

# *KUKA KR5 ipari hegesztő robotkar megfogó rendszerének újratervezése 3D nyomtató technológia felhasználásával Arduino segítségével*

Varga Viktor  
Mechatronikai Tanszék  
Debreceni Egyetem,  
Műszaki Kar  
Debrecen, Magyarország  
viktor.varga6@gmail.com

Erdei Timotei István  
Mechatronikai Tanszék  
Debreceni Egyetem,  
Műszaki Kar  
Debrecen, Magyarország  
timoteierdei@eng.unideb.hu

Dr. habil. Husi Géza  
Mechatronikai Tanszék  
Debreceni Egyetem,  
Műszaki Kar  
Debrecen, Magyarország  
husigeza@eng.unideb.hu

**Absztrakt**—Ezen összefoglaló során egy 3 ujjú megfogó megalkotásának lépései követhetők végig a tervezési fázisoktól kezdve, a megfogó 3D modellezésén, 3D nyomtatásán, programozásán és a Debreceni Egyetem Műszaki Karán található KUKA KR5-ös robotra való felszerelésén keresztül, egészen a robotkarral való munkavégzési feladatok elvégzéséig. Továbbá a megfogott munkadarab átmérőjének számítása, egy ujj által kifejtett erő szenzorral való mérése és a megfogó Arduino alapú vezérlése is bemutatásra kerül.

**Kulcsszavak**— megfogó; 3D nyomtatás; Arduino; robot; FSR szenzor

## I. BEVEZETŐ

Robottechnika óra keretein belül nyílt lehetőség a robotlaboratóriumban található, KUKA KR5 típusú robotkar alapszintű programozásának elsajátítására, illetve felépítésének, felszereltségének megismerésére. Jelenleg egy pneumatikusan vezérelt 2 ujjú megfogó látja el a robot megfogási feladatait. Ezen megfogó 30 és 45 [mm] közötti intervallumban képes megfogásra, és jelenleg egy oktatási célra használt szerszám állandó befogására szolgál. Ez a megoldás az oktatási feladatokat ellátja, viszont kis munkaterülete, illetve kétállapotúsága (nyitott vagy zárt) miatt csak kis tárgyak kezelésére alkalmas, amelyek továbbá nem törekenyek és bírják a rájuk ható erőt. Mivel több megfogó nem áll az egyetem rendelkezésére, a robotkar munkakörének és az oktatási lehetőségek palettájának bővítése céljából alakult ki egy új megfogó szerkezet kialakításának ötlete. A választás egy 3 ujjú párhuzamos mozgású megfogóra esett. A 3D nyomtatók térnyerésének köszönhetően a tervezett 3D modellek fizikai kivitelezése is megtörtént. A vezérlés Bluetooth modul és Android program segítségével került kivitelezésre Arduino felhasználásával. Az alkalmazott szervomotor felépítését kihasználva megfogó ujjainak pozíciói folyamatos vizsgálat alá kerültek, így megállapíthatóvá váltak a megfogott munkadarabok átmérői a megfogási tartományon belül. A megfogó szorítóerejének vizsgálatára egy FSR (Force Sensitive Resistor) erőmérő szenzor került felszerelésre az egyik ujjra, amely megközelítő értéket ad a megfogó szorítóerejéről. A megfogó robotra való integrálást követően

különböző átmérőjű munkadarabok megfogására és áthelyezésére került sor, ezzel a munkavégzési feladatok teljesülésének igényét kielégítve.

## II. MECHANIZMUS BEMUTATÁSA

A 3D modellezés megkezdése előtt, első lépésként, elengedhetetlen volt meghatározni, hogy milyen mechanizmussal legyenek mozgatva a megfogó ujjai. Az egymástól 120°-ra elhelyezett ujjak mozgatásához az alább látható, úgynevezett forgattyús mechanizmus jelentett megoldást [1].

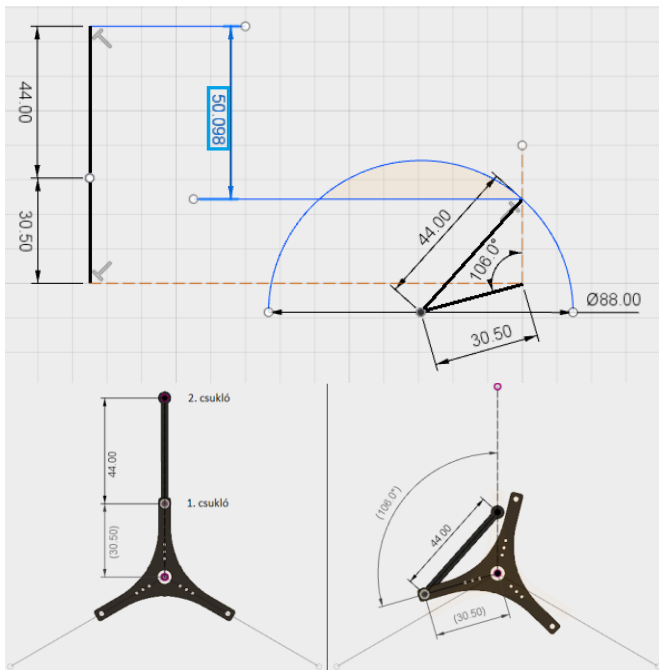


1. ábra: Mechanizmus bemutatása

Ezen mechanizmus segítségével a motor által végzett körmozgás lineáris mozgássá alakítható át. Ilyen mechanizmust használnak többek között a gőzgépeknél, dugattyús motoroknál, dugattyús szivattyúknál és kompresszoroknál is.

