

SZAKDOLGOZAT

Albert Áron

2022

**Magyar Agrár- és Élettudományi Egyetem
Élelmiszertudományi és Technológiai Intézet
Élelmiszeripari Méréstechnika és Automatizálás Tanszék**

**Termelés nyomonkövethetőség
a Dianás cukorka gyártásban
mikrokontrollerek segítségével**

Albert Áron

Budapest

2022

Magyar Agrár- és Élettudományi Egyetem
Élelmiszertudományi és Technológiai Intézet

Szak neve: BSc Élelmiszermérnöki
Élelmiszertechnológiai Automatizálás és Digitalizáció
Szakedolgozat készítés helye: Élelmiszeripari Méréstechnika és Automatizálás

Hallgató:

Albert Áron

A szakedolgozat címe:

Termelés nyomonkövethetőség a Dianás cukorka gyártásban mikrokontrollerek segítségével

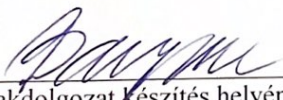
Konzulens:

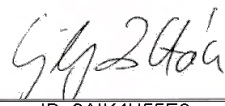
Dr. Gillay Zoltán

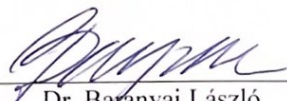
Külső konzulens esetén tanszéki felelős:

Beadás dátuma:

2022.11.02.


szakedolgozat készítés helyének vezetője
Dr. Baranyai László


Signer ID: 9A1K4U55E8
konzulens
Dr. Gillay Zoltán


Dr. Baranyai László
Élelmiszertechnológiai Automatizálás és Digitalizáció

Tartalom

1. Bevezetés	5
2. Célkitűzés	6
3. Irodalmi áttekintés	7
3.1 Dianás cukorka gyártása	7
3.2 Dianás cukor csomagolása	8
3.3 Automatizálási lehetőségek napjainkban	9
3.3.1 Ipari mikrokontrollerek	10
3.3.2 ARDUINO, Raspberry Pi	11
3.3.3 Vállalatirányítási rendszerek	12
3.3.4 RFID	16
3.4 Vezetői döntéshozatal és Operatív HR	18
3.4.1 Operatív HR	20
4 Anyagok és módszerek	21
4.1 Anyagok	21
4.1.1 Segédanyagok	24
5 Eredmények	25
5.1 Fizikai beüzemelés	25
5.1.1 MH-ET Live 3.0 Vonalkód olvasó	25
5.1.2 Arduino UNO R3 és Raspberry Pi Model 3B+	26
5.2 Program felépítése	28
5.2.1 MATLAB tanulmányozása	30
5.2.3 Kapcsolat létrehozása az eszközökkel	37
5.2.4 Adatok átvitele IDE-k között	38
5.2.5 Műveletek elvégzése az adatokkal	40
5.2.6 Adatok mentése	41
5.3 Költségelemzés	42
5.4 Vélemények	42
5.5 További fejlesztések a következő verzióhoz	43
5.5.1 Emberi tényező	43
5.5.2 Internet of Things	43
5.5.3 RFID	44
6. Összefoglalás	46
7. Hivatkozások	47
7.1 Internetes hivatkozások	48

1. Bevezetés

Az Ipar 4.0 (más néven 4. ipari forradalom) megjelenésével a technológiai fejlődés és fejlesztés rohamos iramban kezdett el nőni a különböző iparágakban. Napjainkban láthatóak olyan komplex rendszerek, melyekben mesterséges intelligenciák irányítják és szabályozzák a gyártás folyamatát vagy olyanok, melyekben vállalatok egymás között adnak át hatalmas mennyiségű információt másodpercek alatt.

Ezek előnye, hogy a termelési folyamatok egyre pontosabb szabályrendszerekhez vannak kötve. Ez által a kész termékek minősége egyre magasabb, és egyre több információ gyűjthető be a gyártásról. A begyűjtött információk alapján a vállalatok olyan stratégiákat tudnak alkalmazni, amelyekkel fentebb pozícionálhatják magukat a világgazdaság versenyében.

Ezért az Ipar 4.0 megjelenése hátránnyá vált az olyan vállalatok számára, amik lemaradással küzdenek technológiai fejlettségük terén versenytársaikkal szemben. Egyike ezen vállalatoknak a Ponacco Hungária Kft., melyet a családom alapított 1992-ben, és napjainkban is Dianás cukorka gyártással foglalkozik. A cég örököséként minden gyártási fázisban részt veszek, illetve új megoldásokkal fejleszthetem az üzemet. Az üzemben folyamatosan törekszünk a fejlődésre, az évek során fő célunk az egyes folyamatok gépesítése volt, mely az emberi munkaerő lecserélését jelenti gépekre. Bizonyos folyamatokhoz szükség van az emberi munkaerőre, ami néhány esetben hátrányt jelent.

Úgy vélem, ha egy vállalat minimális feldolgozható adattal rendelkezik, akár egyetlen gyártási- vagy gyártásközi folyamat adatainak begyűjtése és rendszerezése is pozitív változásokat hozhat a cég teljesítményében. Ezek az adatgyűjtések lehetnek akár termékre- akár munkaerőre vonatkozóak. Ehhez viszont szükség van azon berendezések betelepítésére, melyek minimális emberi erőbefektetéssel tudnak adatokat begyűjteni.

2. Célkitűzés

Célom a bevezetés alapján egy olyan rendszer megépítése volt, mely cégünk egyik gyártási folyamatának termelési adatait gyűjti be.

Az általam választott gyártási folyamat a cukorkaszemek becsomagolása. Ez bizonyul a cég azon gyenge pontjának, mely minimális adatgyűjtéssel rendelkezik, illetve ezek a begyűjtött adatok is a dolgozók saját bevallása alapján közölt termelési adatok.

Ezek alapján a feladatom egy olyan rendszer megalkotása volt mely:

- Begyűjti a dolgozók termelési adatait,
- rendszerezi a begyűjtött adatokat,
- pontos információkkal szolgál a vezetőség számára,
- költséghatékony,
- illetve kellően egyszerű ahhoz, hogy gyorsan beilleszthető legyen a gyártásba.

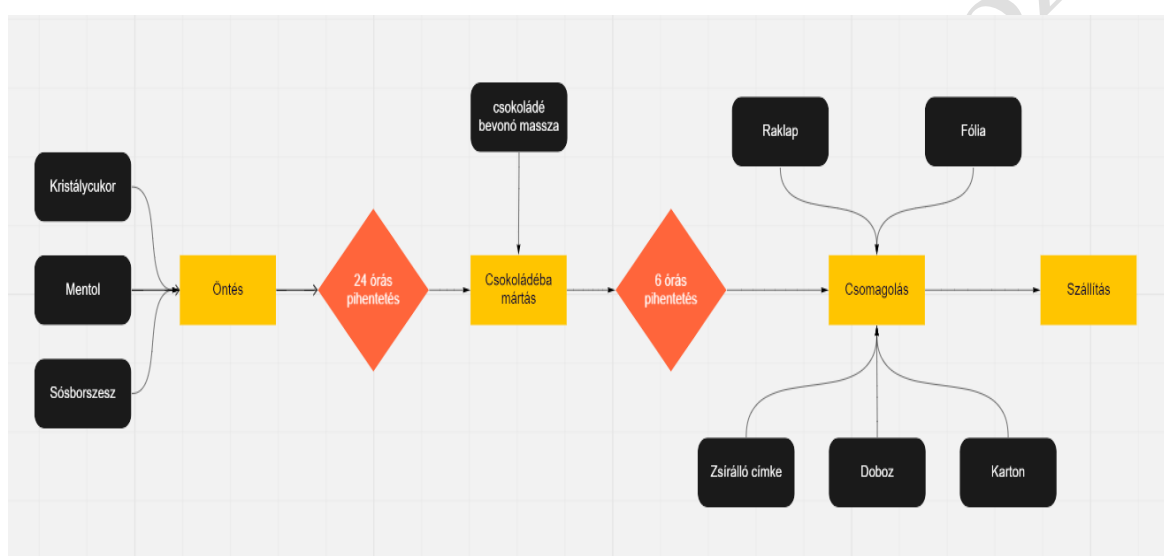
Az előbb felsorolt célok mellett saját célom, hogy egy olyan fejlesztési alappal szolgáljon mely folyamatosan bővíthető a nélkül, hogy vállalatunknak át kelljen terveznie az egész rendszert vagy saját gyártási folyamatát.

3. Irodalmi áttekintés

A Dianás cukorkát 1907-ben kezdték el forgalmazni és ipari mennyiségben gyártani. Az elnevezés a „Diana sósborszesz és kozmetikai cikkek” nevű üzlet miatt maradt meg a köztudatban, mivel itt volt kapható a cukorka (Internet 1). A sósborszesz és mentol tartalmú csokoládéval borított cukorkákat mára csak csekély mennyiségű vállalat gyártja, főként bérgyártásban nagykereskedelmeknek.

3.1 Dianás cukorka gyártása

A Dianás cukorka gyártásához a segédanyagokkal és csomagoláshoz szükséges anyagokkal együtt 5 alapanyagra van szükség.



1. ábra: Dianás cukorka leegyszerűsített gyártási folyamata (Ponacco, 2022)

A termék elkészítése három fázisból áll. Először a mentolt, cukrot, vizet, és sósborszeszt felhasználva, forrpontra melegítik, majd egy erre a célra készített géppel „kiöntik”. Így térnek át a cukorkák a pihentetési fázisra. A termék formájának megtartásához és megfelelő ütemű megkeményedéséhez különböző pihentetési technikák léteznek. Az üzemünk által alkalmazott technika, a 24 órás pihentetést választotta, hogy a cukorkák külső felülete megkeményedjen, belseje pedig folyékony állapotban maradjon. A pihentetés és a nem-megfelelő szemek kiválogatása után, a cukrok átkerülnek a „mártó-egységbe”. A mártó egység egy mártó berendezéssel, illetve egy hűtőalagúttal rendelkezik, ilyen berendezéshez hasonló gépek beszerelésével foglalkozik a Foodgalnz vállalat (Internet 2). A hűtőalagútból kiérkezve, a szemek újabb pihentetési fázison esnek át egy hűtött raktárban a további szilárduláshoz. Ez az üzemünk mérései alapján nagyjából hat órát vesz igénybe. A pihentetés

után a cukorszemek kellően megszilárdulnak, és csomagolható állapotba kerülnek. A csomagolás után a termék készen áll a szállításra.

3.2 Dianás cukor csomagolása

A csomagolási fázis az alapanyagok felhasználásának szempontjából 3 részmunkára bontható. Első sorban a cukorka szemeteket egy csomagológép segítségével a dolgozók szemenként becsomagolják zsírálló címkébe. Ehhez a feladathoz, erre a célra kifejlesztett, félautomata csomagológépeket használnak. Működése 8 állomásból áll, mely állomások közben a munkafolyamat a következőképp történik. A gép felveszi a cukorka szemeteket, körbe tekeri a címkével, majd a címke két végét megfogva becsavarja a szélét. Ez az úgynevezett „dupla-csavarásos” technika. A becsomagolt szemeteket a gép kivezeti egy gyűjtő tálcába, ami elérhető a dolgozók számára. A dolgozók előhajtják a dobozokat, majd megtöltik azokat cukrokkal. Lemérik a dobozok súlyát, majd lezárják a dobozt, és kartonba rakják. Ez után a kartonokat raklapra teszik, körbe tekerik sztreccsfóliával, és előkészítik a szállításhoz szükséges dokumentumokat.

A bérgyártás értelmében az üzem több nagyvállalatnak is szállít terméket, illetve a termékeket nem a saját márkanévvel látjuk el. Ezek értelmében a cukrokat tartalmazó dobozok is eltérőek. Ezeket a méretük, dizájnjuk, vagy a dobozon található vonalkód különbözteti meg. A vállalatunk számára egy termelési folyamat nyomon követéséhez a vonalkódok beolvasása a legkifinomultabb megoldás, ezért fontos alapvető ismerettel rendelkezni a vonalkódokról. A Wikipédia definíciója szerint: „A vonalkód olyan, gépekkel optikailag leolvasható kód, amelynél különböző vastagságú függőleges sötét vonalak és világos közök meghatározott váltakozása fejezi ki az információt” (Internet 3). Az ident.hu weboldal alapján a vonalkódok 2 nagy csoportra bonthatók. Ezek a csoportok az egy-dimenziós és a két-dimenziós vonalkódok. Az egy-dimenziós vonalkódoknak is több alcsoportja van, az EAN 13 jelölést viselő típus a legelterjedtebb a kereskedelemben. Egy 13 karakterből álló kódot foglal magába, melynek első kettő vagy három vonala az országot jelöli, majd az azt követő karakterek pedig a régiót, helységet, termék típust, vagy akár gyártási időt. Minden ország rendelkezik a saját két vagy három számjegyű EAN kódjával, mely szintén megtalálható a weboldalon. Ahogy már említésre került, több fajtája is létezik az egy-dimenziós vonalkódoknak, például UPC E, Code 128, vagy Code 39. Ezek a változatok mind hasonlítanak az EAN kódolású változatra, a bennük hordozható információ mértéke viszont a fajtahoz mérten eltérő lehet. (Internet 4). A 2. ábra mutatja az EAN 13 vonalkód felépítését. Az ábrán látható „5” szám jelöli a gyártó székhelyének jelzett országot

kódját, illetve a „994100” számsor jelöli a régiót illetve a települést, az „159988” számsor a termékkel kapcsolatos specifikációkat jelzi.



2. ábra: EAN 13 vonalkód (Internet 3)

Ezek megjelenése robbanásszerű fejlődést jelentett az iparban. A termékek származási helye ismeretében, a forgalmazók és a vásárlók pontos visszajelzéseket tudnak küldeni a gyártóknak, például termékhiba esetén. Ezzel a technológiával, akár több száz alapanyagot igénylő termék feldolgozása esetén is pontosan visszakövethető, mely alapanyag minősége nem felel meg a feldolgozó vagy a fogyasztó követelményeinek. Az ehhez hasonló technológiák megjelenése mind az ipar ugrásszerű fejlődését jelenti. Az ilyen eseményekre számos példát lehet felsorolni.

3.3 Automatizálási lehetőségek napjainkban

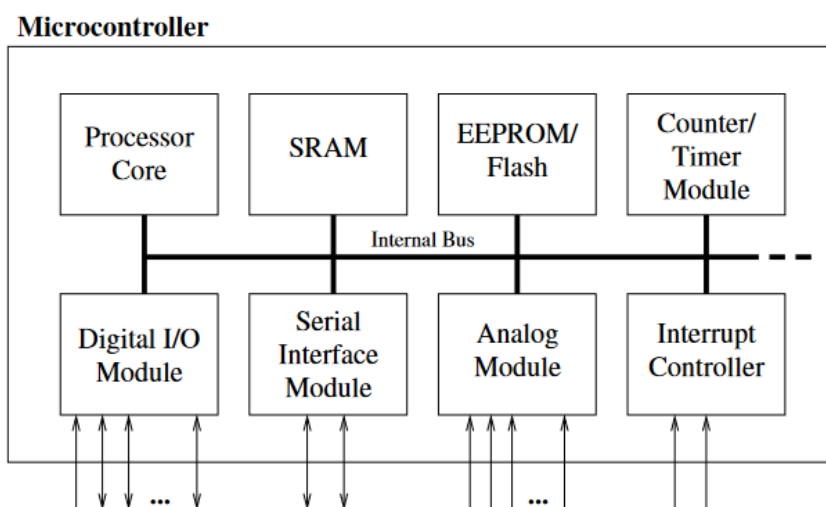
A Britannica online magazin (Internet 5) kategorizálva mutat be számos ipari alkalmazást az automatizálási lehetőségekre. Ezek között találhatóak olyan grandiózus megoldások is, mint egy teljesen önműködő gyár megtervezése és működtetése, robotok használata az ipar különböző területein, mesterséges intelligenciák által irányított alapanyag rendelések, kész áruk eladása, ...stb. Ugyan akkor olyan megoldásokról is hallhatunk, melyek egy már működő cég működését segítik. Érdeemes megemlíteni, hogy az automatizálás nem csak az iparban van jelen. A cikk értekezik az autopilóta üzemmód fogalmáról szállítástechnikában, telekommunikációs rendszerekről, egészségügyi rendszerekről, internetes hálózatokról, önkiszolgáló automatákról. Életünk minden percében találkozunk automatizált rendszerekkel és különböző megoldásokkal. Ahhoz, hogy lépést tudjunk tartani az ipar fejlődésével, nem csak a forradalmi anyagokat kell ismerniük a szakembereknek, hanem azoknak a gépeknek a működését is, melyekkel ezt az anyagot kezelik. Manapság a legelterjedtebb fogalmak az automatizáláson belül szorosan köthetők az Ipar4.0 fogalomtárához, ide tartoznak a termelésvégrehajtási rendszer (mint PLC), vállalatirányítási rendszerek (mint ERP, BI), Big Data, Dolgok hálózata (Internet of Things – IoT), ...stb. A

magyar, újonnan megalapított vállalatok már alkalmazzák az ezekkel a fogalmakkal társított megoldásokat, így a kis- és középvállalkozások folyamatos lemaradással küzdenek. Ahhoz, hogy a kisvállalkozások lépést tudjanak tartani a piac fejlődésével, szükségük van az automatizálási alapismeretek elsajátítására, hogy felzárkózzanak.

