

VARGA ROLAND  
SZAKDOLGOZAT

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM  
GÉPÉSZMÉRNÖKI KAR  
MECHATRONIKA, OPTIKA ÉS GÉPÉSZETI INFORMATIKA TANSZÉK



SZAKDOLGOZATOK



BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM  
GÉPÉSZMÉRNÖKI KAR  
MECHATRONIKA, OPTIKA ÉS GÉPÉSZETI INFORMATIKA TANSZÉK

VARGA ROLAND  
SZAKDOLGOZAT

**Ember és robot kooperációjának demonstrálása  
Sakkozó iiwa robotkar segítségével**

*Demonstrating human-robot collaboration  
With chess-playing iiwa robotic arm*

Konzulens:

*Magyar László*  
tesztmérnök

Témavezető:

*Dr. Czmerk András*  
egyetemi adjunktus

Budapest, 2018

Ide kell befűzni az eredeti feladatkiírási lapot!

# NYILATKOZATOK

## *Beadhatósági nyilatkozat*

A jelen szakdolgozat az üzem által elvárt szakmai színvonalnak mind tartalmilag, mind formailag megfelel, beadható.

Kelt,

Az üzem részéről:

*üzemi konzulens*

## *Elfogadási nyilatkozat*

Ezen szakdolgozat a Budapesti Műszaki és Gazdaságtudományi Egyetem Gépészmérnöki Kara által a Diplomatervezési és Szakdolgozat feladatokra előírt valamennyi tartalmi és formai követelménynek, továbbá a feladatkiírásban előírtaknak maradéktalanul eleget tesz. E szakdolgozatot a nyilvános bírálatra és nyilvános előadásra alkalmasnak tartom.

A beadás időpontja:

*témavezető*

## *Nyilatkozat önálló munkáról*

Alulírott, Varga Roland (XZYX5L), a Budapesti Műszaki és Gazdaságtudományi Egyetem hallgatója, büntetőjogi és fegyelmi felelősségem tudatában kijelentem és sajátkezű aláírással igazolom, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, és dolgozatomban csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a hatályos előírásoknak megfelelően, a forrás megadásával megjelöltem.

Budapest, 2018 .....

*szigorló hallgató*



# TARTALOMJEGYZÉK

<b>1. Bevezetés</b>	<b>1</b>
1.1. Célkitűzés . . . . .	1
1.2. Áttekintés – még nem végleges . . . . .	1
<b>2. Irodalomkutatás</b>	<b>2</b>
2.1. Ipar 4.0 eredete . . . . .	2
2.2. Előnyei . . . . .	2
2.3. Kialakítási alapelvek . . . . .	3
2.3.1. Összekapcsolás . . . . .	3
2.3.2. Információs átláthatóság . . . . .	4
2.3.3. Decentralizált döntéshozatal . . . . .	5
2.3.4. Technikai asszisztens . . . . .	5
2.4. Ember-robot kollaboráció . . . . .	6
2.4.1. Fogalmak tisztázása . . . . .	6
2.4.2. Biztonsági szempontok ember-robot kollaboráció esetén . . .	7
2.4.3. Érzelmi megfontolások . . . . .	10
2.5. KUKA-specifikus biztonsági funkciók . . . . .	10
2.5.1. Használt szakkifejezések . . . . .	11
2.5.2. Biztonság szempontú funkciók . . . . .	14
<b>3. Alkalmazáshoz szükséges műszaki feltételek elemzése</b>	<b>16</b>
3.1. Projekt részletes leírása . . . . .	16
3.2. A megfogó vezérlése . . . . .	17
3.2.1. A vezérlés hardveres kialakítása . . . . .	17
3.2.2. A megfogó szoftveres konfigurációja . . . . .	18
3.3. Sakkbábuk kialakítása, felismerése a képeken . . . . .	20
3.3.1. A sakkbábuk formatervezése . . . . .	20
3.3.2. Bábuk pozíciójának azonosítása . . . . .	23
3.4. QR-kód alapú képfeldolgozás - továbbfejlesztés . . . . .	26
3.4.1. A forráskód build-je . . . . .	26
3.4.2. ZXing könyvtár beimportálása és használata . . . . .	27
3.4.3. QR kód generálás és olvasás . . . . .	28
<b>4. Robotkar kalibráció, referencia felvétel kidolgozása</b>	<b>30</b>
<b>5. Robotkar mozgásának definiálása és programozása</b>	<b>31</b>

<b>6. Sakkalgoritmus beágyazása</b>	<b>32</b>
<b>7. Eredmények értékelése</b>	<b>33</b>





## JELÖLÉSEK JEGYZÉKE

A táblázatban a többször előforduló jelölések magyar és angol nyelvű elnevezése, valamint a fizikai mennyiségek esetén annak mértékegysége található. Az egyes mennyiségek jelölése – ahol lehetséges – megegyezik hazai és a nemzetközi szakirodalomban elfogadott jelölésekkel. A ritkán alkalmazott jelölések magyarázata első előfordulási helyüknél található.

# 1. BEVEZETÉS

## 1.1. Célkitűzés

A szakdolgozat célja megismertetni az olvasóval az Ipar 4.0 (az ipari fejlődés egyik legújabb és legmeghatározóbb trendje) fontos elemét, az ember-robot kollaborációt. A fókuszban egy gyakorlati megvalósítás, egy demo áll, viszont általános irányelvek és módszerek is taglalásra kerülnek. A szakdolgozat feltételezi, hogy az olvasó rendelkezik alapszintű ismeretekkel az Ipar 4.0 főbb elemeit illetően. A dolgozat fő szerepe az Ipar 4.0 egy kisebb részletének praktikus, szemléletes bemutatása, de mégis érdemes rendszerszemléletűen hozzágondolni a többi elemét is, mivel így kaphatunk csak teljes képet ennek funkciójáról.

## 1.2. Áttekintés – még nem végleges

Első körben a 4. ipari forradalom főbb vonásai kerülnek bemutatásra, különös tekintettel az ember-robot együttműködésre (későbbiekben HRC - Human-Robot Collaboration). Ezen belül több balesetmegelőző lépcsőfokról is szó lesz azért, hogy szemléletesebbek legyenek az előnyök. Ezt követően a szakdolgozat magját jelentő demóhoz használt KUKA LBR iiwa 7 R800 robotkarhoz tartozó biztonságtechnikai fogalmakat és specifikációkat ismertetem, mivel ennek keretében beszélhetünk a HRC-vel kapcsolatos funkciókról.

## 2. IRODALOMKUTATÁS

### 2.1. Ipar 4.0 eredete

Az Ipar 4.0 koncepcióját Németországban dolgozták ki, ahol világszinten kiemelkedő a termelési iparág, illetve világvezető a gyártó eszközök területén. Az Ipar 4.0 a német kormány stratégiai kezdeményezése volt, mely nagy mértékben támogatta az ipari szektor fejlesztését. Ilyen értelemben az Ipar 4.0-ra tekinthetünk egy olyan mozgalomként, amelynek célja megőrizni Németország befolyását a gépiparban és az autógyártás területén.[1]

Az alap koncepciót először a Hanover Fair-en<sup>1</sup> prezentálták 2011-ben. A bemutató óta az Ipar 4.0 Németország vezető témája a kutatások területén, egyetemi és ipari környezetekben különféle eseményeken. A fő irányvonal az új technológiákban és koncepciókban rejlő potenciál kihasználása felé mutat, ilyen területek:

- az IoT (Internet of Things<sup>2</sup>) elérhetősége és kihasználása,
- a technológiai és gazdasági folyamatok integrációja cégen belül,
- a valóság virtuális leképezése,
- 'okos' gyárak beleértve az 'okos' gyártást és termékeket.

### 2.2. Előnyei

Amellett hogy a digitalizáció és az új technológiák természetes következménye, az Ipar 4.0 megjelenése szintén kapcsolatban áll azzal a ténnyel, hogy a gyártásban a profit növelésére irányuló kezdeményezések, lehetőségek nagy része kiaknázásra került, új megoldásokat kellett keresni. A gyártási költségek csökkentek a Just-In-Time (röviden JIT) termelés bevezetésével, a lean elveinek alkalmazásával és a gyárak olyan helyre telepítésével, ahol a munkaerő lényegesen olcsóbb. Ha az előállítási költségek minimalizálása a célunk, az Ipar 4.0 egy ígéretes megoldásnak tűnik. Számos forrás alapján az Ipar 4.0 alkalmazása csökkentheti[2]:

- a gyártás költségét 10-30%-kal,
- a logisztikával kapcsolatos kiadásokat 10-30%-kal,

---

<sup>1</sup>A hannoveri a világ egyik legnagyobb kereskedelmi bemutatója. Körülbelül 6500 kiállító és 250.000 látogató vesz részt ezen a rendezvényen.

<sup>2</sup>A Dolgok internete fizikai eszközökből, járművekből, otthoni felszerelésekből és további elektronikát, szoftvert, szenzorokat, aktuátorokat tartalmazó tételekből álló hálózat, amelyek képesek egymással kapcsolatba lépni, adatot fogadni és küldeni.

- a minőségmenedzsmenthez köthető költségeket 10-20%-kal.

Ezekon kívül a koncepció alkalmazásának számos egyéb előnyéről szólhatunk: (1) új termékek piacra kerülési ideje csökken, (2) érzékenyebb reagálás a megváltozott vásárlói igényekre, (3) lehetővé teszi a személyreszabott tömeggyártást az összegyártási költség jelentős növelése nélkül, (4) rugalmasabb és barátságosabb munkakörnyezetet teremt, (5) a természetes erőforrásokat hatékonyabban hasznosítja.