Következő lépésként meg kellett vizsgálni, hogy mi lehet a motor legnagyobb szögelfordulása, ami ütközés nélküli, biztonságos működést biztosít. Az így kapott 106°-os érték mellé meghatározásra került, hogy egy ujj maximális mozgási tartománya 50 [mm] legyen, ami egy kellően nagy megfogási tartományt biztosít. Ezen adatok ismeretében egy Fusion 360-ban rajzolt vázlat segítségével (2. ábra) megállapítható a motor tengelyétől az első csukló középpontjáig való távolság, illetve az első és második csukló között elhelyezkedő elem szükséges mérete [2]. Az ábra bal oldalán a motor 0°-os állása látható, tehát a megfogó teljesen nyitott állapota, a jobb oldalán a

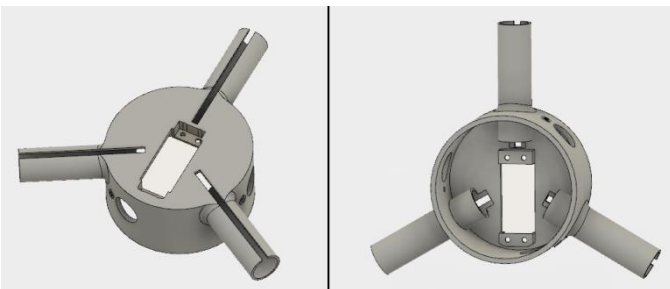
106°-os elfordulás hatására elért, megközelítőleg 50 [mm]-es összehúzódás.



2. ábra: Szükséges méretek meghatározása

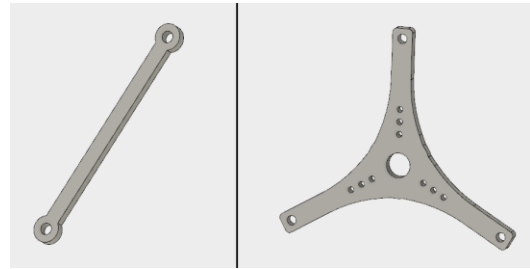
### III. 3D MODELL BEMUTATÁSA

Az előző részben megszabott méretek ismeretében megkezdődhetett az alkatrészek 3D modelljeinek felépítése, amikből felépül a megfogó. A tervezés Autodesk Fusion 360 tervező program használatával valósult meg. A következő ábrákon a megfogót felépítő különálló alkatrészek bemutatása következik. A 3. ábrán a megfogó váza látható, amelyhez a motor kerül rögzítésre, és amelyen az egymástól 120° fokra elhelyezett, a 3 ujj mozgásához szükséges hengeres csúszkák találhatók.



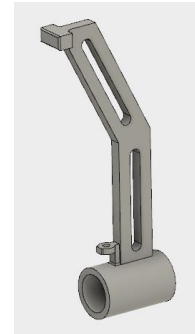
3. ábra: Tartóváz

A következő két alkatrész az előzőekben bemutatott mechanizmus működéséért felelős 44 [mm] nagyságú toló-/húzórud (bal oldali kép) és a motorra felszerelhető szögelfordulást végző alkatrész (jobb oldali kép).



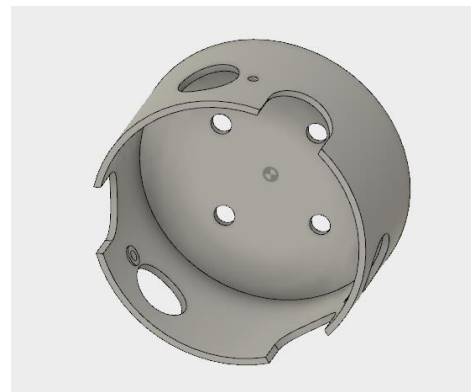
4. ábra: Toló-/húzórud és motor alkatrész

Az 5. ábrán látható a megfogást közvetlenül végző ujj, amely úgy van kialakítva, hogy a megfogó vázában akadálymentesen mozoghasson/csúszhasson.



5. ábra: Ujj

Megoldandó feladatok közé tartozott a KUKA KR5-ös robotkar és a megfogó összeszerelhetőségének megoldása is. A robotkar 6. tengelyének végén 4 furat lett kialakítva, lehetőséget kínálva ezzel szerszámok felszerelésére. A furatok méretének és elhelyezkedésének ismeretében tervezésre került egy adapter (6. ábra), amelyet a robothoz erősítve lehetőséget ad a megfogó hozzászereléséhez is.



6. ábra: Adapter

### IV. ALKATRÉSZEK ÖSSZESZERELÉSE FUSION 360-BAN

Összeszerelés első lépéseként a motortartó váz és a TowerPro MG995 típusú szervomotor összekapcsolása történt meg, majd a motorhoz tartozó alkatrész rögzítése következett. A választott motor pozícionálhatóságával és kis méretéhez képesti nagy erő kifejtésével tökéletesen ellátja a kívánt feladatokat. A következő táblázat a motor néhány fontosabb tulajdonságát mutatja be.

