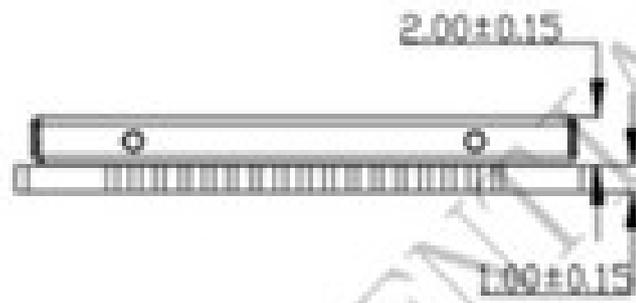
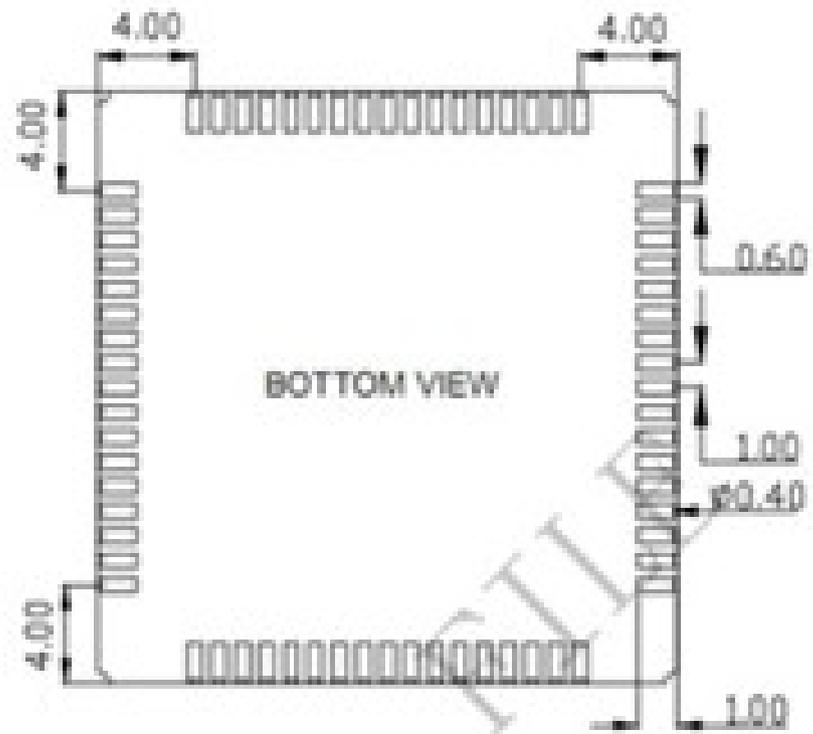
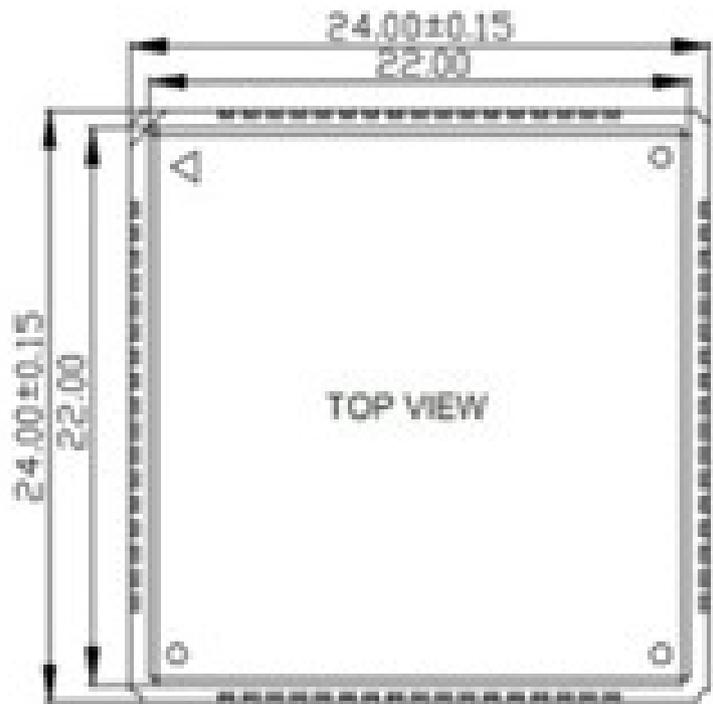


# Лекция

AVR, Arduino

Устройства сопряжения с внешним миром.

(продолжение)



# Sim 900: а зачем мне контроллер?

- Четыре диапазона GSM 850/ 900/ 1800/ 1900 МГц
- - Управление AT командами (GSM 07.07 ,07.05 и фирменные AT команды SIMCom)
- - Аудиокодеки HR, FR, EFR, AMR, подавление эха
- - CSD до 14.4кбит/с
- - PPP-стек
- - Встроенный стек TCP/IP, UDP/IP

- - Протоколы HTTP и FTP
- - Протокол защищенных сокетов SSL
- - Декодирование DTMF-тонов
- - eMail - формирование и отправка электронных писем посредством AT-команд
- - SMS Autorun - исполнение AT-команд, полученных по SMS от определенного абонента
- - 2.5Mb user memory - встроенная память для пользовательских данных
- - MMS - формирование, дополнение пользовательскими файлами и отправка с помощью AT-команд

- - Протоколы HTTP и FTP
- - Протокол защищенных сокетов SSL
- - Декодирование DTMF-тонов
- - eMail - формирование и отправка электронных писем посредством AT-команд
- - SMS Autorun - исполнение AT-команд, полученных по SMS от определенного абонента
- - 2.5Mb user memory - встроенная память для пользовательских данных
- - MMS - формирование, дополнение пользовательскими файлами и отправка с помощью AT-команд

# Технология Embedded AT

Прошивка с поддержкой технологии Embedded AT позволяет записывать в память модуля и исполнять пользовательский код, написанный на языке C. В ряде случаев это позволяет **отказаться** от использования внешнего микроконтроллера!!!!

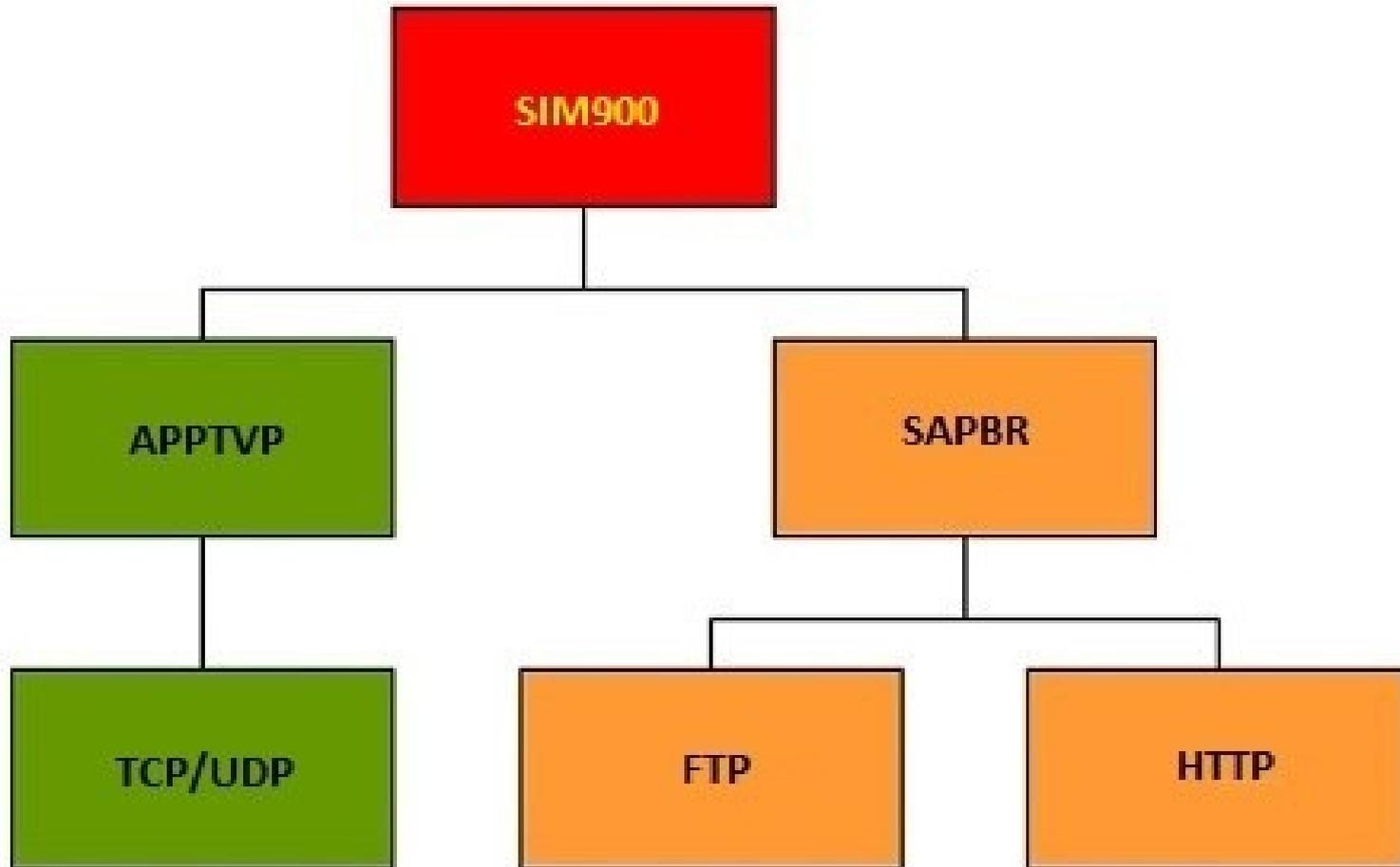
- GSM Трекер можно сделать из одного таракана, практически просто подавая на него питания ( вспомните прошлый курс, когда мы определяли координаты по базам сети!)



# Готовые модули 20-50\$



# AT команды- см курс «Инфстети»



# GPRS

AT+SAPBR=3,1,«CONTYPE»,«GPRS»

- AT+SAPBR=3,1,«APN»,«internet.beeline.ru»
- AT+SAPBR=3,1,«USER»,«beeline»
- AT+SAPBR=3,1,«PWD»,«beeline»
- AT+SAPBR=1,1 — установка GPRS связи
- AT+SAPBR=2,1 — полученный IP адрес
- +SAPBR: 1,1,«10.229.9.115»
- AT+SAPBR=4,1 — текущие настройки соединения
- AT+SAPBR=0,1 — разорвать GPRS соединение
-

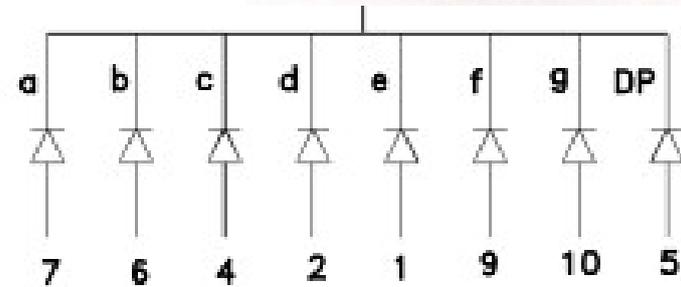
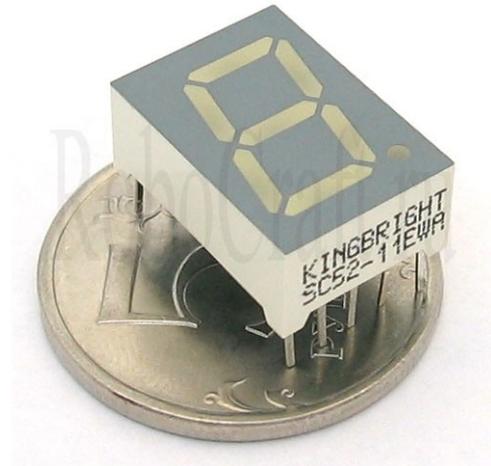
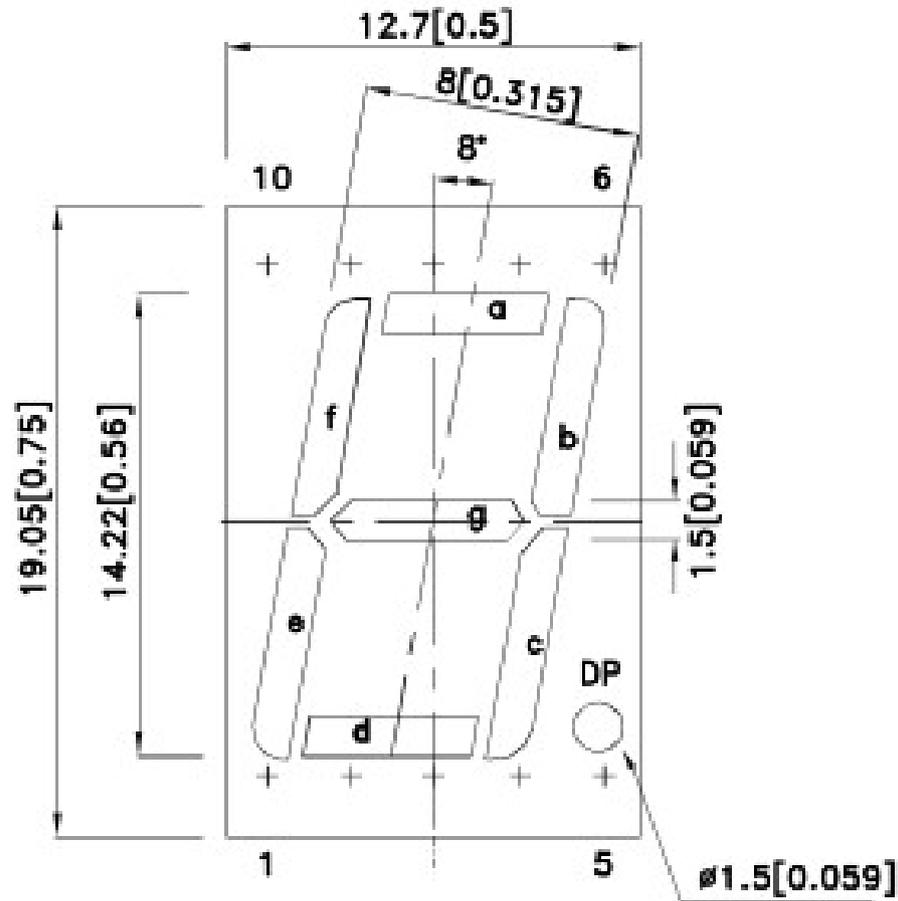
# FTP