## 2.3. Kialakítási alapelvek

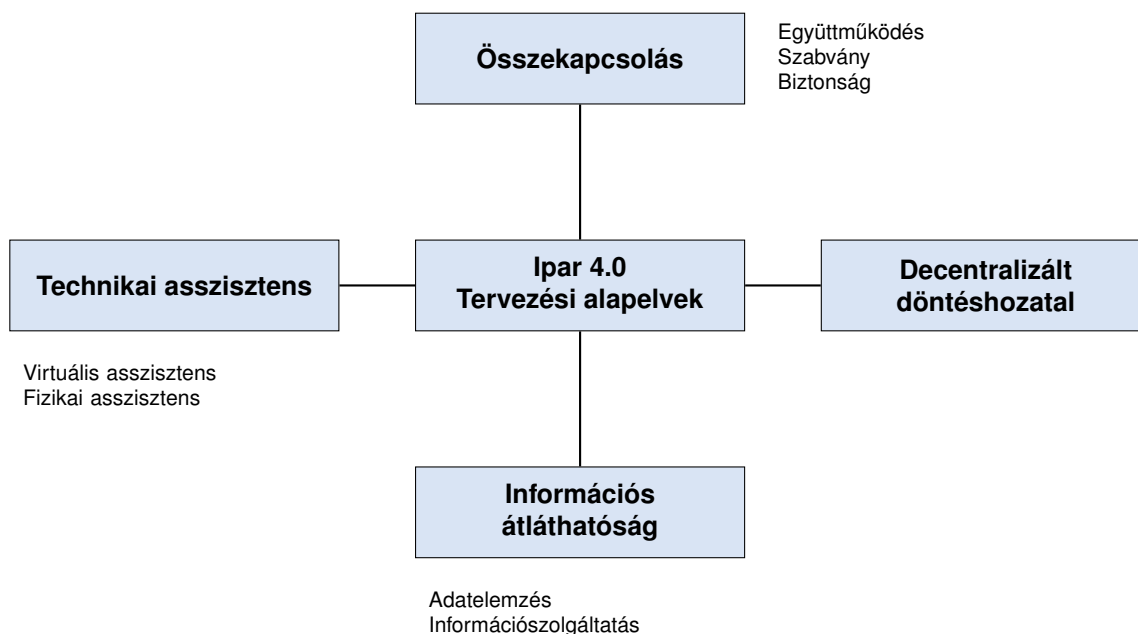
A számos szövegelemzés és átfogó irodalmi áttekintés négy fő dizájn elvet emelt ki, hogy irányvonalat mutasson a szakértőknek és tudósoknak az Ipar 4.0 környezet kialakításához: összekötés, információs átláthatóság, decentralizált döntéshozatal és technikai asszisztens (2.1. ábra). Ezek az alapelvek a következő alfejezetekben kerülnek részletes tárgyalásra az egyetemi és ipari publikációkban használt kifejezések (és következésképpen a kialakítási alapelvek) rövid elemzése után.

Összességében a két különböző típusú publikáció szövegelemzése nem mutat lényeges eltérést, mindkettő típus külön-külön elemzése ugyanazokat a kialakítási alapelveket eredményezi. Azonban szembe tűnő, hogy egyes dizájn elemeket gyakrabban tárgyalnak a gyakorlati publikációkban. Az ember-robot kollaboráció, adat- és információbiztonság és a decentralizált döntéshozatal gyakrabban fordul elő ipari kiadványokban. Az első kettővel kapcsolatos értekezések magas száma rávilágít az Ipar 4.0 eredményes implementálásának legnagyobb kihívásaira amivel az iparban dolgozók szembesülnek. Mindeközben a decentralizált döntéshozatalt tekintik az Ipar 4.0 legproblémásabb elemének, és ezért ez rendkívül részletes és átfogó tárgyalásra kerül.

### 2.3.1. Összekapcsolás

Gépek, eszközök, szenzorok és emberek kapcsolatba lépnek IoT-n (Internet of Things - Dolgok internete) és IoP-n (Internet of People - Emberek internete[3]) keresztül és így formálnak egy IoE-t (Internet of Everything - Minden internete[4]). A vezeték nélküli technológiák kiemelkedő szerepet játszanak az interakciók során, mivel lehetővé teszik az internetes hozzáférést mindenfelé. Az IoE-n keresztül összekötött emberek és eszközök képesek egymással információt megosztani, ami a kollaboráció alapját jelenti a közös célok elérése érdekében. 3 különböző típust különböztethetünk meg az IoE kapcsán: ember-ember együttműködés, **ember-robot kollaboráció** és robot-robot kollaboráció.[5]

A különböző gépek, eszközök, érzékelők és emberek egymás közti interakciója során elengedhetetlen szerepe van a széles körben elfogadott kommunikáci-



**2.1. ábra:** Ipar 4.0 tervezési szempontok

ós szabványoknak. Ezek teszik lehetővé a különböző gyártóktól érkező moduláris eszközök rugalmas kombinálását. Ez a modularizáció az alapfeltétel, hogy az Ipar 4.0 'okos' gyárai alkalmazkodni tudjanak a folyamatosan változó piaci igényekhez vagy a személyreszabott rendelésekhez.

Ahogy nő az IoE-ben részt vevők száma, a monetáris<sup>1</sup> és politikai érdekek meg fogják növelni az ilyen létesítmények elleni káros támadások számát, így az igény is nőni fog a magasabb fokú informatikai biztonság iránt.

### 2.3.2. Információs átláthatóság

Az összekapcsolt objektumok és emberek növekvő számának köszönhetően, a fizikai és a virtuális világ egybeolvadása lehetővé tesz egy újfajta információs modellt[6]. Az érzékelők összekapcsolása révén képezhetünk egy digitális, virtuális leképezést a világunkról.

Az összefüggés-tudatos információ az IoE résztvevői számára elengedhetetlenek a megfelelő döntések meghozatalához. Az ilyen összefüggés-tudatos rendszerek a feladataikat virtuális és a fizikai világból érkező információk alapján látják el. A virtuális világból érkező információkra példák az elektronikus dokumentumok, rajzok, szimulációs modellek. A fizikai világ információi például a pozíció vagy a szerszám állapota. A fizikai világ elemzéséhez az érzékelők felől érkező nyers adatokat magasabb szintű értelmezési és egyéb információval kell kiegészíteni. Ahhoz,

---

<sup>1</sup>pénzhez vagy valutához kötődő

hogy az átláthatóságot fenntartsuk, az adatelemzés eredményeit egy olyan kisegítő rendszerbe kell bevinni, ami minden IoE résztvevő számára elérhető. A folyamatkritikus információk esetén a valós idejű adatszolgáltatás elengedhetetlen.

### **2.3.3. Decentralizált döntéshozatal**

A decentralizált döntések meghozatalának két alappilére az objektumok és emberek összekapcsolása, illetve a termelő létesítményen belülről és kívülről érkező információk átláthatósága. Az összekapcsolt és decentralizált döntéshozó egységek lehetővé teszik a lokális információk globálissal együtti felhasználását egyazon időben, így elősegítve az átgondoltabb döntéshozatalt és így növelve összességében a termelékenységét. Az egyes IoE elemek a feladataikat annyira önállóan látják el, amennyire csak lehet. A feladatok csak kivételek, zavarok vagy ellentmondásos célok esetén kerülnek továbbításra magasabb szintre.

Gyakorlati szempontból a decentralizált döntéshozatalt a kiber-fizikai rendszerek teszik lehetővé. Ezek beágyazott számítógységeinek, szenzorainak és aktuátorainak felhasználásával történik fizikai világ autonóm nyomon követése és az irányítása.

### **2.3.4. Technikai asszisztens**

Az Ipar 4.0 'okos' gyáraiban az ember szerepe alapvetően megváltozik, gépkezelő helyett inkább stratégiai döntéshozóvá és rugalmas problémamegoldóvá válik. A termelési folyamatok növekvő komplexitása miatt, ahol a kiber-fizikai rendszerek összetett hálózatot alkotnak és decentralizált döntéseket hoznak, az embereknek támogató rendszerekre van szükségük. Ezeknek a rendszereknek a szerepe az információk összegyűjtése és megjelenítése egyértelműen és érthetően annak érdekében, hogy az emberek jól megalapozott döntéseket tudjanak hozni, és magas prioritású problémákat tudjanak megoldani rövid időn belül. Jelenleg az embereket főként az okostelefonjaik és táblagépeik kötik össze az IoT-vel[7]. A hordozhatóság kiemelkedően fontossá fog válni a jövőben amint a jelenlegi kihívásokon (mint például az energiaellátás) sikerül felülkerekedni.

Az emberek robotok általi fizikai kisegítése (a robotika területen elért fejlesztésekkel) szintén a technikai asszisztens szerep részét képezi. A robotok számos feladatot képesek elvégezni, amelyek az ember számára kellemetlenek, túl fárasztóak vagy veszélyesek más munkásokra nézve[8]. Az emberek fizikai feladatokban hatékony, sikeres és biztonságos segítésének érdekében szükséges, hogy a robotok az ember társaikkal zökkenőmentesen és intuitívan működjenek együtt[8]. Ezen felül elengedhetetlen, hogy az emberek megfelelő képzsben részesüljenek az adott

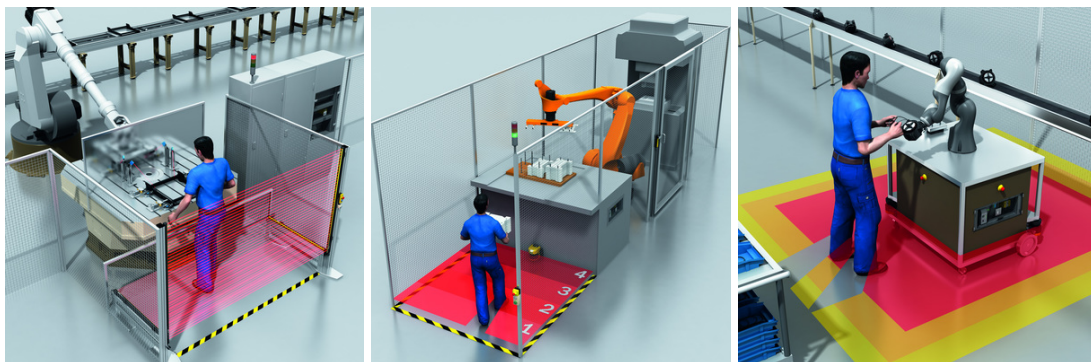
ember-robot kollaborációhoz[9].

