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: «Реализация IoT устройства на базе МК MSP432»

Выполнил:

студент группы 3о-311Б

Авдонин Н.В.

Проверил:

Ассистент каф. 304

Аляутдин Р.Т.

# Задание

Разработать IoT устройство на базе платы MSP432P401R Launchpad, осуществляющее выход в сеть Интернет и обменивающееся данными с сервером. Разработать ВЕБ-интерфейс для управления этим устройством на примере изменения цвета светодиода на плате Launchpad.

Оглавление

[Задание…... 2](#_Toc484141877)

[Введение…. 3](#_Toc484141878)

[Теоретическая часть 5](#_Toc484141879)

[Используемые технологии 5](#_Toc484141880)

[TI RTOS………………….. 5](#_Toc484141881)

[Модуль SIM900D 5](#_Toc484141882)

[Работа с UART… 6](#_Toc484141883)

[Практическая часть 7](#_Toc484141884)

[Устройство клиента 7](#_Toc484141885)

[SIM900D…………... 8](#_Toc484141886)

[Разработка программного обеспечения микроконтроллера 10](#_Toc484141887)

[Устройство сервера 11](#_Toc484141888)

[Принцип обмена данными с клиентом 11](#_Toc484141889)

[Установка и настройка 11](#_Toc484141890)

[Заключение 13](#_Toc484141891)

[Список источников 14](#_Toc484141892)

# Введение

Так как MSP432P401R не обладает встроенным функционалом выхода в сеть Интернет, было решено воспользоваться модулем SIM900D, который может обеспечить выход в Интернет и работать с HTTP посредствам GSM.

Аппаратная часть устройства состоит из:

* Плата на базе микроконтроллера – MSP432P401R Launchpad;
* Модуль SIM900D.

Программная часть состоит из:

* ПО клиента (платы Launchpad);
* ПО сервера.

# Теоретическая часть

## Используемые технологии

При реализации *ПАК* был использован набор различных технологий.

Программа для микроконтроллера была написана на языке программирования С в среде разработки IAR для ARM. Запись программы на МК производилась с помощью отладочной платы XDS110-ET, которая входит в состав MSP‑EXP432P401R LaunchPad.

При реализации API сервера были использованы языки:

* PHP
* MySQL
* JS
* CSS
* HTML

## 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. Благодаря тому, что основа уже протестирована и интегрирована, TI-RTOS позволяет сфокусироваться на разработке конечного приложения. Тем не менее, при необходимости прямой доступ к аппаратной части остается.

## Модуль SIM900D

SIM900D – Компактный GSM/GPRS модуль, работающий в 4х GSM диапазонах со встроенным стеком TCP/IP, UDP/IP и поддержкой протоколов HTTP/FTP. Так же в нем реализованы все необходимые цепи для ввода/вывода аудио для звонков и функционал отправки/получения СМС. На сайте производителя сказано, что данный модуль снят с производства. На смену линейки SIM9XXX пришла линейка SIM8XXX. В ней можно найти более подходящие по энергопотреблению и габаритом решения, а ПО *скорее всего* будет совместимо.

## Работа с UART

Для общения с модулем используется протокол UART TTL уровня. Для работы с UART в TI RTOS есть специальные функции, а в репозитории IAR есть уже готовый пример.

* UART\_Handle - Объект обработчика UART.
* UART\_Params - Объект, в котором указывается скорость (.baudRate), стоп – биты (.stopBits) и т.д.
* UART\_open – функция открытия порта. В нее передается *номер* порта UART и ссылка на структуру параметров.
* UART\_writePolling – функция записи в порт. Была использована функция Polling, потому что вызов двух функций UART\_write может нарушить порядок отправляемых данных и их целостность.
* UART\_readPolling – функция считывания из порта.

Поверх этого API было написано несколько функций-оберток упрощающих обмен данными с модулем SIM900D.

