

УДК 004.415.2.031.43

## Программно-аппаратное средство контроля местоположения

***Строганов И.С.**, студент*

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,  
кафедра «Информационная безопасность»*

***Алешин В.В.**, студент*

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,  
кафедра «Информационная безопасность»*

***Рафиков К.А.**, студент*

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,  
кафедра «Информационная безопасность»*

*Научный руководитель: **Алешин В.А.**, к.т.н., доцент*

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,  
кафедра «Информационная безопасность»*

*[v.aleshin@bmstu.net](mailto:v.aleshin@bmstu.net)*

### Используемые термины и сокращения.

COM-порт – communication port – последовательный порт;

GPS – global position system – система глобального позиционирования;

GPRS – general packet radio service – пакетная радиосвязь общего пользования;

GPRMC – recommended minimum specific GPS data – рекомендуемый минимум навигационных данных;

GSM – global system for mobile communications – глобальная система мобильной связи;

TCP – transmission control protocol – протокол управления передачей

IP – internet protocol – межсетевой протокол;

USB – universal serial bus – универсальная последовательная шина.

## **Введение**

Принцип работы системы спутникового мониторинга заключается в отслеживании и анализе пространственных и временных координат контролируемого объекта [1]. На подвижном объекте устанавливается мобильная система, состоящая из следующих частей:

- 1) модуль для приема спутниковых сигналов;
- 2) модуль передачи данных по сетям операторов мобильной связи;
- 3) модуль хранения;
- 4) микроконтроллер.

Микроконтроллер получает данные координат от приемника спутниковых сигналов, записывает их в модуль хранения и отправляет на сервер с помощью модуля передачи данных. Полученные данные анализируются и выдаются диспетчеру в текстовом виде или с использованием картографической информации.

## **Настройка и моделирование системы**

В данной работе для построения системы контроля местоположения используются следующие модули:

- 1) GPS-приемник GlobatSat GT-100 на базе модуля Sirf Star III;
- 2) GSM/GPRS-модуль Cinterion MC52i, имеющий встроенный стек TCP/IP;
- 3) микроконтроллер PIC 18LF46K22, имеющий два COM-порта.

Микроконтроллер взаимодействует с остальными устройствами через COM-порты. Он получает данные от GPS-приемника, выбирает сообщения GPRMC (рекомендуемый минимум навигационных данных) и отправляет их на сервер с помощью AT-команд GSM/GPRS-модема, используя его встроенный стек TCP/IP.

Для разработки программного обеспечения микроконтроллера используется компилятор CCS PIC C Compiler 5.015.

Моделирование системы выполняется в Proteus 7.10 под управлением операционной системы Windows 8.1.

При подключении GPS-приемника и GSM/GPRS-модуля через USB к персональному компьютеру для них создаются виртуальные COM-порты, что видно в диспетчере устройств (рис. 1).

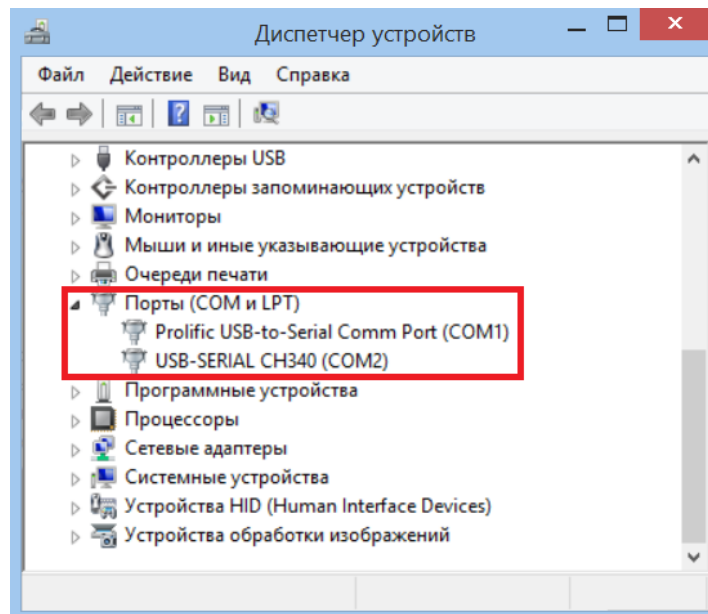


Рис. 1. Виртуальные COM-порты в диспетчере устройств

Настройка виртуальных COM-портов в данном случае заключается в выборе скорости передачи данных и номера порта (рис. 2а, 2б):