### 2.4. Ember-robot kollaboráció

#### 2.4.1. Fogalmak tisztázása

Az ember és a robot közösen végzett feladataikat különböző interakciós szinteken valósíthatják meg, ezeket érdemes egymástól elhatárolni<sup>1</sup> (2.2. ábra):

1. Robot cella (Robotic cell): a robot önállóan végzi a feladatát az embertől kerítéssel elválasztva. Ez esetben nem beszélhetünk ember-robot együttműködésről.
2. Együttes jelenlét (Coexistence): a robot és az ember közel helyezkedik el egymáshoz védőkerítés nélkül, de nincs közös munkaterük. A robotnak van saját meghatározott tere.
3. Szinkronizált munkavégzés (Synchronized work): olyan elrendezés, melyben az ember és a robot osztozik egy közös munkateren, de egyszerre csak egyikük aktív. A munkamenet az ember és a robot jól definiált 'koreográfiája'.
4. Kooperáció (Cooperation): a két „partner” mindegyike a saját feladatával foglalkozik. A munkaterük lehet közös, de nem dolgozhatnak sem ugyanazon a terméken, sem ugyanazon a munkadarabon.
5. Kollaboráció (Collaboration): olyan elrendezés, amely esetén az ember és a robot közösen és szimultán dolgozik egyazon terméken vagy munkadarabon. Tipikusan a robot megfogja, átnyújtja és tartja a munkadarabot amíg a munkás dolgozik rajta.



**2.2. ábra:** Balról jobbra: együttes jelenlét, kooperáció, kollaboráció<sup>2</sup>

---

<sup>1</sup>Forrás: <https://hohmannchris.wordpress.com/2017/02/08/cobots-more-cooperation-than-collaboration/>



### 2.4.2. Biztonsági szempontok ember-robot kollaboráció esetén

A biztonságos ember-robot együttműködés érdekében az elmúlt években különböző stratégiák lettek kifejlesztve. Ezek a módszerek különböző biztonságtípusra építenek, többek közt[10]:

- az ütközésbiztonság érdekében csak 'biztonságos' /kontrollált ütközésre kerülhet sor robotok, emberek és akadályok között. Az emberekre gyakorolt erő/-nyomaték határolása a fő szempont.
- aktív biztonsági rendszer az ember és a berendezés közötti közelgő ütközések időben történő észlelésére és a művelet megszakítására ellenőrzött módon. Távolságérzékelők, látórendszerek és erő/érintkező érzékelők használhatóak erre a célra.
- a hardver elemek működtetése során adaptív biztonsági stratégia a beavatkozáshoz, illetve korrekciós eljárások implementálása, amelyek az ütközéseket a munkafolyamat megszakítása nélkül kerülnek el.

Ilyen irányban nemzeti és nemzetközi szabványokat, irányelveket és törvényeket vezettek be annak érdekében, hogy a rendszerintegrátorok könnyen be tudják építeni a biztonsági metódusokat a saját rendszerükbe. Tekintettel arra, hogy egy kollaborációs munkatér nem csak az ember és a robot tartózkodik, hanem egyéb segédberendezések is megjelennek (például: elektromos csavarbehajtók, elektromos szorítóberendezések stb.) minden egyes cella egyedülálló kockázatot jelent biztonságtechnikai szempontból. Ebből kifolyólag minden berendezés és folyamat esetén be kell tartani az ezekre vonatkozó hatályos törvényeket és szabványokat. A 2.1., 2.2. és 2.3. táblázatok indikatív példákat tartalmaznak az EU legismertebb törvényei és általános normáiból válogatva.

2.1. táblázat: EU Directives

Title	Description
2006/42/EC	Machinery Directive (MD)
2009/104/EC	Use of Work equipment Directive
89/654/EC	Workplace Directive
2001/95/EC	Product Safety Directive
2006/95/EC	Low Voltage Directive
2004/108/EC	Electromagnetic Compatibility Directive (EMC)

<sup>2</sup>Képek forrása: <https://www.safety-solutions.net.au/content/machine/article/safety-solutions-for-intelligent-human-robot-collaboration-990038334>

**2.2. táblázat:** Indicative general standards

Title	Description
EN ISO 12100	Safety of machinery - General principles for design - Risk assessment and risk reduction
EN ISO 13849-1/2	Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design, Part 2: Validation
EN 60204-1	Safety of machinery - Electrical equipment of machines - Part 1: General requirements
IEC 62061	Safety of machinery - Functional safety of safety related electrical, electronic and programmable electronic control systems

**2.3. táblázat:** Robot standards

Title	Description
EN ISO 10218-1	Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots
EN ISO 10218-2	Robots and robotic devices - Safety requirements for industrial robots - Part 2: Robot systems and integration
ISO/PDTS 15066	Robots and robotic Devices - Collaborative Robots

Jelenleg közel 30 aktív uniós irányelv és körülbelül 600 szabvány létezik a biztonságtechnikával kapcsolatosan. Különösen robot cellákkal kapcsolatban 2.3. táblázat előrevetít számos különböző stratégiát a megfelelő biztonság kialakítására. Pár a legfontosabbak közül[10]:

- **Biztonsággal kapcsolatos vezérlőrendszerek teljesítménye:** biztonságkritikus vezérlőrendszer-egységeknek biztosítaniuk kell, hogy az egyes hibákra van oly mértékű tolerancia, hogy a biztonságot nem veszélyezteti.
- **Robotmegállító funkciók:** minden robotnak rendelkeznie kell védőleállító funkcióval, továbbá független vészleállító funkcióval. Külső biztonsági berendezésekkel való kapcsolatot is biztosítani kell.
- **Sebességkorlátozás:** a robot végén található beavatkozó és a szerszámközéppont (TCP - Tool Center Point) sebessége kontrollálható kell legyen. Kollaboratív munkaállomások esetén a TCP sebessége nem haladhatja meg a 250 mm/s-ot.
- **Kollaboratív műveletekre vonatkozó követelmények:** együttműködésre képes robotok kell hogy működtessenek valamilyen látható indikátort amikor a robot kollaboratív műveletet hajt végre. A következő követelmények érvényesek:

- **Biztonsági rendszer által kezdeményezett felügyelt megállás:** a robotnak kötelező megállnia, amikor az ember a kollaboratív munkaterületen tartózkodik. Amint az ember elhagyja ezt a területet, a robot folytathatja az automatikus munkavégzési folyamatot.
- **Kézi vezetés** (Hand guiding): a kézi vezetéshez alkalmazott berendezésnek tartalmaznia kell egy vészleállítót és egy engedélyező szerkezetet. E művelet során biztonságkritikai szempontból tanúsított módon nyomon kell követni a robot sebességét. Számos technológia felhasználható az ember által kézzel történő irányításhoz, mint például az impedancia vagy a merevség szabályozása.
- **Sebesség és pozíció monitorozása**<sup>1</sup>: a robot fent kell tartson egy adott távolságot az üzemeltetőtől. Ezt integrált jellemzők vagy külső inputok kombinációjának figyelésével tudja megvalósítani.
- **Teljesítmény- és erőhatárolás belső dizájnnal:** a teljesítmény-/erőhatároló funkcióknak be kell tartatniuk a szabvány által meghatározott határértékeket. Ha ezt túllépik, megállást kell eszközölniük.
- **Vezérlőrendszer által határolt erő és teljesítmény:** egy vezérlőfunkció biztosítja, hogy a maximális erő és teljesítmény értékeket ne lehessen túllépni.
- **Robot mozgásterének korlátozása:** feleslegesen nagy tér elhatárolása szükséges, ha a robotnak engedélyezett a teljes mozgástartomány kihasználása. A mozgástér leszűkítése lehetséges a robotba integrált rendszerek kihasználásával vagy külső korlátozó eszközök telepítésével, illetve e kettő kombinációjával. Dinamikus korlátozás érhető el vezérlőeszközök (kapcsolók, fényfüggönyök stb.) alkalmazásával, ha további limitációkra van szükség a robotprogram végrehajtása során.
- **Minimális távolság meghatározása:** feladattól függően kockázatelemzés után kerül meghatározásra az ember és a robot közötti minimális távolság. Az elemzés során érdemes figyelembe venni a) a végeffektorokhoz és az esetlegesen ezek által tartott munkadarabokhoz kapcsolódó kockázatokat, b) a munkaterület elrendezését, c) a munkás feladatát és d) a rendszer felhasználhatóságát.
- **Ütközésérzékelés:** a biztonsági funkcióknak meg kell tudniuk határozni mind az ember, mind pedig a robot pillanatnyi pozíciója és sebessége alapján, hogy a minimális távolság alá csökkenhet-e az ember és a robot távolsága (ütközés).

---

<sup>1</sup> folyamatos megfigyelés

- **Esetleges ütközések elkerülése:** ez a funkció lehetővé teszi a robot számára, hogy megelőzze az ütközéseket a) lassítás vagy megállás, b) az meghatározott útja során eszközölt irányváltás és c) egy másik biztonságos útvonalon haladás által.
- **Technológiai és ergonómiai követelmények:** az ember és robot közötti esetleges ütközés esetén megfelelő óvintézkedéseknek kell biztosítaniuk, hogy éles, hegyes vagy durva felületek nem találhatók az érintkezési zónában. Továbbá a környező munkaterületet (ahol az ember a kollaboratív robottal ütközhet) úgy kell kialakítani, hogy a felhasználó számára elegendő tér biztosítva legyen ahhoz, hogy kockázatos helyzeteket elkerülje.

Ezeket a funkciókat szabványosítással foglalkozó szervek dolgozták ki a különböző biztonsági szempontok átfedés nélküli ellenőrzésének érdekében. Átfedések előfordulhatnak az egyes projektek kivitelezésekor; a biztonsági követelményeknek teljes mértékben való megfelelést minden esetben felül kell vizsgálni.

### 2.4.3. Érzelmi megfontolások

Az ember-robot együttműködésnek és a kollaboratív celláknak a kialakítása, valamint az ember és a robot között hatékony feladatmegosztás szorosan kapcsolódik társadalom-fiziológiai kérdésekhez. A mélyebb megértés és az ember viselkedésének megjósolása, valamint a HRI (Human-Robot Interaction - Ember-Robot Interakció) fejlődése részben a kognitív mérnökséghez, pszichológiához és szociológiához kötődik. A bizalom, a terheltség és a kockázatosság a legfontosabb emberi tényezők, amik befolyásolják az automatizálási technológiák alkalmazását. Az emberközpontú robotcella kialakítása során az emberi tényezők, mint a terheltség, éberség, helyzetfelismerés, hibák stb. figyelembe vétele ugyanúgy fontos, mint a kognitív mérnöki szempontok és az ergonómia[11].