# Практическая часть

## Устройство клиента

Клиент представляет собой связку модуля SIM900 и платы MSP432P401R – Launchpad:

### SIM900D

#### Подготовка модуля

Для того, что бы начать работу с SIM900D его нужно немного подготовить. Сначала надо убедиться, что он работает и регистрируется в сети GSM. Лучше всего использовать персональный компьютер c программой - терминалом и адаптер UART<-> USB. Был использован адаптер YP-02 и программа HTerm[2]. Используются *классические* настройки UART:

* Baud: 9600
* Data: 8
* Stop: 1
* Parity: None

Все команды, которые посылаются модулю должны оканчиваться “*\r\n*”. По этому выбираем в “Send on enter” “CR-LF”. Отвечает модуль тоже заканчивая “\r\n” По этому в “Newline at” можно выбрать “CR+LF”.

В качестве антенны был использована медная проволока длинной 103мм и сечением 0.7мм, но лучше использовать *нормальные* антенны, что бы не повредить ВЧ каскад. Питание подается на вывод VBAT модуля через диод Шотки. Так как верхняя граница рабочего напряжения модуля составляет 4.6В, а на диоде Шотки падает примерно 0.4В Модуль можно подключить к 5Вольтовому источнику питания. Важно отметить, что при регистрации в сети модуль может потреблять до 2А. Диод и источник питания *должны быть готовы* к таким нагрузкам.

Что бы запустить модуль нужно подать на него напряжение, а базу транзистора Q1 через резистор опустить на землю, примерно на секунду. Модуль может вывести несколько 0xFF байт. Если модуль отправит 0x00 то это значит что он выключился. Если модуль выключится до регистрации в сети – то он не укажет причину по которой отключился (нехватка тока, скорее всего).

Что бы понять работает модуль вообще – достаточно отправить команду “AT”, на что модуль должен ответить “OK”.

Дальше у него нужно спросить “AT+CREG?” на что модуль должен ответить “+CREG: 0,1” и “OK”. Если ответ отличается то стоит подождать. Если же модуль выводит “ERROR” то стоит включить “AT+CMEE=2” что обеспечит вывод *подробного* отчета об ошибке. Если же все работает, то нужно:

* Отключить вывод отчета об ошибке “AT+CMEE=0”
* Отключить локальное эхо “ATE0”
* Включить обычный режим вывода “ATV0”
* Сохранить эти настройки “AT&W”

Ответом на все это *должно* быть “OK”.

#### Алгоритм обмена данными с модулем SIM900D



В проверку работоспособности входят проверки на отклики на команды:

* AT
* AT+CREG?

В инициализацию входят команды:

* AT+GSMBUSY=1 для игнорирования звонков
* AT+SAPBR=3,1,"CONTYPE","GPRS"
* AT+SAPBR=3,1,"APN","internet.mts.ru"
* AT+SAPBR=3,1,"USER","mts"
* AT+SAPBR=3,1,"PWD","mts"
* AT+SAPBR=1,1
* AT+HTTPINIT
* AT+HTTPPARA="CID",1

Получение последнего запроса с сервера:

* AT+HTTPPARA="URL",http://77.37.152.65/msp/requestget.php”
* AT+HTTPACTION=0
* AT+HTTPREAD

Подробнее *API* сервера описан ниже.

Отправка ответа на запрос:

* AT+HTTPPARA="URL",http://77.37.152.65/msp/requestset.php?statusid=X&respond=Y&id=Z”
* AT+HTTPACTION=0

### Разработка программного обеспечения микроконтроллера

#### Среда разработки

В качестве среды разработки была выбрана IDE IAR для ARM версии 7.50.2.10505. В репозиториях IAR есть готовые примеры проектов для MSP432P401R как под TI RTOS так и без него. Что бы установить примеры нужно:

* В стартовом меню “Help>Information Center” нажать “Example projects”.
* В появившемся списке “Example projects that can be downloaded” найти “Texas Instruments” и нажать “Start Download” напротив.
* Выбрать “Download from IAR systems” и немного подождать.
* Позже нужно обновить этот список и сверху должно появиться “Texas Instruments” в разделе “Installed example projects”.
* Выбрав этот пункт появится еще один список, в котором нужно выбрать “MSP432” и далее “MSP432P4xx Standalone Examples ”.

