



ÓBUDAI EGYETEM  
ÓBUDA UNIVERSITY

KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI KAR  
ELEKTRONIKAI ÉS KOMMUNIKÁCIÓS RENDSZEREK INTÉZET  
MŰSZERTECHNIKAI ÉS AUTOMATIZÁLÁSI TANSZÉK



# SZAKDOLGOZAT



OE-KVK  
2023.

**Horváth Dávid**  
T006540/F112904/K



ÓBUDAI EGYETEM  
ÓBUDA UNIVERSITY

Óbudai Egyetem  
Kandó Kálmán Villamosmérnöki Kar  
Műszertechnikai és Automatizálási Intézet

### SZAKDOLGOZAT FELADATLAP

Hallgató neve: Horváth Dávid

Szakdolgozat száma: SZD2303091208111047

Törzskönyvi száma: T006540/F112904/K

Neptun kódja:

Szak: villamosmérnöki (magyar nyelven)

Specializáció: Műszer-automatika specializáció

A dolgozat címe: Minőségellenőrzés ipari PC és PLC felhasználásával

A dolgozat címe angolul: Quality control using industrial PC and PLC

A feladat részletezése: A gyártás folyamán felmerülő minőségellenőrzési feladat megvalósítása. Ipari PC és PLC felhasználásával képfeldolgozás segítségével. Az eredmény HMI-n vagy monitoron való megjelenítése.

Intézményi konzulens neve: Varga Árpád

A kiadott téma elévülési határideje: 2026. 03. 09.

Beadási határidő: 2023. 12. 15.

A szakdolgozat: Nem titkos.

Kiadva: Budapest, 2023. 10. 24.



Intézetigazgató

A dolgozatot beadásra alkalmasnak találom:

*Varga Árpád*  
belső konzulens

*Béres Zoltán*  
külső konzulens



## HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja, a titkosításra vonatkozó esetleges megkötések mellett.

Budapest, 2023. 12. 12.



*Horváth Dávid*

.....  
hallgató aláírása

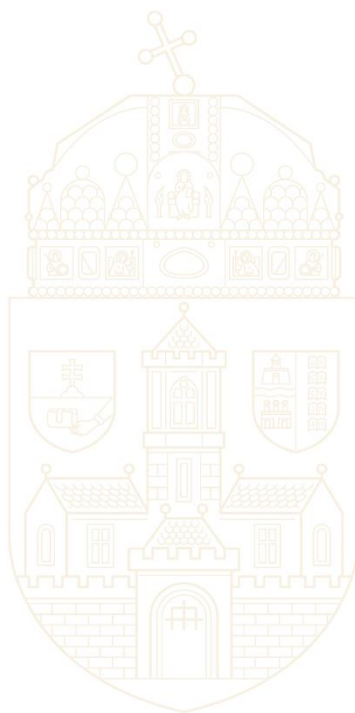


## Tartalom

|        |                                                                  |    |
|--------|------------------------------------------------------------------|----|
| 1.     | BEVEZETÉS .....                                                  | 4  |
| 2.     | A MINŐSÉG FOGALMA, CSOPORTOSÍTÁSA.....                           | 6  |
| 2.1.   | A MINŐSÉG FOGALMA .....                                          | 6  |
| 2.2.   | MINŐSÉGELLENŐRZÉS KÉT NAGY CSOPORTJA .....                       | 6  |
| 2.2.1. | Szubjektív minőségellenőrzés .....                               | 6  |
| 2.2.2. | Objektív minőségellenőrzés .....                                 | 7  |
| 2.3.   | MINŐSÉGELLENŐRZÉSI DÖNTÉSEK.....                                 | 8  |
| 3.     | KÉPFELDOLGOZÁS AZ IPARBAN .....                                  | 9  |
| 3.1.   | ESZKÖZÖK.....                                                    | 10 |
| 3.1.1. | Ipari okoskamera.....                                            | 10 |
| 3.1.2. | Ipari PC .....                                                   | 13 |
| 3.2.   | MEGVILÁGÍTÁSI MÓDSZEREK.....                                     | 14 |
| 4.     | AZ IRÁNYÍTÁSTECHNIKAI PIRAMIS.....                               | 16 |
| 5.     | SIEMENS S7-1500 TERMÉKCSALÁD.....                                | 17 |
| 5.1.   | S7-1500 CPU .....                                                | 17 |
| 5.2.   | S7-1500 1518 MFP PLC.....                                        | 18 |
| 5.3.   | TM-MFP .....                                                     | 18 |
| 5.4.   | SIMATIC HMI.....                                                 | 20 |
| 5.5.   | WEB SZERVER .....                                                | 21 |
| 5.6.   | TIA PORTAL .....                                                 | 22 |
| 6.     | IPARI PC ÉS A TM-MFP GAZDASÁGI ÉS TECHNIKA ÖSSZEHASONLÍTÁSA..... | 24 |
| 7.     | MEGVALÓSÍTÁS .....                                               | 26 |
| 7.1.   | TM-MFP FELADATA .....                                            | 26 |
| 7.1.1. | Python .....                                                     | 26 |
| 7.1.2. | A modulok importálása. ....                                      | 27 |
| 7.1.3. | A globális változó deklarálása.....                              | 27 |
| 7.1.4. | A „find_circles” függvény.....                                   | 27 |
| 7.1.5. | A „measure_circle_diameters” függvény.....                       | 28 |
| 7.1.6. | A „write_measurement_to_file” függvény.....                      | 28 |
| 7.1.7. | A „main” függvény .....                                          | 28 |
| 7.2.   | BLANCPAIN BUS FELADATA .....                                     | 31 |
| 7.2.1. | Az import utasítások: .....                                      | 32 |



|        |                                                   |    |
|--------|---------------------------------------------------|----|
| 7.2.2. | A „clean_and_convert_to_integers” függvény: ..... | 32 |
| 7.2.3. | A „main” függvény: .....                          | 33 |
| 7.2.4. | A „name == "main"” függvény: .....                | 34 |
| 7.3.   | A PLC FELADATA .....                              | 35 |
| 7.4.   | HMI FELADATA.....                                 | 36 |
| 8.     | ÖSSZEGZÉS .....                                   | 37 |
| 9.     | SUMMARY .....                                     | 38 |
| 10.    | RÖVIDÍTÉSEK .....                                 | 39 |
| 11.    | ÁBRAJEGYZÉK .....                                 | 40 |
| 12.    | IRODALOMJEGYZÉK.....                              | 41 |





## 1. BEVEZETÉS

Az ipari folyamatokban a minőségellenőrzés kiemelkedő fontossággal bír annak érdekében, hogy a gyártott termékek megfeleljenek a szigorú minőségi szabványoknak. Ezen szabványoknak való megfelelés nem csupán a vállalat hírnevét erősíti, hanem a végfelhasználók elégedettségét is biztosítja. Ebben a kontextusban egyre növekvő figyelmet kapnak az ipari PC-k és programozható logikai vezérlők (PLC) alkalmazásai, melyek forradalmasítják a minőségellenőrzési folyamatokat.

Az ipari PC-k és PLC-k összekapcsolása a gyártósorokon kiválóan ötvözi a számítógép- és automatizálási technológiák előnyeit. Az ipari PC-k rugalmasságot és számítási kapacitást biztosítanak a magas szintű adatfeldolgozás és analitika elvégzéséhez, míg a PLC-k a valós idejű vezérlés és automatizálás terén emelkednek ki. E két technológia együttes alkalmazása lehetővé teszi a gyártók számára, hogy a minőségellenőrzést precízen és hatékonyan hajtsák végre.

A szakdolgozatom célja, hogy áttekintést nyújtson az ipari PC-k és PLC-k integrált felhasználásának előnyeiről a minőségellenőrzés területén. A továbbiakban részletesen vizsgáljuk ezeknek a technológiáknak az alkalmazási területeit, az elért eredményeket, valamint a gyártási folyamatok hatékonyságára és termékek minőségére gyakorolt hatásait.

A különböző iparágakban egyre növekvő versenyhelyzetben az ipari PC-k és PLC-k által támogatott minőségellenőrzési rendszerek kulcsfontosságúak a vállalati siker szempontjából. Ennek megértése és alkalmazása nem csupán technológiai előnyt jelent, hanem stratégiai tényezőt is a gyártási folyamatok optimalizálásában és a kiváló minőségű termékek előállításában.

A következő fejezetekben részletesen elemzésre kerül az ipari PC-k és PLC-k alkalmazása a minőségellenőrzés területén, bemutatva azok potenciálját és kihívásait. A szakdolgozat célja, hogy hozzájáruljon a gyártási folyamatok hatékonyságának és termékek minőségének javításához, azáltal, hogy kiemeli az integrált technológiai megoldások előnyeit a minőségellenőrzés terén.

Az ipari PC-k és PLC-k integrált felhasználása a minőségellenőrzés területén nem csupán a gyártási folyamatok hatékonyságát növeli, hanem lehetőséget teremt a valós idejű adatgyűjtésre és elemzésre is. Az ipari PC-k által biztosított magas számítási kapacitás lehetővé teszi olyan összetett algoritmusok futtatását, amelyek azonnali visszajelzést adnak a termelés minőségéről. Ezenkívül az ipari PC-k könnyen integrálhatók az adatfelhőbe, ami lehetőséget nyújt a távoli hozzáférésre és az adatok megosztására, felgyorsítva ezzel a döntéshozatali folyamatokat.

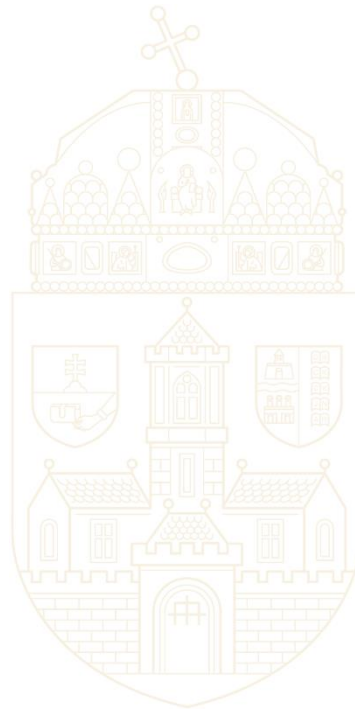
A PLC-k szerepe is kiemelkedő a minőségellenőrzési rendszerekben, különösen a gyártósorok automatizálásában és a szenzorok, mérőeszközök kezelésében. A PLC-k nem



csupán a gépi folyamatokat felügyelik, hanem a rendszeres karbantartást és a hibaelhárítást is segítik, ami minimalizálja a leállásokat és növeli a termelékenységet.

Az alkatrészek vizsgálata egy fontos lépés, ahol az ipari PC-k segítségével könnyedén összehasonlíthatók a valós idejű szenzoradatok az előre meghatározott referenciaértékekkel. Ezáltal a rendszer azonnal azonosítja a potenciális hibákat vagy eltéréseket a gyártási folyamatban.

Ennek eredményeként minimalizálódik a humán beavatkozásból eredő hibalehetőségek száma, és a minőségellenőrzési rendszer még hatékonyabbá válik. Ezen két technológia együttes alkalmazásával a gyártók garantálhatják, hogy az alátétek minden egyes darabja megfelel a szigorú minőségi előírásoknak, hozzájárulva ezzel a végtermék kiváló minőségéhez az autóiipari szektorban.

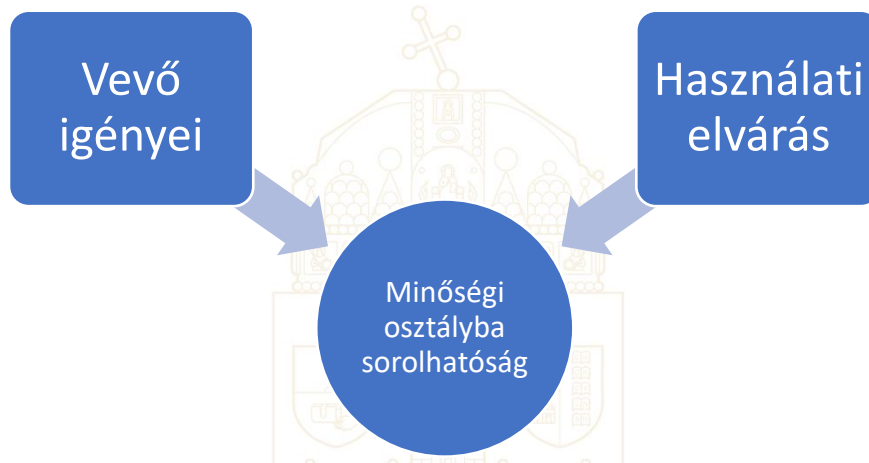


## 2. A MINŐSÉG FOGALMA, CSOPORTOSÍTÁSA

### 2.1. A MINŐSÉG FOGALMA

A minőség a termékek és szolgáltatások jellemzőinek összessége, amelyek meghatározzák azok képességét a fogyasztói elvárásoknak való megfelelésre. Ez nem csak az anyagok és gyártási folyamatok minőségét jelenti, hanem magában foglalja az ügyfélszolgálatot, a tervezést, a műszaki támogatást és minden olyan tényezőt, amely befolyásolja a végső felhasználó elégedettségét. [1]

A minőség nem csupán a hibák hiányát jelenti; inkább a megfelelési szintet, az elvárásoknak való megfelelést tükrözi. A kiváló minőségű termékek és szolgáltatások hozzájárulnak a vállalat hírnevéhez és a vevői lojalitáshoz. [1]



*1. ábra Minőségi osztályba sorolás feltételei Saját ábra*

### 2.2. MINŐSÉGELLENŐRZÉS KÉT NAGY CSOPORTJA

Az ipari minőségellenőrzés csoportok azok a szervezeti egységek vagy csapatok, amelyek felelősek a termékek vagy folyamatok minőségének ellenőrzéséért és biztosításáért az ipari környezetben. Ezek a csoportok kulcsfontosságú szerepet játszanak az ipari folyamatok során, mivel felelősek a termékek megfelelőségének, szabványoknak való megfelelőségéért és az általános minőségirányításért. [1]

A minőségellenőrzés két alapvető csoportba sorolható: szubjektív és objektív minőségellenőrzés.