## 2.5. KUKA-specifikus biztonsági funkciók

A következő részben a KUKA Sunrise OS 1.16[12] biztonsági funkciói kerülnek taglalásra. Az alábbi fogalmak és szempontok egy része már a 2.4.2 fejezetben előkerült általánosságban, viszont gyakorlati szempontok miatt ezek specifikus tárgyalására is szükség van. Az itt leírtak tájékoztató jellegűek, céljuk a szakdolgozat értelmezhetőségének javítása, nem foghatóak fel általános iránymutatásként. Adott OS (Operating System) és egyéb kiegészítő berendezések esetén mindig található ezekhez mellékelt biztonsági utasítás, ezek betartása elengedhetetlen a hatékony és biztonságos működtetéshez.

A biztonsági funkciókat két csoportra lehet osztani:

- Biztonságkritikus funkciók az emberi biztonság fenntartása érdekében  
Az ipari robot biztonságkritikus funkciói megfelelnek a felsorolt biztonsági követelményeknek:
  - EN ISO 13849-1 (2.2. táblázat) szabvány 3-as kategóriája (Category 3), illetve a d teljesítményszint besorolás (Performance Level d)
  - EN 62061 (2.2. táblázat) SIL 2 követelménye Ezek a követelmények azonban csak a következő feltételek fennállásakor teljesülnek:
  - Az ipari robot minden biztonsági szempontból releváns mechanikus, illetve elektromechanikus komponense a biztonságos működés szempontjából le lett tesztelve a beüzemeléskor és legalább egyszer minden évben, hacsak nem a kockázatelemzők másképp határoztak. Ilyen eszközök:
    - \* Vészstop eszköz (Emergency stop device) a smartPAD-en
    - \* Engedélyező kapcsoló a smartPAD-en
    - \* Kulcsos kapcsoló a smartPAD-en
- Nem biztonságkritikus funkciók a gépek védelme érdekében  
Az ipari robot nem biztonságkritikus funkciói nem tartoznak kitüntetett biztonsági követelményekhez.

### 2.5.1. Használt szakkifejezések

**2.4. táblázat:** A használt szakkifejezések

Kifejezés	Leírás
Tengely mozgástartománya (Axis range)	Az a tartomány, amelyben a tengely mozoghat. Ezt a tartományt minden tengelyre külön meg kell határozni.
Megállási út (Stopping distance)	Megállási út = reakcióidő alatt megtett út + fékút. A megállási út része a veszélyzónának.
Munkatér (Workspace)	A beavatkozó (manipulator) ezen tartományon belül mozoghat. A munkatér az egyes tengelyek mozgástartományából származtatott.
Automatikus (AUT) (Automatic)	A programvégrehajtáshoz tartozó üzemmód. A beavatkozó a programozott sebességgel mozog.
A következő oldalon folytatódik	

## 2.4. táblázat – a 11. oldalon kezdődik

Kifejezés	Leírás
Gépkezelő/Felhasználó (Operator/User)	Az ipari robot üzemeltetője lehet a menedzsmen, alkalmazott vagy megbízott ember, aki felelős a robot működtetéséért.
Veszélyzóna (Dangerzone)	A veszélyzóna a munkaterületből és a megállási útból áll.
Élettartam (Service life)	A biztonságkritikus komponensek élettartama az ügyfélnek kiszállítástól számítandó. Az élettartam nem függ attól, hogy a komponens a robotvezérlőbe került beépítésre vagy máshova, vagy akár egyáltalán nem is volt használva, mivel a biztonságkritikus alkatrészek is ki vannak téve előregedésnek a tárolás alatt.
CRR	<p>Irányított Robot Visszatérés (Controlled Robot Retraction)</p> <p>A CRR egy olyan üzemmód, amelybe akkor lehet lépni, ha az ipari robotot megállította a biztonsági vezérlő a következő okok valamelyike miatt:</p> <ul style="list-style-type: none"> <li>• A robot megsértett egy tengelyspecifikus vagy egy descartesi koordinátára vonatkozó szabályt.</li> <li>• A biztonság-orientált szerszám elhelyezkedése a megfigyelt tartományon kívülre esik.</li> <li>• A robot megsérti az erőre vagy a nyomatékra vonatkozó korlátozásokat.</li> <li>• A pozíciószenzor nem 'mastered' vagy 'referenced'.<sup>1</sup></li> <li>• A csuklóban lévő nyomásszenzor nem 'referenced'.</li> </ul> <p>CRR üzemmódba lépés után a robotot újra lehet mozgatni.</p>
KUKA smartPAD	Lásd „smartPAD”
Beavatkozó (Manipulator)	A robotkar és a hozzá telepített elektronika.
A következő oldalon folytatódik	

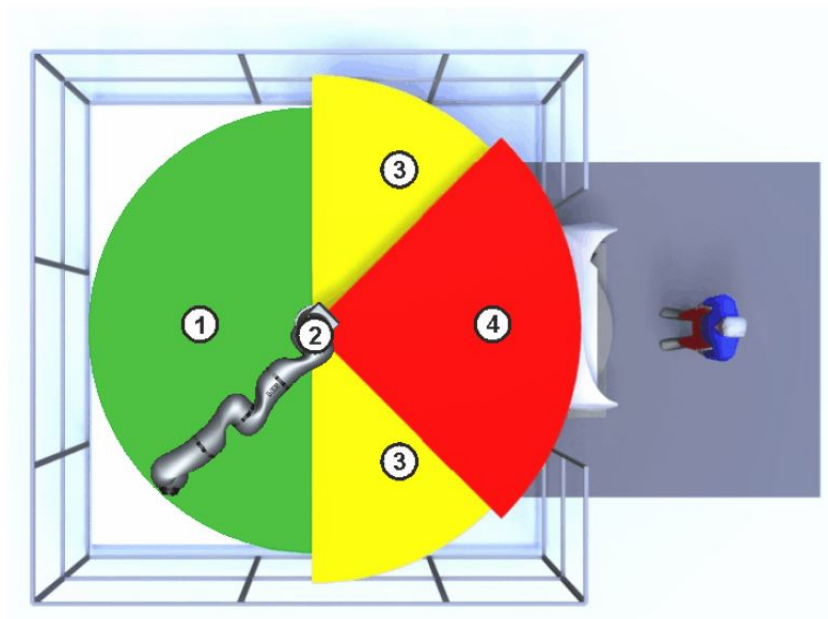
<sup>1</sup>A két kifejezést nem célszerű lefordítani magyarra.

2.4. táblázat – a 11. oldalon kezdődik

Kifejezés	Leírás
Biztonsági sáv (Safety zone)	A beavatkozó nem léphet a biztonsági sávba. Ez a tartomány kívül esik a veszélyzónán.
Biztonsági megállás	A biztonsági megállást a biztonsági vezérlő idézheti elő, így megszakítva a munkamenetet és megállítva a robot mozgását. Biztonsági megállás esetén a programadatok nem vesznek el, így a program folytatódhat a megszakítás pontjától kezdve. A biztonsági megállás lehet 'Stop category 0', 'Stop category 1' vagy 'Stop category 1 (path-maintaining)' (útvonal követés).
smartPAD	A smartPAD egy kézbevehető irányító panel a robotcellához (-állomáshoz). A robotállomás működtetéséhez szükséges összes üzemeltetői és megjelenítési funkciót tartalmazza.
Stop category 0	A motorok kikapcsolnak és a fékek összezárnak.
Stop category 1	A robotkar lefékeződik és közben nem marad a programozott útvonalon. A beavatkozó a motorok segítségével jut álló pozícióba. Amint megállnak a csuklók a motorok kikapcsolnak és a fékek behúzódnak.
Stop category 1 (path-maintaining)	A beavatkozó lefékeződik és közben a programozott útvonalon marad. Amint megállnak a csuklók a motorok kikapcsolnak és a fékek behúzódnak. Ha 'Safety category 1 (path-maintaining)'-et idéz elő a biztonsági vezérlő, akkor ez felügyeli is a megállítás folyamatát. A fékek behúzódnak és a motorok legkésőbb 1 másodperc elteltével kikapcsolnak. Hiba esetén 'Stop category 1' kerül végrehajtásra.
Rendszerintegrátor (System integrator / plant integrator)	A rendszerintegrátorok feladata az ipari robot integrálása egy rendszerbe és ennek hitelesítése.
T1	Teszt üzemmód, manuálisan csökkentett sebesség (Manual Reduces Velocity) ( $\leq 250 \text{ mm/s}$ ) <b>Megjegyzés:</b> kézi vezetés T1-ben esetén a sebesség nem kimondottan csökkentett, inkább limitált biztonság szempontú sebességmonitorozás által.
A következő oldalon folytatódik	

2.4. táblázat – a 11. oldalon kezdődik

Kifejezés	Leírás
	<b>Megjegyzés:</b> a 250 mm/s maximális sebesség a mobil platformokra nem vonatkozik.
T2	Teszt mode, manuálisan állított nagy sebesség (> 250 mm/s megengedett)



**2.3. ábra:** Példa a mozgástartományokra: 1. munkaterület, 2. beavatkozó, 3. megál-  
lási út, 4. biztonsági sáv

### 2.5.2. Biztonság szempontú funkciók

Az alábbi funkciók permanensen jelen vannak az ipari robot esetén:

- Vészleállító eszköz (EMERGENCY STOP device)
- Engedélyező eszköz (Enabling device)
- Az üzemmód lezárása (a kulcsos kapcsoló által)

A következő biztonságszempontú funkciók előre konfiguráltak és a robotvezérlő biztonsági interfészén keresztül bármikor a rendszerbe integrálhatóak:

- Üzemeltető biztonsága (= a fizikailag jelen lévő biztonsági kerítések megfigye-  
lésére alkalmas csatlakozó)
- Külső vészleállító eszköz



- Külső, útvonalkövető megállást előíró eszköz (Stop category 1 (path- maintaining)) (2.4. táblázat)

Egyéb biztonsági funkciók konfigurálása esetleges, pl.<sup>1</sup>:

- Külső engedélyező eszköz
- Tengelyspecifikus munkatér monitorozás
- Descartes-i munkatér megfigyelés
- **Descartes-i védett tér monitorozás**
- Sebesség nyomonkövetés
- **Tengelyekben ébredő nyomatékok megfigyelése**
- **Ütközés detektálása**

Az egyes pontokról további információ a technikai specifikációban található[12].

