ДОДАТОК А

Графічний матеріал атестаційної роботи

Харківський національний університет радіоелектроніки
Кафедра АПОТ

Кваліфікаційна робота

Універсальний цифровий ключ на основі технології RFID

Студент групи СКСм-20-1
Пашков Дмитро Олександрович

Керівник: доцент кафедри АПОТ
Хаханова Ганна Володимирівна

Харків 2021

Мета

Метою даної роботи є розробка універсального цифрового ключа для систем контролю та управління доступом на основі технології RFID

Для досягнення поставленої мети вирішуються наступні задачі:

- аналіз математичного апарату та специфікацій технології RFID;
- проектування та реалізація апаратної частини проекту;
- проектування та реалізація програмної частини проекту;
- аналіз отриманих результатів.

Програмну частину проекту буде написано мовою програмування C.

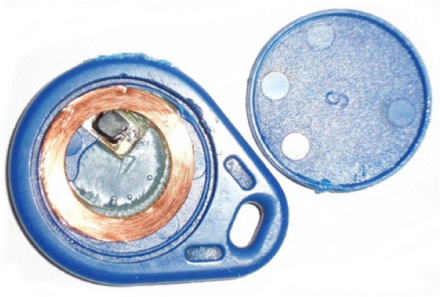
Сфера використання

Системи контролю та управління доступом

Системи контролю і управління доступом, частіше іменовані СКУД, набули широкого поширення на різних об'єктах, де необхідно забезпечити санкціонований вхід / вихід людей і в'їзд / виїзд транспорту в охоронювані зони.

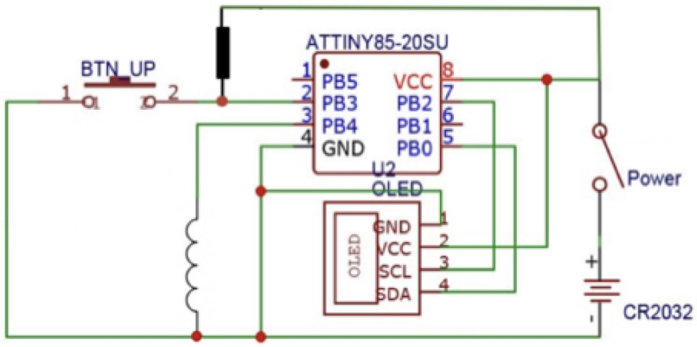
Спосіб ідентифікації

RFID-мітка складається з мікрочіпа, який зберігає інформацію для ідентифікації, і антени, за допомогою якої прилад передає і отримує дані



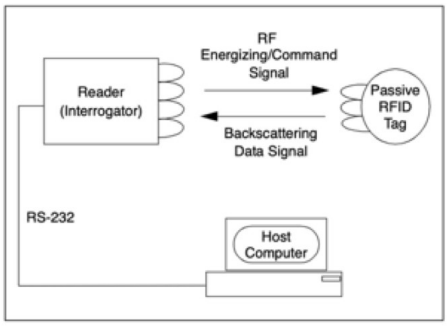
Апаратна частина

- мікроконтролер ATtiny85
- дісплейний модуль OLED 0.96"
- Катушка-антена RFID 125 KHz
- тактова кнопка
- елемент живлення



RFID технологія

RFID або радіочастотна ідентифікація – це спосіб автоматичної ідентифікації об'єктів, в якому за допомогою радіосигналів зчитуються або записуються дані, що зберігаються в так званих транспондерах, або RFID-мітках.



Стандарти RFID

Технологія RFID з моменту свого створення здобула вже більше десятка різних стандартів. Стандарти радіочастот спочатку були прийняті Міжнародною Організацією Стандартизації для діапазону LF (Low Frequency – діапазон з низькою робочою частотою) частоти в 125 кГц. Найбільш популярними вважалися ISO 11784 та ISO 11785 які працюють на частоті в 125KHz.

Також є більш специфічні стандарти, котрі працюють на більших частотах, які доходять навіть до гіга-частотного діапазону.