3.3.1 Ipari mikrokontrollerek

Első sorban az ipari mikrokontrollerek képzik az automatizálási folyamatok nagyrészének alapját. A Hubspire definíciója szerint a mikroprocesszorok olyan apró elektronikus berendezések melyekben megtalálható az aritmetika és logika, illetve áramkör kapcsolási sémára van szükség, hogy olyan funkciókat lásson el, mint a digitális számítógép központi feldolgozó egysége. A gyakorlatban ez az integrált áramkör képes feldolgozni és végrehajtani a programinstrukciókat is, az aritmetikus műveletek végrehajtásán felül (Internet 6). Ez a gyakorlatban, a gépek méretén mutatkozik meg. Az ipari berendezéseket a mikroprocesszorok megjelenésével egyre kompaktabb formában tudták legyártani. Ezek hatására jelentek meg a mikrokontrollerek. A két eszköz között látványos különbség csak az árban, operálási sebességben, és az áram használatban van (Internet 7). Dehner és Wheeler véleménye szerint két csoportra bonthatók a mikrokontrollerek. A végfelhasználókat nézve azok, akik nyílt forráskódú mikrokontrollereket használnak, saját technikai segélynyújtó csoportot hoztak létre fórumokkal és kisebb csoportokkal. Ez által egy automatizálási háttértudással nem rendelkező felhasználó vagy programozó rengeteg helyről találhat megoldást problémájára. Úgy gondolják az ipari mikrokontrollerek tökéletes megoldást nyújtanak nem-túl komplex feladatok megoldására, illetve megfizethetőbbek, mint az ipari automatizálásra optimalizált rendszerek, a PLC-k (Programmable Logic Controller – Programozható logikai vezérlő). A PLC-t előnyben részesítő felhasználók munkáját, a gyártó technikai segélynyújtó részlege próbálja támogatni. Így a programozható logikai vezérlők beüzemeléséről megtalálható tudás csekély mértékben van jelen az ingyenes forrásokon, például fórumokon (Dehner & Wheeler, 2017). Mára számos ipari-mikrokontroller található a piacon. Gridling és Weiss értekezése alapján a felhasználó által megoldandó feladat elvégzésének összetettségéhez mérten kell kiválasztani a megfelelő eszközt. A mikrokontrollerek ugyanis eltérnek adatfeldolgozási gyorsaságban, memória kapacitásban stb., alapvető belső felépítésük viszont megegyezik, ahogy ezt a 3. ábra mutatja. A kontrolleren belül található egyes részek, mint a számláló, időzítő, processzor, memória, analóg-, digitális- és soros interfész- modulok, egy belső busz-hálózattal vannak összekötve, illetve a beavatkozók (pl.: motorok, relék, kijelzők, ...stb.) csatlakoztatásához,

minden eszköz rendelkezik I/O digitális és analóg pin kimenetekkel (Gridling & Weiss, 2007). Ezeket a modulokat bővíteni is lehet, ha a felhasználónak szüksége van rá.



3. ábra: Ipari mikrokontroller belső felépítése. (Gridling és Weiss, 2007)

3.3.2 ARDUINO, Raspberry Pi

A széles választékból 2 manapság a két leggyakoribb mikrokontroller az Arduino és a Raspberry Pi.

Az Arduino egy olyan nyílt forráskódú elektronikus platform, amely fő profilja a felhasználóbarát, könnyen használható hardver és szoftver. Különböző kiegészítő modulokkal számos feladatot el tud látni, például hőmérséklet érzékelés, rádió frekvenciás nyomon követés, zárok működtetése, vagy akár egy Twitter üzenet elküldése. A hardver összekötése huzalozásos alapon működik, a programozáshoz pedig az Arduino saját fejlesztésű szoftverét kell letölteni, mely a feldolgozáson alapul (Internet 8). Az Arduino alapvetően a C/C++ programozási nyelvre épül. Az Arduino programok fejlesztésére rendelkezésre áll egy erre a célra fejlesztett Integrált Fejlesztői Környezet (IDE – Integrált Fejlesztői Környezet). Az IDE az a fejlesztői környezet, mely programok írására készült, és az általános fejlesztői eszközöket tartalmazza egy általános felhasználó felületen, azaz GUI-n (General User Interface) (Internet 9). Ezek mellett mivel nyílt forráskódú a szoftver, a felhasználói közösség rengeteg különböző saját fejlesztésű program csomagot hozott létre az Arduino hivatalos fórum oldalán (Internet 10). Hughes könyve alapján az Arduino mikrokontrollereket otthoni felhasználásra, kisebb projektek kivitelezésére tervezték,

felhasználó barát platformjuk és gazdaságosságuk miatt. Megjelenésük 2005-ben történt (Hughes, 2016).

Az ezt követő évben, egy brit mérnök megpróbálta az embereket közelebb hozni a számítástechnika világához, és elkezdett dolgozni egy olyan egykártyás számítógépen, amely megfizethető, és apró mérete miatt bárhova elvihető és beüzemelhető. 2012-ben így vitte piacra a Raspberry Pi számítógépeket (Patrick Fromaget, 2022). A Raspberrypi.com weboldal alapján ezek a bankkártya méretű számítógépek képesek minden olyan feladat ellátására, amit egy számítógéptől elvár az ember. Segíthetnek megismertetni a fiatalabb generációval a programozás világát, illetve digitális projektek kivitelezését (Internet 11). Internetes böngészésre, fejlesztő felületek futtatására, illetve az ezzel összeköthető eszközökkel való kommunikációra is képes. Ez által ipari alkalmazásra is tökéletes megoldásként szolgálhat, egyszerű feladatok ellátásához.

Az előzőekben felsorolt eszközökön kívül, mára számos megfizethető mikrokontroller kapható a piacon (mint ESP32, BeagleBone, ...stb.), mind a saját karakterisztikaival. Ipari alkalmazásra kisebb mértékben az Arduino illetve a Raspberry Pi tökéletesen alkalmas, nagyobb mennyiségű adat feldolgozására, vagy komplexebb automatizálási feladatok elvégzésére, ennél szofisztikáltabb illetve költségesebb eszközökre van szükség, hogyha a felhasználó a mikrokontrollerekkel szeretne dolgozni. Ezek az eszközök tehát előre megírt feladatokat tudnak teljesíteni, például egy gyártósoron belül. A belőlük kinyerhető és feldolgozható információhoz viszont szükség van egy strukturált környezetre, mely begyűjti ezeket az információkat, és adott esetben feldolgozza azokat. Az ipar a mikrokontrollerek idejére már rendelkezett ezekkel a szisztémákkal, melyeket vállalatirányítási rendszereknek hívnak.

3.3.3 Vállalatirányítási rendszerek

Ilyen mikrokontrollereket összekapcsolva egymással, az ipar eljuthatott a magasabb szintű automatizálásra, ami azt jelenti, hogy a rész munkafolyamatok leautomatizálása helyett, akár teljes munkafolyamatokat képesek lettek működtetni emberi beavatkozás nélkül. Ehhez a technológiai lépéshez köthető a vállalatirányítási rendszerek megjelenése. Tavana értekezése szerint az első vállalatirányítási rendszerek az 1960-as években jelentek meg először. Ezeket a rendszereket a vállalatok maguk tervezték és állították elő, a saját működésük javítására. Ezeket a rendszereket a raktárak működésének optimalizálására tervezték, Inventory Control (IC – Raktár Kontroll) programcsomagok segítségével. Az 1970-es években megjelentek az Essential Requirements Planning (MRP) rendszerek.

Ezekben a rendszerekben a helyiség vagy termék menedzsment egy széleskörű gyártási terv alapján működött. Az MRP rendszereket nevezhetjük a vállalat irányítási rendszerek elődjének. Ezek a rendszerek már képesek voltak a gyártással kapcsolatos folyamatok dokumentálására is. Az első vállalat irányítási rendszer az 1990-es években jelent meg, kombinálva az MRP és IC technológiákat. Így a Vállalat Irányítási (Enterprise Resource Planning, továbbiakban ERP) rendszerek képesek egy vállalat működtetésére, például gyártás, disztribúció, emberi erőforrások, projekt menedzsment, raktár menedzsment, logisztika, karbantartás. Az MRP és IC egyesítése egy rendszerbe az előbb felsoroltak mellett szintén létrehoz kezelhetőséget, illetve átláthatóságot a gyár dolgozói részére (Madjid Tavana et. el. 2020). Így a vállalat irányítási rendszerek létrehozása fontos szerepet játszik a modern vállalatok átalakításában, hogy azok jobban működjenek, illetve elérjék vagy fenntartsák üzleti versenyképességük.

Ezek értelmében mára a vállalatok maguk választhatnak különböző típusú vállalat irányítási rendszerekből. Matende és Ogao szerint ehhez számba kell venniük különböző kritikus sikerességi faktorokat (Critical Success Factor - CSF). Ezek a faktorok az üzembe helyezés és működés sikerességét befolyásolják. Minél több ilyen faktort hagy figyelmen kívül a vállalat, annál kevesebb az esélye az üzembe helyezett rendszer megfelelő működésének. Ilyen faktorok például az adattárolás helyszíne, szoftver nyelvezete, felső menedzsment támogatás, beavatkozás mennyisége, ellenőrző pontok felvétele, dokumentáció, felhasználó barát felület (Matende & Ogao, 2013; Tavana et al., 2020).

3.3.3.1 Felhasználói beavatkozás

A felhasználó barát felület minden faktor közül talán a legkritikusabb pont egy ERP rendszer bevezetésének sikerességéhez, mivel hatással van a felhasználókra a vállalat különböző szintjein, és többnyire a működéséhez különböző felhasználói beavatkozásokra van szükség. Matende és Ogao kutatása alapján az alacsonyabb beosztású dolgozók, értendő itt a gyártósoron dolgozó szakembereket, irodai munkásokat, szállítókat, akár napi szinten kapcsolatba lépnek a software-el. Mivel ezen munkások jelentős hányada nem rendelkezik informatikai háttértudással, vagy olyan készségekkel melyek segítségével gyorsan megértse egy software működését, elengedhetetlen a vállalat számára betanításuk. Ez a folyamat a software felhasználó-barátságához mérten akár hónapokkal is meghosszabbíthatja a sikeres üzembe helyezést (Matende & Ogao, 2013). A betanítási folyamat szoros kapcsolatban van a cég belső információ áramlásának hatékonyságával. Értendő itt a továbbképzések, dokumentációk gyakorisága, és ezek elérhetősége. Ha egy cégen belül a dokumentálás és továbbképzések foka alacsony, a vállalat lemarad versenytársaival szemben és ez hátrányt

jelenthet számára a piaci versenyben. Saide és Mahendrawathi szerint a probléma megszüntetésére a vállalat külsős beavatkozókat, például betanítókat vagy konzulenseket kér fel, ám ezen szervezetek felkérése is költségekkel jár melyek meghaladhatják a vállalatnak ezen célra félretett tőkéjét. Ezen felül felmerülhet, hogy a külsős betanító vállalatok vagy szakemberek sincsenek teljesen tisztában a rendszer működésével, ha annak felülete túl sok, egymástól lényegesen differenciáló szegmensből áll. Például egy egyszerű dokumentációnál a felhasználó belebonyolódhat a folyamatba, és sikertelenül hajtja végre feladatát, vagy olyan problémára talál egy megoldást munkája során, mely egy másik felhasználót érint. Így egy ERP rendszer sikeres implementálásához a cég működése is újra-tervezésre kerülhet (Saide & Mahendrawathi, 2015).

3.3.3.2 Business Intelligencia

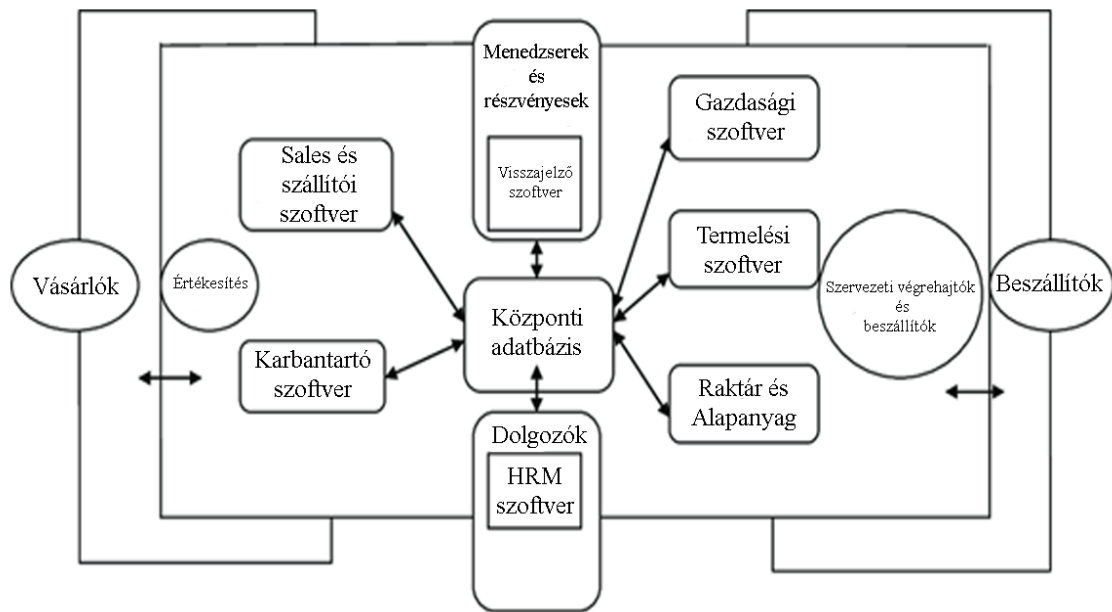
Gyakori esetben a dokumentációkat nem csak a dolgozók végzik az ERP rendszer segítségével, hanem a rendszer is automatikusan dokumentál bizonyos információkat. Ezek az információk lehetnek csekély fontossággal bíróak, vagy akár a vállalat további üzleti tevékenységét is meghatározhatják. Ezen információk rendszerezett és átlátható kinyeréséhez a vállalatok Business Intelligence (BI) software csomagokat fejlesztenek ki, melyek ezeket a feladatokat látják el. Negash alapján a Business Intelligence az a software mely begyűjti, átalakítja, és strukturált adatot szolgáltat különböző forrásokból (Negash, 2004). Nofal és Yusof definíciója alapján pedig a Business Intelligence rendszerekre a releváns információk rövid idejű elérése végett van szükség, a vállalat kapacitásának hatékony kihasználásához (Nofal & Yusof, 2013). Chou megállapítása szerint a Business Intelligence software-ek különböző folyamatokat hajtanak végre, mint adatok felfedezése, adatok kapcsolata, illetve az adatelemzéssel kapcsolatos trendek végrehajtása. Így ábrázol konklúziókat, melyek segítik a vállalati profit emelkedést, illetve a vállalaton belül történő munkavégzés hatékonyságát (Chou et al., 2005). Agostino megállapítása szerint így a Business Intelligence rendszerek számos előnnyel rendelkeznek, például a menedzserek valós időben tudják követni a vállalati pénzmozgást, fejlesztí és segíti az értékasítési osztály munkáját, támogatja a vásárlói kapcsolatok fenntartását, online felületen is elérhető adatokat szolgáltat melyek hozzáférési ideje sokkal rövidebb stb. (Agostino, 2004) Így a vállalat hatékonyabban tudja kezelni az ERP rendszer által szolgáltatott információkat. Az adatfeldolgozás után a vállalat a kinyert információkat valamilyen formában el szeretné tárolni, hogy további megfigyeléseket is végezhesen az idő előre haladtával.

