используется библиотеки «sap.h» и «snp.h», которые входят в состав BLE SDK от TI. Simple Application Processor (SAP) – это набор API, используемых в качестве функций доступа к CC2650 Simple Network Processor. SAP передает по проводам команды, события и вызовы функций от модуля BLE к МК и обратно. Для управления соединением используются различные структуры и функции, название которых начинается с SAP и SNP.

Основные задачи:

* Инициализация GATT (добавление сервисов, характеристик, дескрипторов, и т.д.).
* Инициализация GAP (данные афиширования, параметры соединения, и т.д.).
* Афиширование и ожидание подключения.
* Ответ на GATT-запросы и отправка оповещений, если необходимо.

## Bluetooth Low Energy

Bluetooth Low Energy – энергоэффективная беспроводная передача данных.

Определения и термины:

GATT - Generic Attribute Profile (профиль общих атрибутов) – используется для получения доступа к сервисам и их характеристикам.

GAP - Generic Access Profile (профиль общего доступа) – используется для обнаружения устройства и установки соединения.

Клиент – устройство, которое инициирует запросы и принимает оповещения.

Сервер – устройство, которое принимает запросы и рассылает оповещения.

Характеристика – некоторое значение, которое используется для взаимодействия клиента и сервера.

Сервис – набор характеристик, схожих между собой по выполняемой функции.

Дескриптор – дополнительная информация о характеристике. Характеристика может иметь как ноль, так и несколько дескрипторов. [1]

### CC2650 MODA

CC2650MODA SimpleLink™ Bluetooth® low energy Wireless MCU Module - модуль, предназначенный для работы отдельно или в сопряжении с MSP432P401R. Реализован в виде небольшой печатной платы с портами ввода/вывода, совпадающими с портами MSP432P401R и интерфейсом JTAG.

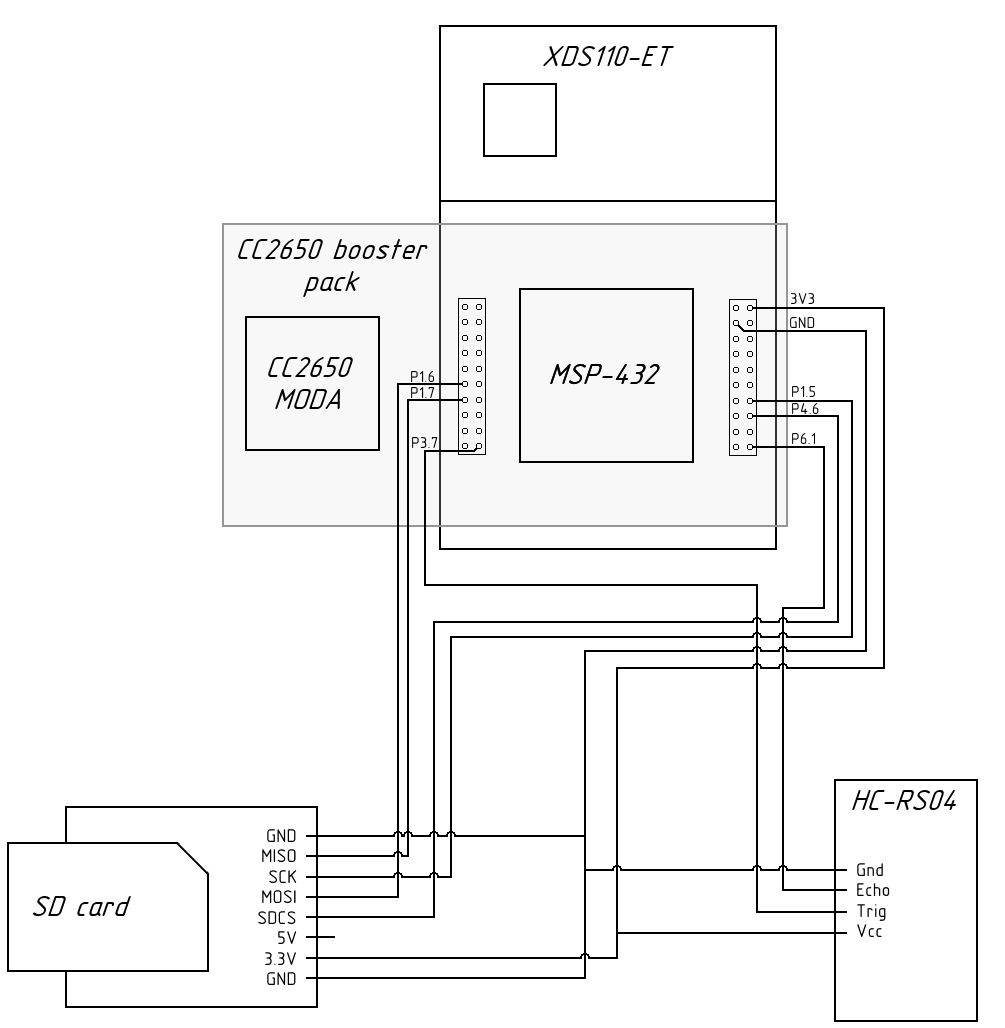
Для беспроводной передачи данных с низким энергопотреблением реализованы интерфейсы Bluetooth Low Energy, ZigBee и 6LoWPAN.

