

# *Autószimulátor tervezése Arduino Pro Micro fejlesztői platformmal*

Szóllósi Gábor  
Debreceni Egyetem, Informatikai  
Kar  
Debrecen, Magyarország  
gabesz9530@gmail.com

Beatrix Papp  
London South Bank  
University  
School of Law and Social  
Sciences,  
London, United Kingdom  
pappb@lsbu.ac.uk

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

**Absztrakt** - Az autószimulátorokat egyaránt használják szórakozásra és az autózvezetői tanulók betanítására, az autózvezetési élményéhez való hozzászoktatására. Már a 80' évek közepén elkezdődött a különböző autószimulátorok gyártása, igaz akkor még csak arcade gépeken futottak a programok. Az egyre fejlődő videójáték ipar megköveteli, hogy minél modernebb, minél több igényt kielégítő játékhordozót és/vagy játékszoftvert dobjanak a piacra. A projektben egy olyan prototípus autószimulátor került megtervezésre és megépítésre, mely mellőzi drága elektronikai hardwareket és könnyen illeszthető a modern számítógépes szoftverek számára. A fő fejlesztési platformnak a PC került megjelölésre.

A kormány egységet egy Arduino Pro Micro board vezérli, melynek a mozgásérzékelése egy MPU6050-es gyorsulásmérővel és giroszkóppal lett megvalósítva.

**Kulcsszavak:** *Autószimulátor; Arduino Micro Pro; ATmega32U4; rFactor; kormány szerkezet; kormány; MPU6050;*

## I. BEVEZETŐ

Manapság egyre elterjedtebbek, a különböző szimulátorok, melyek segítségével való életbeli helyzeteket tudunk biztonságos környezetben tesztelni, például autózvezetés, repülőgép irányítás és más veszélyes tevékenységeket. Ezen szimulátorok segítségével anélkül be tudjuk tanítani az autózvezetőket és a repülőgép pilótákat arra, hogyan reagáljanak egy esetleges ütközés esetén vagy egyáltalán, hogyan kell a járművet irányítani, anélkül, hogy kint a való életben találkozoznának vele először. A kutatás/fejlesztésnek a Debreceni Egyetem adott otthont [12].

## II. AUTÓSZIMULÁTOR

Nagyon sokan összekeverik, hogy az autójátékok és az autószimulátorok ugyanazok. Bár hasonlóak, sokkal több tulajdonságban térnek el, mint amennyiben megegyeznek. Igaz mind a két játékban autót kell irányítani, azonban

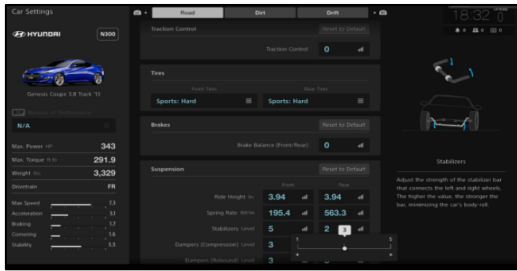
teljesen másként viselkednek a járművek fizikailag. Azokat az autós játékszoftvereket amelyek nem szimulátorok, „Arcade”

játékoknak nevezzük. Az ilyen fajta programok nem követelnek annyira nagy felkészültséget és nem jelentenek akkora kihívást a játékosok számára, inkább a vezetés közbeni ütközésekre, autósüldözésekre koncentrálnak (lásd 1.ábra). Ezek a szoftverek a valóságtól elrugaszkodnak, s nem valós helyszíneken játszódnak, egyfajta „Casual” réteget céloznak meg.



