# ZÁRÓDOLGOZAT

Major Tamás József



# Magyar Agrár- és Élettudományi Egyetem Károly Róbert Campus Műszaki intézet programtervező informatikus (fejlesztő) felsőoktatási szakképzés szak

RFID eszköz megszemélyesítése mikrokontroller segítségével, valamint az adatok kiolvasása és küldése webszerverre

Belső konzulens:

Sike Zoltán mesteroktató

Belső konzulens intézete/tanszéke: Alkalmazott Informatika

Tanszék

Külső konzulens:

Készítette:

Major Tamás József

Gyöngyös 2024

# Tartalom

1.	E	Beve	ezete	és és célkitűzések	2
2.	I	roda	almi	áttekintés	3
3.	A	۹ pr	ojek	t bemutatása	1
	3.1		Elvi	felépítés	1
	3.2	1	Alka	lmazott hardvereszközök bemutatása, azok működése	5
	З	8.2.1	L	Mikrokontrollerek (ESP8266, ESP32)	5
	3	8.2.2	2	RFID (Radio-Frequency Identification) eszköz (kártya, kulcstartó, csuklópánt)	ō
	3	8.2.3	3	Az eszközök összekapcsolása	3
	3.3	1	A fel	lhasználói program telepítése és a projekt működése	Э
	3	8.3.1	L	A kártya program telepítése	Э
	З	8.3.2	2	A kártyaírás folyamata12	2
	З	8.3.3	3	A kártyaolvasás folyamata és megjelenítése webes felületen10	ŝ
	3.4		Fejle	esztői környezetek telepítése, a programok belső működése1	Э
	3.5		Fejle	esztési lehetőségek röviden24	4
4.	Ċ	İssz	efog	glalás 2!	5
5.	F	elh	aszn	nált irodalom	5
6.	Á	Ábra	ijegy	yzék2	7

#### 1. Bevezetés és célkitűzések

A záródolgozat célja egy olyan projekt megvalósításának bemutatása, melyben egy rádióvezérlésű MIFARE kártyára történik adatírás, C# programnyelven megírt telepíthető program segítségével, illetve megvalósul a kártyáról történő adatolvasás is, valamint annak wifi kapcsolaton keresztül webszerverre való feltöltése. Mindezt mikrokontrollerhez (mikróvezérlőhöz) csatlakoztatott író/olvasó eszköz segítségével.

Az elképzelt projekt szimulál egy egyszerű beléptető rendszert, melyhez megszemélyesített RFID kártya kerül kiadásra. A rendszer kiolvassa a kártya azonosítóját, a használó nevét, valamint a kártyakezelés idejét. A projekthez tartozó program képes az adatokat webszerverre küldeni, ahol lehetőség van az adatok megjelenítésén túl, azok visszakeresésre is.

A gyakorlatban ez a következő módon valósul meg:

Két darab mikrokontrollerhez kapcsolódik egy-egy RFID kártyaíró/olvasó eszköz. Az egyik eszközpáros segítségével adatot lehet írni a kártyára (vagy más hasonló szabvány szerint működő eszközre), a másik eszközpárossal pedig ki lehet olvasni a kártya adatait, amit a mikrokontroller wifi kapcsolaton keresztül elküld egy MYSQL adatbázisba. Az adatok vizualizálása webes felületen történik.

A záródolgozat bemutatja a projekt elvi felépítését, valamint azt, hogy milyen hardver eszközök kerültek felhasználásra, azok milyen tulajdonságokkal rendelkeznek, hogyan kell őket összekapcsolni, valamint hogyan működnek.

## 2. Irodalmi áttekintés

A mikróvezérlők napjaink egyik legsokoldalúbb hardver eszközei. Nagyon sokféle területen megbízhatóan felhasználhatók. Számtalan blog és weboldal foglalkozik a mikrokontrollerek beüzemelésével, programozásával, fejlesztői környezetének használatával, illetve annak különböző módjaival.

A programozás közben felmerült akadályok megoldása számos fórumnak témája és kínálnak segítséget a programozók számára.

Az online források nagy része nemcsak az eszközök programozásával foglalkozik, hanem annak a beszerzését is elősegíti, esetenként többféle lehetőséget is biztosít, például az árak összehasonlítását.

A leggyorsabban természetesen az online forrásokból lehet hozzájutni a megfelelő információhoz, egy-egy eszköz műszaki adatlapjához, ami segítséget nyújthat a programozás során. Az eszközök fejlődése olyan mértékű, hogy az írott irodalom ebben a témában többnyire háttérbe szorul.

# 3. A projekt bemutatása

### 3.1 Elvi felépítés

A projekt két fő részből áll. Az egyik a kártyára történő adatírás megoldása. Ennek során a számítógép egy USB kábel segítségével összekapcsolásra került egy mikrovezérlővel, ami közvetlen kábeles összeköttetésben van egy kártyaíró eszközzel. A másik rész a kártyán lévő adatok kiolvasása és helyi routeren keresztül webszerverre továbbítása. A kártyaolvasáshoz nincs szükség számítógépre, elegendő a helyi hálózaton keresztüli wifi kapcsolat és a mikrovezérlő tápellátásának biztosítása. Ez történhet közvetlenül USB töltőadapterrel, vagy USB-A/USB-C kábelen keresztül számítógéphez kapcsolva. Ez utóbbi esetben adatforgalom nincs a kábelen, hanem csak a tápfeszültség biztosítása történik. A második részhez kapcsolódik szorosan a szerveren tárol adatok felhasználók számára történő megjelenítése és keresési funkció biztosítása az adatbázisban. A teljes rendszer, illetve projekt elvi felépítését az alábbi ábra szemlélteti.



**1. ábra:** Elvi felépítés (Forrás: Saját szerkesztés, Draw.io 24.0.4 szoftver felhasználásával)

#### 3.2 Alkalmazott hardvereszközök bemutatása, azok működése

#### 3.2.1 Mikrokontrollerek (ESP8266, ESP32)

A mikrokontrollerek egy vezérléséi célokra használható kisméretű célszámítógépek. Sokoldalú eszközök, mely számos beépített perifériával rendelkeznek. Így tartalmaznak wifi-t, bluetooth-t, analóg és digitális I/O pin-eket, SPI, I2C, UART kommunikációs interfészeket. Típustól függően más-más felépítésűek, de az alaptípusok is számtalan, nagyon jól használható lehetőséggel rendelkeznek. Ideális választás az IoT (Internet of Things) alkalmazásokhoz, szenzoradatok feldolgozásához, otthoni vagy akár ipari automatizálási feladatok megoldásához.