- AT+FTPSERV=«ftp.site.ru» — адрес ftp сервера
- AT+FTPUN=«login» — логин
- AT+FTPPW=«password» — пароль
- AT+FTPPUTPATH="/" — путь
- AT+FTPPUTNAME=«test.txt» — имя файла
- AT+FTPMODE=1 — режим работы ftp
- 
- 1, — пассивный. Стоит по умолчанию
- 0, — активный.
- 
- AT+FTRTYPE=«A» — тип передаваемых данных.
- 
- A, — только ASCII символы.
- I, — двоичный режим.
- 
- AT+FTPPUTOPT=«APPE» — режим открытия файла.
- 
- «APPE» — добавить в конец файла
- «STOR» — создать файл. Если файл существует, то он будет перезаписан.
- «STOU» — создать уникальный файл. Если файл существует, то будет создан файл с заданным именем и дополненный в имени цифрами.
- 
- AT+FTPPUT=1 — открытие ftp соединения
- +FTPPUT:1,1,1260 — сессия на запись открыта. Максимальная длина записи 1260 байт.
- AT+FTPPUT=2,10 — запись 10 символов.
- +FTPPUT:2,10 — после появления приглашения нужно ввести 10 символов. Эха не будет.
- AT+FTPPUT:2,0 — закрытие ftp соединения.

# Примеры прилагаемом в прилагаемом файле

- см. прилагаемый файл

# Индикаторы: 7и сегментик



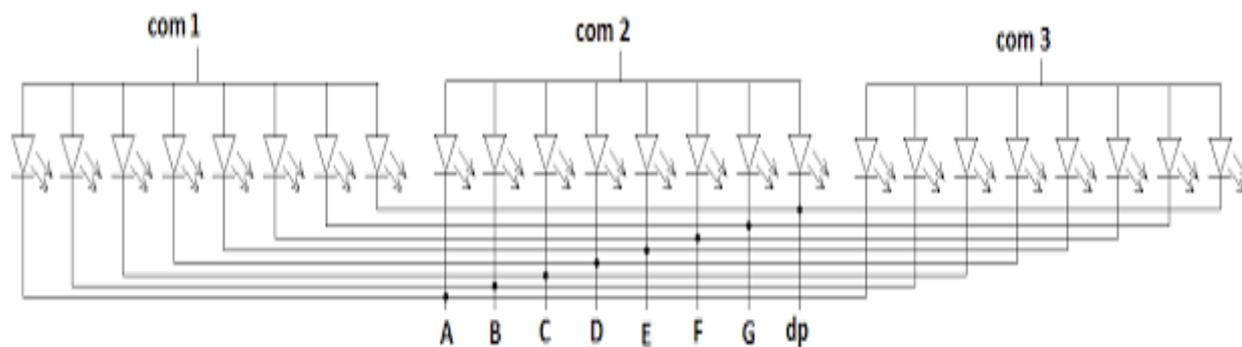
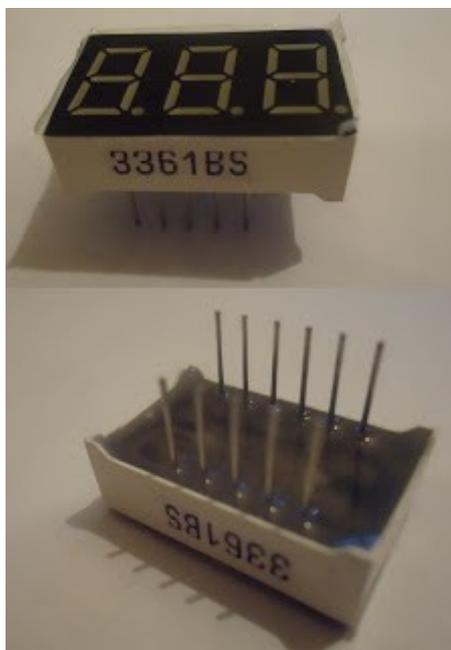
# 7и сегментный индикатор

- Индикатор — это просто 7 светодиодов в едином пластиковом корпусе.
- Можно подключить его напрямую к контроллеру, но тогда будет занято 7 контактов, а в программе будет необходимо реализовать алгоритм преобразования числа из двоичного представления в соответствующие «калькуляторному шрифту» сигналы.
- Нужно помнить про нагрузку- 8 контактов при макс 80ма на порт = 10ма — на сегмент!
-

# Массивчик для сегментов для пересчета цифр=> сегменты

```
int PORT[] = {6, 7, 8, 9, 10, 2, 11, 3, 4, 12, 13, 5}; // ПИНЫ  
byte data[10]={B11111100,B01100000,B11011010,B11110010,  
B01100110,B10110110,B10111110,B11100000,B11111110,B11110110};  
for(int j = 0; j < 8; j++) // Передаем число  
if(segments[temp] & (1 << j)) digitalWrite(PORT, HIGH);  
    digitalWrite(PORT[j], HIGH);  
else digitalWrite(PORT[j], LOW);
```

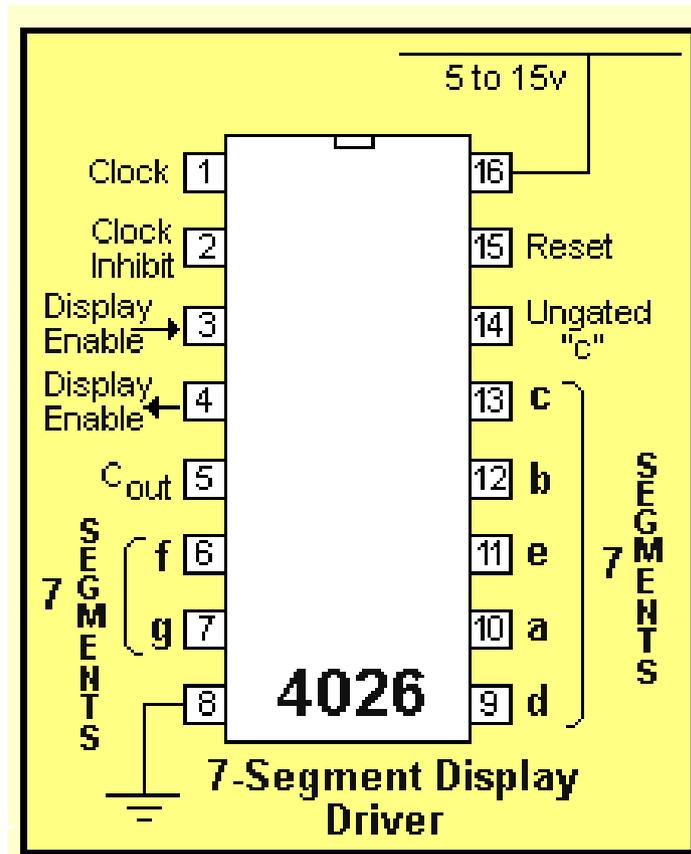
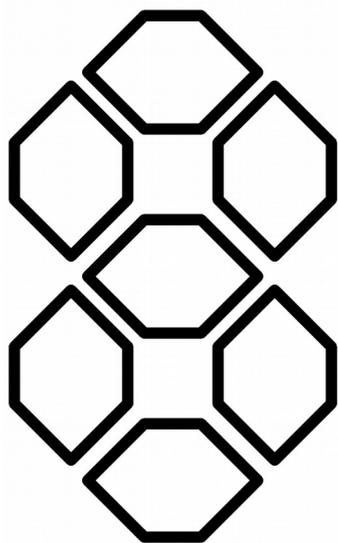
# 3x разрядник





# Не жирно ли 8 портов? CD4026

- Умные люди придумали аппаратный драйвер



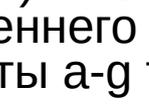
# Драйвер 7и сегментника

- Это простая микросхема с внутренним счётчиком. У неё есть 7 выходов для подключения всех сегментов (a, b, c, d, e, f, g pins), контакт для сбрасывания счётчика в 0 (reset pin) и контакт для увеличения значения на единицу (clock pin). Значение внутреннего счётчика преобразуется в сигналы (включен / выключен) на контакты a-g так, что мы видим соответствующую арабскую цифру.

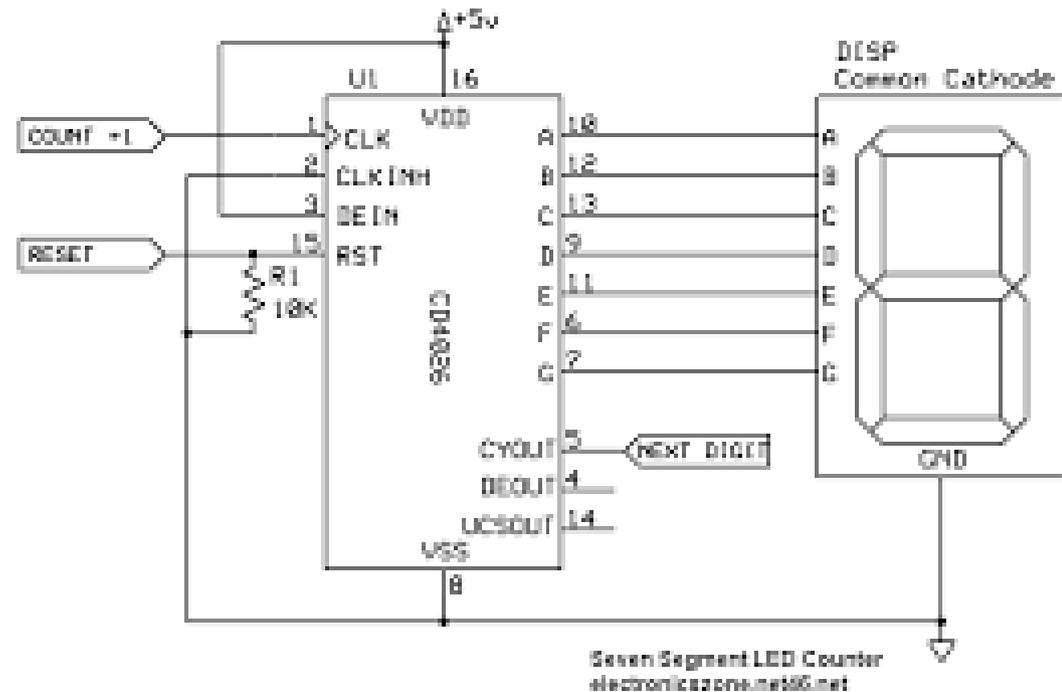
Outputs from the 4026 counter and display driver IC

Count	a	b	c	d	e	f	g	h
0	•	•	•	•	•	•	•	•
1		•	•					•
2	•	•	•	•				•
3	•	•	•					•
4		•	•				•	•
5	•	•	•				•	•
6	•	•	•	•				•
7	•	•	•					
8	•	•	•	•	•			•
9	•	•	•	•	•	•		•

• = segment on. h is used to drive other counters.