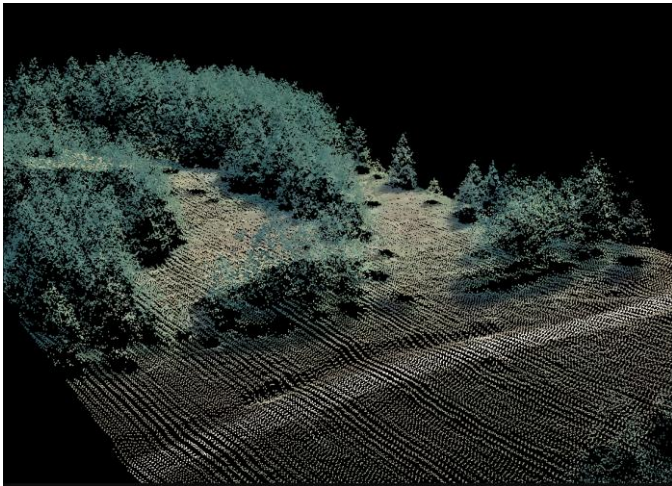
1. ábra Need for Speed arcade szoftver [4]

Az autószimulátorok azonban felkészült összpontosított elmét igényelnek, valóság-hű vezetési élményt biztosítanak. Már a verseny elindítása előtt az autó részletes beállítását követeli (lásd a 2. ábra) meg a játékostól, a vezetés során pedig a legapróbb részletekre is megköveteli a figyelmet, mint például a gumik kopására, időjárás viszonyoknak megfelelően való vezetésre, a többi versenyzővel történő minél kevesebb kontaktusra, hiszen a legapróbb koccanások is komoly károkat okozhatnak az autóban és megváltoztatják a megkövetelt vezetési stílust. Fontos megegyezni, hogy vannak olyan típusok szoftverek, melyek egyben autószimulátor és arcade-nek is számítanak.



2.ábra Gran Turismo Sport autó-beállításai menüje [5]

Ezen a játékoknak a fejlesztői gyakran valós pályákat látogatnak meg azért, hogy be-scenellhesék.



3.ábra LIDAR scan [6]

Ma már bevett szokás, hogy a tervezéseknél a 3D modellezési szakaszt igyekeznek lerövidíteni, ennek eredménye képen a motion capture vagy LIDAR [5] alapú scennelési technikákat alkalmazzák [8].



4.ábra 3D scan [7]

A 3D nyomtatási technológiánál is gyakran ezt a megoldást alkalmazzák, azonban scennelés során létrejövő point cloud komoly számítási teljesítményt igényel a végleges 3D modell rekreációjához.

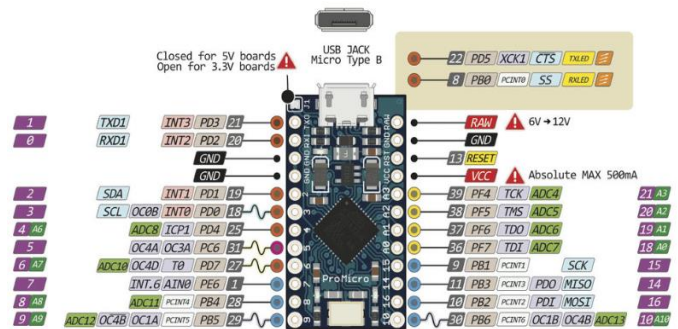
Általánosan elmondható, hogy az autó szimulátorokat, hogy a lehető legjobb valósághű élményt nyújtsák, nem billentyűzetet vagy kontrollert, hanem autószimulátor pedált, kormányt és sebességváltót alkalmaznak. Vannak olyan

játékosok akik csakis ezekkel hajlandók kedvenc játékszoftverüket használni. Magyarországon is elterjedt az ilyen fajta versenyzés, a legnagyobb magyar online közösségnek a Gran Turismo és a Forza Motorsport rajongói a tagjai közt is.