Критерії обрання RFID системи

Технічні параметри цих систем оптимізовані для різних сфер застосування – оформлення квитків, ідентифікація тварин, промислова автоматизація або контроль доступу. Технічні вимоги цих сфер застосування часто збігаються, а це означає, що чітка класифікація відповідних систем не є простою справою. Ситуацію ще більш ускладнює те, що за винятком кількох особливих випадків (ідентифікація тварин, смарт-карти тісного зв'язку), для систем RFID ще не існує жодних обов'язкових стандартів.

Критерії обрання RFID системи

Відстань

Необхідний діапазон застосування залежить від кількох факторів:

– точність позиції RFID мітки;

– мінімальна відстань між декількома транспондерами в практичній експлуатації;

– швидкість транспондера в зоні опитування зчитувача.

Наприклад, у програмах продажу квитків на громадський транспорт – швидкість позиціонування дуже низька, оскільки квиток наводиться до зчитувача вручну кожним окремим користувачем системи.

Мінімальна відстань між декількома транспондерами в цьому випадку відповідає відстані між двома пасажирами, що в'їжджають у транспортний засіб.

Для таких систем є оптимальний діапазон 5–10 см. Більший діапазон у цьому випадку лише спричинить проблеми, оскільки считувач може виявити одночасно декілька квитків пасажирів. Це унеможливить надійну роботу системи.

Критерії обрання RFID системи
Ємність пам'яті

Розмір мікросхеми носія даних, отже і ціновий клас – в першу чергу визначаються його обсягом пам’яті. Постійно закодовані носії даних лише для читання використовуються в чутливих до ціни масових пристроях з низькими вимогами до локальної інформації. Однак за допомогою такого носія даних можна визначити лише ідентичність об’єкта. Подальші дані зберігаються в центральній базі даних керуючого комп'ютера

Критерії обрання RFID системи
Робоча частота та вимоги до безпеки системи

Системи RFID, які використовують частоти приблизно від 100 кГц до 30 МГц, працюють за допомогою індуктивного зв’язку.
Мікрохвильові системи навпаки в діапазоні частот 2,45–5,8 ГГц підключаються за допомогою електромагнітних полів.

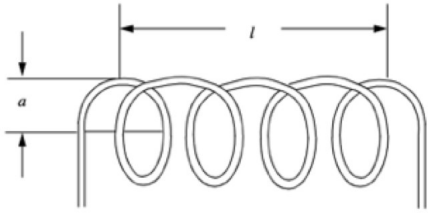
Іншим важливим фактором є чутливість до полів електромагнітних перешкод, таких як ті, які генеруються зварювальними роботами або потужними електродвигунами.

Вимоги до безпеки, які будуть накладені на запланований додаток RFID, тобто шифрування та аутентифікація, повинні бути дуже точно оцінені, щоб виключити будь-які неприємні сюрпризи на етапі впровадження. Для цього слід оцінити стимул, який система представляє потенційному зловмиснику як засіб придбання грошей або матеріальних благ шляхом маніпуляцій.

Для програм з максимальними вимогами безпеки, наприклад банківських програм з електронним гаманцем, слід використовувати тільки транспондери з мікропроцесорами

Індуктивність

Електричний струм, що протікає через провідник, створює магнітне поле. Це змінюване у часі магнітне поле здатне виробляти потік струму через інший провідник. Це змінюване у часі магнітне поле і називається індуктивністю. Індуктивність L залежить від фізичних характеристик провідника. Котушка має більшу індуктивність, ніж прямий дріт з того ж матеріалу, а котушка з більшою кількістю витків має більшу індуктивність, ніж котушка з меншою кількістю витків.



N – кількість витків у котушці;

Ф – Магнітний потік;

I – сила струму.

$$L = \frac{N\psi}{I}$$

Структура даних RFID-мітки

- 9 біт заголовку
- D00-D07 – біти ідентифікатора
- P0-P9 – контрольні біти для рядків
- D08-D39 – біти ідентифікатора клієнта
- PC0-PC1 – контрольні біти для стовпчиків
- S0 – стоп біт

1	1	1	1	1	1	1	1	1
				D00	D01	D02	D03	P0
				D04	D05	D06	D07	P1
				D08	D09	D10	D11	P2
				D12	D13	D14	D15	P3
				D16	D17	D18	D19	P4
				D20	D21	D22	D23	P5
				D24	D25	D26	D27	P6
				D28	D29	D30	D31	P7
				D32	D33	D34	D35	P8
				D36	D37	D38	D39	P9
				PC0	PC1	PC2	PC3	S0

Програмна емуляція
Форумування заголовка

Дані ключа перед відправкою формуються у змінній RFIDdata.