- 1) Prolific USB-to-Serial Comm Port (GPS-приемник): COM1, 4800 бит/с;
- 2) USB-SERIAL CH340 (GSM/GPRS-модуль): COM2, 4800 бит/с.

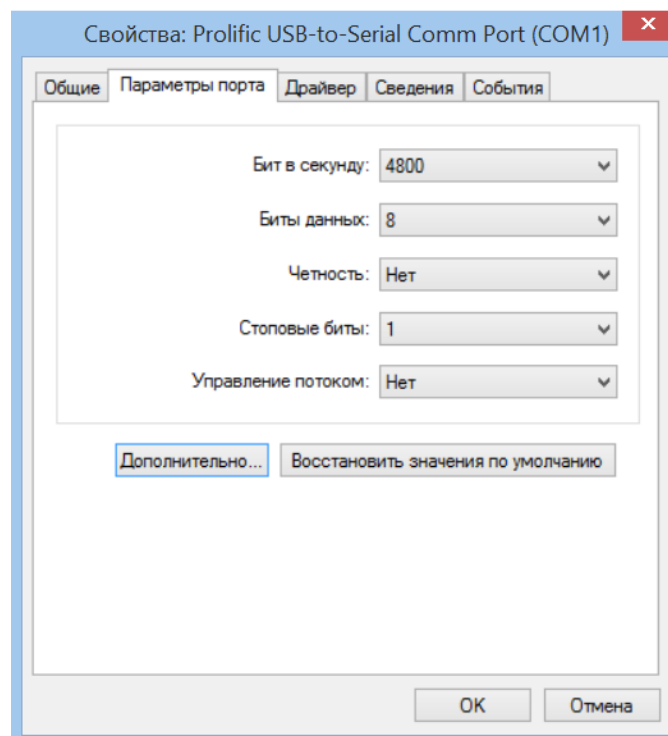


Рис. 2а. Свойства COM-порта в диспетчере устройств

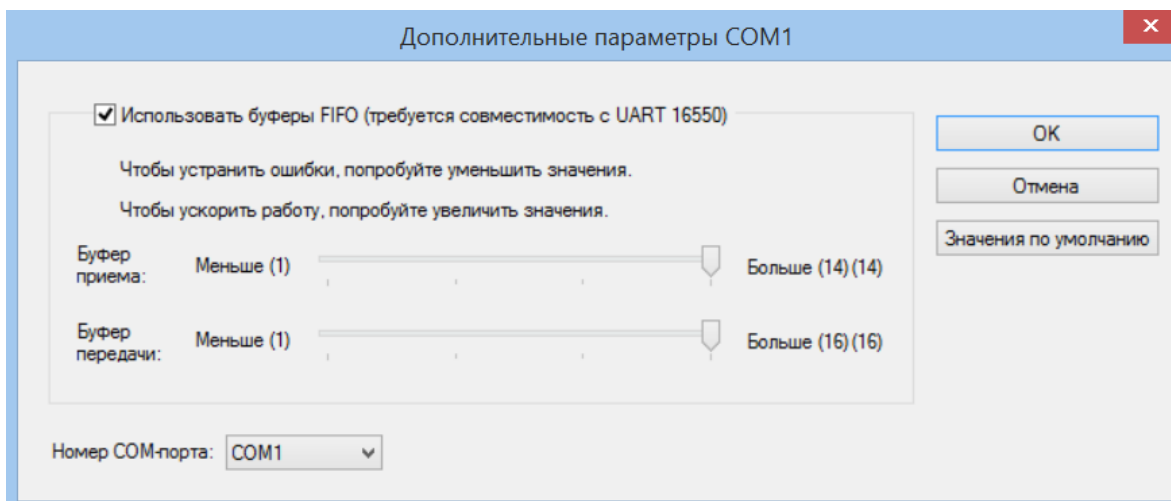


Рис. 26. Дополнительные настройки COM-порта в диспетчере устройств

Модель выбранного микроконтроллера присутствует в Proteus, а моделирование