В основе модуля лежит МК с процессором ARM Cortex-M3, работающем на частоте до 48 МГц, 128 КБ flash-памяти и 8 КБ SRAM, а так же 20 КБ SRAM с низким током утечки. [2]

# Практическая часть

## Устройство ПАК

Функционально ПАК состоит из 4 узлов. Это плата с микроконтроллером, датчик расстояния HC-RS04, модуль с SD картой и модуль BLE CC2650 booster pack.



*Рис. 1. Функциональная схема ПАК*

Модуль CC2650 надевается сверху на ножки MSP‑EXP432P401R. Остальная периферия подключается снизу к тем же разъемам. При этом периферия и СС2650 используют разные ножки для передачи своих сигналов. Таким образом, устройства не конфликтуют между собой ни на физическом уровне, ни на программном.

## Разработка программного обеспечения микроконтроллера

### Среда разработки

В качестве среды разработки была выбрана IDE Code Composer Studio 7.1.0. В ней поддерживается разработка проектов для MSP432, есть встроенные компилятор, отладчик, программа для прошивки МК и программа для измерения энергопотребления “EnergyTrace”.

Для разработки под MSP432 был установлен SDK simplelink\_msp432\_sdk\_1\_30\_00\_40, а для работы с BLE – simplelink\_msp432\_sdk\_bluetooth\_plugin\_1\_10\_00\_42 и ble\_sdk\_2\_02\_01\_18\_setup.

### Реализация программного обеспечения

Было принято решение не реализовывать весь проект с нуля, а взять за основу пример работы с BLE от TI под названием “Project Zero”, который находится в Resource Explorer > Software > SimpleLink SDK Plugins > Connectivity > SimpleLink SDK Bluetooth Plugin > Examples > Development Tools > MSP-EXP432P401R – Rev 2.x (Red) > Bluetooth > project\_zero.

В нем уже реализованы необходимые классы-профили profile\_util и simple\_gatt\_profile, создан поток по управлению и обработке состояний BLE модуля. Так же есть классы data\_service, button\_service и led\_service, отвечающие за передачу данных, оповещения нажатий на кнопки и управление светодиодом. После ознакомления с тем, как реализовываются сервисы, были созданы сервисы data\_service (измененный из примера), manager\_service и stream\_service. Их назначение разобрано далее.

Для компиляции данного проекта необходимо дополнительно настроить пути компиляции и линковки. Для функции usleep используется unistd.h. Для его подключения в настройках проекта (Properties > Build > MSP432 Compiler > Include options) нужно добавить

"${COM\_TI\_SIMPLELINK\_MSP432\_SDK\_INSTALL\_DIR}/kernel/tirtos/packages/ti/sysbios/posix".