---

<sup>1</sup>A vastag betűvel szedett pontok a szakdolgozat fontos elemei

### 3. ALKALMAZÁSHOZ SZÜKSÉGES MŰSZAKI FELTÉTELEK ELEMZÉSE

#### 3.1. Projekt részletes leírása

A projekt célja egy olyan robot demo hardveres és szoftveres kidolgozása, amely képes egy emberrel (továbbfejlesztés után akár egy másik robottal) lejátszani egy sakkjátszmát. A demo az ember-robot kollaboráció bemutatására szolgál, fontos szempont az interakció biztonságos megvalósítása mind az emberre, mind a környező tárgyakra tekintettel.

A megvalósításhoz a következő problémák megoldására van szükség:

1. szükséges biztonsági funkciók beüzemelése,
2. a bábuk helyzetének felismerése az egyes lépések előtt és után,
3. a bábuk megfogása és mozgatása (ide tartozik a kalibráció és a referenciafelvétel),
4. sakkalgoritmus beágyazása a programba,
5. a sakkbábúk és a tábla megtervezése és megvalósítása,
6. jelzés a robotkar számára, ha lépés történt.

A felsorolt pontok a projekt során következőképpen kerültek kidolgozásra:

- A bábuk helyzetének detektálása a projekt során QR-kód kereső és olvasó képfeldolgozó eljárásokon alapul (a bábuk tetején található a kód). A kamera a roboton kerül rögzítésre.
- A bábuk mozgatása egy elektromosan vezérelt, párhuzamos megfogó (gripper) segítségével történik.
- Ahhoz hogy a bábuk megfogása egyszerű legyen, azonos magasságú és azonos módszerrel megfogható bábuk készülnek.
- A biztonsági funkciók főként az tengelyekben ébredő plusz nyomatékok monitorozására és biztonsági zónák definiálására épül.
- A tábla és a robotkar, illetve a megfogó (egy jól definiált pontja) és a robotkar relatív helyzetének kalibrálására a robotvezérlő szoftverben elérhető alapfunkciók kerültek felhasználásra.

- A sakklépés megtörténtét a robotvezérlőhöz kötött külső gombbal tudja a felhasználó jelezni.
- A képek fogadása, feldolgozása és a sakkalgoritmus futtatása mind a robotvezérlőn történik.
- Mivel a robotvezérlőn Java alapú környezet fut magas szinten, így a képfeldolgozó és a sakkozó programok is ebben lettek implementálva.

## 3.2. A megfogó vezérlése

### 3.2.1. A vezérlés hardveres kialakítása

Ahhoz hogy a megfogót (grippert) a robotvezérlőn futó programból lehessen irányítani több kiegészítő eszközre is szükség van a gripperen és a vezérlőegységén kívül. Többféle konstrukcióval is el lehet érni ezt a célt; a projekt során használt összeállítás elemei (3.1):

- **Megfogó (gripper):** párhuzamosan mozgó, két pofájú, elektromos megfogó; pontos típusa: SCHUNK MEG 50 EC[13].
- **Grippervezérlő:** a megfogóhoz tervezett vezérlő; pontos típusa: SCHUNK Controller MEG EC[13]
- **Analóg és digitális I/O modulok**<sup>1</sup>: a robotvezérlő ezen modulok segítségével tudja irányítani a grippervezérlőt. A felhasznált eszközök Beckhoff gyártmányúak. Pontos típusaik:
  - EL1809: 16 csatornás, digitális bemeneti modul<sup>2</sup>
  - EL2809: 16 csatornás, digitális kimeneti modul<sup>3</sup>
  - EL3002: 2 csatornás, analóg bemeneti modul<sup>4</sup>
  - EL4032: 2 csatornás, analóg kimeneti modul<sup>5</sup>
- **EtherCAT<sup>6</sup> Coupler:** az I/O modulok az ún. E-bus-on keresztül kommunikálnak. Ahhoz hogy a robotvezérlőhöz lehessen kötni az E-bus-t, EtherCAT Coupler-re van szükség. Pontos típusa: Beckhoff EK1100<sup>7</sup>.

---

<sup>1</sup>I/O ((Input/Output): bemeneti és kimeneti modulok

<sup>2</sup><https://www.beckhoff.com/english.asp?ethercat/el1809.htm>

<sup>3</sup><https://www.beckhoff.com/english.asp?ethercat/el2809.htm>

<sup>4</sup><https://www.beckhoff.hu/english.asp?ethercat/el3002.htm>

<sup>5</sup><https://www.beckhoff.com/english.asp?ethercat/el4032.htm>

<sup>6</sup>Általános ismertető az EtherCAT-ről: <https://en.wikipedia.org/wiki/EtherCAT>

<sup>7</sup><https://www.beckhoff.com/english.asp?ethercat/ek1100.htm>

- **Tápegység:** a fentebb felsorolt eszközök mindegyike 24 V megtáplálást igényel, ezt a projekt során egy Siemens tápegység szolgáltatja.

### 3.2.2. A megfogó szoftveres konfigurációja

KUKA SunriseOS esetén (és általánosságban a KUKA robotok esetén) a különböző bemeneti, kimeneti és kommunikációs csatornák kezelésére I/O konfigurációs fájl generálására van szükség. Ezeket a fájlokat a WorkVisual nevű program segítségével lehet létrehozni és szerkeszteni. Adott SunriseOS-hez meghatározott a kompatibilis Sunrise Workbench verzió (ebben a programban a legegyszerűbb a robotvezérlőn futó program megírása, feltöltése). Adott Sunrise Workbench verzóval kompatibilis WorkVisual verzóra vonatkozó információkat a Sunrise Workbench-et megnyitva a Help->Sunrise.OS Release Notes menüpont alatt találunk. A szakdolgozathoz felhasznált szoftverek és környezet:

- SunriseOS 16
- Work Visual 5.0.5 build600
- Windows 10 a PC-n

A megfelelő IOConfig elkészítéséhez az alábbi lépések szükségesek:



3.1. ábra: Kép a megfogóhoz tartozó konstrukcióról

1. Ahhoz hogy a WorkVisual verziót összekössük a Sunrise Workbench-csel importálni kell a Workbench-hez tartozó Sunrise.kop fájl a WorkVisual-ba. Ez a fájl a telepített Workbench verzióhoz tartozó mappán belül a 'WorkVisual AddOn' nevű almappában található. Az importálás menete: WorkVisual->Extras->Option package management->Install... gomb. A felugró ablakban lehet kiválasztani a megfelelő Sunrise.kop fájlt és telepíteni. Ahhoz hogy az egyes Beckhoff modulokkal lehessen kommunikálni szükség van a Device description file'-ok importálására. Ezek a fájlok a gyártó oldaláról letölthetőek<sup>1</sup>. A (XML) fájlok importálásához a WorkVisual->Import / Export->Import device description file lehetőséget kell választani (ezt a műveletet adott eszközön csak egyszer kell elvégezni, új projekt esetén ezt a lépést már ki lehet hagyni). A megfelelő fájlok kiválasztása és importálása után szükség van a DTM Catalog frissítésére (WV->Extras->DTM Catalog Management->Search for installed DTMs). A fájlok használatához a 'Known DTMs' részről át kell emelni az elemeket a Current DTM Catalog' részre (3.2).
2. A Sunrise Workbench-ben új projekt létrehozása után a projektre jobb egérgomb->New->I/O Configuration. A WorkVisual automatikusan elindul. A projektre jobb egérgommbal kattintva (WV-ban) a 'Set as active controller' lehetőséget kell választani. A busz struktúrákhoz hozzá kell adni a 'KUKA Extension Bus (SYS-X44)' elemet ahhoz, hogy az EtherCAT kommunikációt inicializáljuk a robotvezérlőben. Ehhez adhatjuk hozzá az EK1100 EtherCAT Coupler-t, ami az összeköttetést biztosítja a robotvezérlőben található EtherCAT hálózat és a modulok között (E-bus). Az egyes modulokat ehhez adhatjuk hozzá a programban. **Fontos:** az egyes fájlokat olyan sorrendben kell hozzáadni, ahogy azok fizikailag kapcsolódnak egymáshoz (3.3 ábra).
3. Ahhoz hogy ezeket a be- és kimeneteket a Sunrise Workbench-ben használni tudjuk szükség van Sunrise I/O Group létrehozására (VW->IO Mapping->Sunrise I/Os->Creates signals at the provider). Az I/O Group-nak tetszőleges nevet adhatunk. Az egyes be- és kimenetek kezeléséhez létrehozhatunk változókat az I/O Group-on belül. A gripper nyitásához és zárásához alapvetően 3 változó deklarálása elegendő<sup>2</sup>), pl.:
  - OpenGripper: bool, output, digital
  - CloseGripper: bool, output, digital

<sup>1</sup>Device description files: <https://www.beckhoff.com/english.asp?download/elconfg.htm>

<sup>2</sup>A gripper nyitásához és zárásához elegendő csak a digitális ki- és bemeneteket használni, de például a pozíció követéséhez használni kell analóg bemeneti modult, a fogóerő programból történő beállításához pedig analóg kimeneti modult.

- Status: bool, input, digital

Az egyes változókat a kívánt bemenetekkel és kimenetekkel Drag and drop módszerrel lehet összerendelni (3.4). Megfelelő összekötés esetén a változók melletti szürke nyilak zölddé változnak.

4. Az elkészült I/O konfigurációt exportálni kell ahhoz, hogy a Sunrise Workbench-ben használni lehessen (WV->File->Import / Export->Export I/O Configuration to Sunrise Workbench project). A Workbench-ben ezek után megjelenik az src' mappában egy ioAccess nevű csomag. Ezen belül található az IO Group-hoz tartozó osztály (pl.: GripperContolIOGroup.java). Ez tartalmazza a szükséges metódusokat a gripper vezérléséhez, ezeket lehet meghívni a programból.