остальных устройств выполняется с помощью компонента COMPIM, который позволяет подключиться к реальному COM-порту компьютера. Настройка компонента сводится к согласованию параметров реального COM-порта и эмулируемого в схеме (рис. 3).

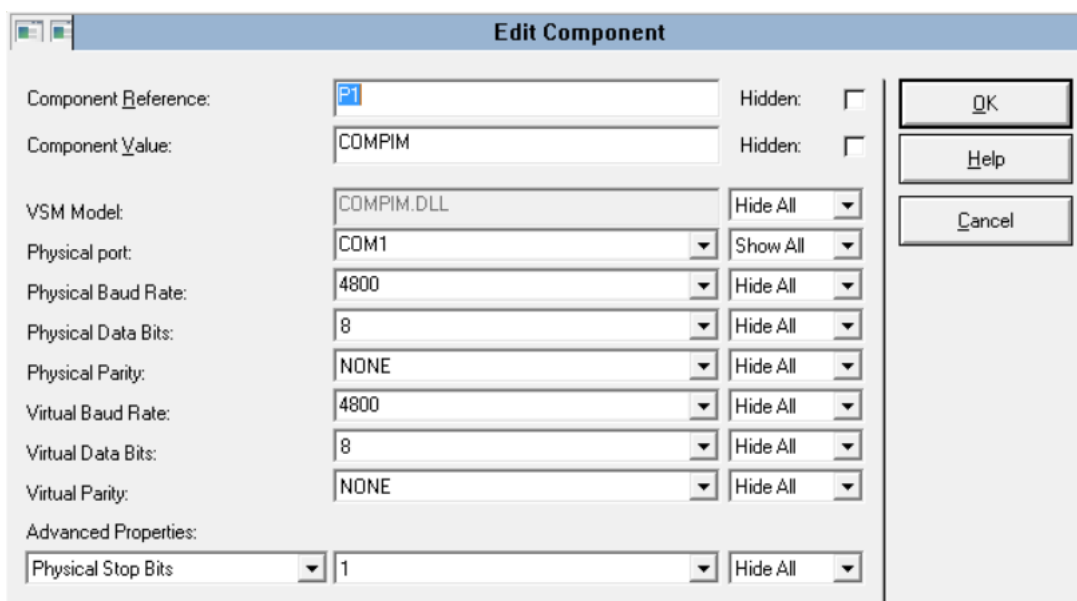


Рис. 3. Свойства компонента COMPIM в Proteus

Схема в Proteus для моделирования системы выглядит следующим образом (рис. 4).