Népszerűségükhöz, a sokoldalúságuk mellett hozzájárul az eszközök ára, a könnyű elérhetősége, a többféle programnyelv használhatóságának lehetősége, a processzor gyorsasága, az alacsony fogyasztása és kis mérete is.

Ebben a projektben kétféle típus került felhasználásra az ESP8266 és az ESP32. Mindkettőt a kínai Espressif Systems fejlesztett ki. Az ESP32, lényegében az ESP8266 továbbfejlesztett változata több új funkcióval. Így: a processzora kétmagos, rendelkezik bluetooth kapcsolattal, az órajele 80 MHz helyett 160 MHz, 17 vezérlésre használható kimenet helyett 34 található, érintés, hőmérséklet és mágneses tér szenzorral rendelkezik.

USB-A/USB-C típusú kábel segítségével csatlakoztathatók a számítógéphez. A kábel alkalmas tápellátást és/vagy adatkapcsoltot is biztosítani az mikrovezérlő és a számítógép között. A fejlesztői felületen megírt program szintén ezen a kábelen keresztül tölthető fel az eszközre.



2. ábra: ESP8266, ESP32 mikrovezérlő (Forrás: https://www.cytron.io)

A modul az érintésmentes kommunikációhoz kifejlesztett eszköz, mely 13,56 MHz-en működik és támogatja (többek között) az ISO/IEC 14443 A/MIFARE szabványt. A modul hatékony kommunikációt biztosít a hasonló szabvány szerint készült kártyákkal (kulcstartókkal, csuklópántokkal).

A hozzá csatlakoztatott vezérlő eszközzel többféle protokollon keresztül képes kommunikálni. Ismeri az SPI, I2C, UART protokollokat. A legjobb teljesítményt a SPI (Serial Peripheral Interface) adja, melynek jellemzője egy master-slave architektúra, a master (mikrokonrtoller) eszköz irányítja a kommunikációt, a slave (iró/olvasó) eszköz pedig válaszol.



3. ábra: MFRC522 RFID író/olvasó eszköz (Forrás: https://www.cytron.io)

# 3.2.2 RFID (Radio-Frequency Identification) eszköz (kártya, kulcstartó, csuklópánt)

Ezek az eszközök áramforrás nélküli passzív elemek. Lényegében egy mikrochipből és egy antennából állnak. A chip képes adatokat tárolni és feldolgozni. Az író/olvasó eszköz elektromágneses teret hoz létre, melybe bekerülve a címke antennája képes a saját chipjét megtáplálni, az adatokat feldolgozni és rádiójel segítségével visszaküldeni az író/olvasó eszköznek. Fő jellemzői:

- működési frekvencia 13,56 MHz,
- adattároló kapacitása 1 KB,
- egyedi UID azonosítója.



4. ábra: RFID kártya, kulcstartó, csuklópánt (Forrás: https://www.temu.com)

Valamennyi említett szabvány szerinti eszköz memóriája az adatokat azonos struktúrába rendezi. Egy 1 KB-os eszköz memóriája 16 szektorból (0-15), szektoronként 4 blokkból (0-3) és blokkonként 16 adatbájtból (0-15) áll.

Ennek ismeretére a programozás során szükség van, mert meg kell határozni, hogy melyik blokkba akarjuk az adatot írni, illetve melyikből akarjuk kiolvasni.

Card U	ID: 7D	B9 1	IC (	23																	
Card Si	AK: 08																				
PICC ty	ype: MI	FARI	E 18	КB																	
Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Acce	ssl	Bit	s
15	63	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0]	0 3	1]	
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0]	0 (	D ]	
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0]	0 (	D ]	
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0]	0 (	D ]	
14	59	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0]	0 3	1]	
	58	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0]	0 (	D ]	
	57	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0]	0 (	D ]	
	56	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0]	0 (	D ]	
	5	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0	0	0]	
	4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0]	0	0]	
0	3	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0]	0	1]	
	2	4D	65	6C	20	47	69	62	73	6F	6E	00	00	00	00	00	00	[ 0	0	0]	
	1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[ 0	0	0]	
	0	7D	В9	1C	C3	1B	08	04	00	62	63	64	65	66	67	68	69	[ 0	0	0]	

**5. ábra:** RFID eszközök adatstruktúrája (Forrás: saját képernyőkép, Arduino IDE – serial monitor)

A fenti esetben az az adatírás a 2. blokkba történik, a kártya egyedi azonosítója (UID) pedig a 0. blokkban.

#### 3.2.3 Az eszközök összekapcsolása

Az író/olvsó eszköz és a mikrokontroller összekapcsolását a tervezett kommunkációs elképzelés alapján kell végezi, ami jelenleg a SPI protokoll. Az iró/olvasó eszköz valamennyi csapjának megtalálható a megfelelő alapértelmezett csapja a mikrokontrollereken is. A vezetékelést ezek alapján kell elvégezni. Esetünkben ez különbözik az írásra illetve az olvasásra használt mikorvezérlők esetében, ugyanais az írásra ESP 8266 típusú, míg az olvasásra ESP32 típusú mikrokonroller került felhasználásra.



**6. ábra: Mikrovezérlők bekötése** (Forrás: Saját szerkesztés, Fritzing szoftver felhasználásával)

A használt csatornák:

SS (Slave Select): jelez a masternek, hogy melyik slave perifériával szeretne kommunikálni,

SCK (Serial Clock): órajelet biztosít a kommunikációhoz,

MOSI (Master Out Slave In): a master (mikrovezérlő) által küldött adatokat továbbítja a slave (író/olvasó eszköz) felé,

MISO (Master In Slave Out): a slave által küldött adatokat továbbítja a master felé.

A VCC és a GND a tápfeszültséget biztosítja, RESET pedig a kikapcsolás és a készülék alaphelyzetbe állítását végzi.



