

- [Home](#)
- [About](#)
- [Video](#)

[Avislab](#)

сайт для палких паяльників

[Raspberry Pi – що це таке? »](#)

« [RFID Reader + ATMEGA + Raspberry Pi](#)

RFID считыватель + ATMEGA + Raspberry Pi

Опубліковано 26.08.2014



У меня в распоряжении появились три различных **RFID** ридера:

- настольный с интерфейсом **USB (125 КГц)**. Поддерживает EM4001, EM4100, EM4200, TK4100;
- **ZNR-A26ID** – для монтажа на стену с интерфейсом **WG26 (125 КГц)**;
- **RFID-RC522** – отдельная плата на базе микросхемы **MFRC522** с интерфейсом **SPI (13,56 МГц)**;



Последний интересен тем, что позволяет читать и писать на карточку.

Настольный RFID reader с интерфейсом USB

Подключаем ридер с **USB** интерфейсом к компьютеру с Windows. Windows видит новое оборудование и сама устанавливает драйвер. **RFID USB-ридер** воспринимается Windows как **HID**-устройство. То есть, как аналог клавиатуры. Открываем **Notepad**, подносим карточку к ридеру. Когда карточка считывается, ридер пищит, мигает зеленым светом, а в окне **Notepad** или любого поля для ввода появляется номер карты. Ридер полностью симулирует ввод номера карты с клавиатуры. Больше ничего он не умеет. Ничего сложного, ничего интересного.

RFID reader для монтажа на стену с интерфейсом WG26

Такой ридер является стандартным прибором для настенного монтажа и контроля доступа в помещение. Он герметичен, наружу выходит 6 проводов:

Красный (**Red**) – **VCC +8..15V**

Черный (**Black**) – **GND**

Зеленый (**Green**) – **Data0**

Белый (**White** 😊) – **Data1**

Серый (**Gray**) – **BEEP**

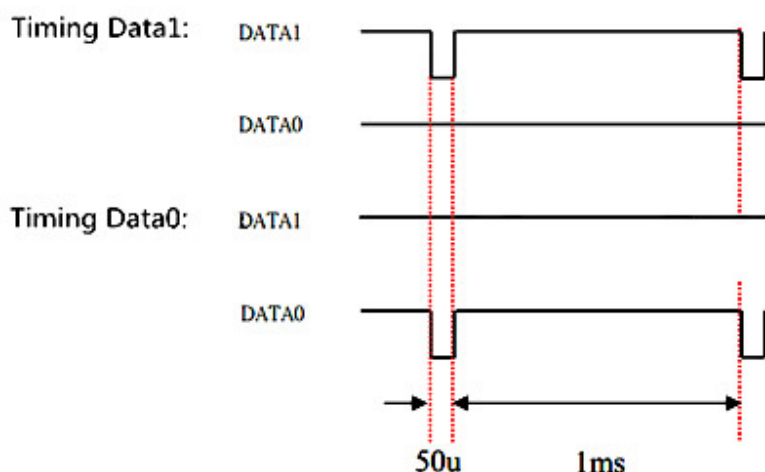
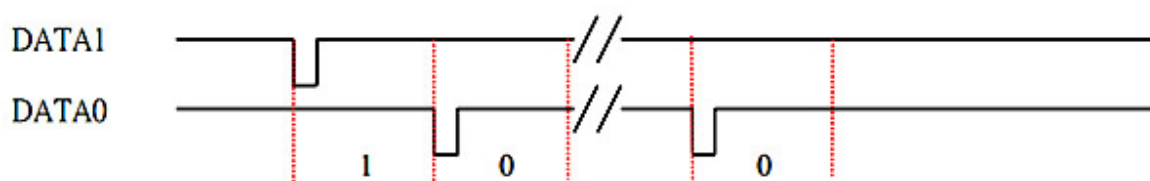
Пурпурный (**Purple**) – неизвестно, видимо **LED**.