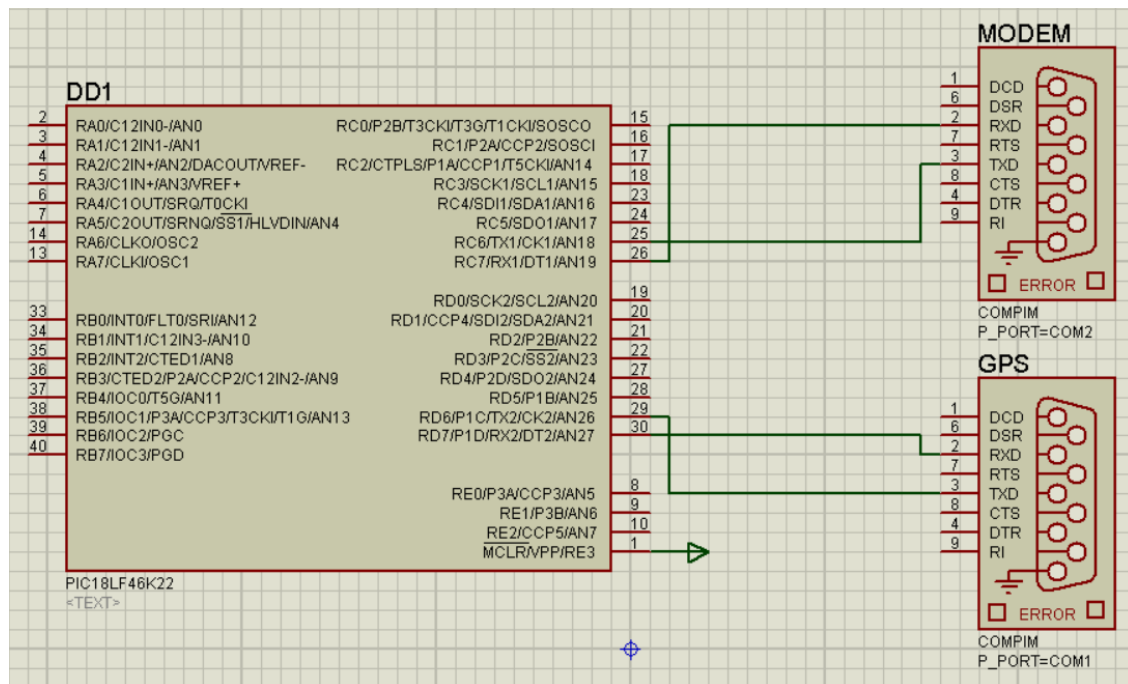


Рис. 4. Схема в Proteus для моделирования системы

### Описание исходного кода программного обеспечения микроконтроллера

Задание тактовой частоты генератора (4 МГц):

```
#use delay(crystal=4000000)
```

Настройка параметров первого COM-порта для подключения GSM/GPRS-модуля:

```
#use RS232(UART1, baud=4800, stream=MODEM, BITS=8, ERRORS)
```

Настройка параметров второго COM-порта для работы GPS-модуля:

```
#use RS232(UART2, baud=4800, stream=GPS, BITS=8, ERRORS)
```

Задержка перед сбором данных с GPS-приемника после очередной отправки на сервер (в миллисекундах):

```
int16 delay = 3000;
```

Шаблон GET-запроса к серверу:

```
char *szServer = "http://u3192s88k.hn3.justhost.ru/cgi-bin/datasaver.cgi?GPRMC=";
```

Функция проверки корректности выполнения AT-команды модема:

```
int8 modem_check_response(char iNumberOfMarks)
```

```
{
    char c, cFlag = 0, cEnd = 0;
    char cFlagOK = 0;
    int8 exit_code = 0;
    while(1)
    {
```

```

    if(cEnd == iNumberOfMarks)
        break;
    c = getc(MODEM);
    if(c == 0x0D)
        cFlag = 1;
    else if(cFlag && c == 0x0A)
        cEnd++;
    else
        cFlag = 0;
    if(c == 'O')
        cFlagOK = 1;
    else if(cFlagOK == 1 && c == 'K')
        exit_code = 1;
    else
        cFlagOK = 0;
}
return exit_code;
}

```

Функция для инициализации модема:

```

int8 modem_init()
{
    // Отключение "эхо" (возвращения команд)
    fprintf(MODEM, "ate0\r");
    if(!modem_check_response(2))
        return 0;

    // Установка типа подключения – GPRS
    fprintf(MODEM, "at^sics=0,conType,GPRS0\r");
    if(!modem_check_response(3))
        return 0;

    // Установка точки доступа для оператора мобильной связи МТС
    fprintf(MODEM, "at^sics=0,apn,internet.mts.ru\r");
    if(!modem_check_response(3))
        return 0;

    // Установка имени пользователя для подключения