Для использования функций по работе с SD картой необходимо подключить в те же настройки компилятора

"${COM\_TI\_SIMPLELINK\_MSP432\_SDK\_INSTALL\_DIR}/source/third\_party/CMSIS/Include"

И в настройки линковщика (Properties > Build > MSP432 Linker > File Search Path)

"${COM\_TI\_SIMPLELINK\_MSP432\_SDK\_INSTALL\_DIR}/source/third\_party/fatfs/lib/fatfs.aem4f"

#### Принцип работы

При запуске микроконтроллера инициализируются драйвера, запускается SYS/BIOS и вызывается функция MSPBLE\_createTask(). В ней создается два потока с помощью функций:

Task\_construct(&MSPBLETask, MSPBLE\_taskFxn, &taskParams, NULL);

Task\_construct(&SensorTask, SensorLoop, &taskParams, NULL);

Первый поток отвечает за инициализацию BLE-стека и работу с модулем CC2650. Он взят неизменным из примера Project\_Zero.

Второй поток реализует основную логику приложения (рис. 2).



*Рис. 2. Основной цикл логики*

#### Data\_service

Этот сервис служит для получения записей за минуту из SD накопителя. Он содержит две характеристики. Первая называется DATA\_STRING\_CHAR0, длина – 20 байт. При записи в нее числа вызывается функция MSPBLE\_processDataServiceCB, в которой устанавливается флаг необходимости передачи минуты, равной записанному числу. Вторая характеристика – DATA\_STREAM\_CHAR1, длина – 8 байт. Она используется для передачи одной секунды из выбранной минуты с помощью оповещений, которые принимает приложение на смартфоне.

Для того чтобы данные не терялись, организована их синхронизация. Первая запись данных в DATA\_STRING\_CHAR0 задает считываемую минуту, а последующие 59 задают номер считываемой секунды. Таким образом, исключается отправка данных тогда, когда принимающая сторона не сможет их принять.

#### Manager\_service

Данный сервис позволяет управлять процессом записи. Единственная характеристика - MAN\_CHAR0, длиной 5 байт. Чтение вернет количество записанных минут, что важно знать для указания получаемой минуты. Запись дает эффект, только если первый символ равен “r” – в таком случае обнуляется счетчик записанных минут и отсчет ведется с начала.

#### Stream\_service

Задача данного сервиса – транслировать текущее значение датчика. Для этого используется характеристика STREAM\_CHAR0 длиной 8 байт. Ее нельзя изменить извне, можно только подписаться на ее обновления, тем самым организовав ежесекундную трансляцию расстояния.

## Приложение для смартфона

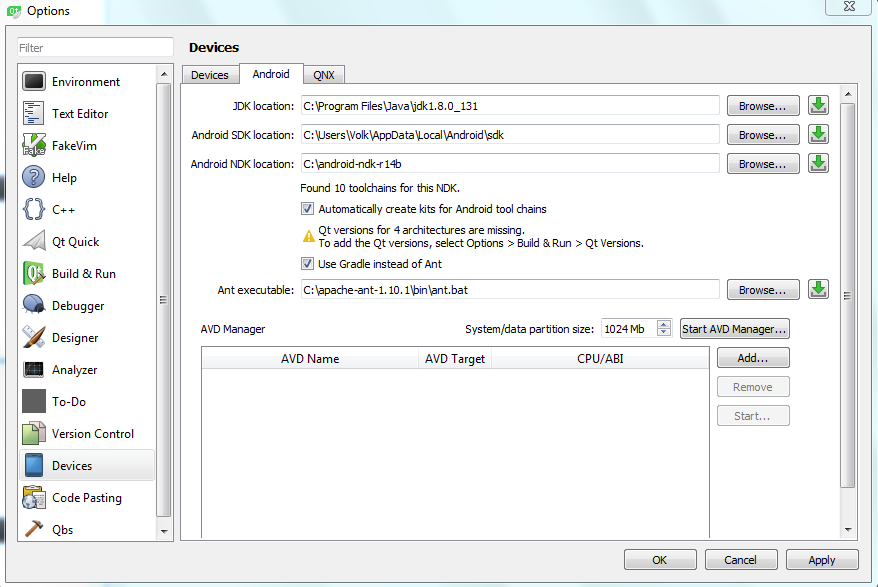
Для реализации приложения на смартфоне был выбран кроссплатформенный фреймворк Qt 5.8.0. Он позволяет быстро разрабатывать GUI приложения, которые будут одинаково работать на разных платформах. Так же благодаря готовым библиотекам работа с Bluetooth Low Energy не вызывает затруднений.