Питание ридера 8..15В <100mA, но сигналы **Data0**, **Data1** уровня 5 В. Передача данных выполняется по протоколу **WG26**

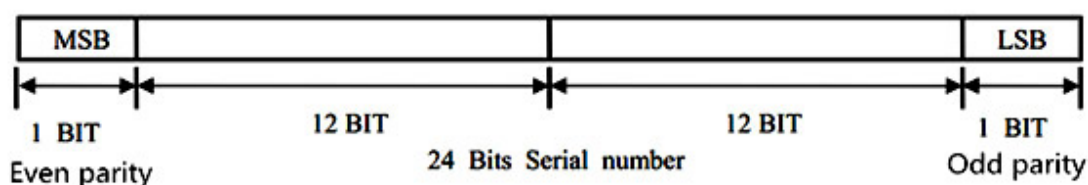
Протокол WG26

Це двопровідний протокол, у якому “0” передаються по одній лінії **Data0**, а “1” по другій – **Data1**. Рівень сигналів **Data0**, **Data1** близький до 5В. Падіння сигналу до 0 на одній з ліній означає передачу одного біта. Якщо 0 на **Data0** – це 0, якщо 0 на **Data1** – це 1. Взагалі, передається 26 біт, тому протокол і називається **WG26**. Аналогічно працює протокол **WG32**. Діаграма роботи протоколу:

Это двухпроводной протокол, в котором “0” передаются по одной линии **Data0**, а “1” по второй – **Data1**. Уровень сигналов **Data0**, **Data1** близок к 5В. Падение сигнала до 0 на одной из линий означает передачу одного бита. Если 0 на линии **Data0** – это 0, если 0 на **Data1** – это 1. Всего, передается 26 бит, поэтому протокол и называется **WG26**. Аналогично работает протокол **WG32**. Диаграмма работы протокола:



Структура данных:

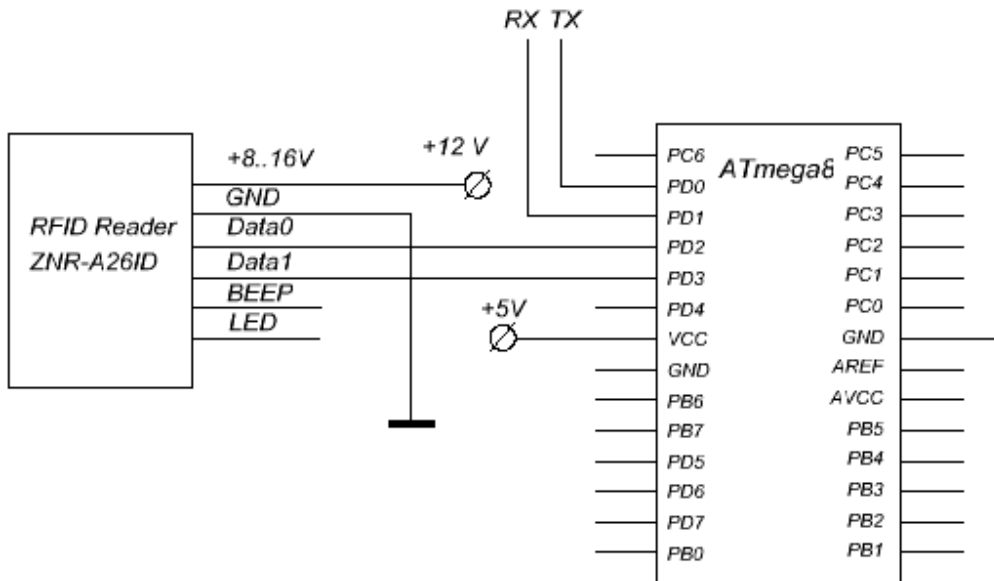


Note 1: MSB is transmitted first.

Note 2: MSB's 12Bits using even parity, LSB's 12Bits using odd parity

Подключаем RFID reader с интерфейсом WG26 к микроконтроллеру ATMEGA

Подключаем **Data0** к **INT0**, **Data1** к **INT1**. Для считывания будем использовать прерывания по спаду. Данные о считывании отправляются по **UART**. схема:



Программа для микроконтроллера:

```

1  #include <stdio.h>
2  #include <avr/interrupt.h>
3  #include "uart.h"
4
5  #define RXUBRR (F_CPU/16/9600)-1
6
7  #define BUFFER_SIZE 11
8  char buffer[BUFFER_SIZE];
9  unsigned long int card_code = 0;
10 char bit_count=0;
11
12 void convert_to_decimal(char* dst, unsigned long src)
13 {
14     int i;
15     for (i = 0; i < 10; i++)
16     {
17         dst[9 - i] = '0' + (int)(src % 10);
18         src /= 10;
19     }
20
21     dst[10] = 0;
22 }
23
24 ISR(INT0_vect) {
25     card_code = card_code << 1;
26     bit_count++;
27 }
28
29 ISR(INT1_vect) {
30     card_code = card_code << 1;
31     card_code = card_code | 0x01;
32     bit_count++;
33 }
34
35 int main( void )
36 {