3.3.3.3 IoT és lokális ERP rendszerek

Tavana és munkatársai két ERP rendszert mutatottak be, a felhő alapú-, illetve a lokális-ERP rendszereket. A lokális rendszer kiépítéséhez a vállalatnak szüksége van megfelelő adattároló eszközökre, melyekkel ezekhez hozzá tudnak férni, míg az IoT (Internet of Things) megjelenése óta, a felhő alapú ERP rendszernek nincs szüksége belső adattárolókra, begyűjtött adatait az internetre, pontosabban a vállalt birtokában lévő felhő rendszerbe feltöltve végezheti műveleteit az adatokkal. A valós idejű információ, előfeltétele a gyors megoldásoknak. Az ERP és IoT integrálása egyenes utat ad az időkésés nélküli adatbegyűjtésnek és az azonnali megoldásoknak egyszerre: ez az ERP-vel lehetséges, ami egyszerű megértést biztosít a fennálló eseményekről, míg az IoT a megfelelő megoldások javaslatával rendelkezik. Valójában, az IoT adathalmazokat készít, melyeket az ERP rendszerek segítségével rendszerezni lehet, a vállalat fejlesztésének céljaként (Tavana et al., 2020).

3.3.3.4 ERP működési sémája

A 4. ábra mutatja be a Davenport meghatározásait a vállalatirányítási rendszerek elemeiről. Ebben látható milyen feladatokat kell ideális esetben elvégeznie egy jól működő ERP-nek. Az alapanyag áramlása alapján 3 fő csoportot lehet megállapítani. A beszerzést-kezelő csoportot, a feldolgozó csoportot, illetve a kész terméket eladó csoportot. A vállalat működése alapján 7 csoport határozható meg. A felső vezetés, a gazdasági részleg, a gyártó részleg, a raktárt kezelő csoport, a humán erőforrások csoportja, a karbantartás illetve az eladásokkal foglalkozó részleg. (Davenport, 1998.) Egy ERP rendszer sikeres beüzemeléséhez így előnyként szolgálhat, ha a vállalat pontosan be tudja határolni a munkafolyamatai határait, így a rendszer egyes elemei és feladatai is pontosan behatárolhatóak.



4. ábra: Vállalatirányítási rendszer felépítése (Davenport, 1998).

Egy ERP rendszer tehát számos feladat elvégzésére képes. Akár a beérkező alapanyagoktól, a termék elkészüléséig nyomon követhetjük az általunk felhasznált anyag áramlását. Dokumentálhatjuk dolgozóink munkateljesítményét, vagy dolgozóként saját munkánkat ellenőrizhetjük. Ez előzőben taglaltak alapján tehát egy ilyen komplex rendszer kiépítése nagyon sok előkészületet igényel, és számos hibalehetőséggel kell számolni akár működése közben is. A vállalatoknak viszont szükségük van a termékeik és dolgozóik nyomon követésére a termelékenység maximalizálásához. Erre manapság igen sok részmegoldás is létezik, egy verziót mégis érdemes kiemelni.

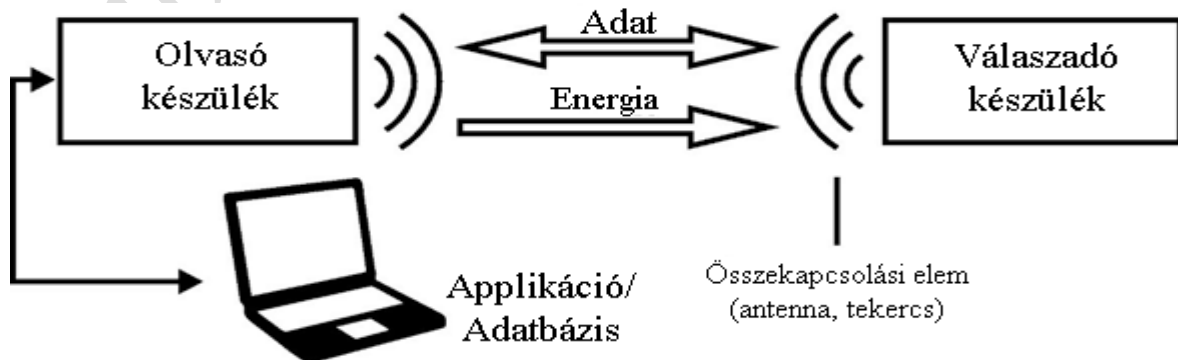
3.3.4 RFID

Parsons megfogalmazása szerint a nyomon követhetőség azon képesség, hogy egy terméket egy egyedi kóddal azonosítsanak, amely lehetővé teszi a szállító vagy a fogyasztó számára, hogy gyártási hiba vagy szennyeződés esetén visszakeresse a termék eredeti alkotórészeit. Az élelmiszeriparban ez magában foglalhatja azt is, hogy mely állatok húsa található meg a felvágottban, vagy honnan származnak a kenyérsütéshez felhasznált búzaszemek (Parsons, 2018). Landt cikkje alapján 1970-re a rádió frekvenciás nyomon követés valósággá vált, és egyre több felsőoktatási intézmény illetve kutató intézet kezdett el foglalkozni ezzel a technológiával. A számos innováció közül kiemelendő Alfred Koelle, Steven Depp, és Robert Freyman elektronikus felismerő rendszere, amit ezzel a technológiával valósítottak meg, 1975-ben. Ez a rendszer két fő elemből állt, egy passzív jelküldőből, melyet tag-nek neveztek el, és egy jelfogadó és feldolgozó rendszerből, mely olyan frekvencia tartományban

dolgozott, aminek hatására képest volt a több tíz méterre lévő jelküldőket is beazonosítani (Landt, 2005). Want cikkje részletes bemutatással szolgál a különböző elérhető RFID technológiákról. A cikk 2 fő részre osztja a beazonosító rendszereket. Léteznek aktív, és passzív rendszerek. Az aktív rendszerek lényege, hogy az információ hordozó működéséhez szükség van valamilyen áramforrásra, míg a passzív adathordozók működéséhez nem szükséges áramforrás, illetve kis méretük miatt sokkal praktikusabbak. Hatótávolság szempontjából szintén két változatot különböztet meg az értekezés. Lehet szó közeli RFID rendszerekről, vagy távoli RFID rendszerekről. A kis hatótávú rendszerek felépítése viszonylag egyszerű, így az ipar ezekre a rendszerekre fejlesztett ki először különböző megoldásokat, illetve szabványokat. A nagy hatótávú rendszer felépítésében a leglényegesebb különbség a frekvenciaérték melyen üzemelnek. Leegyszerűsítve így is magyarázható: Minél magasabb a frekvencia értéke, annál távolabbról beazonosítható az adathordozó (Want, 2006).

3.3.4.1 RFID működése

A rádió frekvenciás beazonosító rendszer működését ismerteti az 5. ábra. Az ábra alapján a rendszer egy adóval, egy vevővel, illetve egy számítógéppel rendelkezik. Az adó a benne található adatot passzív vagy aktív úton közli a vevő felé. A vevő az adott frekvencia tartományon elkezd sugározni a hullámokat, majd a vevő a hatáskörébe érve, visszaküldi a benne található információt. A vevő ezt befogadja, és tovább küldi a feldolgozó egységnek, azaz a számítógépnek. Esetenként az adó rendelkezhet antennával, vagy valamilyen egyéb kiegészítővel, mely segíti a jel kiküldését a vevőnek. Az ábrán az is megfigyelhető, hogy az információ közlése nem kivételesen egy irányba történik. Mára kifejlesztettek olyan technológiákat, melyek lehetővé teszik, hogy az adathordozókban található információk átírássra kerüljenek, azaz módosítani lehet azokat.



5. ábra: RFID rendszer működése (Internet 13)

Suresh és Chakaravarthi munkája alapján az RFID rendszereket számos helyen tudják alkalmazni, mind az iparban, mind az egészségügyben, agrikultúrában stb. (Suresh & Chakaravarthi, 2022). Ezeken belül lehet akár alapanyag, akár munkaerőre irányuló nyomon követést alkalmazni. Viszont ahogy minden technológia alkalmazása egy vállalatnál, az RFID rendszerek sikeres interpretálása is kritikus sikerességi faktorokhoz van kötve. Ting kutatása alapján megállapítható, hogy a faktorok három dimenzióba sorolhatóak. Ezek a dimenziók a technológiai dimenzió, melybe tartozik többek között a megfelelő hardver és szoftver kiválasztása, megfelelő tesztelés, legoptimálisabb információs útvonalak kijelölése. A második dimenzió a menedzsment dimenziója. A menedzsment dimenziójához sorolható a kockázatkezelés hatékonysága, illetve a projekt kivitelezésének menedzsmentje. A harmadik dimenzió a szociális dimenzió, mely a kommunikáció fontosságát taglalja (Ting et al., 2013). Ezeken a faktorokon kívül számos ilyen kritikus pontot érdemes még számításba venni ahhoz, hogy a rendszer megfelelően működjön. Chhetri és munkatársai értekezése egy egészen átfogó példát nyújt az RFID rendszerek alkalmazására. Az általuk vizsgált könyvtári rendszer több elemmel is rendelkezik, ezen elemekhez pedig részletes specifikációt prezentálnak (Chhetri et al., 2019). A nyomonkövetési rendszerek tehát megoldásként szolgálnak a vállalaton belül történő információ közlés megkönnyébbítésére. Azok az információk melyeket egy ilyen RFID rendszer, vagy akár egy vállalatirányítási rendszer közölni tud az operátorokkal, vagy a vezetőséggel, fontos jelentőséggel bírhatnak a cég működésének meghatározásában. De mi a kulcsa egy vállalat effektív működésének?

3.4 Vezetői döntéshozatal és Operatív HR

Chofreh szerint a döntések szintje egy vállalatban magába foglalja a stratégiai, operációs és taktikai döntéseket. A 6. ábra alapján, a stratégiai döntések a kezdeményezésekre és azokra a hosszú távú tervekre van befolyással mely a cég működését érinti. Ezen döntések hatásköre az egész vállalatra kiterjedő, és általában a legfelső menedzsment hozza meg. A taktikai döntések szűk hatáskörűek, és rövid távú megtérülést foglalnak magukban. Ezeket általában a közép szinten lévő menedzserek hozzák meg, hogy stratégiai kezdeményezéseket és terveket valósítsanak meg. Az operációs döntések a cég mindennapi működésében meghozott döntések, melyeket általában az alacsony szinten lévő menedzserek hoznak meg. A vállalatok sikeressége vagy bukása a döntéshozatal alapján, a stratégiai döntések meghozatalán alapszik. Mivel ez a szint a legfontosabb, a döntés hozóknak minden előnyt és hátrányt számba kell venniük egy új rendszer implementálásánál (Chofreh et al., 2020).



6. ábra: Vállalati döntések szintje (Chofreh, 2020).

Az ERP rendszerek sikeres bevezetésében tehát fontos szerepet játszik a Tudás Menedzsment (innenről TM-ként hivatkozva rá). Saide és Mahendrawathi szerint a TM elkezdett figyelmet felkelteni, ahogy megjelentek azok az eszközök melyek képesek támogatni és maximalizálni a vállalat információ átadását a menedzsment minden szintjén, hogy ezzel javítsák a vállalat működését, és a döntés-meghozatalok helyességét (Saide & Mahendrawathi, 2015). A különböző szintű döntések meghozásában tehát különböző típusú információk begyűjtésére van szükség. Ha a cégen belül, jól elkülöníthetően megtalálhatóak a vállalati döntéshozó szintek képviselői, a cég stabilabban működik. Ennek oka, hogy a halmozandó tudás az egyes szinteken nem fog diverzifikálódni, például egy műszak vezetőjének nincs szüksége egy berendezés átfogó ismeretére, ellenben szüksége van a cégben található aktív munkások számára. Ugyan így a karbantartónak nincs szüksége, az éves eladásokkal kapcsolatos információk ismeretére.

Cégünk menedzsmentje 2 szinttel rendelkezik. A vezető hozza a stratégiai és taktikai döntéseket, az egyes munkafolyamatok vezetői pedig az operációs döntéseket (Ponacco Hungária Kft., 2022). A kisvállalkozások működése során, különösen az élelmiszeriparban, fontos szerepet játszik a kríziskezelés. Amíg a vállalat nem rendelkezik egy stabil gyártási tervvel, illetve létszámiánnyal küszköd, mindennapos feladatként tekint a különböző

krízisek gyors és hatékony elhárítására. Csurí Ferenc Gábort idézve: „Mindenki a karizmatikus tulajdonostól várja a felismerést, annak belső kommunikációját és a cselekvést” (Internet 12). Ahhoz, hogy a vezető hatékonyan tudja kezelni a kríziseket, centralizálnia kell az információ áramlást a cégben. Ebben jelenthet nagy előnyt egy vállalatirányítási rendszer, mely a vezető számára szükséges információkat tud szolgáltatni.

3.4.1 Operatív HR

Az ERP és IoT (Internet of Things – a Dolgok Hálójá) rendszerek tehát gyors átláthatóságot, biztosítanak a vezetők számára, így megkönnyítve döntéseiket a vállalat irányelveivel kapcsolatban. Szűcs Ágnes szerint: „Az Operatív HR kontrolling 4 alapfeladattal foglalkozik a vállalatok működésében. Ezek a folyamatok:

- a tárgyévi humánpolitikai elképzelések, paraméterek rövidtávú kidolgozása, az éves gazdasági tervekhez kapcsolódva a humánstratégia kibontása.
- a tárgyévi eszköz- és forrásigények meghatározása a stratégiai tervek előirányzataival összhangban.
- eltéréselemzés: a teljesítmények és az operatív követelmények rendszerezett és részletes szembeállítás, a stratégiai-rézelemzés, az eltérések visszacsatolása a stratégiához.
- folyamatos adatfeldolgozás, illetve a vezetői információs rendszer működtetése”

(Ágnes Szűcs, 2017).

Némely esetben ezt a szerepet a vállalat vezetője tölti be, az ő feladata a dolgozók teljesítményének mérése, a vállalat teljesítményének mérése, illetve ezen adatok alapján a megfelelő vezetői döntés meghozása is. E három feladat elvégzése folyamatos figyelmet igényel, így az egyes feladatok elvégzésében nagyobb a hiba megejtésének lehetősége. Nemeskéri Zsolt megfogalmazása szerint: „A humán kontrolling a vezetés döntéstámogató funkcióit látja el, meghatározott vállalati folyamatokról rendszerezett, elemzett adatok gyűjtésével. Döntési alternatívákat dolgoz ki, melyeket gazdasági és humán elemzésekkel támogat. Prognosztizálja a vezetői döntések következményeit, beleértve a rejtett és hosszútávon felmerülő költségeket is” (Ágnes Szűcs, 2017; Nemeskéri, 2006).

Ezért, egy megfelelően megalkotott vállalatirányítási rendszer, mely rendelkezik elegendő kapacitással a munkafolyamatok során rögzített adatok tárolására, akár szolgálhat az Operatív Humán Erőforrások részleg helyettesítésére, vagy működésének elősegítésére.

4 Anyagok és módszerek

A feladat megoldására fő szempontjaim a gazdaságosság, mobilitás, és hatékonyság, voltak. Ehhez mérten választottam ki a felhasznált anyagokat, és módszereket.

4.1 Anyagok

Elsődlegesen a költséghatékonysághoz az eszközöknek gyorsan beszerezhetőnek, könnyen pótolhatónak, könnyen átalakíthatónak, és olcsónak kell lennie.

- Ezért zártam ki a programozható logikai vezérlőket. Ezen eszközöknek fő feladata a gépesítés, nem pedig az adatgyűjtés és feldolgozás. Ezek értelmében az eszközökhöz kapható adatgyűjtő modulok közül csak kevés felel meg egy ilyen feladat megoldására.
- Alternatív megoldásként szolgálhat, ha PLC-t kötünk egy közelítés érzékelőhöz, mely jelet közvetít, ha egy doboz elhalad előtte, így küldve adatot a központi feldolgozó egységnek. Ez a módszer egy teljesen leautomatizált rendszernél működőképes megoldás lehet, viszont a gyakorlatban, a folyamat emberi beavatkozással valósul meg, így a csalás lehetősége nagyon magas egy ilyen megoldásnál.
- Lehetőségként szolgálhat még, egy kamera beszerelése mely egy képfeldolgozó rendszerrel állapítja meg a dobozok állapotát, és így küldi az adatokat a központi feldolgozó egységbe. Ezek a rendszerek kiszűrhetik a csalás lehetőségét, viszont a program teljes újra tervezést igényelne, ha a cég úgy dönt, új dobozba szeretné csomagolni cukorkáit. Az újra tervezési folyamathoz tartozik a tesztelés is, mely időigényes folyamat egy ipari automatizálásnál.
- Ezért döntöttem az ipari mikrokontrollerek mellett. Olcsó, könnyen beszerezhető és pótolható, kellően mobilis eszközök különböző kivitelben a feladat megvalósításához és plusz feladatokkal történő bővítéshez.