#### 2.2.1. Szubjektív minőségellenőrzés

A szubjektív minőségellenőrzés azt jelenti, hogy a minőség értékelése az emberi érzékszervek és tapasztalatok alapján történik. Ebben a folyamatban olyan tulajdonságokat vizsgálnak, amelyek nem mérhetők műszerekkel, és inkább a humán értékelésen alapulnak. [1]



## 1. Szemrevételezés

A szemrevételezés a termék vagy folyamat vizuális elemzését jelenti. Ez lehet például egy gyártási folyamatnál az alkatrészek külsejének ellenőrzése, a színskombinációk értékelése vagy a felületi hibák vizsgálata. Az emberi szem pontosan érzékel bizonyos jellemzőket, amelyek fontosak lehetnek a minőség szempontjából. [1]

## 2. Érzékszervi értékelés

Az érzékszervi értékelés magában foglalja az összetettebb érzékszervi élményeket, például az ízlelést vagy az illatokat. Élelmiszeriparban gyakran alkalmazzák, ahol az emberi ízlés és illatérzékelés kulcsfontosságú a termék minőségének meghatározásában. [1]

A szubjektív minőségellenőrzés során az emberi tapasztalat és szakértelem nagyban meghatározza a végeredményt, és a minőségértékelés szubjektív jellege miatt fontos, hogy az értékelők megfelelő képzésben és szabványoknak megfelelően végezzék el feladataikat. [1]

### 2.2.2. Objektív minőségellenőrzés

2.2.2. Objektív minőségellenőrzés: Az objektív minőségellenőrzés a műszerek és technológiai eszközök alkalmazásán alapul, hogy a fizikai és kémiai jellemzőket mérjük. Ebben az esetben a mérőeszközök és laboratóriumi berendezések használata biztosítja a precíz és ismételhető méréseket. [1]

#### 1. Fizikai jellemzők mérése

Az objektív minőségellenőrzés mérőeszközök segítségével rögzíti a fizikai tulajdonságokat, például a méreteket, alakot, súlyt stb. Gyakran alkalmazzák gyártási folyamatokban és gyártási vonalakon a konzisztencia és az előre meghatározott paraméterek betartásának biztosítására. [1]

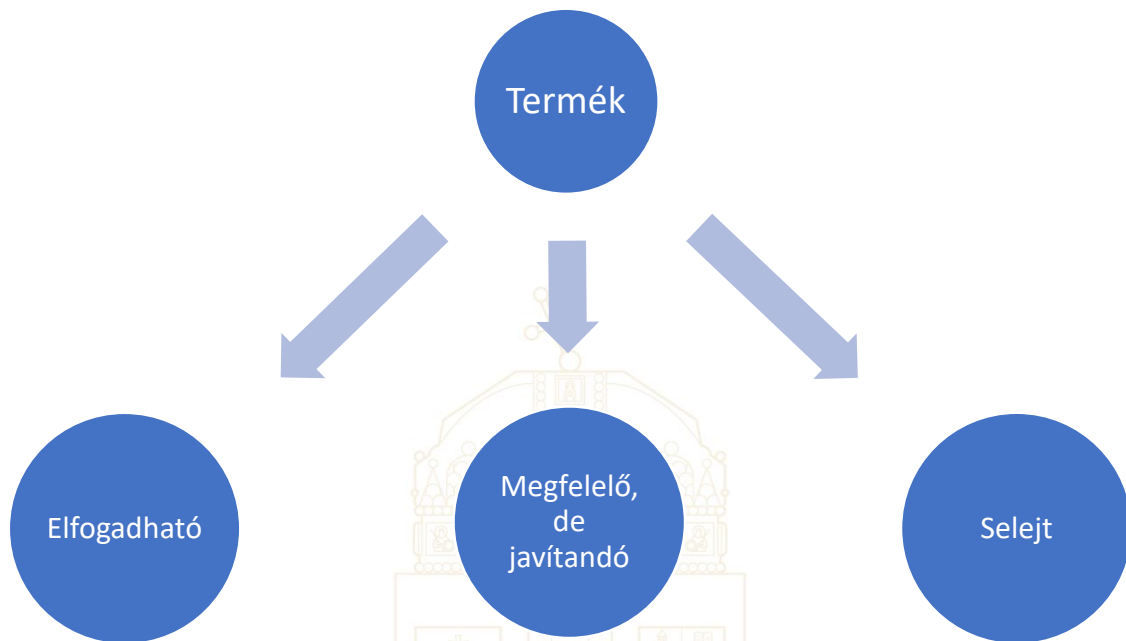
#### 2. Kémiai tulajdonságok mérése

Az objektív minőségellenőrzés laboratóriumi berendezéseket és kémiai elemzéseket használ, hogy megmérje a termék kémiai tulajdonságait. Ez különösen fontos az élelmiszeriparban, gyógyszeriparban és más iparágakban, ahol a kémiai stabilitás és összetétel kulcsfontosságú a minőség szempontjából. [1]

Az objektív minőségellenőrzés általában szigorúan szabályozott paraméterekre és előre meghatározott toleranciákra támaszkodik, és ezáltal objektív kritériumokat biztosít a minőség méréséhez és értékeléséhez. Ez hozzájárul a konzisztens és következetes minőségfenntartáshoz az ipari folyamatok során. [1]

## 2.3. MINŐSÉGELLENŐRZÉSI DÖNTÉSEK

A minőségellenőrzés során meghozott döntések alapvetően befolyásolják a termék sorsát. Az elvégzett ellenőrzések eredményeként a termék vagy szolgáltatás lehet elfogadható, megfelelő, vagy selejt. [1]



2. ábra Termékek besorolása Saját ábra

### 1. Elfogadható termék

Ha a minőségellenőrzés során a termék megfelel minden elvárásnak és szabványnak, elfogadhatónak minősül. [1]

### 2. Megfelelő, de javítandó termék

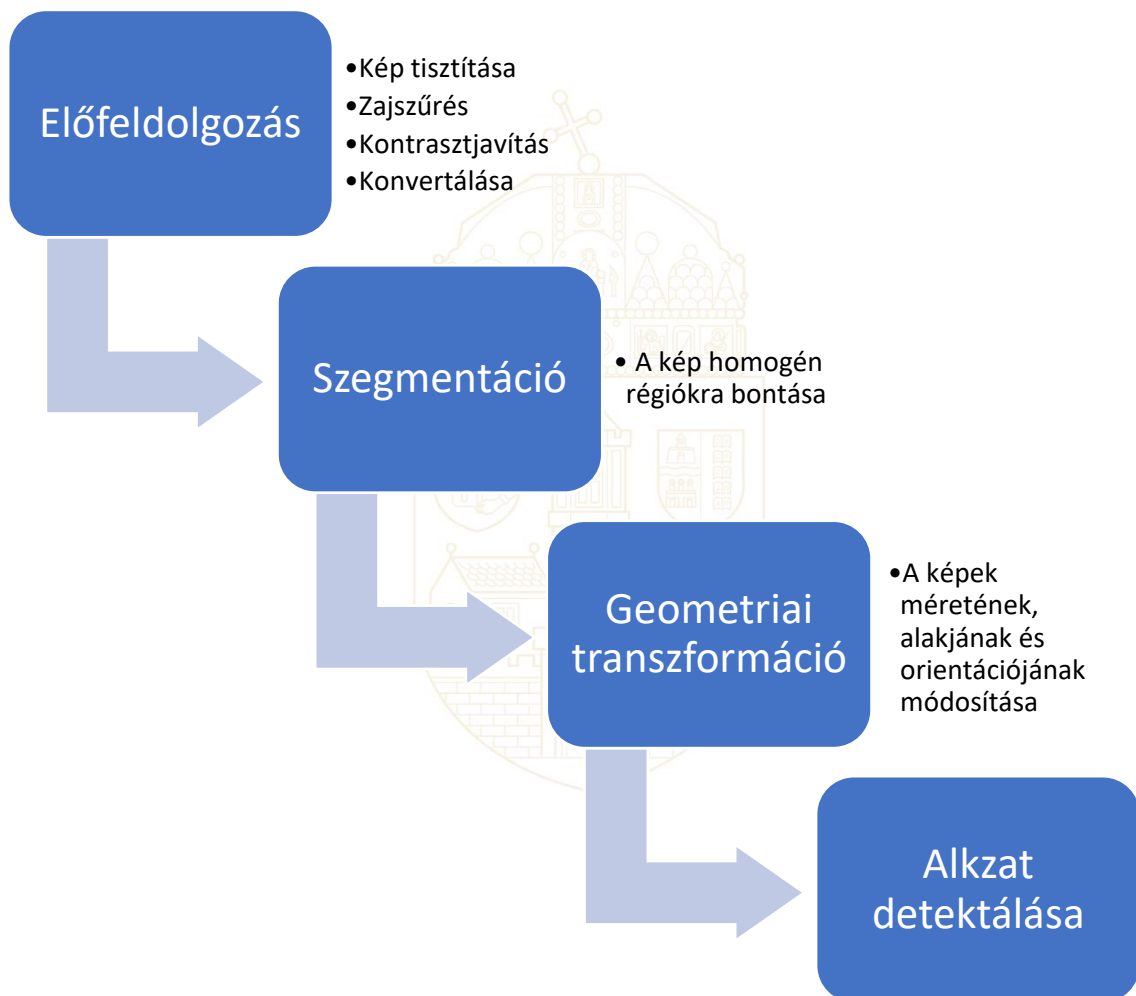
Amennyiben a termék megfelel a minimumkövetelményeknek, de kisebb hibákat vagy eltéréseket észlelnek, esetlegesen javításra szorulhat. [1]

### 3. Selejt termék

A selejtezés akkor következik be, ha a termék nem felel meg a minőségi előírásoknak, és az esetleges hibák, hiányosságok miatt nem használható. [1]

### 3. KÉPFELDOLGOZÁS AZ IPARBAN

A képfeldolgozás a digitális képek elemzésével foglalkozó terület, amely számos ipari, orvosi és informatikai alkalmazásban játszik kiemelkedő szerepet. A képfeldolgozási folyamat számos lépésből áll, kezdve az előfeldolgozással, amely a képek tisztítását, zajszűrését és kontrasztjavítását foglalja magában. Ezt követi a jellemzők kinyerése, ahol a rendszer különböző tulajdonságokat, például színeket és textúrákat azonosít, hogy segítse a későbbi elemzést. [2]



3. ábra A képfeldolgozás lépései Saját ábra

A szegmentáció során a képeket homogén régiókra bontják, ami megkönnyíti az egyes objektumok vagy régiók kezelését és azonosítását. Az objektumfelismerés további

mérföldkő, ahol a rendszer azonosítja és osztályozza a képeken található objektumokat. Az objektumfelismerés kulcsfontosságú az olyan alkalmazásokban, mint az autonóm járművek, arcfelismerés vagy orvosi képfeldolgozás. [2]

A képfeldolgozás egy másik aspektusa a geometriai transzformációk alkalmazása, amelyek segítségével módosíthatjuk a képek méretét, alakját és orientációját. A modern képfeldolgozási módszerek között az egyre széleskörűbben alkalmazott gépi tanulás található, ahol az algoritmusok tanulják meg a képek jellemzőit és összefüggéseit. [2]

A háromdimenziós (3D) képfeldolgozás olyan terület, ahol az objektumokat térbeli információkkal rendelkező képek alapján elemzik. Emellett a mozgásanalízis is fontos képfeldolgozási módszer, amely a mozgásérzékelés és követés révén elemzi a dinamikus változásokat a képeken. [2]

Összességében a képfeldolgozási módszerek széles körben alkalmazhatók és folyamatosan fejlődnek. Az előrehaladó technológiák és a gépi tanulás további lehetőségeket teremtenek a képfeldolgozás terén, elősegítve az innovációt és a fejlett alkalmazások létrehozását a különböző iparágakban. [2]

