Visit [www.ascher-racing.com/OpenSimwheel](http://www.ascher-racing.com/OpenSimwheel) for the latest version of this tutorial.

The OpenSimwheel is an open-source software project to drive a servo motor as a steering wheel (for racing simulations) using off the shelf hardware. Most common is the use of a powerful servo in direct drive mode .

This tutorial intends to lower the barrier of implementing an OpenSimwheel for the casual and not so hardware savvy simracer. A general overview of the interaction of every single part is presented and also discussed in more depth. Furthermore an example part list for a full OpenSimwheel package is shown where minimal Do-It-Yourself effort is needed.

Closing up the story, a step by step tutorial of putting hardware and software in operation is provided as well as a troubleshooting guide.

---

1 steering wheel is directly clamped to the motor shaft – no gear reduction
## Contents

1 **Introduction**  
   1.1 Credits  
   1.2 Overview

2 **Part List**  
   2.1 Servo Drive  
   2.2 Servo  
      2.2.1 Tested Motors  
      2.2.2 Ordering MiGE Servo  
   2.3 Clamping Set  
      2.3.1 Beano's Universal Clamping Set  
      2.3.2 Standard Clamping Set and Adapter  
      2.3.3 3D-printed Clamping Set  
   2.4 Quick Release  
   2.5 24V Power Supply  
   2.6 STM32F4DISCOVERY Controller  
   2.7 Adapter Board  
   2.8 USB to RJ45 Adapter  
   2.9 Cables & Electronic Accessories  
   2.10 Braking Resistor  
   2.11 Servo Mounting Bracket  
   2.12 Example Ordering List

3 **Hardware Setup**  
   3.1 General Wiring  
   3.2 Disco

4 **Software Setup**  
   4.1 General Information  
   4.2 Required Software  
   4.3 Granity  
   4.4 STM32 ST-LINK utility  
   4.5 MMos FFB Tool  
   4.6 Granity – PWM  
   4.7 Uninstalling vJoy  
   4.8 Improving Performance  
      4.8.1 Granity Settings  
      4.8.2 Tuning Torque Controller

5 **Clipping**

6 **Troubleshooting**

7 **Change Log**
1 Introduction

1.1 Credits

The idea of this project was originated by Bernhard Berger at the end of 2013. The project’s documentation can be found at the German simracing community Virtual-Racing.

Another essential part, the USB HID Controller\(^2\), is developed by the user "MMos". This project is also documented at Virtual-Racing.

A lot of effort on the electronic side has been done by Dennis Reimer. With his developments, the amount of work of building oneself an OpenSimwheel is reduced immensely.

With a continuously growing user-base there are more and more helpful individuals sharing knowledge and information. I can’t name everyone in person, but a big thanks to all of you! Be it hardware, software, consulting, discussing or whatever.

1.2 Overview

The OpenSimwheel is a continuously evolving project. This tutorial pictures the current state and recommends proven concepts. Nevertheless it’s of an informative nature only and it’s up to you which parts to chose. An overview of the interconnection between the computer and the servo wheel can be seen in figure 1.

2 Part List

This section lists all relevant parts which are essential to build yourself an OpenSimwheel. Most of the parts will have multiple options to chose from. I will recommend components with (in my humble opinion) the best price-performance ratio and present alternatives. Prices are greatly dependent of your location due to individual shipping, tax and duty costs. Costs from this tutorial are based on Germany. In section 2.12 is an example ordering list for a complete OpenSimwheel package.

---

\(^2\)force feedback controller
2.1 Servo Drive

A Servo Drive is an electronic amplifier to power the servo motor. An excellent drive is the Argon, made by Granite Devices. Get it directly from their onlineshop or search for a distributor. It costs about 550€ and is the most expensive option but with (almost) no restriction to the servo power. There is a very good wiki for this device, too. It’s highly recommended to get four pieces of these heat sinks and a small fan to keep temperatures low.