```

```

fprintf(MODEM, "at^sics=0,user,mts\r");
if(!modem_check_response(3))
    return 0;
// Установка пароля для подключения
fprintf(MODEM, "at^sics=0,passwd,mts\r");
if(!modem_check_response(3))
    return 0;
// Установка неограниченного времени ожидания подключения
fprintf(MODEM, "at^sics=0,inactTO,0\r");
if(!modem_check_response(3))
    return 0;
// Установка протокола аутентификации – PAP
fprintf(MODEM, "at^sgauth=1\r");
if(!modem_check_response(2))
    return 0;
// Установка сервиса – HTTP
fprintf(MODEM, "at^siss=0,srvType,Http\r");
if(!modem_check_response(3))
    return 0;
// Установка идентификатора соединения, указанного при задании типа подключения
fprintf(MODEM, "at^siss=0,conId,0\r");
if(!modem_check_response(3))
    return 0;
// Установка метода запроса – GET
fprintf(MODEM, "at^siss=0,hcMethod,0\r");
if(!modem_check_response(3))
    return 0;
return 1;
}

```

Функция для выполнения GET-запроса на сервер:

```

int8 modem_send_data(char *szRequest)
{
    // Задание параметров профиля интернет-соединения (Internet Service Profile)
    fprintf(MODEM, "%s%s%s", "at^siss=0,address,\"", szRequest, "\"\r");
}

```

```

if(!modem_check_response(3))
    return 0;
// Установка соединения с сервером и выполнение GET-запроса
fprintf(MODEM, "at^siso=0\r");
if(!modem_check_response(5))
    return 0;
// Закрывание соединения
fprintf(MODEM, "at^sisc=0\r");
if(!modem_check_response(3))
    return 0;
return 1;
}

```

Обработчик прерывания от второго COM-порта (обработка поступления данных от GPS-приемника):

```

#INT_RDA2
void modem_send_gps_data()
{
    static char c, i = 0, cBegin = 0, cEnd = 0;
    static char szGPS[85];
    static char data[38][85] = {0};
    static long j = 0;
    const char szGPRMC[] = "$GPRMC";
    char iLength = 6;
    c = getc(GPS);
    if(cBegin < iLength)
    {
        if(c == szGPRMC[cBegin])
            cBegin++;
        else
            cBegin = 0;
    }
    else if(cBegin == iLength)
        cBegin++;
    else if(cBegin == iLength + 1)

```



```

{
    if(c == 0x0D)
        cEnd = 1;
    else if(cEnd == 1 && c == 0x0A)
    {
        szGPS[i-1] = 0;
        cBegin = 0;
        cEnd = 0;
        i = 0;
        disable_interrupts(INT_RDA2);
        char szRequest[160];
        sprintf(szRequest, "%s%s", szServer, szGPS);
        modem_send_data(szRequest);
        return;
    }
    else
        cEnd = 0;
    szGPS[i++] = c;
}
}

```

Главная функция программы:

```

void main()
{
    // Инициализация модема
    modem_init();

    // Разрешение прерываний
    enable_interrupts(global);

    while(1)
    {
        // Разрешение прерываний от UART2
        enable_interrupts(INT_RDA2);
    }
}

```

```

// Задержка на заданное число миллисекунд
delay_ms(delay);
}
}

```

### **Описание исходного кода PERL-скрипта на сервере**

PERL-скрипт (*/cgi-bin/datasaver.cgi*) обрабатывает поступающие GET-запросы, осуществляет разбор полученных сообщений GPRMC и запись в файл следующих данных:

- 1) дата и время фиксации местоположения в формате UTC;
- 2) достоверность полученных данных (A – достоверны, V – недостоверны);
- 3) широта и долгота;
- 4) горизонтальная составляющая скорости относительно земли (в узлах);
- 5) путевой угол (в градусах).