### 3.1.ESZKÖZÖK

#### 3.1.1. Ipari okoskamera

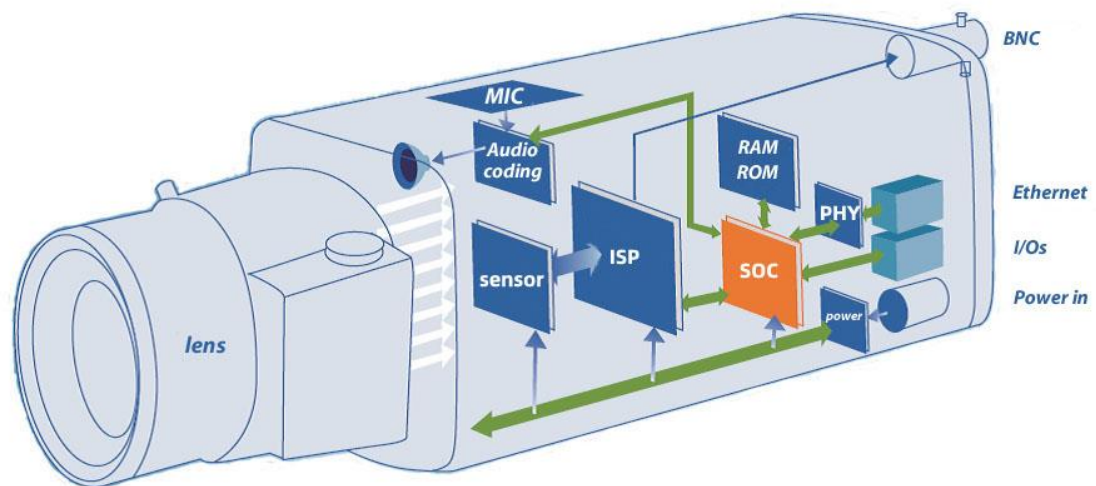
Az "ipari okoskamera" (industrial smart camera) egy olyan speciális típusú kamera, melyet kifejezetten ipari környezetekben való alkalmazásra terveztek. Ezek a kamerák olyan fejlett funkciókkal rendelkeznek, amelyek lehetővé teszik az ipari folyamatok hatékonyabb ellenőrzését, automatizálását és minőségellenőrzését.



4. ábra Ipari okoskamera [13]

Az ipari okoskamerák általában rendelkeznek beépített számítási képességekkel, amelyek lehetővé teszik számukra a képfeldolgozási és elemzési feladatok közvetlenül a kamera eszközén belül. Ez azt jelenti, hogy a kamera önállóan képes döntéseket hozni és adatokat generálni, anélkül, hogy külső számítógépre lenne szükség.

Emellett az ipari okoskamerák gyakran tartalmaznak ipari kommunikációs protokollokat, például Ethernet/IP-t vagy PROFINET-et, amelyek lehetővé teszik a szoros együttműködést más ipari automatizálási rendszerekkel. Ez lehetővé teszi a kamerák integrációját a gyártási folyamatokba és a felsőbb szintű irányító rendszerekbe.



5. ábra Az ipari kamera felépítése [14]

Az ipari okoskamerákat sokféle alkalmazási területen használják, beleértve a minőségellenőrzést, alkatrészazonosítást, követést, rendszerek ellenőrzését és a gyártás folyamatainak felügyeletét. A magas szintű automatizáció és a pontos adatgyűjtés lehetőségét biztosítják, ami kiemelkedően fontos az ipari folyamatok hatékony működése és a gyártott termékek minőségének biztosítása szempontjából.

Az ipari környezetekben történő alkalmazásra tervezve ezek a kamerák rendszerint rendelkeznek különleges tulajdonságokkal, például ellenállósággal a porral, nedvességgel és magas hőmérsékletekkel szemben. Ez biztosítja, hogy a kamerák megbízhatóan működjenek a gyakran szélsőséges ipari környezetekben is.

Az ipari okoskamerák elterjedése az ipari automatizációban kulcsfontosságú, mivel hozzájárulnak a termelékenység növeléséhez, a minőség javításához és az automatizált folyamatok hatékonyabbá tételéhez. Az önálló képfeldolgozás lehetővé teszi az azonnali visszajelzéseket és a hibák gyors azonosítását, ami jelentősen csökkenti a leállási időt és optimalizálja a gyártási folyamatokat.

Az ipari okoskamerák tehát kiemelkedő szerepet játszanak az ipari automatizáció terén, mivel kombinálják a fejlett képfeldolgozási technológiát és a kommunikációs



képességeket, hogy hozzájáruljanak a modern gyártási folyamatok hatékony és megbízható irányításához.

Az ipari okoskamerák általában rendelkeznek flexibilis konfigurációs lehetőségekkel, amelyek lehetővé teszik az alkalmazkodást különböző ipari környezetekhez és alkalmazásokhoz. Ezenkívül a skálázhatóság lehetőséget biztosít arra, hogy a rendszer könnyen bővíthető legyen a növekvő gyártási igényekhez.

Az ipari okoskamerák gyakran rendelkeznek valós idejű funkciókkal, amelyek lehetővé teszik az azonnali döntéseket és reakciókat. Ez kulcsfontosságú az olyan gyártási folyamatoknál, ahol azonnali beavatkozásra van szükség a hibák kijavításához vagy a rendszer optimalizálásához.

Egyes ipari okoskamerák támogatják az integrált működést a felhőalapú rendszerekkel. Ez lehetővé teszi a távoli elérhetőséget, valós idejű adatok megtekintését és elemzését, valamint a távoli konfigurációt. A felhőintegráció hozzájárulhat a távoli felügyelethez és az adatok analitikai kihasználásához.

Az egyes ipari okoskamerák képesek alkalmazni mesterséges intelligencia (AI) és gépi tanulás (ML) technológiákat. Ez növelheti a kamerák önálló döntési képességeit, és lehetővé teszi az adaptívabb és intelligensebb működést, például az automatikus hibadetekció vagy a rendszer optimalizálása terén.

Az ipari okoskamerák tervezése során gyakran figyelnek az energiahatékonyságra is. Alacsony energiafogyasztásuk lehetővé teszi hosszabb üzemidőt és csökkenti az energiafelhasználási költségeket, különösen olyan területeken, ahol nagy számú kameraegység működik egyszerre.

### 3.1.2. Ipari PC

A SIMATIC IPC-ket komplex automatizációs és vizualizációs feladatokhoz tervezték. A rendszerek alapja a legmodernebb PC technológia, ideértve az Intel Core i processzorokat (6. generáció) Turbo Boost 2.0-val, hyper-threading-gel és virtualizációs technológiával. [4]



6. ábra Siemens ipari rack PC [15]

A processzorba integrált memória- és grafikavezérlő kivételes memória- és grafikateljesítményt biztosít, és a maximális teljesítmény érdekében Intel C236 chipsettel, DDR4 memóriával, amely támogatja a kétsatornás technológiát. [4]

A rendszerek magas adatátviteli sebességeket kínálnak, beleértve a serial ATA szilárdtest-meghajtót akár 480 GB-ig, serial ATA merevlemezeket akár 2 TB-ig, Dupla Gigabit Ethernet-t és a PCI Express 3.0 technológiát. [4]

Az uptime csökkentése érdekében a SIMATIC IPC-k 24 órás üzemre tervezett, magas rendelkezésre állású rendszerek, hatékony önellenőrzéssel rendelkeznek. [4]

A RAID1 (tükördisk rendszer) vagy RAID5 (striping pár) révén magas adatbiztonságot biztosít, további tartalék merevlemezzel. A "hot swap" meghajtóhelyek lehetővé teszik a merevlemez cseréjét működés közben. A rendszer gyors azonosítást és merevlemezcsere lehetőségét kínálja a HDD riasztó LED-ek segítségével RAID konfigurációk esetén. Egyéb jellemzők közé tartozik az SSD opció RAID1 mellett, a redundáns tápellátás "hot swap" funkció, ami a tápegységmodul cseréjét teszi lehetővé működés közben. [4]



A SIMATIC IPC-k rendkívül rugalmas és felhasználóbarát a beüzemelés, üzemeltetést és karbantartást tesznek lehetővé az előre telepített és aktivált operációs rendszer, valamint az Intel Active Management Technology (IAMT) funkció segítségével. A rendszer gyorsan helyreállítható a vele szállított helyreállító DVD segítségével, alacsony zajszinttel és egyedi kialakítással rendelkezik. Univerzálisan implementálható ipari munkaállomásként vagy szerverként. [4]

Az Intel Hybrid többmonitoros támogatás akár 5 monitorral való csatlakozáshoz opcionális PCIe x16 grafikus kártyával. A rendszer rendkívül flexibilis, bővíthető integrált interfészekkel rendelkezik, PCI és PCI Express. [4]

A SIMATIC IPC magas ipari kompatibilitással és kompakt kialakítással rendelkezik, kiváló elektromágneses kompatibilitással és korrózióállósággal ipari környezetekben és háztartási, üzleti és kereskedelmi környezetekben való használatra. [4]

Ezenkívül megfelel a porvédelemnek a túlnyomásos szellőztetési koncepcióval, elől elhelyezett ventilátorral és por szűrővel, speciális merevlemeztartókkal és kártya tartókkal a rezgés és ütés elleni védelem érdekében. [4]

A SIMATIC IPC jelentős költségcsökkentést tesz lehetővé a magas beruházási biztonság révén, legalább 2 év elérhetőséggel és 3 év garanciával a pótalkatrészekre. A rendszert SIMATIC komponensekkel tesztelték és világszerte cULus tanúsítvánnyal rendelkezik. [4]

A rendszer támogatja a régi interfészeket (PS/2, COM, LPT), kompatibilis az eszközgenerációk közötti telepítéssel, és világszerte elérhető szerviz és támogatás mellett biztosítja a felhasználók számára a biztonságos és hatékony használatot. [4]

### 3.2.MEGVILÁGÍTÁSI MÓDSZEREK

A megvilágítás kulcsfontosságú tényező a képfeldolgozásban, és meghatározza, hogy milyen minőségben tudjuk mérni a vizsgált tárgy tulajdonságait. A megvilágításnak számos változata létezik, és ezek kombinációja lehet a legoptimálisabb eredmények eléréséhez. [3]

#### 1. Felső Megvilágítás

A felső megvilágítás során a tárgyat felülről világítjuk meg. Ennek két alapvető esete lehet: sötét látómező és világos látómező. A sötét látómező esetén, ha a kamera optikai tengelye közel merőleges a tárgy felületére, és a fényforrás iránya kicsi szöget zár be azzal, a kép sötét lesz. Világos látómező esetén a kamera felé visszaverődő fénysugarak miatt a látómező világos lesz. A megvilágítás fénysugarai lehetnek párhuzamosak vagy diffúzok, és a diffúz fény különösen alkalmas a repedések és mélyedések detektálására. [3]

## 2. Alsó Megvilágítás

Az alsó megvilágítás előnyös lehet több esetben is. Például átlátszó objektumok esetén az átvilágítás segíthet a belső részek jól láthatóvá tételében. Ha a tárgyon lyukak vannak, vagy a körvonalát szeretnénk ellenőrizni, az alsó megvilágítás kiválóan alkalmazható. A hátulról jövő fény lehet diffúz vagy kollimátor használatával párhuzamos, amely elősegíti az alakra vonatkozó pontos méréseket a képfeldolgozó algoritmusok számára. [3]



7. ábra Kondenzátor képe normál háttérmegvilágítás és kollimátor használatával [3]

## 3. Kombinált Megvilágítás

A legjobb eredmények elérése érdekében gyakran alkalmaznak kombinált megvilágítást, amely különböző típusú megvilágítások kombinációját jelenti. Például felülről történő megvilágítást lehet párhuzamos alsó megvilágítással kombinálni az optimális mérési eredmények elérése érdekében.

Ez a megvilágítási technikák kiválasztása alapvető fontosságú a sikeres képfeldolgozáshoz és a pontos mérések eléréséhez. A különböző megvilágítási beállítások lehetővé teszik a rendszer számára, hogy lássuk a tárgyakat különböző szögekből, és így optimalizálhassuk a képalkotást és a minőségi méréseket.