Alternatively, it’s possible to use the cheaper VSD-E Servo Drive for about 200€. It’s maximum current is lower, thus producing less torque. Furthermore, an external power supply is needed which has to be considered when comparing prices!

The third option will be (not released yet at this time) the IONI servo drive. All servo drives can be seen in figure 2.

2.2 Servo

2.2.1 Tested Motors

The servos shown in table (1) have been tested and are in use as OpenSimwheel motor.
### Table 1: Servo Options

<table>
<thead>
<tr>
<th>Servo</th>
<th>MiGE 130ST-M10010</th>
<th>MiGE 130ST-M15015</th>
<th>Lenze MCS12H15L</th>
</tr>
</thead>
<tbody>
<tr>
<td>holding torque</td>
<td>10 N m</td>
<td>20 N m</td>
<td>11.4 N m</td>
</tr>
<tr>
<td>peak torque</td>
<td>20 N m</td>
<td>30 N m</td>
<td>29 N m</td>
</tr>
<tr>
<td>rated speed</td>
<td>1000 /min</td>
<td>1500 /min</td>
<td>1500 /min</td>
</tr>
<tr>
<td>rated voltage</td>
<td>220 V</td>
<td>220 V</td>
<td>158 V</td>
</tr>
<tr>
<td>torque coefficient</td>
<td>2.20 N m/A</td>
<td>1.58 N m/A</td>
<td>1.40 N m/A</td>
</tr>
<tr>
<td>rated current</td>
<td>4.5 A</td>
<td>9.5 A</td>
<td>8.2 A</td>
</tr>
<tr>
<td>rotor inertia</td>
<td>19.4 kg cm$^2$</td>
<td>27.7 kg cm$^2$</td>
<td>7.3 kg cm$^2$</td>
</tr>
<tr>
<td>encoder steps</td>
<td>10000</td>
<td>10000</td>
<td>16000</td>
</tr>
<tr>
<td>shaft diameter</td>
<td>22 mm</td>
<td>22 mm</td>
<td>19 mm</td>
</tr>
<tr>
<td>price</td>
<td>$\approx$ 245 USD</td>
<td>$\approx$ xyz USD</td>
<td>$\approx$ 900€</td>
</tr>
</tbody>
</table>

Figure 2: Servo Drives ([granitedevices.com](http://granitedevices.com))
The first one (referred as "small MiGE") might be the weakest of these three at first glance but delivers an incredible performance considered its low price tag. The second servo (referred as "big MiGE") scores with its high peak torque of 30 Nm but be prepared for more rotor inertia thus less details. The ultimate top scorer is the Lenze servo motor. About equal peak torque level as the big MiGE combined with the least rotor inertia and highest encoder resolution. These values are mirrored in the price of course.

2.2.2 Ordering MiGE Servo

If you chose to keep the budget tight, that’s how you order one of the MiGE servos:

- contact Lisa Zhan directly via mail (address below)
- ask for the servo and shipping costs
- kindly request to leave the keyway unmounted

Buying servos in pair is a money safer because shipping costs don’t raise by factor two!

Lisa Zhan  
Foreign trade sales representative  
Hang Zhou Mige Electric CO., ltd  
Website: www.hzmgdj.com.cn  
E-Mail: hzmgdjzhan@gmail.com  
phone: +86 571 5757 6990  
fax: +86 571 8230 8685  
mobile: +86 137 5826 1177

2.3 Clamping Set

The connection between the servo shaft and a 70 mm bolt pattern Quick Release or wheel, can be accomplished by multiple ways.

2.3.1 Beano’s Universal Clamping Set

There is a special OpenSimwheel clamping set designed by Beano shown in figure 3. It fits servo shaft diameters of both 19 mm and 22 mm.
2.3.2 Standard Clamping Set and Adapter

A proven and self centering clamping set standard is COM-B. This is a clamping set for both MiGE servos (22 mm shaft); Lenze servo has a 19 mm shaft and needs this one. Both are available for around 15 €.

