10. RFID kártyás azonosítás MF-RC522-es RFID olvasó segítségével

Ennek a projektnek a témája az RFID kártya vagy RFID tag azonosítása. Itt egy beléptető rendszert fogunk létrehozni mely egy bizonyos azonosító számmal vagyis ID-vel rendelkező kártya számára biztosít belépési vagy hozzáférési jogot bármilyen általunk megadott funkcióhoz. A belépési jog engedélyezését vagy elutasítását két darab led dióda segítségével vizualizáljuk. Tetszőlegesen használhatunk még piezo hangsugárzót is hangjelzésként. Természetesen minden értesítést, illetve szükséges információt olvasható formában egy kijelzőn is megjelenítünk.

Szükséges eszközök: Számítógép, Arduino IDE, Arduino Uno, egy db 2x16 vagy 3x20 karakteres LCD kijelző I2C adapterrel, egy db MF-RC522-es RFID olvasó, egy db RFID kártya, egy db RFID tag, szükség szerint egy db piezo hangsugárzó, kettő db led dióda lehetőleg piros és zöld színű, egy db próbapanel, USB kábel és vezetékek.

Az MF-RC522-es olvasó megfelelő használatához szükséges az *MFRC522.h* header fájl implementálása, valamint nem megkell jegyeznünk, hogy mivel ez az olvasó az *SPI* buszrendszeren keresztül kommunikál az Arduino-val ezért az *SPI.h* könyvtárat is illeszük be. Ezek mellet helyet kap még a szokásos *Wire.h* és *LiquidCrystal_I2C.h* páros is.

```
#include<SPI.h> //Az SPI kommunikációs protokoll használatához szükséges könyvtár
#include<Wire.h> //Wire könyvtár beillesztése az I2C busz használatához
#include<LiquidCrystal_12C.h> //Az 12C Folyékony kristályos LCD kijelző kezelő könyvtára
#include <MFRC522.h> //Az RFID olvasó használatához szükséges könyvtár
#define SS_PIN 10 //Az SS Soros bemeneti port definiálása
#define RST_PIN 9 //AZ RST vagyis nullázó port definiálása
#define LED_G 8 //Az engedélyezést jelző led portjának a definiálása
#define LED_R 7 //Az elutasítást jelző led portjának a definiálása
int piezo = 0; //A piezo elem által használt kimenet tárolása egy globális integer változóban
MFRC522 mfrc522 (SS_PIN, RST_PIN); //Típus definiálás az MFRC522-es RFID olvasónak
LiquidCrystal I2C lcd(0x27, 20, 4); //Az általunk használt kijelző karakterkészlete 20 karakter és 3 sor
void setup()
 Serial.begin(9600); //A soros porton történő kommunikáció bitrátája
 SPI.begin(); //Az SPI buszon történő kommunikáció indítása
 mfrc522.PCD Init(): //Az MFRC522-es RFID olvasó modul inicializálása
 pinMode (LED G, OUTPUT); //Az engedélyezést jelző LED diódát tartalmazó Pin kimenetté alakítása
 pinMode (LED_R, OUTPUT); //Az elutasítást jelző LED diódát tartalmazó Pin kimenetté alakítása
   inMode(piezo, OUTPUT); //A piezo elemet tartalmazó pin kimenetté alakítása
 lcd.init(); //Az LCD kijelző inicializálása
 lcd.backlight(); //Az LCD kijelző háttérvilágításának bekapcsolása
  lcd.setCursor(6, 0); //Kurzor pozicionálás ez esetben 6. karakter a 0. sorban
 lcd.print("UDOVOZOLJUK"); //Megadott karakterlánc kiíratása
 lcd.setCursor(0, 1); //Kurzor pozicionálás ez esetben 0. karakter a 1. sorban
 lcd.print("A RENDSZER AKTIV"); //Megadott karakterlánc kiíratása
  Serial.println("A RENDSZER AKTIV"); //Megadott karakterlánc kiíratása a soros portra
  lcd.setCursor(9, 2); //Kurzor pozicionálás ez esetben 9. karakter a 2. sorban
 lcd.print(":)"); //Megadott karakterlánc kiíratása
 delay(5000); //Várakozás 5 másodpercig
  Serial.println("Tap your card!"); //Megadott karakterlánc kiíratása
  lcd.clear(); //Az LCD kijelző tartalmának a törlése
  lcd.setCursor(6, 0); //Kurzor pozicionálás ez esetben 6. karakter a 0. sorban
 lcd.print("ERINTSE"); //Megadott karakterlánc kiíratása
 lcd.setCursor(8, 1); //Kurzor pozicionálás ez esetben 8. karakter a 1. sorban
  lcd.print("A"); //Megadott karakterlánc kiíratása
 lcd.setCursor(8, 2); //Kurzor pozicionálás ez esetben 8. karakter a 2. sorban
 lcd.print("KARTYAJAT"); //Megadott karakterlánc kiíratása
 delay(700); //Várakozás 700 milliszekundum ideig
```