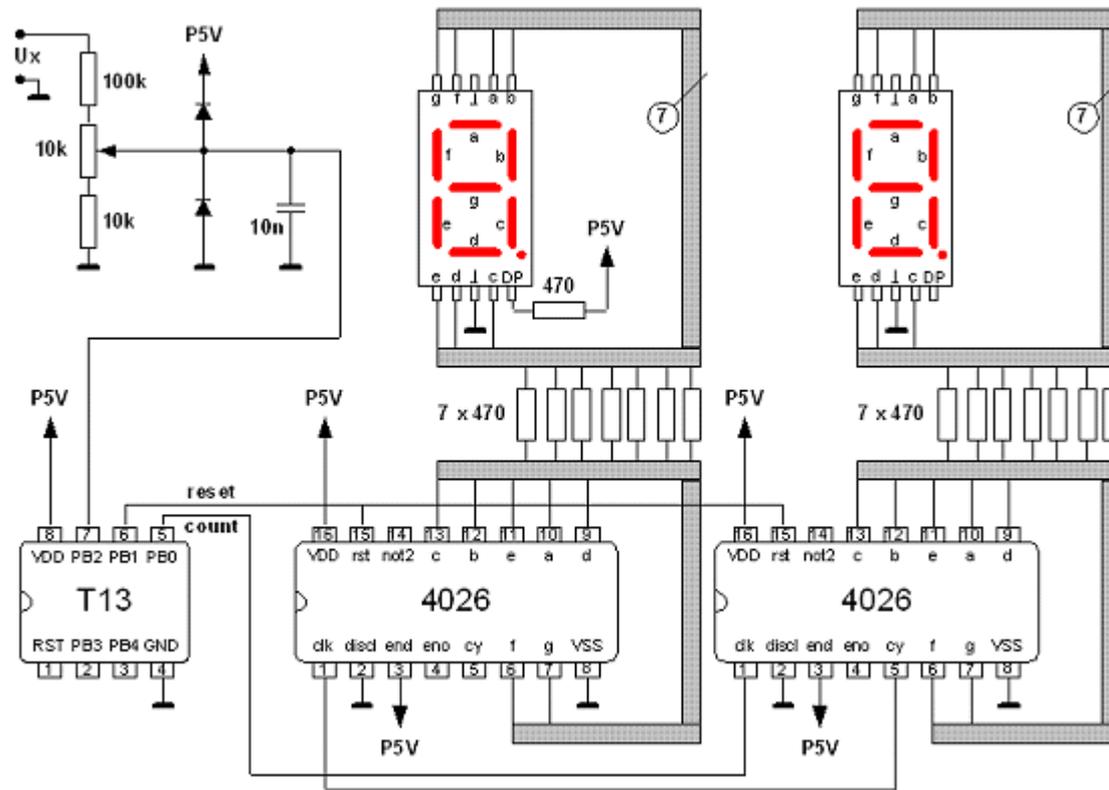
7-segment display



# Каскадирование

- На микросхеме есть ещё один выход, обозначенный как «÷10». Его значение всё время LOW за исключением момента переполнения, когда значение счётчика равно 9, а его увеличивают на единицу. В этом случае значением счётчика снова становится 0, но выход «÷10» становится HIGH до момента следующего инкремента. Его можно соединить с clock pin другого драйвера и таким образом получить счётчик для двузначных чисел. Продолжая эту цепочку, можно выводить сколь угодно длинные числа, используя всего 2 шины.

# Схемотехническое решение



```

#define CLOCK_PIN 2
• #define RESET_PIN 3
•
• /*
• * Функция resetNumber обнуляет текущее значение
• * на счётчике
• */
• void resetNumber()
• {
•     // Для сброса на мгновение ставим контакт
•     // reset в HIGH и возвращаем обратно в LOW
•     digitalWrite(RESET_PIN, HIGH);
•     digitalWrite(RESET_PIN, LOW);
• }
•
• /*
• * Функция showNumber устанавливает показания индикаторов
• * в заданное неотрицательное число `n` вне зависимости
• * от предыдущего значения
• */
• void showNumber(int n)
• {
•     // Первым делом обнуляем текущее значение
•     resetNumber();
•
•     // Далее быстро «прокликаем» счётчик до нужного
•     // значения
•     while (n-- > 0) {
•         digitalWrite(CLOCK_PIN, HIGH);
•         digitalWrite(CLOCK_PIN, LOW);
•     }
• }
•
• void setup()
• {
•     pinMode(RESET_PIN, OUTPUT);
•     pinMode(CLOCK_PIN, OUTPUT);
•
•     // Обнуляем счётчик при старте, чтобы он не оказался
•     // в случайном состоянии
•     resetNumber();
• }
•
• void loop()
• {
•     // Получаем количество секунд в неполной минуте
•     // с момента старта и выводим его на индикаторы
•     showNumber((millis() / 1000) % 60);
•     delay(1000);
• }
•
•
•

```

# Матрица светодиодная

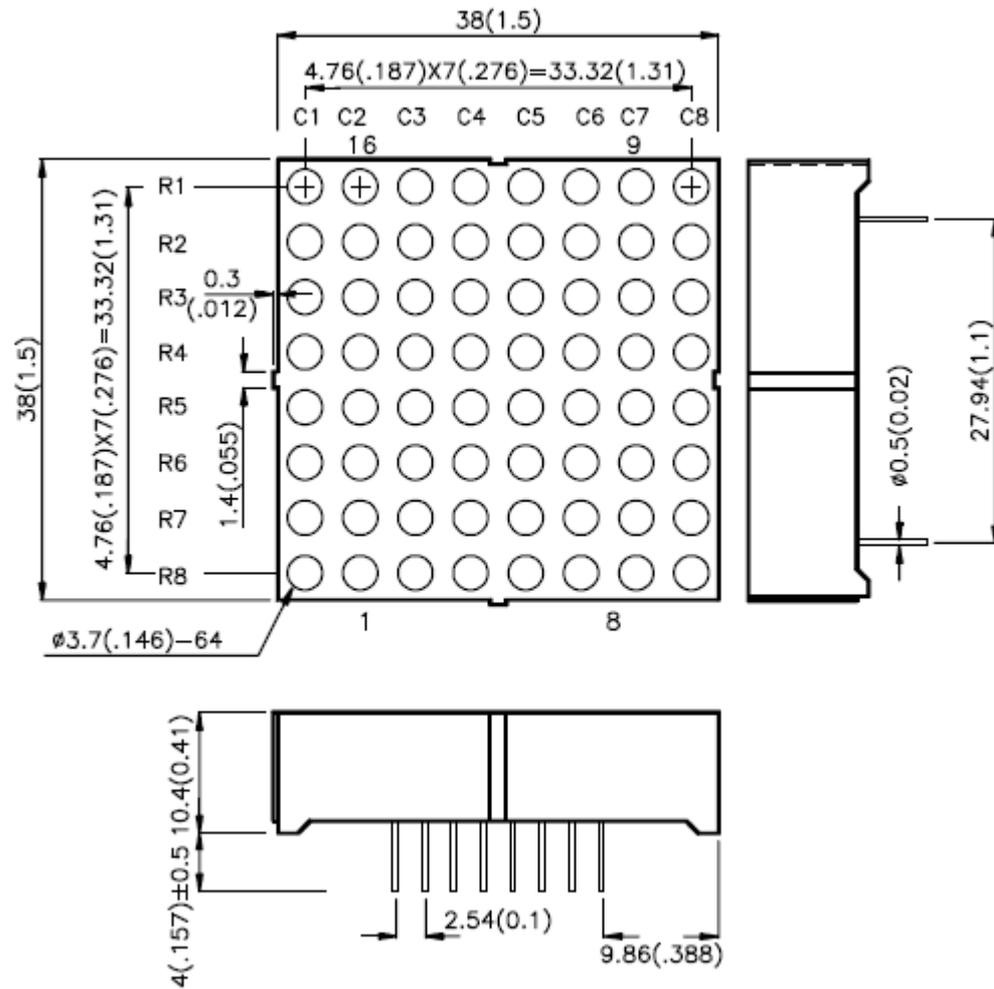
- Разрядность матрицы разная. Бывают одно/много/ полноцветные (RGB)



# Пример: ТС 15-11

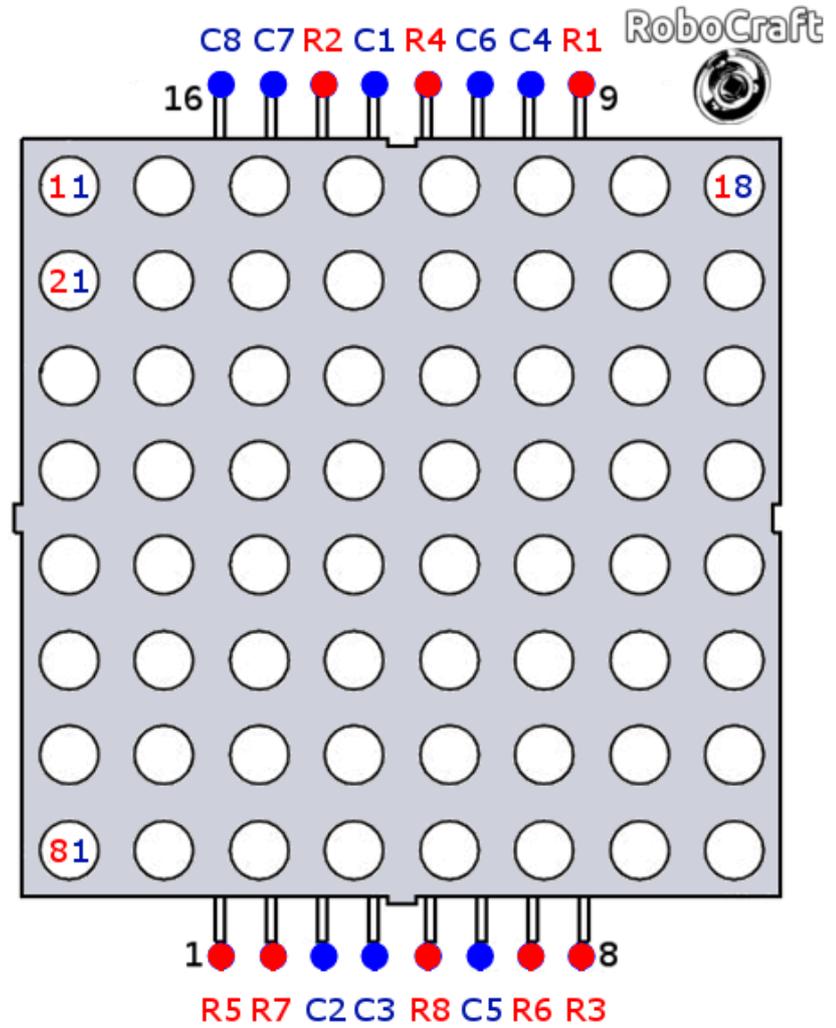
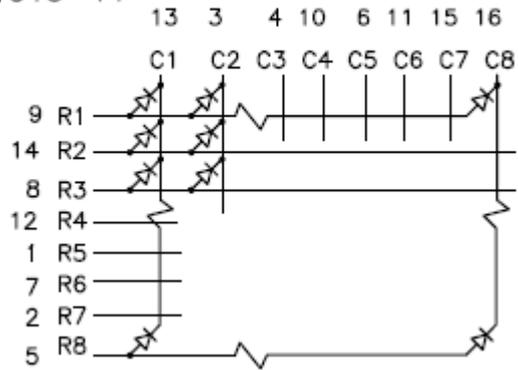