They are normally used in mechanical engineering and not supposed to directly mount a 70 mm bolt pattern QR or wheel on it. The connection of both bolt patterns is accomplished by a further adapter for around 20 €.
2.3.3 3D-printed Clamping Set

Derek Speare has designed a clamping set which can be 3D-printed. A good source for printing services is www.shapeways.com with a price of about 80 USD for this set made out of stainless steel. The required STL-data files can be downloaded.

2.4 Quick Release

A Quick Release is highly recommended when using multiple wheel rims. Direct drive servo wheels operate not only at high torque levels but also vibrate at a large frequency band. This means that any mechanical play of the quick release system is destructive. Usually, cheap systems always have at least a tiny bit of play which can be enough to screw up the experience. But there is an affordable Quick Release for simracing which is designed to have absolutely zero play at all: Q1R for about 60€ (each side); see figure 6

2.5 24V Power Supply

The Argon servo drive needs for its logic part a 24V power supply (minimum 13 W). This power supply for around 12€ fits perfectly; see figure 7
Figure 6: Holger Buchfink’s Q1R Quick Release

(a) closed

(b) 50 and 70 mm available

Figure 7: 24 V power supply (source: reichelt.de)

Figure 7: 24 V power supply (source: reichelt.de)
2.6 STM32F4DISCOVERY Controller

2.7 Adapter Board

The connection between the Disco controller board and the Argon is documented in appendix B and can either be wired directly or via an additional adapter board seen in figure 8. It does the same job as wiring everything manually but requires no knowledge about any diagrams and works out of the box. Available for around 35€ at info@ib-reimer.de.

2.8 USB to RJ45 Adapter

In case Argon is the chosen servo drive, a USB to RJ45 Adapter (see figure 9) needs to be used to flash firmware. This adapter can be purchased for around 20€ either directly from Granite Devices or via a distributor. Flashing the firmware can be done by multiple ways (e.g. serial PCI-e card) but the USB solution is the easiest and cheapest.

2.9 Cables & Electronic Accessories

In case you don’t want to build the electronic connection by yourself, Dennis Reimer gives the option to build a complete cable harness for the OpenSimwheel.

In case you want to do it yourself, this is what is needed:
• D-SUB 15-pol (1€)
• D-SUB 15-pol housing (2€)
• RJ45 CAT6 patch cable 0.5 m (6€)
• RJ45 CAT6 patch cable 1.5 m (5€)
• 2-pole 1.5 mm² cable for 24 V (1€)
• 2 x power cables with open end (4€)
• wire sleeves (1€)

2.10 Braking Resistor

A braking resistor is not needed in the first place. But in case of high force feedback settings and quick steering inputs there is a chance of over charging the capacitors of the servo drive stage. A 100 W 100Ω resistor will avoid these over voltage peaks. Voltage peaks occur due to the servo motor acting as a generator. This resistor (see figure 10) for around 15€ does the job.

Attention when using a braking resistor! With wrong settings (see section 4.3) the braking resistor can get dangerously hot!

2.11 Servo Mounting Bracket

The servo can be attached to the sim rig in any possible way. There are dozens of solutions out there like e.g. a milled servo plate to attach in vertical orientation to an aluminum profile rig. Another option might be my mounting bracket which can be found at Ascher-Racing, see figure 11.
Figure 10: braking resistor (source: conrad.de)

(a) bolt to vertical surface
(b) bolt to horizontal surface

Figure 11: servo mounting brackets

page 12 of 33
### 2.12 Example Ordering List

See table 2 for an example ordering list.

