

**МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН**

**ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ ИМЕНИ МУХАММАДА АЛ-ХОРАЗМИЙ**

**Кафедра “Аппаратное и программное  
обеспечение систем управления в телекоммуникации”**

**Мирзаева М.Б., Абасханова Х.Ю.**

**Методические пособие по выполнению лабораторных работ по предмету  
“ВСТРОЕННЫЕ СИСТЕМЫ И ИХ ПРОГРАММНОЕ  
ОБЕСПЕЧЕНИЕ”**

**Ташкент 2019**

**Авторы: Мирзаева М.Б., Абасханова Х.Ю.**

**«ВСТРОЕННЫЕ СИСТЕМЫ И ИХ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ»**  
/ТУИТ Ташкент, 2019. 48с.

Рассмотрены 6 лабораторных работ на основе платы Arduino, в которых показано, как сделать так, чтобы электронные устройства могли обмениваться между собой данными и реагировать на команды. Показано, как обмениваться данными с помощью RS-232 интерфейсом, взаимодействующими по сети; как использовать устройства Bluetooth, Ethernet, WI-FI, RFID и GSM модули для беспроводного получения информации от различных датчиков и др. Рассмотрены языки программирования Arduino, Processing и PHP.

Методические пособие рассмотрены на заседании научно - методического совета ТУИТ имени Мухаммада ал-Хоразмий и рекомендованы к печати (протокол № \_\_\_\_\_ от \_\_\_\_\_ г.)

Ташкентский университет информационных технологий имени Мухаммада  
ал-Хоразмий, 2019

## СОДЕРЖАНИЕ

Введение.....		4
Лабораторная работа №1	От Arduino к Arduino с использованием последовательного интерфейса.....	5
Лабораторная работа №2	Проектирование приема и передачи данных при помощи интерфейса Bluetooth во встроенных системах связи.....	10
Лабораторная работа №3	Передача данных по установленным сетям Ethernet.....	17
Лабораторная работа №4	Проектирование приема и передачи данных при помощи интерфейса Wi-Fi модуля ESP8266 во встроенных системах связи.....	28
Лабораторная работа №5	Проектирование приема и передачи данных при помощи RFID на примере RC522 во встроенных системах связи.....	38
Лабораторная работа №6	Подключение GSM модуля SIM800L к Arduino.....	42
	Литература .....	46

## ВВЕДЕНИЕ

Все, к чему прикасаются руки человека на работе или в быту, изготовлено для удобства. На данный момент наука об встроенных системах — огромный объем информации, начиная от теоретических работ на передовом фронте науки и заканчивая сугубо практическими знаниями инженерно-технического плана.

Чтобы понять встроенные системы и использовать их в собственных нуждах, совсем необязательно изучать все премудрости технических постулатов и понимать хитрые коды. В этом заключается работа ученых и инженеров: привести сложную теорию в разряд доступной практики. Ведь никто не изучает теорию двигателей внутреннего сгорания, чтобы сесть за руль или починить мелкие неисправности.

Чтобы разобраться в бытовом применении встроенных систем, необходимо знать достаточно простые правила, сформулированные на страницах этой методички. Именно для этого она и предназначена — помочь студентом, кто, не имея диплома инженера, хочет самостоятельно решать проблемы, возникающие с встроенными системами, проводить практические работы, понимать суть функционирования и устройство встроенных систем. Список необходимых инструментов, область их применения и условия их работы — все это будет рассмотрено как можно подробнее, с пошаговым описанием всех необходимых действий, подкреплено теоретическими знаниями, наглядными фотографиями рисунками.

## От Arduino к Arduino с использованием последовательного интерфейса

**Цель работы:** Изучить принцип работы *serial port* и научиться от Arduino к Arduino с использованием последовательного интерфейса.

### Нужные приборы:

1. Arduino UNO – 2 шт
2. Светодиоды
3. Соединительные провода

### Теоретическая часть

Набор функции **Serial** служит для связи устройства Arduino с компьютером или другими устройствами, поддерживающими последовательный интерфейс обмена данными. Все платы Arduino имеют хотя бы один последовательный порт (UART, иногда называют USART). Для обмена данными **Serial** используют цифровые порты ввода/вывода 0 (RX) и 1 (TX), а также USB порт. Важно учитывать, что если вы используете функции **Serial**, то нельзя одновременно с этим использовать порты 0 и 1 для других целей.

Среда разработки **Arduino** имеет встроенный монитор последовательного интерфейса (**Serial monitor**). Для начала обмена данными необходимо запустить монитор нажатием кнопки **Serial monitor** и выставить ту же скорость связи (*baudrate*), с которой вызвана функция *begin* ().

Плата **Arduino Mega** имеет три дополнительных последовательных порта: **Serial 1** на портах 19 (RX) и 18 (TX), **Serial 2** на портах на портах 17 (RX) и 16 (TX), **Serial 3** на портах на портах 15 (RX) и 14 (TX). Чтобы использовать эти порты для связи с компьютером понадобится дополнительные адаптеры USB – to - serial, т.к. они не подключены к встроенному адаптеру платы Mega. Для связи с внешним устройством через последовательный интерфейс соедините TX порт вашего устройства с RX портом внешнего устройства и RX порт вашего устройства с портом TX внешнего и соедините «землю» на устройствах.

### Важно!

- Не подключайте эти порты напрямую к RS232 порту, это может повредить плату.
- Функции **Serial:**
- **Serial.begin ();** // инициализирует соединение
- **Serial.end ();** // закрывает соединение
- **Serial.available ();** // получает количество байт, доступных для чтения из последовательного интерфейса связи
- **Serial.read();** // считывает очередной доступный байт из буфера последовательного соединения

- **Serial.flush();** // ожидает окончания передачи исходящих данных (для Arduino 1.0+)
- **Serial.print();** // передает данные через последовательный порт как ASCII текст
- **Serial.println();** // передает данные через последовательный соединение как ASCII текст с следующим за ним символом переноса строки
- **Serial.write();** // функция передаст данные как бинарный код через последовательное соединение
- **Serial.peek();** // возвращает следующий доступный байт (символ) из буфера входящего последовательного соединения, не удаляя его из этого буфера

### **Serial.print()**

Передает данные через последовательный порт как ASCII текст. Эта функция может принимать различные типы данных. Так целые числа выводятся соответствующими им символами ASCII. Вещественные выводятся с помощью двух ASCII символов, для целой и дробной части. Байты передаются как символ с соответствующим номером. Символы и строки отсылаются как есть.

Пример:

- **Serial.print(78)** передается как "78"
- **Serial.print(1.23456)** передается как "1.23"
- **Serial.print(byte(78))** передается как "N" (т.к. в таблице ASCII "N" под 78 номером)
- **Serial.print('N')** передается как "N"
- **Serial.print("Hello world.")** передается как "Hello world."

С помощью второго опционально параметра можно задать базис (систему счисления) для чисел. Допустимые значения BYTE, BIN (двоичный), OCT (восьмеричный), DEC (десятеричный), HEX (шестнадцатеричный). Для вещественных (дробных) чисел второй параметр задает количество знаков после запятой.

Пример:

- **Serial.print(78, BYTE)** выводит "N"
- **Serial.print(78, BIN)** выводит "1001110"
- **Serial.print(78, OCT)** выводит "116"
- **Serial.print(78, DEC)** выводит "78"
- **Serial.print(78, HEX)** выводит "4E"
- **Serial.println(1.23456, 0)** выводит "1"
- **Serial.println(1.23456, 2)** выводит "1.23"
- **Serial.println(1.23456, 4)** выводит "1.2346"

## **Синтаксис**

`Serial.print(val)`

`Serial.print(val, format)`

## **Параметры**

- `val`: данные для передачи через последовательное соединение
- `format`: базис для целых чисел или количество знаков после запятой для вещественных

## **ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ**

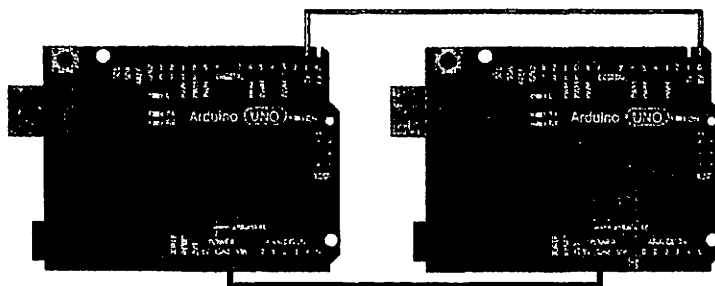
### **От Arduino к Arduino с использованием последовательного интерфейса**

Обмен данными между двумя платами Arduino - очень полезная фишка для многих проектов. Например, можно один Arduino управляет моторами, а второй используется для подключения сенсоров и передачи управляющих сигналов на первый микроконтроллер. Реализовать обмен данными между двумя Arduino можно с использованием с помощью последовательного (`serial`) интерфейса. Для этого будут использоваться контакты RX и TX.

### **Схема подключения**

На рисунке ниже показана схема подключения двух контроллеров Arduino. На схеме показаны две платы Arduino Uno, но можно использовать и Arduino Mega, если использовать соответствующие контакты RX и TX.

Обратите внимание, что необходимо объединить контакты Gnd. В противном случае, обмен данными происходить не будет! При подключении контакт TX подключается к RX, а RX - к TX.



1.1- Рис. Серил порт.

