



ОПУБЛИКОВАТЬ СТАТЬЮ



ENGLISH



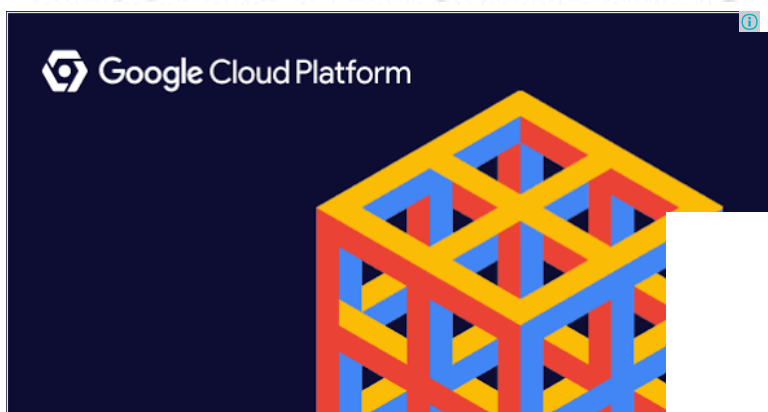
Войти

Только \$5 Прототип печатной платы
Время выполнения 2~3 дня

Заказать сейчас

Главная » Микроконтролле

Интерфейс связи между SIM300 и ATmega32



Start your free trial

Deploy on an infrastructure protected by more than 700 top experts

GSM/GPRS модуль [SIM300](#), может применяться в устройствах, где необходима связь на большие расстояния. Например, роботом в Москве человек управляет сидя в Краснодаре! Или фермер включает водяной насос на рисовом поле из своего дома, расположенного за несколько километров от поля! Есть несколько вариантов связи с устройством:

Простая связь на основе SMS сообщений:

Включение/выключение устройства при помощи простых SMS команд. Для управления устройством можно использовать любой мобильный телефон.

Охранный/пожарная сигнализация, которая информирует хозяина о чрезвычайной ситуации в доме при помощи SMS.

Связь на основе вызовов:

“Умная” охранный/пожарная сигнализация, которая вызывает полицию или пожарных и сообщает о чрезвычайной ситуации при помощи заранее записанных голосовых сообщений.

Связь с использованием интернета (GPRS):

Пользователь может управлять устройством с любого ПК/планшета/мобильного телефона, подключенного к интернету. Например, информационные дисплеи, установленные на трассах, управляются из центральной диспетчерской.

Робот, управляемый через интернет. Такой робот доступен с любого устройства подключенного к интернету из любой точки мира.

Портативные устройства, установленные в транспортных средствах, которые подключаются к интернету с помощью GPRS модуля SIM300 и добавляют текущую позицию (с помощью GPS (Global Position System, Глобальная Система Позиционирования)) на сервер. Эти данные сохраняются в базу данных на сервере вместе с идентификатором автомобиля. Для просмотра маршрута автомобиля можно соединиться с сервером с компьютера при помощи World Wide Web (Всемирной Паутины).

Преимущества использования модуля SIM300

Набор [SIM300 Kit](#) является полностью самостоятельным модулем с разъёмом SIM-карты, блоком питания и т.д. Этот модуль может быть легко связан с дешевыми микроконтроллерами AVR/PIC/8051. Связь с микроконтроллером осуществляется через асинхронный последовательный порт. Это основной тип последовательной связи, который аппаратно поддерживается большинством микроконтроллеров. Данные передаются бит за битом и собираются в байты. На высоком уровне это выглядит как простой текстовый поток. Всего потоков два: один от микроконтроллера к SIM300 и другой от SIM300 к микроконтроллеру. Команды передаются как

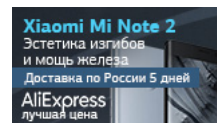
Призовой фонд
на декабрь 2017 г.

1. **Спектроанализатор**
Arinst SSA-TG LC
Крокс
2. **Термометр Relsib WT51**
Рэлсиб
3. **1000 руб**
PCBWay
4. **100 руб.**
От пользователей

присоединиться

Публикуя
статью или
обзор прибора
на сайте вы:

Заметили ошибку в тексте,
схеме или печатной плате?

Выделите ее
и нажмите

простой текст.

Если вы никогда не использовали последовательную передачу данных и не слышали о ней, то желательно разобраться в её работе и попрактиковаться на более простых примерах.