### Среда разработки

Так как основной язык для написания Qt-программ – С++, то он же использовался для создания приложения на смартфоне. Так как С++ - не нативный язык для Android, то Qt для компиляции использует Android SDK, JDK, Android NDK.

Для создания приложения была установлена версия Qt 5.8.0 for Android (Windows 32-bit). После установки в настройках Android в QtCreator (Options > Devices > Android) были указаны пути к jdk1.8.0\_131, Android SDK и JDK, как показано на рисунке 3.

В качестве средства автоматической сборки между Ant и Gradle был выбран Gradle. Для его корректной работы в папку C:\Users\%USERNAME%\AppData\Local\Android\sdk\tools скопировать папку templates из C:\Program Files\Android\Android Studio\plugins\android\lib.



*Рисунок 3. Настройки QtCreator*

### Разработка приложения

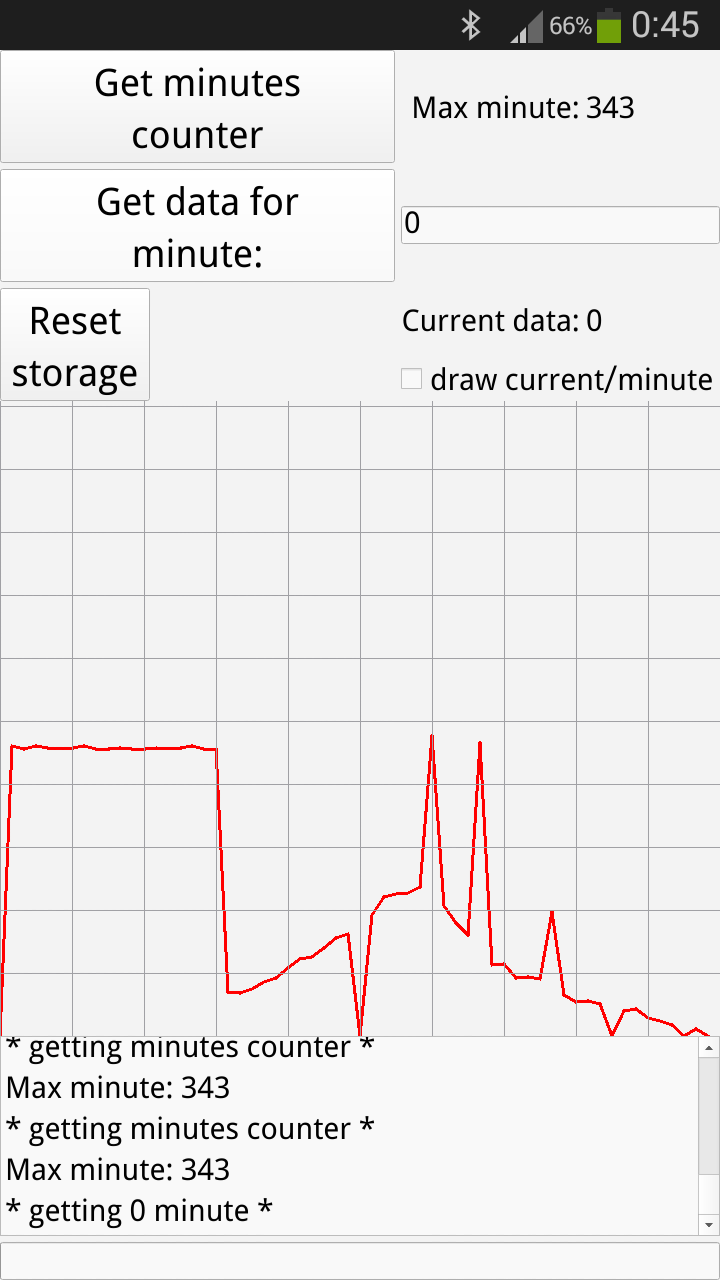
При создании проекта был выбран шаблон Qt Widgets Applications, а базовым классом – QWidget.

