УДК 004

Эмуляция RFID меток с использованием микроконтроллеров семейства Arduino

Болтовский Гавриил Александрович
Приамурский государственный университет им. Шолом-Алейхема
Студент

Аннотация
Целью данной статьи является создание устройства, позволяющего эмулировать различные RFID метки, работающие на частоте 125 кГц. Для этого был создан специальный модуль, позволяющий передавать сигнал на необходимой частоте, а также устройство для его тестирования. Результатом исследования станет готовое устройство с подробным описанием принципов его построения.
Ключевые слова: RFID, arduino, встроенная разработка

Emulation of RFID tags using Arduino family microcontrollers

Boltovskiy Gavriil Aleksandrivich
Sholom-Aleichem Priamursky State University
Student

Abstract
The purpose of this article is to create a device that allows you to emulate various RFID tags operating at a frequency of 125 kHz. For this purpose, a special module was created that allows transmitting a signal at this frequency, as well as a device for testing it. The result of the study will be a ready-made device with a detailed description of the principles of its construction.
Keywords: RFID, arduino, embedded development

1. Введение

1.1 Актуальность исследования

Эмуляция RFID меток – это процесс, при котором одно устройство имитирует поведение другого устройства в системе, что может быть использовано для различных целей, таких как тестирование, анализ, взлом или создание новых функций. Для эмуляции RFID может быть использован микроконтроллер семейства Arduino, который позволит генерировать и передавать радиосигналы, соответствующие протоколам и стандартам RFID.

Это способствует изучению основ технологии RFID, ее принципов работы, протоколов и стандартов. Позволяет лучше понять возможности и ограничения технологии, а также развить навыки анализа и тестирования систем RFID и обучаться различным алгоритмам, связанным с RFID, таким как кодирование, декодирование, шифрование, аутентификация и т.д.

1.2 Обзор исследований

С. В. Горельченко, рассмотрел принципы функционирования метода радиочастотной идентификации, каналы утечки информации, приводящие к нарушению конфиденциальности данных, хранящихся на RFID - метках и считывателях, а также средства, способные привести к выведению из строя систем радиочастотной идентификации[1]. Также можно использовать RFID метки для оптимизации работы предприятия, что в своих статьях описывают А. А. Мертинян [2] и А. И. Багиров [3]. Г. А. Боев описал процесс разработки электронного замка для лабораторного комплекса [4].

1.3 Цель исследования

Создать устройство, позволяющее эмулировать RFID метки, работающие на частоте 125 гигагерц.

1.4 Постановка задачи

На первом этапе разрабатывался модуль, посылающий сигнал. Затем было собрано само устройство. И конечной задачей являлось написание прошивки для работы.

2. Методы и результаты исследования

Для создания данного устройства необходим собственный модуль, позволяющий транслировать сигнал. Сборка данного модуля происходит по схеме. Схема и внешний вид модуля представлен на рисунке (рис. 1).

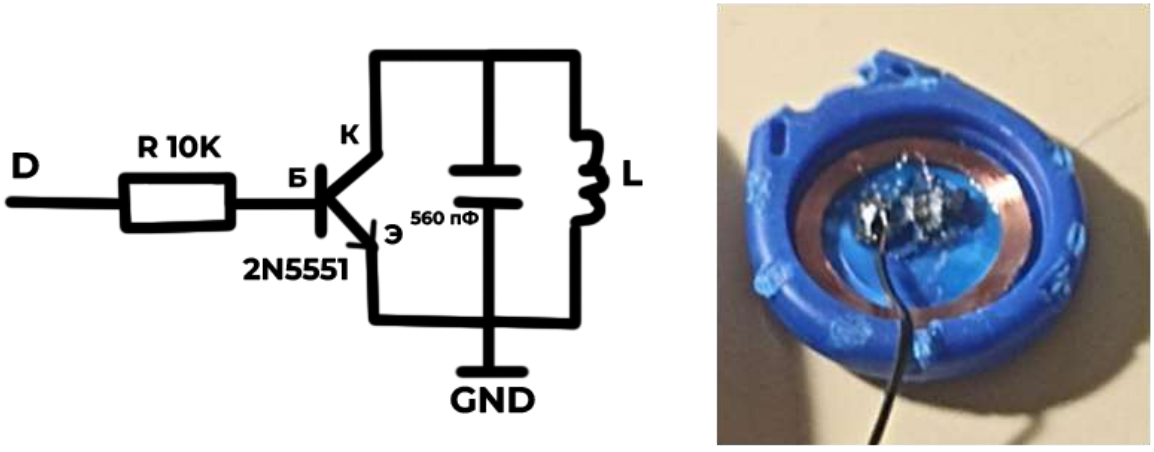

Рисунок 1 – Схема и внешний вид модуля

Для модуля требуется конденсатор ёмкостью 560 пикофарад и n-p-n транзистор. В реализации модуля, используется 2 конденсатора, подключённых параллельно, а также транзистор 2N5551. Для антенны используется брелок с другой RFID меткой, работающей на частоте 125 кГц. Сборка осуществлялась навесным монтажом. Полученный модуль имеет два выхода: земля и сигнальный контакт.