Связь с модулем SIM300 при помощи AVR UART

Аппаратная часть микроконтроллера, используемая для последовательной связи, называется UART, и мы используем его для связи с модулем SIM300 (Также он может использоваться для связи с другими устройствами, например считывателями RFID, GPS модулями, сканерами отпечатков пальцев и т.д.). UART является очень распространенным способом связи в мире электроники, мы написали для него чистую и простую библиотеку, которую мы применяем во всех своих проектах с использованием UART.

Поскольку байт от SIM300 может прийти на микроконтроллер в любое время, то, что произойдет, если микроконтроллер занят чем-то другим? Чтобы решить эту проблему, мы сделали буферизацию входящих данных, основанную на прерываниях. Буфер находится в оперативной памяти микроконтроллера. У него есть функция, позволяющая определить количество байт в очереди.

Ниже приведены функции библиотеки AVR USART:

void USARTInit(uint16_t ubrrvalue)

Инициализация аппаратной части AVR USART. Значением параметра **ubrrvalue** устанавливается желаемая скорость передачи данных. По умолчанию скорость передачи данных для SIM300: 9600 бит/сек. Для микроконтроллера AVR работающего на частоте 16 МГц значение **ubrrvalue** для такой скорости должно быть **103**.

char UReadData()

Чтение одного символа из очереди. Если в очереди ничего нет, то ответ 0.

void UWriteData(char data)

Записывает один байт данных на линию Tx, используя функцию UWriteString ().

uint8_t UDataAvailable()

Сообщает количество данных в очереди FIFO.

void UWriteString(char *str)

Записывает строку в Си стиле, оканчивающуюся нуль символом в линию Tx.

Пример 1: UWriteString("Hello World !");

Пример 2: char name[]="Avinash !"; UWriteString(name);

void UReadBuffer(void *buff,uint16_t len)

Копирует содержимое FIFO буфера в память, определенную buff, количество скопированных данных определяется параметром len. Если по UART в FIFO буфер пришло меньше данных, чем надо (в соответствии с параметром len), то оставшееся место будет заполнено нулями.

Примеры:

```
char gsm_buffer[128];
UReadBuffer(gsm_buffer,16);
```

Приведенный выше пример будет считывать 16 байт данных (если они есть) из FIFO буфера в переменную **gsm_buffer**. Обратите внимание, что **gsm_buffer** выделен массив 128 байт, поскольку позже нам может потребоваться более 16 байт. Таким образом, этот буфер можно будет использовать для чтения до 128 байт в дальнейшем.

Функция, показанная выше, обычно применяется вместе с UDataAvailable ().

```
while(UDataAvailable())<16)
{
//Do nothing
}
```

```
char gsm_buffer[128];
UReadBuffer(gsm_buffer,16);
```

Фрагмент кода показанный выше, ждет пока в буфере накопится 16 байт данных, а затем считывает их.

void UFlushBuffer()

Отменяет ожидание данных FIFO буфером. Прежде чем отправлять новую команду GSM модулю, сначала отмените ожидание данных FIFO буфером.

Приведенные выше функции используются для отправки и получения текстовых команд от GSM модуля SIM300.

Набор AT команд для SIM300

Теперь, когда Вы знакомы с основами библиотеки AVR USART и её использованием для инициализации USART, отправки и получения данных, настало время, чтобы изучить команды модуля SIM300 и как отправлять их и принимать ответы. SIM300 имеет несколько функций: отправка текстового сообщения, звонок и т.д. Каждая из этих функций выполняется после определённой команды, и SIM300 имеет свой набор команд.

Все команды SIM300 начинаются с префикса **AT+** и заканчиваются **Carriage Return** (сокращенно, возврат каретки). ASCII код CR - 0x0D (десятичное 13). Все команды, которые вы отправляете



Надежные
промышленные
контроллеры

ipс2u.ru



Лампы для любых
проекторов

hiteklamp.ru



Бесплатное
обучение
от MTBankFX

mtbank.by



Содействие в подборе финансовых услу...

Продажа блоков
питания 12/24V

diodlenta.by



Яндекс Директ

Поддержи
авторов статей

SIM300, будут возвращаться по TX линии SIM300. То есть если вы отправляете команду 7 байт (включая завершающий CR), то вы сразу получите эти 7 байт в буфер по UART. Если вы не получили её, то это значит, что-то не в порядке!

Первая функция, которую мы изучим будет **SIM300Cmd(const char *cmd)**, она выполняет следующие действия:

- Пишет команды, заданные параметром **cmd**.
- Добавляет **CR** после команды.
- Ожидает возврата команды, и если она приходит до тайм-аута, она отвечает **SIM300_OK** (константа, определенная в sim300.h). Если возврата ждали слишком долго, а его не было, она отвечает **SIM300_TIMEOUT**.