1. táblázat: Szervomotor adatok [3]

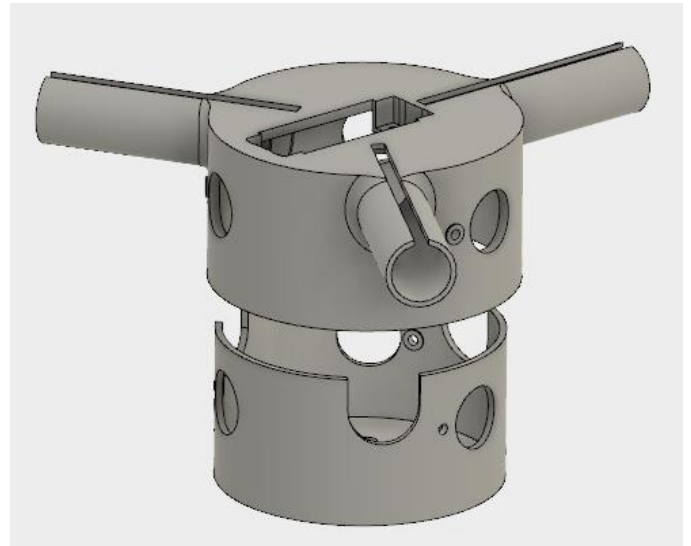
	Forgatónyomaték [Nm]	Gyorsaság [s/60°]	Tömeg [g]
TowerPro MG995	1	0,16	55

Következő lépésként a tartóváza oldalról becsúztatott ujjak toló-/húzóruddal való összeszerelésére került sor, melynek másik vége a forgómozgást végző alkatrészhez kapcsolódik. Ezennel elkészült az összeszerelt megfogó 3D modellje (7. ábra).



7. ábra: Összeszerelt 3D modell

A megfogási tartomány a megfogó teljesen nyitott és teljesen zárt pozíció közti tartomány, amely 9 [mm] és 109 [mm] közötti átmérőjű tárgyak megfogását teszi lehetővé. A megfogó és az adapter, amely a robotkarral való összeszerelést teszi lehetővé, összezsúszatható és a kialakított furatokon keresztül csavarokkal összeerősíthető (8. ábra).



8. ábra: Adapter összeillesztése a megfogóval

## V. ALKATRÉSZEK 3D NYOMTATÁSA

A nyomtatás egy magánszemélytől került megrendelésre, akivel egyeztetésre kerültek a fontosabb nyomtatási paraméterek. Általánosságban az ABS rugalmasabb, hőállóbb, keményebb, illetve törésre kevésbé hajlamos anyag, mint a PLA, de mivel a tervezett megfogónak nem szükséges a magas hőállóság és nem lesznek nagy tömegek mozgatva vele, így PLA anyagból való nyomtatás (9. ábra) is kifogástalan működést biztosít. Könnyebb nyomtathatósága és széles színválasztéka miatt designerek és hobbi 3D nyomtatók körében közkedvelt [4].



9. ábra: PLA anyagból készült 3D nyomtatott alkatrészek

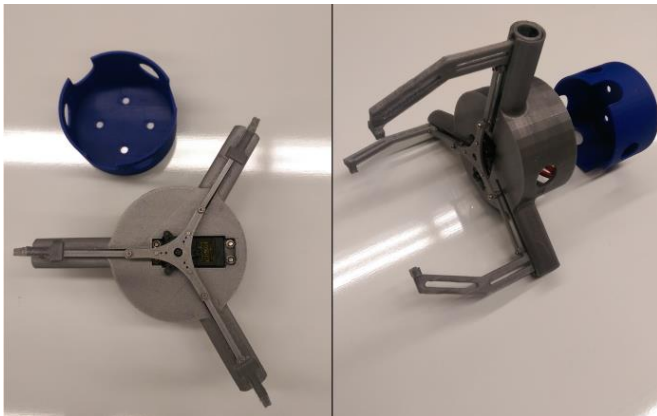
Az Ultimaker Cura program, amelybe az alkatrészeket egyenként be lehet illeszteni, és amelyben a nyomtatási paraméterek beállításra kerülnek, megadja a következő fontos információkat a nyomtatásról, amiket az alábbi táblázat foglal össze a 9. ábra alkatrész számozása alapján [5].

2. táblázat: Nyomtatási adatok alkatrészenként

Alkatrész	Kitöltési	Nyomtatási	Tömeg	Anyagár
-----------	-----------	------------	-------	---------

	tényező (%)	idő	(g)	(Ft)
1	30	2ó 48p	31,1	165
2	30	6ó 54p	82,1	435,1
3 (3db)	30	2h 54p	23,7	126,6
4	30	10p	1,6	8,5
5 (3 db)	30	6p	1,2	6,3
$\Sigma$	-	12h 52p	139,7	741,5

A megfogó és a szervomotor együttes tömege csavarok nélkül 194,7 [g], amely a robotkar 5 [kg]-os terhelhetőségének mindössze a 3,89 [%]-át teszi ki. A kinyomatott elemek a 3D modell összeszerelési lépéseit követve a valóságban is összeszerelésre kerültek. Az ujjak megfogási felületének jobb tapadása érdekében gumirétegek felhelyezésére került sor.