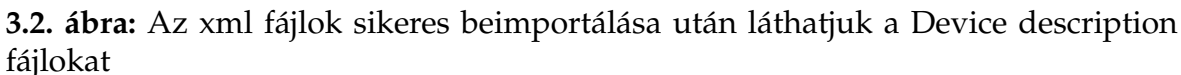
### 3.3. Sakkbábúk kialakítása, felismerése a képeken

A sakkprogram működtetéséhez elegendő a bábúk új pozíciójának felismerése azután, miután az ember lépett. Elég csak az ember bábuját figyelni (a robot vezeti magának a bábuk pillanatnyi helyzetét). A sakkbábúk felismerére többfajta módszer is megfelelő lehet. A képfeldolgozást lehet saját vagy szabványos mintára alapozni (pl.: QR-kód alapú felismerés - 3.4 fejezet), illetve színes kamerakép esetén adott színek keresésére. A szakdolozat során a második módszer került beágyazásra, viszont a sakkbábuk formai kialakításakor a QR-kód alapú felismerhetőség is fontos szempont volt, így biztosítva az ilyen irányú továbbfejlesztés lehetőségét.

#### 3.3.1. A sakkbábuk formatervezése

Ahhoz hogy a megfogó fel tudja emelni a bábukat könnyebség, ha a bábuk egyforma magasak, vagy ha a megfogó mindegyik bábut egy kitültetett, egy magasságban lévő részénél fogja meg (akár a talprészénél vagy alá is nyúlhatna). A talprésznél megfogás jó megoldás lehet, ha maguk a bábuk színesek, és így biztosított a képeken való felismerhetőségük. A QR-kódos továbbfejlesztéshez viszont elkerülhetetlen, hogy a QR-kód a bábuk tetején legyen elhelyezve. Ezen megfontolások alapján a bábuk végleges konstrukciója (3.5 ábra):

- Mindegyik bábu tetején található egy négyzet alapú, szélességéhez viszonyítva alacsony hasáb, amire az adott színt/színeket/QR-kódot rá lehet ragasztani.
- Ennél a hasábnál fogva emeli meg a megfogó a bábút, így kevésbé kell lenyúlnia a robotkarnak, kisebb az esély arra, hogy egy másik bábut felborít.



- A hasáb 6 mm vastag, hogy a megfogó viszonylag nagy terhelését (min. 50 N) elviselje.
- Mindegyik bábu egyforma magas (40 mm), így a robotkarral felemelés programja és a képfeldolgozás is nagy mértékben leegyszerűsödik.
- A bábuk viszonylag alacsonyak és a talprészük vastagított, tömör, hogy kevésbé legyenek borulékonyak.
- Annak érdekében, hogy a bábukat kevesebb támaszanyag felhasználásával lehessen nyomtatni a bábuk tetején elhelyezkedő hasábok külön lettek kinyomtatva és csak utólag lettek a bábukra felragasztva.



**3.5. ábra:** A sakkbábúk 3D-s terve





3.6. ábra: A kinyomtatott sakkbábuk

### 3.3.2. Bábuk pozíciójának azonosítása

A bábuk pozíciójának azonosításához szükség van valamilyen kalibrációs eljárásra, hogy a képek egyes részeihez a sakktábla mezőit tudjuk rendelni. A mezők definiálásához kötődő módszer alapjait OpenCV-s függvények jelentik. A kidolgozott eljárás lényege, hogy a kalibrációs képen egy sakktáblaminta belső sarokpontjait (azon sarokpontok, amelyek 4 mezőnek együttes sarokpontjai) határozza meg a képen. A képkoordináták alapján történik az egyes mezők felosztása az élek mentén szomszédos pontok képzeletbeli összekötésével. Mivel a perspektivikus hatások elhanyagolhatók a tábla mezőre nézve, így 2 szemközti pontra illesztett téglalap jó közelítés és az egyes képek kivágásához. Fontos elem, hogy a kalibrációs sakkminta mezőszámai vízszintes és függőleges irányban különbözzenek, mert így fogja a sarokpontkereső függvény mindig ugyanabban a sorrendben visszaadni a pontok koordinátáját. A sorszámozás a minta bal felső részénél lévő, fekete mező sarkától indul és soronként halad (a hosszabb élek mentén) a 3.7 ábrán látható módon<sup>1</sup>. Az algoritmus az elmosódásokra, nagyobb szögekre (50-70°) és a papír gyűrődéseire nem érzékeny.

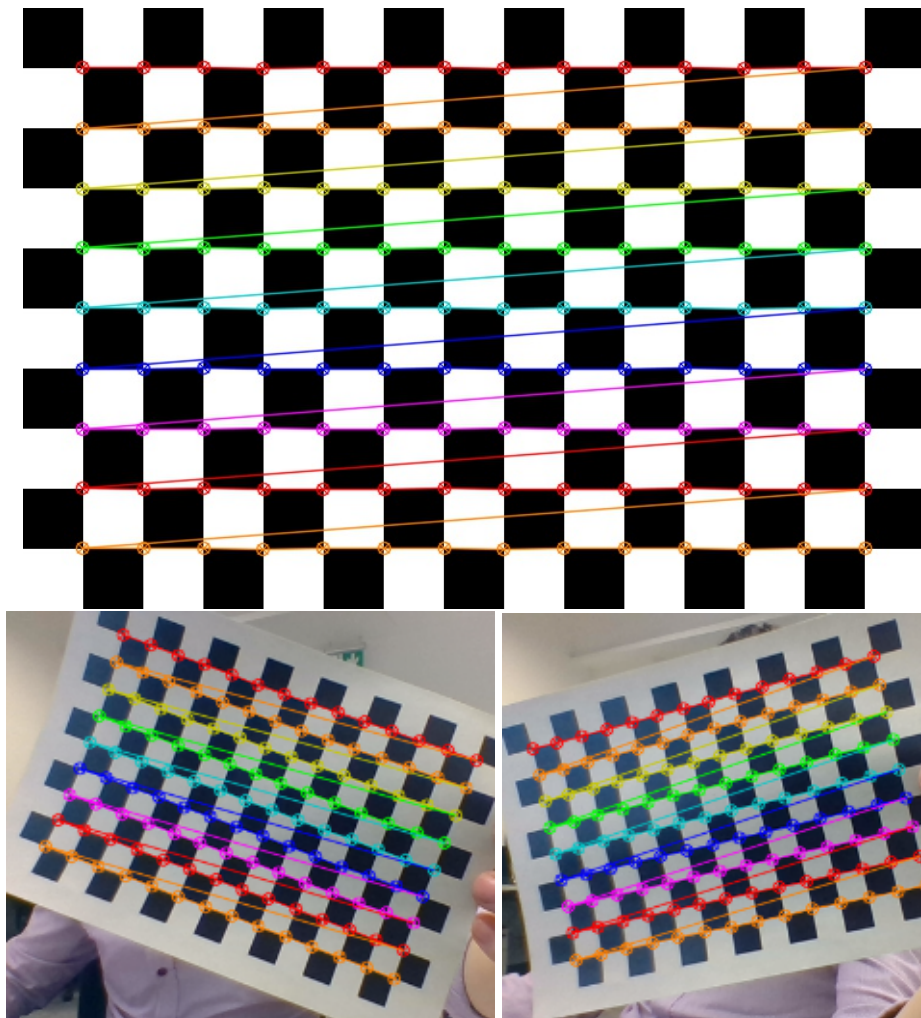
Fontos elem ezeken túl, hogy a kalibrációs lap abban a síkban helyezkedjen el, ahol a bábukon a zöld minta, így biztos megfelelő helyen keressük a bábukat. Ha az alsó tábla szintjére helyeznénk a kalibrációs lapot, akkor az egyes mezőkre helyezett bábukon lévő minták átlóghatnának a szomszédos mezőkre. Ezt mutatja be a 3.8

<sup>1</sup><http://stadatum.blogspot.com/2016/05/corner-identification-in-computer.html>

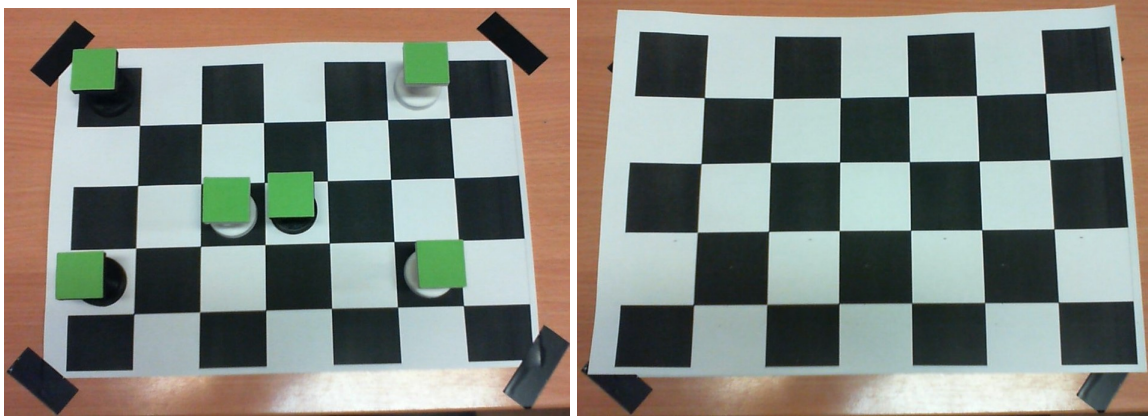
ábra. Ezek alapján a kalibrációs eljárás lépései:

1. Néhány sakkbábú táblára helyezése.
2. A bábuk teteje által kifeszített síkra egy sakktáblamintás papír ráhelyezése úgy, hogy az egyes mezők az alsó sakktábla mezői felett legyenek (mindegyik alsó sakktáblamező felett legyen egy belső sarokpontokkal határolt mező).
3. Kép készítése abból a pozícióból és szögből, ahova a robotkat a kamerával minden képkészítés előtt vissza fog térni.
4. A képen a sarokpontok keresése és a mezők meghatározása automatikusan történik a programban, de a sarokpontok számát vízszintes és függőleges irányban előre definiálni kell.