<table>
<thead>
<tr>
<th>part</th>
<th>ordering</th>
<th>≈ price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argon Servo Drive</td>
<td>CNC-Spezialist</td>
<td>550 €</td>
</tr>
<tr>
<td>Servo MiGE 130ST-M10010</td>
<td>see section 2.2.2</td>
<td>350 €</td>
</tr>
<tr>
<td>19 mm COM-B Clamping Set</td>
<td>Holger Buchfink</td>
<td>15 €</td>
</tr>
<tr>
<td>OpenSimwheel Clamping Set Adapter</td>
<td>Holger Buchfink</td>
<td>20 €</td>
</tr>
<tr>
<td>Q1R Quick Release</td>
<td>Holger Buchfink</td>
<td>150 €</td>
</tr>
<tr>
<td>24 V Power Supply</td>
<td>reichelt.de</td>
<td>12 €</td>
</tr>
<tr>
<td>STM32F4DISCOVERY Controller</td>
<td>ebay.de</td>
<td>17 €</td>
</tr>
<tr>
<td>Disco Adapter Board</td>
<td>Dennis Reimer</td>
<td>35 €</td>
</tr>
<tr>
<td>USB to RF45 Adapter</td>
<td>CNC-Spezialist</td>
<td>24 €</td>
</tr>
<tr>
<td>D-SUB 15-pol</td>
<td>reichelt.de</td>
<td>1 €</td>
</tr>
<tr>
<td>D-SUB 15-pol housing</td>
<td>conrad.de</td>
<td>2 €</td>
</tr>
<tr>
<td>RJ45 CAT6 patch cable 0.5m</td>
<td>conrad.de</td>
<td>6 €</td>
</tr>
<tr>
<td>RJ45 CAT6 patch cable 1.5m</td>
<td>conrad.de</td>
<td>5 €</td>
</tr>
<tr>
<td>2-pole 1.5mm2 cable</td>
<td>conrad.de</td>
<td>1 €</td>
</tr>
<tr>
<td>2 x power cables with open end</td>
<td>reichelt.de</td>
<td>4 €</td>
</tr>
<tr>
<td>wire sleeves</td>
<td>conrad.de</td>
<td>1 €</td>
</tr>
<tr>
<td>Braking Resistor</td>
<td>conrad.de</td>
<td>15 €</td>
</tr>
<tr>
<td>Servo Mounting Bracket</td>
<td>Ascher-Racing</td>
<td>69 €</td>
</tr>
</tbody>
</table>

\[ \sum \approx 1200 € \]

Table 2: Example Ordering List

### 3 Hardware Setup

#### 3.1 General Wiring

Appendix B shows the technical wiring diagram of the complete OpenSimwheel package. Credits goes to Beano for creating this diagram.

Due to high currents of the servo power cable, try to separate it from all other cables. Use shielded cables and ferrit cores for the servo feedback cable. In case of using an aluminum profile rig and problems still occur, try to ground the rig.
3.2 Disco

The STM32F4DISCOVERY Controller has two USB sockets. Figure 12 clarifies the two different types.

**Mini-USB** is needed for flashing MMos firmware and power supply. In case Dennis’ adapter board is used, this mini-USB connection is for (MMos) firmware upgrades only, as the adapter board provides the power from the Argon. In contrast, **micro-USB** provides data connection to the PC and has to be connected at any time.

Dennis’ adapter board can be directly plugged-in the Disco controller. Take care of the orientation as there are two possible ways. See appendix D to see how it fits together.

4 Software Setup

4.1 General Information

Most important thing is safety! Make sure the servo motor is clamped rigidly and the shaft can spin without destroying anything (e.g. no coiled cable from rig to wheel). Be always prepared for the wheel to rev up or oscillate. If something unexpected happens, don’t grab the wheel by reflex but use the STO\(^3\) (emergency switch) instead.

Read carefully and follow the instructions step by step as described in following sections.