Для установки примеров для TI RTOS достаточно следовать инструкции[1].

#### Реализация программного обеспечения

Поверх API, описанного в теоретической части было создано несколько функций для упрощенного общения с модулем, согласно схеме определенной выше:

* SIM900\_PollCommand – Оправляет команду в модуль с “\r\n”, пока тот не ответит “OK”. Полезно для обработки команд типа AT и AT+CREG?. Иногда некоторые команды (например, AT+HTTPINIT) могут выдать ошибку.
* SIM900\_ReadRespond – Считывает ответ модуля со всеми “\r\n” в нужных местах и исключает их из целевого буфера.
* SIM900\_WriteCommand – Посылает модулю команду в нужном формате.

Так как модулю требуется некоторое время на обработку команд, были вставлены задержки между командами для исключения ошибок.

## Устройство сервера

Принцип обмена данными с клиентом 

Для клиента важны всего две станицы:

* requestget.php:

В тексте этой станицы находится PHP скрипт, который выбирает первый из запросов в таблице оных, статус которого равен нулю (ожидание). И выводит его на страницу в формате:

ID REQUEST\r\n

Где ID – id запроса, а REQUEST его текст из таблицы запросов.

Если же таких запросов не было найдено (все выполнены, а новых еще не поступало) – то ID будет равно 0 и REQUEST будет “null”

* requestset.php:

В этот скрипт нужно передать 3 параметра:

* + ID – id запроса, который был получен из requestget.php
  + RESPOND – текст ответа на запрос, например “Done.”
  + STATUSID – Новый статус запроса: 1 – успешное выполнение 2 – Ошибка во время выполнения.

Пользователь же может создавать запросы используя ВЕБ-интерфейс представленный страницами requestview.html, requestcancel.html и requestcreate.html.

### Установка и настройка

Сервер представляет собой обычный компьютер с ОС Ubuntu 16.04.1 – server и классическим набором LAMP:

* Linux
* Apache
* MySQL
* PHP

Все это устанавливается командой sudo apt-get install lamp-server^ Все должно заработать, и при вводе в браузере вашего IP Нужно только включить обработку PHP скриптов в HTML коде[3], и создать необходимые структуры в MySQL:

* Запускаем MySQL и вводим пароль, заданный при установке:

>mysql –u root –p

>\*\*\*\*\*\*\*\*

* Далее создаем БД и *переходим* в нее:

CREATE DATABASE msp

USE msp

* Создадим таблицы:

CREATE TABLE request (id int, dateserver datetime, statusid int, request varchar(80), respond varchar(80))

CREATETABLE status (id int, meaning varchar(16), img varchar(32))

* И заполним таблицу статусов:

INSERT INTO status (id, meaning, img) values (0, ‘Awaiting’, ‘http://i.imgur.com/fTZrRqt.png’)

INSERT INTO status (id, meaning, img) values (1, ‘Success’, ‘http://i.imgur.com/mrXagtY.png’)

INSERT INTO status (id, meaning, img) values (2, ‘Error’, ‘http://i.imgur.com/r95mOMF.png’)

INSERT INTO status (id, meaning, img) values (3, ‘Canceled’, ‘http://i.imgur.com/JlkDKcS.png’)

* И создадим пользователя:

CREATE USER ‘board’@’localhost’ IDENTIFIED BY ‘keyword’

# 

# Заключение

В ходе выполнения данной курсовой работы было разработано IoT устройство на базе микроконтроллера MSP432P401R и модуля SIM900 и создан ВЕБ-интерфейс для взаимодействия с ним. Были изучены принципы работы с TI RTOS и IDE IAR для ARM.

# Список источников

[1] Инструкция по установке TI-RTOS для IAR <http://processors.wiki.ti.com/index.php/Creating_TI-RTOS_Applications_in_IAR_Embedded_Workbench>

[2] Программа-терминал <http://www.der-hammer.info/terminal/>

[3] Включение скриптов внутри HTML страниц <https://stackoverflow.com/questions/6295141/server-not-parsing-html-as-php>

[4] Репозиторий <https://github.com/BatyaStudios/CourseWork6>