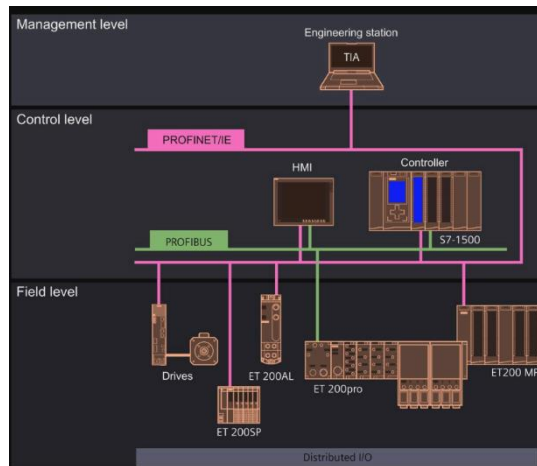
Az általános szabályok és a specifikus alkalmazásokhoz szükséges megvilágítási technikák megismerése segít abban, hogy a képfeldolgozási rendszer hatékonyan alkalmazható legyen az ipari környezetekben és más területeken. [3]

| Termék/feladat                                  | Megvilágítás típusa                                    |
|-------------------------------------------------|--------------------------------------------------------|
| íves felületű termékek                          | diffúz megvilágítás                                    |
| közel sima, de megbillenthető, fényes felületek | diffúz megvilágítás                                    |
| karcok detekciója közel sima felületeken        | koaxiális vagy<br>sötét látómezős megvilágítás         |
| termék meglétének detekciója                    | alsó megvilágítás vagy<br>sötét látómezős megvilágítás |
| átlátszó termék vizsgálata                      | alsó megvilágítás,<br>esetleg kollimátorral            |

8. ábra Néhány tipikus feladathoz a megfelelő megvilágítás [3]

## 4. AZ IRÁNYÍTÁSTECHNIKAI PIRAMIS

Az irányítástechnológiában az irányítás különböző szintjeit különböző módon nevezik el. A "management level," "control level," és "field level" kifejezések az irányítástechnikai



9. ábra Az irányítástechnikai piramis [5]

piramis alapján hivatkoznak az irányítási szintekre. Az irányítási szinteket általában az alábbi módon nevezik el:

### 1. Management Level (Menedzsment Szint):

Ez a legfelső irányítási szint, ahol a stratégiai és üzleti döntések születnek. Ezen a szinten az irányító rendszer kommunikál az üzleti tervezéssel és a vállalatirányítási rendszerekkel.

### 2. Control Level (Vezérlési Szint):

A középső szint, ahol a folyamatokat és rendszereket irányítják. Az irányító rendszer itt dolgozza fel az információkat a terepen lévő szenzoroktól, és döntéseket hoz a rendszer irányítására.

### 3. Field Level (Terepi Szint):

A legalsó szint, ahol a fizikai folyamatok és eszközök találhatók. Ezen a szinten történik a terepi adatgyűjtés, a szenzorok és aktuátorok vezérlése. **Hiba! A hivatkozási forrás nem található.**

Ezek a kifejezések segítenek leírni és elkülöníteni az irányítás különböző szintjeit az irányítástechnikai rendszerekben. A piramis struktúra azt tükrözi, hogy minél magasabbra emelkedünk a piramison, annál inkább a stratégiai és üzleti döntéseket, míg minél lejjebb megyünk, annál inkább a fizikai folyamatok irányításával és ellenőrzésével foglalkozunk. **Hiba! A hivatkozási forrás nem található.**

## 5. SIEMENS S7-1500 TERMÉKCSALÁD

A SIMATIC S7-1500 automatizálási rendszer megfelelő rugalmasságot és teljesítményt kínál számos vezérlőalkalmazáshoz gép- és üzemtervezési területen. A skálázható konfiguráció lehetővé teszi, hogy a PLC-t a helyi feltételekhez igazítsa az Ön számára. [5]



10. ábra SIMATIC S7-1500 PLC CPU-k [5]

### 5.1.S7-1500 CPU

Az S7-1500-ban elérhető standard motion control és technológiai funkciók mellett a SIMATIC S7-1500 technológiai CPU-k további funkciókat kínálnak Önnek, például kibővített szinkron működést és kinematikai funkciókat. A SIMATIC S7-1500 automatizálási rendszer megfelel az IP20 fokú védelemnek, és száraz környezetben való használatra és vezérlőszekrénybe való telepítésre szolgál. A SIMATIC S7-1500R/H CPU-k (redundáns vagy hibatűrő CPU-k) lehetőséget biztosítanak rendszerének elérhetőségének növelésére. Annak érdekében, hogy szükség esetén átválthasson a primer CPU-ról a másodlagos CPU-ra, a felhasználóprogramot szinkronban dolgoztatják el két CPU-n. [5]

A rendszer moduláris szerkezetet használ, ami lehetőséget biztosít a rendszer további bővítésére további modulokkal és kiegészítőkkel. Az integrált biztonsági funkciók



11. ábra S7 1500 1518 MFP PLC és néhány modullal [5]

szolgáltatják a gép- és felhasználóvédelmet, miközben a TIA Portal integrált fejlesztői környezet egy egységes platformot biztosít a programozáshoz és konfigurációhoz. [5]

## 5.2.S7-1500 1518 MFP PLC

Az S7-1500 1518 MFP PLC a Siemens Totally Integrated Automation (TIA) rendszerének kiemelkedő modellje, a magas szintű vezérlési feladatokhoz és kiterjedt ipari alkalmazásokhoz tervezve. A rendszer teljesítményének magas szintje, a korszerű multicore technológia alkalmazása és a gazdag erőforrások lehetővé teszik a bonyolult vezérlési folyamatok gyors és hatékony végrehajtását. [6]

Ezen kívül az S7-1500 1518 MFP modell kiterjedt kommunikációs képességekkel rendelkezik, támogatva az Ethernet, a Profinet és más ipari kommunikációs protokollokat. Ez a sokoldalúság lehetővé teszi a PLC integrációját a gyártási hálózatokba, hozzájárulva a gyártási folyamatok összehangolt működéséhez. [6]

Az S7-1500 1518 MFP PLC kifejezetten támogatja a gyártási folyamatokat és hozzájárul a termelékenység növeléséhez. Az egységes fejlesztői környezet, a magas szintű biztonsági funkciók és a kiterjedt kommunikációs lehetőségek révén ez a PLC modell megfelel az ipari kihívásoknak és számos ipari szektorban alkalmazható a megbízható és hatékony automatizálás érdekében. [6]

## 5.3.TM-MFP

A TM MFP (Multifunkcionális Platform technológiamodul) egy olyan modul az automatizálási mérnöki alkalmazásokban, ami azt jelenti, hogy olyan alkalmazásokhoz használható, amelyek közvetlenül kiértékelik és továbbfeldolgozzák az adatokat és információkat a CPU-ból, de nem integrálódnak közvetlenül a tényleges vezérlési feladatba. A technológiamodul lehetővé teszi a csatlakoztatott érzékelők, például kamerák és robotrendszerek adatainak feldolgozását, valamint az adatokat a CPU felhasználói



12. ábra TM-MFP modul [8]

programjából. A TM MFP-en előre telepítették a Linux alapú SIMATIC Industrial OS (IndOS) operációs rendszert. [7]

A beépített processzor Graphics Processing Unit (GPU) képes futtatni és gyorsítani az AI rutinokat. Ez lehetővé teszi nagy mennyiségű adat gyors és hatékony feldolgozását, valamint folyamatok optimalizálását. [7]

A technológiamodul támogatja a Docker szoftvert (alapértelmezetten nincs előre telepítve) szoftverek és algoritmusok betöltéséhez, például a TM MFP-be bináris formában. [7]

A TM MFP technológiai modul szorosan beépíti az Edge Computing funkcionalitását a SIMATIC S7-1500 termékcsaládba. Az általánosan felhasználható TM MFP platform lehetővé teszi, hogy különböző, egymástól független alkalmazásokat integráljunk, mint például:

- Rugalmas használat opcionális szoftvercsomagokkal, például Proneta esetén
- Lehetőség SIMATIC Flow Creator és más Edge alkalmazások használatára
- Nyitottság testreszabott magas szintű programnyelv-kiterjesztésekhez, például C/C++ és Python
- Kizárólagos interfészek használata speciális alkalmazásokhoz

A technológiamodul TM MFP a SIMATIC Industrial OS alapján készült, és az Edge futási idő révén kifejezetten fel van készítve a Siemens Industrial Edge alkalmazások irányítási szintjéhez. A modul központi és skálázhatóan csatlakoztatható bármely SIMATIC S7-1500 CPU-hoz (1511-től 1518-ig). Az alaplap busz lehetővé teszi hatékony adatcserét valós időben. [7]

A multifunkcionális platform előnyeit ötvözi a felhő megoldások előnyeivel és a helyi megoldások erősségeivel. Ezenkívül különböző adatmentési koncepciók alkalmazhatók, beleértve a helyi, felhőalapú és hibrid megoldásokat. [7]

A TM MFP előre telepített és konfigurált Linux-alapú SIMATIC Industrial OS operációs rendszerrel rendelkezik. A kommunikáció a TM MFP és a CPU között I/O kommunikációval az hátlapi buszon keresztül, TCP, UDP. A TM MFP lehetővé teszi a következő alkalmazások csatlakozását és implementálását: [7]

- Felhőrendszerek
- Robotrendszerek
- Optikai rendszerek
- Mesterséges intelligencia (AI)
- Magasszintű programozási nyelvek: C, C++ és Python
- MathWorks MATLAB

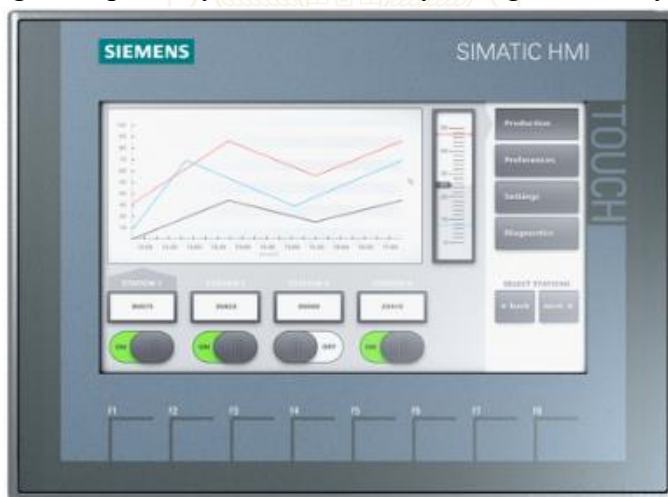
## 5.4.SIMATIC HMI

A SIMATIC HMI (Human-Machine Interface) a Siemens által kifejlesztett és gyártott ipari automatizálási megoldás, melynek célja az ember-gép interfész hatékony és intuitív kialakítása az ipari környezetben. Az ipari automatizálás terén a SIMATIC HMI eszközök kulcsfontosságú szerepet játszanak, lehetővé téve az operátoroknak, hogy egyszerűen és hatékonyan kommunikáljanak a gépekkel és az automatizált rendszerekkel. [8]

Ezen eszközök között szerepelnek érintőképernyők és Panel PC-k, melyek lehetővé teszik az operátoroknak, hogy intuitív érintéssel interakcióba lépjenek a rendszerekkel. A SIMATIC HMI kínálata széles skáláját foglalja magába a különféle méretű és típusú kijelzőknek, így alkalmazkodik az ipari környezetek változatos igényeihez. [8]

A rendszer támogatja a SCADA (Supervisory Control and Data Acquisition) technológiát is, amely lehetővé teszi az operátoroknak, hogy felügyeljék és ellenőrizzék a folyamatokat, valamint gyűjtsék és elemezzék a termelési adatokat. Emellett a SIMATIC HMI könnyen integrálható más Siemens automatizálási eszközökkel, például a SIMATIC PLC-kkel (Programmable Logic Controller), lehetővé téve egy összehangolt és egységes automatizációs rendszer kialakítását. [8]

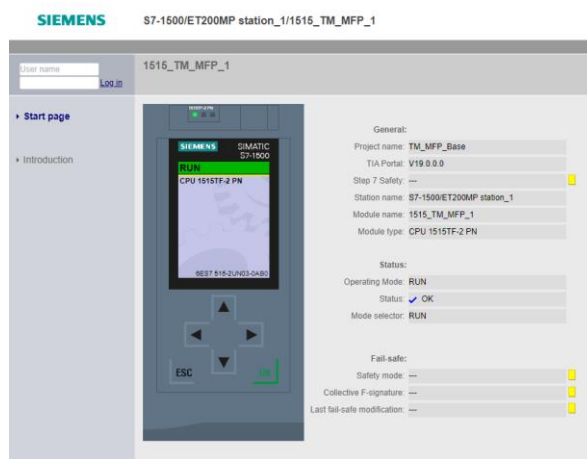
A rugalmas programozhatóság lehetőséget ad a felhasználóknak saját alkalmazások és felhasználói felületek kialakítására, ezáltal testreszabhatóvá téve a megoldást az egyedi ipari igényekhez. Az adatgyűjtés és riportolás funkciók segítenek az információk hatékony elemzésében és a döntéshozatalban, hozzájárulva a gyártás hatékonyságának növeléséhez. [8] Összességében a SIMATIC HMI rendszer a modern ipari automatizáció kihívásaira válaszolva fejlett technológiát kínál, amely optimalizálja az ember-gép interfészt, és elősegíti az ipari folyamatok hatékony és rugalmas irányítását. [8]



13. ábra Standard simatic HMI [16]

## 5.5. WEB SZERVER

A Siemens S7-1500 OLC (Online Change) Web szervere olyan kulcsfontosságú elem, amely a Siemens SIMATIC S7-1500 sorozatú programozható logikai vezérlők számára fejlesztett egyedi funkció. Ennek a technológiának a fő célja a távoli hozzáférés és konfiguráció lehetőségeinek kibővítése, amelyeket egy intuitív webes felületen keresztül lehet elérni. Ezt az újítást a Siemens TIA (Totally Integrated Automation) Portal