10. ábra: Összeszerelt megfogó

## VI. MEGFOGÓ ARDUINO ALAPÚ VEZÉRLÉSE

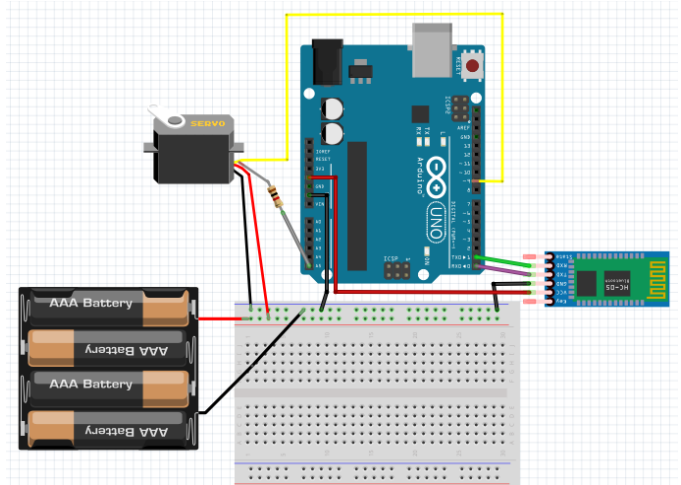
A szervomotor működésénél fogva, ha elfordulás közben akadályba ütközik mielőtt elérte a programban megadott szöveget, továbbra is folyamatosan az adott pontba akar fordulni, mivel a benne található potenciométer még nem érte el a pozícióját, ami a szöghöz tartozik [6]. Ezen jelenség bekövetkezésénél a szervomotor felforrósodhat, és ezáltal meghibásodik. A problémát levetítve a feladatra megállapítható, hogy a megfogó egyszerű  $0^\circ$ -tól  $106^\circ$ -ig való összezárása különböző méretű alkatrészek megfogásához nem biztonságos a motor élettartamát tekintve, mivel ha félúton megakadnak az ujjak a megfogandó alkatrészben, a motor továbbra is a  $106^\circ$ -os végső helyzetét akarja elérni, ezzel károsítva azt. Kitűzött célá vált tehát elérni azt, hogy ha a megfogási területen belül bármekkora átmérőjű tárgy kerül megfogásra, a motor a megszorulását követően álljon be abba a pozícióba ahol megakadt.

Ezt a szervomotorban található potenciométer vizsgálatával sikerült megvalósítani. A motor szétszerelése után a potenciométer signal lábára forrasztott vezetékkel, amely az Arduino analóg bemenetére lett kötve, és a megfelelő programsor segítségével vizsgálhatóvá vált a potenciométer, és egyben a motor állása 0 és 1024 értékek közt. Ekkor a motort  $0^\circ$  és  $106^\circ$ -os véghelyzetekbe küldve megállapítható azok 0 és 1024 közötti értékei. „Map” programparancs segítségével az így kapott  $0^\circ$ -nak megfelelő 122-es érték  $20^\circ$ -

ra lett átalakítva, illetve a  $106^\circ$ -nak megfelelő 302-es érték  $126^\circ$ -ra (12. ábra). Ezen értékek egy *val* nevű változóba lettek elmentve. A *val* változó értéke tehát a motor bármely állásánál plusz 20-as értéket fog mutatni, aminek a későbbiek fontos szerepe lesz a programban.

A program (12. ábra) lényege, hogy  $0^\circ$ -tól kezdve  $106^\circ$ -ig egy for ciklus *x* változójának segítségével minden egyes fokba beállításra kerül a motor. Az előzőekben leírtak eredményeképp  $0^\circ$ -nál a motor potenciométerének *val* változóba mentett értéke 20. Ahogy akadálymentesen fordul a motor, úgy arányosan változik a *val* értéke is. Azonban ha a motor elakad egy útjába kerülő alkatrészben, az azt jelenti, hogy a *val* érték a potenciométer elakadása miatt nem változik. Eközben a for ciklus *x* értéke, ami a motor forgását biztosítja, folyamatosan növekszik, és így a 20 fokos eltérés az *x* és *val* értékek között elkezd csökkenni egészen addig, amíg az *x* egyenlő nem lesz a *val* értékkel. Amikor ez az állapot bekövetkezik, a for ciklus megáll és a motor *x*-17 fokra áll be. A kivonás azért szükséges, mert az *x* érték jelenleg  $20^\circ$ -al nagyobb, mint a tényleges állása a motornak. Az ujjak kismértékű mozgása miatt  $20^\circ$  kivonása helyett csak  $17^\circ$  kivonására került sor a szükséges szorítás elérése érdekében.