## Особенности программы

При использовании серийного интерфейса, данные передаются в байтах. Эти байты считываются вторым Arduino по одному. Если мы передаем символы, преобразовать их в байты не проблема. Но если мы передаем и символы, и цифровые значения, данные могут интерпретироваться некорректно, так как число и символ могут принимать одинаковые значения в байтах. Кроме того, числа могут не поместиться в один байт... Самос простое решение этой проблемы - не передавать символы и числа одновременно.

### Простой скетч

Можно настроить передачу данных между двумя Arduino с использованием примера Physical Pixel.

Загрузите скетч Physical Pixel, который можно найти в Arduino IDE в папке: File>>Examples>>Communication. на ваш Arduino.

На вторую плату Arduino загрузите следующий скетч:

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.print('H');  
  delay(500);  
  Serial.print('L');  
  delay(500);  
}
```

Когда код начинает работать, светодиод на 13 пине Arduino должен загораться и тухнуть с частотой 0.5 Гц. Для того, чтобы удостовериться в работоспособности скетча, можете изменить значение задержки (delay). Символ 'H' в приведенном выше скетче зажигает светодиод, а символ 'L' отвечает за отключение светодиода. Можно расширить этот список символов для выполнения других действий. Но этот код недостаточно гибкий и для решения более комплексных и сложных задач может не подойти.

### Расширенный скетч

Есть второй вариант скетча для передачи данных от одного Arduino к другому. Для этого все данные с Arduino, который передает информацию, преобразовываются в символы. Вторая плата Arduino считывает поступающие данные тоже в символах. На практике данные, конечно,



передаются в байтах, но наш Arduino отлично справляется с конвертацией символов в байты и обратно. Скетч для Arduino, передающего данные, скетч для микроконтроллера, который передает данные, конвертирует символы в байты и, если необходимо, преобразовывает числовые значения в символы, перед тем как превратить их в байты. Пример скетча приведен ниже.

```
const int ledPin = 13; // the pin that the LED is attached to
byte incomingByte; // a variable to read incoming serial data into
void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}
void loop() {
  if (Serial.available() > 0) {
    incomingByte = Serial.read();
    Serial.print(incomingByte);
    if (incomingByte == 'H') {
      digitalWrite(ledPin, HIGH);
    }
    if (incomingByte == 'L') {
      digitalWrite(ledPin, LOW);
    }
  }
}
```

**Задание:** Передать личные данные (Ф.И.О. и группу) от Arduino к Arduino с использованием последовательного интерфейса.

#### **Вопросы:**

1. Для чего используется Serial порт на Arduino?
2. Перечислите функции Serial порт?
3. Разница между RX и TX?
4. В какой функции Serial передает данные через последовательный соединение как ASCII текст с следующим за ним символом переноса строки?
5. Что используется для целых чисел или количество знаков после запятой для вещественных на платформе ArduinoUno?

## Лабораторная работа №2

### Проектирование приема и передачи данных при помощи интерфейса Bluetooth во встроенных системах связи.

**Цель работы:** Получение знаний по проектированию приема и передачи данных при помощи протокола Bluetooth во встроенных системах, на примере Arduino

#### Нужные приборы:

- Arduino Uno R3
- Набор проводов ПАПА-МАМА
- HC-06 Bluetooth
- Телефон на базе Android (С поддержкой Bluetooth)
- Необходимое программное обеспечение –Bluetooth client

#### Теоретическая часть

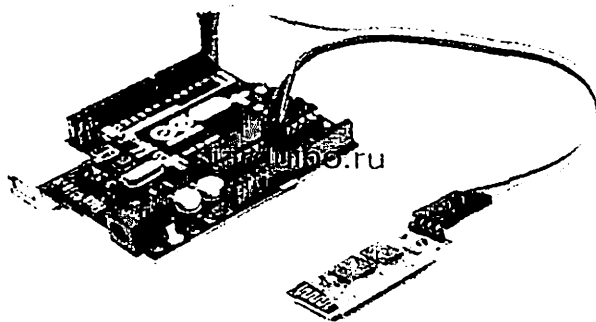
Очень часто в ваших проектах возникает необходимость в дистанционном управлении или передачи данных с ваших телефонных гаджетов.

Один из самых популярных и распространенных методов обмена данными посредством Bluetooth.

Сегодня мы разберем простые примеры как можно подключить Bluetooth модуль к Arduino и настроить дистанционное управление с телефона.

Подключать Bluetooth модуль к микроконтроллеру Arduino удобнее всего с помощью проводов ПАПА-МАМА.

Со стороны управляемого устройства, такого как Arduino, этот модуль выглядит как обычный последовательный интерфейс. Поэтому вы можете отладить всё общение с устройством на компьютере, а потом просто подключить этот bluetooth-модуль, и всё заработает как надо. С HC-06 вы можете управлять роботом прямо со своего смартфона. Поставив на телефон или планшет одну из многочисленных программ для управления через bluetooth, вы можете превратить его в настоящий продвинутый джойстик, и ваш робот сможет послушно поворачивать вслед за поворотом смартфона. Точно так же можно связаться со своим устройством с ноутбука, или с любого другого прибора, который может подключаться к bluetooth-устройствам.

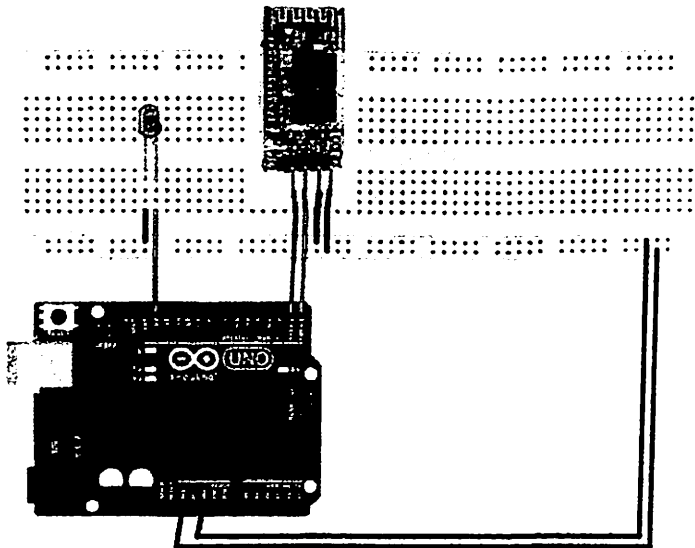


2.1- Рис. Подключения Bluetooth модуль к микроконтроллеру.

<b>Arduino</b>	<b>Bluetooth</b>
Pin 1 (TX)	RXD
Pin 0 (RX)	TXD
GND	GND
5V	VCC

Будьте внимательны, подключать нужно *TX -> RXD*, *RX -> TXD*. Теперь необходимо записать пробный код программы:

Во время загрузки скетча необходимо, что бы Bluetooth модуль был отключен от микроконтроллера Arduino. В противном случае скетч не запишется, потому что связь с Bluetooth модулем происходит по одному и тому же порту RX и TX, что и USB.



2.2- Рис. Схема подключения.

```

int ledPin = 2;
int ledPin2 = 3;
const int step = 1;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  digitalWrite(ledPin, LOW);
}