7. ábra: MFRC522 pin kiosztása (Forrás: Saját szerkesztés, Fritzing szoftver felhasználásával)

# 3.3 A felhasználói program telepítése és a projekt működése

## 3.3.1 A kártya program telepítése

A működő C# programkód elkészítése után lehet felépíteni új projektként a telepítőcsomagot.

Ennek eredményeként létrejönnek a már futtatható telepítő fájlok:

🗸 🔤 CharpDataToArduino_S	^	Név	Módosítás dátuma	Típus	Méret
🗸 📙 CharpDataToArduino		😽 RFID_write_setup.msi	2024. 03. 31. 23:02	Windows Installer csomag	665 KB
> 📙 bin		💽 setup.exe	2024. 03. 31. 23:02	Alkalmazás	550 KB
> 📙 obj					
Properties					
✓ RFID_write_setup					
📙 Debug					
Release					

8. ábra: Létrehozott telepítő fájlok (Forrás: saját képernyőkép, Windows Intéző)

A .msi kiterjesztésű telepítőfájlra kettőt kattintva elindul a telepítési folyamat. A Windows klasszikus telepítési varázslója vezet végig a folyamaton. Valamennyi lépés után a *Next* gombra kattintva tudunk továbblépni.



**9. ábra:** RFID\_write\_setup program telepítése – 1. lépés (Forrás: saját képernyőkép, Windows telepítő varázsló)

Először meghatározásra kerül a telepítési könyvtár. Itt látható az elérési útvonalban az a *Manufacturer* név, amit megadtunk a projekt készítése közben, illetve látható magának a telepítő készítéskor adott projektnév is. Itt kell meghatároznunk, hogy a programot csak az éppen bejelentkezett felhasználó, vagy majd mindenki használhatja-e. A beállítások után a *Next* gombra kattintva léphetünk a következő ablakba.

₩ RFID_write_setup -	-	×	Solution Explorer	<b>~</b>	ųΧ
Select Installation Folder	[		Image: Search Solution Explore       Image: Solution CharpDa	☞ (Ctrl+é) ataToArduino' (2 of 2 proje	』 " ァー cts)
The installer will install RFID_write_setup to the following folder. To install in this folder, click "Next". To install to a different folder, enter it below	or click "Brows	se".	<ul> <li>CharpDataTo/</li> <li>Properties</li> <li>Properties</li> <li>Properties</li> <li>Properties</li> <li>Properties</li> <li>Properties</li> <li>App.config</li> <li>Form1.cs</li> </ul>	Arduino ; }	
<u>C:\Program Files (x86)</u> <u>MT_GHF0SV\RFID_write_setup\</u> Install RFID_write_setup for yourself, or for anyone who uses this computer: <u>Everyone</u> Just me	Browse Disk Cost		<ul> <li>Form 1.</li> <li>Form 1.</li> <li>C # Program.cz</li> <li>RFID_write_set</li> <li>Potected D</li> <li>Micross</li> <li>System</li> <li>Primary ou</li> <li>tech_rfid_iot</li> </ul>	Designer.cs resx s tup Dependencies oft .NET Framework .Net.Http.dll utput from CharpDataToArd conico	duino (.
< <u>B</u> ack <u>N</u> ext >	Cance	el	▲ Solution Explorer Git 6	Changes	
			Properties <b>RFID_write_setup</b> Deg <b>Bill</b> Description DetectNewerInstalled InstallAIIUsers Keywords Localization Manufacturer ManufacturerUrl PostBuildEvent	♥ ployment Project Propertie True False English (United States) MT_GHFOSV	<b>₽</b> × s ▼

**10. ábra:** RFID\_write\_setup program telepítése – 2. lépés (Forrás: saját képernyőkép, Windows telepítő varázsló és Visual Studio 2022 fejlesztői környezet) A telepítés előtt még egy ellenőrző ablakot látunk majd ismét a *Next*-re kell kattintani, ezután elindul a telepítési folyamat

🙀 RFID_write_setup			_		×
Confirm Installation					
The installer is ready to install RFID_write	_setup on your con	nputer.			
Click "Next" to start the installation.					
	< <u>B</u> ack	<u>N</u> ext >		Cano	cel

**11. ábra:** RFID\_write\_setup program telepítése – 3. lépés (*Forrás: saját képernyőkép, Windows telepítő varázsló*)

A telepítés során az asztalon megjelenik a Visual Studio-ban a telepítőkészlethez rendelt saját ikon. A telepítés végén üzenet jelenik meg, a sikeres installálásról, melyet bezárhatunk a *Close* gombbal. Ezután már használható a program.

🕼 RFID_write_setup —		×	
Installation Complete			
RFID_write_setup has been successfully installed.			
Click "Close" to exit.			
Please use Windows Update to check for any critical updates to the .NET Framew	ork.		
< Back Close	Cance	el	

**12. ábra:** RFID\_write\_setup program telepítése – 4. lépés (*Forrás: saját képernyőkép, Windows telepítő varázsló*)

A telepítés után a *Programfájlok (x86)* mappában létrejönnek a futtatáshoz szükséges állományok.

📙   🛃 📮   RFID_write_setup							-		×	
Fájl Kezdőlap Megosztá:	s Néze	t							~ 🕐	
← → × ↑ 🦲 « Helyi lemez (C:) » Programfájlok (x86) » MT_GHFOSV » RFID_write_setup 🗸 ⊘ Keresés: RFID_write_setup										
	^	Név ^	Módosítás dátuma	Típus		Méret				
Y Gyors eleres		📧 CharpDataToArduino.exe	2024. 03. 31. 21:13	Alkalmazás		13 KB				
Asztal	*	🖓 CharpDataToArduino.exe.config	2024. 03. 31. 23:02	XML Config	guration File	1 KB				
Letöltések	*	System.Net.Http.dll	2024. 03. 31. 23:02	Alkalmazás	kiterjesztés	85 KB				
🖆 Dokumentumok	*	tech_rfid_iconico	2024. 03. 31. 23:02	ICO fájl		67 KB				
📰 Képek	*									
bedandó_tervezet										
🔄 Camtasia Studio										
🔒 Záródolgozat										
záródolgozat										
🔵 OneDrive - Personal										
💻 Ez a gép										
🗊 3D objektumok										
Asztal										
Dokumentumok										
E Kének 4 elem	*									

**13. ábra:** RFID\_write\_setup program telepítése – 5. lépés (Forrás: saját képernyőkép, Windows Intéző)

## 3.3.2 A kártyaírás folyamata

Duplán kattintva az asztalon található ikonra, a program elindul és megjelenik a GUI (Graphical User Interface). Itt egy egyszerű űrlap segítségével könnyen el lehet végezni az adatírást a számítógéppel kapcsolatban lévő mikrovezérlőn keresztül.