Az általam választott ipari mikrokontroller, az Arduino UNO R3 (Internet 14), mely rendelkezik analóg és digitális kimenetekkel, képes adatok gyűjtésére és továbbítására, USB kimenettel rendelkezik mely megengedi, hogy adatokat közöljük az eszközzel a központi feldolgozó egységre. Ezen a kábelén keresztül az eszköz könnyen és gyorsan programozható. 69x57mm-es méretei pedig megfelelően kicsik ahhoz, hogy gyorsan átszerelhető legyen bármely helyre az üzem területén. 12V áramfelvételével átlagos fogyasztással rendelkezik. Az eszközhöz számos modul kapható, ezek mind kompatibilisek egymással. Programozó felülete egyszerűen átlátható, könnyen megtanulható, és nyílt

forráskódja miatt rengeteg kód található az interneten a különböző modulokhoz melyek ingyen felhasználhatóak és módosíthatóak.



7. ábra: Arduino UNO R3 mikrokontroller (Internet 14)

Az Arduino mikrokontrollerekhez is kapható távolság érzékelő, illetve kamera modul, viszont ezek a megoldások az előzőekben az emberi beavatkozási lehetőség miatt elvetésre kerültek. A mérendő dobozok viszont rendelkeznek szabványosított vonalkódokkal (EAN-13). Ezek meghatározására található a piacon egy eszköz, az MH-ET LIVE Scanner 3.0 vonalkód olvasó modul (Internet 15).

A 7. ábrán látható vonalkód olvasó a beépített kamera és lézeres olvasó segítségével képes 1 dimenziós és 2 dimenziós vonalkódok, illetve QR-kód beolvasására is. Mikro-USB összekötéssel, képes áramot felvenni, illetve kommunikálni a hozzá csatlakoztatott eszközzel, így küldve adatokat. Az interneten megtalálható dokumentáció segítségével személyre szabható a működése, például beolvasási mód (Folyamatos, szakaszos, tárgy érzékelésére történő olvasás), késleltetés (Két beolvasás között kötelezően eltelendő idő), segédfény (Az eszköz rendelkezik egy 4 LED-égből álló segédlámpával, így szabályozható az általa kibocsájtott fény erőssége), adatküldési lehetőségek, ...stb.



8. ábra: MH-ET LIVE Scanner 3.0 vonalkód olvasó modul (Internet 15)

A központi feldolgozó egységben található az a program, mely feldolgozza a kapott információkat, és a megfelelő módon rendszerezi azokat, illetve műveleteket végez velük. Ahhoz, hogy a dolgozók és a vezetőség átlássa az információ áramlását, és kezelni tudja azt, egy felhasználó felület (GUI – General User Interface) kialakítása szükséges. Ehhez operációs rendszerrel rendelkező eszközre van szükség.

- Az asztali számítógépek illetve notebook-ok és laptopok drága megoldásnak számítanak, egy ilyen feladat elvégzéséhez.
- A piacon már találni olyan nagyobb teljesítményű tableteket, melyek rendelkeznek operációs rendszerrel, például Lenovo Thinkpad X1, Surface Go 3, Acer Switch 5, ...stb. Előnyük, hogy könnyen mozgathatóak, és képesek egy asztali számítógép teljesítményét biztosítani. Hátrányuk azonban, hogy áruk túl magas ahhoz, hogy egy vállalat költséghatékonyan pótolni tudja ezeket a berendezéseket.
- A harmadik alternatíva, egy olyan nagy kapacitású ipari mikrokontroller, melyre feltölthető egy operációs rendszer, és így funkcionál számítógépként.

Én e mellett döntöttem mivel a mikrokontrollerek, habár teljesítményüket nézve nem olyan hatékonyak, mint az erre az asztali számítógépek, laptopok, vagy notebookok, elegendőnek bizonyulnak kisebb ipari automatizálásra tervezett programok lefuttatására. Méretük és kialakításuk könnyen kezelhetővé teszik ezeket, és alacsony áruk miatt könnyen és gazdaságosan pótolhatóak. Ilyen eszköznek számít a Raspberry Pi model 3B+ (9. ábra), mely egy nagy teljesítményű mikrokontroller, illetve rendelkezik az előbb felsorolt tulajdonságokkal. Saját operációs rendszerével alkalmas adatokat fogadni más eszközöktől a benne található 4 USB csatlakozóval, így klaviatúrát és egeret is lehet rá csatlakoztatni. HDMI bementével bármely, szintén HDMI bemenettel rendelkező monitor csatlakoztatható.



9. ábra: Raspberry Model 3B+ hozzá tartozó védőburkolattal (Saját kép)

4.1.1 Segédanyagok

Az egyes eszközök (Raspberry Pi model 3B+, Arduino UNO R3, MH-ET LIVE Scanner 3.0) összekapcsolásához segédanyagként USB, illetve Breadboard Jumper kábeleket használtam.

Az Arduino UNO R3 működéséhez egy, az eszköz mellé kapható USB kábelre van szükség. A Raspberry Pi model 3B+ működéséhez szükség van 1db tápegységre, mely árammal látja el a készüléket, 1db Egérre, illetve 1db klaviatúrára, ami a beüzemeléshez és a működéshez szükséges, illetve 1db HDMI összekötő kábelre és monitorra, amivel láthatjuk az eszköz által kiírt információkat.

A szigeteléshez és a munkavédelmi előírásoknak való megfeleléshez, egy erre a célra kialakított műanyag védőburkolatot készítettem az eszközöknek. A burkolattal az eszköz rögzíthető a munkaasztalra, és külső mechanikai hatás (pl.: ütés, rázás) hatására a benne lévő eszköz nem sérül.

A beszerzett anyagokkal és segédanyagokkal megkezdődhet a rendszer összeállítása. Fontos szempont egy rendszer beüzemelésénél egy kezdetleges működési ábra megalkotása, mely tartalmazza:

- A rendszer működésének fontosabb elemeit,
- A működési elemek sorrendjét, azaz a program lefutásának menetét,
- Az egyes pontoknál meghatározott megoldandó problémákat,
- A problémákon belül felmerülő egyéb hibalehetőségeket,
- Továbbfejlesztéshez szükséges kikapuk megtervezését a rendszerben.

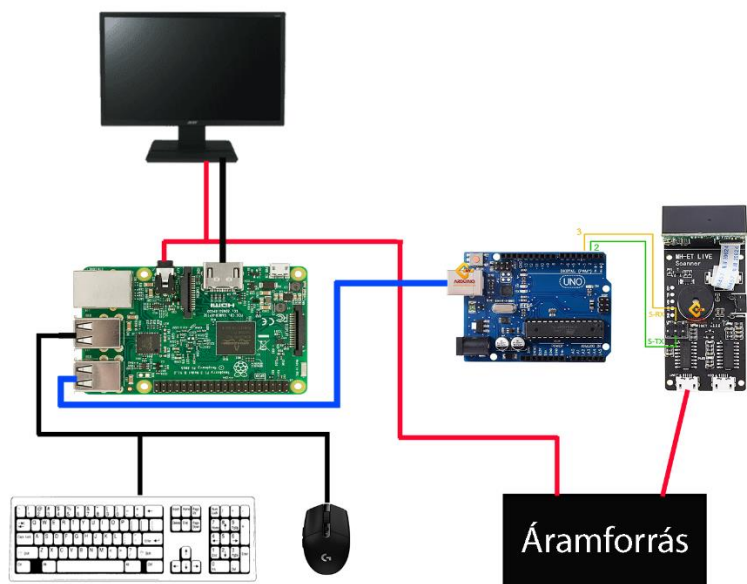
Ezen pontok megvalósításához elkészítettem egy bekötési ábrát (10. ábra), illetve a program működésének ábráját (14. ábra).

Az ábrák elkészítése után a megvalósítást két részre bontottam, a fizikai beüzemelésre, illetve a szoftver összeállítására.

5 Eredmények

5.1 Fizikai beüzemelés

A fizikai beüzemeléssel kezdtem. Az általam elkészített bekötési ábra (10. ábra) tartalmazza azokat a műszereket, melyeket az első verzióba terveztem beépíteni.



10. ábra: Bekötési séma a termelés-nyomonkövető programhoz. (Internet 16-20)

ennek első pontja a műszerek működésének megismerése, és szoftveres terhelhetőségük vizsgálata.

5.1.1 MH-ET Live 3.0 Vonalkód olvasó

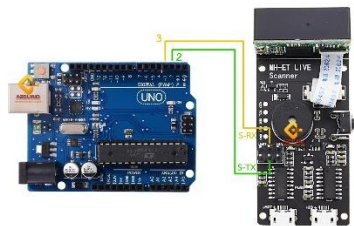
Az MH-ET Live vonalkód olvasó előnye, hogy rendelkezik egy előre telepített programmal, aminek segítségével nagyon könnyen és hatékonyan letesztelhető működése. Az interneten fellelhető PDF formátumú dokumentációjában találhatóak olyan QR kódok, amelyekkel a különböző funkciói állíthatók be az egy feladatokhoz, például beolvasás sebessége, folyamatos vagy szakaszos beolvasási mód, segédvilágítás bekapcsolása, kikapcsolása stb. Nem rendelkezik be- vagy kikapcsoló gombbal, ha bekötésre kerül, azonnal áramot kap és bekapcsol. Az előre telepített programja USB csatlakozás által tud adatot szolgáltatni a vevőnek.

Ez könnyen ellenőrizhető. Miután összekapcsoltam a számítógéppel a vonalkód olvasót, megnyitom a vezérlőpult ablakot a számítógépen, majd az eszközök almenü alatt megnézem, hogy a számítógép érzékelt-e a készüléket. Amennyiben az USB kommunikáció menüpontban látható a kamera, megnyitom a számítógép soros portját, és beolvasok egy

vonalkódot. A vonalkód számának meg kell jelennie a számítógép soros portján. Amennyiben ez megtörtént, az eszköz működése megfelelő. Abban az esetben, ha a vezérlőpult nem érzékeli a műszert viszont az bekapcsolt állapotban van, a műszer kap áramot, de a kommunikáció nem létesül. Ekkor érdemes egy driver csomagot telepíteni a számítógépre majd újraindítani a számítógépet, és újra összekapcsolni a vonalkód olvasóval. Ha a számítógép nem érzékeli a kamerát, és áramot sem kap, akkor érdemes másik USB kábellel összekapcsolni. Egy megtört vagy kontaktos kábel nem tud áramot biztosítani az eszköznek, ezért az gyakran lekapcsolódhat, vagy rá se csatlakozik a számítógépre. Ha a kábelcsere ellenére sem működik a készülék, a másik áramot biztosító csatlakozóját érdemes használni. Ha nem tud egyik USB portján sem kommunikációt létesíteni, másik megoldást kell választani.

5.1.2 Arduino UNO R3 és Raspberry Pi Model 3B+

Ehhez szükséges egy mikrokontroller, mint például Arduino UNO R3, vagy Raspberry Pi model 3B+. Ezek az eszközök rendelkeznek jumper kábel bemenetekkel. A vonalkód olvasó előnye, hogy USB csatlakozás mellett jumper kábellel is működtethető. A 11. ábrán (Internet 20) látható a vonalkód olvasó Arduino UNO R3 mikrokontroller segítségével jumper kábellel történő bekötés. A két jumper kimenet működteti a kommunikáció módját, kettő kimenet pedig áramot szolgáltat a készüléknek. Ha nem működik az USB kapcsolat, az eszköz használható mikrokontroller segítségével, viszont a beolvasás paramétereit programozás útján kell beállítani. Ha elengedhetetlen az USB kapcsolat létesítése, az eszköz cseréire szorul.



11. ábra: Arduino UNO R3 és MH-ET Live 3.0 bekötési ábrája (Internet 20)

Ahhoz, hogy a vonalkód olvasó jumper kábellel való megfelelő működését le tudjam ellenőrizni, szükséges a mikrokontrollerek ellenőrzése is. Következő ellenőrizendő eszközök tehát a Raspberry Pi model 3b+ illetve az ARDUINO UNO R3 mikrokontroller,

melyek működésének ellenőrzéséhez az eszközökhöz tartozó USB kábeleket szükséges használni. A modellekhez léteznek hivatalos, és nem hivatalos másolatok is. A nem hivatalos másolatok kevésbé költségesek, viszont felmerülhet, hogy nem megfelelően működnek vagy komplikáltabb a beüzemelésük. A Raspberry hivatalos oldalán megtalálható a készülékekhez tervezett OS (Operating System – Operációs Rendszer) ami teljesen ingyenes. Az OS feltelepítéséhez egy mikro SD kártyára van szükség, 4gb szabad tárhellyel. Amikor letöltöttem az OS-t, behelyezem az SD kártyát a számítógépbe, és elindítom a letöltött telepítőprogramot. A programon belül kiválaszthatom melyik verziót szeretném használni az OS-ből és kiválasztom az SD kártyát, mint a rendszer tárhelyét. A telepítés után az SD kártya tartalmazza az OS-t, és azt behelyezve a Raspberry készülékbe futtatható lesz.

Ahhoz, hogy lássam működés közben, a készülékhez csatlakoztatok egy monitort HDMI kábel segítségével, illetve egy egeret USB csatlakozással. Ha a monitoron látható a kezdő képernyő, az egerrel irányítható a kurzor, és megnyithatóak az alapprogramok, a rendszer működésre kész. A 12. ábrán látható ezen bekötések ábrája. Az ábra jobb oldalán található az ARDUINO mikrokontroller, és a laptop képernyőjének jobb oldalán pedig az ahhoz tartozó ARDUINO program kezdő képernyője. Ha a program, alsó fekete felületén nem látható piros szöveg, a program illetve az eszköz működésre kész. Az ábra bal oldalán található a Raspberry Pi 3B+ mikrokontroller illetve a laptop képernyőjének bal oldalán pedig a készülékre telepített operációs rendszer kezdőképernyője látható.



12. ábra: ARDUINO és Raspberry készülékek bekötése és tesztelése
(Internet 19, Internet 21, Internet 22)

A Raspberry készülék internet kapcsolat nélkül a pontos idő és a programok telepítése és beállítása nem lehetséges. A MATLAB és Excel programok telepítéséhez, és a szoftverfrissítések kezeléséhez internet kapcsolatra van szükség, hogy a készülék operációs rendszerén keresztül le tudja tölteni a megfelelő kiegészítéseket, illetve be tudja állítani a pontos időt. A készülék csatlakozni tud Wifi hálózatra is, az én készülékemen viszont

megsérült a bele épített Wifi modem. Ehhez megoldás az Ethernet kapcsolat létrehozása. Egy Ethernet kábel segítségével összekacsoltam a routert a készülékkel, így az automatikusan kialakította a kapcsolatot az internettel, és beállította a pontos időt. A 13. ábra bal oldalán található az Ethernet kábel, jobb oldalán pedig a Raspberry készülék és internetet szolgáltató router összekötése. A működő Raspberry készüléket összekötöm a monitorral, egérrel, illetve a klaviatúrával. Mikor az ábrán látható kezdőképernyő megjelenik, az OK gombra kattintva bezárom a felugró ablakot. Ez után az ábra bal oldalán látható Ethernet kimenettel rendelkező kábellel összekötöm a routert, illetve a Raspberry készüléket. Ez után az eszköz automatikusan csatlakozik az internetre, és beállítja a pontos időt. Ez után az operációs rendszerben letölthetőek a program felépítéséhez használatos programok.