Для ускорения разработки из примера Bluetooth Low Energy Scanner Example были взяты классы CharacteristicInfo, Device, DeviceInfo, ServiceInfo. Они предоставляют несколько удобных функций для работы с классами QLowEnergyCharacteristic, QLowEnergyController, QLowEnergyService, и так далее. Но осуществлять всю работу через эти классы невыгодно – приходится писать много кода, не относящегося непосредственно к логике, а предоставляющего доступ к нужным элементам. Поэтому в классе Widget, где производится работа с графическим интерфейсом, добавлены объекты классов с приставкой QLowEnergy... .

В файле проекта adnroidBLE.pro к строке “QT += core gui” нужно добавить “bluetooth”.

При запуске приложения вызывается функция startDeviceDiscovery() у экземпляра класса Device, которая начинает поиск BLE-устройств. После успешного нахождения устройства с названием “MSP SD dst MAI” начинается процесс обнаружения сервисов устройства. Если были обнаружены все сервисы, то каждый из них по очереди исследуется на наличие необходимых характеристик.

Когда найдены все сервисы и раскрыты все характеристики, включается прием оповещений от устройства и графический интерфейс становится активным для взаимодействия.



*Рисунок 4. Графический интерфейс приложения.*

### Возможности приложения

Приложение может считывать общее количество минут, записанных на SD карту, передавать номер минуты для считывания, и сбросить записанные данные.

Приложение принимает оповещения от МК. МК каждую секунду отправляет текущее значение датчика расстояния, а также при запросе получения данных за указанную минуту.

Так как приложение разработано специально для данного ПАК, то пользователю не требуется что-либо вводить или выбирать. Весь доступный функционал представлен в виде кнопок и элементов информации в виде текстовых меток, графиков и прогресс-баров.

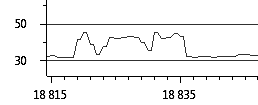
## Замеры энергопотребления

Для измерения энергопотребления использовалась программа EnergyTrace, входящая в состав CCS 7.1.0. Потребляемая энергия на графиках приводится в Ваттах. Так как напряжение, потребляемое ПАК равно 3,3 вольта, то для получения потребляемого тока необходимо разделить полученные Ватты на 3,3.

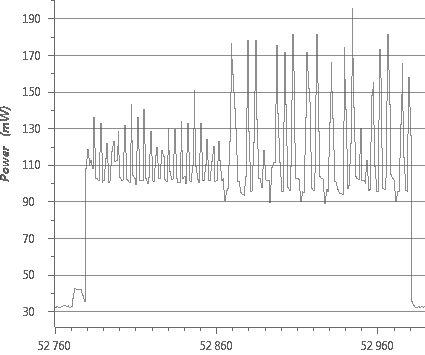
С помощью отдельного амперметра было выяснено, что модуль SD карты потребляет 4,7 мА, а датчик расстояния – 2,6 мА, поэтому из всех полученных значений было вычтено 7,3 мА

Были получены следующие результаты:

При замере расстояния в течении 0,1 мс потребляется ток около 5,7 мА.