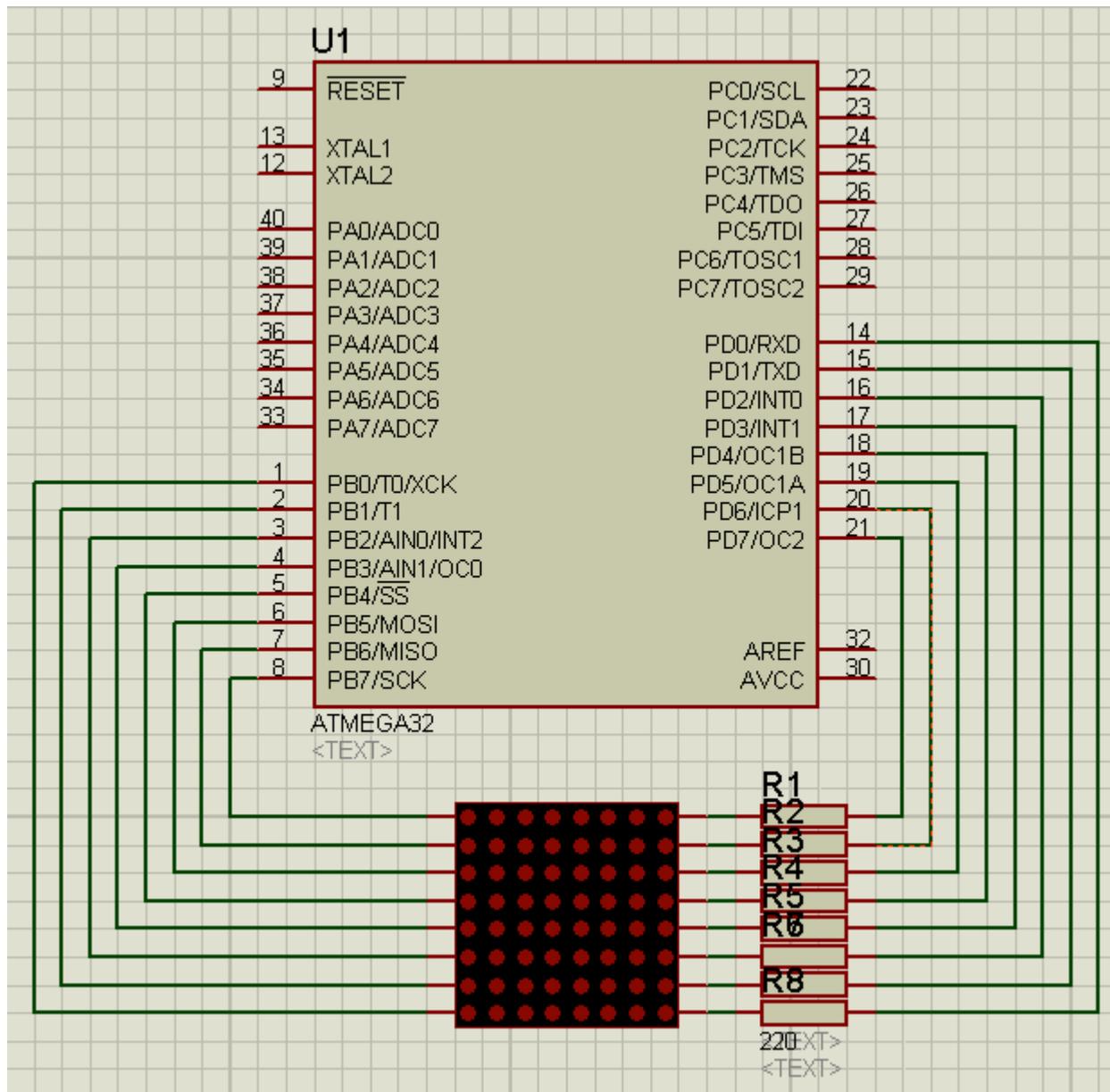
Package Dimensions



Internal Circuit Diagram

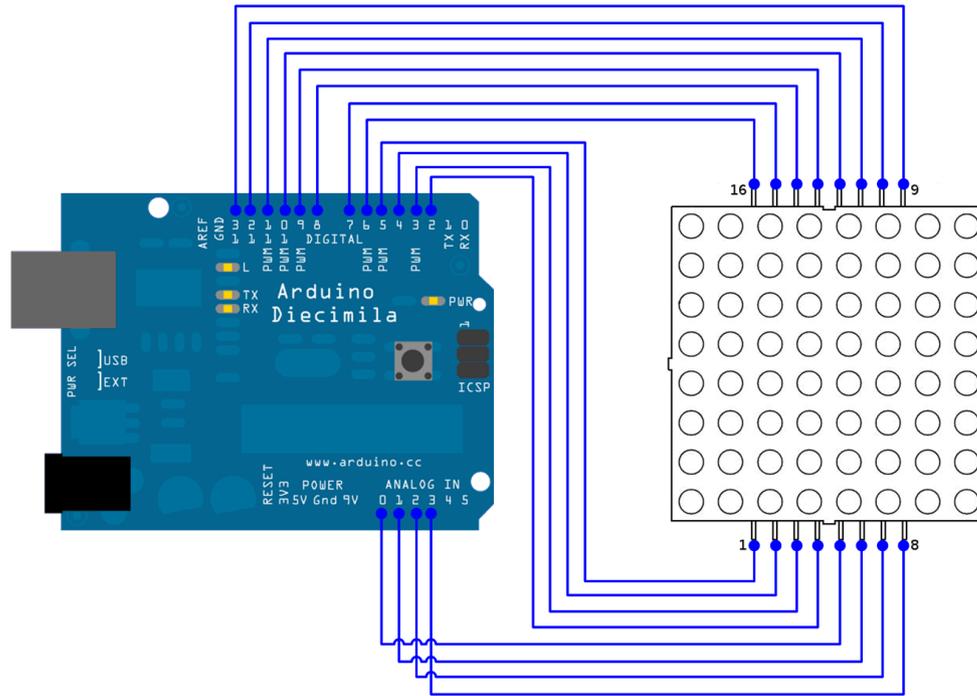
TC15-11



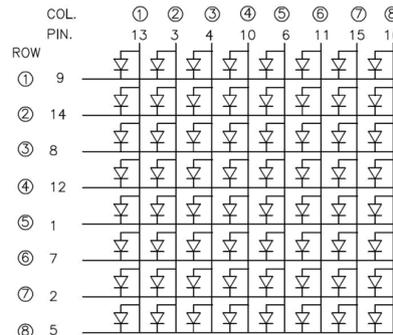


## Direct wiring an Arduino to an LED matrix

// pin[xx] on led matrix connected to nn on Arduino (-1 is dummy to make array start at pos 1)  
 int pins[17]= {-1, 5, 4, 3, 2, 14, 15, 16, 17, 13, 12, 11, 10, 9, 8, 7, 6};



Column and row anode and cathode wiring to pins in matrix



**Примечание:**  
 ! В TC15-11 диоды включены наоборот !

NB There probably should be some current limiting resistors in here. I don't take responsibility if you blow up your Arduino and/or your LED matrix

# Работать с матрицей также удобно через массив

- `boolean A[8][8] =`
- `{0,0,1,1,1,1,0,0,`
- `0,1,0,0,0,0,1,0,`
- `0,1,0,0,0,0,1,0,`
- `0,1,0,0,0,0,1,0,`
- `0,1,1,1,1,1,1,0,`
- `0,1,0,0,0,0,1,0,`
- `0,1,0,0,0,0,1,0,`
- `0,1,0,0,0,0,1,0}`

# Вывод массива

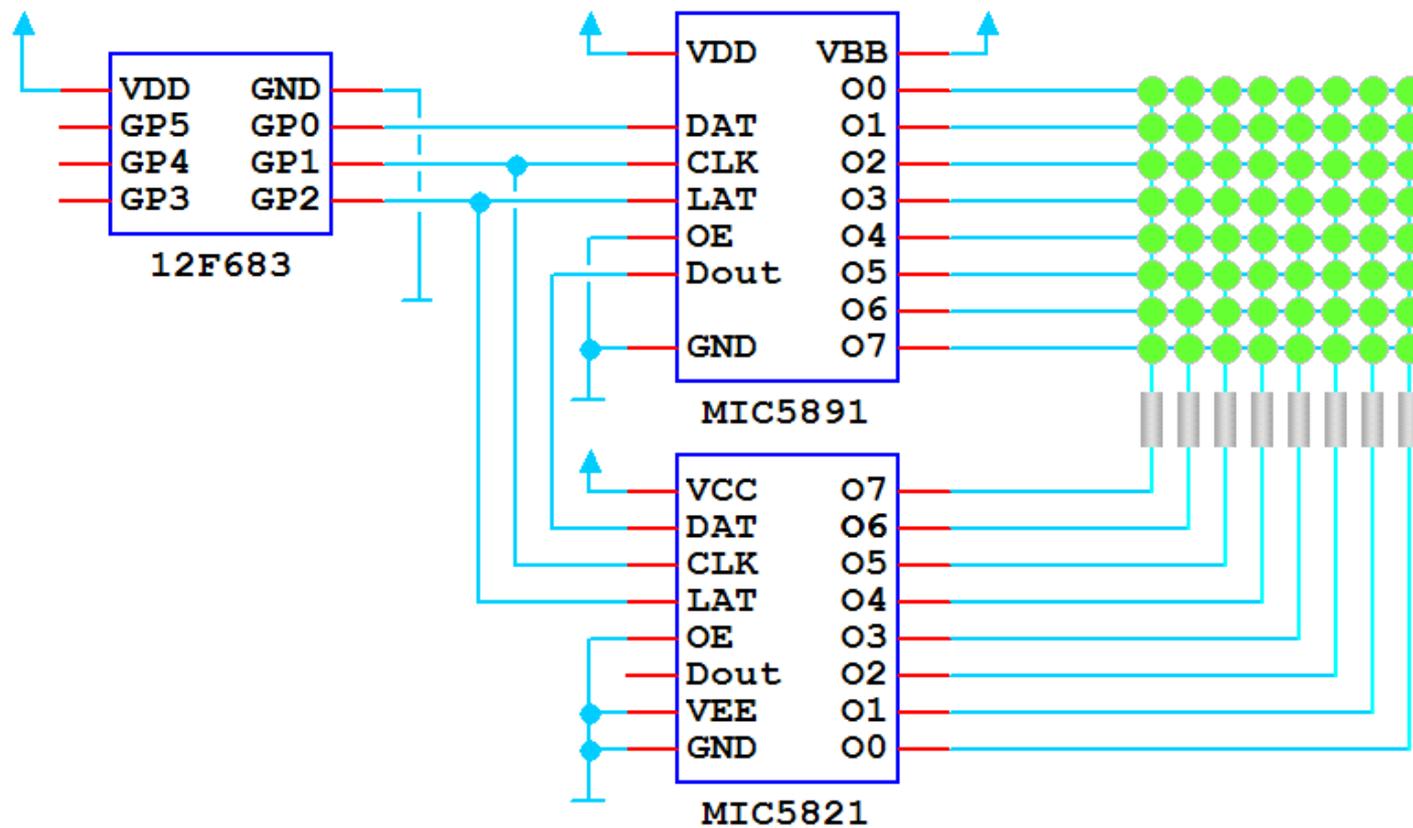
```
• {
• int i, j;
• for (int c=0; c <v; c ++ ) // Тот самый цикл задержки отрисовки информации
• {
•   for (i=0; i<8; i++)
•     for (j=0; j<8; j++)
•       {
•         if (screen[i][j]= =1) // Если элемент массива = 1
•           {
•             digitalWrite(row[i], LOW);
•             digitalWrite(col[j], HIGH);
•             delayMicroseconds(dms);
•             digitalWrite(row[i], HIGH);
•             digitalWrite(col[j], LOW); // То включить светодиод           в проверяемой позиции, задержать включенным, выключить светодиод.
•           }
•         else
•           {
•             digitalWrite(row[i], HIGH);
•             digitalWrite(col[j], LOW);
•             delayMicroseconds(dms);
•             digitalWrite(row[i], HIGH);
•             digitalWrite(col[j], LOW);
•           }
•         } // Иначе выключить светодиод в проверяемой позиции, задержать включенным, обновить экран. (даже если светодиод и был выключенным в этой позиции, то
•         все равно его нужно «выключить» для того, чтобы проверка в цикле и действия проходили с одинаковой задержкой. Если в позиции 1 – то выполняются действия, если в
•         позиции 0 – все равно должно что-то выполниться либо выполниться задержка, сопоставимая с задержкой выполнения включения светодиода)
•       }
•   }
• }
•
•
•
•
```

# Бегущая строка

- `void scroll(boolean screen[8][8]) // В качестве параметра передаем наш двухмерный массив`
- `{`
- `boolean buf[8][1];`
- `for (int i=0; i<8; i++)`
- `{`
- `buf[i][0]=screen[i][0]; // Считали 1ю колонку в буфер`
- `}`
- `for (int i=0; i<8; i++)`
- `for (int j=0; j<8; j++)`
- `{`
- `screen[i][j]=screen[i][j+1]; // Сдвинули матрицу на один столбец влево`
- `}`
- `for (int i=0; i<8; i++)`
- `{`
- `screen[i][8-1]=buf[i][0]; // Записали содержимое буфера (первую колонку) в конец матрицы`
- `}`
- `}`

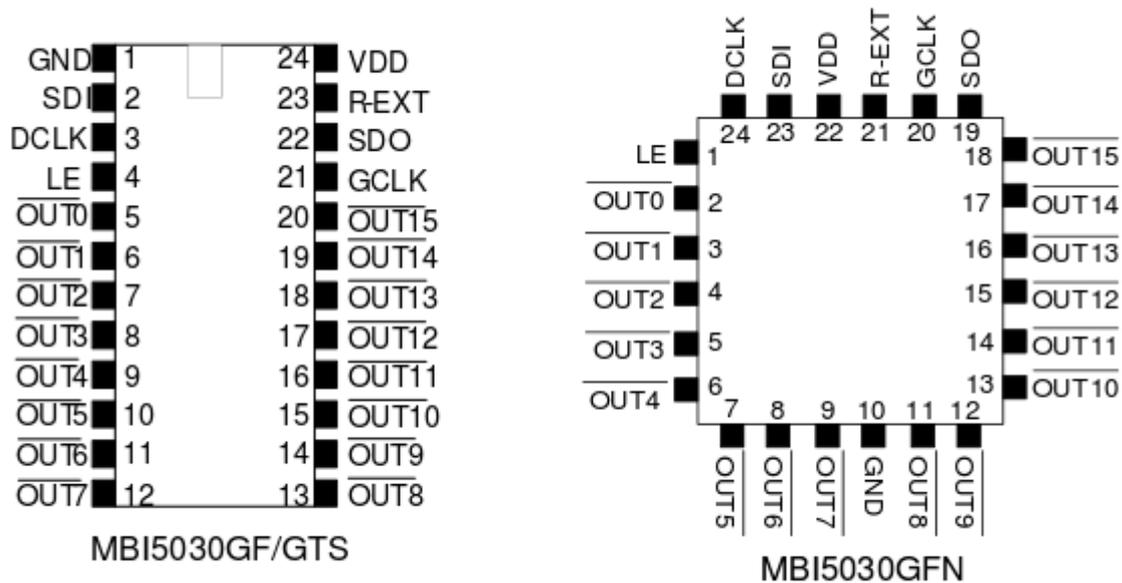
# Но 16 выводов на одну панель совсем уж неприлично...