A képfeldolgozáshoz szükséges osztályokat az 'imgprocess' csomag tartalmazza (az OpenCV-s és a QR-kódhoz tartozó .jar fájlokat a Build path-hoz hozzá kell



3.7. ábra: A sakktáblaminta feltérképezett pontjai



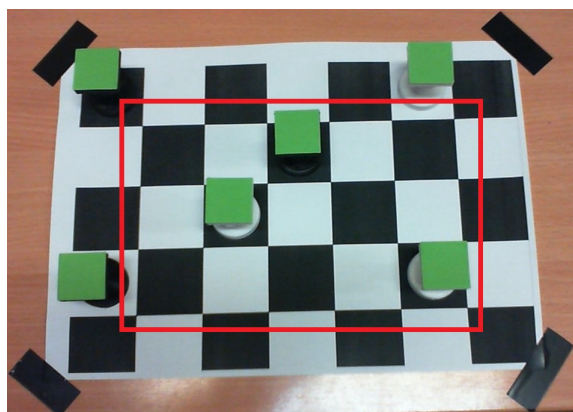
3.8. ábra: Kalibrációs kép felvétele

még adni). A sakkbábuk pozíciójának felismerését végző metódusok a `ScanPiecesPosition` osztályban vannak implementálva. Az osztályt példányosítani a (`BufferedImage`) kalibrációs képpel lehet, megadva még a horizontális és vertikális sarokpontok számát függvényparaméterként. A példányosítás hatására lefut a sarokpontkereső függvény. A `ScanPosition` függvényt erre a példányra meghívva visszakapjuk a fehér bábuk helyét egy boolean mátrix formájában, ahol a mátrix egyes elemei a tábla egyes mezői. A kép mezőkre darabolását a `CropImages` privát függvény végzi, amely a kalibráció során meghatározott sarokpontok alapján darabolja fel a képet: a kapott képek a mezők bal felső és jobb alsó pontjai által meghatározott téglalapon belüli részek.

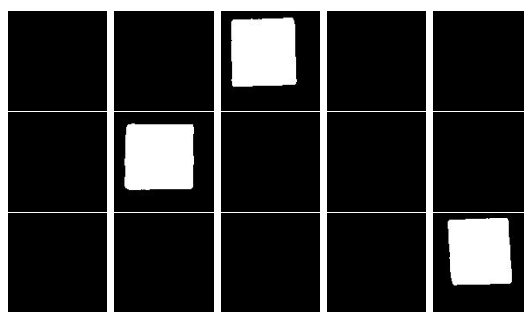
A kapott képekről a `FindGreen` függvény dönti el, hogy található-e rajtuk jelentősebb kiterjedésű zöld szín. A zöld színre szűrést RGB színtartományban vizsgálja a program (HSV színskálába transzformálással a szűrés még finomítható és kevésbé lesz érzékeny a különböző fényviszonyokra is). Az egyes képek minden pixele esetén kiszámol a program egy értéket a következő képlet alapján:

$$value = greenComponent - \frac{redComponent}{2} - \frac{blueComponent}{2}$$

Ha ez a 'value' érték egy megadott határértéknél nagyobb (jelenleg 40 egy 0-tól 255-ig terjedő skálán), akkor a pixel színét a szűrő algoritmus fehérre állítja, ellenkező esetben feketére. Utolsó lépésként a szűrő algoritmus minden mezőre külön-külön kiszámítja a fehér és a fekete pixelek számának az arányát. Ha ez az érték 10%-nál nagyobb egy mező esetén, akkor ott található bábu, azaz a hozzá tartozó boolean mátrix elem `true` értékű lesz. A visszaadott mátrixot a sakkprogram kapja majd meg, ami ez alapján és az előző állás alapján dönti el, hogy mi volt a lépés.



(a) A feldolgozandó kép



(b) A zöld színre szűrés eredményeként kapott képek

**3.9. ábra:** Zöld színre szűrés (a bal oldali képen a pirossal bekeretezett rész lett kalibrálva, azokat a mezőket vizsgálja)

## 3.4. QR-kód alapú képfeldolgozás - továbbfejlesztés

Ahhoz, hogy sakkjátékot tetszőleges állapotból lehessen kezdeni, illetve folytatni szükség van a bábuk helyének felismerésén túl azok típusának egyértelmű felismerésére. Erre alkalmas megoldás lehet a bábuk tetejére helyezett QR-kód, amelyben kódolva van a bábu típusa. A QR-kódok kezelésére jó választás a nyílt forráskódú ZXing („Zebra Crossing”) program<sup>1</sup>. Ez a Java könyvtár (library) alkalmas különböző formátumú, egy- és kétdimenziós vonalkódokkal kapcsolatos képfeldolgozásra, amelynek csak egyik eleme a QR-kód olvasás és generálás.

### 3.4.1. A forráskód build-je

A könyvtár egyszerű használatához célszerű .jar fájlokat generálni a forráskódból. Az első lépés a Github-on elérhető forráskód letöltése vagy klónozása a saját számítógépre. A programkód számos mappába és almappába van rendszerezve az egyes moduloknak megfelelően (pl.: core/ és javase/). **Fontos:** a mapparendszert olyan helyre tegyük a számítógépen, melynek elérési útjában nem található szóköz karakter (ékezetes karakter is probléma lehet)! Minden Java alapú modul esetén található egy pom.xml fájl, amit Apache Maven<sup>2</sup> segítségével lehet használni.

Szükségünk van megfelelő java verzió telepítésére. A JRE (Java Runtime Environment) helyett a JDK (Java Development Kit) valamelyik verzióját (a projekt jdk 10.0.2 verziót használ) érdemes telepíteni, ha fejlesztői funkciókat is igénybe szeretnénk venni (a JRE csomagot ez már tartalmazza). Az Apache Maven telepítése után szükségünk van az ehhez és a JDK-hoz tartozó környezeti változók be-

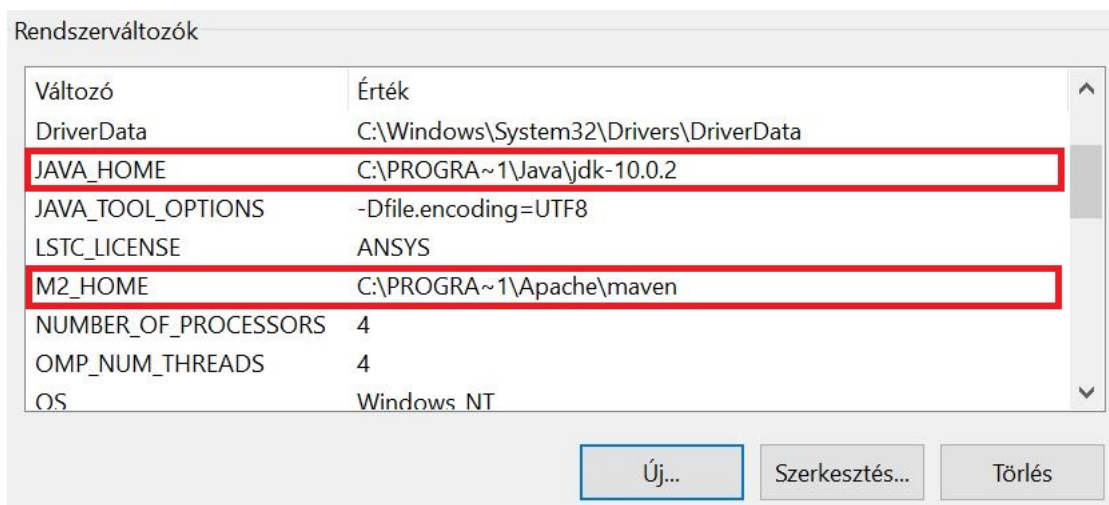
<sup>1</sup>További információk és forráskód: <https://github.com/zxing/zxing>

<sup>2</sup>Az Apache Maven program ingyenesen letölthető: <https://maven.apache.org/>



állítására. Ezt a Vezérlőpult->Rendszer->Speciális rendszerbeállítások->Környezeti változók...->Rendszerváltozók címszó alatt tehetjük meg. Szükségünk van egy 'JAVA\_HOME' és egy 'M2\_HOME' változóra (3.10. ábra).

Parancssorban navigáljunk a ZXing projekt gyökeréhez és futtassuk a 'mvn install' parancsot a fordításhoz, a tesztekhez és az összes modul felépítéséhez. A „-DskipTests” paraméter hozzáadásával a unit tesztek kihagyhatjuk. Szükség lehet a '-Drat.ignoreErrors=true' paraméterre a licensz tesztekkel kapcsolatos problémák ignorálásához. A build folyamat akkor mondható sikeresnek, ha mindegyik modul felépítése sikeres ('ANDROID\_HOME' környezeti változó beállítása nélkül az Androidhoz kapcsolódó modulokat nem build-eli) (3.11. ábra).