14. ábra A webszerver online felülete Forrás: TIA

softvercsaládjában kezelik, és rendkívül fontos szerepet játszik az ipari automatizációs folyamatok optimalizálásában. [9]

Az egyik kiemelkedő jellemzője a S7-1500 OLC Web szervernek a távoli elérés lehetősége. Ez azt jelenti, hogy a felhasználók képesek távolról bejelentkezni a vezérlőhöz, és konfigurálni, illetve módosítani a vezérlő programjait anélkül, hogy fizikailag jelen lennének a helyszínen. Ez a rugalmasság különösen hasznos lehet olyan ipari környezetekben, ahol a vezérlők elhelyezkedése távoli vagy nehezen hozzáférhető területeken történik. [9]

A S7-1500 OLC Web szerver rendelkezik egy felhasználóbarát webes felülettel, amely egyszerű és hatékony módját kínálja a vezérlő programjainak és beállításainak kezelésére. A grafikus interfész segítségével a felhasználók könnyen navigálhatnak és módosíthatnak a programokon, így a konfigurációs folyamat gyors és gördülékeny. [9]

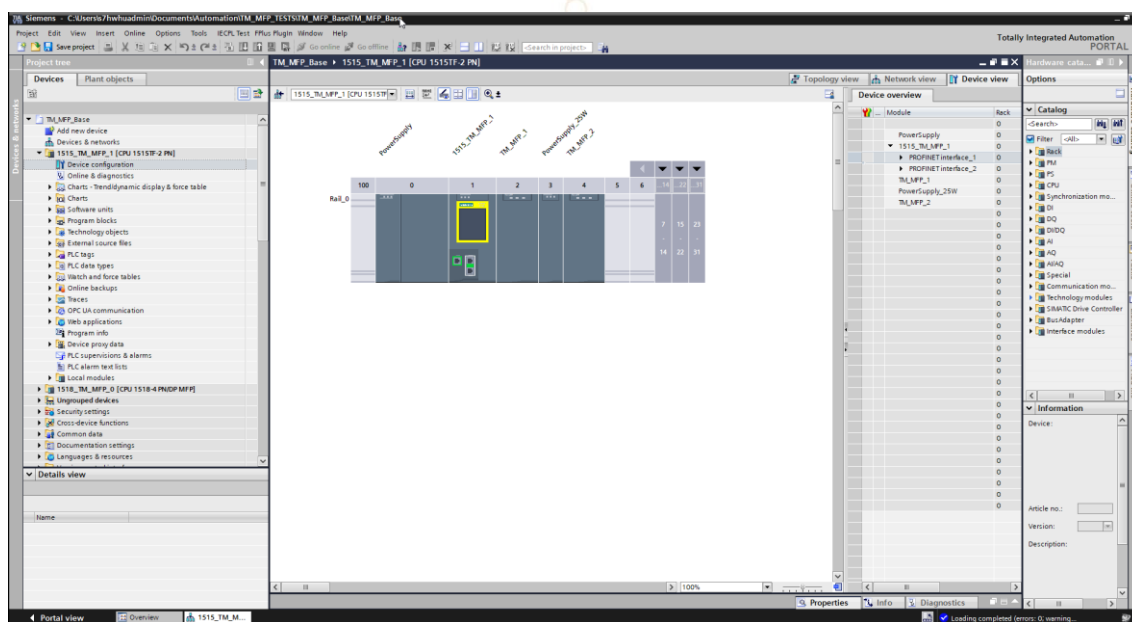
Az "Online Change" (Éles változtatás) funkció kiemelkedő előnye, mivel lehetővé teszi a programok módosítását anélkül, hogy le kellene állítani a vezérlő működését. Ez kritikus fontosságú az ipari folyamatok folytonosságának és hatékonyságának fenntartása szempontjából, mivel minimalizálja az üzemeltetési leállásokat és a termelési idő veszteségét. [9]

A S7-1500 OLC Web szerver könnyen telepíthető és konfigurálható a Siemens TIA Portal környezetben. Ez az egyszerűség hozzájárul a rendszer gyors és hatékony bevezetéséhez, minimalizálva a beavatkozásokat és a leállási időt. [9]

Biztonsági szempontból a S7-1500 OLC Web server számos beépített biztonsági intézkedést tartalmaz, amelyek védelmet nyújtanak a jogosulatlan hozzáférés és a kiberbiztonsági fenyegetések ellen. Ez biztosítja a vezérlő és a hozzá kapcsolódó rendszerek megbízhatóságát és biztonságát. [9]Ezen összességében a Siemens S7-1500 OLC Web server jelentős lépést jelent az ipari automatizáció terén, lehetővé téve a rugalmas és hatékony távkezelést és konfigurációt. A felhasználók könnyedén kihasználhatják a távoli elérés és az online változtatások előnyeit, miközben minimalizálják a leállási időt és maximalizálják a gyártási folyamatok hatékonyságát. [9]

## 5.6. TIA PORTAL

A Siemens Totally Integrated Automation (TIA) Portal kiemelkedő szoftveres fejlesztői környezetként emelkedik ki az ipari automatizáció terén. Ez a komplex szoftveres megoldás egyetlen, egységes felhasználói felületen keresztül integrálja az ipari automatizációs rendszerek tervezéséhez, programozásához, beüzemeléséhez és karbantartásához szükséges eszközöket és funkciókat.



15. ábra A TIA Portal képe Forrás: TIA Portal

A TIA Portal tervezési környezetének egyik fő előnye az egységes fejlesztői környezet. A felhasználók könnyen navigálhatnak a különféle automatizációs területeken, például PLC programozásban, HMI tervezésben és meghajtókezelésben, mindezt egyetlen, felhasználóbarát grafikus interfész segítségével.

A PLC programozás terén a TIA Portal támogatja a különböző Siemens PLC sorozatokat, mint például a SIMATIC S7-1200 és S7-1500. Ez a funkcionalitás lehetővé teszi a programozók számára a hatékony és testreszabott programok kialakítását, amelyek kiemelkedő teljesítményt nyújtanak az ipari automatizációs folyamatokban.



Az ember-gép interfész (HMI) tervezése is a TIA Portal egyik integrált eleme. Az eszköz segítségével a felhasználók könnyedén létrehozhatnak és testreszabhatnak vizualizációs felületeket, optimalizálva a felhasználói élményt és az operációt.

A TIA Portal az integrált meghajtókezelés révén támogatja a Siemens SINAMICS meghajtókat, így a felhasználók könnyen konfigurálhatják és kezelhetik a meghajtókat az egységes környezetben.

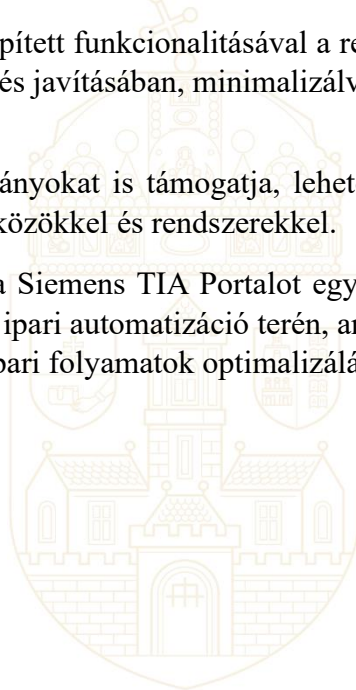
A rendszer további előnye az automatikus hálózati konfiguráció, ami jelentősen megkönnyíti a hálózati eszközök felismerését és konfigurálását a rendszer beüzemelése során.

A TIA Portal kifejezetten támogatja a digitalizációt az ipari környezetekben. A szoftver lehetővé teszi az ipari folyamatok teljes életciklusának optimalizálását, ami hozzájárul az üzemek hatékonyságához és a gyártási folyamatok rugalmasságához.

A diagnosztikai eszközök beépített funkcionalitásával a rendszer segíti az üzemeltetőket a hibák gyors azonosításában és javításában, minimalizálva ezzel az üzemeltetési leállási időt.

A TIA Portal a nyitott szabványokat is támogatja, lehetővé téve a könnyű integrációt harmadik féltől származó eszközökkel és rendszerekkel.

Mindezek együttesen teszik a Siemens TIA Portalot egy kivételesen erőteljes és teljes körű fejlesztői környezetté az ipari automatizáció terén, amely támogatja a vállalatokat a digitális átalakulásban és az ipari folyamatok optimalizálásában.



## 6. IPARI PC ÉS A TM-MFP GAZDASÁGI ÉS TECHNIKA ÖSSZEHAISONLÍTÁSA



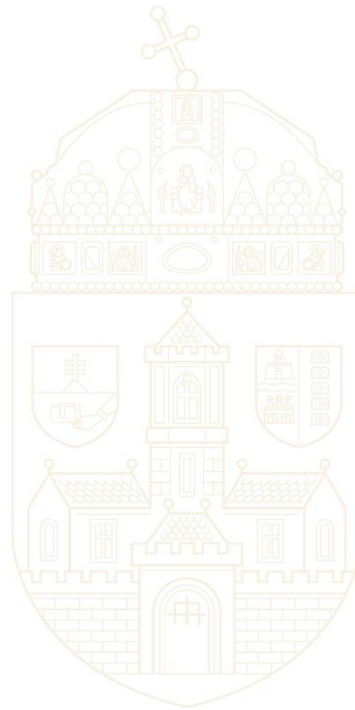
16. ábra A három vizsgált eszköz összehasonlítása Forrás: Saját ábra

Az összehasonlító elemzés alapján a TM MFP egy multifunkcionális platform, amely különböző alkalmazásokhoz használható az automatizálási mérnöki területen, ötvözi az Edge Computing és a felhőalapú megoldások előnyeit, kifejezetten az ipari alkalmazásokhoz, különös tekintettel a Siemens SIMATIC S7-1500 termékcsaláddal való integrációra. A TM MFP rugalmas, és számos alkalmazási területen használható a teljesítmény és a sokoldalúság révén.



Az ipari PC és az ipari okos kamera általánosságban is használhatók ipari alkalmazásokban, de az alkalmazás specifikus követelményektől függ, hogy melyik eszköz a legmegfelelőbb. A Siemens ipari PC széles körű csatlakoztathatóságot és általánosabb alkalmazhatóságot kínál, míg az ipari okos kamera elsősorban a képfeldolgozás és a vizuális felismerés területén kiemelkedik.

Az ipari okos kamerák nemcsak a képfeldolgozásban és a vizuális felismerésben, hanem a teljesítményükben is kiemelkednek, hogy hatékonyan megfeleljenek az ipari alkalmazások specifikus követelményeinek. Az összehasonlításban szem előtt kell tartani ezeket a szempontokat az optimális választás kiválasztásakor a konkrét alkalmazási igényekhez.





## 7. MEGVALÓSÍTÁS

### 7.1. TM-MFP FELADATA

A TM-MFP-n történik a kép beolvasása, átalakítása és feldolgozása. Az eszköz Display partjára kötött monitoron az átalakítás előtti és utáni kép, valamint a detektált élt megjelenítő kép is látszik a mért értékek mellett. Ez a megjelenítő az esetleges gyors hibakeresésnél nyújt segítséget

#### 7.1.1. Python

A Python programozási nyelv és a Linux operációs rendszer szoros kapcsolatban állnak, és ennek köszönhetően kiváló együttműködést biztosítanak számos területen. Elsősorban a Python nyelv beépített támogatása a Linuxhoz teszi rendkívül könnyen hozzáférhetővé. A Python interpreter előre telepítve van a legtöbb Linux disztribúcióban, így a fejlesztők gyorsan és egyszerűen kezdhetik meg a Python kódok írását és végrehajtását Linux rendszeren.

Mind a Python, mind a Linux rendszerek elkötelezettek a nyílt forráskódú elvek mellett, és élénk fejlesztői közösségek támogatják őket. A közösségi támogatás révén mindkét technológia folyamatosan fejlődik, és a fejlesztők számára számos hasznos könyvtár, modul és eszköz áll rendelkezésre.

A Python kiválóan alkalmazható shell scriptingre és automatizálásra a Linux rendszeren belül. A könnyen olvasható és érthető szintaxisa, valamint a dinamikus jellege segíti a fejlesztőket a hatékony scriptek írásában, amelyek elősegítik a rendszeradminisztrációt, a folyamatok automatizálását és a rendszerfelügyelet terén.

Az egyik kiemelkedő előnye a Pythonnak az, hogy platformfüggetlen, tehát ugyanazt a kódot könnyedén futtathatjuk különböző operációs rendszereken, beleértve a Linuxot. Ennek köszönhetően a fejlesztők egyszerűen áthelyezhetik kódjaikat különböző környezetek között.