Решение — старый знакомый — драйвер 7и сегментника + точка

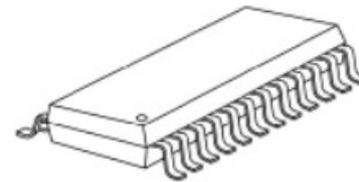


# С RGB так просто уже не получится

т. к. нужно еще и управлять яркостью («ШИМить»). Нужен специальный драйвер, например MBI5030

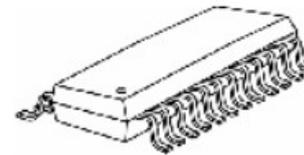


## Small Outline Package



GF: SOP24-300-1.00

## Thin Shrink SOP

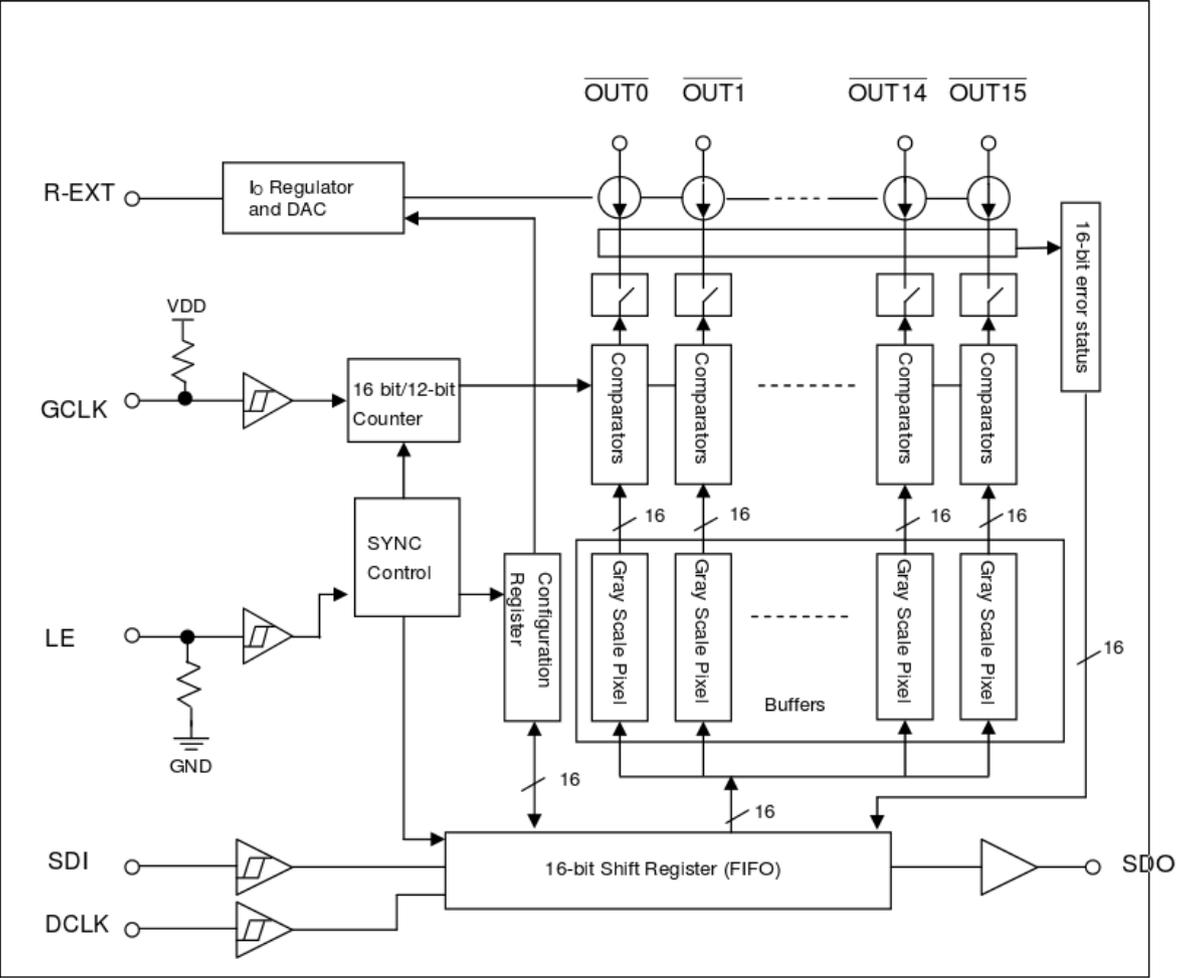


GTS: TSSOP24-173-0.65

## Quad Flat No-Lead



GFN: QFN24-4\*4-0.5



## Terminal Description

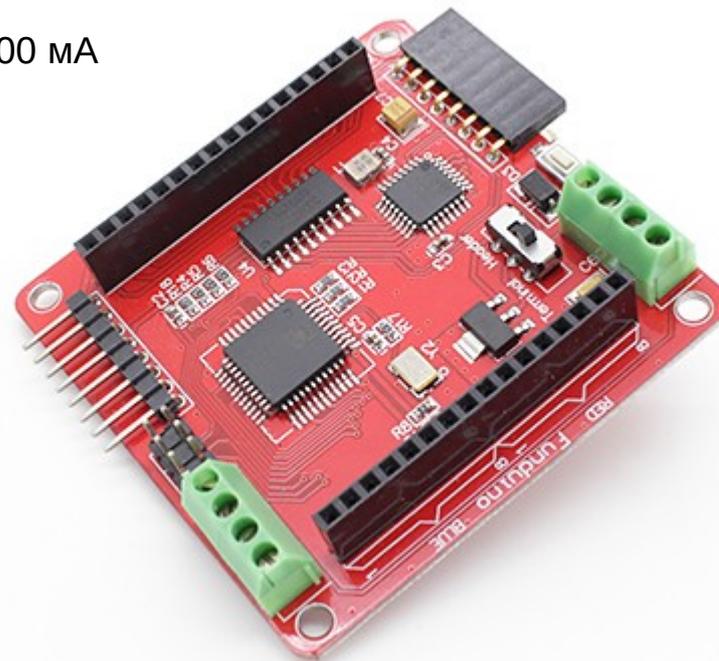
Pin Name	Function
GND	Ground terminal for control logic and current sink
SDI	Serial-data input to the shift register
DCLK	Clock input terminal used to shift data on rising edge and carries command information when LE is asserted.
LE	Data strobe terminal and controlling command with DCLK
$\overline{\text{OUT0}} \sim \overline{\text{OUT15}}$	Constant current output terminals
GCLK	Gray scale clock terminal Clock input for gray scale. The gray scale display is counted by gray scale clock comparing with input data.
SDO	Serial-data output to the receiver-end SDI of next driver IC
R-EXT	Input terminal used to connect an external resistor for setting up output current for all output channels
VDD	3.3V/5V supply voltage terminal

# Развертка

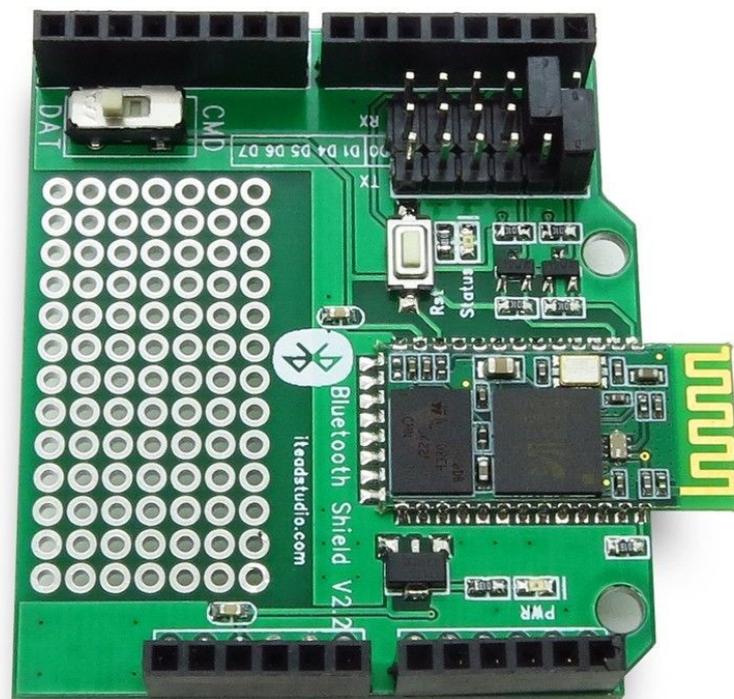
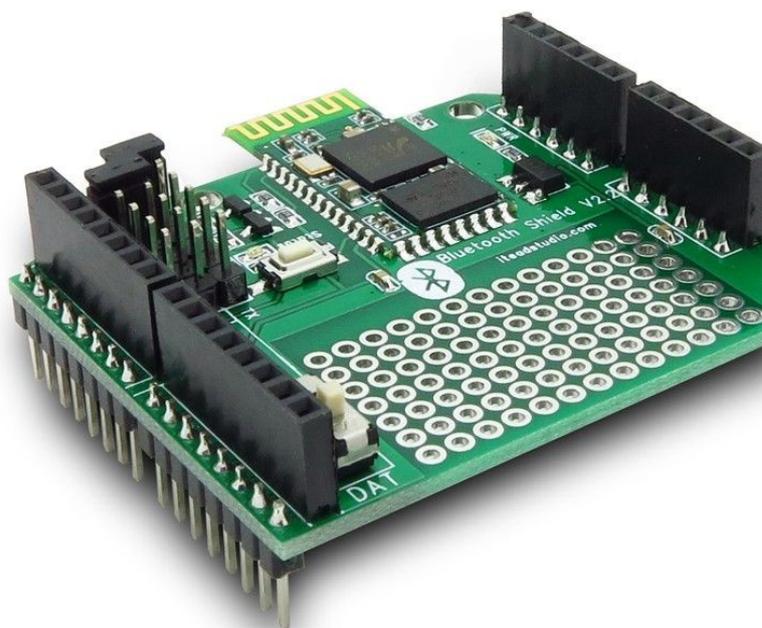
- Зачем микроконтроллер? В системе появляется развертка, т. е. Работаем по принципу телевизора.
- Развертка через строку слегка усложняет программирование, но заметно улучшает восприятие.

# Для Arduino есть готовый шилд ( и не один)

- Colorduino V2.0 драйвер управления RGB светодиодной матрицей
- 8+6 bit цветовой уровень по каждому каналу цвета.
- Тактовая частота поддержки ШИМ - 16 мГц
- Интерфейсы GPIO (дискретных сигналов) и ADC (АЦП)
- Аппаратные интерфейсы UART и I2C с поддержкой каскадного соединения
- 24 канала со стабилизированным выходным током 60 мА
- 8 каналов с максимальным выходным током 500 мА
- Мозг: ATMeга 328P



# Синий зуб ( в схему не полезем )



# SLAVE

```
#include <NewSoftSerial.h> //Software Serial Port
```

```
#define RxD 6
```

```
#define TxD 7
```

```
#define DEBUG_ENABLED 1
```

```
NewSoftSerial blueToothSerial(RxD,TxD);
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  pinMode(RxD, INPUT);
```

```
  pinMode(TxD, OUTPUT);
```

```
  setupBlueToothConnection();
```

```
}
```

# SLAVE

```
void loop()
{
  char recvChar;
  while(1){
    if(blueToothSerial.available()){//check if there's any data sent from the remote bluetooth shield
      recvChar = blueToothSerial.read();
      Serial.print(recvChar);
    }
    if(Serial.available()){//check if there's any data sent from the local serial terminal, you can add
the other applications here
      recvChar = Serial.read();
      blueToothSerial.print(recvChar);
    }
  }
}
```

# SLAVE

```
void setupBluetoothConnection()
{
  blueToothSerial.begin(38400); //Set BluetoothBee BaudRate to default baud rate 38400
  blueToothSerial.print("\r\n+STWMOD=0\r\n"); //set the bluetooth work in slave mode
  blueToothSerial.print("\r\n+STNA=SeedBTSlave\r\n"); //set the bluetooth name as
  "SeedBTSlave"
  blueToothSerial.print("\r\n+STOAUT=1\r\n"); // Permit Paired device to connect me
  blueToothSerial.print("\r\n+STAUTO=0\r\n"); // Auto-connection should be forbidden here
  delay(2000); // This delay is required.
  blueToothSerial.print("\r\n+INQ=1\r\n"); //make the slave bluetooth inquirable
  Serial.println("The slave bluetooth is inquirable!");
  delay(2000); // This delay is required.
  blueToothSerial.flush();
}
•
•
```

# Ethernet (Wiznet W5100)



# Наиприятнейшая штукавина

Target Applications The W5100 is well suited for many embedded applications, including:

-Home Network Devices: Set-Top Boxes, PVRs, Digital Media Adapters

-Serial-to-Ethernet: Access Controls, LED displays, Wireless AP relays, etc.