13. ábra: Ethernet kábel (bal), és Ethernet internetösszeköttetés Raspberry készülékkel (jobb)
(Internet 16-19, Internet 23-24)

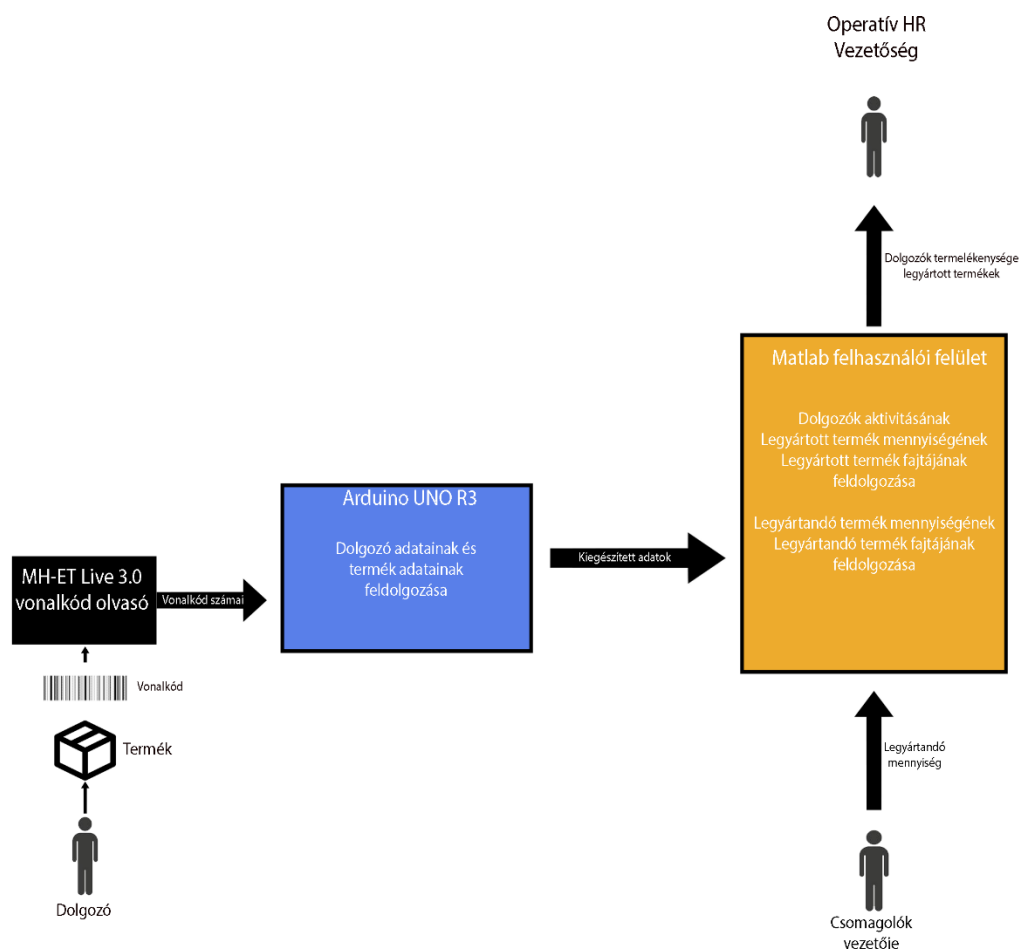
Miután leteszteltem a vonalkód olvasót és a mikrokontrollert is, összekötöttem azokat a 8. ábra alapján, hogy együttes működésüket is megvizsgáljam. Az USB kapcsolatot ugyan úgy megvizsgáltam a Raspberry számítógépre, illetve a vonalkód olvasóra is. Amikor védett körülmények között megfelelő a működés, elkezdtem vizsgálni a rendszert a gyártósor körülményeivel.

5.2 Program felépítése

A tesztek után a vonalkód olvasóhoz és a számítógéphez használatos programot kezdtem megírni. A 14. ábra mutatja a program felépítését, illetve azokat az fő adatcsoportokat melyeket az egyes részeknek fel kell dolgoznia és továbbítania kell. Az alapot a termelés nyomon követhetősége adja, illetve hogy a program megfelelő kiindulásként szolgáljon egy komplexebb vállalatirányítási rendszer megalkotásához. Szintén a projekt fő szempontjai közé tartozik, hogy a meglévő rendszert ne befolyásolja a tesztfázis, vagyis ne vesszenek el

olyan értékes adatok, amelyeket a vállalat már begyűjtött és rendszerezett. A felsorolt szempontoknak való megfelelés céljából az adatgyűjtés formáját Microsoft Excel fájlokba gyűjtöttem. Így az adatok elkülönítve kerülnek begyűjtésre a cég rendszere által gyűjtött adatoktól, és módosításukkal a vállalat központi adatbázisa nem sérül, e mellett az Excel fájlok kis tárhelyet foglalnak el az SD kártyán. Másik előnye az Excel program használatának a tesztfázis kivitelezéséhez, hogy az program saját beépített függvényeivel módosíthatóak az adatok, így kevesebb programozást igényel a projekt számára kulcsfontosságú adatok kinyerése.

A programot, mely kommunikál a vonalkód olvasóval, és adatokat közvetít a számítógépre, MATLAB IDE-n írom meg. A MATLAB IDE előnye, hogy komplex matematikai műveletek megoldására építették, így az adatfeldolgozáshoz ez a legkommerciálisabb megoldás. A felület egyszerűen kezelhető, és felhasználóbarát megoldásaival nincs szükség alacsony szintű programozás ismeretére a feladat megoldásához.



14. ábra: termelés nyomonkövethetőség rendszer felépítése és működése

A rendszer összeállítása 6 lépésből áll:

- Először a MATLAB programok felépítését kell megismerni. Minden programozási nyelv rendelkezik saját szintaktikákkal melyekre nagy figyelmet kell szentelni ahhoz, hogy a felépített program minimális hibával fusson.
- Második lépésként egy GUI megalkotása szükséges mindazokkal a gombokkal, táblázatokkal jelzésekkel, melyekkel a felhasználó értesülni tud az adatok beérkezéséről, illetve bizonyos esetekben módosítani tudja azokat, vagy meg tudja adni azokat a paramétereket melyekkel a program bizonyos funkcióit elindítja. Ezek mellett kellően egyszerűnek kell lennie a GUI felépítésének ahhoz, hogy a kezeléshez szükséges betanulási folyamat minimális időt vegyen igénybe, illetve a dolgozó ne tudjon előállítani olyan paramétereket a programban, melyekkel az adatfeldolgozás folyamata után a kinyert adatok fals eredményt adjanak a vezetőség számára.
- Harmadik lépés a kapcsolat kialakítása a vonalkód olvasóval, illetve a biztonsági programok beillesztése abban az esetben, ha a kapcsolat megszakad.
- Negyedik lépés a vonalkód olvasó által küldött adatok beolvasása a programba.
- Ötödik lépés a beolvasott adatok értelmezése és a megfelelő műveletek elvégzése a nyers adatokkal.
- Az utolsó lépés, hogy az elrendezett adatokat a program egy Excel fájlba lemenve, elérhetővé tegye a vállalat vezetőinek, illetve leegyszerűsített külalakjával könnyen értelmezhetővé és rendszerezetté tegye akár a dolgozóknak is.

5.2.1 MATLAB tanulmányozása

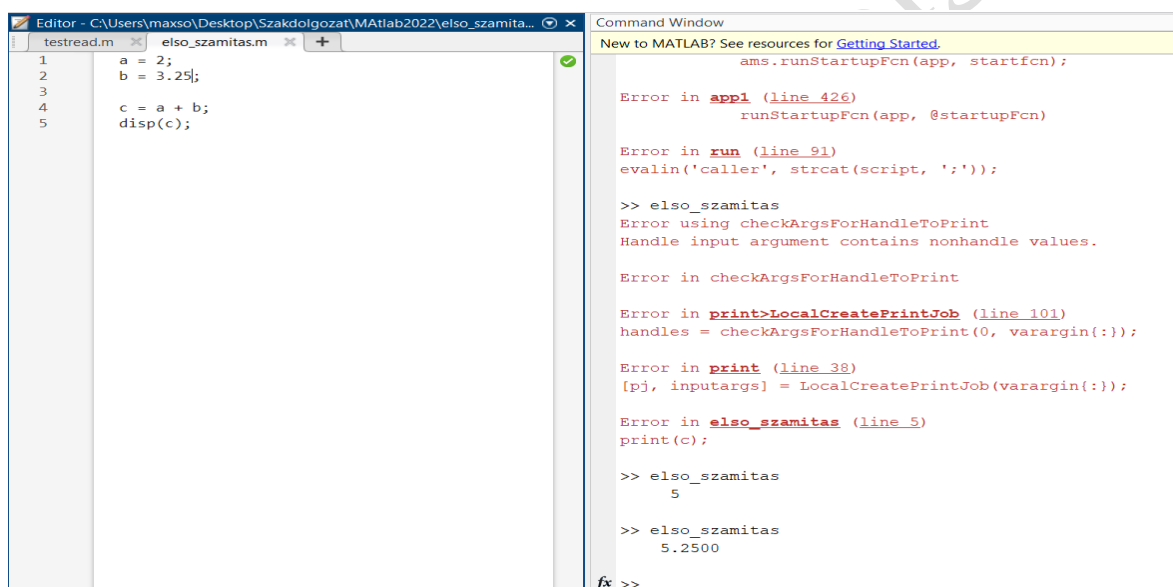
Először tehát a MATLAB által használt programozási nyelvet, és az IDE (lásd: 2.3.2) sajátosságait kezdtem el megismerni. Az IDE rendelkezik saját dokumentációval, illetve különböző internetes fórumokon is találhatóak olyan megoldások, melyek felhasználhatóak a saját projektem megtervezésére és megalkotására. Először a dokumentációt átolvasva alapszintű programokat futtattam le, hogy lássam milyen megoldásokat tudok alkalmazni a projekt során kinyert nyers adatokkal történő számítások elvégzésére. A 15. ábrán látható a legelső program, amit a felületen megírtam. A programozási nyelvekben alapvető feladat az adatok deklarálása, akár szöveget deklarál a programozó, akár számot. A MATLAB által használt nyelvezet automatikusan meghatározza az adat formáját és úgy alakítja az adattípust, hogy megjeleníthető legyen a végeredmény. Az adattípusok mellett érdemes

megemlíteni a matematikai műveletek elvégzéséhez szükséges szabályokat. Bizonyos IDE-k szabályokhoz kötik a matematikai műveletek elvégzését. Ezek a szabályok lehetnek:

- Zárójel segítségével elvégezni a műveletet
- Függvényben elvégezni a műveletet.

Ezek megbonyolíthatják a beüzemelést, viszont a MATLAB egy matematikára optimalizált fejlesztői környezet, így matematikai műveletek elvégzéséhez csak a műveletek sorrendjének írásban történő meghatározásával kell tisztában lenni ahhoz, hogy saját megoldást írjunk.

Ezek alapján a további komplexebb számítások is sokkal könnyebben elvégezhetők, és a programban található beépített függvények használatával nagyon egyszerűen leírhatóak.



The screenshot shows the MATLAB Editor with a file named 'elso_szamitas.m' containing the following code:

```
1 a = 2;  
2 b = 3.25;  
3  
4 c = a + b;  
5 disp(c);
```

The Command Window shows the execution of the script, displaying the result of the addition:

```
>> elso_szamitas  
5  
  
>> elso_szamitas  
5.2500  
fx >>
```

15. ábra: Egyszerű összeadás a MATLAB programban

Az alap számítások mellett szükségem volt a MATLAB felületen létrehozható táblázatok felépítésének és kezelésének megismerésére. Ezek a műveletek nem sokban különböznek az egyéb IDE-k függvényeitől. Listákat létrehozva és strukturálva létrehozhatóak a táblázatok, majd ezek egyes elemeit kiemelve elvégezhetőek a megadott műveletek. A táblázat elemei a végső adatfeldolgozásban fontos szerepet játszanak, így több verziót is alkottam a táblázatok jellegére. Ezekhez a verziókhoz összegyűjtöttem azokat a szempontokat melyeket a programnak meg kell jelenítenie és fontos információval bír a teljesítmény, és így a bérézés meghatározásához. Az általam választott pontok:

- Dolgozó neve,

- Egy nap alatt becsomagolt cukorka mennyisége dobozszámban,
- Az időintervallum, amin belül a dolgozó aktív volt.

A meghatározott pontok alapján meghatároztam azokat az adatforrásokat, amelyekkel ezek kiszámíthatók majd ezekre az adatokra készítettem egy táblázatot. Az első verzióban a táblázat sorai folyamatosan bővültek a következő oszlopokban:

- Dolgozó neve,
- Becsomagolt doboz száma,
- Pontos idő a becsomagolt doboz lejelentéséhez.

Az első teszt végén az első sor tartalmazta mikor fejezte be a dolgozó a legelső doboz becsomagolását, az utolsó sor pedig a legutolsó doboz becsomagolását és annak pontos idejét. A végső táblázat megtekintése után látható volt, hogy az adatok mennyisége miatt a formátum túl bonyolult egy egyszerű áttekintéshez ezért ez a táblázat további egyszerűsítésre szorult. A gyors és hatékony döntéshozatalhoz minimális végső adatmennyiségre van szükség, ami a lehető legtöbb nyers adatot tartalmazza.

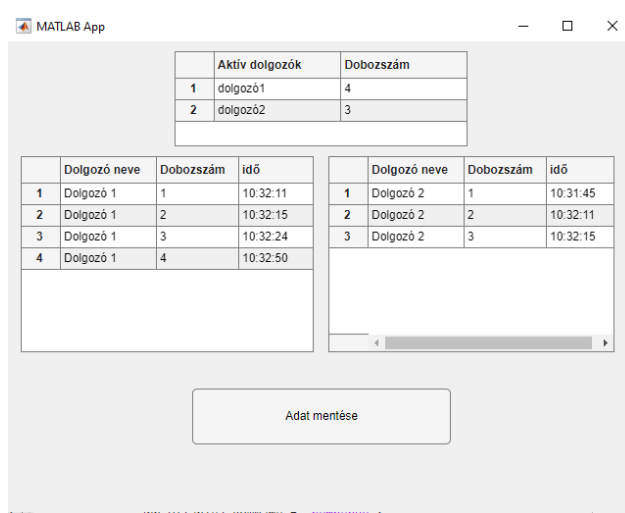
Az esetben, ha két munkás dolgozik egyazon gépen melyre egy vonalkódolvasó van kötve, egy oszlopban jelenik meg a két név felváltva vagy egymás mellett. Ez által a következő probléma a nevekhez tartozó adatok szétválasztása. A programnak fel kell ismernie mely dobozszám tartozik az első dolgozóhoz a második dolgozóhoz vagy mindkettőhöz, e mellett szét kell választania az összes becsomagolt dobozszámot az egyes dolgozókra, hogy ne a gép által becsomagolt dobozszám mennyiséget kapja meg a vezetőség, hanem a dolgozókhoz tartozó dobozszámot. Ennek értelmében létrehoztam egy második listát, mely számszerűsíti:

- Az aktív dolgozókat (1 vagy 2),
- Az összes becsomagolt dobozszámot,

Ez által nyomon-követhetővé vált hány munkás dolgozott a csomagoló gépen, illetve hány dobozt csomagoltak be egy adott időtartamon belül. Az egy dolgozóra vetített becsomagolt dobozszám meghatározására tovább módosítottam a táblázatkezelő program felépítését. Minden dolgozóhoz rendeltem egy azonosító kódot, és az azonosító kód tartalmazta az először elkészített táblázatot. Így a munkafolyamat végén a program 3 táblázatot hozott létre, mely a 16. ábrán látható, és melynek részei:

- Két táblázat tartalmazta a dolgozók adatait a saját becsomagolt dobozszámukkal, a hozzá tartozó időpontokkal, illetve a saját nevükkel.
- A harmadik táblázat összegezte az összes becsomagolt dobozszámot, illetve a dolgozók számát.

A rendszer működött, az adatok a vezetőség számára kellően egyszerűek lettek azoknak a döntéseknek a meghozatalához, mely a termeléssel kapcsolatos, illetve a dolgozó maga is nyomon tudta követni saját termelékenységét.



16. ábra: Összegyűjtött tesztadatok táblázatokba gyűjtése MATLAB szoftverben

A fejlesztés következő lépése az adatoptimalizáció volt. A program minden esetben 3 táblázatot készített, így ha egy napon csak egy dolgozó csomagolt, a program által létrehozott üres táblázat a tárhelyet foglalta, illetve abban az esetben ha a vezetőség minden adatot látni akart, 3 Excel fájlt kellett megnyitnia. Ez a megoldás jól működik egy napnyi termelés elbírálásánál, viszont a hónapok során összegyűlt Excel fájlok átláthatatlanná teszik az adatkezelést. Ezek mellett az Excel csak a pontos időt tartalmazta magában, a dátumot nem. Ezek által akár két hónap után is fölöslegessé válnak az adatok, mivel a vezetőség nem tudná meghatározni melyik táblázat tartalmazza az aznapi adatokat. Ennek megoldására beillesztettem egy pontos dátumot meghatározó függvényt, és egy másik függvénnyel az Excel fájlok nevét a pontos dátumra állítottam. Ahhoz, hogy három fájl helyett egyet kapjon a vezetőség, az Excelben használható munkafüzetekeket használtam fel.

A táblázatot kezelő program így egy fájlt alkotott melynek neve az aktuális dátum lett, és a három megközelítésből kapott adatokat a fájlban belül három munkafüzetbe helyezte. Így a

vezetőség könnyen megtalálhatja mely napon mely munkások dolgoztak, és milyen hatékonyak akár egyedül, akár csapatban.