A megfogó vezérlését egy HC-06 típusú Bluetooth modul, egy Arduino és egy ingyen letölthető Android alkalmazás szolgáltatja [7]. Az Android alkalmazás Android típusú okostelefonon fut, amelyről két érték kiküldése lehetséges a Bluetooth modulnak, ami soros porton keresztül továbbítja a jelet az Arduino számára [8]. Első lenyomásra egy „+”, második lenyomásra egy „-” jelet küld az Arduino-nak. A „+” jel hatására elindul az ujjak összehúzódása, azaz a megfogó mechanizmus, „-” jelle pedig egyszerűen visszaáll a megfogó teljesen nyitott állapotba. A kapcsolási rajz Fritzing program segítségével került megrajzolásra (11. ábra) [9].



11. ábra: Kapcsolási rajz

A motor tápellátását egy külső tápegység biztosítja mivel az Arduino nem képes kellő nagyságú áramerősséget biztosítani. A szervomotor potenciométerének stabilizálásához és vizsgálatához szükséges volt egy 1 [k $\Omega$ ]-os ellenállás bekötése is, máskülönben a vizsgált érték nagymértékben ugrált különböző értékek között. Továbbá a kívánt működés

eléréséhez a motort, tápegységet, Arduinot és Bluetooth modult közös földre kellett kötni.

```
void loop() {
  if (Serial.available()) { // van-e soros kommunikáció
    bluetooth = Serial.read(); // bluetooth jel beolvasása
    Serial.println(blueooth); // bluetooth jel kiírása

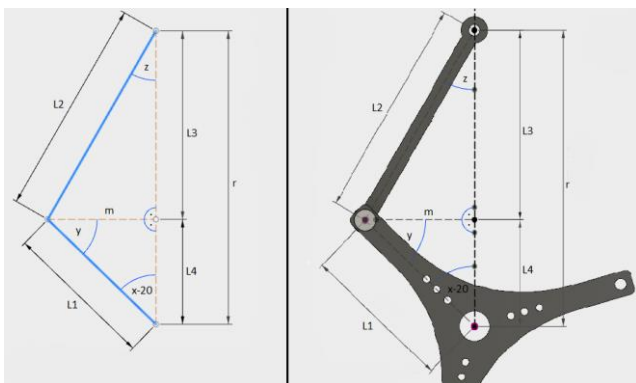
    if (bluetooth == '+'){ // ha bluetooth jel értéke '+'
      for (int x=0 ; x < 107; x++){
        myservo.write(x); // servo 0°-tól való mozgatása 106°-ig
        int val = analogRead(A5); // servo potméterének vizsgálata
        val = map(val, 122, 302, 20, 126); // val változó átváltása 12° és 118°-ra
        Serial.print(x); // x kiírása
        Serial.print(" ");
        Serial.println(val); //val kiírása

        if (x == val) // ha elakad a servo, az x értéke utoléri a val értékét
        {
          myservo.write(x-17); //servo megállítása az akadási pontnál
          Serial.println(x-17);
          break;
        }
        delay(35);
      }
    }
    else { // ha bluetooth jel értéke '-'
      myservo.write(0); //servo 0°-ra áll, kinyit a megfogó
    }
  }
}
```

12. ábra: Megfogó Arduino programja

## VII. MEGFOGOTT TÁRGY ÁTMÉRŐJÉNEK SZÁMÍTÁSA

A szervomotor megakadt állapotában való megállításával, és ezen pozíciójának vizsgálatával minden megfogási műveletet követően kiszámolható a megfogott tárgy átmérője. A 12. ábrán bemutatott programból, a motor szögállását jelentő  $x$  változó további felhasználásával lehetséges a megfogott tárgy átmérőjének megállapításához szükséges számolás elvégzése. A számolás menete paraméteresen kerül bemutatásra. Az alábbi ábrán láthatóak a számolás képleteiben használt jelölések. A szemléletesség érdekében a jelölések vázlatosan (bal oldali ábra), és vizuálisan (jobb oldali ábra) is fel lettek tüntetve.



13. ábra: Számításhoz szükséges jelölések

$$\sin(x - 20) = \frac{m}{L1} \quad (1.)$$

Az előző részben leírtak alapján az  $x$  értéke a motor megakadásának pillanatában  $20^\circ$ -al több mint a tényleges, ezért került kivonásra ez az érték.

$$m = \sin(x - 20) * L1 \quad (2.)$$

$$\sin(z) = \frac{m}{L2} \quad (3.)$$

$$z = \arcsin\left(\frac{m}{L2}\right) \quad (4.)$$

$$L3 = \cos(z) * L2 \quad (5.)$$

$$L4 = \cos(x - 20) * L1 \quad (6.)$$

$$r = L3 + L4 \quad (7.)$$

$$d = (L3 + L4 - 20) * 2 \quad (8.)$$

$L1$  és  $L2$  állandó értékek.

A 8. képletben látható kivonás az ujjak érintkező felületének  $20$  [mm]-es eltolását kompenzálja a számolt értékhez képest. A 2-vel való szorzás a sugár értékéből a megfogott munkadarab átmérőjének méretét eredményezi [14]. A használt jelzések deklarálását követően a bemutatott számolás programba való implementálására került sor. Az alább látható Arduino programba bevitt számolás helyenként összevont formában került megírásra a helytakarékoság érdekében.