-Parallel-to-Ethernet: POS / Mini Printers, Copiers

-USB-to-Ethernet: Storage Devices, Network Printers

-GPIO-to-Ethernet: Home Network Sensors -Security Systems: DVRs, Network Cameras, Kiosks

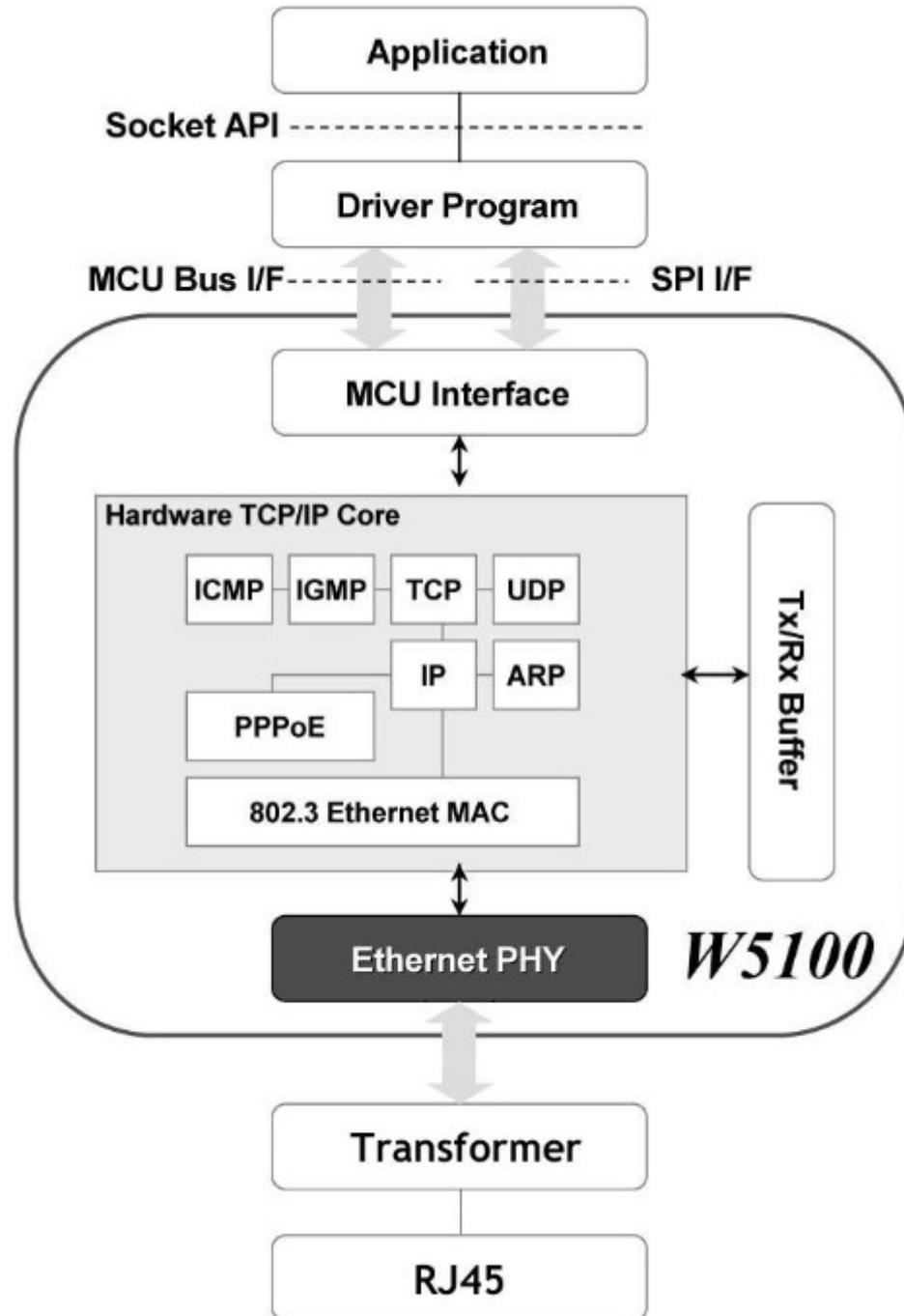
-Factory and Building Automations

-Medical Monitoring Equipments -Embedded Servers

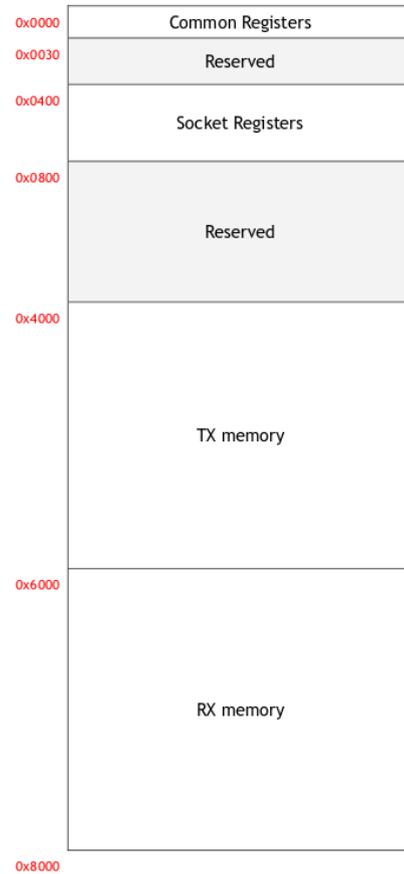
# Которая много чего умеет

- Support Hardwired TCP/IP Protocols : TCP, UDP, ICMP, IPv4 ARP, IGMP, PPPoE, Ethernet
- -10BaseT/100BaseTX Ethernet PHY embedded
- -Support Auto Negotiation (Full-duplex and half duplex)
- -Support Auto MDI/MDIX
- -Support ADSL connection (with support PPPoE Protocol with PAP/CHAP Authentication mode)
- -Supports 4 independent sockets simultaneously
- - Not support IP Fragmentation
- - Internal 16Kbytes Memory for Tx/Rx Buffers
- - 0.18  $\mu\text{m}$  CMOS technology
- - 3.3V operation with 5V I/O signal tolerance
- - Small 80 Pin LQFP Package
- - Lead-Free Package
- - Support Serial Peripheral Interface(SPI MODE 0)
- - Multi-function LED outputs (TX, RX, Full/Half duplex, Collision, Link, Speed)

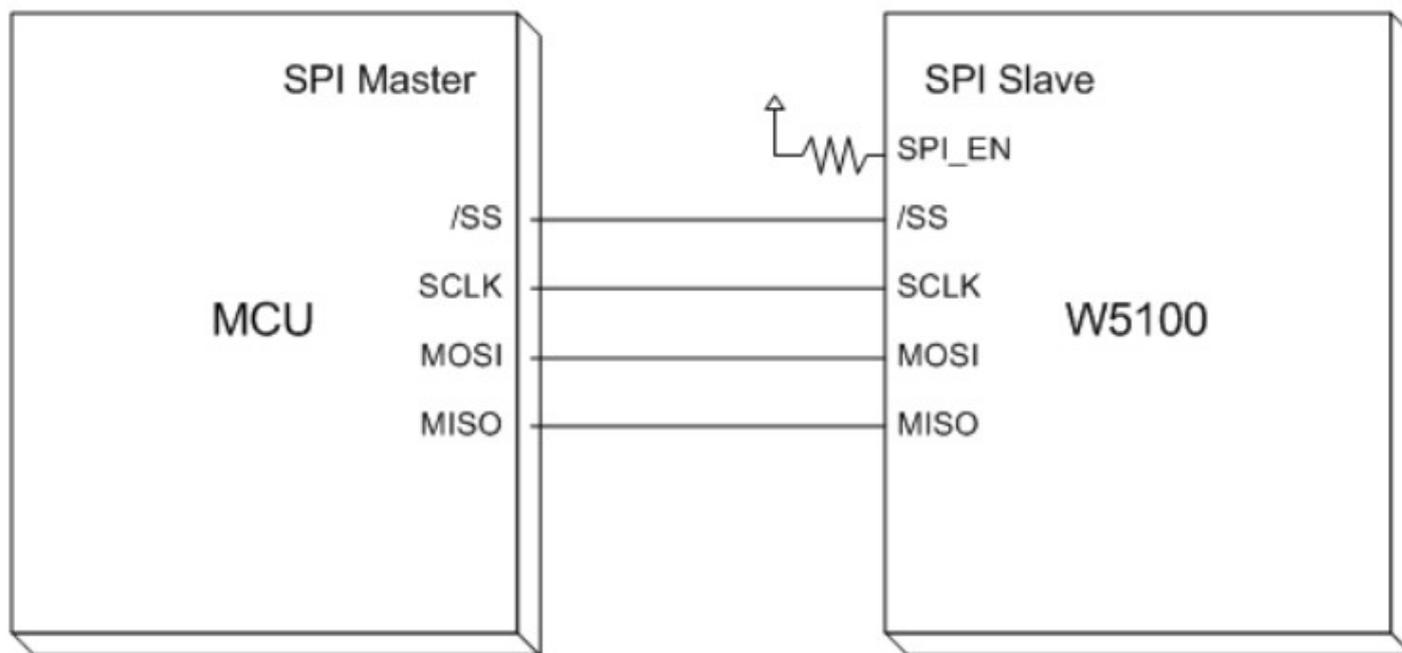
# Block Diagram



# Memory Map



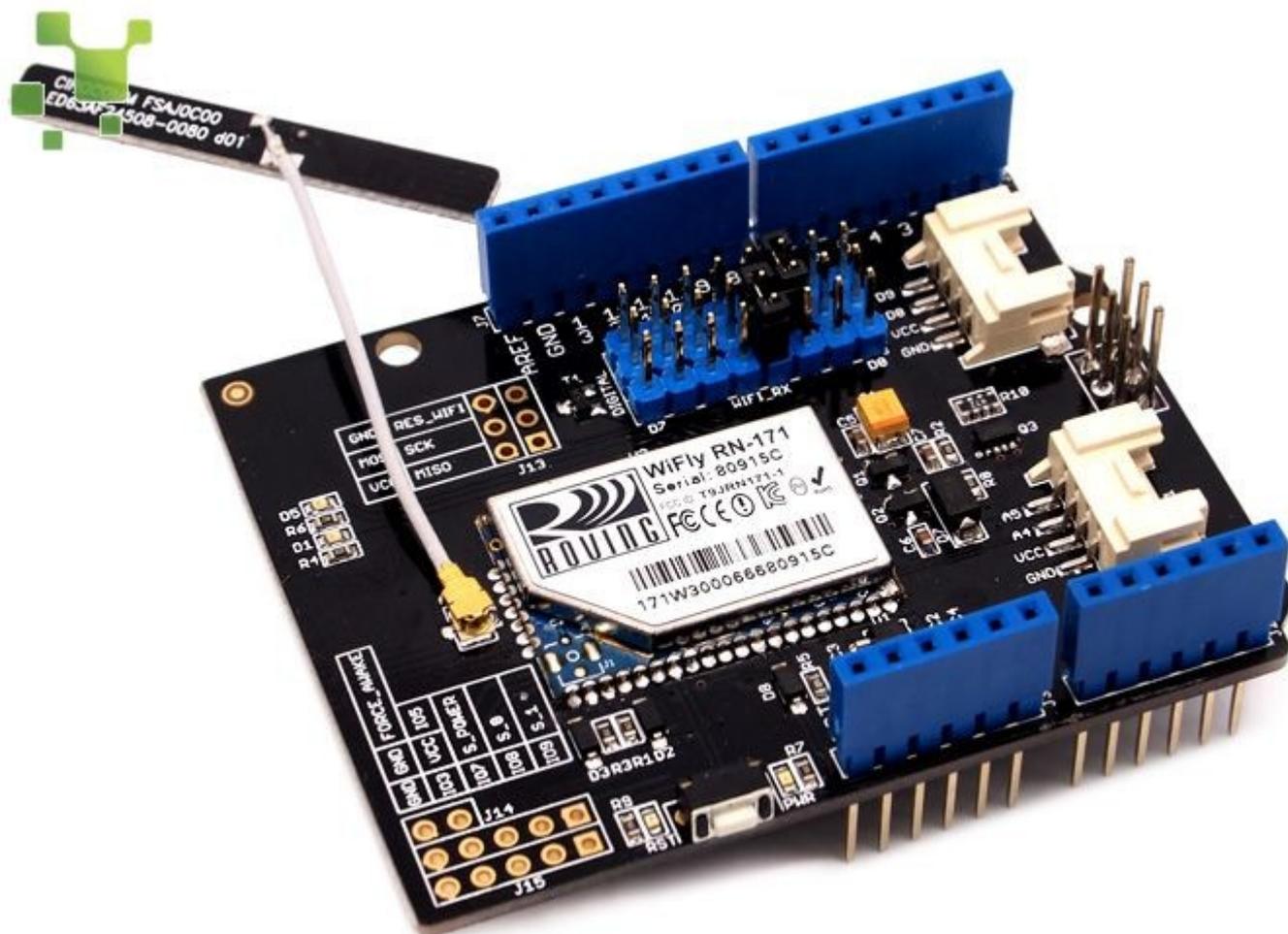
# Как подключиться? SPI!