Az SPI busz használatánál négy darab pinre van szükségünk, az Arduino Uno esetében ezek a pinek a következők SS soros bemenet a 10-es pin, MOSI Master OUT SLAVE IN kimenet a 11-es pin, MASTER IN SLAVE OUT bemenet a 12-es pin és végül az SCK vagyis az órajel kimenet pedig a 13-as pin. A könyvtárak utána fontos ezen pinek közül kettőnek a definiálása. A #define SS PIN 10 parancs a 10-es pinre csatlakoztatott soros bemenetet a #define RST PIN 9 pedig a 9es pinre csatlakoztatott nullázó, vagyis reset kimenetet határozza meg. Majd ugyanezzel az eljárással a piros és zöld led diódáink által használt kimeneteket is meghatározhatjuk. Amennyiben szeretnénk hangjelzést is hozzáadni a projektünkhöz deklaráljunk egy piezo nevezetű integer típusú globális változót a csatlakozásra szánt pint tetszőlegesen választhatjuk ki. Ezután egy típus definiálás és az SPI busz SS és RST pinjeinek a hozzárendelése az MF-RC522-es olvasóhoz történik. Következhet a void setup() szekció. Ebben a projektben szeretnénk használni a soros port által kapott lehetőségeket ezért egy Serial.begin(9600) paranccsal kezdünk, ám mivel az SPI buszt is szeretnénk használni az SPI.begin() funkciót is meg kell hívnunk. Az mfrc522.PCD_Init() funkcióval az RFID olvasónkat tudjuk inicializálni ez történik a következő sorban. Mivel tudjuk, hogy a led diódáknak és az esetlegesen használt piezo hangsugárzónak kimenetre van szüksége ezért az Arduino 7-es, 8-as és a piezo elemre csatolt pinjét kimenet kell alakítanunk. Ezt a folyamatot követi az lcd inicializálás és háttérvilágításának bekapcsolása. Mivel azt szeretnénk, hogy a program már ezen szekciójában a kijelző üzenetet jelenítsen meg ezért a kurzort a kívánt helyre kell helyeznünk annak érdekében, hogy az üzenetet meg tudjuk jeleníteni. Ebben az esetben egy üdvözlő képernyőt szeretnénk készíteni, némi animációval. A megjelenítésre szánt üzenetet a következőképpen szeretnénk megjeleníteni az ÜDVÖZÜLJÜK karakterláncot a kijelző 0-ik sorának a közepére, vagyis a 6-ik karaktertől kell kezdenünk a kiíratást. Majd a kijelző 1-es sorába a RENDSZER AKTÍV szöveget. Mivel ebben a projektben ezúttal egy 3x20 karakteres kijelzőt használtunk az előzőleg megjelenített üzenet alá még van lehetőségünk valamit megjeleníteni, legyen az egy :) hangulat jel. Ez az üzenet öt másodpercig látható a kijelzőn majd törlésre kerül, mivel felváltja az ÉRINTSE A KÁRTYÁJÁT üzenet, melyet három pont karakter követ egymás után 700 milliszekundum várakozás után.

```
lcd.setCursor(12, 2); //Kurzor pozicionálás ez esetben 12. karakter a 2. sorban
 lcd.print("."); //Megadott karakterlánc kiíratása
 delay(700); //Várakozás 700 milliszekundum ideig
 lcd.setCursor(13, 2); //Kurzor pozicionálás ez esetben 13. karakter a 2. sorban
 lcd.print("."); //Megadott karakterlánc kiíratása
 delay(700); //Várakozás 700 milliszekundum ideig
 lcd.setCursor(14, 2); //Rurzor pozicionálás ez esetben 14. karakter a 2. sorban
 lcd.print("."); //Megadott karakterlánc kiíratása
void loop() //ciklus
 if ( !mfrc522.PICC IsNewCardPresent())//Feltétel vizsgálat, ha új kártya van jelen:
   return; //kilépés a feltétel vizsgálatból
 if (! mfrc522.PICC ReadCardSerial())//Feltétel vizgsálat kártya ID-jének kiolvasása
   return; //kilépés a feltétel vizsgálatból
 Serial.print("UID tag:"); //Megadott karakterlánc kiíratása a soros portra
 String content = ""; //String típusú üres karakterláncot tartalmazó változó
 byte letter; //Byte típusú változó
 for (byte i = 0; i < mfrc522.uid.size; i++) //for ciklus mely véqiqhalad a kártya ID-ján:
   Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? "0" : ""); //Az ID ben lévő számok Hexadecimális stringre alakítása
   Serial.print(mfrc522.uid.uidByte[i], HEX); //Az átalakított ID kiíratása a soros portra
content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ")); //konkatenálása az ID vel és 0x hozzarendelése</pre>
   content.concat(String(mfrc522.uid.uidByte[i], HEX)); //A konkatenáció kiíratása a soros portra
 Serial.println(); //Üres karakterlánc kiíratása a soros portra
 Serial.print("UZENET:"); //Megadott karakterlánc kiíratása a soros portra
 content.toUpperCase(); //A változó tartalmának nagy karakterekre alakítása
 if (content.substring(1) == "F0 1E 78 7A") //Feltétel vizsgálat:
   Serial.println("BELEPES ENGEDELYEZVE"): //Megadott karakterlánc kiíratása a soros portra
   Serial.println(); //Üres karakterlánc kiíratása a soros portra
   lcd.clear(); //Az LCD kijelző tartalmának a törlése
   lcd.setCursor(3, 1); //Kurzor pozicionálás ez esetben 3. karakter a 1. sorban
```