There is a known incompatibility of the MMos FFB Tool with vJoy and Bodnar controllers. This FFB tool is only needed to set up the wheel or making changes (e.g. damping), not for actual driving (you can then use your Bodnar controllers of course). In this case, simply

\(^3\) safe torque off
unplug all controllers when using MMos FFB Tool. If vJoy is installed on your computer you’ll have to uninstall it (see section 4.7)

4.2 Required Software

These programs, drivers or firmwares are required:

1. Granity
2. STM32 ST-LINK utility
3. MMos FFB
4. Virtual COM Port driver
5. Visual C++ Redistributable
6. Argon core firmware
7. Argon I/O firmware
8. motor settings file

4.3 Granity

This section guides through setting up Argon with Granity software. The drive stage has Dip-switches (see figure 13) to set the device address that identifies the device on a multidrop communication bus. In our case, we need this to change between normal operation and firmware upgrade mode. Set the Dip-switch as follows:

normal operation 00011 (device adress 1)
firmware upgrade 00001 (device adress 255)

1. download Granity software
2. extract the files to a new folder and start the program by clicking Granity.exe.
3. set Argon Dip-switches to firmware upgrade mode (see figure 13)
4. connect Argon and PC via RJ45 patch cable and Simple Motion V2 adapter
5. Windows recognizes it as a serial port normally; if not install the Virtual COM Port driver manually
6. power on the Argon
7. Connect Tab → select the SMV2 adapter

*SimpleMotion V2 USB adapter driver – in most cases Windows detects it automatically
*often already installed through Windows update
8. connect
9. select Argon at adress 255
10. select core firmware
11. restart Argon
12. select Argon at adress 255
13. select I/O firmware
14. turn off Argon
15. set device adress 1 (DIP 00011)
16. turn on Argon & connect
17. load settings from file
18. select motor setting drc-file
19. save settings to drive
20. disconnect

This are not the final settings and will be changed after MMos FFB is installed to the Disco controller (see section 4.6)

If a braking resistor is used the Granity F0V parameter needs to be increased. Otherwise, the Argon would constantly heat the resistor, even without load!

---

*Over voltage fault threshold*
4.4 STM32 ST-LINK utility

This software is required to flash the MMos FFB hex-file to the Disco controller board. See figure 14 for screenshots.

1. connect mini-USB cable to Disco board
2. install STM32 ST-LINK utility
3. connect
4. erase chip
5. program and verify
6. select MMos hex-file
7. start flashing
8. disconnect
9. unplug mini-USB cable
4.5 MMos FFB Tool

If the Disco board is set up for the first time do the following steps to configure the wheel and its center position and then restart the MMos FFB Tool.

**Do not click DFU Mode!**

1. connect micro-USB to Disco board
2. download MMos FFB and start MMos_FFB_2014.exe
3. green/ red button → MMos firmware ok/ not ok
4. click Setup; see figure 16
5. PWM Modus → PWM & Dir
6. PWM Frequenz → 3.4 kHz
7. Encoder CPR: MiGE/ Lenze → 10000/ 16384
8. Index auswerten → tick
9. Tasten 1-16 → STM32 Pins
10. Speichern (save)
11. Reset durchführen? → No
12. Abbrechen (cancel)
13. Konfiguration (configuration) → tick Datei → Speichern → tick Eprom → Speichern
14. Beenden (quit)

Start this tool again to set the center point of the wheel. This has to be done only once.

1. tick Datei → Laden
2. turn wheel to center position
3. tick Lenkung zentrieren und Offset speichern
4. tick Datei → Speichern
5. tick Eprom → Speichern
6. Beenden

Now, switch off the Argon and Disco to see what the initial startup is like. Once under power, the wheel starts to oscillate carefully to initialize itself. It’s called phasing and after about 20 s a clicking sound indicates phasing finished. The Disco shows a blue LED flashing which means that it waits for you to turn the wheel until the index is found. Once you’ve done this, the LED turns off and the wheel is automatically centered and ready to go.

Try to bolt your wheel to the servo shaft in a way that the servo index is near the top (12 o’clock). This is better for the Disco board to find the index reliably.
If this initial configuration process was successful you can start the MMos FFB Tool to make adjustments to the force feedback. The tool is only needed for this process and is not required for driving. Once you are happy with your settings, this tool is not needed anymore.