Для проверки работоспособности модуля можно использовать проверочную прошивку (рис. 2).

```
#define ANTENNA 2
#define CARD_ID 0x00000BDDCC

volatile int bit_counter=0;
volatile int byte_counter=0;
volatile int half=0;

uint8_t data[8];

void data_card_ul() {
  uint64_t card_id = (uint64_t)CARD_ID;
  uint64_t data_card_ul = (uint64_t)0x1FFF; //first 9 bit as 1
  int32_t i;
  uint8_t tmp_nybble;
  uint8_t column_parity_bits = 0;
  for (i = 9; i >= 0; i--) { //5 bytes = 10 nybbles
    tmp_nybble = (uint8_t) (0x0f & (card_id >> i*4));
    data_card_ul = (data_card_ul << 4) | tmp_nybble;
    data_card_ul = (data_card_ul << 1) | ((tmp_nybble >> 3 & 0x01) ^ (tmp_nybble >> 2 & 0x01) ^\
      (tmp_nybble >> 1 & 0x01) ^ (tmp_nybble  & 0x01));
    column_parity_bits ^= tmp_nybble;
  }
  data_card_ul = (data_card_ul << 4) | column_parity_bits;
  data_card_ul = (data_card_ul << 1); //1 stop bit = 0
  for (i = 0; i < 8; i++) {
    data[i] = (uint8_t)(0xFF & (data_card_ul >> (7 - i) * 8));
  }
}

void setupTimer1() {
  noInterrupts(); // Clear registers
  TCCR1A = 0;
  TCCR1B = 0;
  TCNT1 = 0;
  OCR1A = 4095;
  TCCR1B |= (1 << CS10);
  TIMSK1 |= (1 << OCIE1A);
  interrupts();
}

void setup() {
  pinMode(ANTENNA, OUTPUT);
  data_card_ul();
  setupTimer1();
}

void loop() {
}

ISR(TIMER1_COMPA_vect) {
      TCNT1=0;
    if (((data[byte_counter] << bit_counter)&0x80)==0x00) {
        if (half==0) digitalWrite(ANTENNA, LOW);
        if (half==1) digitalWrite(ANTENNA, HIGH);
    }
    else {
        if (half==0) digitalWrite(ANTENNA, HIGH);
        if (half==1) digitalWrite(ANTENNA, LOW);
    }

    half++;
    if (half==2) {
        half=0;
        bit_counter++;
        if (bit_counter==8) {
            bit_counter=0;
            byte_counter=(byte_counter+1)%8;
        }
    }
}
```

Рисунок 2 – Тестовая прошивка

В тестовой прошивке указывается код эмулируемой RFID метки в шестнадцатеричном формате (вторая строка). Считать код можно при помощи модуля RDM6300. Схема подключения модуля приведена на рисунке (рис. 3).