```

```

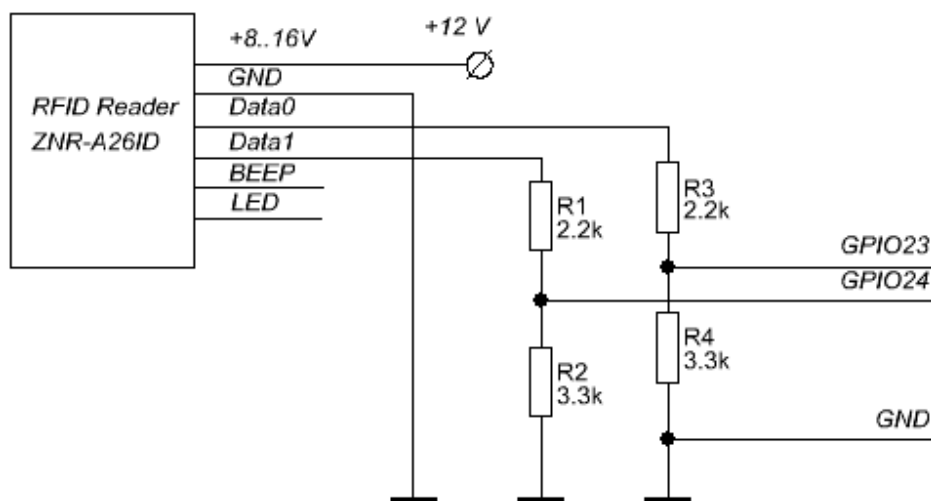
37  uart_init(RXUBRR);
38
39  MCUCR |= (1<<ISC01); // INT0 (falling edge)
40  MCUCR |= (1<<ISC11); // INT1 (falling edge)
41  GICR = ((1<<INT0)|(1<<INT1)); // Enable INT0, INT1
42
43  sei();
44  uart_puts("Started\r\n");
45
46  while(1) {
47      if (bit_count == 26) {
48          card_code = card_code >> 1;
49          card_code = card_code & 0x00ffffff;
50
51          uart_puts("Card number: ");
52
53          convert_to_decimal(buffer, card_code);
54          uart_puts(buffer);
55
56          uart_puts("\r\n");
57
58          card_code=0;
59          bit_count=0;
60      }
61  }
62  }

```

[Скачать текст программы](#)

Подключаем RFID reader с интерфейсом WG26 к Raspberry PI

Raspberry Pi не может работать с уровнем входных сигналов 5В. Говорят, что это может вывести **Raspberry Pi** из строя. Я не проверял. Я использовал простую схему для согласования уровней сигналов. Схема подключения FRID reader WG26 + Raspberry Pi:



Пример для считывания на **Python**:

```

1  import RPi.GPIO as GPIO
2
3  _DATA0PIN = 23
4  _DATA1PIN = 24
5
6  GPIO.setmode(GPIO.BCM)
7  GPIO.setup(_DATA0PIN , GPIO.IN, pull_up_down=GPIO.PUD_UP)

```

```
8  GPIO.setup(_DATA1PIN , GPIO.IN, pull_up_down=GPIO.PUD_UP)
9
10 cardnumber = 0
11 bit_count = 0
12 for x in range(0, 26):
13     data0 = GPIO.input(_DATA0PIN )
14     data1 = GPIO.input(_DATA1PIN )
15     while ( (data0 == 1) and (data1 == 1) ):
16         data0 = GPIO.input(_DATA0PIN )
17         data1 = GPIO.input(_DATA1PIN )
18
19     cardnumber = cardnumber << 1
20
21     if data1 == 1:
22         cardnumber = cardnumber | 0x01
23     bit_count = bit_count + 1
24
25     while ( (data0 == 0) or (data1 == 0) ):
26         data0 = GPIO.input(_DATA0PIN )
27         data1 = GPIO.input(_DATA1PIN )
28
29 cardnumber = cardnumber >> 1
30 cardnumber = cardnumber & 0x00ffffff
31 print cardnumber
```