The tool works like this schema: Datei → Laden → adjusting parameters → Eprom → Speichern → Beenden

See figure 15 for a starting point to adjust the wheel settings. To perform best, don’t set effect filters too high as this induces latency and increases oscillations.

**Lenkwinkel** degrees of rotation

**Lenkung zentrieren** set center position

**Lenkanschlag** hardstop

**Verstärkung** set ramp angle of hardstop

**Kraft Maximum** hardstop force

**Alle Effekte** all effects

**Effekt Filter** general filter

**Kraft Minimum** minimal force

**Kraft Maximum** maximum force

**Feder Effekte** spring effect

**Zentrierfeder an** switch on spring effect

**Federkraft** spring force

**Dämpfungs Effekte** damper effect [overwritten by iRacing's ingame damper slider!]

**Effekt Filter** effect filter

**Dämpfer an** switch on damper effect

**Dämpferkraft** damping force

**Reibungs Effekte** friction effect

**Effekt Filter** effect filter

**Reibung an** switch on friction effect

**Reibungskraft** friction force

**Trägheits Effekte** inertia effect

**Effekt Filter** effect filter

**Trägheit an** switch on inertia effect
4.6 Gravity – PWM

The Gravity parameters from section 4.3 loaded via drc-file are set to [CRI] Setpoint input = Serial. This now has to be changed to Pulse Width Modulation and saved to the drive.

The reason for not setting it to PWM at first hand: Without the MMos Disco controller connected to the PC and Gravity set to PWM input, the wheel could spin at full speed. The Argon can’t differentiate between 0 and 1 without reference.

4.7 Uninstalling vJoy

In case this software is installed, you need to remove it because the MMos FFB Tool will crash. Use an appropriate program like for example CCleaner for uninstallation and run the Registry-Cleaner afterwards. Reboot the computer and do the following commands:

1. press windows button
2. type cmd.exe
3. press Ctrl + Shift + Enter (start as admin)
4. type pnputil -e | find /n "Shaul" to find remains of Shaul Eizikovich’s vJoy
5. type pnputil /e to list all not directly with Windows implemented drivers
6. type pnputil /f /d "example name" to delete the driver
7. type pnputil -e | find /n "Shaul" to see if it’s still there

If the MMos FFB Tool still keeps crashing, a Bios update (likely with ASRock) or complete new installation of Windows might help.

4.8 Improving Performance

4.8.1 Gravity Settings

There are two important parameters in Gravity which have a great influence to the feeling:

[CIS] This option ticks the setpoint smoothing. It’s recommended to not use it.
Torque Band Width should be set as high as possible and as low as needed. A high setting might feel notchy, almost like driving with an impact wrench.

### 4.8.2 Tuning Torque Controller

This section shows how to fine tune the Argon torque controller for optimal performance. This means finding the correct gain values for the torque controller inside the Argon to achieve a proper response from a torque setpoint change. Visit the official Granitedevices Wiki for a step by step manual.

These gain values, namely MR and ML, can be taken from the motor datasheet. Due to manufacturing tolerances these parameters may not be quite accurate hence testing can increase the overall performance.

The test itself is very simple and can be done in a few minutes. The only change required in the hardware setup is to constrain the rotation of the motor shaft. The Argon will output full torque for a fraction of a second and the torque achieved by the servo motor will be plotted. A free spinning shaft affects the data as the torque achieved is only balanced with the wheel inertia.

See figure 18 for a simple design to clamp the motor shaft.

Figure 17 shows the influence of both parameters MR and ML regarding torque response. In figure 19 the torque response of my MiGE 130ST-M10010 with optimal settings is shown. The parameters are optimal for this very motor but can be a good starting point for testing other motors.

### 5 Clipping

Clipping is a form of distortion that limits a signal once it exceeds a threshold. When talking about clipping of force feedback wheels this signal threshold is the torque of the steering wheel.

Figure 20 shows a schematic signal clipping where the amplitude of a sine wave rises until it reaches the system limit (clipping threshold). To avoid this effect, and at the same time maximize the details of the ffb signal, the torque threshold has to be set correctly.