3.10. ábra: Szükséges környezeti változók beállítása

```
[INFO] Reactor Summary:
[INFO]
[INFO] ZXing 3.3.4-SNAPSHOT ..... SUCCESS
[INFO] ZXing Core ..... SUCCESS
[INFO] ZXing Java SE extensions ..... SUCCESS
[INFO] ZXing zxing.org web app 3.3.4-SNAPSHOT ..... SUCCESS
[INFO] -----
[INFO] BUILD SUCCESS
```

3.11. ábra: Sikeres build végeredménye

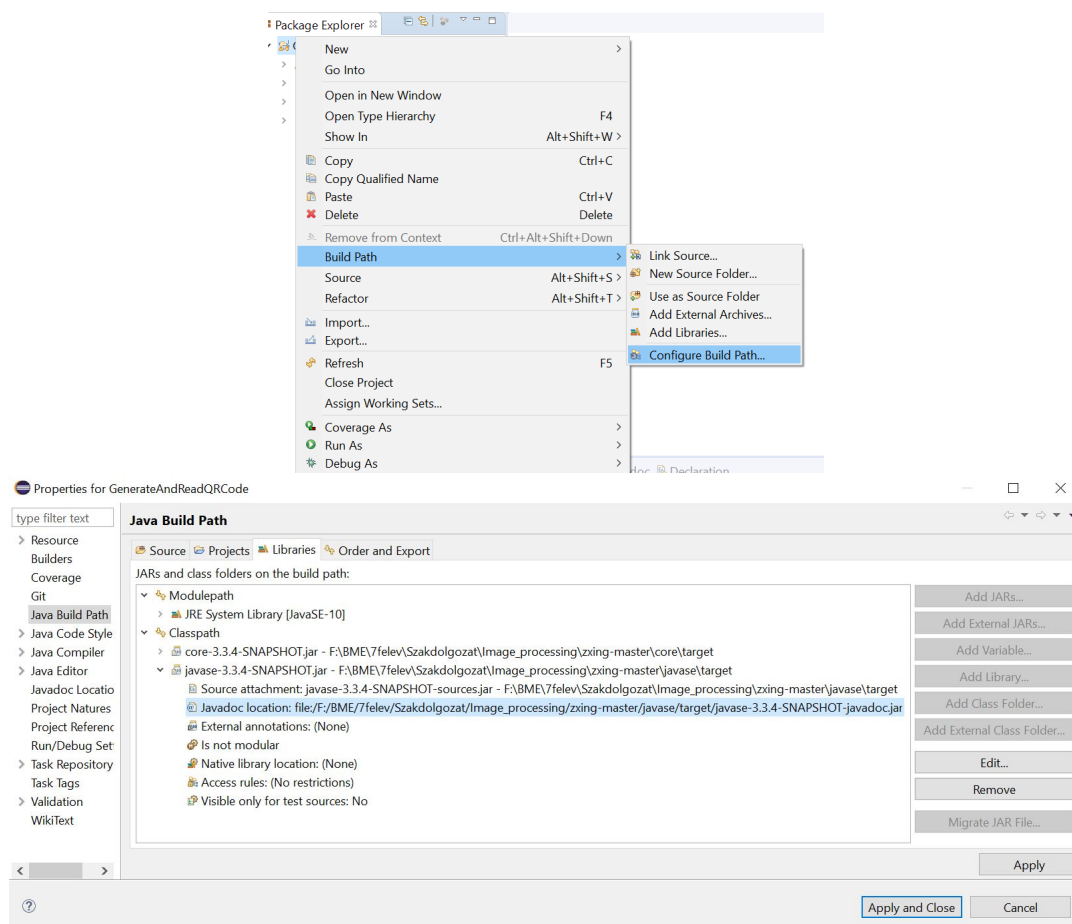
A lefordított .jar fájlokat ezt követően az egyes modulokon belül találjuk. Például a lefordított *core/* kód helye a *core/target/core-x.y.z.jar*. Ezeket lehet beimportálni a képfeldolgozást megvalósító projektbe.

### 3.4.2. ZXing könyvtár beimportálása és használata

Az i18n robotkart programozni Sunrise Workbench használatával a legegyszerűbb, ami egy JAVA Eclipse platformú szoftver. Emiatt praktikus okokból a szakdolgo-

zat képfeldolgozási és sakkalgoritmus beágyazási része túlnyomó részt Eclipse-ben történt (verzió: 4.9.0)<sup>1</sup>.

A könyvtár beimportálásához létrehoztam egy java projektet. Erre jobb egérgombbal kattintva elnavigáltam a 'Configure Build Path...' menüponthoz (3.12. ábra). A 'Classpath'-t kiválasztva jobb oldalt aktivizálódik az 'Add External Jars...' gomb. Kiválasztottam a *core/* és a *javase/* modulok .jar fájljait. Ezeken belül lehetőség van forráskód (Source attachment) és dokumentáció (Javadoc location) csatolására (3.12. ábra). 'Apply and Close' után megjelennek a csomagok a hivatkozott könyvtárak (Referenced libraries) pont alatt.



3.12. ábra: ZXing hozzáadása a hivatkozott könyvtárakhoz

### 3.4.3. QR kód generálás és olvasás

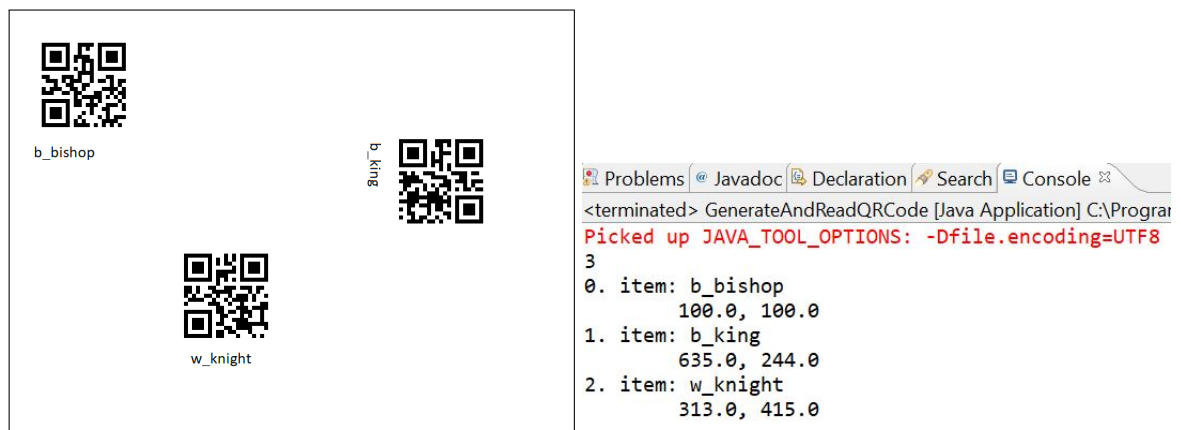
A ZXing könyvtárt felhasználó, ehhez a projekthez szükséges elemek az 'imgprocess' csomagban találhatóak a 'QRCodeMethods.java' fájlban. A függvények leírása a 3.1 táblázatban találhatóak<sup>2</sup>.

<sup>1</sup>Link: <https://www.eclipse.org/>

<sup>2</sup>Forráskód: <https://github.com/rolandvarga601/Szakdolgozat>

Függvény neve	Paraméterek	Leírás
createQRCode	<ul style="list-style-type: none"> <li>String qrCodeData - a kódolt szöveg</li> <li>String filePath - a képek mentési helye</li> <li>String charset - karakterkódolás</li> <li>int qrCodeheight - a kép magassága</li> <li>int qrCodewidth - a kép szélessége</li> </ul>	Nincs visszatérési értéke, a QR-kódot tartalmazó képet a megadott elérési helyre menti.
readQRCode	BufferedImage image - a kép amelyen a QR-kód található	Visszaadja a képen található QR-kódba kódolt szöveget.
readMultiQRCode	String FileName - a kép elérési útja, amelyen a QR-kódok találhatóak	Képes egyazon képen különböző szögekben elhelyezkedő QR-kódok felismerésére és dekódolására (3.13 ábra). Visszatérési érték: Result[], amely tömb elemei tartalmazzák többek között az egyes QR-kódok helyét a képeken és a kódolt szöveget.

3.1. táblázat: QRCodeMethods.java függvényei



3.13. ábra: Példa a több QR-kódot tartalmazó kép kiolvasására

## 4. ROBOTKAR KALIBRÁCIÓ, REFERENCIA FELVÉTEL KIDOLGOZÁSA



## 5. ROBOTKAR MOZGÁSÁNAK DEFINIÁLÁSA ÉS PROGRAMOZÁSA

## 6. SAKKALGORITMUS BEÁGYAZÁSA

## 7. EREDMÉNYEK ÉRTÉKELÉSE

### HIVATKOZÁSOK

- [1] Andreja Rojko. Industry 4.0 concept: Background and overview. 2017.
- [2] Thomas Bauernhansl, Jörg Krüger, Gunther Reinhart, and Günther Schuh. Wgp-standpunkt industrie 4.0. Technical report, Wissenschaftliche Gesellschaft für Produktionstechnik, 2016.
- [3] T. Vilarinho, B. A. Farshchian, J. Floch, and B. M. Mathisen. *A Communication Framework for the Internet of People and Things Based on the Concept of Activity Feeds in Social Computing*. July 2013.
- [4] F. J. N. d. Santos and S. G. Villalonga. Exploiting local clouds in the internet of everything environment. In *2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, March 2015.
- [5] C. Wesch-Potente G. Schuh, T. Potente and Annika Hauptvogel. Sustainable increase of overhead productivity due to cyber-physical systems. In *Proceedings of the 11th Global Conference on Sustainable Manufacturing*, March 2013.
- [6] Henning Kagermann. Change through digitization : value creation in the age of industry 4.0. 2015.
- [7] J. Miranda, N. Mäkitalo, J. Garcia-Alonso, J. Berrocal, T. Mikkonen, C. Canal, and J. M. Murillo. From the internet of things to the internet of people. *IEEE Internet Computing*, 19(2):40–47, Mar 2015.
- [8] M. Awais and D. Henrich. Human-robot interaction in an unknown human intention scenario. In *2013 11th International Conference on Frontiers of Information Technology*, pages 89–94, Dec 2013.
- [9] S. Jaschke. Mobile learning applications for technical vocational and engineering education: The use of competence snippets in laboratory courses and industry 4.0. In *2014 International Conference on Interactive Collaborative Learning (ICL)*, pages 605–608, Dec 2014.
- [10] George Michalos, S Makris, Panagiota Tsarouchi, Toni Guasch, Dimitris Kontovrakis, and George Chryssolouris. Design considerations for safe human-robot collaborative workplaces. *Procedia CIRP*, 37:248–253, 12 2015.

- [11] Olessia Ogorodnikova. Human weaknesses and strengths in collaboration with robots. *Periodica Polytechnica Mechanical Engineering*, 52(1):25–33.
- [12] KUKA Deutschland GmbH. *System Software - KUKA Sunrise.OS 1.16 - Operating Instructions for End Users*, 2018.
- [13] SCHUNK GmbH & Co. KG. *Electrical parallel gripper MEG 50 EC Assembly and Operational Manual*, 3 2016. Rev. 02.01 EN.

**HRC** ..... Human Robot Collaboration

**IoT** ..... Internet of Things

**IoE** ..... Internet of Everything

**TCP** ..... Tool Center Point

**HRI** ..... Human-Robot Interaction