5.2.2 Felhasználói felület létrehozása

Az adatok rendszerezése után a második lépés egy felhasználói felület létrehozása. A felhasználói felület fő célja, hogy az adatokat élőben nyomon tudják követni a dolgozók vagy a vezetőség tagjai, illetve egyéb a munkakörhöz tartozó adatokat tudjanak megadni. Ennek értelmében a felhasználói felület (GUI – General User Interface) első verziója az előbbieken megalkotott három táblázatot mutatta, melyen:

- A felső táblázat az összesített adatokat,
- Az alsó két táblázat pedig a dolgozók termelékenységét mutató adatokat írta ki.

A program így be tudta gyűjteni az adatokat, viszont ezeknek lementése Excel formátumba nem történik meg. Erre a feladatra egy gombot hoztam létre a felületen, mely megnyomásával mind a három táblázat adatsora lementhető. A létrehozott GUI így egyszerű, és ellátja fő feladatát.

A csomagolóknak viszont feladatai nem csak a csomagolásból állnak. A munkások és vezetőjük felelőssége, hogy a cukorszemeket a megfelelő design-al rendelkező dobozba csomagolják a megfelelő mennyiségben. Amennyiben ez nem ilyen módon történik meg, a becsomagolt cukorszemek vagy kibontásra kerülnek, ami hatalmas veszteséget okoz a cégnek csomagolóanyag tekintetében, vagy raktározzák az előkészített mennyiséget. A raktározás a csomagolótérben lévő szabad hely miatt nem engedi nagyobb mennyiségek elhelyezését, illetve a multinacionális cégek által átvett termékek lejáratí idõtartamához kötött 2/3 szabálya értelmében, ez hosszabb ideig nem megengedhetõ a cégnek.

A lejáratí idõvel kapcsolatos szabályok értelmében a termék átadási idõpontja nem lépheti át a termék teljes fogyaszthatósági idejének 1/3 részét, azaz az átadási idõpontban a termék teljes fogyaszthatósági idejének 2/3 része még fenn kell álljon az átadás/leszállítás idõpontjában. Ehhez készítettem a 17. ábrát, mely leegyszerûsítve mutatja be a szabályt. Ennek értelmében egy 180 napos fogyaszthatósági idõvel rendelkező termék esetében az átadási határidõ a 60. nap, mivel még 120 nap hátra van a termék lejártáig, ami 2/3 része a teljes fogyaszthatósági idejének. E mellett fontos figyelembe venni a rendelt mennyiséget mivel, ha a rendelési határidõre nem érkezik meg a termék a megrendelt mennyiség, a beszállító vállalat köteles ködbért fizetni a késedelmi idõ alapján. Ezek miatt a hatékony termelés nem csak a mennyiségtõl függ, hanem a termék jellemzõitõl is.

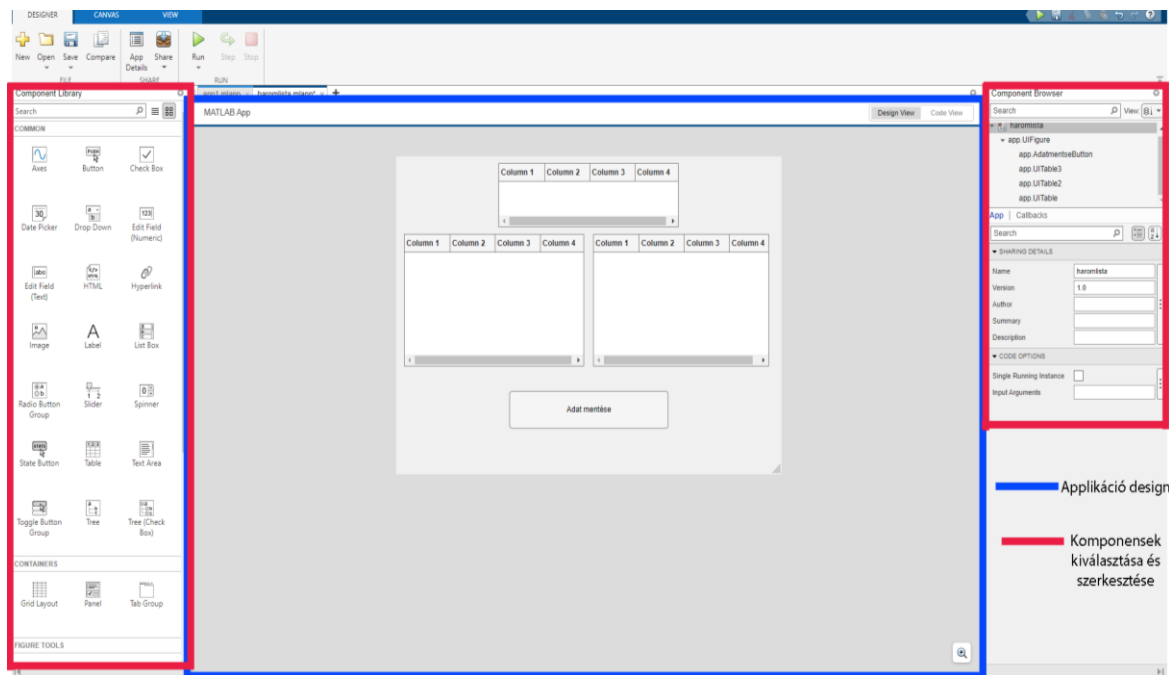


17. ábra: Termék fogyaszthatósági idejéhez kötődő átvételi szabály ábrázolása

(Internet 25-27)

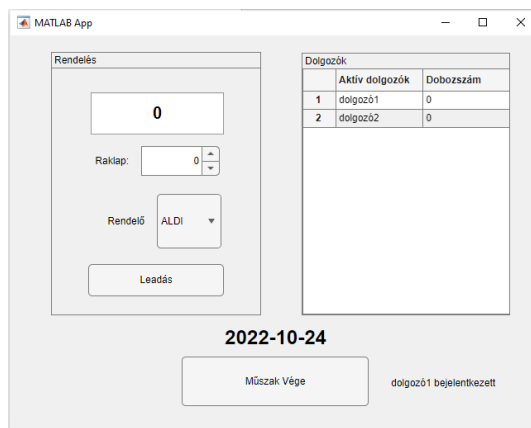
A program beépítése előtt a cég ezt a problémát papír alapon oldotta meg, ebben kritikus pont, ha a papír elveszik, a csomagolók akár a nem megfelelő berendelőhöz készült dobozba is csomagolhatják a cukorszemeket. A GUI második verziójában ezt a rendszerező felületet alkottam meg. Ehhez a MATLAB IDE-n belül található felhasználói felület készítéséhez alkalmazható komponenseket tekintettem át. Ezen lehetőségek egy része a 18. ábrán látható. A három táblázat mellé beillesztettem egy Checkbox (jelölőnégyzet) felületet, amelyben kiválaszthatja a vezető, melyik típusú dobozt szeretné becsomagoltatni a dolgozókkal. A Checkbox hátránya, hogy több elem is egyszerre kijelölhető, így minden elemet negálni szükséges a megfelelő működéshez. A mennyiségre egy klaviatúrával beírható felületet alkottam, melybe beírható a megrendelt mennyiség dobozszám formában. A felületet az összegző (felső) táblázat mellé tettem, viszont így a teljes helyet kitöltötte és további bővítésekre nincs lehetőség, a Checkbox negálását pedig egy új elem beírása esetén újra kellene alkotni, ezért a helyett más megoldást kell alkalmazni.

A MATLAB GUI alkotó felülete rendelkezik egy „DropDown” (leugró ablakok) elemmel, melybe beírhatóak a típusok, és nem engedi egynél több elem kiválasztását. A cégben különböző kiszerelésű dobozokba is csomagolják a cukorkát, így egy raklapra más mennyiségű doboz kerülhet a megrendelőtől függően.



18. ábra: MATLAB felhasználói felület készítő felület

Ezért a harmadik verzióban mely a 19. ábrán látható, a DropDown menü elemeit összekötöttem az arra a márkára vonatkozó egy raklapra kerülő dobozok számával, és egy szintén a MATLAB IDE-ben megtalálható „Spinner” elemmel így a vezetőnek már csak a raklapszámot kell megadnia, klaviatúra használata nélkül. A Spinner által begörgetett számmal és a Drop Down menüben kiválasztott megrendelővel a program kiírja, hány darab doboz került rendelésre. E mellett a beírható felület megmaradt így, ha nem egész raklapnyi dobozmennyiség került megrendelésre, az manuálisan beírható. A további bővítésre való tekintettel a három táblázatból csak az összegző táblázatot hagytam a programban megjelenítésre, illetve egy gombot helyeztem a felületre, mellyel automatikusan lementhető a termelés. Továbbá szövegdobozzal megjeleníttetem az aktuális dátumot.



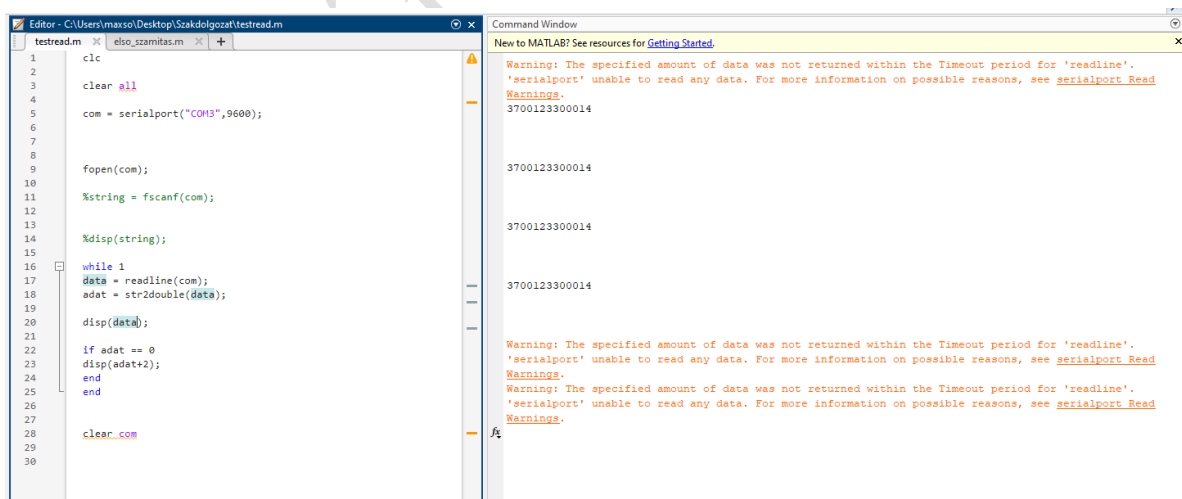
19. ábra: Termelés nyomonkövetési program végső felhasználó felülete működés előtt.

5.2.3 Kapcsolat létrehozása az eszközökkel

Miután elkészült a felhasználói felület a beírt programokkal, elkezdtem beállítani a vonalkód olvasót. Ennek a fázisnak első lépése volt, hogy a laptopra kötve a vonalkód olvasó által közölt adat a MATLAB felületén jelenjen meg.

A MATLAB rendelkezik egy USB port kommunikációs bővítménnyel. A kommunikáció létesítése 3 lépésből áll. Először meghatároztam, az elérhető portok közül mely portot foglalja le a program a kommunikáció létesítésére. Ehhez a `serialportlist('available')` kódsort használtam, amellyel a MATLAB kiírja mindazon USB portok nevét melyek nincsenek használva, vagyis nincs eszköz csatlakoztatva hozzájuk. Ez után a kiválasztott portra rákötöttem a vonalkód olvasót és deklarálva a kiválasztott portot, lefoglaltam azt a kommunikációhoz. Harmadik lépésben elneveztem a leolvasott adatot, és a `readline()` funkcióval folyamatosan leolvastam a közölt adatot a vonalkód olvasóról. A programban módosítható az olvasás sűrűsége, meg lehet határozni mikor olvasson a program, vagy milyen esetekben ne írjon adatot a központi konzolra. A folyamat lefutása a 20. ábrán látható.

Az ábra tartalmazza a `readline` függvényt, illetve a `disp()` függvényt, mely arra szolgál hogy az ábra jobb oldalán található konzolra írja ki a program a beolvasott adatot. Ez esetben az adat egy Nestlé Aqarel szénsavmentes ásványvíz vonalkódja. Az ábrában továbbá látható egy feltétel, mely szerint ha a beolvasott adat egyenlő nullával, a kiírandó adathoz a program adjon kettőt.



The screenshot shows the MATLAB environment. The Editor window displays a script named `testread.m` with the following code:

```
1 clc
2
3 clear all
4
5 com = serialport("COM3", 9600);
6
7
8
9 fopen(com);
10
11 %string = fscanf(com);
12
13
14 %disp(string);
15
16 while 1
17     data = readline(com);
18     adat = str2double(data);
19
20     disp(data);
21
22     if adat == 0
23         disp(adat+2);
24     end
25
26
27
28 clear com
29
30
```

The Command Window shows the output of the script, displaying the serial number `3700123300014` multiple times. There are also warning messages in the Command Window:

```
Warning: The specified amount of data was not returned within the Timeout period for 'readline'.
'serialport' unable to read any data. For more information on possible reasons, see serialport Read
Warnings.
3700123300014

Warning: The specified amount of data was not returned within the Timeout period for 'readline'.
'serialport' unable to read any data. For more information on possible reasons, see serialport Read
Warnings.
3700123300014

Warning: The specified amount of data was not returned within the Timeout period for 'readline'.
'serialport' unable to read any data. For more information on possible reasons, see serialport Read
Warnings.
3700123300014
```

20. ábra: MATLAB USB kommunikáció lefutása

A Raspberry Pi készülék hátránya, hogy 4 USB porttal rendelkezik és ezekből szükség van két portra, az egér és a klaviatúra használatához. A három csomagoló állomásból így maximálisan csak kettőt lehet közvetlen ráilleszteni a készülékre és további bővítések, illetve

fejlesztések esetén a port kommunikáció a különböző eszközökkel sokkal bonyolultabb lenne. Megoldás erre a vonalkód olvasó és az Arduino Uno mikrokontroller összekötése. Az Arduino és vonalkód olvasó közötti kapcsolat kialakítása komplikáltabb.

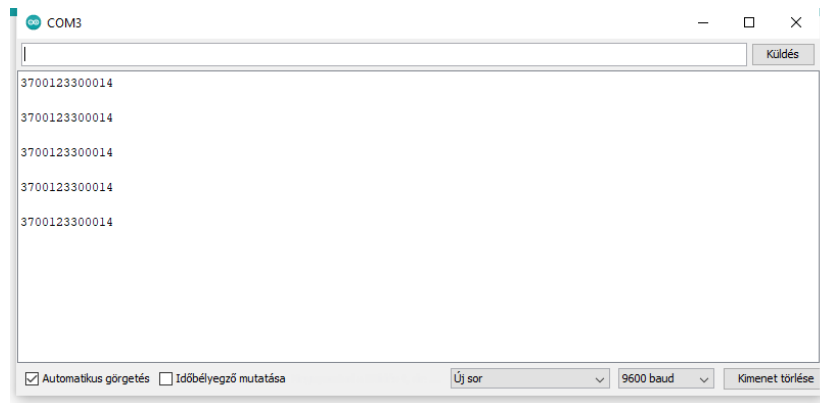
A jumper kábeleket használva összekötöttem a két eszközt, és meghatároztam mely helyeken fog folyni a kommunikáció. EAN 13 vonalkód esetén egy olyan programot alkottam mely meghatározza, hogy a vonalkód olvasó hány karaktert szeretne közölni. Ha a közölni kívánt karakterek száma 13, egy feltétel rendszerbe illeszttem az egyes karaktereket. Ha a vonalkód számai megegyeznek az általam megadott számokkal, a program kiírja az általam közölni kívánt adatot a soros portra. Mivel a jumper kábelek csak adatot vesznek át a vonalkód olvasótól, a beolvasás gyorsasága vagy fajtája nem változik az Arduino program futtatása során. Leteszteltem a programot különböző beolvasási módokkal, és beállítottam azt a módot illetve gyorsaságot, amely a legkevesebb hibalehetőséget hordozza magában. Tapasztalati úton a folyamatos olvasás 1 másodperces olvasásközi leállással volt a legideálisabb a gyors de hatékony olvasáshoz. Az olvasási módok letesztelése a gyakorlati szempontok alapján volt fontos lépés a projekt kivitelezésében.