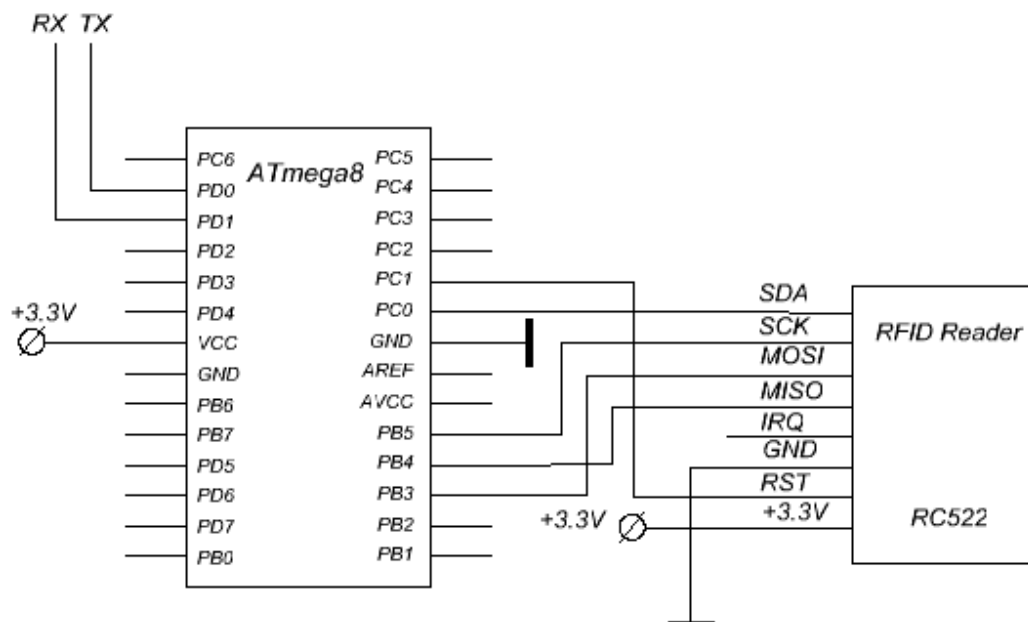
В этих примерах не используются **MSB** и **LSB** биты. Если у Вас будет такая необходимость, Вы можете доработать этот код.

Плата RFID reader на базе микросхемы MFRC522 с интерфейсом SPI

Цей зчитувач досить популярний і цікавий тим, що використовує найбільш популярну робочу частоту **13,56 МГц**. До того ж має дуже широкий функціонал, який дозволяє реалізувати різні задачі у яких задіяні можливості запису та зчитування інформації з картки. Якщо уважно почитати документацію до микросхеми MFRC522, на якій базується ця плата, стане ясно, що можливості микросхеми дуже широкі. [Скачати документацію до микросхеми MFRC522](#).

Этот считыватель достаточно популярен и интересен тем, что использует наиболее популярную рабочую частоту **13,56 МГц**. К тому же имеет очень широкий функционал, который позволяет реализовать различные задачи в которых задействованы возможности записи и считывания информации с карты. Если внимательно почитать документацию к микросхеме **MFRC522**, на которой базируется эта плата, станет ясно, что возможности микросхемы очень широкие и позволяют расширить спектр использования технологии **RFID** за пределы привычных границ. [Скачать документацию к микросхеме MFRC522](#).

Подключаем RFID reader MFRC522 к микроконтроллеру ATMEGA



RFID-RC522 Atmega8

SDA	PC0
SCK	SCK
MOSI	MOSI
MISO	MISO
IRQ	NONE
GND	GND
RST	PC1
3.3V	3.3V

Я не смог найти готовую библиотеку для **GCC**, поэтому взял библиотеку для **Arduino** и переписал. Если найдете ошибки, извините. Программа для МК:

```

1  #include <stdio.h>
2  #include <util/delay.h>
3  #include "uart.h"
4
5  #include "MFRC522.h"
6
7  #define RXUBRR (F_CPU/16/9600)-1
8
9  #define BUFFER_SIZE 2
10 char buffer[BUFFER_SIZE];
11
12 int main( void )
13 {
14     uchar card_num[5];
15     uchar i;
16
17     sei();
18
19     uart_init(RXUBRR);
20     MFRC522_Init();
21
22     uart_puts("RC522 started\r\n");
23
24     while(1) {