2

A program ciklusában egy feltétel vizsgálattal indítunk. Mely azt vizsgálja, hogy jelen van-e új kártya, ha igen akkor kilépünk az első feltétel vizsgálatból és egy újabb feltételhez ugrunk, ahol ennek a kártyának az azonosító számát olvashatjuk ki az *mfrc522.PICC_ReadCardSerial* funkció segítségével. A soros portra juttatunk egy üzenetet mely után szeretnénk kiíratni a kártyánk azonosítóját, de előbb ezt még egy változóban kell tárolnunk, ezért létrehozunk egy *string* típusú nem globális változót. Majd ezt követően egy *byte* típusút is. Egy *for* ciklus segítségével a byte változónk értékét addig növeljük még nagysága kisebb, mint az mfrc522.uid.size értéke. A ciklus minden egyes lefutásában ezt az értéket kiíratjuk a soros portra a következő feltétel vizsgálat mellett (*mfrc522.uid.uidByte[i] < 0x10 ? "0" : "")* melynek köszönhetően az RFID kártyánk vagy tag-ünk azonosítója hexadecimális formában kerül kiíratásra. Ezt követi egy érték adó művelet, ahol egy string típusú változóba helyezzük az azonosítónk hexadecimális formáját, ehhez a művelethez a *content.concat* parancsot használjuk mivel, konkatenációról vagyis összefűzésről van szó. A tárolt azonosítót jelenítsük meg a soros porton a *string* változó tartalmát érdemes a kiíratás előtt csupa nagy betű formájúra alakítani, ez által könnyebben elkülöníthető a programkódban, ezt a *content.toUpperCase()* funkció segítségével valósíthatjuk meg. Mivel a