Felmerülhet egy esemény, mely során a dolgozó előre becsomagol 7 darab dobozt, és egyszerre szeretné beolvasni azokat. A megállás nélküli folyamatos beolvasás következtében egy dobozt több alkalommal is beolvasott a készülék, így az veszteséget okozna a vállalatnak a gyakorlatban. Az MH-ET Live 3.0 rendelkezik egy olyan móddal mely során az olvasás bekapcsol abban az esetben, ha a beépített kamera közelébe tesznek valamit. Ez a mód nem minden alkalommal érzékelt a doboz közelségét, és vagy nem kapcsolt be, vagy ahogy az előző mód, többször regisztrálta ugyan ahhoz a dobozhoz tartozó vonalkódot.

5.2.4 Adatok átvitele IDE-k között

Következő lépés volt az Arduino által megjelenített információ továbbítása a MATLAB programba. Mivel az adatátvitelhez 2 IDE-t is használok, számos lehetőség van az adatok továbbítására.

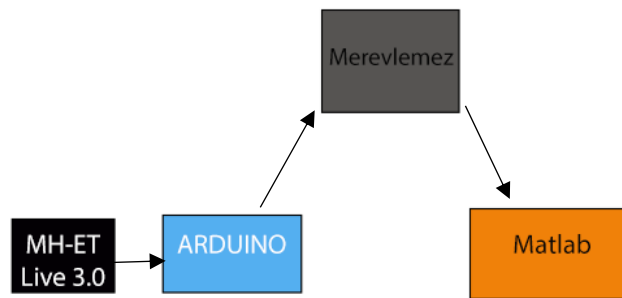
Először a MATLAB programban minden lehetségesen beolvasott vonalkódhoz egy azonosító számot jelöltem meg. Az azonosító számokat ezután kiírtam az Arduino soros portjára a 21. ábrához hasonlóan, majd a soros portra kiírt adatokat olvastam be a programba. Ez hatékony kommunikációs módszernek bizonyult, viszont egy új vonalkód beolvasása esetén mind a két programot bővíteni kell a meghatározott műveletekkel.



21. ábra: Vonalkód beolvasása Arduino mikrokontroller segítségével.

Második opció a vonalkódok kiírása az Arduino soros portjára, és azok beolvasása a MATLAB programba. Mivel az Arduino C# nyelven működik, a MATLAB nem ugyan azt az adatot olvassa be, mint amit az Arduino kiír, ezért ezekkel nem végezhetőek el műveletek. Ahhoz, hogy az adatokkal műveleteket lehessen elvégezni, ismerni kell a C# nyelv adatraktározási sajátosságát, amit Pointereknek neveznek.

A C# nyelv sajátossága, hogy az ezen a programozási nyelven írt változókat, legyen az integer, float, string, vagy char formátumú, a program a merevlemez egy szegmensébe helyezi el. Tekintsünk a merevlemezre egy könyvespolc ként, az adatokra pedig könyvekként. A könyveket a polc különböző helyeire teszi a program, a Pointerek pedig azok a cetlik, melyek megmutatják a könyvek pontos helyét, ezt a metodikát ábrázolja a 22. ábra. Mivel az Arduino a soros portra string vagy char formátumba írja ki az adatokat, a konvertálás szám formátumba kritikus pont lehet, illetve bővítés esetén a MATLAB program adatfeldolgozási részét újra kellene írni. Ezért a Pointerek használatával az Arduino által beolvasott adatformát fogom használni. Kiírva a pointert az Arduino soros portjára, és beolvasva azt a MATLAB programban, megkapom az adatom helyét a merevlemezén, így nem kell konvertálni az adatokat, és nem kell plusz műveleteket írni sem a MATLAB, sem az Arduino programjába.



22. ábra: Adatok feldolgozása Pointerek használatával

A Pointerek konvertálásához a beolvasott értékeket átalakítottam bájtokba. A beolvasás után egy függvény bájt értékbe konvertálja a vonalkód olvasó által kiírt Integer értéket, és ehhez adja meg a Pointert. A Pointert ezután kiküldi a soros portra, majd a MATLAB program beolvasva a pointer értékét, átveszi a számot, és vissza konvertálja az eredeti Integer értékre. A műveletet lehetséges Float formátumú számokra is alkalmazni.

5.2.5 Műveletek elvégzése az adatokkal

Negyedik lépés volt az adatokkal kapcsolatos műveletek elvégzése. A programnak kettő lényeges adatot kell megkülönböztetnie, amik:

- A dolgozók egyedi azonosító kódját,
- A dobozok EAN13 vonalkódját.

Az Arduino beolvasó program további bonyolítása nélkül, egy EAN13 kód generátorral minden dolgozóhoz létrehoztam egy egyedi vonalkódot. A vonalkódot beolvasva a MATLAB programba megadtam a vonalkódhoz tartozó nevet. Ez után a feldolgozandó adatok a doboz vonalkódjai és azok számszerűsítése a táblázatban, illetve a dolgozók vonalkódjai és az azokhoz tartozó nevek feltüntetése a megfelelő táblázatokban.

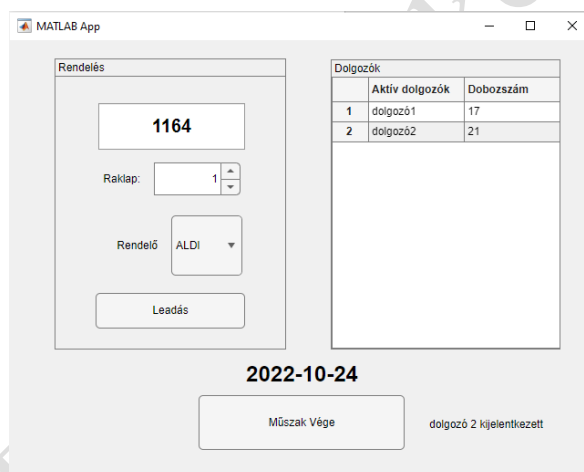
A dobozok vonalkódjaihoz a számszerűsítés mellett hozzá kell kötni mely márkából származnak ahogy az, az előre elkészített táblázatokban szerepel. Így egy doboz beolvasásának alkalmával a programnak:

- Egy számot kell küldenie,
- A számhoz tartozó megrendelő nevét kell meghatároznia.

Felépítése alapján tehát ha a megrendelő már szerepel a listában, a hozzá tartozó számhoz hozzáadja a beolvasott dobozt a program. Amennyiben nem szerepel a listában, a program egy új sort alkotva az új berendelő nevével elkezd a beolvasások számolását, e mellett a dolgozók adatait hozzáköti a megfelelő listához. Ez által a program külön tartja nyilván az

aktív dolgozókat, illetve a beolvasott dobozok számát, és csak abban az esetben köti össze azokat, ha a dolgozó bejelentkezett az állomásra.

Ez által a dobozok akkor is beolvashatóak, ha egy dolgozó sem jelentkezett be, ami hibának számít a programban. Ennek megoldása, hogy a dolgozó csak akkor olvashatja be dobozát, ha bejelentkezett az állomásra, így nem vesznek el adatok a termelés nyomon követésben. Ezzel párhuzamosan előfordulhat, hogy a dolgozó bejelentkezve marad az állomáson így az aznapi termelés az ő neve alá kerül. A programba ezért beillesztettem egy biztonsági kijelentkező függvényt, melynek szerepe, hogy a műszak után automatikusan minden dolgozót inaktív állapotba kapcsol, és lementi az aznapi termelést. Ezek mellett a rendelések leadását összekötöttem a beolvasott dobozok számával (23. ábra). Mikor leadásra kerül a rendelés, a megrendelt dobozok száma megjelenik egy panelen, és minden beolvasás alkalmával a panel száma 1 értékkel csökken. Így valós időben nyomon követhető hány doboz beolvasására van még szükség a rendelés teljesítéséhez.



23. ábra: Termelés nyomonkövetési program végső felhasználó felülete működés közben.

5.2.6 Adatok mentése

A program lefutása után a lementett adatokat megtekintettem. Az adatok pontosak voltak, így kérvényeztem a vezetőségtől egy 3 napos tesztidőszakot, ami alapján láthatom a dolgozók reakcióját az új fejlesztésre, illetve kikérhetem véleményüket a rendszer felépítésével kapcsolatban. Fontos szempont, hogy a dolgozók ne csak megértsék a rendszer működését, hanem alkalmazni is tudják azt.

5.3 Költségelemzés

A rendszer elkészülte után összevetésként érdemes megemlíteni a kiválasztott anyagokkal és segédanyagokkal keletkező összköltséget, illetve összehasonlítani ezt az előbb felsorolt alternatív megoldások költségeivel.

- Az MH-ET LIVE Scanner 3.0 vonalkódolvasó modul (Internet 28) ÁFA nélkül 15.114Ft.
- A Raspberry Pi model 3B+ (Internet 29) 24.990Ft-tól beszerezhető, a hozzá tartozó védőburkolat pedig 2.490Ft-tól kapható. A hozzá tartozó segédanyagoknál:
 - a tápegység 4.990Ft,
 - az egér 990Ft (Internet 30),
 - a klaviatúra beszerzése nem szükséges, mivel a program nem igényel gépelést,
 - HDMI bemenettel rendelkező monitorral és hozzá tartozó kábellel pedig rendelkezik a vállalat.
- Az Arduino UNO R3 nettó 6.203Ft-ért kapható, (Internet 31)
- a hozzá kötendő jumper kábelek (30db.), amikből 2db.-ra van szükség, 990Ft-ba kerülnek.
- Az ehhez szükséges mikro-USB kábel 900Ft.

Összeadva az árakat, a rendszer teljes beszerzési költsége:

$$15.114 + 24.990 + 2.490 + 4.990 + 990 + 6.203 + 990 + 900 = \underline{51.767\text{Ft.}}$$

Ezek mindegyike könnyen beszerezhető termék, illetve a rendszer legérzékenyebb pontjához, amit a vékony jumper kábeleknek tekintek, tartozik 28 másik kábel, így könnyen pótolható szakadás esetén.

Ezzel összevetve:

- egy Windows operációs rendszerrel rendelkező táblagép (Internet 32) ára 80.000Ft,
- Egy PLC (pl.: Siemens LOGO) nettó 41.959Ft-ba kerül. (Internet 33)
- Egyéb, vállalatok által megtervezett és betelepített adatgyűjtő és feldolgozó rendszerek pedig ennél is költségesebbek.

A költségelemzés 2022 október 24.-én készült, így az árak megváltozhattak.

5.4 Vélemények

A tesztek során a dolgozók könnyen alkalmazkodtak az új munkamenethez. A felhasználói felület kellően egyszerűnek bizonyult a gyors megértéshez, a vonalkódok leolvasásának figyelembevétele pedig csak minimális adatvesztést okozott.

A programnak köszönhetően a vezetőség adatokat tud elemezni a dolgozók teljesítményének elbírálására, illetve új módszerekkel tudja értékelni termelékenységüket. A rendszer működése közben kisebb adatkorrekcióra volt szükség, egyes esetben a program lefutása lassabb volt, mint a beolvasott dobozmennyiség, így pozitív irányba kellett korrigálni a táblázatokat. Ez azt indikálja, hogy a dolgozók nehezen tudnak több dobozt beolvasni a valós becsomagolt darabszámnál, ami a cél teljesítését jelenti. Ellenben a további adatoptimalizálással kellően gyors lehet a program ahhoz, hogy ne legyen szükség utókorrekcióra.

5.5 További fejlesztések a következő verzióhoz

5.5.1 Emberi tényező

Egyes dolgozók csomagolási stratégiája hasonlított a 2.3 pontban leírtakhoz. A beolvasás késleltetése megakadályozta a nagyobb hibákat beolvasás közben, viszont a dolgozók maguk olvashatják be leragasztott dobozaikat közvetlen érintkezve a vonalkód olvasóval. A jelenlegi verzió fejlesztésére egyszerű megoldás lehet egy futószalag beépítése terelőkkel. A futószalag és terelők segítségével a dolgozók nem olvashatnak be egy dobozt több alkalommal, így nem tudják módosítani saját termelésük. A futószalag sebességének beállításával a dobozok nem haladnak túl gyorsan a beolvasáshoz, illetve a feltorlódás ellen a futószalag kimeneti vége egy ládában érne véget egy csúszdával, így a dobozok sem sérülnek.

5.5.2 Internet of Things

Második fejlesztési pont az adatok átvitele egy IoT rendszerbe, hogy a vezetőség tagjainak ne legyen szükséges fizikálisan jelen lenni a termelésben a valós idejű adatok megtekintéséhez. A MATLAB rendelkezik IoT függvénycsomaggal, ez által pár függvényt alkalmazva illetve a Raspberry készüléket egy Ethernet kábellel csatlakoztatva az internethez, az Excel táblázatokat folyamatosan fel tudná tölteni a hálózatra. Ezzel párhuzamban a vezetőség is tudna adatokat közölni a dolgozókkal. Lemásolva a rendelés megadása felületet a MATLAB programból, és létrehozva egy listát az IoT rendszerben, a vezetőségnek nem kell felhívnia a műszakvezetőt ahhoz, hogy megadja melyik megrendelőtől hány raklapnyi dobozt szükséges becsomagolni. Egy megrendeléseket feldolgozó programot elkészítve, a vezetőség bármely számítógépéről leadható lenne a rendelés, amit az IoT rendszer dolgozna fel egy listával. A lista elemei ezután visszakerülnek a termelés-nyomonkövetési programba, így a dolgozók mindig tudhatják mi a következő feladat. Ez csökkenti az állásidőket.

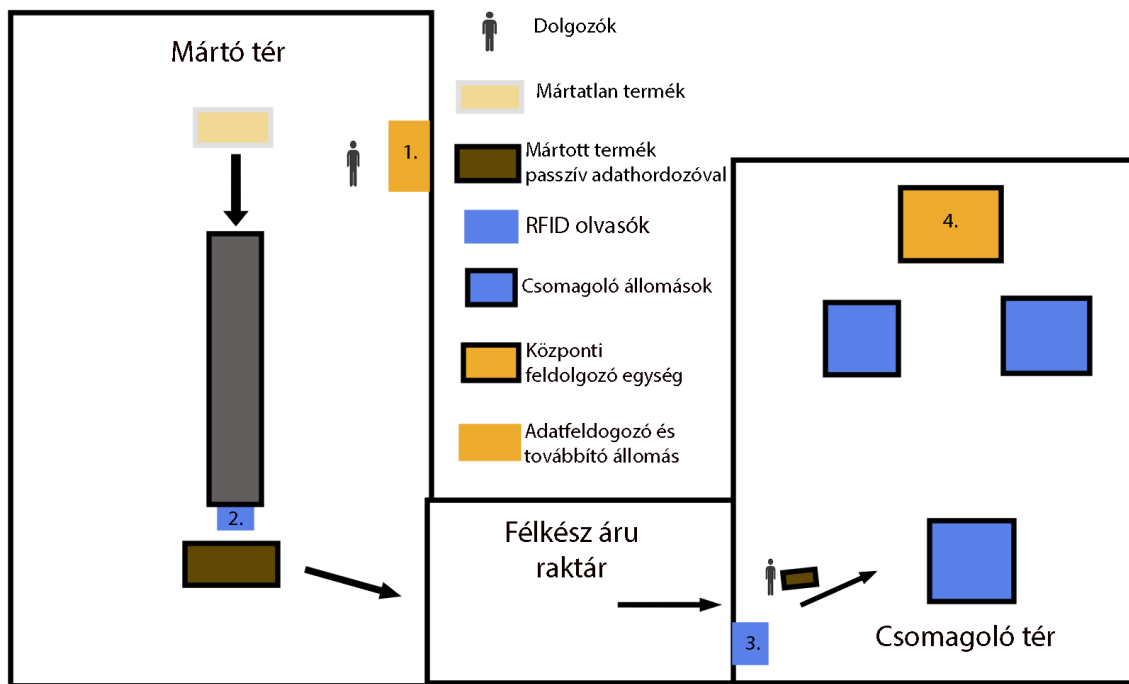
5.5.3 RFID

A rádiófrekvenciás nyomonkövetés napjainkban egyre népszerűbb megoldás az ipar különböző területein. Az egyszerű kártyás beléptető-kapun át egészen egy termék helyének centiméterre pontos meghatározásán keresztül számos esetben alkalmazzák ezt a technológiát. A nyomonkövetési program lehetséges fejlesztései közé tartozik az RFID technológia implementálása is. A vállalat 3 fajta Dianás cukorkát csomagol be. Azt, hogy mely cukorka megy a dobozba a csomagolás előtt határozzák meg közvetlen. Ennek hátránya, hogy mind a háromfajta cukorból megfelelő mennyiségnek kell készen állnia a raktárban, a csomagolás teljesítéséhez. Mivel a cukorkákat ládába szórják, egy egyszerű automatizált rendszerrel pontosíthatóak a mennyiségek.