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
  digitalWrite(ledPin2, HIGH);
  delay(1000);
  digitalWrite(ledPin2, LOW);
  delay(1000);
}

```

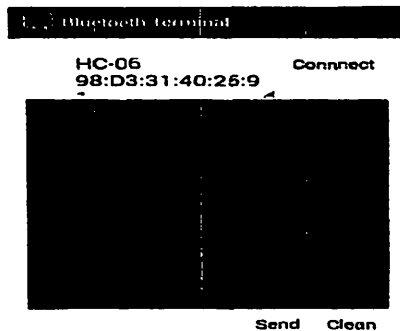
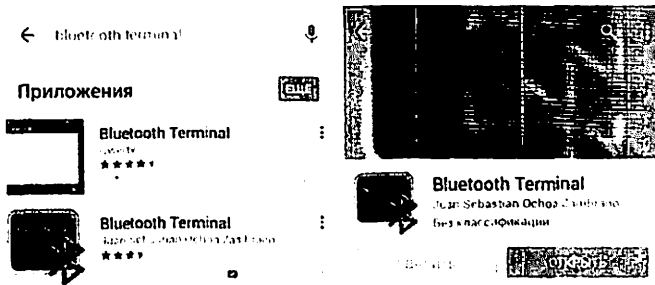
После того как скетч записан и Bluetooth модуль подключен к Arduino, можно перейти к следующему шагу.

### Подключение Bluetooth к телефону:

Желательно в качестве источника питания для Arduino использовать не USB, а внешний блок питания на 9 В.

1. Включаем Bluetooth на телефоне и ищем новые устройства
2. Находим в списке расстройств "HC-06" и подключаемся к нему.
3. Телефон спросит пин-код, необходимо ввести "1234" или "0000"
4. Ура. Устройство подключено.

Теперь нужно скачать bluetooth terminal на ваш телефон. Мы рассмотрим на примере платформы Android.



Вы можете установить разные bluetooth терминалы, как правило они отличаются только разными дизайнами, функционал от этого не меняется. Так же можно найти и терминал и для продуктов ios. После того как мы

установили терминал, запускаем его выбираем наш bluetooth модуль HC-06 и подключаемся к нему.

#### **Описание:**

Bluetooth модуль для подключения устройств к Arduino по Bluetooth. Режим работы пассивный.

Для работы просто выбираем поиск на другом устройстве, и находим его как HC-06. Пин базовый 1234. После подключения к Arduino, данный Bluetooth выступает в роли порта, т.е. все что мы пошлем на последовательный порт будет отсылался а этот Bluetooth, также все то что мы будем слать на этот Bluetooth из вне, будет приниматься на Arduino как через порт.

#### **Пины:**

STATE - выводит сигнал от светодиода.

RXD - контакт для получения данных. На него мы посылаем данные.

TXD - контакт для отправки данных.

GND - земля

VCC - питание 3.3 v

EN - питание вкл/выкл, если подать единицу модуль будет включен. Если ноль, то модуль будет отключен.

#### **Настройка:**

Для изменения базовых настроек, имя, пин. Модуль нужно подключить к ПК, через переходник. И через ПК, AT командами изменять настройки. На данный момент я этого не делал, т.к. не имсю переходника.

#### **Но вот список команд для изменения:**

AT - возвращает ОК

AT+VERSION - посмотреть версию

AT+NAMEууу - изменить имя устройства.

AT+PINxxxx - задать другой пин, базовый 1234

AT+BAUDx - установить скорость.

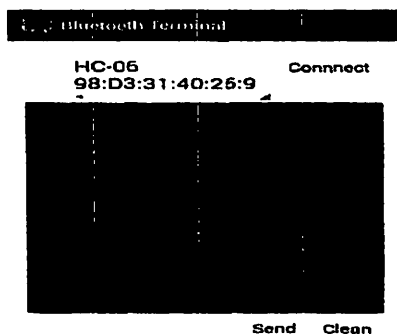
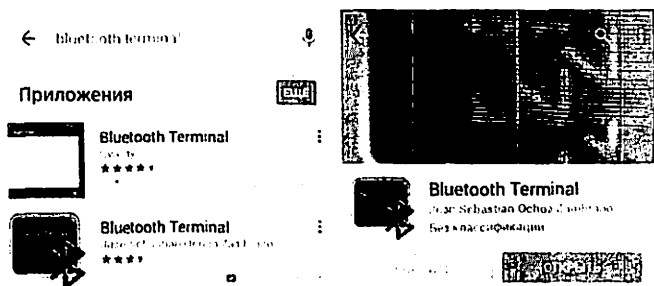
После того как скетч записан и Bluetooth модуль подключен к Arduino, можно перейти к следующему шагу.

### Подключение Bluetooth к телефону:

Желательно в качестве источника питания для Arduino использовать не USB, а внешний блок питания на 9 В.

1. Включаем Bluetooth на телефоне и ищем новые устройства
2. Находим в списке расстройств "HC-06" и подключаемся к нему.
3. Телефон спросит пин-код, необходимо ввести "1234" или "0000"
4. Ура. Устройство подключено.

Теперь нужно скачать bluetooth terminal на ваш телефон. Мы рассмотрим на примере платформы Android.



Вы можете установить разные bluetooth терминалы, как правило они отличаются только разными дизайнами, функционал от этого не меняется. Так же можно найти и терминал и для продуктов ios. После того как мы

установили терминал, запускаем его выбираем наш bluetooth модуль HC-06 и подключаемся к нему.

#### **Описание:**

Bluetooth модуль для подключения устройств к Arduino по Bluetooth. Режим работы пассивный.

Для работы просто выбираем поиск на другом устройстве, и находим его как HC-06. Пин базовый 1234. После подключения к Arduino, данный Bluetooth выступает в роли порта, т.е. все что мы пошлем на последовательный порт будет отсылался а этот Bluetooth, также все то что мы будем слать на этот Bluetooth из вне. будет приниматься на Arduino как через порт.

#### **Пины:**

STATE - выводит сигнал от светодиода.

RXD - контакт для получения данных. На него мы посылаем данные.

TXD - контакт для отправки данных.

GND - земля

VCC - питание 3.3 v

EN - питание вкл/выкл, если подать единицу модуль будет включен. Если ноль, то модуль будет отключен.

#### **Настройка:**

Для изменения базовых настроек, имя, пин. Модуль нужно подключить к ПК. через переходник. И через ПК, AT командами изменять настройки. На данный момент я этого не делал, т.к. не имею переходника.

#### **Но вот список команд для изменения:**

AT - возвращает OK

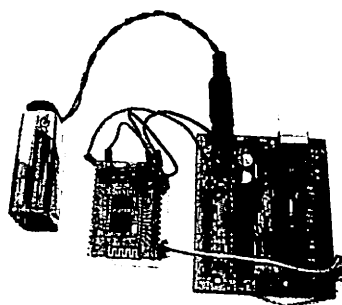
AT+VERSION - посмотреть версию

AT+NAMEууу - изменить имя устройства.

AT+PINxxxx - задать другой пин, базовый 1234

AT+BAUDx - установить скорость.





2.3- Рис. Источника питания для Arduino использовать не USB

В данном примере, по команде с телефона мы отправляем 1 и светодиод на 13 порту включается, 0 его отключает, а \* заставляет его мигать с интервалом в 1 секунду.

```
char val;
int LED = 13;
void setup() {
  Serial.begin(9600);
  pinMode(LED,OUTPUT); }
void loop() {
  if (Serial.available())
  {
  val = Serial.read();
  Serial.print(val);
  if (val=='1')
  {
  digitalWrite(LED, HIGH);
  }
  else if (val=='0')
  {
  digitalWrite(LED, LOW);
  }
  }
}
```

**Задание:**

Включить 3 светодиода. При включение каждого светодиода надо дат команду Ф.И.О.

- 1 светодиод- Фамилия
- 2 светодиод – Имя
- 3 светодиод - Отчество

**Вопросы:**

1. Нарисуйте и объясните схему подключения Bluetooth модуля к Arduino.
2. Какова распиновка HC-06 Bluetooth?
3. Каковы плюсы и минусы использования Bluetooth протокола?
4. Где и в каких масштабах можно встретить встроенные Bluetooth модули?
5. Какие необходимое ПО вы знаете для работы с Bluetooth на Arduino?

## Лабораторная работа №3

### Передача данных по установленным сетям Ethernet

**Цель работы:** Изучение модули Ethernet, принцип ее работы и применение Ethernet в разных отраслях связи.

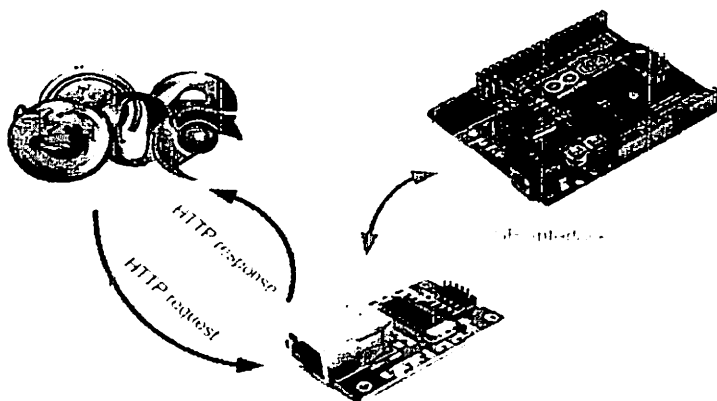
#### Нужные приборы

- Ethernet модуль
- Arduino Плата
- Кабели
- Светодиоды
- Потенциометр
- Резисторы
- Кабель UTP с разъемом 8P8C, используемый в Ethernet-сетях стандартов 10BASE-T, 100BASE-T(x) и 1000BASE-T(x).

#### Теоретическая часть

##### NC28J60 Ethernet модуль для Arduino

Предназначен для сборки устройства управления электрическими приборами через интернет и передачи данных от датчиков для отображения на странице сайта. При этом с сайта можно управлять приборами и проверять выполнение команд. Например, можно изготовить управление поворотом видекамеры с помощью шагового двигателя и с экрана ПК наблюдать за видеосъемкой.



3.1 - Рис. NC28J60 Ethernet модуль для Arduino

Модуль работает в локальной сети TCP/IP и в сети Internet. Используя аппаратные возможности модуля ENC28J60 и программу микроконтроллера создают простой веб-сервер. Модуль связывает МК через интерфейс SPI с сетью TCP/IP. Управление приборами автоматике через internet подключенными к ENC28J60 может производит одновременно или поочередно несколькими операторами с различных персональных компьютеров или мобильных устройств. ENC28J60 Ethernet модуль для Ардуино может служить не только частью прибора автоматике, но и наглядным пособием для изучения сети TCP/IP и методов построения веб-сервера.

Ведущий компонент – микросхема ENC28J60. На плате смонтирована розетка для кабеля сети TCP/IP. Розетка содержит трансформатор, обеспечивающий гальваническую развязку модуля от кабеля TCP/IP и 2 светодиода, свечение которых свидетельствует о обмене данными. Имеется кварцевый резонатор частоты 25 МГц. Вилка подключения к МК имеет 10 контактов.

Индикатор питания. ENC28J60 Ethernet модуль для Ардуино может работать в составе Raspberry Pi или другого микроконтроллерного устройства. Хорошо согласуется с МК STM32 благодаря напряжению питания одного уровня. От МК требуется только подавать пакеты для отправки и забирать принятые через интерфейс SPI.

### **Характеристики:**

Питание напряжение - 3,14 – 3,45 В

ток. максимальный - 250 мА

номинальный - 170 мА

Интегрированный MAC и 10 Base-T PHY

Полный и полудуплекс

Поддержка одного 10 Base-T порта с автоопределением полярности и коррекцией

Авто выравнивание генерации контрольной суммы

Автоотброс ошибочных пакетов

Программируемая функция повтора передачи при ошибке

Программируемые функции Padding и генерирование CRC

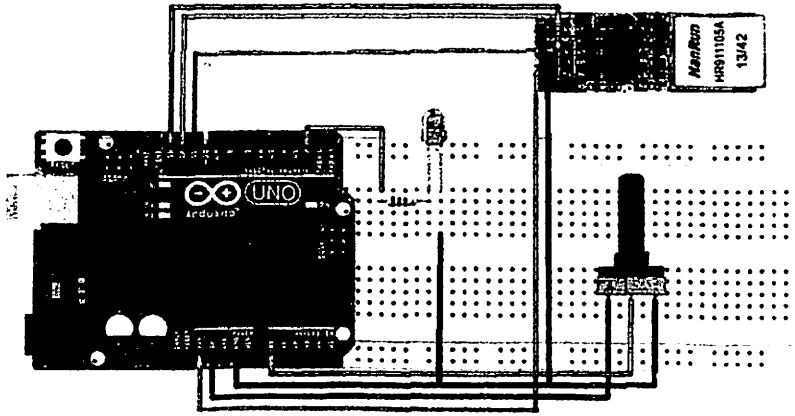
Программируемая функция фильтрации ошибочных пакетов

Тактовая частота SPI до 20 МГц

Размеры платы: 55 x 36 мм

### **Соединения между модулем Ethernet ENC28J60 и Arduino**

модуль ENC28J60	ArduinoUno
VCC	3.3V Pin
GND	GND Pin
SS (CS)	Pin 10
MOSI (SI)	Pin 11
MISO (SO)	Pin 12
SCK	Pin 13



3.2 - Рис. Подключения ENC28J60 модуля с Arduino

Следует отметить, что соединения действительны для Arduino Uno или на основе ATmega328 (Arduino Mini Pro, Arduino Nano и т.д.) в качестве модуля ENC28J60 использует пластины связи SPI, так что если вы работаете с Mega, вы должны подключаться к соответствующему контакту SPI.

**Пример:**

Программа для модуля Ethernet ENC28J60 когда вы программируете этот модуль вам нужно, чтобы загрузить и импортировать наш Arduino IDE библиотека Ethercard.

После импорта библиотеки мы написали следующий эскиз:

```
#include <EtherCard.h>

static byte mymac[] = {0xDD,0xDD,0xDD,0x00,0x01,0x05};
static byte myip[] = {192,168,1,177};
byte Ethernet::buffer[700];

const int ledPin = 2;

char* StatusLed="OFF";

void setup () {

  Serial.begin(9600);
```

```

Serial.write("Laboratory Module ENC28J60");

if(ether.begin(sizeof(ether.net::buffer), mymac, 10))
  Serial.println("Не удалось получить доступ к контроллеру Ethernet");
else
  Serial.println("Ethernet Controller инициализирован");

if(ether.staticSetup(myip))
  Serial.println("Не удалось установить IP - адрес");

Serial.println();

pinMode(ledPin, OUTPUT);
digitalWrite(ledPin, LOW);
}

static void homepage() {
  BufferFiller bfill = ether.tcpOffset();
  bfill.emit_p(PSTR("HTTP/1.0 200 OK\r\n"
    "Content-Type: text/html\r\nPragma: no-cache\r\nRefresh: 5\r\n\r\n"
    "<html><head><title>Naylamp Mechatronics</title></head>"
    "<body>"
    "<div style='text-align:center;'>"
    "<h1>Laboratory Module ENC28J60</h1>"
    "Delay Encoder : $I Milliseconds"
    "<br /><br />STATUS LED: $$<br />"
    "<a href='?status=ON'><input type='button' value='ON'</a>"
    "<a href='?status=OFF'><input type='button' value='OFF'</a>"
    "<br /><br />Encoder: $D (MAX 1024)"
    "<br /><br />"
    "</body></html>"
    ), millis()/1000, StatusLed, analogRead(0));

  return bfill.position();
}

void loop() {

  wordlen = ether.packetReceive();
  wordpos = ether.packetLoop(len);

  if(wordpos) {

```

```

if(strstr((char *)Ethernet::buffer + pos, "GET/?status=ON") != 0) {
    Serial.println("Comando ON recibido");
    digitalWrite(ledPin, HIGH);
    StatusLed = "ON";
}

if(strstr((char *)Ethernet::buffer + pos, "GET/?status=OFF") != 0) {
    Serial.println("Comando OFF recibido");
    digitalWrite(ledPin, LOW);
    StatusLed = "OFF";
}
ether.httpServerReply(homePage());
}
}
}

```

#### Код на Русском

- #include<EtherCard.h>
- 
- статический байт mymac [] = {0xdd, 0xdd, 0xdd, 0x00, 0x01, 0x05};
- статический байт MyIP [] = {192, 168, 1, 177};
- байт Ethernet :: буфер [700];
- 
- const int ledPin = 2;
- 
- char \* EstadoLed = "OFF" ;
- 
- void setup() {
- 
- // Инициализация выводов (9600);
- // Инициализация вывода ( "Test Module ENC28J60" );
- 
- // Инициализация выводов (SizeOf Ethernet :: буфера, mymac, 10)
- // Инициализация нескольких серийных портов ( "Не удалось получить доступ к контроллеру Ethernet" );
- // Инициализация
- // Инициализация ( "Ethernet Controller инициализирован" );
- 
- // Инициализация (MyIP)
- // Инициализация ( "Не удалось установить IP - адрес" );



- 
- Подготовительный `pinMode ()`;
- 
- `pinMode (ledPin, OUTPUT)`;
- `digitalWrite (ledPin, LOW)`;
- }
- 
- `static void setup () {`
- 
- `BufferFillerether.topOffsetbfill = ()`;
- `bfill.emit_p (PSTR ("HTTP / 1.0 200 OK\r\n"`
- `"Content-Type: Текст / html\r\nPragma: безcachernRefresh: 5m\r\n"`
- `"<HTML><HEAD>< название >Nay lampМехатроника</ title></"`
- `голова>"`
- `"< тело >"`
- `"<DIV стиль = \"выравнивания текста: центр;\">"`
- `"<h1> Модуль тестирования ENC28J60 </ h1 >"`
- `" Время пектало: $ L секунды "`
- `"<br /><br /> Государство LED: S $ <br />"`
- `"<A href = \"? Status = ON\"> Тинвходного -кнопка \"\" Значение`
- `= \"ON\"></a>"`
- `"<a href = \"?status=OFF\"><входнойтип =\" кнопка \"значение =\"`
- `OFF \"></a>"`
- `"<br /><br /> Потенциометр: $ D (1024) "`
- `"<br /><br />"`
- `"<a href = \"http://www.naylampmechatronics.com/\">`
- `www.naylampmechatronics.com </a>"`
- `"</ body></ html>"`
- `) . write (0) / 1000, EstadoLed, analogRead (0));`
- 
- `return bfill. emit_p ();`
- }
- 
- `intu протоклетки () {`
- 
- `Слово Len = ether.packetReceive ();`
- `Слово-поз = ether.packetLoop (LEN);`
- 
- `return (поз) {`
-

- если (strstr(( символ \*) Ethernet :: буфер + поз. "GET/? статус = ON" ) != 0) {
- Serial.println ( "ПОЛУЧИЛ по команде" );
- digitalWrite (ledPin, ВЫСОКИЙ);
- EstadoLed = "ON" ;
- }
- 
- 
- если (strstr(( символ \*) Ethernet :: буфер + поз. "GET/? состояние = OFF" ) != 0) {
- Serial.println ( "Command OFF ПОЛУЧИЛ" );
- digitalWrite (ledPin, LOW);
- EstadoLed = "OFF" ;
- }
- ether.httpServerReply (домашней страницы ()): //веб-страница
- //сайт
- }
- }

Теперь объясните основной код:

```
статический байт mac [] = {0xdd, 0xdd, 0xdd, 0x00, 0x01, 0x05};
статический байт MyIP [] = {192, 168, 1, 177};
```

Здесь мы устанавливаем наш Mac и IP, мы можем поставить любое значение, избегая дублирования с другого оборудования, подключенного к нашей сети.

В ничтожной настройке () инициализировать модуль Ethernet и отчет через последовательный порт, здесь также настроить выходной контакт светодиода.

В функции статического слова домашней страницы () мы пишем наш веб - сайт, который должен быть в формате HTML:

```
статическое слово домашней страницы () {
```

```
  BufferFiller ether.tcpOffsetbfill = ();
  bfill.emit_p (PSTR ( "HTTP / 1.0 200 OK\r\n"
    "Content-Type: Text / html\r\nPragma: безeachernRefresh: 5\r\n\r\n"
    "<HTML><HEAD><название>NaylampМехатроника</ title></"
    голова>"
    " <тело> "
```

```

" <DIV стиль = "выравнивания текста: центр:"> "
" <h1> Модуль тестирования ENC28J60 </ h1> "
" Время истекло: $ L секунды "
" <br /><br /> Государство LED: $ $ <br /> "
" <a href="/?status=ON"><входной тип = "кнопка" значение =
"на"></a> "
" <A HREF = "/? состояние = OFF "><входной тип = " значение = "
OFF кнопка "></a>"
" <br /><br /> Потенциометр: $ D (1024)"
" <br /><br /> "
" <a href='http://www.naylampmechatronics.com/'>
www.naylampmechatronics.com </a> "
" </ body></ html> "
), Минус () / 1000, StatusLed, analogRead (0));

```

```

    return bfill, по желанию ();
}

```

Обратите внимание, что для отправки переменных данных должны использовать маркеры, а затем отправить длинный тип данных, строку и десятичнойсо ответственно.

Мы также устанавливаем браузер, чтобы перезагрузить страницу каждые пять секунд это сделано с "Refresh: 5", вам нужны данные - закон показываются постоянно меняется. Время обновления зависит от типа отображаемого данных. В некоторых случаях в этом нет необходимости, или выбрать, чтобы поставить кнопку, чтобы обновить. Если вы не просто хотите, чтобы удалить "Обновить: 5" в функции веб-сайта.

Следующее сравнение, чтобы увидеть, если данные уже были отправлены из браузера с помощью метода GET.