```

```

25     _delay_ms(500);
26     if ( MFRC522_Request( PICC_REQIDL, card_num ) == MI_OK ) {
27         if ( MFRC522_Anticoll( card_num ) == MI_OK ) {
28             uart_puts("Card number:\r\n");
29             for (i = 0; i < 5; i++ ) {
30                 sprintf(buffer, "%x ", card_num[i]);
31                 uart_puts(buffer);
32             }
33             uart_puts("\r\n");
34         }
35     }
36 }
37 }

```

[Скачать пример с библиотекой](#)

Подключаем RFID reader MFRC522 к Raspberry PI

Схема підключення така:

Name Pin # Pin name

SDA	24	GPIO8
SCK	23	GPIO11
MOSI	19	GPIO10
MISO	21	GPIO9
IRQ	None	None
GND	Any	Any Ground
RST	22	GPIO25
3.3V	1	3V3

Роспиновка Raspberry Pi:



Проверяем файл `/etc/modprobe.d/raspi-blacklist.conf`

Строки, в имеющие отношение к **spi**, должны быть закомментированы. Пример файла: `/etc/modprobe.d/raspi-blacklist.conf`:

```

#blacklist spi-bcm2708
blacklist i2c-bcm2708

```

Обновляем систему:

```
apt-get update
```

Устанавливаем python-dev:

```
apt-get install python-dev
```

```
git clone https://github.com/lthiery/SPI-Py
```

```
cd SPI-Py
```

```
sudo python setup.py install
```

```
cd ..
```

Устанавливаем MFRC522-python:

git clone <https://github.com/mxgwx/MFRC522-python>

Запускаем пример:

cd MFRC522-python

python Read.py

Подносим карточку и радуемся работе устройства! Там же Вы найдете скрипт **Write.py**, демонстрирующий запись и считывание данных с карты.

Успехов!

Читайте начало: [RFID. Что это такое?](#)

Рубрика: [Корисно знати](#), [Схеми і прошивки Raspberry Pi – що це таке?](#) »

« [RFID Reader + ATMEGA + Raspberry Pi](#)

Хочешь стать тестировщиком?

Теория + практика. Обучение у профи. Стань тестировщиком с "ШАГ"!

Перейти в itstep.by/курсы-тестировщиков



Translate

Выбрать язык ▼

Технологии  Переводчик

Категорії

- [Brushless Motors](#) (14)
- [Raspberry Pi](#) (53)
- [STM32](#) (53)
- [Довідники](#) (5)
- [Книги про електроніку](#) (18)
- [Корисно знати](#) (80)
- [Новини](#) (9)
- [Початківцям](#) (77)
- [Схеми і прошивки](#) (126)

Недавні записи

- [Keil uVision5 – IDE для STM32](#)
- [IAR Workbench – IDE для STM32](#)
- [DFPlayer Mini – MP3 модуль з портом UART](#)
- [ST-Link Програмувальники для STM8 / STM32](#)
- [23. STM32. Програмування STM32F103. Option bytes](#)

Останні коментарі

- Vit до [7. STM32. Програмування STM32F103. ADC](#)
- andre до [7. STM32. Програмування STM32F103. ADC](#)
- Vit до [7. STM32. Програмування STM32F103. ADC](#)
- andre до [BME280 – датчик атмосферного тиску з гігрометром](#)
- Igor до [BME280 – датчик атмосферного тиску з гігрометром](#)



Архіви

Архіви

Tags

[ATMega](#) [AVR](#) [battery](#) [BLDC](#) [bmp](#) [BMP085](#) [books](#) [Brushless](#) [DC-DC](#) [DHT11](#) [displays](#) [DMA](#) [EB-500](#) [Examples](#) [EXTI](#) [GPIO](#) [GPS](#) [I2C](#) [LCD](#) [LED](#) [meteo](#) [motor](#) [MPU-6050](#) [news](#) [NVIC](#) [programmator](#) [purchase](#) [PWM](#) [Python](#) [Raspberry](#) [RFID](#) [RS-232](#) [RTC](#) [sensors](#) [servo](#) [SMD](#) [solar](#) [ST-Link](#) [STM32](#) [TIM](#) [Timer](#) [UART](#) [USB](#) [vario](#) [WiFi](#)

© 2011-2017 Андрій Корягін, Кременчук, Україна

Powered by [WordPress](#) and [WordPress Theme](#) created with Artisteer.