Ha minden ládára helyeznek egy passzív adathordozót, mely egy azonosító számot tartalmaz, illetve két vevőt melyek közül egy található a csokoládé mártó állomáson a másik pedig a csomagolóterben. Ez által a vezetőség láthatja mekkora mennyiségű cukorkával rendelkeznek az egyes fajtákból, illetve azt is láthatják melyik dolgozó mártotta a cukorkákat, így nagyobb mennyiségű selejt keletkezése során gyorsan és hatékonyan tudnak eljárni.

A rendszer így nem csak a termékre összpontosít, hanem a vezetőségnek is információval tud szolgálni a dolgozók munkateljesítményéről. Ehhez az Arduino RFID modulja tökéletes megoldás költséghatékonyasága és pótolhatósága miatt. A programokat bővítve nincs szükség más módosításokra a működéshez. A beolvasás, pointerek használata, illetve táblázatok kezelése a szükséges háttértudás a bővítés elvégzésére. A 24. ábrán látható az elméleti alapja a fejlesztés megalkotásának.

- Első pontban a Mártó térben dolgozó munkások meghatározzák melyik fajtájú cukrot mártják be csokoládéval, illetve beolvassák saját kártyájukat, hogy a program fel tudja jegyezni melyik munkás dolgozott az adott mennyiséggel. Ennek a selejtkelkezés megelőzése szempontjából fontos szerepe lehet.
- Második pontban a mártóállomáson áthaladó cukrok a passzív adathordozóval rendelkező ládába esnek, mely adatokat beolvassa a gépbe üzemelt olvasó. Ezek az adatok bekerülnek a raktározási rendszerbe.
- Harmadik pontban a dolgozók a raktárból kihozva a terméket, beolvassák azt a második olvasón, így az kikerül a rendszerből.
- Az előző három pont adatforgalmát a negyedik pontban látható központi feldolgozó egység raktározza.



24. ábra: RFID nyomonkövetési fejlesztés folyamatábrája

6. Összefoglalás

Napjainkra az élelmiszeripar egyes ágazataiban folyamatos fejlődés látszik az Ipar 4.0 által létrejött új technológiák implementálásában. A megfelelő mennyiségű tőkével rendelkező nagyvállalatok sikeresen alkalmazzák az újításokat, ezzel egyre nagyobb szerepet szerezve a világgazdasági verseny élelmiszeripari szegmensében.

Míg a nagyvállalatok folyamatosan fejlődnek, a kisvállalkozások egyre nagyobb lemaradással küzdenek termelésük optimalizálásában. Ilyen vállalatok közé tartozik a Ponacco Hungária Kft. is, melyet családom alapított. Az új generáció képviselőjeként egy olyan költséghatékony termelés nyomkövetési rendszert próbáltam kifejleszteni, mely első verziójában a termelés egyik fázisában játszik szerepet, és a dolgozók termelékenységét jegyzi fel. A mérendő fázisként a cukorszemek becsomagolását választottam, melynek nagy hányada emberi munkaerővel történik.

Irodalmi áttekintésemben tanulmányoztam a cég gyártási folyamatát, az iparban is alkalmazható egyszerű technológiai újításokat, illetve a termelés nyomkövetésének Emberi Erőforrásokkal kapcsolatos vonatkozásait.

A rendszert a költséghatékonyság megtartása miatt mikrokontrollerek segítségével helyeztem üzembe. E mellett a pontos mérések és az emberi tényező minimalizálása számított a rendszer két kritikus pontjának. Ezeket a pontokat a pontos idő figyelembevételével, illetve a termékek vonalkódjával kötöttem össze.

Beüzemelés után a rendszer hatékonyan tudta ellátni feladatát, e mellett kellően egyszerűnek bizonyult a gyors betanuláshoz. A mikrokontrollerek gyorsan pótolható alanyagoknak számítanak a piacon, illetve költséghatékonyságuk miatt akár tartalék alkatrészeket is beszerezhetnek a kisvállalatok tőkéjük nagyobb mennyiségű csökkenése nélkül.

Az üzembe helyezés után fellépő kisebb hibák megoldására több fejlesztési alternatívát is feltártam elméleti szinten.

Véleményem szerint az ehhez hasonló alacsony költségvetésű rendszerek kialakításával a kisvállalatok is versenyképesek tudnak maradni mind a hazai piacon, mind az iparban alkalmazott új technológiák terén.

7. Hivatkozások

- Ágnes Szűcs. (2017). *HR kontrolling szerepe a tudatos emberi erőforrás menedzsmentben*.
- Agostino, R. (2004). *Business Intelligence: Solving the ERP Overload*.
www.CFOProject.com
- Augarten, S. (1983). The Most Widely Used Computer on a Chip The TMS 1000. In *State of the Art : A Photographic History of the Integrated Circuit* (p. 38). <http://smithsonianchips.si.edu/augarten/p38.htm>
- Chhetri, M., Sinha Thakur, R., Chhetri Assistant Librarian, M., & Ranjan Sinha Thakur, U. (2019). *Implementation of RFID Technology in Libraries: A Case Study in UPES Library Implementation of RFID system in Libraries: A case study in UPES library*. <https://digitalcommons.unl.edu/libphilprac>
- Chofreh, A. G., Goni, F. A., Klemeš, J. J., Malik, M. N., & Khan, H. H. (2020). Development of guidelines for the implementation of sustainable enterprise resource planning systems. *Journal of Cleaner Production*, 244. <https://doi.org/10.1016/j.jclepro.2019.118655>
- Chou, D. C., Bindu Tripuramallu, H., & Chou, A. Y. (2005). BI and ERP integration. *Information Management & Computer Security*, 13(5), 340–349. <https://doi.org/10.1108/09685220510627241>
- Davenport, T. H. (n.d.). *Putting the Enterprise into the Enterprise System*.
- Dehner, B., & Wheeler, T. (2017, November 1). *Micro-controllers versus PLCs: Which one belongs in your plant?* Control Engineering.
- Gridling, G., & Weiss, B. (2007). *Introduction to Microcontrollers*.
- Hughes, J. M. (2016). *Arduino: A Technical Reference*.
<https://www.pdfdrive.com/arduino-a-technical-reference-a-handbook-for-technicians-engineers-and-makers-e187689503.html>
- Landt, J. (2005). *The history of RFID*.
https://web.archive.org/web/20170809131526id_/http://www.sepaco-tech.com/modules/Manager/Articles/the%20history%20of%20rfid.pdf
- Matende, S., & Ogao, P. (2013). Enterprise Resource Planning (ERP) System Implementation: A Case for User Participation. *Procedia Technology*, 9, 518–526. <https://doi.org/10.1016/j.protcy.2013.12.058>
- Negash, S. (2004). Business Intelligence. *Communications of the Association for Information Systems*, 13. <https://doi.org/10.17705/1CAIS.01315>
- Nemeskéri, Z. (2006). *Humán kontrolling (Dr. András Klára PhD értekezése alapján)*.
- Nofal, M. I., & Yusof, Z. M. (2013). Integration of Business Intelligence and Enterprise Resource Planning within Organizations. *Procedia Technology*, 11, 658–665. <https://doi.org/10.1016/j.protcy.2013.12.242>

- Parsons, D. (2018, January 31). *Nyomon követés az élelmiszergyártásban*. Foodtech.Hu. <http://www.foodtech.hu/elelmiszer-gyartas/nyomon-kovetes-az-elelmiszergyartasban-20180126>
- Patrick Fromaget. (2022). *The Epic Story of the Raspberry Pi*. <https://raspberrytips.com/raspberry-pi-history/>
- Ponacco Hungária Kft. (2022). *Céges működés*.
- Saide, & Mahendrawathi, E. R. (2015). Knowledge Management Support for Enterprise Resource Planning Implementation. *Procedia Computer Science*, 72, 613–621. <https://doi.org/10.1016/j.procs.2015.12.170>
- Suresh, S., & Chakaravarthi, G. (2022). RFID technology and its diverse applications: A brief exposition with a proposed Machine Learning approach. *Measurement*, 195, 111197. <https://doi.org/10.1016/j.measurement.2022.111197>
- Tavana, M., Hajipour, V., & Oveisi, S. (2020a). IoT-based enterprise resource planning: Challenges, open issues, applications, architecture, and future research directions. In *Internet of Things (Netherlands)* (Vol. 11). Elsevier B.V. <https://doi.org/10.1016/j.iot.2020.100262>
- Tavana, M., Hajipour, V., & Oveisi, S. (2020b). IoT-based enterprise resource planning: Challenges, open issues, applications, architecture, and future research directions. In *Internet of Things (Netherlands)* (Vol. 11). Elsevier B.V. <https://doi.org/10.1016/j.iot.2020.100262>
- Ting, S. L., Tsang, A. H. C., & Tse, Y. K. (2013). A framework for the implementation of RFID systems. *International Journal of Engineering Business Management*, 5(1), 1–16. <https://doi.org/10.5772/56511>
- Want, R. (2006). An introduction to RFID technology. In *IEEE Pervasive Computing* (Vol. 5, Issue 1, pp. 25–33). <https://doi.org/10.1109/MPRV.2006.2>

7.1 Internetes hivatkozások

- Internet 1: <https://hvg.hu/magyarmarka/20050325diana>
- Internet 2: <https://www.foodglanz.eu/>
- Internet 3: <https://hu.wikipedia.org/wiki/Vonalk%C3%B3d>
- Internet 4: <https://ident.hu/vonalkod-alapok>
- Internet 5: <https://www.britannica.com/technology/automation/Computer-process-control>
- Internet 6: <https://www.hubspire.com/what-is-a-cpu-and-what-is-its-function/>
- Internet 7: <https://www.eletimes.com/microprocessor-vs-microcontroller-what-is-the-difference>
- Internet 8: <https://www.arduino.cc/en/Guide/Introduction>

Internet 9: <https://www.redhat.com/en/topics/middleware/what-is-ide>

Internet 10: <https://create.arduino.cc/projecthub>

Internet 11: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>

Internet 12: <https://www.hrportal.hu/hrblog/interim-management-amp-consulting/hogy-ne-krizismanagerrol-szoljon-a-megbizas-20200107.html>

Internet 13: <https://www.hellermannntyton.com/hu/kompetenciak/rfid-koevetes#>

Internet 14: https://techfun.sk/hu/produkt/arduino-uno-r3-original/?lang=hu¤cy=HUF&gclid=CjwKCAjw79iaBhAJEiwAPYw_oCB731ZVo1BLe2usXmZBB8WBqkldH2elDkAdwQqwooqNPHm0cUYq_oGxoCKeAQAvD_BwE

Internet 15: <https://www.tinytronics.nl/shop/en/sensors/optical/cameras-and-scanners/mh-et-live-barcode-scanner-v3.0>

Internet 16: <https://p1.akcdn.net/full/442222999.lg-27bk550y-b-w.jpg>

Internet 17: <https://astorekw.com/web/image/product.template/15440/image>

Internet 18: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRSdsGkSVZh4XUTZTtyeOuwggpI6vuW6eZdFA&usqp=CAU>

Internet 19: <https://image.coolblue.be/600x315/products/492827>

Internet 20: <https://www.cybertice.com/product/3825/%E0%B9%82%E0%B8%A1%E0%B8%94%E0%B8%B9%E0%B8%A5%E0%B8%AD%E0%B9%88%E0%B8%B2%E0%B8%99-%E0%B8%9A%E0%B8%B2%E0%B8%A3%E0%B9%8C%E0%B9%82%E0%B8%84%E0%B9%89%E0%B8%94-%E0%B8%84%E0%B8%B4%E0%B8%A7%E0%B8%AD%E0%B8%B2%E0%B8%A3%E0%B9%8C%E0%B9%82%E0%B8%84%E0%B9%89%E0%B8%94mh-et-live-scanner-v3-barcode-reading-board-qr-code-scanner-reader-mod>

Internet 21: <https://cdn.alza.cz/Foto/f5/NA/NA578m.jpg>

Internet 22: <https://robotechshop.com/shop/arduino/arduino-board/arduino-uno-r3-china/>

Internet 23: https://m.media-amazon.com/images/I/71TBFus4toS.AC_SY355.jpg

Internet 24: <https://www.easy-shop.hu/media/catalog/product/cache/f9624d942b096256dc066cb72376286b/4/1/41zf6tj-xml.jpg>

Internet 25: <https://www.colorironline.com/images/imgcolor/desenho-de-fabrica-20-para-colorir.jpg>

Internet 26: <https://thumbs.dreamstime.com/b/types-warehouse-icons-set-outline-illustration-types-warehouse-icons-web-types-warehouse-icons-set-outline-style-124913874.jpg>

Internet 27: https://www.nicepng.com/png/detail/393-3938414_cargo-truck-facing-right-vector-truck.png

Internet 28: https://www.hestore.hu/prod_10041540.html

Internet 29: <https://malnapc.hu/>

Internet 30: <https://teleonline.hu/vezetekes-optikai-eger-ergonomikus-kivitel->

[pd971?gclid=Cj0KCQjwntCVBhDdARIsAMEwACnAC_92FLirDfs1KVZ0hLzSxY4zle0jkSJhF-T-L_Z13O_dmxeJ_VsaAkqeEALw_wcB](https://teleonline.hu/vezetekes-optikai-eger-ergonomikus-kivitel-pd971?gclid=Cj0KCQjwntCVBhDdARIsAMEwACnAC_92FLirDfs1KVZ0hLzSxY4zle0jkSJhF-T-L_Z13O_dmxeJ_VsaAkqeEALw_wcB)

Internet 31:

https://www.hestore.hu/prod_10035528.html?gross_price_view=1&source=gads&gclid=Cj0KCQjwwfiaBhC7ARIsAGvcPe5eMuTWvnRNLBab73Q7qpwPdaXNXk3rYYaTjfKtuGbx4NST8VWrRusaAqbwEALw_wcB

Internet 32: https://www.jebeto.hu/toshiba-portege-z20t-c-05-11?utm_source=google_shopping&utm_medium=cpp&utm_campaign=direct_link&gclid=Cj0KCQjwntCVBhDdARIsAMEwACkOGmVCyD5CFskLXil1D7KfVTKRfefYOMRRmrFdfQyd7e4c98_DjbsaAq1XEALw_wcB

Internet 33: https://aonline.hu/Siemens-LOGO-Basic-kijelzovel-alapegyseg-12/24RCE?gclid=Cj0KCQjwwfiaBhC7ARIsAGvcPe63fdu2or9T2Ec72uVldlbEvNr_xeX85aHouNCToPBWY34LFtkWeB4aAqFfEALw_wcB

KONZULTÁCIÓS NYILATKOZAT

A ALBERT ARON (név) (hallgató Neptun azonosítója: GFYCZW)
konzulenseként nyilatkozom arról, hogy a
~~záródolgozatot/szakdolgozatot/diplomadolgozatot/portfóliót~~¹ áttekintettem, a hallgatót az
irodalmi források korrekt kezelésének követelményeiről, jogi és etikai szabályairól
tájékoztattam.

A ~~záródolgozatot/szakdolgozatot/diplomadolgozatot/portfóliót~~¹ a záróvizsgán történő védésre
javaslom / nem javaslom².

A dolgozat állam- vagy szolgálati titkot tartalmaz: igen nem^{*3}

Kelt: 2022. év 10. hó 26. nap


Belső konzulens

¹ A megfelelő dolgozattípus meghagyása mellett a többi típus törlendő.

² A megfelelő aláhúzendő.

³ A megfelelő aláhúzendő.

NYILATKOZAT

a szakdolgozat nyilvános hozzáféréséről és eredetiségéről

A hallgató neve: Albert Áron
A Hallgató Neptun kódja: GFYCZW
A dolgozat címe: Termelés nyomonkövethetőség a Dianás cukorka gyártásban
mikrokontrollerek segítségével
A megjelenés éve: 2022
A konzulens tanszék neve: Élelmiszeripari Méréstechnika és Automatizálás Tanszék

Kijelentem, hogy az általam benyújtott záródolgozat/szakdolgozat/diplomadolgozat/portfólió¹ egyéni, eredeti jellegű, saját szellemi alkotásom. Azon részeket, melyeket más szerzők munkájából vettem át, egyértelműen megjelöltem, s az irodalomjegyzékben szerepeltettem.

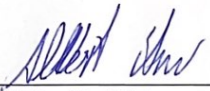
Ha a fenti nyilatkozattal valótlan állítottam, tudomásul veszem, hogy a Záróvizsga-bizottság a záróvizsgából kizár és a záróvizsgát csak új dolgozat készítése után tehetek.

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 szellemi tulajdonkezelési szabályzatában megfogalmazottak érvényesek.

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

Kelt: 2022 év 10 hó 26 nap


Hallgató aláírása

¹ A megfelelő dolgozattípus meghagyása mellett a többi típus törölendő.