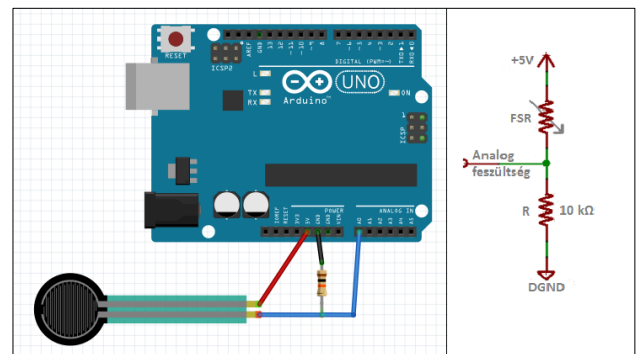
```
m = sin(radians (x-20))*L1;
w = m/L2;
z = asin(w);
d = (((cos(radians(x-20)) * L1) + (cos(z) * L2)) - 20) * 2;
Serial.print("A megfogott tárgy atmeroje = ");
Serial.println(d);
```

14. ábra: Megfogott munkadarab átmérő számolása

A soros monitorra kiírt eredmény a 17. ábrán látható bekeretezve.

## VIII. SZORÍTÓERŐ MEGHATÁROZÁSA FSR 400 ERŐMÉRŐ SENZOR HASZNÁLATÁVAL

A megfogó egy ujjának szorítóereje egy FSR 400 (Force Sensitive Resistor) típusú erőmérő szenzor használatával került meghatározásra. Ez tulajdonképpen egy nyomás hatására értékét változtató ellenállás, amely  $0,2$  [N] és  $20$  [N] közötti erők mérésére alkalmas [10]. A szenzorral való erőmérés egy egyszerű áramkör kialakításával elvégezhető.



15. ábra: Kapcsolási és sematikus kapcsolási rajz [11]

A szenzor egyik lába  $5$  [V]-os tápfeszültségbe, a másikat egy lehúzó ellenálláson keresztül a földre lett kötve. Továbbá az állandó lehúzó ellenállás és változó értékű FSR ellenállás közé az Arduino analóg bemenete kerül csatlakozásra a szenzor értékének vizsgálata érdekében [15]. A bekapcsolási küszöbértéket meghaladó erő hatására az FSR ellenállás értéke körülbelül  $100$  [k $\Omega$ ]. Ahogy további nyomás alá kerül a szenzor, az FSR ellenállás csökken, így az FSR és a lehúzó

ellenállás együttes összege 100 [kΩ]-ról 10 [kΩ]-ra tud lecsökkenni. Ez azt jelenti, hogy az áramerősség mindkét ellenálláson növekszik, ami az állandó 10 [kΩ]-os ellenálláson áthaladó feszültség növekedését eredményezi [11]. Az áramkör kiépítése után az adatok kinyeréséért felelős program bemutatása következik.

```
int fsrAnalog; int fsrVolt; unsigned long fsrEllenallas;

void setup() {
  Serial.begin(9600);
}

void loop() {
  fsrAnalog = analogRead(A0);
  Serial.print("Analog ertekek = ");
  Serial.println(fsrAnalog);

  fsrVolt = map(fsrAnalog, 0, 1023, 0, 5000); //fsrAnalog ertekek (0 - 1023) atvaltja [mV]-ra (0 - 5000)
  Serial.print("fsrVolt ertekek [mV]-ban = ");
  Serial.println(fsrVolt);

  if (fsrVolt == 0) {
    Serial.println("Nincs nyomás");
  }
  else {
    // FSR = ((Vcc - V) * R) / V      R = 10 [kΩ], Vcc = [5V]
    fsrEllenallas = 5000 - fsrVolt; // VCC = 5 [V] = 5000 [mV]
    fsrEllenallas *= 10000; // 10 [kΩ] ellenallas
    fsrEllenallas /= fsrVolt;
    Serial.print("FSR ellenallas [Ω]-ban = ");
    Serial.println(fsrEllenallas);
  }
  Serial.println("-----");
  delay(1000);
}
```

16. ábra: FSR 400 erőmérő Arduino program

Az első sorban a változók deklarálása történt. A void loop-  
on belül először a szenzor 0-1023-ig terjedő analóg értéke kerül beolvasásra és kiíratásra a soros monitorra. A szenzor nyomásának erősségével változik az analóg érték, ami felfogható a szenzoron áthaladó 0 [V]-tól 5 [V]-ig terjedő értéknek, ezért az analóg érték a map programsor segítségével átváltásra kerül 0 [mV]-tól 5000 [mV]-ra. A későbbi számolás érdekében lett a [V] érték [mV]-ba váltva. Ezt követően a szenzoron áthaladó feszültség nagysága is kiíratásra került a soros monitorra. Nyomás hiányában „Nincs nyomás” üzenet látható a monitoron. A lehúzó ellenálláson áthaladó feszültség értékét a következő feszültségosztó egyenlettel lehet kiszámolni [11].