# Как с ней работать? Ehernet.h!

- Разбираем стандартные примеры набора скейтчей Ehernet и «WebClient» «WebServer»

WiFi опустим, с точки зрения программирования, он похож на ETH, но скорость работы позволяет использовать UART.







```

#include <Wire.h>
#include <ADXL345.h>
.
.
• ADXL345 adxl; //variable adxl is an instance of the ADXL345 library
.
• void setup(){
•   Serial.begin(9600);
•   adxl.powerOn();
•
•   //set activity/ inactivity thresholds (0-255)
•   adxl.setActivityThreshold(75); //62.5mg per increment
•   adxl.setInactivityThreshold(75); //62.5mg per increment
•   adxl.setTimeInactivity(10); // how many seconds of no activity is inactive?
•
•   //look of activity movement on this axes - 1 == on; 0 == off
•   adxl.setActivityX(1);
•   adxl.setActivityY(1);
•   adxl.setActivityZ(1);
•
•   //look of inactivity movement on this axes - 1 == on; 0 == off
•   adxl.setInactivityX(1);
•   adxl.setInactivityY(1);
•   adxl.setInactivityZ(1);
•
•   //look of tap movement on this axes - 1 == on; 0 == off
•   adxl.setTapDetectionOnX(0);
•   adxl.setTapDetectionOnY(0);
•   adxl.setTapDetectionOnZ(1);
•
•   //set values for what is a tap, and what is a double tap (0-255)
•   adxl.setTapThreshold(50); //62.5mg per increment
•   adxl.setTapDuration(15); //625us per increment
•   adxl.setDoubleTapLatency(80); //1.25ms per increment
•   adxl.setDoubleTapWindow(200); //1.25ms per increment
•
•   //set values for what is considered freefall (0-255)
•   adxl.setFreeFallThreshold(7); //(5 - 9) recommended - 62.5mg per increment
•   adxl.setFreeFallDuration(45); //(20 - 70) recommended - 5ms per increment
•
•   //setting all interrupts to take place on int pin 1
•   //I had issues with int pin 2, was unable to reset it
•   adxl.setInterruptMapping( ADXL345_INT_SINGLE_TAP_BIT,  ADXL345_INT1_PIN );
•   adxl.setInterruptMapping( ADXL345_INT_DOUBLE_TAP_BIT,  ADXL345_INT1_PIN );
•   adxl.setInterruptMapping( ADXL345_INT_FREE_FALL_BIT,   ADXL345_INT1_PIN );
•   adxl.setInterruptMapping( ADXL345_INT_ACTIVITY_BIT,    ADXL345_INT1_PIN );
•   adxl.setInterruptMapping( ADXL345_INT_INACTIVITY_BIT,  ADXL345_INT1_PIN );
•
•   //register interrupt actions - 1 == on; 0 == off
•   adxl.setInterrupt( ADXL345_INT_SINGLE_TAP_BIT, 1);
•   adxl.setInterrupt( ADXL345_INT_DOUBLE_TAP_BIT, 1);
•   adxl.setInterrupt( ADXL345_INT_FREE_FALL_BIT,  1);
•   adxl.setInterrupt( ADXL345_INT_ACTIVITY_BIT,   1);
•   adxl.setInterrupt( ADXL345_INT_INACTIVITY_BIT, 1);
• }
•

```

```
void loop(){
```

```
    //Boring accelerometer stuff
```

```
    int x,y,z;
```

```
    adxl.readXYZ(&x, &y, &z); //read the accelerometer values and store them in variables x,y,z
```

```
    // Output x,y,z values
```

```
    Serial.print("values of X , Y , Z: ");
```

```
    Serial.print(x);
```

```
    Serial.print(" , ");
```

```
    Serial.print(y);
```

```
    Serial.print(" , ");
```

```
    Serial.println(z);
```

```
    double xyz[3];
```

```
    double ax,ay,az;
```

```
    adxl.getAcceleration(xyz);
```

```
    ax = xyz[0];
```

```
    ay = xyz[1];
```

```
    az = xyz[2];
```

```
    Serial.print("X=");
```

```
    Serial.print(ax);
```

```
    Serial.println(" g");
```

```
    Serial.print("Y=");
```

```
    Serial.print(ay);
```

```
    Serial.println(" g");
```

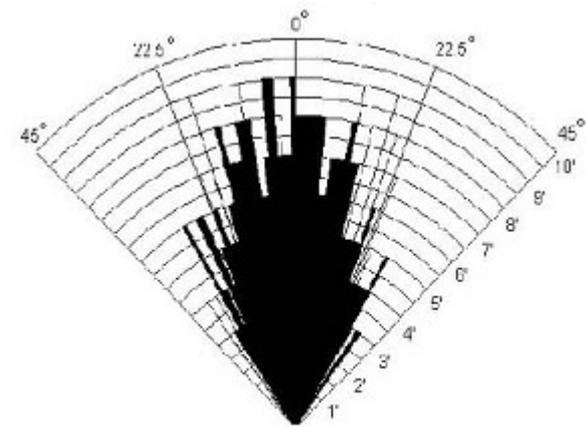
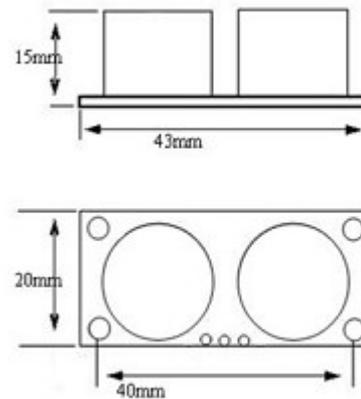
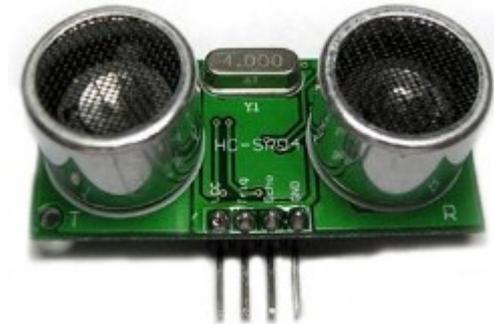
```
    Serial.print("Z=");
```

```
    Serial.println(az);
```

```
    Serial.println(" g");
```

# Датчик ультразвуковой (HC-SR04)

- Расстояние: 2 см ~ 5 м
- Обзор ~ 30 градусов
- Разрешающая способность до 0.3 см

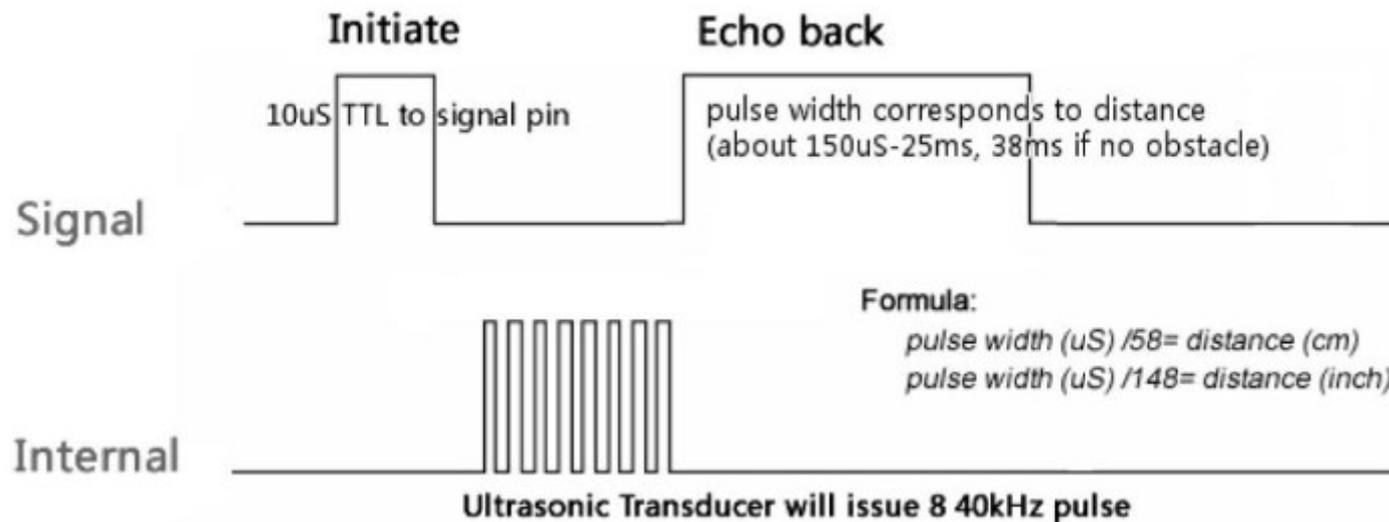


*Practical test of performance,  
Best in 30 degree angle*

Library:

<http://iteadstudio.com/store/images/produce/Robot/HCSR04/Ultrasonic.ra>

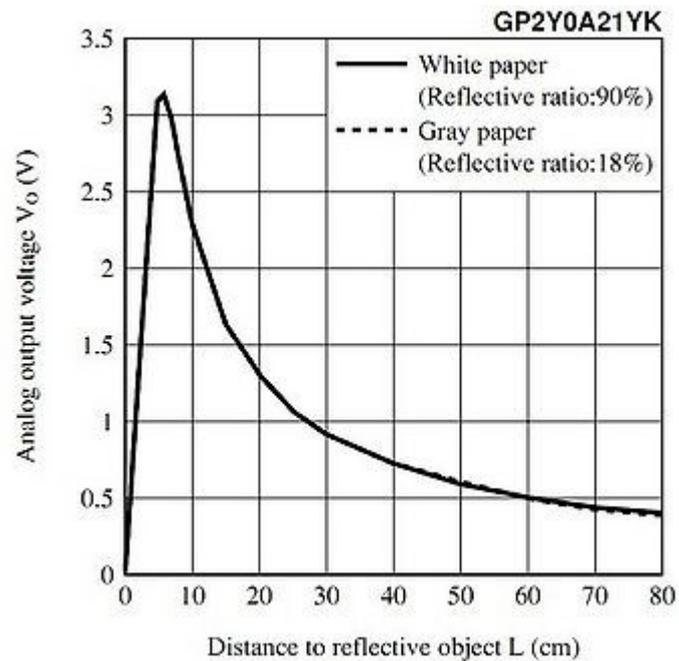
- The recommend cycle period should be no
- less than 50ms. If a 10 $\mu$ s width trigger pulse is sent to the signal pin, the Ultrasonic module will
- output eight 40kHz ultrasonic signal and detect the echo back
- If no obstacle is detected, the output pin will give a 38ms high level signal



# Пример работы

- #include "Ultrasonic.h"
- #include <LiquidCrystal.h>
- LiquidCrystal lcd(11, 10, 9, 4, 5, 6, 7);
- Ultrasonic ultrasonic(12,13);
- 
- void setup() {
- lcd.begin(16, 2);
- lcd.print("testing...");
- }
- 
- void loop()
- {
- lcd.clear();
- lcd.setCursor(0, 0);
- lcd.print(ultrasonic.Ranging(CM));
- lcd.print("cm");
- 
- delay(100);
- }
-

# Инфракрасный датчик (аналоговый)



# NFC RFID брелки/сенсоры

- Применяются для беспроводной идентификации. Работают на частоте 13.56 мегагерц. Требуют NFC приемник. Дальность связи до 10 см. На карте до 8K EEPROM



# Near Field Communication

- Near field communication, NFC («коммуникация ближнего поля», «ближняя бесконтактная связь») — технология беспроводной высокочастотной связи малого радиуса действия, которая дает возможность обмена данными между устройствами, находящимися на расстоянии около 10 сантиметров анонсирована в 2004 г.