```

    если (strstr( сервер + * ) Ethernet :: буфер + поз, "GET/? status = ON" )! = 0)
    {
        Серийный .Println( "ПОЛУЧИЛ по команде" );
        digitalWrite (ledPin, ВЫСОКИЙ );
        StatusLed= "ON" ;
    }

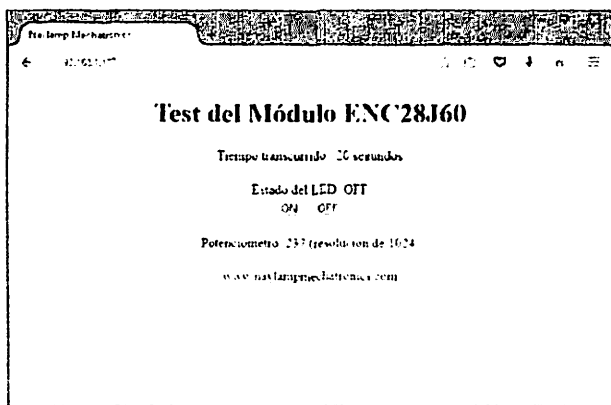
```

В этом случае мы сравниваем ли он получил данные с GET? Status = ON метод следующим образом, это происходит, если кнопка ON нажата в браузере (см кнопка HTML код в предыдущей функции)

После того, как написано скит мы загрузили наш Arduino.

Просто не хватает подключения модуля ENC28J60 Ethernet с еще кабельный модем, маршрутизатор, коммутатор или ПК.

Теперь из веб - браузера нашего ноутбука, телефона или планшета, мы получаем доступ к нашему Arduino набрав IP (для нашего примера 192.168.1.177) в браузере.



### Задание:

№	Порты для Encoder	Порты для LED	Количество Светодиодов
1	A1	2	1
2	A2, A4	3,8	2
3	A3,A5,A1,A2,A0,A4	2-8	6
4	A1,A3,A4,A5	2,4,6,7	4
5	A2,A0,A3	6,7,8	3
6	A4, A0	2,9	2
7	A0	2	1
8	A1-A4	3-7	4
9	A0, A5	8,9	2
10	A5,A4,A2,A1,A0,A3	3-9	6
11	A0-A3	4-8	4
12	A0	2	1
13	A5,A3,A0,A2	2-6	4
14	A3,A4,A5	7,8,9	3
15	A0	2	1
16	A4,A3	5,7	2
17	A5,A0,A1	4,5,6	3

18	A2,A3,A4,A5	1,5,6,8	4
19	A1,A3	8,9	2
20	A5	5	1

**Вопросы:**

1. Что такое Ethernet?
2. Принцип работы Ethernet?
3. Характеристика Ethernet модуля?
4. Схема подключения Arduino и Ethernet?
5. Сфера применения Ethernet?

## Лабораторная работа №4

### Проектирование приема и передачи данных при помощи интерфейса WI-FI модуля ESP8266 во встроенных системах связи.

**Цель работы:** Получение знаний по проектированию приема и передачи данных при помощи протокола Wi-Fi во встроенных системах, на примере Arduino.

#### Нужные приборы:

- 1 плата Arduino Pro Mini или ее аналог
- Набор проводов ПАПА-МАМА
- 1 модуль ESP8266.
- 1 цифровой датчик температуры DS18B20.
- 1 резистор 4,7 кОм.
- 1 источник питания (от 3.3 В до 12 В. В данном примере использовалось напряжение 9В).

#### Теоретическая часть

Модуль на чипсете ESP8266 — это простой и дешёвый способ добавить в своё устройство функции беспроводной связи через Wi-Fi.

Используйте ESP8266, чтобы управлять своим устройством дистанционно или чтобы снимать показания с сенсоров через интернет. Подключите свой гаджет к социальным сетям или реагируйте на данные, которые получаете через API от веб-сервисов.

В семействе модулей ESP8266 есть много разновидностей. Представленный модуль — ESP-01 V090. У него антенна встроена на плату, а на ножки дополнительно выведены 2 GPIO-порта свободного назначения.

#### Взаимодействие:

Управляющее устройство общается с ESP8266 через UART (Serial-порт) с помощью набора AT-команд. Поэтому работа с модулем тривиальна для любой платы с UART-интерфейсом: используйте Arduino, RaspberryPi, что душе угодно.

Работа над приёмом и передачей данных выглядит, как взаимодействие с сырым TCP-сокетом или с serial-портом компьютера.

Более того, модуль можно перепрошивать. Программировать и загружать прошивки можно через Arduino IDE, точно так же, как при работе с Arduino. Реакция на AT-команды — это просто функция штатной прошивки, устанавливаемой на заводе. А вы можете написать свою собственную, если того требует проект. Поскольку на модуле есть 2 порта ввода-вывода общего назначения, вы можете обойтись вовсе без управляющей платы: просто подключите периферию непосредственно к ним.

Для того, чтобы среда Arduino IDE научилась прошивать ESP8266 достаточно добавить директорию с конфигурацией платформы в папку со своими скетчами.

Для физического соединения при прошивке вам понадобится USB-Serial адаптер или плата Arduino/Iskra, настроенная в режим USB-моста.

#### Питание:

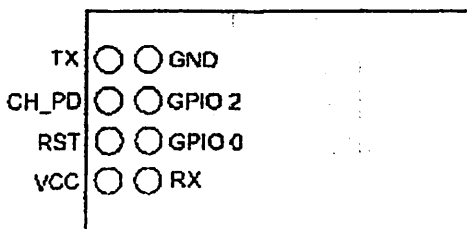
Родное напряжение модуля — 3,3 вольта. Его пины не толерантны к 5 вольтам. Если вы подадите напряжение выше, чем 3,3 вольта на пин питания, коммуникации или ввода-вывода, модуль выйдет из строя.

Поэтому для передачи данных на модуль с 5-вольтовых управляющих плат используйте делитель напряжения, чтобы перевести напряжение в допустимый диапазон. Делитель из двух резисторов одинакового номинала (например, 10 кОм) подойдёт.

Никаких посредников для приёма данных не нужно. Сигнал в 3,3 В как есть будет воспринят управляющей платой, как логическая единица.

Питайте модуль ровными 3,3 вольтами. Их можно получить с отдельного регулятора напряжения или с пина 3.3V на некоторых платах Arduino.

Модуль потребляет в пике 220 мА. Регулятора напряжения, используемого на пятивольтовых платах Arduino для пина 3.3V, может оказаться недостаточно. Обратите внимание на характеристики своей платы. Например, Arduino Uno и Arduino Leonardo могут выдать не более 50 мА с пина 3.3V, поэтому с ними нужно обязательно использовать внешний регулятор; а IskraNeo может выдать до 800 мА, поэтому с ней можно питать ESP8266 прямо от платы.



4.1 - Рис. Распиновка ESP8266 модуля.

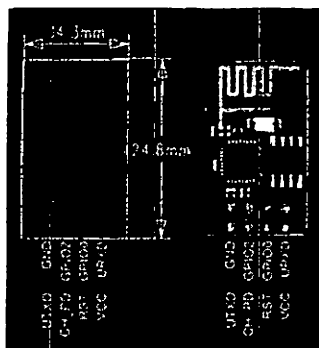
Из-за расположения ножек вплотную в 2 ряда, модуль нельзя установить на breadboard'e. Используйте макетную плату под пайку или провода с разъёмами «мама» для подключения к пинам модуля.

## Характеристики

Основные характеристики Wi-Fi модуля ESP8266:

- поддержка WiFi протоколов 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- встроенный стек TCP/IP
- встроенный TR переключатель, balun, LNA, усилитель мощности и соответствие сети
- встроенный PLL, регуляторы, и система управления питанием
- выходная мощность +20.5 дБм в режиме 802.11b
- поддержка диверсити антенн
- ток утечки в выключенном состоянии до 10 мкА
- SDIO 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4µs guard interval
- пробуждение и отправка пакетов за время до 22 мс
- потребление в режиме Standby до 1.0 мВт (DTIM3)

Контакты (пины) Wi-Fi модуля ESP8266:



4.2- Рис. Распиновка Wi-Fi модуля ESP-01.

Vcc — питание, +3,3 В (максимум 3,6В)

GND — общий

TXD — передача данных (уровень 3,3В)

RXD — приём данных (уровень 3,3В)

CH\_PD — выключение модуля (низкий уровень активный, для включения модуля следует подать Vcc)

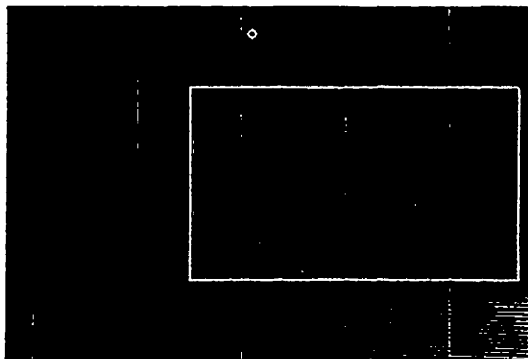
GPIO0 — 0 вывод общего назначения

GPIO2 — 2 вывод общего назначения



RST — сброс модуля (низкий уровень активный)

### Пример использования:



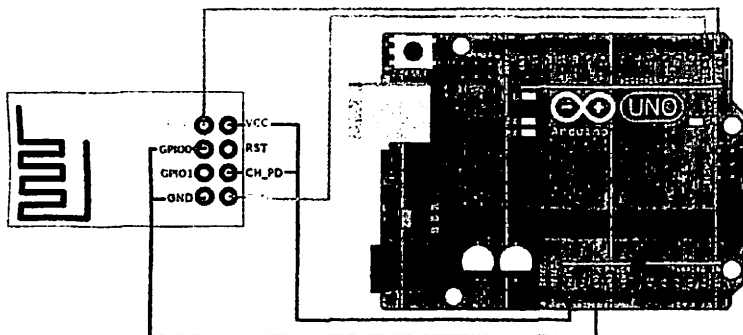
В этой части возникают основные проблемы. Обратите внимание, что существует две версии модуля ESP8266. На первой версии встроены сигнальный светодиоды, которые расположены рядом с пинами. На второй (более новой) модели светодиоды находятся рядом с антенной. В этом проекте мы будем использовать более новую версию модуля ESP8266.

Советую сразу загрузить Firmware V0.922, с помощью которого можно устанавливать скорость передачи данных на 9600.

Для проверки соединения можно использовать USB-TTL кабель и терминал CoolTerm. Команда для изменения скорости передачи данных: AT+STOBAUD=9600

Ниже приведены пины, которые использовались на ESP 8266 для подключения USB-TTL. Для питания ESP8266 использовался пин 3.3V VCC на Arduino. Обратите внимание, что максимальная сила тока на пине 3.3V VCC составляет 150 мА, максимальный ток на ESP8266 достигает 240 мА. Но, за неимением другого источника тока, можно использовать и этот вариант. Обычно модуль ESP8266 использует 70мА.

Не забудьте подключить GPIO0 к GND при загрузке нового firmware. После использования, отключите этот коннектор для нормальной работы.



4.3- Рис. Схема подключения ESP 8266 модуля к Arduino

**Настройка Arduino и скетч подключение ESP8266 к Arduino:**

UTXD --> RX (USB-TTL)  
 CH\_PD <--> VCC  
 RST  
 VCC --> VCC (источник питания):

---

GND --> GND (источник питания)  
 GPIO2  
 GPIO0  
 URXD --> TX (USB-TTL)

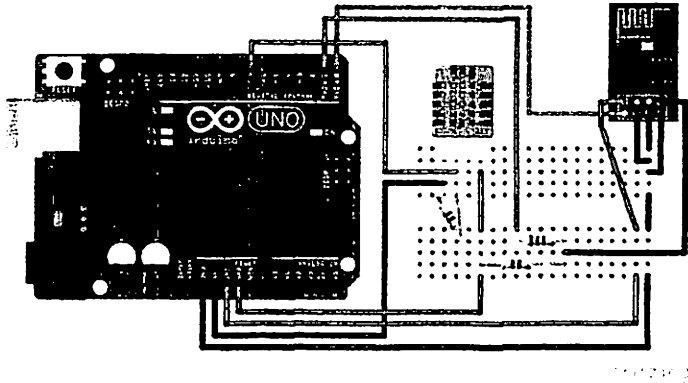
---

UTXD --> RX на Arduino  
 CH\_PD <--> VCC  
 RST  
 VCC --> VCC на Arduino

---

GND --> GND на Arduino  
 GPIO2  
 GPIO0  
 URXD --> TX на Arduino

---



4.4 - Рис. Подключение цифрового датчика температуры к Arduino.

---

Arduino GND --> DS18B20 GND(1) --> DS18B20 VDD(3)  
 DS18B20 DQ(2) --> 4.7кОм R --> VCC на Arduino 3.3В

---