### III. HARDWARE EGYSÉGEK

Az Arduino egy szabad szoftveres, nyílt forráskódú elektronikai fejlesztőplatform, ami úgy lett kialakítva, hogy a különböző projekteknél az elektronikus eszközök minél könnyebben hozzáférhetőek és kezelhetőek legyenek. Széles tömegek számára elérhető, mivel olcsó, könnyen beszerezhető, és Open-Source minősítésű, továbbá egyszerűen programozható, és csatlakoztatható más eszközökhöz. A fejlesztői platform az úgynevezett IDE-ből (integrált fejlesztői környezet), és az Arduino Board-okból áll. Előbbi segítségével maga a program írható és tesztelhető a számítógépen, utóbbi pedig egy elektronikus eszköz, amelyre az előzőleg elkészített programok kerülnek feltöltésre a számítógépen keresztül, majd elektronikus eszközök vezérelhetők a segítségével. Ebben a projektben egy Arduino Pro Micro típus lett felhasználva (az 5.ábrán látható), mivel ez rendelkezik a projekthez szükséges USB-porttal, így magát a boardot lehet egér vagy billentyűzetként is használni. A board memóriájának a kapacitása 32kB. A választást megerősítette az is, hogy ez a board rendelkezik ATmega32U4 processzorral is [9].

Ugyancsak az elősegítette ezen board választását az, hogy rendelkezik I2C bemenetekkel, amelyek ideálisak, ha egy gyorsulásmérő szenzort szeretnénk hozzákapcsolni, ebben az esetben egy MPU6050 típusú giroszkópot.



5.ábra Arduino Pro Micro fejlesztői modul [10]

### IV. TERVEZÉSI SZEMPONTOK

Elsődleges célpont az volt, hogy olyan kormányt tudjunk készíteni, amely nem úgy viselkedik, mint például egy billentyűzet, ami csak a két végpontot tudja érzékelni, vagy teljesen jobbra vagy pedig teljesen balra tekeredik, hanem hogy fokozatosan tudjuk szabályozni mind a két oldalra való kormányzást.

A kormányt egy a már ezt megelőző projektben elkészített pedálrendszerhez kellett rögzíteni, hogy pontosan olyan élményt adhasson mintha egy autóban ülénk. Ugyan csak nagy figyelmet igényelt, hogy mindezt egy Arduino Micro Pro mikrokontrollerrel tudjuk működtetni és hogy a kormány elforgatását érzékelő MPU6050 kompatibilis legyen a használt



Arduino board-dal, hogy a számítógéphez közvetlenül tudjuk csatlakoztatni.

Továbbá a flexibilitás is a főbb prioritások közé esett, a jövőbeni tovább-fejlesztési lehetőségek miatt, hogy hozzá csatlakozhat legyen sebességváltó is.

#### V. AZ AUTÓSZIMULÁTOR RENDSZER FELÉPÍTÉSE

A kormány elkészítésekor felhasznált alkatrészek nagy része bontott autóból származott, hogy ezzel is a megtervezett autó szimulátor élet hűbb legyen.

A hosszú fémrúd, amellyel a pedál és a kormány össze lett kötve, egy autó ülésének a hátra tolásához használt sínből lett elkészítve, amely két részre flexelése után és a csapágytartóhoz forrasztása után lett a szerkezet alapja, a pedálrendszerhez a minél nagyobb mobilitás végett csak csavarral lett odaszorítva, nem pedig forrasztva viszont a szilárdsága, így is garantált.

A fémalkatrészek összemunkálása során nem gáz, hanem elektromos hegesztőt került használatra, ami nem csak kevésbé szennyezi a környezetet, de ilyen kis méretű fémalkatrészek forrasztásánál gazdaságosabb is. A kormány szerkezet maga is forrasztva lett, hiszen a kormány és a csapágy szerkezet nem ugyanabból a gépjárműből származtak, így ezek is igényeltek kisebb átalakítást.



6. ábra A kormány szerkezete

A kormányra fel lett szerelve egy az esetleges túltekerést megállító csavar, mely egyfajta szabályzóként funkcionál (lásd 6. ábra).

Az áramkör megépítéséhez egy Arduino Pro Micro mikrokontroller lett használva, ami ATmega32U4-es processzorral van ellátva (ez a programozását tekintve nagyon fontos, mivel a Joystick.h library csakis ilyen típusú rendszeren egységgel működik). A mozgás feltérképezhetősége végett, a board-hoz lett csatlakoztatva egy MPU6050 típusú giroszkóp [10].