# I2C(SPI) NFC контроллер

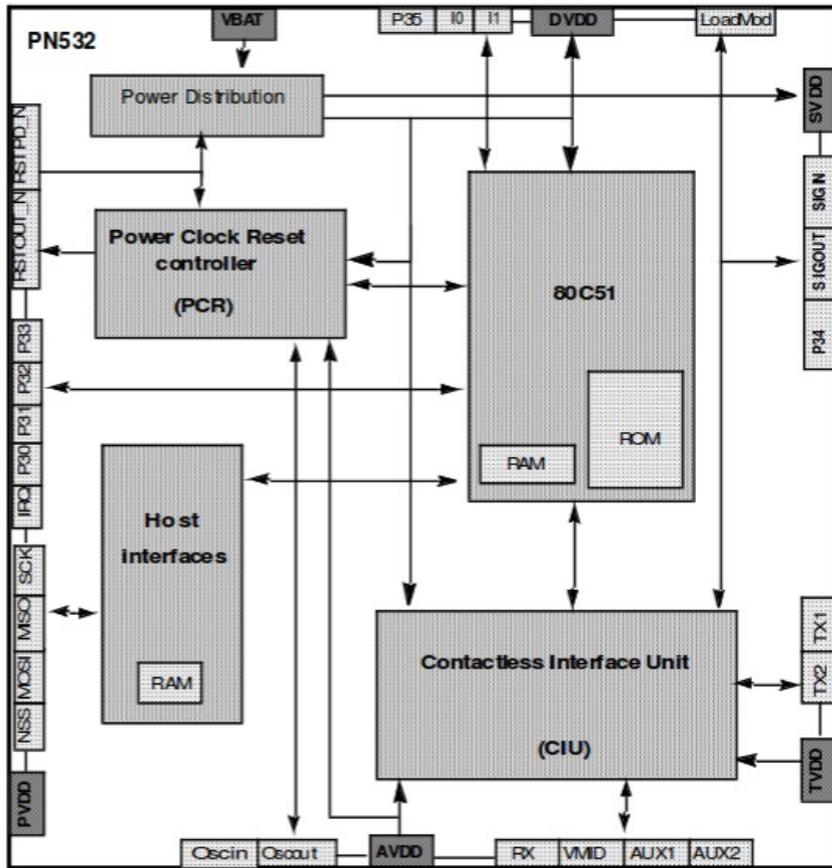


Fig 1. Block diagram of PN532

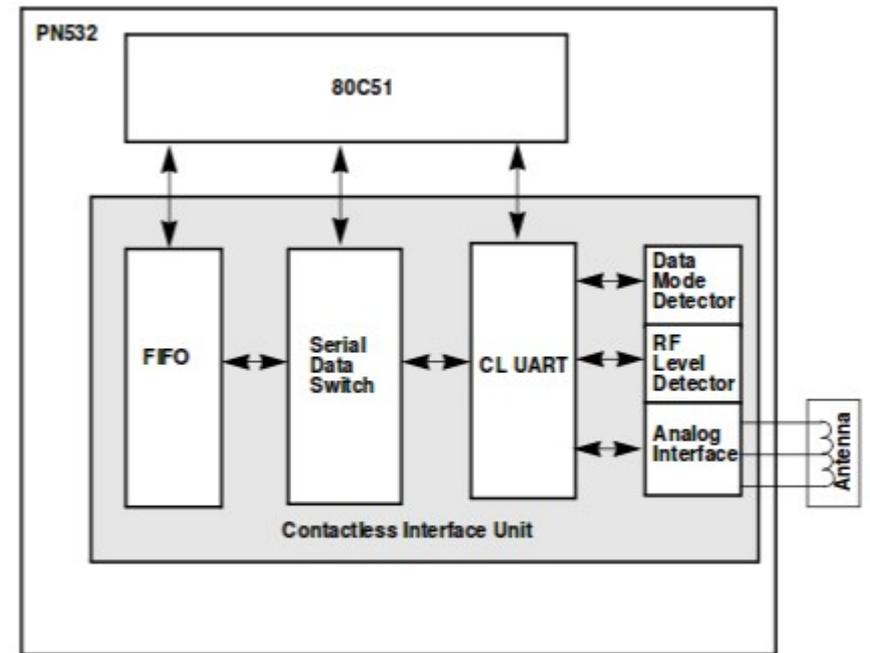
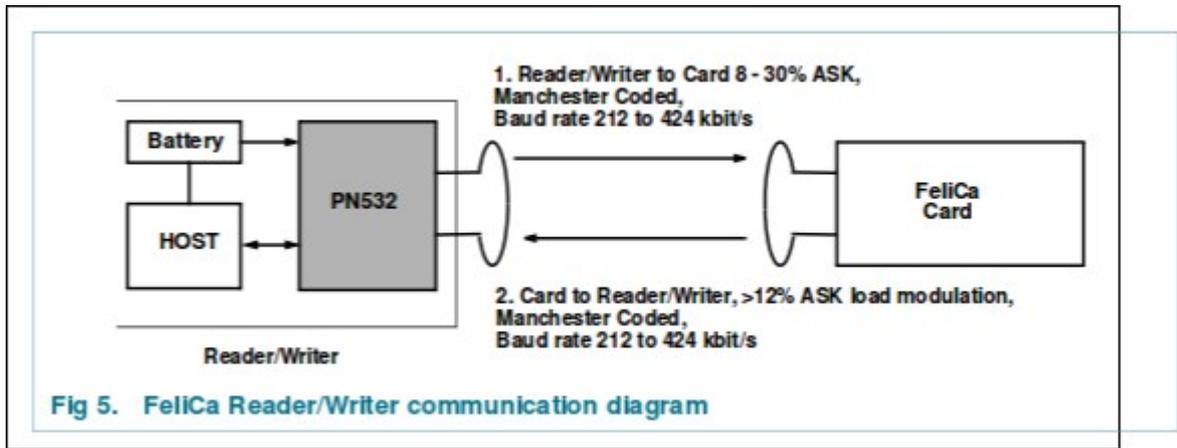


Fig 2. Simplify Contactless Interface Unit (CIU) block diagram



# Пример кода - reader

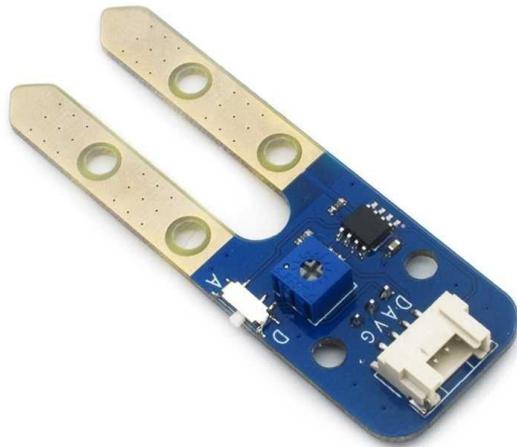
- #include "PN532\_HSU.h"
- #include "PN532.h"
- #include "NfcAdapter.h"
- 
- PN532\_HSU interface(Serial1);
- NfcAdapter nfc = NfcAdapter(interface);
- 
- void setup(void) {
- Serial.begin(115200);
- Serial.println("NDEF Reader");
- nfc.begin();
- }
- 
- void loop(void) {
- Serial.println("\nScan a NFC tag\n");
- if (nfc.tagPresent())
- {
- NfcTag tag = nfc.read();
- tag.print();
- }
- delay(5000);
- }

# Запись

- На Tag карту можно записывать данные
- Тех, кто кинется копировать транспортные карты сразу хотелось бы огорчить:
  - RFID это только технология связи, алгоритм работы в серьезных применениях основан на криптографии, ключ лежит внутри карты и нигде не светится.
  - Не надо нарушать законы...

# Датчик влаги

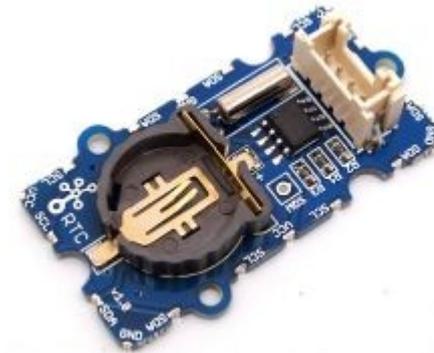
- По сути измеряет сопротивление между электродами и преобразует в UART



# Модуль RTC DS1307

Автономные часы, которые продолжают «тикать» после отключения питания микроконтроллера.

- - Часы реального времени (RTC) действуют до 2100 г. (типоразмер батареи CR1225 или CR1220)
- - 56-Byte, батарейной, энергонезависимой (NV) оперативной памяти для хранения данных
- - Последовательный интерфейс

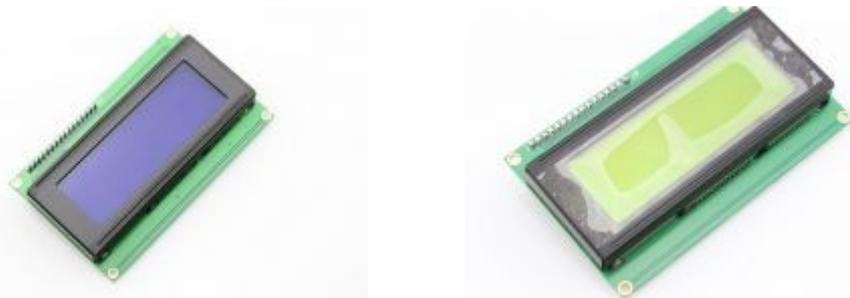


# Алфавитно-цифровой дисплей I2C

- 

- 

- 



- Идея- передаем строчку дисплею, а дальше контроллер дисплея сам разбирается что с ней делать.
- `I2C_LCD_Library`

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x20,20,4); // set the LCD address to 0x20 for a 16 chars and 2 line display
```

```
void setup()
```

```
{
```

```
  lcd.init();           // initialize the lcd
```

```
  lcd.init();
```

```
  // Print a message to the LCD.
```

```
  lcd.backlight();
```

```
  lcd.setCursor(3,0);
```

```
  lcd.print("Hello, world!");
```

```
  lcd.setCursor(2,1);
```

```
  lcd.print("Ywrobot Arduino!");
```

```
    lcd.setCursor(0,2);
```

```
  lcd.print("Arduino LCM IIC 2004");
```

```
    lcd.setCursor(2,3);
```

```
  lcd.print("Power By Ec-yuan!");
```

```
}
```

```
void loop()
```

```
{
```

```
}
```

```

//YWROBOT
//Compatible with the Arduino IDE 1.0
//Library version:1.1
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#if defined(ARDUINO) && ARDUINO >= 100
#define printByte(args) write(args);
#else
#define printByte(args) print(args,BYTE);
#endif

uint8_t bell[8] = {0x4,0xe,0xe,0xe,0x1f,0x0,0x4};
uint8_t note[8] = {0x2,0x3,0x2,0xe,0x1e,0xc,0x0};
uint8_t clock[8] = {0x0,0xe,0x15,0x17,0x11,0xe,0x0};
uint8_t heart[8] = {0x0,0xa,0x1f,0x1f,0xe,0x4,0x0};
uint8_t duck[8] = {0x0,0xc,0x1d,0xf,0xf,0x6,0x0};
uint8_t check[8] = {0x0,0x1,0x3,0x16,0x1c,0x8,0x0};
uint8_t cross[8] = {0x0,0x1b,0xe,0x4,0xe,0x1b,0x0};
uint8_t retarrow[8] = { 0x1,0x1,0x5,0x9,0x1f,0x8,0x4};

LiquidCrystal_I2C lcd(0x20,20,4); // set the LCD address to 0x20 for a 16 chars and 2 line display

void setup()
{
  lcd.init();           // initialize the lcd
  lcd.backlight();

  lcd.createChar(0, bell);
  lcd.createChar(1, note);
  lcd.createChar(2, clock);
  lcd.createChar(3, heart);
  lcd.createChar(4, duck);
  lcd.createChar(5, check);
  lcd.createChar(6, cross);
  lcd.createChar(7, retarrow);
  lcd.home();

  lcd.print("Hello world...");
  lcd.setCursor(0, 1);
  lcd.print(" i ");
  lcd.printByte(3);
  lcd.print(" arduinos!");
  delay(5000);
  displayKeyCodes();
}

// display all keycodes
void displayKeyCodes(void) {
  uint8_t i = 0;
  while (1) {
    lcd.clear();
    lcd.print("Codes 0x"); lcd.print(i, HEX);
    lcd.print("-0x"); lcd.print(++i+16, HEX);
    lcd.setCursor(0, 1);
    for (int j=0; j<16; j++) {
      lcd.printByte(++j);
    }
    i+=16;

    delay(4000);
  }
}

void loop()
{
  •
}

```

# e-Paper Panel

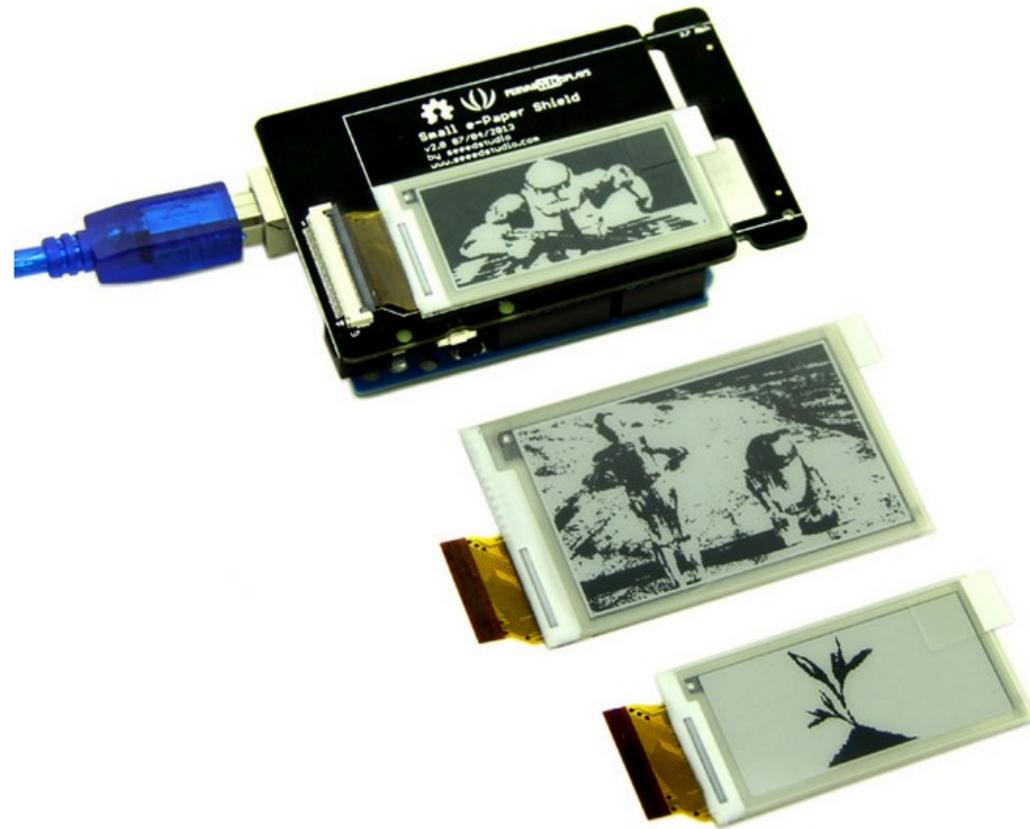
- «Электронная бумага» - элемент, который требует энергии только для обновления информации.
- Угол обзора 180°
- Изображение в отраженном свете
- В общем случае ч/б ( или несколько градаций серого)
- Неспешное обновление



2.7" e-Paper Panel

Model Name	EM027AS011	EM027AS012	EM027AS013
Outline (mm)	70.42(H) x 45.80(V) x 1.00(T)		
Resolution	264 x 176		
Contrast Ratio	10:1	7:1	7:1
Interface	SPI		
Operation Temperature	0~50°C		
Storage Temperature	-20~60°C		
Refresh Time(room temp.)	3 sec		
Pixel Pitch(mm)	0.217 x 0.217 (117dpi)		
Pin Number	40 pins(Connector STARCONN 089H40 or Compatible)		
Input Voltage	3.0 V		
FPC	0.5mm pitch with stiffener	0.5mm pitch with stiffener	0.5mm pitch <b>without</b> stiffener
FPL	V220	V110	V110

# Подключение к контроллеру в общем случае потребует драйвер



# Что почитать(источники идей)

- <http://nextable.ru/144-4-razryadnyy-7-segmentnyy-indikator-i-arduino.html>
- <http://greenoakst.blogspot.ru/2011/11/arduino-7-spi.html>
- <http://habrahabr.ru/post/129569/>
-

# Что почитать(источники идей)

- <http://nextable.ru/144-4-razryadnyy-7-segmentnyy-indikator-i-arduino.html>
- <http://greenoakst.blogspot.ru/2011/11/arduino-7-spi.html>
- <http://habrahabr.ru/post/129569/>
-