```

#include
#include
#include
#include
#define ONE_WIRE_BUS 8
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
#define SSID "[YOUR_SSID]"
#define PASS "[YOUR_PASSWORD]"
#define IP "184.106.153.149" // thingspeak.com
String GET = "GET
/update?key=[THINGSPEAK_KEY]&field1=";
SoftwareSerial monitor(10, 11); // RX, TX
void setup()
{
  monitor.begin(9600);
  Serial.begin(9600);
  sensors.begin();
  sendDebug("AT");

```

```

delay(5000);
if(Serial.find("OK")){
monitor.println("RECEIVED: OK");
connectWiFi();
}
}
void loop(){
sensors.requestTemperatures();
float tempC = sensors.getTempCByIndex(0);
tempC = DallasTemperature::toFahrenheit(tempC);
char buffer[10];
String tempF = dtostrf(tempC, 4, 1, buffer);
updateTemp(tempF);
delay(60000);
}
void updateTemp(String tempF){
String cmd = "AT+CIPSTART="TCP\,";
cmd += IP;
cmd += "\.80";
sendDebug(cmd);
delay(2000);
if(Serial.find("Error")){
monitor.print("RECEIVED: Error");
return;
}
cmd = GET;
cmd += tempF;
cmd += "\r\n";
Serial.print("AT+CIPSEND=");
Serial.println(cmd.length());
if(Serial.find(">")){
monitor.print(">");
monitor.print(cmd);
Serial.print(cmd);
}else{
sendDebug("AT+CIPCLOSE");
}
if(Serial.find("OK")){
monitor.println("RECEIVED: OK");
}else{
monitor.println("RECEIVED: Error");
}
}
}

```

```

void sendDebug(String cmd){
monitor.print("SEND: ");
monitor.println(cmd);
Serial.println(cmd);
}
booleanconnectWiFi(){
Serial.println("AT+CWMODE=1");
delay(2000);
String cmd="AT+CWJAP=\"";
cmd+=SSID;
cmd+="\". \"";
cmd+=PASS;
cmd+="\"";
sendDebug(cmd);
delay(5000);
if(Serial.find("OK")){
monitor.println("RECEIVED: OK");
return true;
}else{
monitor.println("RECEIVED: Error");
return false;
}
}
}

```

**Совет! После того как вы провели testing/monitoring, загрузите тот же скетч без серийного монитора. Обработываетлучше.**

```

#include
#include
#include
#define ONE_WIRE_BUS 8
OneWireoneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
#define SSID "[YOUR_SSID]"
#define PASS "[YOUR_PASSWORD]"
#define IP "184.106.153.149" // thingspeak.com
String GET = "GET
/update?key={THINGSPEAK_KEY}&field1=";
void setup()
{
Serial.begin(9600);
sensors.begin();
Serial.println("AT");
delay(5000);

```

```

if(Serial.find("OK")){
connect WiFi();
}
}
void loop(){
sensors.requestTemperatures();
float tempC = sensors.getTempCByIndex(0);
tempC = DallasTemperature::toFahrenheit(tempC);
char buffer[10];
String tempF = dtostrf(tempC, 4, 1, buffer);
updateTemp(tempF);
delay(60000);
}
void updateTemp(String tempF){
String cmd = "AT+CIPSTART=\"TCP\",\"";
cmd += IP;
cmd += "\.80";
Serial.println(cmd);
delay(2000);
if(Serial.find("Error")){
return;
}
cmd = GET;
cmd += tempF;
cmd += "\r\n";
Serial.print("AT+CIPSEND=");
Serial.println(cmd.length());
if(Serial.find(">")){
Serial.print(cmd);
}else{
Serial.println("AT+CIPCLOSE");
}
}
booleanconnectWiFi(){
Serial.println("AT+CWMODE=1");
delay(2000);
String cmd="AT+CWJAP=\"";
cmd+="SSID:
cmd+="\".\\"";
cmd+="PASS;
cmd+="\"";
Serial.println(cmd);
delay(5000);

```

```

void sendDebug(String cmd){
  monitor.print("SEND: ");
  monitor.println(cmd);
  Serial.println(cmd);
}
booleanconnectWiFi(){
  Serial.println("AT+CWMODE=1");
  delay(2000);
  String cmd="AT+CWJAP=\"";
  cmd+=SSID;
  cmd+="\".\\\"";
  cmd+=PASS;
  cmd+="\"";
  sendDebug(cmd);
  delay(5000);
  if(Serial.find("OK")){
    monitor.println("RECEIVED: OK");
    return true;
  }else{
    monitor.println("RECEIVED: Error");
    returnfalse;
  }
}
}

```

**Совет! После того как вы провели testing/monitoring, загрузите тот же скетч без серийного монитора. Отрабатываетлучше.**