In order to find this certain value, first one has to understand how the signal from the simulation is handled by the servo motor or any FFB-wheel: The simulation has to scale the force feedback torque value on a range from \(-100\%\) to \(100\%\). It cannot directly demand a specific torque (like e.g. 3.4 Nm) but has to use this uniteless normalization instead.

---

7 Motor Resistance
8 Motor Inductance
Figure 17: Tuning Torque Controller for MiGE 130ST-M10010
Figure 18: clamping servo motor shaft

Figure 19: 130ST-M10010 perfect settings: MR 2.6; ML 11.5
This torque range equals the system range from figure 20 and means in simple words: “When the calculated torque from the simulations equals \( x \) N m the reported value to the wheel is 100%”.

If you set this torque threshold very high there would be hardly any clipping. On the other hand the wheel would be very light and information (or detail) is lost in a similar way like when the calculated torque gets clipped.

The said torque threshold is different for every car and every track. To make things even more complicated it also changes when the cars setup or track conditions change. Finding the right torque threshold value can be done by looking at the telemetry file of a clean lap. This can be seen in figure 21. It’s a recorded lap with iRacing’s Lotus 79 at Mosport. As you can see, the torque range goes up to 30 N m with occasionly spikes from heavy curbs. A good setting for the in-game threshold would be 28 – 30 N m, because this covers most of the torque band with clipping only a few peaks.

And here comes the benefit of a high torque servo motor: The in-game clipping threshold is set to the servo’s peak torque. That means all torques (people often refer to force) are mapped 1:1 and are not scaled. But this only works if the maximum servo torque is high enough compared to the simulated steering rack torque by the game physics engine.

It’s easier with numbers:

- car example requires 19 N m at maximum lateral acceleration
- in-game torque threshold set to 19 N m means that the simulations reports 100% of “force” when reaching 19 N m
- wheel \( F \) is capable of maximum 7.3 N m. It delivers this amount of torque when the simulations reports 19 N m in-game. It’s scaled by 1:2.6
- wheel \( O \) is capable of maximum 19 N m and is scaled 1:1
During a slow turn only 5 N m are requested by the simulation.

- wheel \( F \) would scale this value and only \( 5/2, 6 = 1.92 \) N m are output by the wheel
- wheel \( O \) would output true 5 N m as it’s unscaled

That’s the next advantage of a strong direct drive wheel. It’s not about the peak torques and bashing over curbs. It is the enhanced detail which is perceived at low forces.

On top there is a large headroom for high peaks which makes the whole force feedback experience more believable. Imagine racing at the peak lateral acceleration through a corner. When hitting the curb at the apex, this additional torque can be felt at the steering wheel. A low powered wheel at a lower clipping range would output 100% in both situations: before and during hitting the curb, hence it is not felt in the wheel.

Setting this torque threshold explicit in-game is only possible with iRacing (as far as I know). To do this, simply change a setting in the iRacing’s \texttt{app.ini} (my Documents). Set \texttt{DisplayLinearInNm = 1} in order to change the FFB-slider from the uniteless number to a torque value in N m.

See figure 22 for a correlation between these two methods. Horizontal axis is the value of the normal iRacing FFB slider; vertical axis is the value in N m when set the \texttt{app.ini}-setting to \texttt{DisplayLinearInNm = 1}.

6 Troubleshooting

\textbf{MMos Tool keeps crashing} make sure vJoy is uninstalled (see section 4.7) properly. Unplug all controllers when starting this tool. Make sure you have the latest BIOS
version. If nothing helps a fresh installation of Windows helps.

7 Change Log

0.9 – December 10, 2014
Figure 23: Granity

A  Granity Screenshots

B  Disco Wiring
If Bernhard Berger API SW is used, use the PC RS-485 connection method, omit the STM32F4 Discovery method. Only ONE method is to be used!

If the STM32F4 Discovery Microcontroller is used, flash with MMos FW and connect this as per schematic. DO NOT use the API method then!!

