



УДК 004.657

## РАЗРАБОТКА СИСТЕМЫ ХРАНЕНИЯ И ПЕРЕДАЧИ МЕДИЦИНСКИХ ДАННЫХ НА ОСНОВЕ RFID ТЕХНОЛОГИИ

Ксения Опришко<sup>1</sup>, Всеволод Новиков<sup>1</sup>

<sup>1</sup> Херсонский национальный технический университет

Адрес для переписки: Ксения Опришко, студентка

Место работы: Херсонский национальный технический университет, Херсон, Украина

Email: [kseniaoprishko@mail.ru](mailto:kseniaoprishko@mail.ru)

**Аннотация.** Работа посвящена разработке системы хранения и передачи медицинских данных на основе RFID технологии. Как аналога действующих систем хранения информации на бумажном носителе или жестком диске.

**Ключевые слова:** Arduino, RFID, медицинские данные

**Введение.** С началом реформирования медицины на Украине возникает вопрос о замене старых систем хранения медицинских данных с бумажных носителей на электронные. Однако предпочтение отдаётся либо хранению данных на жестком диске, либо на облачном сервере, что несет в себе некоторые неудобства в случае отсутствия сети/интернет подключения в отдельном подразделении больницы. Так же возникают проблемы в случае взаимодействия пациента с частными медицинскими учреждениями, которые не будут иметь доступа к базе данных больницы где зарегистрирован пациент. Соответственно необходимо разработать мобильную систему хранения данных которая дополняла бы существующую систему, не исключая/заменяя ее.

**Материалы и методы исследования.** Существует множество вариантов электронных носителей информации, работающих на разных технологиях передачи данных по типу NFC, RFID, устройства с технологией Thunderbolt 3 с USB-C разъемом и т.д. Однако для удобного взаимодействия с различными типами устройств необходимо разработать систему с беспроводной передачей данных, которая кроме того будет отвечать требованиям безопасности данных. Для нашего устройства мы воспользуемся технологией RFID в виду того, что метку можно сделать как для считывания на малых расстояниях [1] (чтобы считывать к примеру, личные данные медицинской карты пациента) и средняя/дальняя идентификация (к примеру, для регистрации/фиксации/учета пациентов в различных подразделениях медицинского учреждения, учет и доступ к определенным

услугам за счет стационарных считывателей). Сама технология может использоваться для безопасного хранения медицинских данных так как данная технология показала свою эффективную безопасность при использовании в банковской сфере [2].

В качестве аппаратной платформы для создания нашей системы мы воспользуемся платформой Arduino ввиду ее универсальности и сравнительно более низкой себестоимостью по сравнению с готовыми решениями считывателей радиометок. В качестве модуля считывания/записи данных мы воспользуемся модулем RC552 [3].

Для начала проведем компьютерное моделирование принципиальной схемы в ISIS Proteus(Рис.1).