🖳 Adatírás RFID kártyára	_			×
KAPCSOLAT				
Kapcsolódás az íróhoz Kapcsolat bontása				
STATUSZ: NINCS KAPCSOLAT				
NÉV:		Írás	kártyá	ra
SOROS PORT MONITOR :				

**14. ábra:** Kártyaíró program a programindítás után (Forrás: saját képernyőkép, Adatírás RFID kártyára – GUI)

Először fel kell építeni a kapcsolatot a számítógép és a mikrovezérlő között. Ezt úgy tehetjük meg hogy rákattintunk a *Kapcsolódás az íróhoz* gombra. Ekkor meghívódik egy metódus, ami megnyitja az előre definiált soros port-ot, az előre definiált bitsebességgel, valamint vizuálisan láthatóvá teszi, hogy a kapcsolódás megtörtént. Ha a kapcsolódás során hiba keletkezik, akkor hibaüzenet jelenik meg.

🖳 Adatírás RFID kártyára		_		×
KAPCSOLAT				
Kapcsolódás az íróhoz	Kapcsolat bontása	]		
STATUSZ:	KAPCSOLÓDVA			
NÉV:		Ír	ás kárty	ára
SOROS PORT MONITOR :				

**15. ábra:** Kártyaíró program a kapcsolat felépítése után (Forrás: saját képernyőkép, Adatírás RFID kártyára – GUI)

🔛 Adatírás RFID kártya	ára	_		$\times$
KAPCSOLAT				
Kapcsolódás az ír	róhoz Kapcsolat bontása			
STATUSZ:	NINCS KAPCSOLAT			
NÉV:			Írás kárty	ára
SOROS PORT MONI	TOR :			
	TOR : A következő port nem létezik: "COM5".	×		

**16. ábra:** Kártyaíró program hiba esetén (Forrás: saját képernyőkép, Adatírás RFID kártyára – GUI)

A kapcsolódás után beírható a név, a megfelelő űrlapmezőbe, majd rá kell kattintani az *Írás kártyára* gombra. Ekkor megjelenik egy ellenőrző üzenet és egy felhívás, hogy a felhasználó közelítse a kártyát az íróeszközhöz.

赔 Adatírás RFID kártyára		-	-		×
KAPCSOLAT					
Kapcsolódás az íróhoz	Kapcsolat bontása				
STATUSZ:	KAPCSOLÓDVA				
NÉV: Mel Gibson			Írá	s kártya	ára
SOROS PORT MONITOR :					
A következő adatokat írom a l Mel Gibson	kártyára:				
Kérem közelítsd a kártyát az íí Várd meg a 'Kártyaírás kész!'	róhoz! üzenetet!				

 17. ábra: Kártyaíró program az adatbevitel után, közvetlenül a fizikai adatírás előtt.
 (Forrás: saját képernyőkép, Adatírás RFID kártyára – GUI)

Közelítve a kártyát az írófejhez megtörténik a kártya azonosítása, az adatok kártyára írása, majd annak ellenőrzése, illetve az írt adatok fizikai visszaolvasása. Ezeket a folyamatokat a mikrovezérlő végzi. Futás közben üzeneteket küld a soros port-ra, ami a GUI "SOROS PORT MONITOR" elnevezésű text boxában látható.

SOROS PORT MONITOR :	
A következő adatokat írom a kártyára: Mel Gibson	^
Kérem közelítsd a kártyát az íróhoz! Várd meg a 'Kártyaírás kész!' üzenetet!	
**Kártya detektálás** Kártya UID azonosító: 7D B9 1C C3	
Adatírás Hitelesítés sikeres! Adatírás rendben!	
Ellenőrzés	
Adatok olvasása Hitelesítés sikeres! Olvasás rendben!	*

**18. ábra:** Kártyaíró program a kártyaírás közben – 1. (Forrás: saját képernyőkép, Adatírás RFID kártyára – GUI)

Ha rendben volt, minden akkor megjelennek a kiolvasott kártyaadatok, valamint a "Kártyaírás kész!" üzenet.

SOROS F	SOROS PORT MONITOR :									
		^								
A kártya	adatai:									
Card UID	D: 7D B9 1C C3									
Card SA	K: 08									
PICC type	e: MIFARE 1KB									
Sector B	lock 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 AccessBits									
15 6	3 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									
61	00 00 00 00 00 00 00 00 00 00 00 00 00									
60	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0									
14 5	9 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									
57	00 00 00 00 00 00 00 00 00 00 00 00 00									
56	00 00 00 00 00 00 00 00 00 00 00 00 00									
13 5	5 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									
53	00 00 00 00 00 00 00 00 00 00 00 00 00	v -								
1 50										

**19. ábra:** Kártyaíró program a kártyaírás közben – 2. (Forrás: saját képernyőkép, Adatírás RFID kártyára – GUI)

	SOROS PORT MONITOR :										
	2	11	00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]	^							
		10	00 00 00 00 00 00 00 00 00 00 00 00 00								
		9	[000] 000000000000000000000000000000000								
		8	00 00 00 00 00 00 00 00 00 00 00 00 00								
	1	7	00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF FF [ 0 0 1 ]								
		6	00 00 00 00 00 00 00 00 00 00 00 00 00								
		5	00 00 00 00 00 00 00 00 00 00 00 00 00								
		4	00 00 00 00 00 00 00 00 00 00 00 00 00								
	0	3	00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]								
		2	4D 65 6C 20 47 69 62 73 6F 6E 00 00 00 00 00 00 [ 0 0 0 ]								
		1	00 00 00 00 00 00 00 00 00 00 00 00 00								
		0	7D B9 1C C3 1B 08 04 00 62 63 64 65 66 67 68 69 [ 0 0 0 ]								
	<i></i>	- ( - ( - (		÷							
ľ	karty	airas	KESZ!								
				~							

**20. ábra:** Kártyaíró program a kártyaírás befejezése után. (Forrás: saját képernyőkép, Adatírás RFID kártyára – GUI)

Látható, hogy az előre definiált a 2-es számú blokkba került be az űrlapon beírt név, hexadecimálisan kódolva.