A Python és Linux együttműködése tehát sokoldalú és hatékony, és számos területen alkalmazzák, beleértve a webfejlesztést, adatelemzést, mesterséges intelligenciát és gépi tanulást. E kombináció eredményeként a fejlesztők és rendszergazdák hatékonyan és eredményesen dolgozhatnak a fejlesztési, tesztelési és rendszermenedzsment folyamatok során, amiért ez a kombináció népszerű választás a technológiai közösségben. [11]

### 7.1.2. A modulok importálása.

A kódot a modulok importálásával kezdjük.

```
import cv2
import numpy as np
import datetime
import time
```

17. ábra Az OpenCV, NumPy, datetime és time modulok importálása. Forrás: Saját kód

### 7.1.3. A globális változó deklarálása.

Ezeket a változókat több függvényből is kell tudni olvasni ezért lettek globálisként meghatározva.

```
# Élsimított, szürkeárnyaltos és élszűrt képek deklarálása az egész kódban
blurred = None
gray = None
edges = None
szorzo = 1.5
```

18. ábra A globális változók Forrás: Saját kód

### 7.1.4. A „find\_circles” függvény

Ez a függvény megkeresi a képen található köröket. Először a képet szürkeárnyaltossá alakítja, majd élsimítást és élszűrést hajt végre. Az élek megtalálásához a Canny élszűrőt használja. Ezután megtalálja a kontúrokat, kiszámolja a kontúrok minimális befoglaló körét, és kiválasztja a külső és belső kört a távolság alapján a kép középpontjától. Végül visszatér a külső és belső körrel.

```
def find_circles(image, min_radius):
    global blurred, gray, edges
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    blurred = cv2.GaussianBlur(gray, (13, 13), 0)
    edges = cv2.Canny(blurred, 80, 200)

    contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    outer_circle = None
    inner_circle = None

    # Kép középpontja
    image_center = (image.shape[1] // 2, image.shape[0] // 2)

    for contour in contours:
        (x, y), radius = cv2.minEnclosingCircle(contour)
        center = (int(x), int(y))
        radius = int(radius)

        if radius >= min_radius:
            # Kör középpontja
            circle_center = center

            # Távolság a kép középpontjától
            distance_to_image_center = np.sqrt((circle_center[0] - image_center[0]) ** 2 +
                                                (circle_center[1] - image_center[1]) ** 2)

            # Ha az outer_circle még nincs beállítva, vagy a jelenlegi kör közelebb van a kép középpontjához,
            # akkor ez az outer_circle.
            if outer_circle is None or distance_to_image_center < np.sqrt((outer_circle[0][0] - image_center[0]) ** 2 +
                                                                           (outer_circle[0][1] - image_center[1]) ** 2):
                outer_circle = (center, radius)
            # Ellenkező esetben ez a belső kör.
            else:
                inner_circle = (center, radius)

    return outer_circle, inner_circle
```

19. ábra A „find\_circles” függvény Forrás: Saját kód

### 7.1.5. A „measure\_circle\_diameters” függvény

Ennek a függvénynek a célja egy adott kör átmérőjének mérésére szolgáló információ megjelenítése a képen és a képernyőn. A kör kétvonalú kijelzése, valamint az átmérő értékének kiírása a képre.

```
def measure_circle_diameters(image, circle, label, offset_y):  
    # Rajzolj egy kört a bemeneti képre a megadott kör paraméterek alapján  
    cv2.circle(image, circle[0], circle[1], (0, 255, 0), 2)  
  
    # Számold ki a átmérőt milliméterben a megadott konverziós tényezővel (szorzo)  
    diameter_mm = circle[1] * szorzo * 0.264583  
  
    # Jelenítsd meg a átmérő információt szöveggént a képen  
    cv2.putText(image, f"{label} Atmero: {diameter_mm:.2f} mm", (10, 30 + offset_y),  
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)
```

20. ábra A „measure\_circle\_diameters” függvény Forrás: Saját kód

### 7.1.6. A „write\_measurement\_to\_file” függvény

A függvény kiírja a mérési eredményeket egy naplófájlba, ideértve a külső és belső átmérőt is, valamint egy időbélyegzőt a mérés időpontjáról. A mért értékek milliméterben vannak kifejezve.

```
def write_measurement_to_file(log_file_path, log_file_path_2, outer_circle, inner_circle):  
    # Aktuális időbélyeg készítése  
    timestamp = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S.%f")[:-3]  
  
    # Kimeneti fájl megnyitása írásra, "w" mód  
    with open(log_file_path, "w") as log_file:  
        # Belső és külső atmero milliméterben  
        inner_diameter_mm = inner_circle[1] * szorzo * 0.264583  
        outer_diameter_mm = outer_circle[1] * szorzo * 0.264583  
  
        # Belső és külső atmero információk írása a log fájlba  
        log_file.write(f"{timestamp} - Külső Átmérő: {outer_diameter_mm:.2f} mm\n")  
        log_file.write(f"{timestamp} - Belső Átmérő: {inner_diameter_mm:.2f} mm\n")  
  
    # Második kimeneti fájl megnyitása hozzáfűzésre, "a" mód  
    with open(log_file_path_2, "a") as log_file:  
        # Belső és külső atmero milliméterben, 1.4-es szorzóval  
        inner_diameter_mm = inner_circle[1] * 1.4 * 0.264583  
        outer_diameter_mm = outer_circle[1] * 1.4 * 0.264583  
  
        # Belső és külső atmero információk írása a második log fájlba  
        log_file.write(f"{timestamp} - Külső Átmérő: {outer_diameter_mm:.2f} mm\n")  
        log_file.write(f"{timestamp} - Belső Átmérő: {inner_diameter_mm:.2f} mm\n")  
  
    # Az időbélyeg visszaadása  
    return timestamp
```

21. ábra A „write\_measurement\_to\_file” függvény Forrás: Saját kód

### 7.1.7. A „main” függvény

- Beállítja a kamera felbontását és inicializálja a video forrást. A „cap” változó egy cv2.VideoCapture objektum, amely a video forrás (kamera) kezelését biztosítja. Itt a 0 a video forrás indexét jelenti, és ebben az esetben a beépített kamerára utal. Ha több kamera van csatlakoztatva, akkor más értéket is megadhat, például 1, 2 stb.

```
# Kamera felbontásának beállítása
resolution = (640, 480)

# Videoforrás inicializálása (0 a beépített kamera, de megadható más forrás is)
cap = cv2.VideoCapture(0)
cap.set(3, resolution[0]) # 3-as kód: szélesség beállítása
cap.set(4, resolution[1]) # 4-es kód: magasság beállítása
```

22. ábra A "main" függvény kódrészlet: video forrás inicializálása és a kamera beállítása  
Forrás: Saját kód

- Folyamatosan olvassa be a képkockákat a kamerából és írja ki a képernyőre. A „ret” változó „True” értéket kap, ha a képkocka sikeresen beolvasásra került. A „frame” változó tartalmazza a beolvasott képkockát. Az olvasás után a frame\_2 változó is ugyanazt a képkockát tartalmazza, így két változóban kerül eltárolásra az aktuális képkocka, amit a cv2.imshow("Kamera kep ", frame\_2) sorban megjelenít, a frame változó tartalmát később használja fel.

```
# Képkockák olvasása a kamerából
ret, frame = cap.read()
ret, frame_2 = cap.read()

# Kép megjelenítése a kamerából
cv2.imshow("Kamera kep ", frame_2)
```

23. ábra A "main" függvény kódrészlet: képkockák beolvasása a kamerából és kiírtatása a képernyőre Forrás: Saját kód

- Meghívja a már korábban bemutatott „find\_circles” függvényt a képkocka elemzésére.

A find\_circles függvénynek a következő paramétereket kell megkapnia:

- image: Ez a paraméter a képet jelenti, amelyen a függvény körkeresést végez. A frame változó tartalmazza az aktuális képkockát a kameráról, és ezt kell átadni a függvénynek. Ez jelen esetben a „frame” változó.
- min\_radius: Ez a minimális sugár értékét jelenti, amelyet a függvény használ a keresés során.

```
# Külső és belső körök keresése a képen
outer_circle, inner_circle = find_circles(frame, min_radius)
```

24. ábra A "main" függvény kódrészlet: kör keresése a képen Forrás: Saját kód

- Ha talált külső és belső kört, meghívja a „measure\_circle\_diameters” függvényt a képen és a képernyőn való megjelenítéshez.

A „measure\_circle\_diameters” függvény a következő paramétereket várja:

- **image:** Ez a paraméter a képet jelöli, amin módosításokat végez a függvény. A „frame” változó tartalmazza az aktuális képkockát a kameráról, és ez kerül átadásra a függvénynek.
- **circle:** Ez a paraméter tartalmazza a körciklus jellemzőit, mint például a középpont és a sugár. A `find_circles` függvény által visszaadott „outer\_circle” változót használod itt.
- **label:** Ez a szöveg a körre kerül kiírásra a képre. A példában "Belso", azaz belső kör.
- **offset\_y:** Ez egy eltolás, amellyel a szöveg pozícióját a képen módosítod. A példában 0 értéket adtál meg, tehát a szöveg a kör közepén jelenik meg.

Tehát a „`measure_circle_diameters(frame, outer_circle, "Belso", 0)`” hívás azt jelenti, hogy a belső kör adatait (`outer_circle`) vizualizálódik a `frame` képkockán, ahol a kör közepén a "Belso" szöveg jelenik meg. A 0 értékkel eltolod a szöveget, tehát a kör közepén helyezkedik el.

```
# Ha találtunk külső és belső kört
if outer_circle and inner_circle:
    # Átmérők mérésének és kiírásának függvényének meghívása
    measure_circle_diameters(frame, outer_circle, "Belso", 0)
    measure_circle_diameters(frame, inner_circle, "Kulso", 20)
```

25. ábra A "main" függvény kódrészlet: A mérés és a kiírás függvényének meghívása

Forrás: Saját kód

- Kiírja a mért eredményeket a naplófájlba és a képre.

A „`write_measurement_to_file`” függvénynek a következő paramétereket kell megkapnia:

- **log\_file\_path:** Az első naplófájl elérési útvonala, ahol a külső és belső átmérőket tárolja. Ez az elérési útvonal a kódban előre meg lett adva.
- **log\_file\_path\_2:** A második naplófájl elérési útvonala, ahova más változókkal vagy beállításokkal a mért értékeket menti. Ez is előre meg lett határozva a kódban.
- **outer\_circle:** A külső kör adatait tartalmazó tuple, amely a következőket tartalmazza: ((középpont\_x, középpont\_y), sugár).
- **inner\_circle:** A belső kör adatait tartalmazó tuple, ugyanúgy, mint az `outer_circle`.

A „`cv2.putText`” függvénynek a következő paramétereket kell megkapnia:

- **frame:** Az eredeti képkocka, amelyre kiírja az időbélyegzőt.
- **f"Merés időpontja: {timestamp}"**: Az időbélyegző szövege, amely a mért időpontot tartalmazza. A `timestamp` változó tartalmazza a `write_measurement_to_file` függvény által visszaadott időbélyegzőt.
- **(10, 70)**: Az időbélyegző helye a képen. Itt (10, 70) koordinátákat adtál meg, tehát az időbélyegző az (10, 70) pozíciójára kerül a képen.

- `cv2.FONT_HERSHEY_SIMPLEX`: A betűtípus, itt a `FONT_HERSHEY_SIMPLEX`.
- `0.5`: A betűméret.
- `(0, 255, 0)`: A szövegszín, itt zöld (RGB formátumban).
- `2`: A betűvastagság.

```
# Mérés eredmények naplózása és időbélyegző kiírása a képre
timestamp = write_measurement_to_file(log_file_path, log_file_path_2, outer_circle, inner_circle)
cv2.putText(frame, f"Meres időpontja: {timestamp}", (10, 70), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
```

26. ábra A "main" függvény kódrészlet: mérési eredmények naplózása Forrás: Saját kód

- A képkockákat megjeleníti külön ablakban az eredeti, élsimított, szürkeárnyaltos és élszűrt formában.

A „`cv2.imshow`” függvénynek a következő paramétereket kell megkapnia:

- `window_name`: Ez a paraméter a képernyőn megjelenő ablak nevét határozza meg. Minden megnyitott ablaknak egyedi nevet kell adni. Ha ugyanazt a nevet használjuk egy másik ablak megnyitására, az új ablak felülírja a régit.
- `image`: Ez a paraméter a megjelenítendő képet vagy képkockát jelenti. A kép lehet színes (BGR formátumban) vagy szürkeárnyaltos. Ha a kép mérete nagy, akkor a kép méretezése is automatikusan megtörténik, hogy elférjen az ablakban.

```
# Képek megjelenítése az eredeti, élsimított, szürkeárnyaltos és élszűrt formában
cv2.imshow("Eredeti Kep", frame)
cv2.imshow("Elsimított Kep", blurred)
cv2.imshow("Szürkeárnyaltos Kep", gray)
cv2.imshow("Elkep", edges)
```

27. ábra A "main" függvény kódrészlet: képek megjelenítése a képernyőn Forrás: Saját kód

- Kilép a programból, ha a felhasználó megnyomja a "q" billentyűt.

```
# Kilépés a ciklusból, ha a felhasználó megnyomta a 'q' billentyűt
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
```

28. ábra A "main" függvény kódrészlet: a programból való kilépés Forrás: Saját kód

## 7.2.BLANCPAIN BUS FELADATA

Az adatok küldése a PLC és a linux között backpain buszon történik. A backpain buson történő adat küldéshez használt python alkalmazás a TM-MFP modulon fut.