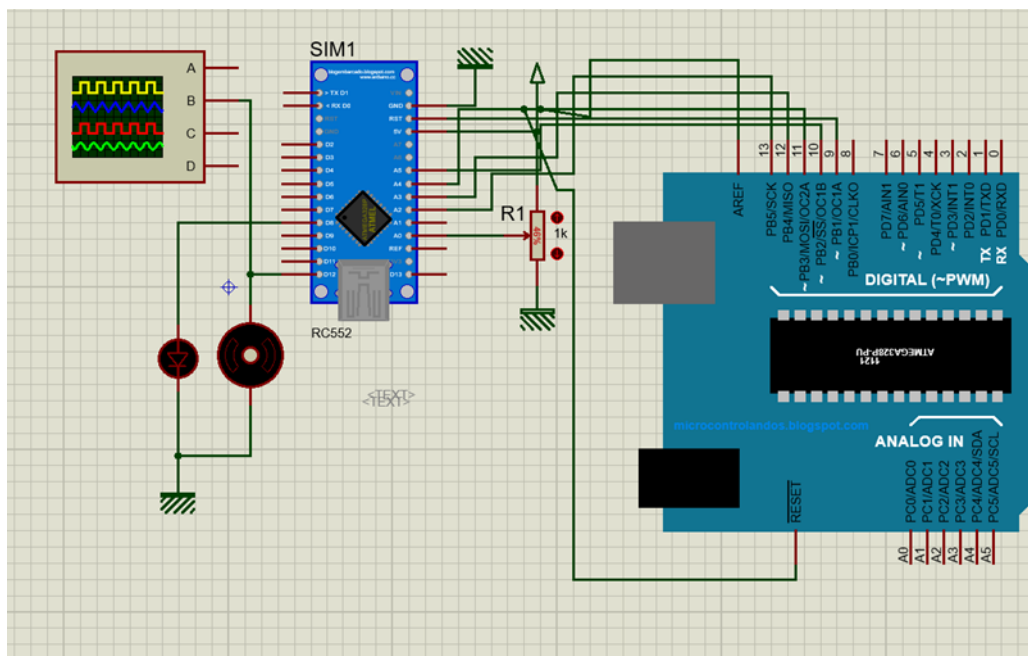


Рис. 1. Моделирование принципиальной схемы устройства

Затем проводим сборку нашего устройства (Рис.2.)

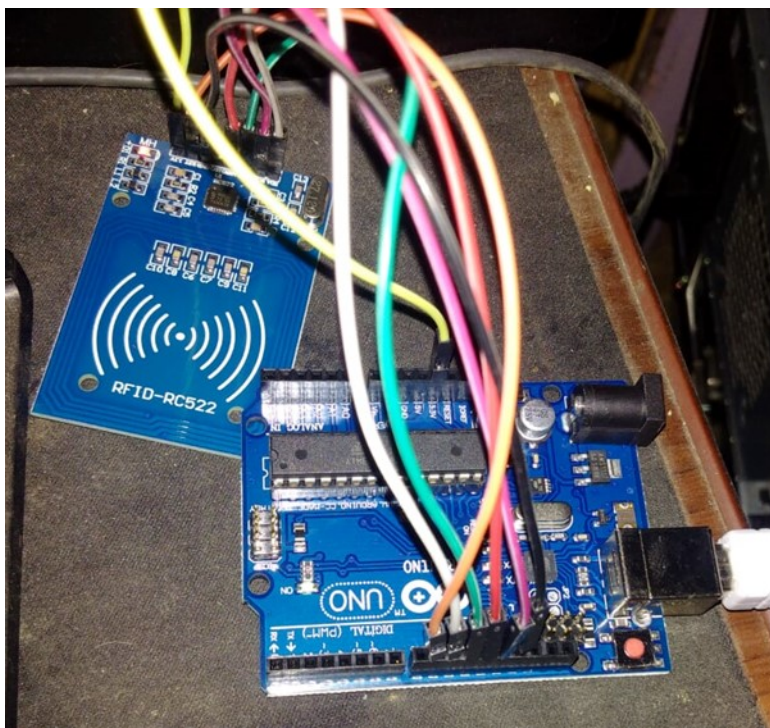


Рис. 2. Электронные компоненты устройства в сборе

Устройство будет запитано через USB ввиду того, что здесь используется стабилизация напряжения.

Следующим этапом является программная часть нашей системы. На метку мы можем записать до 512 кб данных [4-5]. В качестве номера карты пациента будет использоваться UID метки.

Проверяем и загружаем скетч на платформу (Рис. 3.)