Исходный код PERL-скрипта:

```

#!/usr/bin/perl

use strict;

print "Content-type: text/html\n\n";

# Получение строки GET-запроса
my $query = $ENV{QUERY_STRING};

# Декодирование запроса и формирование ассоциативного массива
my @array = split(/&/=/, $query);
for (@array)
{
    $_ =~ s/\+//g;
    $_ =~ s/%([a-zA-F0-9]{2})/chr(hex($1))/ge;
}
my %hash = @array;

# Разбор полученного сообщения GPRMC

```

```
my ($utctime, $correctness, $latitude, $latitude_NS, $longitude, $longitude_WE, $speed, $direction, $date) = split(/,/,$hash{GPRMC});
```

```
# Запись полученных значений в файл (data.txt)
```

```
my $file = '../data.txt';
```

```
if(!-e $file && open(FL, "> $file"))
```

```
{
```

```
    my $mode = 0640;
```

```
    close(FL);
```

```
    chmod($mode, $file);
```

```
}
```

```
if(open(FL, ">> $file"))
```

```
{
```

```
    print(FL "Date: $date\nTime: $utctime\nCorrectness: $correctness\nLatitude: $latitude\n$latitude_NS\nLongitude:    $longitude    $longitude_WE\nSpeed:    $speed\nDirection: $direction\n\n");
```

```
    close(FL);
```

```
}
```

### **Тестирование взаимодействия микроконтроллера с GPS-приемником в автономном режиме**

Схема для тестирования взаимодействия микроконтроллера с GPS-приемником в автономном режиме выглядит следующим образом (рис. 5).