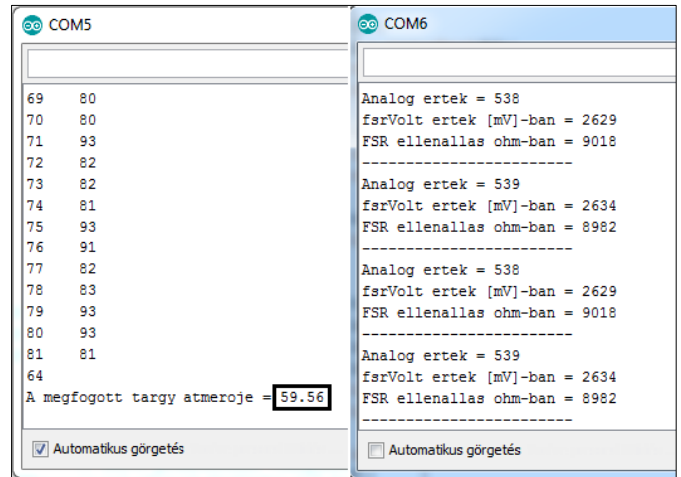
$$U_r = U \left( \frac{R}{R + FSR} \right) \quad (9.)$$

Az egyenletet átalakítva megkapható az FSR ellenállásra vonatkozó összefüggés.

$$FSR = \frac{(U - U_r) * R}{U} \quad (10.)$$

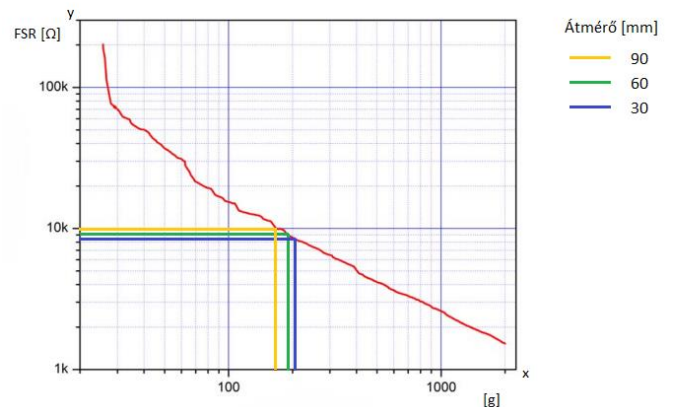
$$U = 5 [V], R = 10 [k\Omega]$$

Ezt az egyenletet levezetve az Arduino programban, eredményül megkapjuk az FSR szenzor ellenállását [Ω]-ban (17. ábra). A megfogó összeszorító erejének méréséhez egy 30 [mm], egy 60 [mm] és egy 90 [mm] átmérőjű próbadarab került 3D nyomtatásra [13]. Ezt követően mindhárom munkadarab összeszorításra került a megfogóval, miközben az FSR szenzor az egyik ujj tapintó felületére lett erősítve. A következő ábrán a 60 [mm] átmérőjű munkadarab befogását követő mérési eredmények láthatóak.



17. ábra: FSR ellenállás mérési eredménye

Bal oldalt látható a már korábban bemutatott átmérő számítás, jobb oldalt pedig az adott átmérőhöz tartozó mért értékek. Az FSR ellenállás értékek elengedhetetlenek a szenzorra ható erő megállapításához, ugyanis a kapott értékeket a következő diagram y tengelyére helyezve, majd ezen pontok magasságában a görbével kapott metszéspontokat levetítve, megkapjuk az x tengely értékeit. 100-al való osztás után jutunk a [N]-ban értendő eredményhez.

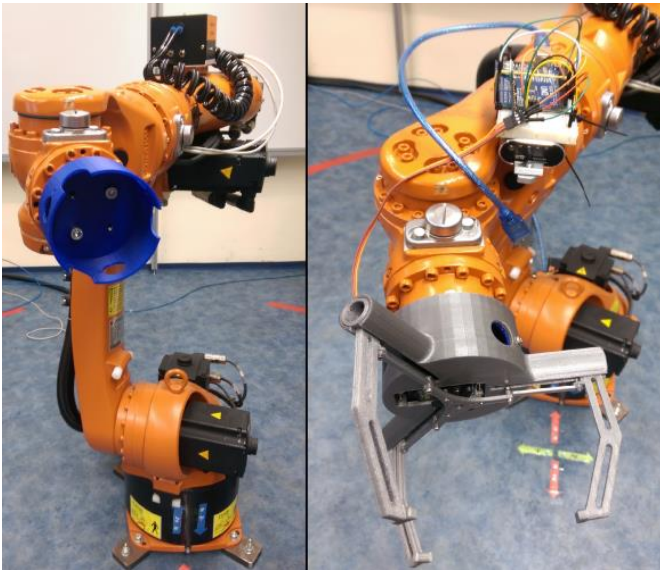


18. ábra: FSR ellenállás-erő diagram [12]

Az értékeket leolvastva látható, hogy körülbelül 1,7 [N], 1,9 [N] és 2,05 [N] nagyságú erőt fejt ki egy ujj a különböző átmérőjű darabokon.