Egy egyszerű kódkonvertáló weboldallal visszaellenőrizhető az adatok helyessége.

Paste hex numbers or drop file	
4D 65 6C 20 47 69 62 73 6F 6E	\$ //
Character encoding	
ASCII	~
$\checkmark$ Convert X Reset $\uparrow_{\downarrow}$ Swap	
Mel Gibson	<b>↓</b>

**21. ábra:** Adat ellenőrzés online konvertáló program felhasználásával. (Forrás: saját képernyőkép, https://www.rapidtables.com/convert/number/hex-to-ascii.html)

#### 3.3.3 A kártyaolvasás folyamata és megjelenítése webes felületen

A kártyaolvasás folyamata a fejlesztői környezet – Arduino IDE 2.3.2 – soros monitorán követhető, amennyiben a kártya olvasását végző eszközpárost a számítógép USB portjával összekapcsoljuk. Így látható a wifi kapcsolat felépülése után a helyi routertől kapott IP cím,

a kártyaolvasó inicializálása, majd az eszköz várakozása a kártya közelítésére, mely másodpercenkénti dátum és idő kiíratásával kerül szemléltetésre.



**22. ábra:** A fejlesztői környezet (Forrás: saját képernyőkép, Arduino IDE 2.3.2)

A számítógéphez történő csatlakozás nem feltétele az olvasási műveletnek. Az eszköz ugyanúgy működik más tápellátásról (például USB töltő) is, de természetesen ilyenkor az olvasás folyamat vizuálisan nem kontrollálható semmilyen módon, csak az eredménye olvasható ki a webszerver adatbázisából.

Az adatok megjelenítése webes felületen, internetes böngészőprogram segítségével történik. Az oldal URL címét beírva a címsorba, egy egyszerű két menüpontból álló weblap jelenik meg. Az egyik menüpont a *Teljes eseménylista*, mely táblázatos formában jelenít meg valamennyi kártyakezelést csökkenő sorendben (DESC), azaz a weboldal első sorában jelenik meg az utolsó kártyakezelés. A táblázat tartalmazza az ID azonosítót (adatbázisban automatikusan generált ismétlődés nélküli növekvő sorszám), az UID azonosítót (az RFID eszköz gyárilag kódolt azonosítója), a nevet, valamint kártyahasználat időpontját.

Shome.testhosting.hu/index.phi × +												
$\leftarrow$ $\rightarrow$ C $\bigcirc$ $\land$ Nem b	는 - > C 🍙 🛆 Nem biztonságos shome.testhosting.hu/index.php 으 🗘 🛱 Inkognitó											
Teljes eseménylista Keresés												
		reije	es esemei	nyiista								
	ID UID Név Kezelési időpont											
	52         7DB91CC3         Mel Gibson         2024-04-04 21:06:48											
	51	7DB91CC3	Mel Gibson	2024-04-01 19:52:02								
	50	B26AA383	Sárga Kártya	2024-04-01 19:51:09								
	49	7326FD12	Kék Kártya	2024-04-01 19:49:52								

**23. ábra:** Webfelület a kezelések megjelenítésére (Forrás: saját képernyőkép, Chrome böngésző)

A második menüpont a *Keresés*. Itt van lehetőség az adatbázis adatainak lekérdezésére egy egyszerű űrlap segítségével. Névre és dátumok közötti esemény keresésre van lehetőség. Az opciókat beírva, majd a *Keress…* gombra kattintva, külön táblázatban jelennek meg a keresési feltételek, alatta a találatok száma, valamint táblázatos formában az események növekvő sorrendben (ASC)

Új keresési adatok megadásával a régi keresési feltételek és találatok törlődnek és újra indul a lekérdezés.

v S shome.testhosting.hu/keres.ph	× +						— C	x נ		
$\epsilon  o \mathbf{C}$ $\widehat{\mathbf{m}}$ $\mathbb{A}$ Nem biztons	ságos s	home.testhosting.h	u/keres.php?ker_n	ev=&kezd_datum=2024-04-0	4&veg_da @	२ 🖈	🔒 Inkogni	tó :		
Teljes eseménylista Keresés								Ì		
			<u>Keresé</u>	S				- 1		
C	S== N	1el Gibson				- 1				
l d	2	2024.04.04.						- 1		
l i	2	2024. 04. 04.						- 1		
			Keresés					- 1		
		Ke	resési felt	ételek:						
	ID	Dátum-tól		Dátum-ig				- 1		
		2024-04-04				- 1				
-	_							- 1		
	Találatok:									
	Találat	ok száma: 1 db.								
	ID	UID	Név	Kezelési idő						
	52 7DB91CC3 Mel Gibson 2024-04-04 21:06:48									
shome.testhosting.hu/keres.php	-							-		

**24. ábra:** Webfelület a keresés megjelenítésére (Forrás: saját képernyőkép, Chrome böngésző)

#### 3.4 Fejlesztői környezetek telepítése, a programok belső működése

A projekt működéséhez több egymással kommunikáló, önálló program megírására volt szükség.

- 1. program a számítógépre, mely alkalmas a felhasználótól adatokat bekérni és továbbítani arra a mikrovezérlőre,
- program az író mikrovezérlőre (ESP8266), mely az adatírást fizikailag végrehajtja az íróeszköz segítségével,
- 3. program az olvasó mikrovezérlőre (ESP32), mely az adatolvasást hajtja végre, csatlakozik a nyílt internethez, és elküldi az adatokat,
- program a webszerverre, mely az adatok feldolgozását és megjelenítését végzi internetböngésző segítségével.

#### A GUI program felépítése.

Az 1. pontban meghatározott program C# programnyelven készült, Visual Studio 2022 fejlesztői környezetben. Egy Windows Forms App (.NET Framework) projekt segítségével készült el a GUI és a telepíthető állományok. A program megírásához, az alapértelmezett osztályokon kívül szükséges volt a számítógép portjainak kezelésére használható System.lo.Ports névtér osztály is.