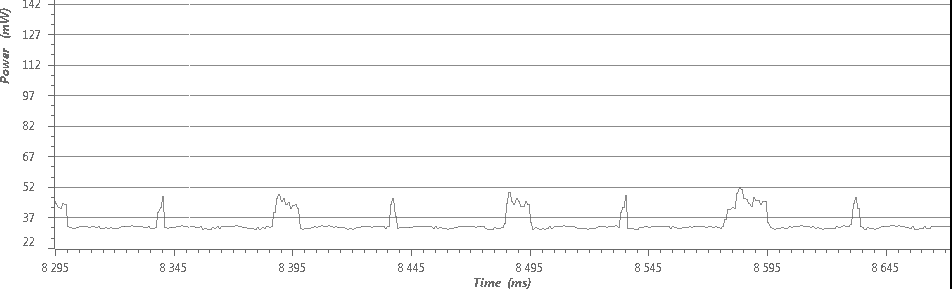


При записи минуты в течении 0,2 секунды потребляется ток от 25,7 мА до 37,7 мА.

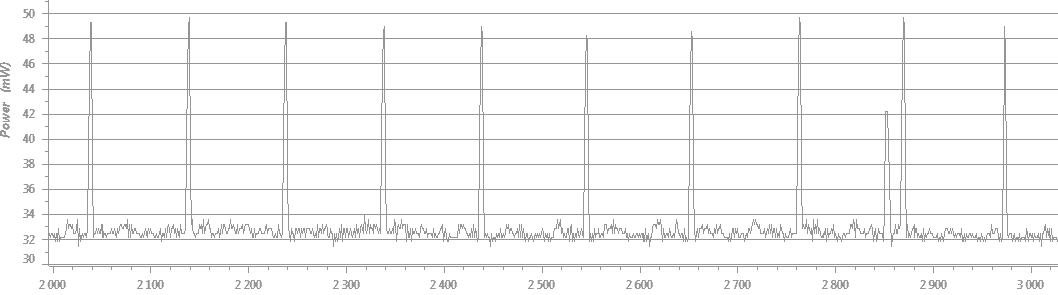


При передаче минуты примерно раз в 0,1 секунды происходит передача данных (широкие столбцы) длительностью 0,01 секунды и с потребляемым током 5,7 мА. Таких передач – 60, что соответствует времени ожидания получения минуты в приложении на смартфоне, равному примерно 6 секунд.

В начале передач происходит короткое обращение к карте памяти длительностью 0,01 секунды и потреблением 25,7 мА.



При отсутствии явных действий энергопотребление выглядит следующим образом:



Большую часть времени МК потребляет около 2,7 мА, что довольно экономно. При этом за секунду происходит 10 скачков до 7 мА и 1 скачок до 5,7 мА. Единичный скачок связан с работой таймера для получения данных от датчика расстояния, а остальные происходят из-за рассылки BLE. При наличии активного подключения эти скачки учащаются в два раза до 20 в секунду.

Таким образом, BLE в данных тестах потребляет не более 5 мА, что соответствует заявленной потребляемой мощности < 15 мА. [1]

### Оптимизация

Оптимизация энергопотребления путем использования функций из библиотеки power.h оказалась неэффективной. МК не переходит в состояние сна, а понижение производительности не снижает энергопотребление, зато замедляет процессы (в частности, записи на SD карту – самый «дорогого» по току), что дает отрицательный результат оптимизации.

# Заключение

В результате данной курсовой работы были изучены основы работы с TI RTOS, разработано устройство на основе микроконтроллера MSP432P401R с периферией и модулем CC2650 MODA. Были изучены принципы работы Bluetooth Low Energy и создано приложение для смартфона для управления микроконтроллером через BLE.

Список использованной литературы

[1]. Википедия – свободная энциклопедия [Электронный ресурс]. – https://en.wikipedia.org/wiki/Bluetooth\_Low\_Energy (дата обращения: 23.05.2017)

[2]. Texas Instruments [Электронный ресурс]. – http://www.ti.com/product/cc2650moda (дата обращения: 23.05.2017)

Приложение

Исходный код программы для микроконтроллера и приложения может быть получен из git-репозитория на сайте github: https://github.com/Wasd96/BLE-MSP432-Qt/tree/master.