Please note Line Colors used in J2.1 and J2.1 is based on 568A patch cable standard as found in Australia. Work according to pin numbers and NOT color codes if your cable is different. NEVER use a cross-over cable, you will cause damage to the ARGON Drive.
C Registration at Virtual Racing

Visit Virtual Racing to register and see figure 25 for an English translation.

D Disco Adapter Board Connections
Figure 24: STM32F4DISCOVERY Wiring
Figure 25: translation of VR-registration
**Belegungsplan für Adapterplatine**

**Übersicht**

- X1 – Flachbandkabel Verbindung zur Argon
- X2 – SPI Erweiterungsstecker für MCP Zusatzchip 1
- X3 – SPI2 Erweiterungsstecker für MCP Zusatzchip 2
- X4 – Pedalanschluss
- X5 – Anschluss für 16 Button (Low-Active/Masseaktiv)
- X6 – Anschluss für Gängeausgang zum Motor
- X7 – Anschluss für andere Motorenstufen, z.B. VSD-E
- X8 – Encoderausgang zur Argon
- X9 – Encodereingang vom Motor
- X10 – Lüfteranschluss 5V

**J1 – Jumper für PWM-Direction**

<table>
<thead>
<tr>
<th>X2</th>
<th>Pin #</th>
<th>Belegung</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+5V</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SCK</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MISO</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MOSI</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X3</th>
<th>Pin #</th>
<th>Belegung</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+3V</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SCK</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MISO</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MOSI</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X4</th>
<th>Pin #</th>
<th>Belegung</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+3V</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Data 1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>+3V</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Data 2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>+3V</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Data 3</td>
<td></td>
</tr>
</tbody>
</table>

**X5**

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Belegung</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+5V</td>
</tr>
<tr>
<td>2</td>
<td>Button</td>
</tr>
<tr>
<td>3</td>
<td>Button</td>
</tr>
<tr>
<td>4</td>
<td>Button</td>
</tr>
<tr>
<td>5</td>
<td>Button</td>
</tr>
<tr>
<td>6</td>
<td>Button</td>
</tr>
<tr>
<td>7</td>
<td>Button</td>
</tr>
<tr>
<td>8</td>
<td>Button</td>
</tr>
<tr>
<td>9</td>
<td>+3V</td>
</tr>
</tbody>
</table>

**X6**

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Belegung</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+3V</td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
</tr>
<tr>
<td>3</td>
<td>Data 1</td>
</tr>
<tr>
<td>4</td>
<td>+3V</td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
</tr>
<tr>
<td>6</td>
<td>Data 2</td>
</tr>
<tr>
<td>7</td>
<td>NC</td>
</tr>
<tr>
<td>8</td>
<td>NC</td>
</tr>
<tr>
<td>9</td>
<td>NC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X7</th>
<th>Pin #</th>
<th>Belegung</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PWM 1N</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PWM 2N/PWM Inv-Dir</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>PWM 3N</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>PWM 3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>PWM 1/PWM Dir</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>NC</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>NC</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>NC</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>NC</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X8/X9</th>
<th>Pin #</th>
<th>Belegung</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hall W</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Hall V</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Hall U</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>B+</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>B-</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>A-</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>A+</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>+5V Out</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Z-</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Z+</td>
<td></td>
</tr>
</tbody>
</table>

**Poti-Anschlussmöglichkeiten**

<table>
<thead>
<tr>
<th>Anschluss #</th>
<th>Belegung 1</th>
<th>Belegung 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+3V</td>
<td>GND</td>
</tr>
<tr>
<td>2</td>
<td>Data</td>
<td>Data</td>
</tr>
<tr>
<td>3</td>
<td>GND</td>
<td>+3V</td>
</tr>
</tbody>
</table>

**J1**

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Belegung</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>Dir</td>
</tr>
<tr>
<td>2-3</td>
<td>Inv-Dir</td>
</tr>
</tbody>
</table>