Az modul egy USB vezetékkel lett a géphez csatlakoztatva. A három darab toló potenciométer CDE23N-60-B5K típusú és lineáris karakterisztikájú, maximum üzemi feszültsége 500V és a toleranciája +-20%-os. A tolópálya

hosszúsága 60mm [4]. Az áramköri egységeket forrasztással kötöttük össze.

Mint, ahogy korábban említésre került a pedál rendszer kialakítása és megtervezése ezen projekt egy korábbi állomásánál valósult meg. A pedálok autó ülések háttámlájának a merevítőjéből készült, a szerkezet maga fém saroklemezekből lett összehegesztve, amik összemunkálására elektromos nem pedig gázos hegesztő lett használva, a fémrészek flexszel lettek méretre vágva. Miután elkészült a hegesztés, hogy forrasztások minél erősebb legyen ezért ponthegesztővel is meg lett dolgozva, a nagyobb erőbírás végett.



7. ábra A kormány szerkezete

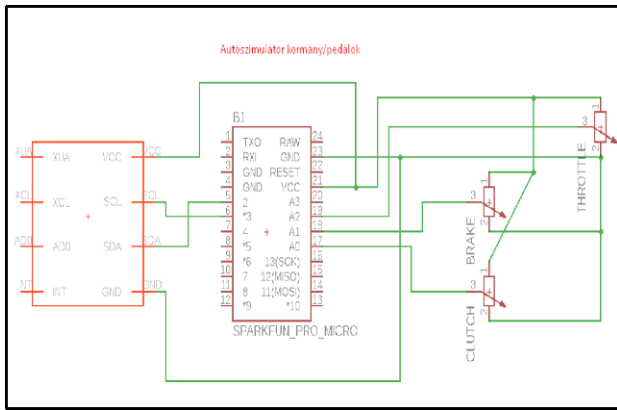
A pedálok úgy lettek elhelyezve, hogy lábbal kényelmesen lehessen váltani közöttük. A kis fogók 7cm-esek, így, ha teljesen rátaposunk a gázra akkor sem teszünk kárt a potméterekben.

Összeszerelt autószimulátor egység.



8. ábra Az elkészült Autószimulátor

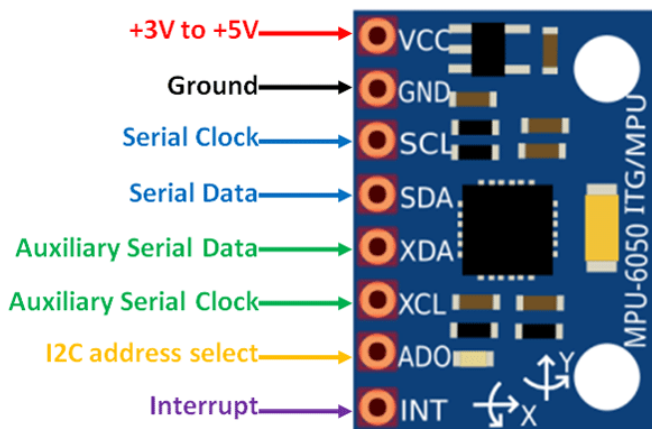
A három darab potenciométer 1-es számozású bemenetei a VCC feszültségre vannak kötve, amely 5V-os nagyságú ebben az esetben. A 2-es számozású kimenetek az Arduino GND bemenetéhez vannak hozzákötve, azaz leföldelve. Mivel a 3-as számozású potméterkimenetek a board egyes változó bemenetére vannak rákötve, ez a 3 bemenet alkotja a program változóinak az értékét. A gáz az A0-s, a fék az A1-es a kuplung pedig az A2-es bemenetre van rákötve.