### 7.2.1. Az import utasítások:

- sys: A Python rendszermódulja, szükséges a rendszerrel kapcsolatos funkciókhoz.
- time: Időzítési műveletekhez használt modul.
- dbus: D-Bus kommunikációhoz szükséges modul.
- dbus\_client: Valószínűleg a saját modul, amely egy D-Bus klienst implementál.
- struct: Az struct modul az adatok csomagolására és kicsomagolására szolgál, például a bájtfolyamok és egész számok közötti átalakításokhoz.
- re: A reguláris kifejezések (regular expressions) módulja, ami a szövegek mintázásához és kereséséhez használt.

```
import sys
import time
import dbus
import dbus_client
import struct
import re
```

29. ábra Az import utasítások Forrás: Saját kód

### 7.2.2. A „clean\_and\_convert\_to\_integers” függvény:

A függvény célja, hogy egy fájlból beolvasott sorokat tisztítsa és átalakítsa egész számokká. Az eredményt egy új fájlba írja (output\_integers.txt) és visszaadja az egész számokat egy listában. A tisztítás során eltávolítja a kötőjelet, a kettőspontot, a pontot és az abc kis- és nagybetűit, és a számokat különválasztja. Két vagy annál hosszabb számok esetén a számjegyeket két számjegyű részekre bontja. A konvertálás során figyel a ValueError-re, és a konvertálhatatlan értékeket figyelmen kívül hagyja. A fájl nem találása esetén vagy ha a konvertálás nem sikerül, hibaüzeneteket ír ki, majd kilép a program.

```
def clean_and_convert_to_integers(file_path, output_file_path="output_integers.txt"):
    try:
        with open(file_path, 'r') as file:
            # Fájl sorainak feldolgozása
            lines = file.readlines()

            # Az összegyűjtött számokat int-típusúvá konvertáljuk
            integers = []
            for line in lines:
                # Karakterek cseréje és tisztítás
                cleaned_line = re.sub(r'[:-:~a-zA-Z]', ' ', line)
                # Szétválasztjuk a számokat, majd azokat átalakítjuk int-típusúvá
                for num in cleaned_line.split():
                    if len(num) >= 3:
                        # Ha 3 vagy több számjegyű, két számjegyű részekre szedjük szét
                        integers.extend([int(num[i:i+2]) for i in range(0, len(num), 2)])
                    else:
                        try:
                            # Próbáljuk meg átalakítani az értéket int-té
                            integers.append(int(num))
                        except ValueError:
                            print(f"Warning: Unable to convert '{num}' to integer. Skipping...")

            # Kiírjuk az int értékeket egy fájlba
            with open(output_file_path, 'w') as output_file:
                for integer_value in integers:
                    output_file.write(f"{integer_value} ")

            return integers

    except ValueError as e:
        print(f"Error: Failed to convert data to integer. {e}")
        sys.exit(1)
    except FileNotFoundError:
        print(f"Error: File '{file_path}' not found!")
        sys.exit(1)
```

30. ábra A „clean\_and\_convert\_to\_integers” függvény Forrás: Saját kód

### 7.2.3. A „main” függvény:

A végtelen ciklusban egy D-Bus klienszt hoz létre és kapcsolódik hozzá. Ha a kapcsolat létrejön, a program fő ciklusában marad. Ebben a ciklusban a „clean\_and\_convert\_to\_integers” függvény segítségével számokat olvas be és elküldi a D-Bus kliens segítségével. A D-Bus kliens a „write\_all” módszerrel írja ki az összes bájtot, majd várakozik egy rövid időre (0.02 másodpercre) az adatok feldolgozása között. Ha a PLC IO adatok érvényesek, az adatokat elküldi, ellenkező esetben hibaüzenetet ír ki, és visszatér az újra próbálkozás külső ciklusába.

```
def main():
    while True:
        client = dbus_client.DBusClient()

        try:
            client.connect()
        except dbus.DBusException:
            print("Error: D-Bus client is not connected!")
            time.sleep(5)
            continue # Újrapróbálkozás

        print("Connection successful!") # Sikeres kapcsolat esetén kiírjuk az üzenetet

        while True:
            # Fájlból származó int értékek elküldése
            int_values = clean_and_convert_to_integers("send_data.txt")

            io_data_valid = client.get_io_data_state()
            if io_data_valid is True:
                # Minden int_value érték elküldése egyetlen blokkban
                byte_values = [(int_value >> (i * 8)) & 0xFF for int_value in int_values for i in range(1)]
                client.write_all(byte_values)
                time.sleep(0.02)
            else:
                print("PLC IO data are not valid! PLC is not configured or is not in RUN state. Retrying in 5 seconds...")
                time.sleep(5)
                break # Visszatérés az újrapróbálkozás külső ciklusába
```

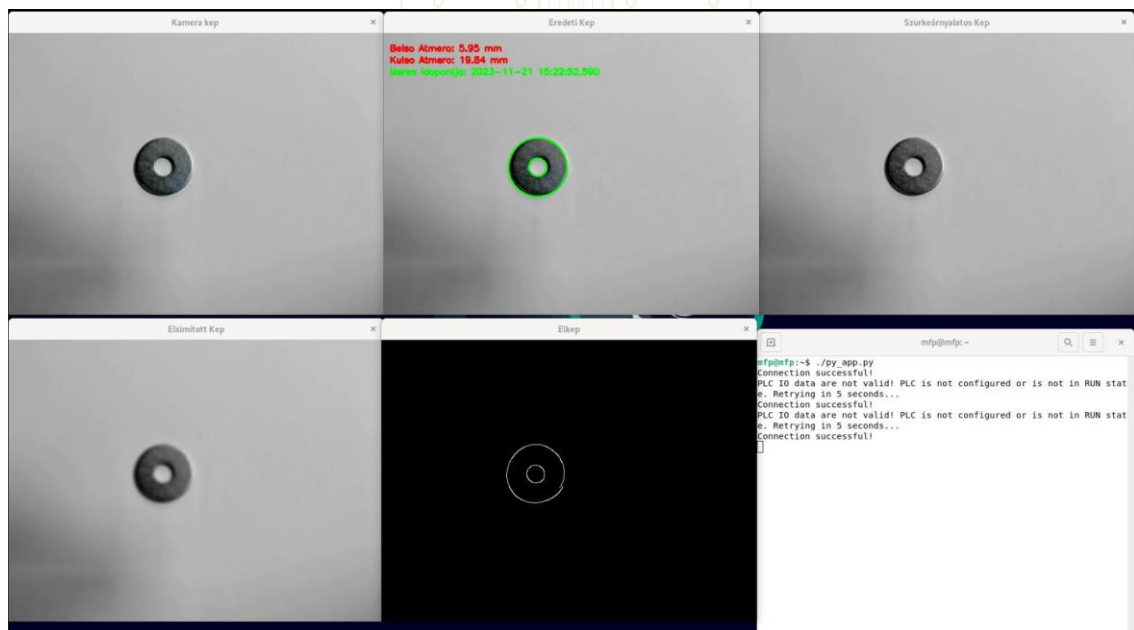
31. ábra A „main” függvény Forrás: Saját kód

#### 7.2.4. A „name == \"main\"” függvény:

Ellenőrzi, hogy a fájl közvetlenül fut-e, és ha igen, meghívja a main függvényt. Ha a fájl importálva van, akkor nem hajtja végre a kódot. A sys.exit(main()) meghívja a main függvényt, majd a visszatérési értékét használja a program leállításához. Ha a main visszatérési értéke nem 0, akkor a program hibakóddal zárul.

```
if __name__ == "__main__":
    sys.exit(main())
```

32. ábra A „name == \"main\"” függvény Forrás: Saját kód



33. ábra A TM-MFP kijelzőjén látható kép Forrás: Saját kép

A „Kamera kép” folyamatosan frissül, arra szolgál, hogy ellenőrizni lehessen, az eszköz nem fagyott le, illetve mit lát. Az „Eredeti kép” csak akkor frissül, ha új mérés történik, ezen az ablakon lehet látni a mért eredményeket és a mérés idejét is. A „Szürkearnyalatos kép” és az „Elsimitott kép” a képalakítási folyamat bemutatása és ellenőrzése miatt kerül megjelenítésre. Az „Elkép” a rendszer által detektált vonalakat jeleníti meg. A parancssorban a backplane busz programjának az üzenetei látszanak.

### 7.3.A PLC FELADATA

A PLC fogadja az byte típusú adatokat, amit 22 Tag-ben ezt Int típusúvá alakítja ["Result\_DB"."0":= BYTE\_TO\_INT("b\_0")] és tárolja egy Data Block-ban.

| Default tag table |      |           |         |                          |                                     |                                     |                                     |               |
|-------------------|------|-----------|---------|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|---------------|
|                   | Name | Data type | Address | Retain                   | Acces...                            | Writa...                            | Visibl...                           | Monitor value |
| 1                 | b_0  | Byte      | %IB0    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#14         |
| 2                 | b_1  | Byte      | %IB1    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#17         |
| 3                 | b_2  | Byte      | %IB2    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#0A         |
| 4                 | b_3  | Byte      | %IB3    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#05         |
| 5                 | b_4  | Byte      | %IB4    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#01         |
| 6                 | b_5  | Byte      | %IB5    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#37         |
| 7                 | b_6  | Byte      | %IB6    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#25         |
| 8                 | b_7  | Byte      | %IB7    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#38         |
| 9                 | b_8  | Byte      | %IB8    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#09         |
| 10                | b_9  | Byte      | %IB9    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#06         |
| 11                | b_10 | Byte      | %IB10   | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#4B         |
| 12                | b_11 | Byte      | %IB11   | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#14         |
| 13                | b_12 | Byte      | %IB12   | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#17         |
| 14                | b_13 | Byte      | %IB13   | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#0A         |
| 15                | b_14 | Byte      | %IB14   | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#05         |
| 16                | b_15 | Byte      | %IB15   | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#01         |
| 17                | b_16 | Byte      | %IB16   | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#37         |
| 18                | b_17 | Byte      | %IB17   | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#25         |
| 19                | b_18 | Byte      | %IB18   | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#38         |
| 20                | b_19 | Byte      | %IB19   | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#09         |
| 21                | b_20 | Byte      | %IB20   | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#13         |
| 22                | b_21 | Byte      | %IB21   | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#54         |
| 23                | b_22 | Byte      | %IB22   | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 16#00         |

34. ábra A Tag-be érkező adatok Forrás: TIA Portal

Az idő, amit a Linux oldali applikáció átküld a könnyebb kezelhetőség érdekében DTL adat típussá alakítja

```

1 "Result_DB"."time".YEAR := "Result_DB"."0" * 100 + "Result_DB"."1";
2 "Result_DB"."time".MONTH := "Result_DB"."2";
3 "Result_DB"."time".DAY := "Result_DB"."3";
4 "Result_DB"."time".HOUR := "Result_DB"."4";
5 "Result_DB"."time".MINUTE := "Result_DB"."5";
6 "Result_DB"."time".SECOND := "Result_DB"."6";
7 "Result_DB"."time".NANOSECOND := "Result_DB"."7";

```

35. ábra Az idő átalakítástát végző kód Forrás: Saját kód

| Result_DB |            |           |                          |                      |
|-----------|------------|-----------|--------------------------|----------------------|
|           | Name       | Data type | Start value              | Monitor value        |
| 1         | Static     |           |                          |                      |
| 2         | time       | DTL       | DTL# 1970-01-01-00:00:00 | DTL# 2023-10-05-0... |
| 3         | YEAR       | UInt      | 1970                     | 2023                 |
| 4         | MONTH      | USInt     | 1                        | 10                   |
| 5         | DAY        | USInt     | 1                        | 5                    |
| 6         | WEEKDAY    | USInt     | 5                        | 5                    |
| 7         | HOUR       | USInt     | 0                        | 1                    |
| 8         | MINUTE     | USInt     | 0                        | 55                   |
| 9         | SECOND     | USInt     | 0                        | 37                   |
| 10        | NANOSECOND | UDInt     | 0                        | 56                   |
| 11        | 0          | Int       | 0                        | 20                   |
| 12        | 1          | Int       | 0                        | 23                   |
| 13        | 2          | Int       | 0                        | 10                   |
| 14        | 3          | Int       | 0                        | 5                    |
| 15        | 4          | Int       | 0                        | 1                    |
| 16        | 5          | Int       | 0                        | 55                   |
| 17        | 6          | Int       | 0                        | 37                   |
| 18        | 7          | Int       | 0                        | 56                   |
| 19        | 8          | Int       | 0                        | 9                    |
| 20        | 9          | Int       | 0                        | 6                    |
| 21        | 10         | Int       | 0                        | 75                   |

36. ábra A RESULT\_DB tartalma (részlet) Forrás: TIA Portal

#### 7.4.HMI FELADATA

A HMI kizárólag a mért értékek és a mérés idejének a kezelő fele történő megjelenítését végzi, nincs bevitel ezért jelszóval sincs védve.