Примечание: Все зависимые функции SIM300 хранятся в файле sim300.c. Образцы и константы хранятся в sim300.h

Работа с SIM300Cmd ()

```
1 int8_t SIM300Cmd(const char *cmd)
2 {
3     UWriteString(cmd); //Send Command
4     UWriteData(0x0D); //CR
5
6     uint8_t len=strlen(cmd);
7
8     len++; //Add 1 for trailing CR added to all commands
9
10    uint16_t i=0;
11
12    //Wait for echo
13    while(i < 10*len)
14    {
15        if(UDataAvailable() < len)
16        {
17            i++;
18            _delay_ms(10);
19            continue;
20        }
21        else
22        {
23            //We got an echo
24            //Now check it
25            UReadBuffer(sim300_buffer,len); //Read serial Data
26
27            return SIM300_OK;
28        }
29    }
30
31    return SIM300_TIMEOUT;
32 }
```

За командой обычно следует ответ. Форма ответа такая: <CR><LF><response><CR><LF>
LF - Line Feed, его ASCII код 0x0A (10 в десятичной системе)

Таким образом, во время ожидания ответа после отправки команды могут случиться три вещи:

- Ответа нет в течении длительного времени. Вероятной причиной может быть то, что SIM300 не подключен к микроконтроллеру.
- Ответ получен, но не тот, который ожидался. Причиной может быть неисправность последовательной линии, неверно установленная скорость передачи данных или микроконтроллер, работающий с неверной частотой.
- Получен правильный ответ.

Например, команда **Get Network Registration** (Регистрация в сети) выполняется следующим образом: Command String (Команда): **"AT+CREG?"**

Response (Ответ): **+CREG: , OK**

Вы видите правильный ответ 20 байт. То есть после отправки команды "AT + CREG?" необходимо ждать получения 20 байт или пока истечет определенное время. Второе условие выполняется во избежание зависания, если SIM300 неисправен. То есть вместо того, чтобы вечно ждать ответа, будет выдана ошибка, если SIM300 отвечает слишком долго (это называется тайм-аут)

Если получен правильного ответа, то мы анализируем переменную для получения информации о регистрации в сети.

В зависимости от текущего состояния регистрации в сети значение может быть: 0 - Не зарегистрирован, сейчас SIM300 не ищет нового оператора для регистрации. 1 - Зарегистрирован в домашней сети. 2 - Не зарегистрирован, сейчас SIM300 ищет нового оператора для регистрации. 3 - В регистрации отказано. 4 - Неизвестно. 5 - Зарегистрирован, роуминг.

Работа с SIM300GetNetStat ()

```
1 int8_t SIM300GetNetStat()
2 {
3     //Send Command
4     SIM300Cmd("AT+CREG?");
5
6     //Now wait for response
7     uint16_t i=0;
8
9     //correct response is 20 byte long
10    //So wait until we have got 20 bytes
11    //in buffer.
12    while(i<10)
13    {
14        if(UDataAvailable()<20)
15        {
```

```

16         i++;
17         _delay_ms(10);
18     }
19     continue;
20 }
21 else
22 {
23     //We got a response that is 20 bytes long
24     //Now check it
25     UReadBuffer(sim300_buffer,20);    //Read serial Data
26
27     if(sim300_buffer[11]=='1')
28         return SIM300_NW_REGISTERED_HOME;
29     else if(sim300_buffer[11]=='2')
30         return SIM300_NW_SEARCHING;
31     else if(sim300_buffer[11]=='5')
32         return SIM300_NW_REGISTERED_ROAMING;
33     else
34         return SIM300_NW_ERROR;
35 }
36 }
37 }
38
39 //We waited so long but got no response
40 //So tell caller that we timed out
41
42 return SIM300_TIMEOUT;
43 }

```

Точно так же реализована функция: `int8_t SIM300IsSIMInserted()`

При другом типе ответов мы не знаем заранее точный размер ответа как в приведенной выше команде. Например, это команда Get Service Provider Name(Получение названия оператора (провайдера)), где длина имени оператора неизвестна заранее. Это может быть MTS, Beeline и т.п.. Для решения этой проблемы мы пользуемся тем, что перед и после ответа находится **CR LF**. Таким образом, мы просто записываем в буфер все символы до тех пор, пока мы не встречаем **CR**, что означает конец ответа.