9. ábra A rendszer kapcsolási rajza [3]

Ahhoz, hogy a kormány is működőképes legyen, azt az Arduino Micro Pro 2-es és 3-es port-jára kellett csatlakoztatni, mivel ezen a két bemeneten van értelmezve az I2C protokoll, melyet a giroszkóp is használ. Az I2C protokoll (Integrated-Integrated Circuit), ezt akkor szokták alkalmazni, amikor IC van kötve IC-hez.

A giroszkóp rendelkezik továbbá egy INT pinnel is, ezt az esetleges beavatkozáshoz lehet használni, ha szeretnénk megszakítani a működését.



10. ábra Az MPU-6050 giroszkóp [3]

Az arduino board 2-es pin-jére lett rákötve a giroszkóp SDA kimenete azaz a Serial Data, ami a soros adatokat szolgáltatja. A 3-as portra lett rákötve az SCL azaz Serial Clock, ami pedig a soros órajelet szolgáltatja. Az összes boardba menő vezeték, közvetlenül a board-hoz lett hozzáferrasztva. Az áramkör forrasztásához külön ilyen eszközök forrasztására használatos eszköz lett alkalmazva (amely a 10. ábrán látható is).

## VI. TESZTELÉS

A tesztelés Windows környezetben zajlott, azon belül is Windows 8.1 x64 típusú operációs rendszeren. A tesztelések megkezdése előtt frissíteni kellett a drivereket, mivel a rendszer Arduino Leonardo billentyűzetként tartotta számon.

Mielőtt bármilyen szoftvert elindítanánk meg kell győződnünk, hogy kontrollerként kezeli az Arduino-akat a számítógép.

Az autószimulátor tesztelésre egy ismert autószimulátor az rFactor-ban került sor [2]. Azért erre a program esett a választás mivel, alacsony rendszerigényű és közel áll a klasszikus értelemben vett, autó szimulátorokhoz, illetve FreeTrial verzió volt.

Az autószimulátor isiMotor2 engine-t használ.

A program rendszer követelménye:

- CPU: 1.4 GHz
- RAM: 512 MB
- VGA: 128 MB VRAM
- HDD: 2GB [2].

A programon belüli elindítása során, a tesztelés menüpont alatt lett elindítva egy tesztpálya, ahol a pálya betöltése után, a SETTINGS menüpont alatt be lehet állítani a KEY BINDING-ot, azaz, hogy mivel lehessen a gázt a féket a kormányzást használni. Itt be lett állítva, hogy a jobb oldali pedál a gáz, a fék a középső, a bal oldali pedál pedig a kuplung legyen, a kormányt pedig a projekt keretein megépített kormányval tudjuk irányítani.

A szoftver autokuplung-ot használ alapvetően, de ezt ki lehet kapcsolni, amint elindul a tesztkör, így már a program el is várja, hogy minden egyes sebességváltáskor használjuk a kuplungot.

Throttle	JOY1 S0-
Brake	JOY1 Y-
Steer Left	JOY1 Z-
Steer Right	JOY1 Z+
Shift Up	UP_ARR
Shift Down	DN_ARR
Neutral	-EMPTY-
Clutch	JOY1 X-
Toggle AI Control	L_KEY

11. ábra Alkalmazott beállítások

A 11. ábrán látható a tesztelésnél használt beállítások, mint ahogy az ábrán is látható a kuplung (bal szélső pedál) kapta a JOY1 X- nevet, a fék (középső pedál) kapta a JOY1 Y- nevet és a gáz (jobb szélső pedál) a JOY1 S0- nevet. A kanyarodást a JOY1 Z-(balra) és a JOY1 Z+(jobbra) viszont a felfelé vagy lefelé váltás gombjai maradtak a billentyűzetben használtak. A tesztelést megkönnyítette, hogy a menüben már volt egy olyan beépített jelző, amin ellenőrizni lehetett, hogy a gáz a fék és a kuplung mennyire érzékeny, és ha ez nem megfelelő, még lehetett korrigálni rajta a szoftveren belül egy másik menüpont alatt.