**25. ábra:** System.IO.Ports class (Forrás: saját képernyőkép, Visual Studio 2022)

A programban először deklarálásra kerülnek a változók, majd pedig inicializálásra kerül az űrlap. Betöltéskor a *Kapcsolódás az íróhoz* gomb engedélyezésre, a *Kapcsolat bontása* és az *Írás kártyára* gombok tiltásra kerülnek, valamint piros színnel kerül megjelenítésre a *NINCS KAPCSOLAT* üzenet. A *Kapcsolódás az íróhoz* gomb hívja meg azt a metódust, amely meghatározza a soros port-ot (COM5), annak sebességét (9600 bit/sec), beállítja a karakterkódolást (Encoding.UTF8), valamit gondoskodik a vizuális megjelenítésről, végül megnyitja a soros port-ot (serialPort1.Open()). A *Kapcsolat bontása* gombhoz rendelt metódus a vizuális elemek megjelenítésén túl még a port bezárását végzi el (serialPort1.Close()), az űrlap végleges bezárását végrehajtó "*X*", pedig csak port bezárást.

Az *Írás kártyára* gomb metódusa a beviteli textbox adatait beolvassa egy változóba és küldi a soros port-ra (serialPort1.Write(dataOut);) Mind a négy esetben a hibakezelést a try-catch utasításpár kezeli és küld szükség esetén hibaüzenetet.

További három metódus van, ami a mikrovezérlőről a soros port-ra érkező adatokat kiolvassa és a *Soros port monitor*-nak elnevezett richTextBox-ban, scrollorozható módon meg is jeleníti.

#### Mikrokontrollerek programozása.

Az 2. és 3. pontban meghatározott program az Arduino platformhoz fejlesztett nyelven készült. Ez a programnyelv egyedi, de a C nyelvekre jellemző szerkezeti elemekre és struktúrára épül. Ez egy általánosan elfogadott környezet mikrovezérlők kezelésére. Rendelkezik kódírásra alkalmas felülettel és Serial Monitor-ral, mely segítségével a működés azonnal monitorozható (ld.: 19. kép).

#### A kártyaíró program felépítése.

Az alap könyvtárakon kívül a programba include-olni kell az SP kommunikációhoz szükséges könyvtárat és a kártya író/olvasó eszköz programozásához szükséges könyvtárat (<SPI.h>, <MFRC522.h>). Utána történik a mikrovezérlők tűinek a definiálása, a szükséges változók deklarálása, illetve a kezdőértékadás. Itt történik meg annak meghatározása is, hogy az űrlapon beírt adatok, a kártya melyik blokkjába kerüljenek beírásra (esetünkben a 2.).

A következő szakaszban el kell végezni az eszköz beállításait, inicializálni kell a kártyaolvasót, illetve azt a bitsebességet, amit a kommunikáció során használunk. Fontos, hogy a C# programban is és a mikrovezérlő programjában is ugyanazt a bitsebességet kell beállítani (9600 bit/sec). Eltérő sebesség esetén a kommunikáció nem működik.

A következő rész a void loop() függvény, mely folyamatosan ismétlődik, így képes figyelni a soros port-ot és szükség esetén függvényt hívni. Ha adat érkezik azokat betölti egy sztring változóba és visszaküldi a számítógépre, ahol megjelenítésre került a *Soros port monitor*-on,

20

továbbá meghívja a kártya írását végző függvényt, mely először a sztringet átírja hexadecimális alakban, majd beírja a fentebb meghatározott blokkba. Külön definiált függvény végzi az a már megírt RFID eszközre írt adatok ellenőrzését. Kiolvassa az adatokat és visszaküldi a soros porton keresztül a *Soros port monitor*-ra. Ezzel az írási folyamatnak vége.

#### A kártyaolvasó program felépítése.

Az olvasási oldalon az alapértelmezett könyvtárakon, a kommunikációhoz és a kártyaolvasó eszköz működéséhez szükséges könyvtárakon kívül importálni kell a dátum és idő kezeléséhez szükséges könyvtárat is (<time.h>). Ez az NTP (Network Time Protocol) időadat lekéréséhez, illetve a kártyakezelés időpontjának pontos megállapításához szükséges. Továbbá külön könyvtárra van szükség a wifi kapcsolat felépítéséhez, illetve a http kapcsolat kezeléséhez is. (<WiFi.h>, <HTTPClient.h>)

A program következő részében szintén szükséges a mikrovezérlő csapjainak és az alkalmazott változók definiálása. Itt kerül deklarálásra egy szabadon választott biztonsági kulcs, mely 13 karakterből áll és az interneten történő adattovábbítás azonosítására szolgál, valamint itt kerül meghatározása az a blokk is (2), amelyből ki kell olvasni a megfelelő adatot.

Program következő részében itt is a beállítások történnek meg. Így meg kell határozni a soros port sebességét és be kell beállítani wifi kapcsolatot. Itt kell definiálni egy változóban annak a php fájnak az elérési útvonalát is, ami a webszerveren fogadja az olvasást végző mikrovezérlő által küldött adatokat.

#### (const char serverName = "http://shome.testhosting.hu/post-data.php")

A program törzsében először lekérése kerül az NTP időbélyeg, majd pedig ellenőrzésre kerül a wifi kapcsolatos státusza. Ha nincs wifi kapcsolat felépítve, akkor megkísérni az olvasó mikrovezérlő az újra csatlakozás, addig amíg nem lesz működő hálózat. Ez a programfutás során is folyamatosan vizsgálatra kerül. Annak megszakadása esetén újra csatlakozik a mikrovezérlő a routerhez.

Ezután az előre definiált blokkból egy ciklus kiolvassa az adatokat, majd átalakítja a sztringé.

Ekkor következik egy POST üzenet összeállítása, mely tartalmazza biztonsági kulcsot, a kártya UID azonosítóját, a sztringet ami az adott blokkból kiolvasásra került, valamint a NTP alapú kezelési időt.

21

A világhálóhoz, a wifi-n és a helyi routeren csatlakoztatott mikrovezérlő, az adatokat továbbítja webszerverre, hol annak feldolgozása megkezdődik.

A fejlesztői felület soros monitorán, a folyamat végig követhető. Így látható a wifi kapcsolat sikeressége, illetve a kapott helyi IP cím.