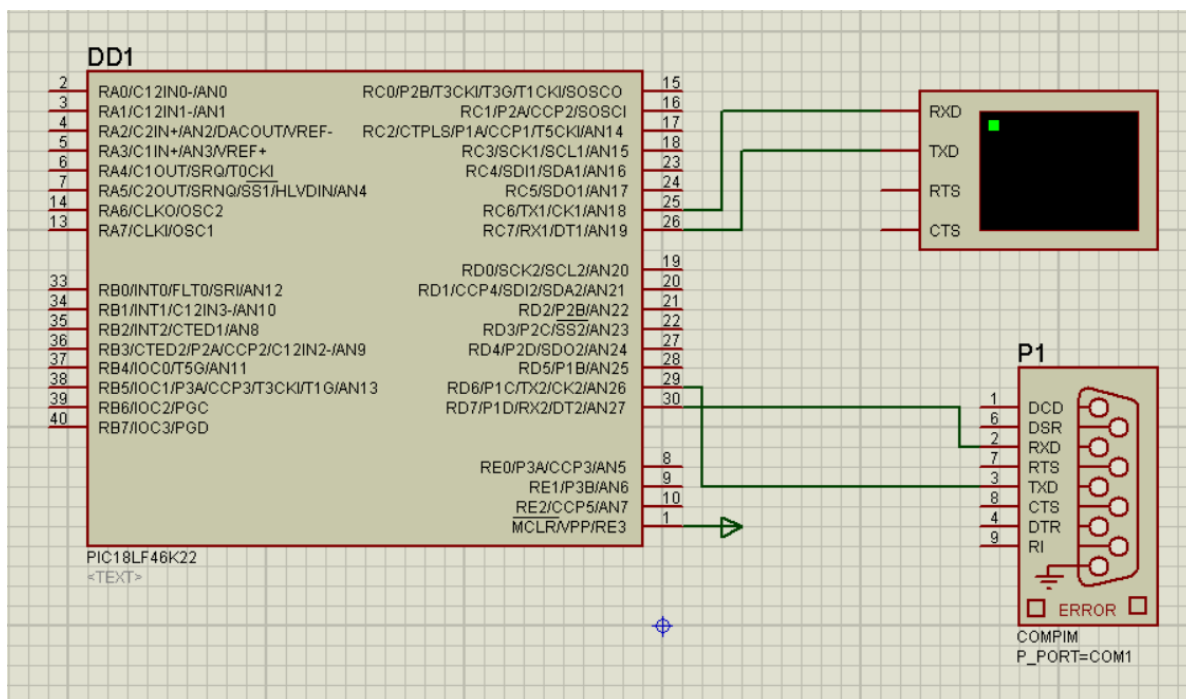


Рис. 5. Схема для тестирования взаимодействия микроконтроллера с GPS-приемником

Считывая данные с GPS-приемника, микроконтроллер формирует GET-запросы, которые в данном случае выводятся на терминал (рис. 6).

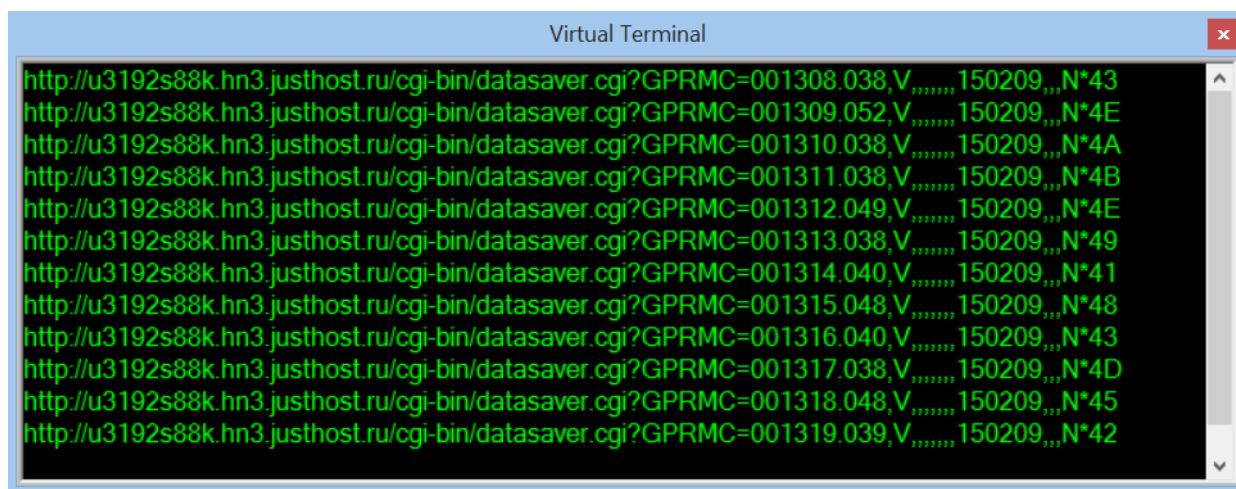


Рис. 6. Тестовые GET-запросы от микроконтроллера на основе данных GPS

## Выводы.

Разработан прототип системы контроля местоположения с использованием микроконтроллера, GPS-приемника и GSM/GPRS-модуля.

В рабочей версии системы должны быть также учтены вопросы безопасности, связанные с проверкой со стороны сервера подлинности источника поступающих

сообщений, а также возможность использования картографической информации для отображения текущего местоположения контролируемого объекта.

### **Список литературы**

1. Википедия – свободная энциклопедия. Режим доступа: <http://wikipedia.org> (дата обращения 07.04.2015).
2. Уилмсхерст Т. Разработка встроенных систем с помощью микроконтроллеров PIC: Принципы и практические примеры. СПб.: КОРОНА-БЕК, 2008. 544 с. [Wilmshurst T. Designing Embedded Systems with PIC Microcontrollers: Principles and Applications. Newnes, 2007. 583 p.].
3. Шпак Ю.А. Программирование на языке С для AVR и PIC микроконтроллеров. СПб.: КОРОНА-БЕК, 2011. 546 с.