soros porton már megjelenítettük a kívánt azonosítót ezért most már egy egyszerű feltétel vizsgálattal hozzárendelhetjük a belépési vagy hozzáférési jogot. A feltétel azt vizsgálja, hogy a content.substring(1) -ben tárolt azonosító megegyezik-e az általunk előbb megjelenített azonosítóval amennyiben igen a kijelzőn a *BELÉPÉS ENGEDÉLYEZVE* üzenet jelenik meg.

```
lcd.print("BELEPES ENGEDELYEZVE"); //Megadott karakterlánc kiíratása
 delay(1000); //Várakozás 1 másodpercig
  digitalWrite(LED_G, HIGH); //Engedélyezést jelző Led bekapcsolva
  tone (piezo, 500): //Piezo elem megszólaltatása 500 hz frekvencian
  delay(300); //Várakozás 300 milliszekundum ideig
 noTone (piezo); //Piezo eleme elhallgattatása
 delay(10000); //Várakozás 10 másodpercig
 digitalWrite(LED G, LOW); //Engedélyezést jelző Led kikapcsolva
else //Egyébként
  Serial.println("BELEPES MEGTAGADVA"); //Megadott karakterlánc kiíratása a soros portra
  Serial.println(); //Üres karakterlánc kiíratása a soros portra
 lcd.clear(); //Az LCD kijelző tartalmának a törlése
 lcd.setCursor(3, 1); //Kurzor pozicionálás ez esetben 3. karakter a 1. sorban
 lcd.print("BELEPES MEGTAGADVA"); //Megadott karakterlánc kiíratása
  digitalWrite(LED_R, HIGH); //Elutasítást jelző Led bekapcsolva
 delay(1000); //Várakozás 1 másodpercig
  tone(piezo, 1200); //Piezo elem megszólaltatása 1200 hz frekvencian
 delay(300); //Várakozás 300 milliszekundum ideig
 noTone (piezo); //Piezo eleme elhallgattatása
 delay(3000); //Várakozás 3 másodpercig
 digitalWrite(LED_R, LOW); //Elutasítást jelző Led kikapcsolva
}
```

3.

}

Miután a kijelző igazolta a belépés engedélyezését jelezzük ezt a zöld led dióda bekapcsolásával is, mint tudjuk ezt a *digitalWrite(LED_G, HIGH)*_paranccsal tehetjük meg. Amennyiben egy piezo elemet is használtunk akkor azzal is jelezzünk egy tetszőleges frekvencián. Visszatérve az azonosítót vizsgáló feltételünkhöz, amennyiben az azonosító nem egyezik meg az általunk megadottal abban az esetben a *BELÉPÉS MEGTAGADVA* üzenet kerül megjelenítésre és a vörös led diódát kapcsoljuk aktív állapotba egy általunk megszabott időre, a jelenlegi esetben ez az idő három másodperc. Majd az Arduino eszközünk újraindítását követően az azonosítás folyamata újra végbe megy.



4

Az MF-RC522-es RFID olvasónk megfelelő bekötéséhez segítségünkre lehet a fenti 4.ábra. Itt láthatjuk, hogy mely csatlakozó mely pinre csatlakozik. Ami az olvasót illeti érdemes az *SPI* busz által használt csatlakozókat bekötni először mivel elég közel vannak elhelyezve egymáshoz. Az *RST* csatlakozó az Arduino 9-es pinjére csatlakozik az *SS* a 10-es pinre, a *MOSI* a 11-es pinre a *MISO* a 12-es pinre és végül az *SCK* a 13-as pinre csatlakozik. Ezután beköthetjük a tápfeszültséget is az olvasó részére ám nem szabad megfeledkeznünk arról, hogy 3.3 volt feszültséget használjunk a korábban megszokott 5 volt helyett. Miután az olvasó GND-jét csatlakoztattuk a közös földhöz kössük be a led diódáinkat is, ahogyan az ábrán láthatjuk itt egy RGB dióda van használva. A dióda *R*-rel jelölt lábát kössük az Arduino 7-es pinjére a *G*-vel jelölt lábát pedig a 8-as pinre, végül csatlakoztassuk a GND-t is a megfelelő helyre. A helyes bekötést eredményezően az áramkör működése a következő.