```

#include
#include
#include
#define ONE_WIRE_BUS 8
OneWireoneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
#define SSID "[YOUR_SSID]"
#define PASS "[YOUR_PASSWORD]"
#define IP "184.106.153.149" // thingspeak.com
String GET = "GET
/update?key=[THINGSPEAK_KEY]&field1=";
void setup()
{
  Serial.begin(9600);
  sensors.begin();
  Serial.println("AT");
  delay(5000);

```

```

if(Serial.find("OK")){
connectWiFi();
}
}
void loop(){
sensors.requestTemperatures();
float tempC = sensors.getTempCByIndex(0);
tempC = DallasTemperature::toFahrenheit(tempC);
char buffer[10];
String tempF = dtostrf(tempC, 4, 1, buffer);
updateTemp(tempF);
delay(60000);
}
void updateTemp(String tenmpF){
String cmd = "AT+CIPSTART="TCP",";
cmd += IP;
cmd += ".80";
Serial.println(cmd);
delay(2000);
if(Serial.find("Error")){
return;
}
cmd = GET;
cmd += tenmpF;
cmd += "\n\n";
Serial.print("AT+CIPSEND=");
Serial.println(cmd.length());
if(Serial.find(">")){
Serial.print(cmd);
}else{
Serial.println("AT+CIPCLOSE");
}
}
booleanconnectWiFi(){
Serial.println("AT+CWMODE=1");
delay(2000);
String cmd="AT+CWJAP=";
cmd+=SSID;
cmd+="\n";
cmd+=PASS;
cmd+="\n";
Serial.println(cmd);
delay(5000);

```









```

if{Serial.find("OK"){
return true;
} else{
return false;
}
}

```

#### Задание:

1. Каждый должен написать своё Ф.И.О.
2. 1-7 варианты создают 3 
3. 8-14 варианты создают 3 
4. 15-22 варианты создают 2  и 1 
5. 23-30 варианты создают 1  и 2 

#### Вопросы:

1. Нарисуйте и объясните схему подключения Wi-fi модуля к Arduino.
2. Какова распиновка Wi-fi?
3. Каковы плюсы и минусы использования Bluetooth протокола?
4. Где и в каких масштабах можно встретить встроенные Wi-fi модули?
5. Какое необходимос ПО вы знаете для работы с Wi-fi на Arduino?

## Лабораторная работа №5

### Проектирование приема и передачи данных при помощи RFID на примере RC522 во встроенных системах связи.

**Цель работы:** Изучить RFID модуль, научиться подключать его к Arduino.

#### Нужные приборы:

1. Контроллер Arduino UNO R3 + USB кабель;
2. Считыватель RFID RC522;
3. Беспроводная RFID метка;
4. Среда ArduinoIDE;
5. Соединительные провода.

#### Теоритическая часть

Радиочастотная идентификация (RFID) – это технология автоматической бесконтактной идентификации объектов при помощи радиочастотного канала связи. Базовая система RFID состоит из:

- радиочастотной метки;
- считывателя информации (ридера);
- компьютера для обработки информации.

Идентификация объектов производится по уникальному цифровому коду, который считывается из памяти электронной метки, прикрепляемой к объекту идентификации. Считыватель содержит в своем составе передатчик и антенну, посредством которых излучается электромагнитное поле определенной частоты. Попавшие в зону действия считывающего поля радиочастотные метки «отвечают» собственным сигналом, содержащим информацию (идентификационный номер товара, пользовательские данные и т. д.). Сигнал улавливается антенной считывателя, информация расшифровывается и передается в компьютер для обработки. Подавляющее большинство современных систем контроля доступа (СКД) использует в качестве средств доступа идентификаторы, работающие на частоте 125 кГц. Это проксимити - карты доступа (только чтение), самыми распространенными являются карты EM-Marin, а также HID, Indala. Карты этого стандарта являются удобным средством открывания дверей и турникетов. Но не более. Эти карты не обладают никакой защищенностью, легко копируются и подделываются и, соответственно, ничего не дают для защиты объекта от несанкционированного проникновения.

Настоящую защиту от копирования и подделки обеспечивают такие идентификаторы, в чипах которых реализована криптографическая защита. Это бесконтактные смарт-карты, работающие на частоте 13.56 МГц, наиболее распространенными из них являются карты Mifare®.

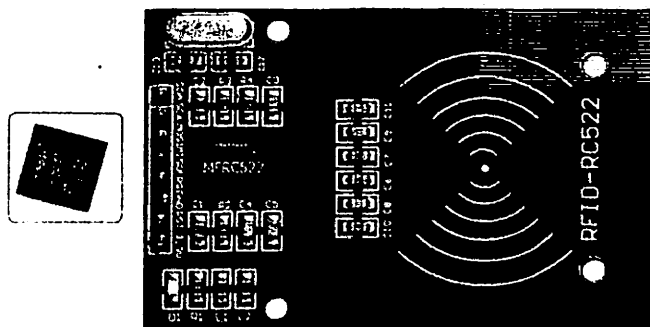
В картах этих стандартов криптозащита организована на высоком уровне, и подделка таких карт практически невозможна. Модуль RFID-RC522 выполнен на микросхеме RC522 фирмы NXP.

Эта микросхема обеспечивает двухстороннюю беспроводную (до 6 см) коммуникацию на частоте 13,56 МГц. RFID - это сокращение от "Radio Frequency IDentification" и переводится как "радиочастотная идентификация".

Микросхема MFRC522 поддерживает следующие интерфейсы подключения:

- SPI (Serial Peripheral Interface, последовательный интерфейс для связи периферийных устройств), обеспечивает скорость передачи данных до 10 Мбит/сек;
- двухпроводной интерфейс I2C, скорость до 3400 кбод в режиме High-speed, до 400 кбод в режиме Fast;
- последовательный UART (аналог RS232), скорость до 1228,8 кбод.

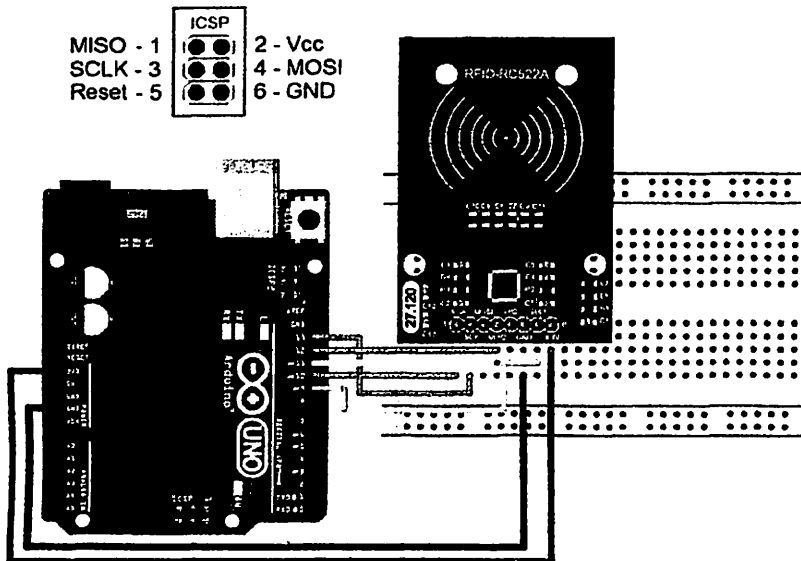
С помощью данного модуля можно записывать и считывать данные с различных RFID-меток: брелков от домофонов, пластиковых карточек-пропусков и билетов на метро и наземный транспорт, а также набирающих популярность NFC-меток.



5.1 - Рис. RFID модуль

#### Основные характеристики:

- основан на микросхеме RC522;
- напряжение питания: 3,3 В;
- потребляемый ток: 13–26 мА;
- рабочая частота: 13,56 МГц;
- дальность считывания: 0~60 мм;
- интерфейс: SPI, максимальная скорость передачи 10 МБит/с;
- размер: 40×60 мм;
- чтение и запись RFID-меток.



5.2- Рис. Схема подключения RFID модуля.

**Контакты для подключения:**

VCC	Питание – 3.3V
RST	Линия сброса
GND	Заземление
MISO	Данные от ведомого к ведущему, SPI
MOSI	Данные от ведущего к ведомому, SPI
SCK	Тактовый сигнал, SPI
SS	Выбор ведомого, SPI
:RQ	Линия прерывания

### Скетч:

```
#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN 7
#define SS_PIN 8
MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup() {
  SPI.begin(9600);
  pinMode(SS_PIN, OUTPUT);
  mfrc522.PCD_WritePCDRegister(0, 0);
  mfrc522.PCD_WritePCDRegister(1, 0);
}

void loop() {
  if (!mfrc522.PCD_ReadPCDRegister(0)) {
    return;
  }
  if (!mfrc522.PCD_ReadPCDRegister(1)) {
    return;
  }
  mfrc522.PCD_WritePCDRegister(0, (&mfrc522.uid));
}
```

### Вопросы:

1. Нарисуйте и объясните схему подключения RFID модуля к Arduino.
2. Какова распиновка RFID?
3. Каковы плюсы и минусы использования RFID?
4. Где и в каких масштабах можно встретить встроенные RFID модули?
5. Какое необходимое ПО вы знаете для работы с RFID на Arduino?

## Лабораторная работа №6

### Подключение GSM модуля SIM800L к Arduino

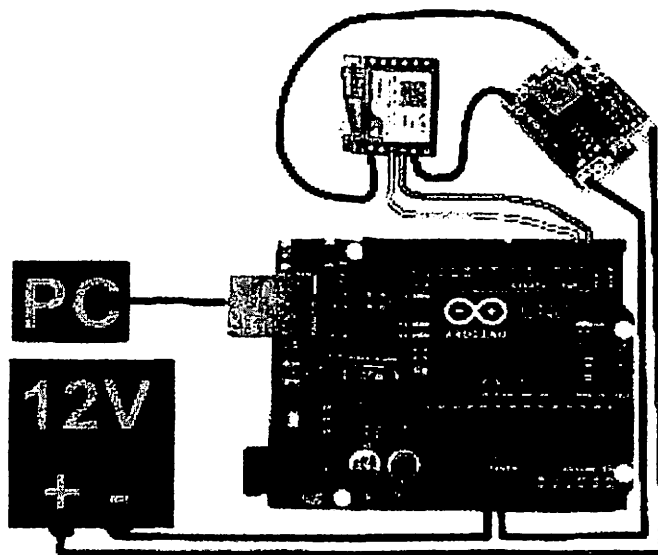
**Цель работы:** Изучить GSM модуль, научиться подключать его к Arduino.

#### Нужные приборы:

- Контроллер Arduino UNO R3 + USB кабель
- GSM модуль SIM800L (питание от 3.7 В до 4.2В)
- Преобразователь напряжения понижающий (DC-DC step-down converter)
- Батарея 12В (или любой источник питания от 6В до 20В)
- Соединительные провода

#### Теоритическая часть

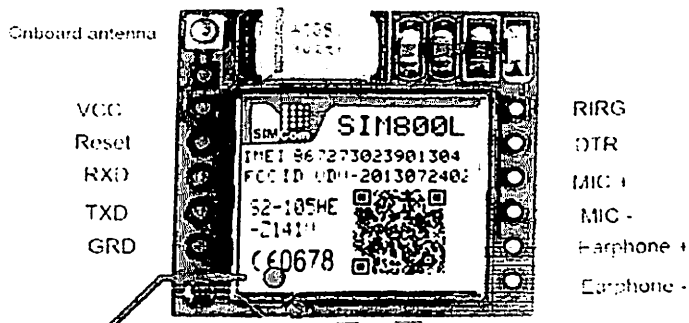
Схема соединения. GSM модуля SIM800L к Arduino



6.1 - Рис. Схема подключения GSM модуля SIM800L.

RF antenna socket

Network signal indicator



6.2- Рис. GSM модуль SIM800L

Даем питание к GSM модулю от батареи 12 В через преобразователь:

- от 12В "минус" идет на ардуино в "GND", от "GND" в преобразователь напряжения во "входящий минус".
- от 12В "плюс" идет в преобразователь напряжения во "входящий плюс".

Стоит отметить, что 5В от Arduino опасны для работы модуля, так как приведут его к поломке, а 3.3 В помогут реагировать на пользовательские команды, но к сети подключение будет отсутствовать.

Перед подключением к преобразователю напряжения необходимо его настроить выставив выходное напряжение на любое в диапазоне 3.7 В - 4.2В. От преобразователя напряжения выходящие контакты подключаем к GSM модулю, соблюдая полярность.

TX и RX контакты на GSM модуле соединяем с 2 и 3 цифровыми контактами на Ардуино. Если необходимо подключить несколько GSM модулей к Arduino, то используйте другие пины и прописывайте в скетче через SoftwareSerial.

#### Необходимые AT-команды

- AT+CMGF=1 — устанавливает текстовый режим смс-сообщения
- AT+IFC=1, 1 — устанавливает программный контроль потоком передачи данных

- AT+CPBS="SM" — открывает доступ к данным телефонной книги SIM-карты
- AT+CNMI=1.2.2.1.0 — включает оповещение о новых сообщениях. новые сообщения приходят в следующем формате: +CMT: "<номер телефона>", "", "<дата, время>", а на следующей строчке с первого символа идёт содержимое сообщения

**Скетч:**

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3); // RX, TX
void setup() {
  Serial.begin(19200); //Скорость порта для связи Arduino с компьютером
  Serial.println("Goodnightmoon!");
  mySerial.begin(19200); //Скорость порта для связи Arduino с GSM модулем
  mySerial.println("AT");
}

void loop() {
  if (mySerial.available())
    Serial.write(mySerial.read());
  if (Serial.available())
    mySerial.write(Serial.read());
}

//Процедура для отправки СМС
void sms(String text, String phone) {
  Serial.println("SMS send started");
  mySerial.println("AT+CMGS=\"" + phone + "\"");
  delay(1000);
  mySerial.print(text);
  delay(300);
  mySerial.print((char)26);
  delay(300);
  Serial.println("SMS send finish");
  delay(3000);
}
```

Процедуру добавлять в конец скетча и вызывать её из основного цикла так: `sms(String("текст СМС на англ."),String("+номер телефона"));`

Только отправлять СМС рекомендую по какому ни будь событию и не просто так в цикле.



### Проверка

Включаем питание затем через монитор порта, выставив скорость 19200 и обязательно опцию "новая строка".

Вводите команду "AT!" и нажмите ENTER. Должна появиться информация от модели модуля.

#### Задание:

№	Порты для LED	Количество Светодиодов	Задержка мсек
1	2	1	200
2	3,8	2	100
3	2-8	6	500
4	2,4,6,7	4	2000
5	6,7,8	3	300
6	2,9	2	450
7	2	1	1000
8	3-7	4	850
9	8,9	2	600
10	3-9	6	520
11	4-8	4	430
12	2	1	1000
13	2-6	4	670
14	7,8,9	3	780
15	2	1	1000
16	5,7	2	1500
17	4,5,6	3	440
18	1,5,6,8	4	250
19	8,9	2	500
20	5	1	200

#### Вопросы:

1. Нарисуйте и объясните схему подключения GSM модуля к Arduino.
2. Объясните распиновку GSM модуля.
3. Сфера применения.

## Список литературы

1. Мирзиёев Ш.М. Танқидий таҳлил, қатъий тартиб-интизом ва шахсий жавобгарлик – ҳар бир раҳбар фаолиятининг қундалик қондаси бўлиши керак. Ўзбекистон Республикаси Вазирлар Маҳкамасининг 2016 йил якунлари ва 2017 йил истиқболларига бағишланган мажлисидаги Ўзбекистон Республикаси Президентининг нутқи. // Халқ сўзи газетаси. 2017 йил 16 январь, № 11.
2. Авторы: Х. Ю. Абасханова, Х.У.Алиев. «ВСТРОЕННЫЕ СИСТЕМЫ»/ ТУИТ. Ташкент, 2015.
3. Ключев А.О., Кустарев П.В., Ковязина Д.Р., Петров Е.В. Программное обеспечение встроенных вычислительных систем. – СПб.: СПбГУ ИТМО, 2009.– 212 с.
4. Платунов А.Е. Постников Н.П. Высокоуровневое проектирование встраиваемых систем. – СПб.: НИУ ИТМО. 2011. – 121 с.
5. 2017й. У.Б.Амирсидов, Х.Ю.Абасханова. Микропроцессорлар. Олий ўқув юртлари учун ўқув қўлланма. Тошкент. 2015. -350 бет .
6. John Catsoulis. Designing Embedded Hardware. O'Reilly, May 2005. Pages: 400

## Дополнительные литература

1. Мирзиёев Ш.М. Эркин ва фаровон, демократик Ўзбекистон давлатини биргаликда барпо этамиз..
2. Мирзиёев Ш.М. Қонун устуворлиги ва инсон манфаатларини таъминлаш – юрт тараққиёти ва халқ фаровонлигининг гарови. 2017.
3. 2017 Гололобов В. - С чего начинаются роботы. О проекте Arduino для школьников (и не только) – Москва, 2011.
4. Ключев, А.О., Ковязина Д.Р., Кустарев, П.В., Платунов, А.Е. Аппаратные и программные средства встраиваемых систем. Учебное пособие [Текст] / А.О. Ключев, П.В. Кустарев, А.Е. Платунов. – СПб.: СПбГУ ИТМО. 2010. – 290 с.
5. C/C++ Embedded System Design Tools. Microsoft Corporation, 2014.

Методические указания предназначены для студентов бакалавриатуры по направлению 5350100 – Телекоммуникационные технологии («Телекоммуникация», «Мобильные системы», «Телерадиовещание»)

Методические указания рассмотрены на заседании кафедры «А и ПОСУ в Т» и рекомендованы для рассмотрения на научно-методическом совете факультета (протокол № 35 от 15.05.2018 г.)

Методические указания рассмотрены на заседании факультета «Телекоммуникационные технологии» и научно - методический совета. (протокол № 10 от 29.05.2018 г.)

Методические указания рассмотрены на заседании научно - методического совета ТУИТ имени Мухаммада ал-Хоразми и рекомендованы к печати (протокол № 11(112) от 24.06.2018 г.)

Авторы:



М.Б.Мирзаева

Рецензенты:

Х.Ю.Абасханова

Ш.Ю.Джаббаров,

Редактор:



А.И.Абдуллаев

С.С.Парсиев

Формат 60x84 1/16. Печ.лист 3.  
Заказ № 232. Тираж 40.  
Отпечатано в "Редакционно издательском"  
отделе при ТУИТ.  
Ташкент ул. Амир Темур, 108