Arduino | Arduino 1.8.1

Файл Правка Скетч Инструменты Помощь

```

Dumpinfo $
/*
 * -----
 * Typical pin layout used:
 * -----
 *
 * MFR522 Arduino Arduino Arduino Arduino Arduino
 * Reader/PCD Uno/101 Mega Nano v3 Leonardo/Micro Pro Micro
 * Signal Pin Pin Pin Pin Pin Pin Pin
 * -----
 * RST/Reset RST 9 5 D9 RESET/ICSP-5 RST
 * SPI SS SDA(SS) 10 53 D10 10 10
 * SPI MOSI MOSI 11 / ICSP-4 51 D11 ICSP-4 16
 * SPI MISO MISO 12 / ICSP-1 50 D12 ICSP-1 14
 * SPI SCK SCK 13 / ICSP-3 52 D13 ICSP-3 15
 */

#include <SPI.h>
#include <MFR522.h>

constexpr uint8_t RST_PIN = 9; // Configurable, see typical pin layout above
constexpr uint8_t SS_PIN = 10; // Configurable, see typical pin layout above

MFR522 mfrc522(SS_PIN, RST_PIN); // Create MFR522 instance

void setup() {
  Serial.begin(9600); // Initialize serial communications with the PC
  while (!Serial); // Do nothing if no serial port is opened (added for Arduinos based on ATMEGA32U4)
  SPI.begin(); // Init SPI bus
  mfrc522.PCD_Init(); // Init MFR522
  mfrc522.PCD_DumpVersionToSerial(); // Show details of PCD - MFR522 Card Reader details
  Serial.println(F("Scan PICC to see UID, SAK, type, and data blocks..."));
}

void loop() {
  // Look for new cards
  if (!mfrc522.PICC_IsNewCardPresent()) {
    return;
  }

  // Select one of the cards
  if (!mfrc522.PICC_ReadCardSerial()) {
    return;
  }

  // Dump debug info about the card; PICC_HaltA() is automatically called
  mfrc522.PICC_DumpToSerial(&mfrc522.uid);
}

```

Компиляция завершена

Скетч использует 8230 байт (25%) памяти устройства. Всего доступно 32256 байт.  
Глобальные переменные используют 286 байт (13%) динамической памяти, оставляя 1762 байт для локальных переменных. Максимум: 2048 байт.

Рис.3. Результаты компиляция скетча

Проверяем возможность считать/записать данные с метки (Рис. 4.)

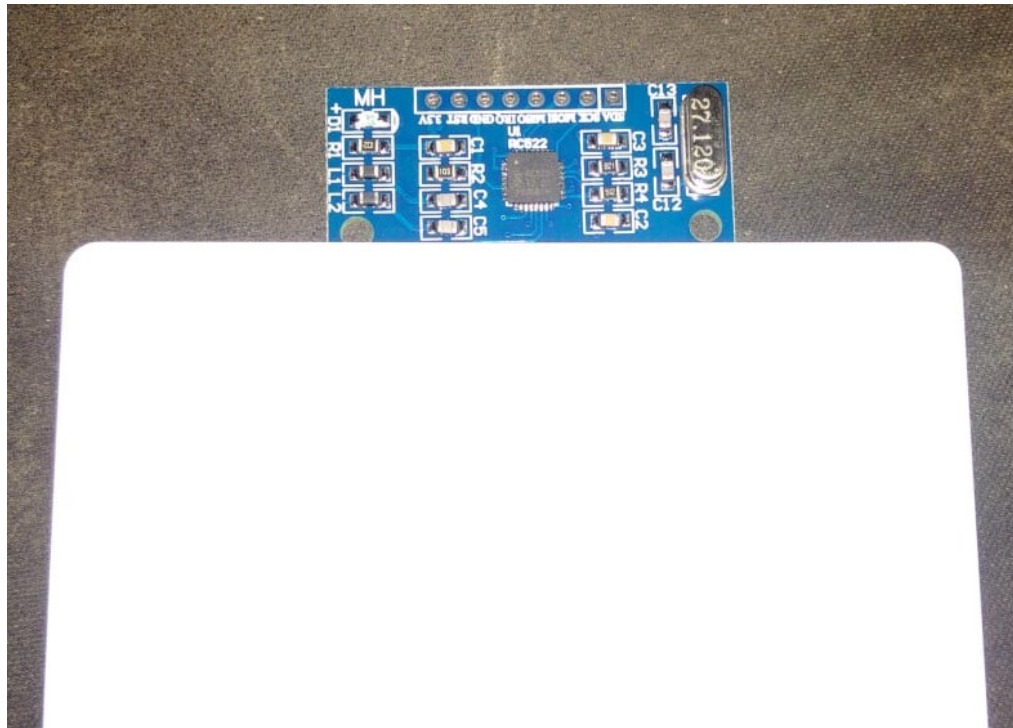


Рис.4. Размещение RFID метки на считывающем модуле RC552

При мониторинге последовательного порта мы получаем данные (Рис. 5.)

```

COM8 (Arduino/Genuino Uno)

Firmware Version: 0x12 = (unknown)
Scan PICC to see UID, SAK, type, and data blocks...
Card UID: 34 EF 0F B1 D1 5C 96
Card SAK: 00
PICC type: MIFARE Ultralight or Ultralight C
Page 0 1 2 3
0 34 EF 0F 5C
1 B1 D1 5C 96
2 AA 00 70 08
3 00 00 00 00
4 45 DB 40 03
5 18 D3 3A 00
6 22 20 56 26
7 05 58 C0 01
8 2F 03 FE 40
9 00 00 00 00
10 94 F7 CC C0
11 22 20 56 26
12 05 58 C0 01
13 2F 03 FE 40
14 00 00 00 00
15 94 F7 CC C0