Функція WriteHeader сформує так званий “заголовок” даних. Для успішної комунікації між ключем та зчитувачем – перщі 9 біт сигналу мають бути встановлені в логічну одиницю

```
void WriteHeader(void)
{
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
}
```

Програмна емуляція
Запис даних

Метод WriteData – займається формуванням даних у форматі прийнятному для RFID зчитувача.

Метод формує молодші та старші значуші біти ідентифікатора клієнта (біти D00 – D07), а також код ключа RFID-картки (біти D08 – D39).

Також метод бере на себе відповідальність за прораховування та запис контрольних сум для рядків бітів (біти P0 – P9)

```
void WriteData(byte input)
{
  byte data;
  byte rowsum=0;
  for (int i=4; i>0; i--)
  {
    if ((input& 1<<i-1) ==0)
       {
         data=0;
    }
     else
    {
       data=1;
       rowsum++;
       colsum[i-1]++;
}

    RFIDdata[datapointer++]= data;
  }
 if ((rowsum%2)==0)
 {
  RFIDdata[datapointer++]=0;
 }
 else
 {
   RFIDdata[datapointer++]=1;
 }
}
```

Програмна емуляція

Формування контрольної суми

Метод WriteChecksum – займається не лише формуванням бітів парності для чотирьох стовпчиків даних.

В самому кінці метода, після ітеративного циклу for, в сформований масив даних для відправки на считува добавляється останній бітдорівнює логічному нулю.

Цей біт умовно називається “стоп” бітом та сигналізує RFID считовачу про кінець передачі даних.

```
void WriteChecksum(void)
{
 byte data;
 byte rowsum=0;
 for (int i=4; i>0; i--)
  {

  if ((colsum[i-1]%2) ==0)
  {
    RFIDdata[datapointer++]=0;
     #ifdef SERIALDEBUG
      Serial.print((int)0);
     #endif
    }

  else
  {
    RFIDdata[datapointer++]=1;
     #ifdef SERIALDEBUG
     Serial.print((int) 1);
     #endif
   }
 }

 // write the stop bit
 RFIDdata[datapointer++]=0;

 #ifdef SERIALDEBUG
   Serial.print((int)0);
    #endif
}
```

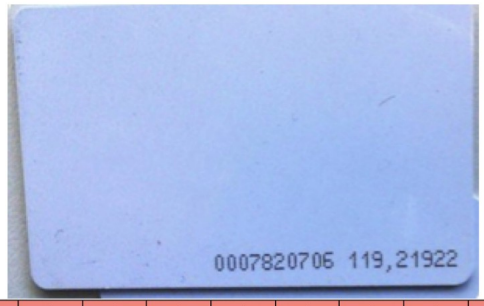
Тестування

Послідовність біт емуляції картки наведено у більш зручному для сприйняття форматі таблиці

Якщо ми розглянемо перший рядок, то комбінація з перших чотирьох біт буде **0010**. Кількість одиниць в рядку не парна, тому контрольний біт встановлено в **1**.

В другому рядку з комбінацією **1100**, кількість одиниць навпаки парна, тому контрольний біт встановлено в **0**. За аналогією перевіряємо усі десять рядків, усі дані збігаються

За аналогією перевіряємо контрольні біти **PC0-PC3**. Усі дані збігаються



1	1	1	1	1	1	1	1	1
				0	0	1	0	1
				1	1	0	0	0
				0	0	0	0	0
				0	0	0	0	0
				0	1	1	1	1
				0	1	1	1	1
				0	1	0	1	0
				0	1	0	1	0
				1	0	1	0	0
				0	0	1	0	1
				0	1	1	0	0

Тестування

	D08;D11	D12;D15	D16;D19	D20;D23	D24;D27	D28;D31	D32;D35	D36;D39
BIN	0000	0000	0111	0111	0101	0101	1010	0010

Останнім кроком візьмемо біт за номерами D08 – D39, тобто безпосередній ідентифікатор клієнта, та подивимось чи коректно його було сформовано. Для цього відкинемо контрольні біти, та запишемо дані у таблицю.