A tesztkör az elején még automata módban van, azaz nem engedi a játékost beleavatkozni a játék menetében, de amint a gépkocsi elhagyja a box utcát akkor már teljesen a vezető kezében van az irányítás. Az alapszoftver csak egy pályát tartalmaz, de egy teljes körön belül a pedálok és a kormány minden kombinációja tesztelésre került.

## VII. AZ ARDUINO PRO MICRO PROGRAMJA

A mikrokontroller programja az Arduino IDE, egy a kifejezetten Arduinok programozására való szoftverrel készült, aminek a neve Arduino IDE [9].

A szoftver feltelepítése után, az *Eszközök* menüpontra belül a *Processzorok* fülénél kell kiválasztani a programozni kívánt eszközt, de mivel az alapértelmezettek között nem szerepelt az Arduino Pro Micro, így azt egy külön csomag letöltésével kellett hozzáadni. A Sparkfun [11] csomagjában már megtalálható egy ATmega32U4 típusú processzor, amely teljesen úgy működik, mint a programozni kívánt darab.

Miután ez is be lett állítva, a következő teendő a Joystick.h library letöltése volt, amely a github.com [1] webes felületről került letöltésre, és tartalmazza mindazon függvényeket, metódusokat amelyek elengedhetetlenek, hogy a pedál működni tudjon. Mivel a webes felületen lévő egy újabb 2.0.4-es verzió, így ez nem minden rendszeren kompatibilis, így az 1.0-s verzió az ami letöltésre került. A kormányzáshoz használt MPU6050 érzékelő könyvtárát és hozzá kellett adni a programhoz, mint header fájlt, így az a használatához szükséges függvények is hozzá lettek adva. Ahhoz, hogy használni tudjuk a programban ezt az érzékelőt, külön deklarálni kell egy változót, itt egy MPU6050 típusú mpu változó került deklarálásra. Ezen könyvtárak segítségével GamePad-ként tudjuk kezelni a szerkezetet.

A fejlesztői szoftveren belül be kell állítani, hogy melyik porton helyezkedik el a programozni kívánt Processzor, de hogy ezt be tudjuk állítani, először meg kellett győződni, hogy melyik portra lett csatlakoztatva, amit az operációs rendszer eszközközkezelőjén keresztül lehet megtenni. Miután ez is be lett állítva, el lehet kezdeni az eszköz programozását. Először hozzá kell adni a Joystick könyvtár mind a 3 header fájlját.

```
//=====
//=====
// Joystick (Gamepad)

class Joystick_
{
private:
  uint8_t joystickId;
  bool autoSendState;
  int8_t xAxis;
  int8_t yAxis;
  uint16_t buttons;

public:
  Joystick_(uint8_t initJoystickId);

  void begin(bool initAutoSendState = true);
  void end();

  void setXAxis(int8_t value);
  void setYAxis(int8_t value);

  void setButton(uint8_t button, uint8_t value);
  void pressButton(uint8_t button);
  void releaseButton(uint8_t button);

  void sendState();
};
```

12. ábra Részlet a Joystick3.h header fájlból

A programban változóként be lett vezetve a 3 használt bemenet, az Arduino A0 bemenete a gáz, az A1-es a fék, az A2-es bemenet pedig a kuplung értékét tárolja. A bemenetek

értékeit az analogRead() függvénnyel tudjuk kezelni, ami 0 és 5 volt közti feszültséget mér és azt 0 és 1023 közti egész számmá bontja le, de ezt az intervallumot tudjuk változtatni az analogReference() függvénnyel. Mind a 3 változóhoz tartozik egy *ertek* nevű változó ahol a az analogRead()-el bekért értékeket. Ugyancsak rendelkezik mind a gáz, a fék és a kuplung egy *ertekbyte1* és egy *byte2* változóval, ezek abban vannak a segítségünkre, hogy milyen kimenetet biztosítson a programunk.