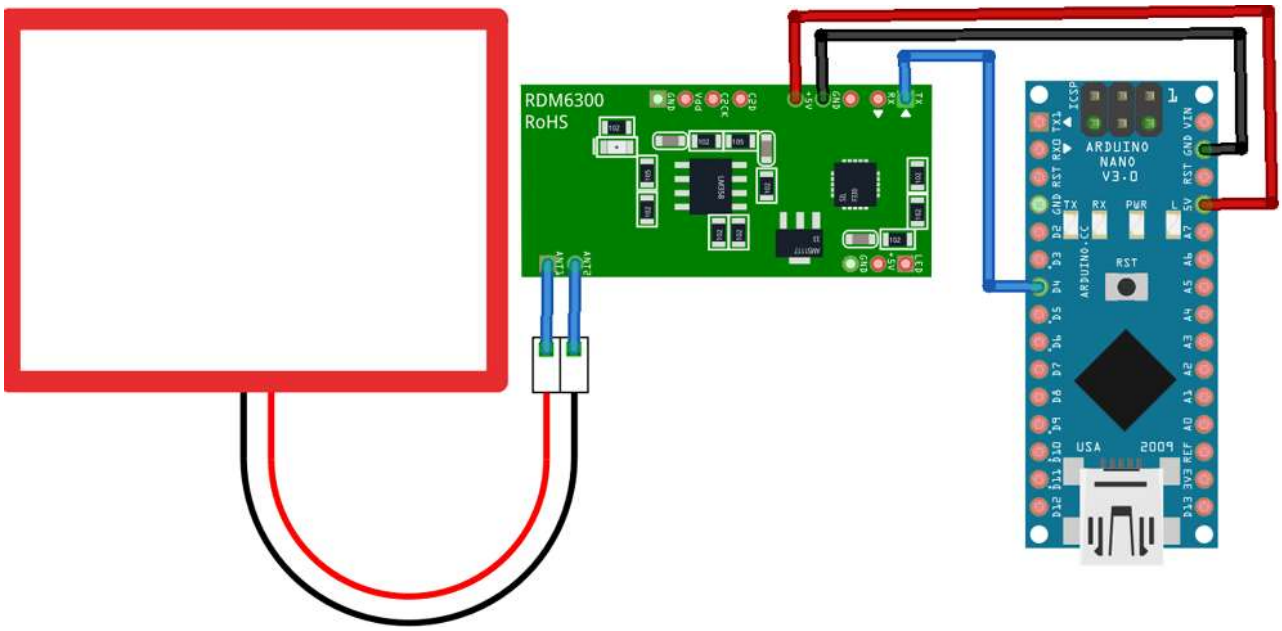

Рисунок 3 – Схема подключения модуля RDM6300

Собранное устройство с подключённым модулем RDM600 прошивается прошивкой, которую можно найти в GitHub репозитории [5].

Прошивка работает следующим образом: в зону действия антенны считывателя подносится RFID метка, а в мониторе порта появляется соответствующий ей код в шестнадцатеричном формате.

На данном этапе становится возможным оценка работоспособности модуля, а также дальности его действия.

Убедившись в работоспособности модуля, можно приступать к сборке устройства. Финальная версия эмулятора должна содержать элементы управления, источник питания, дисплей.

В качестве элементов управления выбраны стоп-кнопки. Источник питания собран из модуля зарядки TP40561A и аккумулятора, выдающего напряжение 4,7 вольт. В проекте используется OLED дисплей разрешением 128x64 пикселей. Схема собранного устройства и его внешний вид представлен на рисунке (рис. 4).

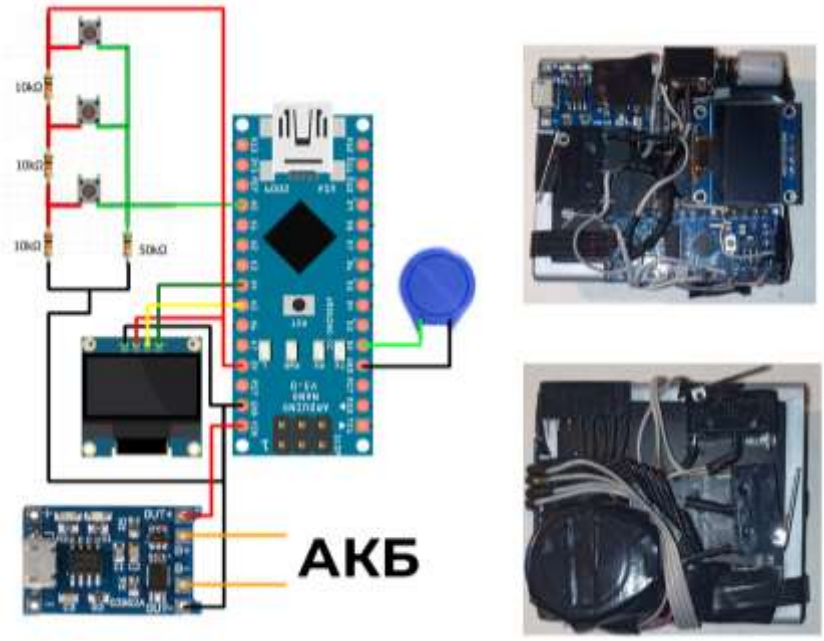

Рисунок 4 – Схема и внешний вид устройства

После сборки устройства для него необходимо написать прошивку. В проекте используются библиотеки EncButton [6], AnalogKey, для обработки нажатий на клавиши и библиотека U8glib [7] для работы с OLED дисплеем.