```

Рис. 5. Получение данных с RFID метки

На последнем этапе клонируем данные с готовой RFID метки на пустую, для проверки возможности записи данных (Рис.6.)

```

COM8 (Arduino/Genuine Uno)

Firmware Version: Ox12 * (unknown)
Scan PICC to see UID, SAK, type, and data blocks...
Card UID: F0 B5 DF 2B
Card SAK: 08
PICC type: MIFARE 1KB
Sector Block 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 AccessBits
15 63 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
62 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
61 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
59 00 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
58 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
57 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
56 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
14 55 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
54 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
53 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
52 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
12 51 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
49 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
48 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11 47 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
46 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
45 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
44 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
10 43 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
42 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
41 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
9 40 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
39 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
38 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
37 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
36 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
8 35 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
34 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
33 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
7 31 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
29 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
28 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
6 27 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
26 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
25 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
24 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
5 23 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
22 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
21 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
4 19 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
17 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
3 16 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
15 00 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]

```

Рис. 6. Полученные данные с RFID метки после клонирования другой метки

После извлечения метки данные пропадают с последовательного порта (Рис. 7)

```

Card UID: F0 B5 DF 2B
Card SAK: 08
PICC type: MIFARE 1KB
Sector Block 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 AccessBits
15 63 PCD_Authenticate() failed: Timeout in communication.
14 59 PCD_Authenticate() failed: Timeout in communication.
13 55 PCD_Authenticate() failed: Timeout in communication.
12 51 PCD_Authenticate() failed: Timeout in communication.
11 47 PCD_Authenticate() failed: Timeout in communication.
10 43 PCD_Authenticate() failed: Timeout in communication.
9 39 PCD_Authenticate() failed: Timeout in communication.
8 35 PCD_Authenticate() failed: Timeout in communication.
7 31 PCD_Authenticate() failed: Timeout in communication.
6 27 PCD_Authenticate() failed: Timeout in communication.
5 23 PCD_Authenticate() failed: Timeout in communication.
4 19 PCD_Authenticate() failed: Timeout in communication.
3 15 PCD_Authenticate() failed: Timeout in communication.
2 11 PCD_Authenticate() failed: Timeout in communication.
1 7 PCD_Authenticate() failed: Timeout in communication.
0 3 PCD_Authenticate() failed: Timeout in communication.

```

Рис. 7. Данные с последовательного порта после извлечения RFID метки

**Результаты исследования.** В результате проведенного исследования мы собрали устройство считывания цифровых данных на базе технологии RFID. Которое возможно использовать в медицинских учреждениях, оно обладает сравнительно низкой себестоимостью и определенное защищенностью данных.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ:**

1. Историография радиочастотной идентификации (RFID)-российские корни // Современные наукоемкие технологии, Бондаревский А.С., Золотов Р.В. 2010. - №8, - стр. 11-15
2. RFID – PROTOKOLLE, VORLESUNG RFID SYSTEMS, MICHAEL GEBHART, TU PRESS, LONDON 2011
3. Levis P., Madden S., Polastre J. and dr. «TinyOS: An operating system for wireless sensor networks» // W. Weber, J.M. Rabaey, E. Aarts (Eds.) // In Ambient Intelligence. — New York, NY: Springer-Verlag, 2005. — 374 p.
4. Максим Власов. RFID: 1 технология – 1000 решений: Практические примеры использования RFID в различных областях. — М.: Альпина Паблишер, 2014. — 218 с. — ISBN 978-5-9614-4879-5.
5. Клаус Финкенцеллер. Справочник по RFID. — М.: Издательский дом «Додэка-XXI», 2008. — 496 с. — ISBN 978-5-94120-151-8.