**26. ábra:** A létrejött wifi kapcsolat és a helyben kiosztott IP cím (Forrás: saját képernyőkép, Arduino IDE – serial monitor)

Látható továbbá a kártya olvasásának folyamata, az összeállított POST üzenet tartalma és a küldés sikerességére vonatkozó http válaszkód.

Card UID: 7D B9 1C C3																		
Card SAK: 08																		
PICC type: MIFARE 1KB																		
Secto	r 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	[001]
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[000]
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[000]
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[000]
14	59	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[001]
	58	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[000]
	57	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[000]
			~~~	~~~			~~~	DD			~~~	DD	DE		DD			
1	7	00	00	00	00	00	00	F.F.	07	80	69	F.F.	FF	FF	F.F.	FF	FF	[001]
	6	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[000]
	5	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[000]
	4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[000]
0	3	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[001]
	2	4D	65	6C	20	47	69	62	73	6F	6E	00	00	00	00	00	00	[000]
	1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[000]
	0	7D	В9	1C	С3	1B	08	04	00	62	63	64	65	66	67	68	69	[000]
Data	in Block	:2	>	Me	l Gi	bso	n											
7DB91	.CC3																	
2024	2024 April 04, Thursday 21:16:41																	
httpRequestData: api_key=tPmAT5Ab3j7F9&uid_az=7DB91CC3&nev=Mel Gibson&kez_ido=1712258195																		
HTTP	HTTP Response code: 200																	

**27. ábra:** Írási folyamat és POST üzenet küldése (Forrás: saját képernyőkép, Arduino IDE – serial monitor)

Hibátlan adatátvitel esetén a 200-as kódot kapjuk vissza.

#### Az adatok feldolgozása és megjelenítése a webszerveren.

A webszerverre küldött adatok feldolgozásához szükséges html és php kódok Visual Studio Code fejlesztői környezetben készültek. Az alapbeállításokon túl egyéb összetevő telepítésére nem volt szükség.

Ahhoz, hogy az adatok feldolgozása rendben megtörténjen a web szerveren először létre kell hozni egy megfelelő tartalmú adatbázist.

Jelen esetben az adatok beírása *rfid* adatbázis, *adatok* táblájába történik az alábbi szerkezetben:





Az érkező adatokat, a már említett php fájl (post-data.php) fogadja. Ebben kerül definiálása ugyanaz a biztonsági kulcs, ami az adatolvasó mikrovezérlőjében.

A program először összeveti a két biztonsági kulcsot és ha egyezik, akkor folytatja a műveletet. Az érkező UID azonosítót, nevet és kezelési időt egy-egy változóba írja. Az egyszerű keresési funkció biztosítására két dátum változó kerül deklarálásra. Az egyik az óra percet is tartalmazza, a másik pedig csak a napot. Mindkettő a POST üzenetben beérkező kezelési időből kerül beállításra.

Ezután a sikeres adatbázishoz történő csatlakozás esetén az érkező adatok beírásra kerülnek az *adatok* táblába, majd a művelet befejezése után pedig bezárásra kerül az adatbázis kapcsolat.

Az adatbázisadatok vizuális megjelenítését táblázatos formában, html és php kód biztosítja, mely magába foglalja az sql lekérdezéseket is.

#### 3.5 Fejlesztési lehetőségek röviden

A projekt mind szoftveresen, mint hardveresen továbbfejleszthető.

Szoftveres fejlesztés tekintetében elképzelhető lehet a lekérdezések további finomítása, akár névre, akár napszakra, időpontra, hónapra vetítve előre definiált listák formájában. A lekérdező űrlapra felvehető még a kártya egyedi kódja, az UID azonosítója. Jelenleg egy adatblokk használata történik meg csak, de az 1 KB-os eszközön további 63 blokk van, ami ugyan nem írható 100%-ban, de jelentős mozgásteret biztosít további adatok rögzítésére. Azaz, ha a program további űrlapmezőkkel bővítésre kerül, akkor a beírt adatok, mind rákerülhetnek a RFID eszközre, annak kapacitásáig.

Hardveres tekintetben hozzárendelhető az olvasó oldali blokkhoz például megjelenítéshez bármilyen vizuális eszköz például led vagy LCD kijelző, mely jelzi a kiolvasási folyamat végét, vagy például a jogosultság hiányát vagy megfelelőségét, a wifi kapcsolat felépülését stb. Fejlesztési irány lehet például elektromos zárt beüzemelése, mely nyílik a megfelelő jogosultság esetén vagy akár rögzítheti a rendszer a tiltott kártyakezelést is.

Figyelembe véve, hogy a mikrovezérlőknek szinte minden csapja szabadon programozható kiés bemenetre egyaránt, így egy eszközzel számtalan lehetőség nyílik. Ezek megvalósítása elsősorban programozást igényel, illetve minimális hardveres fejlesztést.

24

# 4. Összefoglalás

A projektben sikeresen megvalósult a célkitűzésben megfogalmazott, szimulált elképzelés. A GUI-n keresztül megszemélyesített RFID kártya adatai kiolvashatók és továbbíthatók egy távoli szerverre, ahol biztosított azok megjelenítése és korlátozott visszakeresése.

A projektben is alkalmazott mikrovezérlők számtalan egyéb lehetőséget is biztosítanak teljesen más témában is. Fel lehet dolgozni velük szenzorok adatait és hasonló formában meg lehet jeleníteni; kapcsolni lehet hozzájuk mobiltelefonos applikációt szintén az adatok megjelenítésére; különböző eszközöket lehet ki- és bekapcsolni, illetve szabályozni; alkalmasak motorvezérlésre; az egymás közötti mind rádiós, mind bluetooth-os közvetlen kapcsolatra; vagy akár minimális webserver is telepíthető rá, így működőképessé tehető velük egy szerver-kliens kapcsolat.