Якщо біти ідентифікатора клаєнта перевести в десяткову систему числення, то ми отримаємо число 7820706, що повністю збігається з ідентифікатором клієнта картки, котру ми моделювали, одже процес емуляції картки пройшов вдало.

Висновки

Першим чином було проаналізовано математичний апарат та було проведено аналіз технології RFID. Було обрано стандарт ISO 14223, виходячи з робочої частоти, необхідної робочої відстані, вимог до безпеки системи та ємності пам'яті.

Цифрові відбитки ключів не займають багато пам’яті, і однією з цілей даної кваліфікаційної роботи було спроектувати апаратну частину проекту з компактними розмірами для зручного транспортування та використання в повсякденному житті. Мікроконтролер серії ATtiny найкраще підходить для цього завдання.

Програмна частина проекту написано мовою програмування C. Розроблена програма займає 1862 байти пам’яті, що приблизно дорівнює 22% від всього обсягу пам’яті мікроконтролера. Програма є досить ефективною з точки зору використання ресурсів, та має потенціал для розширення та модернізації.

В ході написання проекту так було проаналізовано стандарти радіочастотної комунікації, такі як RFID, та вивчені принципи їх роботи. Проаналізовані переваги та недоліки систем побудованих навколо цієї технології.

Дякую за увагу

ДОДАТОК Б

Лістінг виконавчого файлу

```
#include <stdio.h>
#include <EEPROM.h>
#include <avr/interrupt.h>
#include <avr/io.h>
#include <avr/sleep.h>

#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
//#include <Adafruit_SSD1306.h>
#define OLED_RESET 6
//Adafruit_SSD1306 display(OLED_RESET);

#define NUMFLAKES 10
#define XPOS 0
#define YPOS 1
#define DELTAY 2


#define LOGO16_GLCD_HEIGHT 16
#define LOGO16_GLCD_WIDTH  16
static const unsigned char PROGMEM logo16_glcd_bmp[] =
{ B00000000, B11000000,
  B00000001, B11000000,
  B00000001, B11000000,
  B00000011, B11100000,
  B11110011, B11100000,
  B11111110, B11111000,
  B01111110, B11111111,
  B00110011, B10011111,
  B00011111, B11111100,
  B00001101, B01110000,
  B00011011, B10100000,
  B00111111, B11100000,
  B00111111, B11110000,
```

```
  B01111100, B11110000,
  B01110000, B01110000,
  B00000000, B00110000
};

#if (SSD1306_LCDHEIGHT != 32)
//#error("Height incorrect, please fix Adafruit_SSD1306.h!");
#endif

// Пин к которому подключена катушка - пин A5 (19 цифровой)
#define COIL 2



byte facility[2] = { 0x02, 0x0C };
byte cardID[8] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
int colsum[4] = { 0, 0, 0, 0}; // storage for the column checksums

// delay between symbols when we are transmitting
int bittime = 256;

byte RFIDdata[128];

int clock = 0; // storage for the current state of our clock signal.

byte datapointer = 0;
byte state;

//*****************************************************
#define METAKOM_CYFRAL 0xFFFFFFFFFF
#define METAKOM_1 0x365A1140BE
#define CYFRAL_1 0x01FFFFFFF
#define VIZIT_1 0x565A1140BE
#define VIZIT_2 0x365A398149
#define ELTIS 0x0
#define LIFT 0x0B57814601
#define TOILET 0x0400711335 //53

//uint64_t   universalID[8]   =   {1099511627775,   233439314110,   8589934591,
370878267582, 233441952073, 0, 48712730113, 5675447};
uint64_t universalID[] =
{
```

```
  METAKOM_CYFRAL,
  METAKOM_1,
  CYFRAL_1,
  VIZIT_1,
  VIZIT_2,
  0,
  LIFT,
  TOILET
};
uint32_t nibleMask = 15;

#define BTN_UP   3
#define BTN_DWN  5
#define BTN_SEL  4

int8_t keyNumber = 0;

void setup()
{
  Serial.begin(9600);

}