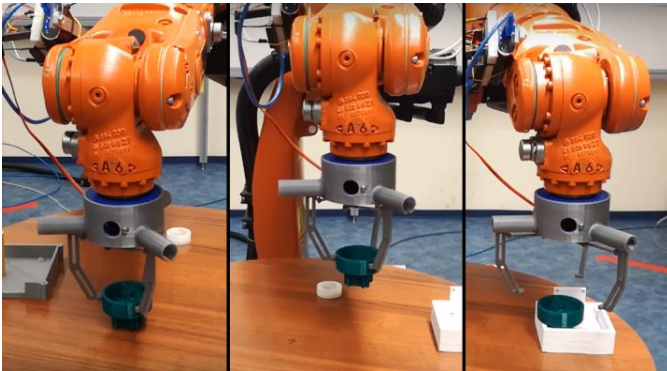
## IX. MUNKAVÉGZÉS

A megfogó felszerelését a robotkarra a már korábban bemutatott, 6. ábrán látható adapter biztosítja, amely a kialakított furatokon keresztül a robottal és a megfogóval is összeerősíthető. Továbbá az Arduino, a tápegység és a szükséges áramkör is rögzítésre került a robotkarra.



19. ábra: Megfogó felszerelése KUKA KR5-re

A felszerelést követően különböző átmérőjű munkadarabok megfogása és áthelyezése került végrehajtásra.



20. ábra: Munkavégzés

## X. ÖSSZEGZÉS

A feladat során kitűzött célok és ötletek megvalósultak, és ezek működése a dolgozat pontjaiban magyarázatra és bemutatásra kerültek. A megfogó alkalmas pozíciójának detektálására, a megfogott tárgy átmérőjének számítására, egy ujj erő kifejtésének mérésére és munkavégzésre KUKA KR5 robotkarra való felszerelést követően.

## KÖSZÖNETNYILVÁNÍTÁS

A publikáció elkészítését az EFOP-3.6.1-16-2016-00022 számú projekt támogatta. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.

## HIVATKOZÁSOK

- [1] „Forgattyús mechanizmus,” [Online]. Available: [https://hu.wikipedia.org/wiki/Forgatty%C3%BAs\\_mechanizmus](https://hu.wikipedia.org/wiki/Forgatty%C3%BAs_mechanizmus). [Hozzáférés dátuma: 27 11 2017].

- [2] „Autodesk Fusion 360,” [Online]. Available: <https://www.autodesk.com/products/fusion-360/overview>. [Hozzáférés dátuma: 27 11 2017].
- [3] „TowerPro MG995 Servo,” [Online]. Available: <https://servodatabase.com/servo/towerpro/mg995>. [Hozzáférés dátuma: 27 11 2017].
- [4] „PLA vagy ABS? Ez itt a kérdés!,” [Online]. Available: [http://freedee.blog.hu/2014/11/26/pla\\_vagy\\_abs\\_ez\\_itt\\_a\\_kerdes](http://freedee.blog.hu/2014/11/26/pla_vagy_abs_ez_itt_a_kerdes). [Hozzáférés dátuma: 27 11 2017].
- [5] „Ultimaker Cura software,” [Online]. Available: <https://ultimaker.com/en/products/ultimaker-cura-software>. [Hozzáférés dátuma: 27 11 2017].
- [6] F. Reed, „How Do Servo Motors Work,” [Online]. Available: <https://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html>. [Hozzáférés dátuma: 27 11 2017].
- [7] „Arduino Bluetooth controller,” [Online]. Available: <https://play.google.com/store/apps/details?id=com.giumig.apps.bluetoothserialmonitor>. [Hozzáférés dátuma: 27 11 2017].
- [8] „Arduino,” [Online]. Available: <https://www.arduino.cc/>. [Hozzáférés dátuma: 27 11 2017].
- [9] „Fritzing,” [Online]. Available: <http://fritzing.org/home/>. [Hozzáférés dátuma: 27 11 2017].
- [10] „Force Sensitive Resistor (FSR), Overview,” [Online]. Available: <https://learn.adafruit.com/force-sensitive-resistor-fsr/overview>. [Hozzáférés dátuma: 27 11 2017].
- [11] „Force Sensitive Resistor (FSR), Using an FSR,” [Online]. Available: <https://learn.adafruit.com/force-sensitive-resistor-fsr/using-an-fsr>. [Hozzáférés dátuma: 27 11 2017].
- [12] „Interlink Electronics FSR Force Sensing Resistors,” [Online]. Available: [http://www.produktinfo.conrad.com/datenblaetter/500000-524999/503368-an-01-en-DRUCKSENSOR\\_FSR\\_400.pdf](http://www.produktinfo.conrad.com/datenblaetter/500000-524999/503368-an-01-en-DRUCKSENSOR_FSR_400.pdf). [Hozzáférés dátuma: 27 11 2017].
- [13] T. I. Erdei, Zs. Molnár, N. C. Obinna, G. Husi, „A Novel Design of an Augmented Reality Based Navigation System & its Industrial Applications,” 15th IMEKO TC10 – Technical Diagnostics in Cyber-Physical Era Budapest, Hungary, 6 – 7 June, 2017 - Organised by: MTA SZTAKI – Hungarian Academy of Sciences - Institute for Computer Science and Control.
- [14] Zs. Molnár, T. I. Erdei, N. C. Obinna, G. Husi, „A novel design of an air-cushion vehicle and its implementation,” MATEC Web Conf. Volume 126,
- [15] T. I. Erdei, Zs. Molnár, N. C. Obinna, G. Husi, „Surveillance and Security System in the Building Mechatronics Research Center,” Electrical Engineering and Mechatronics Conference EEMC'16 - 7th-18th-19th March, 2016.