Для упрощения обработки таких команд, мы сделали функцию
SIM300WaitForResponse (uint16_t timeout)

Эта функция ждет ответа от SIM300 (конец ответа обозначается CR) и сообщает размер ответа, в то время, когда сам ответ копируется в глобальную переменную `sim300_buffer []`.

Если ответ не получен до тайм-аута, то ответ 0. Время тайм-аута в миллисекундах можно задать параметром **timeout**. Она не считает запаздывающие LF или последние OK , они остаются в UART FIFO буфере. Поэтому перед возвратом мы используем команду **UFlushBuffer ()**, чтобы удалить их из буфера.

Работа с SIM300WaitForResponse (uint16_t timeout)

```

1  int8_t SIM300WaitForResponse(uint16_t timeout)
2  {
3      uint8_t i=0;
4      uint16_t n=0;
5
6      while(1)
7      {
8          while (UDataAvailable()==0 && n

```

Работа с SIM300GetProviderName (char *name) Функция выполняет следующие действия:

1. Очищает USART буфер, чтобы удалить все ошибки или ответы.
- 2.
3. Отправляет команду "AT + CSPN?" используя функцию SIM300Cmd ("AT + CSPN?");
- 4.
5. Затем она ждет ответа, используя функцию SIM300WaitForResponse ()
- 6.
7. Если мы получаем не нулевой ответ, она разбирает его чтобы получить название оператора.
- 8.

Подобным образом реализованы следующие функции:

```

> uint8_t SIM300GetProviderName(char *name)
>
> int8_t SIM300GetIMEI(char *imei)
>
> int8_t SIM300GetManufacturer(char *man_id)
>
> int8_t SIM300GetModel(char *model)
>

```

```

1  uint8_t SIM300GetProviderName(char *name)
2  {
3      UFlushBuffer();
4
5      //Send Command
6      SIM300Cmd("AT+CSPN?");
7
8      uint8_t len=SIM300WaitForResponse(1000);
9
10     if(len==0)
11         return SIM300_TIMEOUT;
12
13     char *start,*end;
14     start=strchr(sim300_buffer, ' ');
15     start++;
16     end=strchr(start, ' ');
17
18     *end='\0';

```

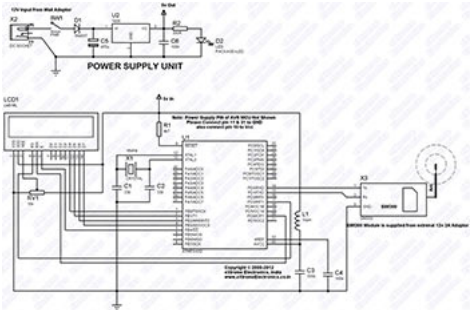


```
19  
20     strcpy(name,start);  
21  
22     return strlen(name);  
23 }
```

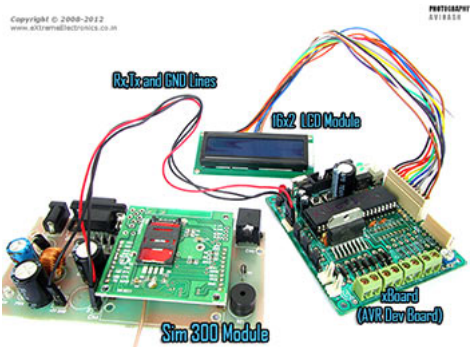
SIM300 и ATmega32. Аппаратная часть

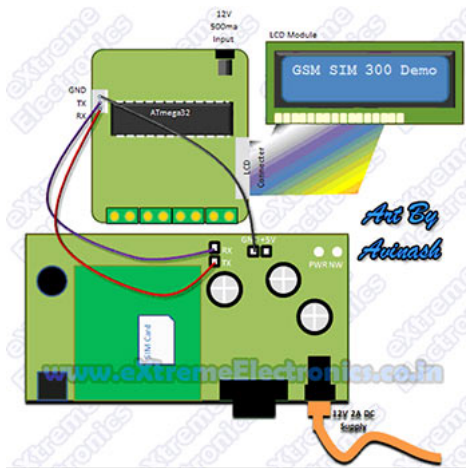
Для демонстрации связи с SIM300 используя AVR ATmega32, нам понадобятся следующие компоненты:

- ATmega32 с обвязкой – регистром сброса, ISP штырьками, кварцем 16 МГц.
- источник +5В для питания ATmega32 и ЖК-дисплея.
- символьный ЖК-дисплей 16x2 для индикации результатов.
- модуль SIM300.