Amint elkészült a program, akkor a kezelői felületen található pipával ellátott gombra kattintva lefordítjuk a programot. Ha ez sem produkál semmilyen hiba kimenetet, akkor fel is lehet tölteni a vezetékesen csatlakoztatott Arduino boardra, ezt a jobbra mutató nyílal ellátott gombbal tudjuk megtenni. Ha ezt is sikeresen végrehajtotta az IDE akkor tájékoztat bennünket a kimeneten.

## VIII. ÖSSZEGZÉS

Az autószimulátor kormányának tervezése és megépítése és beprogramozása sikeresen befejeződött. Az építéshez a rendelkezésre álló alapanyagok lettek felhasználva, más autó alkatrészekből. Az elkészített autószimulátor rendszer megfelelően működött a tesztek során.

További fejlesztési lehetőség közé sorolandó a sebességváltó megtervezése, megépítése és hozzá illesztése a meglévő rendszerhez.

## IX. KÖSZÖNETNYÍLVÁNÍTÁS

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.

Szeretnék köszönetet mondani Szöllősi László édesapámnak, aki nélkül a fémszerkezet nem jöhetett volna létre, ezen kívül Erdei Timotei Istvánnak a projekt elkészítése közben felmerült kérdések gyors válaszadására és Szigili Istvánnak, aki megmutatta, hogyan is kell összerakni egy a projektben használthoz hasonló áramkört, és biztosította az elektronika forrasztásánál használatos eszközöket.

## X. HIVATKOZÁSOK

- [1] Matthew Heironimus, (2017, Dec 16.) [Online]. Available: <https://github.com/MHeironimus/ArduinoJoystickLibraryT>. I.
- [2] Google, (2017, Dec 17). [Online]. Available: <http://www.rfactor.net/>
- [3] Google, (2017, Dec 17.) [Online]. Available: <http://www.jensws.com/pedals/>
- [4] Google, (2017 Dec 20) [Online]. Available: [https://cdn.vox-cdn.com/thumbor/K\\_jm2HXfDH81FH1hJ3soZ4fSAGQ=/10x0:2858x1602/1600x900/cdn.vox-cdn.com/uploads/chorus\\_image/image/55189151/needforspeed.0.png](https://cdn.vox-cdn.com/thumbor/K_jm2HXfDH81FH1hJ3soZ4fSAGQ=/10x0:2858x1602/1600x900/cdn.vox-cdn.com/uploads/chorus_image/image/55189151/needforspeed.0.png)
- [5] Google, (2017 Dec 20) [Online]. Available: <https://i.ytimg.com/vi/ExtkVSn2HjU/maxresdefault.jpg>
- [6] LIDAR, (2017 Dec 20) [Online]. Available: [http://www.pinsdaddy.com/aerial-lidar-companies\\_Qy2\\*ecImtrxxS3nGFR34IH\\*iq%7C\\*z%7CTAlu5SOnykPbQ/](http://www.pinsdaddy.com/aerial-lidar-companies_Qy2*ecImtrxxS3nGFR34IH*iq%7C*z%7CTAlu5SOnykPbQ/)

- [7] Motion Capture, (2017 Dec 20) [Online]. Available: <https://www.xsens.com/tags/motion-capture/>
- [8] 3D Scan, (2017 Dec 20) [Online]. Available: <https://www.3ders.org/images2017/german-museum-partners-formwerk3d-3d-scan-rare-bust-ancient-egyptian-pharaoh-akhenaten-1.jpg>
- [9] Arduino IDE, (2017 Dec 20) [Online]. Available: <https://www.arduino.cc/>
- [10] MPU6050, (2017 Dec 20) [Online]. Available: [https://store.invensense.com/datasheets/invensense/MPU-6050\\_DataSheet\\_V3%204.pdf](https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3%204.pdf)
- [11] SparkFun, (2017 Dec 20) [Online]. Available: <https://www.sparkfun.com/>
- [12] T. I. Erdei, Zs. Molnár, N. C. Obinna, G. Husi, „AGV cyber physical navigation system,” FMTÜ - XXII. International Scientific Conference of Young Engineers - Kolozsvár, 23-24.03.2017.