void loop(void)
{
  Serial.println(keyNumber);




  if (digitalRead(BTN_SEL) == HIGH)
  {

    facility[0] = universalID[keyNumber] >> 36 & nibleMask;
    facility[1] = universalID[keyNumber] >> 32 & nibleMask;

    for (uint8_t i = 0; i < 8; ++i ) //забиваем десятичный ID карты в массив
    {
        cardID[i] = (universalID[keyNumber] >> ((7 - i) * 4)) & nibleMask; //в
нулевой элемент массива надо записывать старший разряд, поэтому 7-i
    }
```

```
    EmulateCard();  // start card emulation
  }

}
void WriteHeader(void)
{
  // a header consists of 9 one bits
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
  RFIDdata[datapointer++]=1;
}


void WriteData(byte nibble)
{
  byte data;
  byte rowsum=0;
  for (int i=4; i>0; i--)
  {
    if ((nibble& 1<<i-1) ==0)
    {
      data=0;
    }
    else
    {
      data=1;
      rowsum++;  // increment the checksum value
      colsum[i-1]++; // increment the column checksum
    }


    RFIDdata[datapointer++]= data;
    #ifdef SERIALDEBUG
      Serial.print((int) data);
    #endif
```

```
  }
  // write the row checksum out
  if ((rowsum%2)==0)
  {
    RFIDdata[datapointer++]=0;
    #ifdef SERIALDEBUG
      Serial.print((int)0);
    #endif

  }
  else
  {
    RFIDdata[datapointer++]=1;
    #ifdef SERIALDEBUG
      Serial.print((int)1);
    #endif
  }

    #ifdef SERIALDEBUG
      Serial.println();
    #endif

}


void WriteChecksum(void)
{
  byte data;
  byte rowsum=0;
  for (int i=4; i>0; i--)
  {
    if ((colsum[i-1]%2) ==0)
    {
      RFIDdata[datapointer++]=0;
     #ifdef SERIALDEBUG
      Serial.print((int)0);
     #endif
    }
    else
    {
      RFIDdata[datapointer++]=1;
      #ifdef SERIALDEBUG
```

```
      Serial.print((int) 1);
      #endif
    }
  }

  // write the stop bit
  RFIDdata[datapointer++]=0;

      #ifdef SERIALDEBUG
      Serial.print((int)0);
      #endif

}




void BuildCard(void)
{
  // load up the RFID array with card data
  // intitalise the write pointer
  datapointer=0;

  WriteHeader();
  // Write facility
  WriteData(facility[0]);
  WriteData(facility[1]);

  // Write cardID
  WriteData(cardID[0]);
  WriteData(cardID[1]);
  WriteData(cardID[2]);
  WriteData(cardID[3]);
  WriteData(cardID[4]);
  WriteData(cardID[5]);
  WriteData(cardID[6]);
  WriteData(cardID[7]);

  WriteChecksum();
}
```

```
void TransmitManchester(int cycle, int data)
{

  if(cycle ^ data == 1)
  {
    digitalWrite(COIL, HIGH);
  }
  else
  {
    digitalWrite(COIL, LOW);
  }
}

void EmulateCard(void)
{
  #ifdef SERIALDEBUG
  Serial.println("Emulate Card Entered");
  #endif  // enter a low power modewritedataLEDS(0);  // turn off the LEDs

  BuildCard();

  while (1)
  {
    for(int i = 0; i < 64; i++)
    {
      TransmitManchester(0, RFIDdata[i]);
      delayMicroseconds(bittime);
      TransmitManchester(1, RFIDdata[i]);
      delayMicroseconds(bittime);
    }
  }
}
```

Відомість кваліфікаційної роботи

№	Позначення	Найменування	Дод. відомості
		Текстові документи	
1	ГЮІК. 46741Х.013 ПЗ	Пояснювальна записка	80 с.
		Графічні документи	
2		Слайд-презентація	20 слайдів
		Носії інформації	
3		Компакт-диск (CD)	1 шт.

Змін.	Арк.	Номер докум.	Підп.	Дата	ГЮІК. 46741Х.013 ВД			
Розроб.		Пашков Д.О.			Універсальний цифровий ключ на основі технології RFID	Літ	Аркуш	Аркушів
Перевір.		Хаханова Г. В.				У		
Н.контр.		Хаханова Г. В.				ХНУРЕ Кафедра АПОТ		
Затв.		Чумаченко С.В.			Відомість кваліфікаційної роботи			