37. ábra A HMI-n megjelenő kép Forrás: TIA Portal

## 8. ÖSSZEGZÉS

A szakdolgozat célja, hogy a Siemens S7 1500 PLC és a hozzá tartozó TM MFP modul segítségével gyártás során minőség ellenőrzés megvalósítása.

Megéri ezt a rendszert használni, el tudja látni a feladatot?

Milyen egyéb eszközökre van szükség?

Mekkora gondot és bizonytalanságot jelent az állandóan változó fényviszony?

Milyen programozási nyelveket lehet használni?

Megéri a rendszert használni hiszen árához képest jó teljesítményt nyújt. Mivel az ipari PC-vel szemben a TM MFP minden esetben a PLC mellett található így a távolságból kifolyó meghibásodások kiküszöbölhetőek, illetve adatátviteli idő csökken, eszközhiba esetén mivel minden PLC mellé kell tenni egy ilyen eszközt, aminél képfeldolgozást akarunk végezni ezért kisebb idő és anyagi veszteséggel lehet pótolni a meghibásodott eszközt. A rendszerről könnyen lehet biztonsági mentést készíteni egy pendrive-ra, így a rendszer könnyen, rövid idő alatt átállítható más alkatrész esetén.

Fontos szempont a kamera kiválasztása. Az eszközhöz USB-n vagy Etherneten keresztül lehet az ilyen eszközöket csatlakoztatni. A legjobb kamerák erre a célra, amik telecentrikus optikával rendelkeznek. A rendszerél ki kell zárni, hogy a külső fények megzavarja a mérést, ezért szükség van megvilágításra, amit a tárgy négy oldaláról kell elhelyezni, hogy ne legyen árnyék.

A TM MFP Debian Linux alapú industrial OS-t használ, amelyet a Siemens fejleszt. Az eszköz a C++ és a Python programozási nyelveket támogatja.

Python egy platformfüggetlen nyelv, így a Debian Linuxon kívül más operációs rendszereken is használható. Előre telepítve van a Python interpreter. Ez azt jelenti, hogy a Python alkalmazásokat könnyen futtathatod Debian rendszeren anélkül, hogy külön telepítened kellene a Python környezetet. Python egy nyílt forráskódú nyelv, és rendelkezik egy aktív fejlesztői közösséggel.

A rendszer nem rendelkezik beépített akkumulátorral ezért minden indításkor biztosítani kell az időszinkront a TM MFP-nek. Ezt NTP szerver alkalmazásával lehet megtenni. A log-ok szempontjából elengedhetetlen, hogy a PLC és a TM MFP oldalon megegyezzen az idő.

Az eszköz folyamatos fejlesztés alatt áll, mivel 2023 tavaszán került forgalomba. A modul jelenleg csak NTP szerverrel történő idő szinkronnal rendelkezi, hasznos lenne a bbus-on keresztüli időszinkron a PLC-vel. Nem rendelkezik alarmokkal sem az SSD kapacitását, sem a rendszer hőmérsékletét tekintve.



## 9. SUMMARY

The aim of the thesis is to implement quality control during manufacturing using the Siemens S7 1500 PLC and the associated TM MFP module.

Is it worth using this system? Can it fulfill the task?

What other tools are needed?

How much trouble and uncertainty does the constantly changing lighting conditions pose?

What programming languages can be used?

It is worthwhile to use the system since it offers good performance in relation to its cost. Unlike industrial PCs, the TM MFP is always located next to the PLC, so malfunctions can be eliminated remotely, reducing data transfer time. In case of device failure, since each PLC requires such a device for image processing, a malfunctioning device can be replaced with minimal time and financial loss. Backing up the system on a USB drive is easy, allowing for quick reconfiguration in the event of component replacement.

The choice of the camera is a crucial factor. Devices can be connected to the TM MFP via USB or Ethernet. The best cameras for this purpose are those with telecentric optics. External light interference must be eliminated from the system, requiring illumination from all four sides of the object to avoid shadows.

The TM MFP uses a Debian Linux-based industrial OS developed by Siemens. The device supports C++ and Python programming languages.

Python is a platform-independent language, so it can be used on operating systems other than Debian Linux. The Python interpreter comes pre-installed, making it easy to run Python applications on Debian without the need for a separate installation. Python is an open-source language with an active developer community.

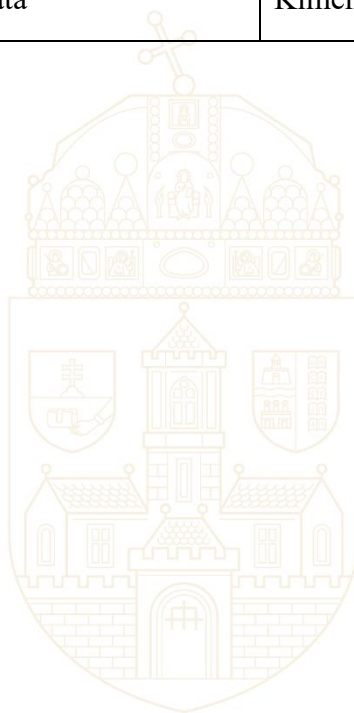
The system does not have a built-in battery, so time synchronization must be ensured for the TM MFP at every startup. This can be achieved using an NTP server application. Consistent time on both the PLC and TM MFP sides is crucial for logs.

The device is under continuous development since its release in spring 2023. The module currently only supports time synchronization with an NTP server, and it would be beneficial to have time synchronization via bbus with the PLC. It lacks alarms for both SSD capacity and system temperature.



## 10. RÖVIDÍTÉSEK

| Betűszó | Leírás                                        |                                             |
|---------|-----------------------------------------------|---------------------------------------------|
| PLC     | Programmable Logic Controller                 | Programozható Logikai Vezérlő               |
| OS      | Operating System                              | Operációs rendszer                          |
| TM MFP  | Technology Module<br>Multifunctional Platform | Technológiai Modul<br>Multifunkciós Felület |
| PC      | Personal Computer                             | Személyi Számítógép                         |
| HMI     | Human Mashine Interface                       | Ember Gép Összekötő                         |
| IO data | Input Output data                             | Kimeneti Bemeneti adat                      |





## 11. ÁBRAJEGYZÉK

|                                                                                                                            |    |
|----------------------------------------------------------------------------------------------------------------------------|----|
| 1. ábra Minőségi osztályba sorolás feltételei Saját ábra .....                                                             | 6  |
| 2. ábra Termékek besorolása Saját ábra .....                                                                               | 8  |
| 3. ábra A képfeldolgozás lépései Saját ábra .....                                                                          | 9  |
| 4. ábra Ipari okoskamera [13] .....                                                                                        | 10 |
| 5. ábra Az ipari kamera felépítése [14] .....                                                                              | 11 |
| 6. ábra Siemens ipari rack PC [15].....                                                                                    | 13 |
| 7. ábra Kondenzátor képe normál háttérmegvilágítás és kollimátor használatával [3] .....                                   | 15 |
| 8. ábra Néhány tipikus feladathoz a megfelelő megvilágítás [3] .....                                                       | 15 |
| 9. ábra Az irányítástechnikai piramis [5] .....                                                                            | 16 |
| 10. ábra SIMATIC S7-1500 PLC CPU-k [5] .....                                                                               | 17 |
| 11. ábra S7 1500 1518 MFP PLC és néhány modullal [5] .....                                                                 | 17 |
| 12. ábra TM-MFP modul [8] .....                                                                                            | 18 |
| 13. ábra Standard simatic HMI [16].....                                                                                    | 20 |
| 14. ábra A webszerver online felülete Forrás: TIA Portal .....                                                             | 21 |
| 15. ábra A TIA Portal képe Forrás: TIA Portal.....                                                                         | 22 |
| 16. ábra A három vizsgált eszköz összehasonlítása Forrás: Saját ábra .....                                                 | 24 |
| 17. ábra Az OpenCV, NumPy, datetime és time modulok importálása. Forrás: Saját kód.....                                    | 27 |
| 18. ábra A globális változók Forrás: Saját kód.....                                                                        | 27 |
| 19. ábra A „find_circles” függvény Forrás: Saját kód.....                                                                  | 27 |
| 20. ábra A „measure_circle_diameters” függvény Forrás: Saját kód.....                                                      | 28 |
| 21. ábra A „write_measurement_to_file” függvény Forrás: Saját kód.....                                                     | 28 |
| 22. ábra A "main" függvény kódrészlet: video forrás inicializálása és a kamera beállítása Forrás: Saját kód.....           | 29 |
| 23. ábra A "main" függvény kódrészlet: képkockák beolvasása a kamerából és kiírtatása a képernyőre Forrás: Saját kód ..... | 29 |
| 24. ábra A "main" függvény kódrészlet: kör keresése a képen Forrás: Saját kód .....                                        | 29 |
| 25. ábra A "main" függvény kódrészlet: A mérés és a kiírás függvényének meghívása Forrás: Saját kód.....                   | 30 |
| 26. ábra A "main" függvény kódrészlet: mérési eredmények naplózása Forrás: Saját kód.....                                  | 31 |
| 27. ábra A "main" függvény kódrészlet: képek megjelenítése a képernyőn Forrás: Saját kód ...                               | 31 |
| 28. ábra A "main" függvény kódrészlet: a programból való kilépés Forrás: Saját kód.....                                    | 31 |
| 29. ábra Az import utasítások Forrás: Saját kód.....                                                                       | 32 |
| 30. ábra A „clean_and_convert_to_integers” függvény Forrás: Saját kód.....                                                 | 33 |
| 31. ábra A „main” függvény Forrás: Saját kód.....                                                                          | 34 |
| 32. ábra A „name == "main"” függvény Forrás: Saját kód .....                                                               | 34 |
| 33. ábra A TM-MFP kijelzőjén látható kép Forrás: Saját kép .....                                                           | 34 |
| 34. ábra A Tag-be érkező adatok Forrás: TIA Portal.....                                                                    | 35 |
| 35. ábra Az idő átalakítást végző kód Forrás: Saját kód .....                                                              | 35 |
| 36. ábra A RESULT_DB tartalma (részlet) Forrás: TIA Portal.....                                                            | 36 |
| 37. ábra A HMI-n megjelenő kép Forrás: TIA Portal.....                                                                     | 36 |





- [8] SIMATIC HMI Communication for Windows-based Systems User's Guide Release 12/01 Forrás: [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKewiFxriwo-SCAxVWtaQKHYshByMQFnoECA0QAQ&url=https%3A%2F%2Fsupport.industry.siemens.com%2Fcs%2Fattachments%2F1405528%2Fcomb1V6\\_e.pdf&usg=AOvVaw22huYOFIEm6p8CP6WWLvWJ&opi=89978449](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKewiFxriwo-SCAxVWtaQKHYshByMQFnoECA0QAQ&url=https%3A%2F%2Fsupport.industry.siemens.com%2Fcs%2Fattachments%2F1405528%2Fcomb1V6_e.pdf&usg=AOvVaw22huYOFIEm6p8CP6WWLvWJ&opi=89978449)
- [9] Examples for the SIMATIC S7-1200 / S7-1500 Web Server V2.1, 08/2018 Forrás: [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKewiN9rHep-SCAxUVG-wKHxUEBhcQFnoECBoQAQ&url=https%3A%2F%2Fcache.industry.siemens.com%2Fdl%2Ffiles%2F496%2F68011496%2Fatt\\_959527%2Fv2%2F68011496\\_Examples\\_for\\_S7WebServer\\_DOC\\_v21\\_en.pdf&usg=AOvVaw3nk9CM7917EddM\\_tLVYzP2&opi=89978449](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKewiN9rHep-SCAxUVG-wKHxUEBhcQFnoECBoQAQ&url=https%3A%2F%2Fcache.industry.siemens.com%2Fdl%2Ffiles%2F496%2F68011496%2Fatt_959527%2Fv2%2F68011496_Examples_for_S7WebServer_DOC_v21_en.pdf&usg=AOvVaw3nk9CM7917EddM_tLVYzP2&opi=89978449)
- [10] SIMATIC STEP 7 and WinCC Engineering V18 System Manual 11/2022 Forrás: [support.industry.siemens.com/dl/files/056/109815056/att\\_1121875/v5/STEP\\_7\\_WinCC\\_V18\\_enUS\\_en-US.pdf](https://support.industry.siemens.com/dl/files/056/109815056/att_1121875/v5/STEP_7_WinCC_V18_enUS_en-US.pdf)
- [11] <https://docs.python.org/3.12/>
- [12] [https://www.cognex.com/library/media/press-release-media/2020/in-sight-d900-integrated-light-in-hand-left.jpg?sc\\_lang=en](https://www.cognex.com/library/media/press-release-media/2020/in-sight-d900-integrated-light-in-hand-left.jpg?sc_lang=en)
- [13] <https://www.ipsecu.com/articles/the-internal-structure-of-the-security-camera/>
- [14] <https://mall.industry.siemens.com/mall/fr/be/Catalog/Products/10398031>
- [15] <https://www.distrelec.hu/hu/ktp700-basic-hmi-panel-800-480-ip65-siemens-6av2123-2gb03-0ax0/p/30024699>