# 5. Felhasznált irodalom

- aharshac repository. Letöltés dátuma: 2024. 02. 02. forrás: <u>https://github.com/aharshac/EasyNTPClient</u>
- arduino.cc. Letöltés dátuma: 2024.02.11. forrás: https://docs.arduino.cc/programming/
- arduino.cc. Letöltés dátuma: 2024. 02. 10. forrás: <u>https://www.arduino.cc/reference/en/</u>
- arduino.cc. Letöltés dátuma: 2024. 01. 22. forrás: <u>https://www.arduino.cc/reference/en/libraries/easy-mfrc522/</u>
- BoostMyTool csatorna. Letöltés dátuma: 2024. 03. 10. forrás: https://www.youtube.com/watch?v=NOkBUoP54b8
- electronicshub.org. Letöltés dátuma: 2024. 03. 08. forrás: https://www.electronicshub.org/arduino-rc522-rfid-module/
- www.instructables.com. Letöltés dátuma: 2024. 01. 21. forrás: https://www.instructables.com/C-Serial-Communication-With-Arduino/
- Last Minute Engineers. Letöltés dátuma: 2024. 02. 05. forrás: <u>https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/</u>
- 9. www.nxp.com. Letöltés dátuma: 2024. 01. 22. forrás: https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf
- 10. Random Nerd Tutorials honlapja. Letöltés dátuma: 2024. 03. 25. forrás: <u>https://randomnerdtutorials.com/security-access-using-mfrc522-rfid-reader-with-arduino/</u>
- 11. Random Nerd Tutorials honlapja. Letöltés dátuma: 2024. 03. 25. forrás: https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/
- 12. Random Nerd Tutorials honlapja. Letöltés dátuma: 2024. 01. 20. forrás: https://randomnerdtutorials.com/projects-esp32/
- 13. Wikipédia. Letöltés dátuma: 2024. 04. 05. forrás: <u>https://hu.wikipedia.org/wiki/Mikrovez%C3%A9rl%C5%91</u>
- 14. zintechideas repository Letöltés dátuma: 2024. 03. 12. forrás: https://github.com/zintechideas/RFID\_Arduino/blob/main/RFID\_write.ino

# 6. Ábrajegyzék

1. ábra: Elvi felépítés
2. ábra: ESP8266, ESP32 mikrovezérlő5
3. ábra: MFRC522 RFID író/olvasó eszköz 6
4. ábra: RFID kártya, kulcstartó, csuklópánt7
5. ábra: RFID eszközök adatstruktúrája7
6. ábra: Mikrovezérlők bekötése
7. ábra: MFRC522 pin kiosztása
8. ábra: Létrehozott telepítő fájlok9
9. ábra: RFID_write_setup program telepítése – 1. lépés9
10. ábra: RFID_write_setup program telepítése – 2. lépés
11. ábra: RFID_write_setup program telepítése – 3. lépés
12. ábra: RFID_write_setup program telepítése – 4. lépés
13. ábra: RFID_write_setup program telepítése – 5. lépés
14. ábra: Kártyaíró program a programindítás után12
15. ábra: Kártyaíró program a kapcsolat felépítése után
16. ábra: Kártyaíró program hiba esetén 13
17. ábra: Kártyaíró program az adatbevitel után, közvetlenül a fizikai adatírás előtt 14
18. ábra: Kártyaíró program a kártyaírás közben – 1 14
19. ábra: Kártyaíró program a kártyaírás közben – 2 15
20. ábra: Kártyaíró program a kártyaírás befejezése után
21. ábra: Adat ellenőrzés online konvertáló program felhasználásával
22. ábra: A fejlesztői környezet 16
23. ábra: Webfelület a kezelések megjelenítésére 17
24. ábra: Webfelület a keresés megjelenítésére18
25. ábra: System.IO.Ports class 19
26. ábra: A létrejött wifi kapcsolat és a helyben kiosztott IP cím
27. ábra: Írási folyamat és POST üzenet küldése 22
28. ábra: Webszerver adatbázisának a szerkezete 23

#### NYILATKOZAT

#### a záródolgozat nyilvános hozzáféréséről és eredetiségéről

A hallgató neve:	Major Tamás József
A Hallgató Neptun kódja:	GHFOSV
A dolgozat címe:	RFID eszköz megszemélyesítése mikrokontroller segítségével, valamint az adatok kiolvasása és küldése webszerverre
A megjelenés éve:	2024
A konzulens intézetének neve:	Műszaki Intézet
A konzulens tanszékének a neve:	Alkalmazott Informatika Tanszék

Kijelentem, hogy az általam benyújtott záródolgozat egyéni, eredeti jellegű, saját szellemi alkotásom. Azon részeket, melyeket más szerzők munkájából vettem át, egyértelműen megjelöltem, és az irodalomjegyzékben szerepeltettem.

Ha a fenti nyilatkozattal valótlant állítottam, tudomásul veszem, hogy a záróvizsga-bizottság a záróvizsgából kizár és a záróvizsgát csak új dolgozat készítése után tehetek.

A leadott dolgozat, mely PDF dokumentum, szerkesztését nem, megtekintését és nyomtatását engedélyezem.

Tudomásul veszem, hogy az általam készített dolgozatra, mint szellemi alkotás felhasználására, hasznosítására a Magyar Agrár- és Élettudományi Egyetem mindenkori szellemitulajdon-kezelési szabályzatában megfogalmazottak érvényesek.

Tudomásul veszem, hogy dolgozatom elektronikus változata feltöltésre kerül a Magyar Agrárés Élettudományi Egyetem könyvtári repozitori rendszerébe. Tudomásul veszem, hogy a megvédett és

- nem titkosított dolgozat a védést követően

- titkosításra engedélyezett dolgozat a benyújtásától számított 5 év eltelte után nyilvánosan elérhető és kereshető lesz az Egyetem könyvári repozitori rendszerében.

Kelt: Gyöngyös, 2024 év április hó 21 nap

Hallgató aláírása

#### NYILATKOZAT

Major Tamás (hallgató Neptun azonosítója: GHFOSV) konzulenseként nyilatkozom arról, hogy a zárodolgozatot áttekintettem, a hallgatót az irodalmi források korrekt kezelésének követelményeiről, " jogi és etikai szabályairól tájékoztattam.

A záródolgozatot a záróvizsgán történő védésre javaslom / nem javaslom<sup>1</sup>.

A dolgozat állam- vagy szolgálati titkot tartalmaz:

igen <u>nem\*</u>2

2

\*.

Kelt: 2024 év április hó 20. nap

belső konzulens

<sup>1</sup> A megfelelő aláhúzandó.
 <sup>2</sup> A megfelelő aláhúzandó.