Мы использовали отладочную плату [Xboard](#), поскольку она имеет ATmega32 с обвязкой, источник +5В и ЖК-дисплей.





Демонстрационный исходный код для AVR и SIM300

Демонстрационный исходный код написан на языке C и скомпилирован с использованием бесплатного AVR-GCC компилятора, используя последнюю [Atmel Studio 6 IDE](#). Проект разделен на следующие модули:

- » Библиотека ЖК-дисплея
 - Файлы lcd.c, lcd.h, myutils.h, custom_char.h
 - Её работа заключается в контроле стандартного ЖК-дисплея 16x2.
 - Более подробную информацию можно найти по [ссылке](#).
- » Библиотека USART
 - Файлы usart.c, usart.h
 - Её работа заключается в контроле аппаратного USART микроконтроллера AVR. Включает в себя функции инициализации USART, отправки/приема символов, отправки/приема строк.
- » Библиотека SIM300
 - Файлы sim300.c, sim300.h

Пошаговая настройка проекта AS6

Создайте новый проект AS6 под названием "Sim300Demo".

Используя solution explorer (дерево проектов) создайте папку с именем "lib" в текущей папке.

Внутри папки "lib" создайте папки "LCD", "USART" и "SIM300".

Скопируйте файлы (с помощью проводника Windows) lcd.c, lcd.h, myutils.h, custom_char.h в папку lcd.

Скопируйте файлы (с помощью проводника Windows) usart.c, usart.h в папку USART

Скопируйте файлы (с помощью проводника Windows) sim300.c, sim300.h в папку SIM300.

Добавьте файлы lcd.c, lcd.h, myutils.h, custom_char.h в проект с помощью solution explorer (дерева проектов).

Добавьте filesusart.c, usart.h в проект с помощью solution explorer (дерева проектов).

Добавить файлы sim300.c, sim300.h в проект с помощью solution explorer (дерева проектов).

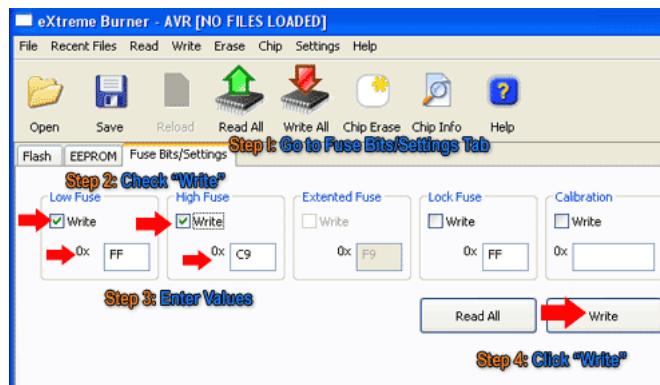
Определите значение F_CPU = 16000000 использования AS6.

Скопируйте и вставьте основной файл Sim300Demo.c в программу.

Скомпилируйте проект чтобы получить hex файл.

Прошейте Xboard с помощью USB программатора.

Если вы используете новый микроконтроллер ATmega32, установите LOW FUSE на 0xFF и HIGH FUSE на 0xC9.



Что делает демонстрационная программа?

Инициализирует ЖК-дисплей и модуль SIM300.

Проверяет, что модуль SIM300 подключен к USART и реагирует должным образом.

Отображает IMEI SIM300 модуля.

Отображает ID производителя

Проверяет наличие SIM-карты.

Ищет GSM сеть и устанавливает соединение. Для этого должна быть активная SIM-карта.

Показывает название оператора, например MTS или Megafon.





Возможные проблемы



Нет изображения на ЖК-дисплее

Убедитесь, что в проекте AVR Studio установлена тактовая частота 16 МГц (16000000Hz)
Отрегулируйте контрастность потенциометром.
Нажмите кнопку сброса несколько раз.
Включите/выключите устройство несколько раз.
Подключайте ЖК-дисплей только так, как показано на схеме.

Во время инициализации SIM300 появляется ошибка "No Response (Нет ответа)"

Проверьте целость Rx, Tx и GND линий между SIM300 и Xboard.
Убедитесь, что микроконтроллер работает на частоте 16 МГц.
Установите фьюзы точно, как описано выше.