В начале прошивки (рис. 5) указываются значения для каждой кнопки, их можно получить через функцию analogRead(). Инициализируется дисплей, а также прописываются коды для каждой эмулируемой RFID метки (cardIDs). В массиве menuItems указываются названия для каждой опции в меню.

```
#include <EncButton.h>
EncButton<EB_TICK, VIRT_BTN> UP_BTN;
EncButton<EB_TICK, VIRT_BTN> SELECT_BTN;
EncButton<EB_TICK, VIRT_BTN> DOWN_BTN;


#include <AnalogKey.h>
int16_t sigs[3] = {
  1023, 601, 299
  };

// указываем пин, количество кнопок и массив значений
AnalogKey<A0, 3, sigs> keys;
// инициализируем антенку
#define ANTENNA 2

// всё что касается ключей
uint32_t cardIDs[] = {
  0x0000000000,
  0x0000000000,
  0x0000000000,
  0x0000000000,
  0x0000000000,
  0x0000000000,
  0x0000000000,
  0x0000000000,
  0x0000000000,
  0x0000000000,
  0x0000000000,
  0x0000000000,
  0x0000000000
};
volatile int bit_counter=0;
volatile int byte_counter=0;
volatile int half=0;

uint8_t data[8];
```

Рисунок 5 – Код прошивки

Функции DrawMenu() и draw() (рис.6) отвечают за отрисовку меню на дисплей и выбор нужной опции.

```
101  void drawMenu() {
102    u8g.setFont(u8g_font_6x10);  // Шрифт для текста меню
103
104    for (int i = 0; i < MENU_ITEMS; i++) {
105      // Отображение каждого пункта меню
106      if (i == selectedMenuItem) { // Выделение выбранного пункта
107        u8g.drawBox(0, i * MENU_HEIGHT, MENU_WIDTH, MENU_HEIGHT);
108        u8g.setColorIndex(0);  // Цвет текста выбранного пункта
109      } else {
110        u8g.setColorIndex(1);  // Цвет текста остальных пунктов
111      }
112      // Отображение текста пункта меню
113      u8g.drawStr(2, (i+1) * MENU_HEIGHT - 3, menuItems[i]);
114    }
115  }
116
117
118  void draw(void) {
119    u8g.firstPage();
120    do {
121       Serial.println("""рисуем""");
122      drawMenu();  // Отображение меню
123    } while (u8g.nextPage());
124  }
```

Рисунок 6 – Функции DrawMenu() и draw()

В блоке setup() происходит вызов функции draw() для отрисовки дисплея (рис. 7), поворачивается дисплей, а так же инициализируется антенна.

```
127  void setup() {
128    u8g.setRot270(); // переворачиваем дисплей как нам нужно
129    draw(); // отрисовываем первый раз, чтобы небыло артифактов
130    Serial.begin(9600);
131    pinMode(ANTENNA, OUTPUT); // определяем антенну
132
133  }
```

Рисунок 7 – Блок setup() прошивки

В блоке loop() (рис. 8) происходит опрос кнопок, отвечающих за перемещение по меню, а при нажатии на кнопку SELECT, происходит вызов функции data_card_ul() с кодом, который соответствует выбранной опции меню.

```
void loop() {
  UP_BTN.tick(keys.status(0));
  SELECT_BTN.tick(keys.status(1));
  DOWN_BTN.tick(keys.status(2));

  if (UP_BTN.click()){
    selectedMenuItem--;
    if (selectedMenuItem < 0) {
      selectedMenuItem = MENU_ITEMS - 1;
    }
    draw();
  } else if (DOWN_BTN.click()){
    selectedMenuItem++;
    if (selectedMenuItem >= MENU_ITEMS) {
      selectedMenuItem = 0;
    }
    draw();
  }

  if (SELECT_BTN.click()){
    Serial.println(selectedMenuItem);
    data_card_ul();
    setupTimer1();
  }
}
```

Рисунок 8 – Функция loop()

Проверить работоспособность устройства можно с помощью модуля RDM6300. Схемы всех собранных устройств и прошивок для них можно найти в репозитории проекта [8].

3. Выводы

Таким образом, было создано устройство, позволяющее эмулировать RFID метки, работающие на частоте 125 кГц.

Библиографический список

1. Горельченко, С. В. Исследование технологии радиочастотной идентификации, технологии RFID и анализ системы безопасности RFID // Научно-исследовательский центр "Вектор развития". 2021. № 2. С. 347-353.
2. Мертинян, А. А. использование RFID меток для оптимизации работы предприятия // Научный электронный журнал Меридиан. 2019. № 15(33). С. 105-107.
3. Багиров, А. И. RFID-технология автоматизации склада // Научный журнал. 2020. № 5(50). С. 10-13.
4. Боев, Г. А. Разработка электронного замка для лабораторного комплекса // Modern Science. 2019. № 9-1. С. 247-250.
5. Github [Электронный ресурс]. URL: https://github.com/arduino12/rdm6300 (дата обращения: 4.07.2023)
6. Github [Электронный ресурс]. URL: https://github.com/GyverLibs/EncButton (дата обращения: 4.07.2023)
7. Github [Электронный ресурс]. URL: https://github.com/olikraus/u8glib (дата обращения: 4.07.2023)
8. Github [Электронный ресурс]. URL: https://github.com/Gavriilbolt/Ardurobo/tree/master/RFIDemu (дата обращения: 4.07.2023)