Ошибки компилятора

Многие люди используют уже написанные и скомпилированные программы. У них отсутствует опыт и они не знакомы с основами программирования и компиляции. Ознакомится с компиляторами и их работой на разных платформах (PC / MAC / Linux) будет отличным началом. Встроенные системы не подходят для изучения основ. Они предназначены тем, кто имеет эти навыки и просто пользуется ими.
Убедитесь, что все файлы библиотеки ЖК-дисплея добавлены к проекту.
Убедитесь, что AVR-GCC установлен. (Дистрибутив Windows называется WinAVR)
Убедитесь, что в проекте AVR Studio указан AVR GCC.

Общие советы для новичков

Используйте готовые отладочные платы и программаторы.
Попробуйте изучать [AVR с самого начала](#), основываясь на статьях и учебниках.

Список радиоэлементов

Обозначение	Тип	Номинал	Количество	Примечание	Магазин
U1	МК AVR 8-бит	ATmega32	1		Поиск в LCSC
U2	Линейный регулятор	LM7805	1		Поиск в LCSC
D1	Выпрямительный диод	1N4007	1		Поиск в LCSC
D2	Светодиод		1		Поиск в LCSC
C1, C2	Конденсатор	22 пФ	2		Поиск в LCSC
C3, C4, C6	Конденсатор	0.1 мкФ	3		Поиск в LCSC
C5	Электролитический конденсатор	470 мкФ	1		Поиск в LCSC
RV1	Подстроечный резистор	10 кОм	1		Поиск в LCSC
R1	Резистор	4.7 кОм	1		Поиск в LCSC
R2	Резистор	330 Ом	1		Поиск в LCSC
X1	Кварцевый резонатор	16 МГц	1		Поиск в LCSC
X2	Разъем для питания		1		Поиск в LCSC
X3	Модуль	SIM300	1		Поиск в LCSC
LCD1	LCD-дисплей	LM016L	1		Поиск в LCSC
L1	Катушка индуктивности	10 мкГн	1		Поиск в LCSC
SW1	Выключатель		1		Поиск в LCSC

[Скачать список элементов \(PDF\)](#)

[Оригинал статьи](#)

Прикрепленные файлы:

• [SIM300_AVR.rar](#) (977 Кб)

Теги: GSM GPRS Микроконтроллер AVR Atmel Studio Перевод

Заказать разработку печатной платы к данной статье

ИБП
Промышленные!
3 - 540 кВт!

Что случилось
сегодня в
Минске?


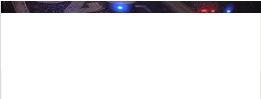

Производство
электроники-
ProSMD

Блок-модули
Containex


Жилые **блоки** любых
размеров. Стальной
каркас. Тепло и звуко-
изоляция.


[Онлайн-запрос](#)
[Аренда мебели](#)
[Дополнительное
оснащение](#)








Яндекс.Директ


 none

 Опубликовано: 2012 г.

 3

 0

 PDF

 Вознаградить


Я собрал

0

0

Отслеживать передвижение своих сотрудников поможет [gps жучок](#).

Комментарии (3) | Я собрал (0) | Подписаться




Дмитрий


08.08.2014 01:48


#

Приборчик может по sms передавать место положения (gps) ?

Ответить

 0






Evgraf


18.12.2015 00:36


#

Может, если к контроллеру подвести gps трекер

Ответить

 0






дмитрий


09.10.2017 09:13

#

Тема ещё жива? Как вы судите по символу конца строки? Если например в ответе от модема таких разделителей может быть 3 штуки. Нашли первый и отправили в кофемолку. И в результате ничего не нашли, потому что нужная инфа например содержится после второго 0x0d в строке

Ответить

 0



Добавить комментарий

Имя:

E-mail: не публикуется




Уведомлять об ответах на мой комментарий

B

i

U

S



Ваш комментарий

Текст:

Защита от спама:

В чем измеряется сила тока?

Ответ на вопрос

Файлы:


Выбрать файлы

Файл не выбран

Для выбора нескольких файлов используйте CTRL



Я согласен с [правилами публикации комментариев](#)

Оставить комментарий




Программатор Pickit3


Купить 18 \$





вверх

1999-2017 СМИ Сайт-ПАЯЛЬНИК 'cxem.net' 6+
Свидетельство о регистрации СМИ Эл № ФС 77 - 59178 от 03.09.2014 выдано Роскомнадзором
При использовании материалов сайта, обязательна ссылка на сайт ПАЯЛЬНИК и первоисточник










Размещение рекламы Вакансии

154172
62319
25005



Участник
Rambler's
TOP 100

http://cxem.net/mc/mc151.php

9/9