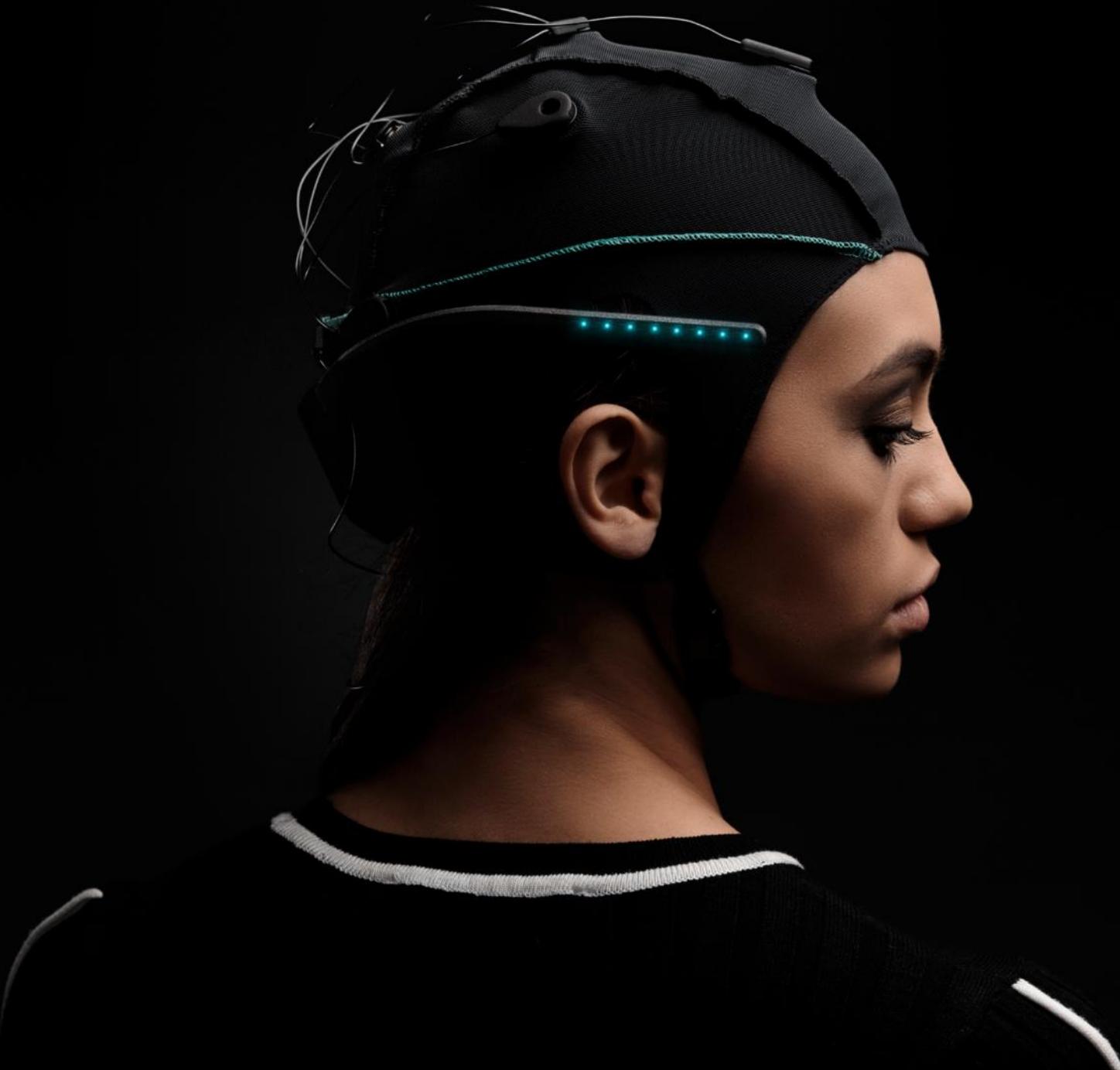




# unicorn

## HYBRID BLACK

USER MANUAL





User Manual for Unicorn Brain Interface Hybrid Black

Version Name: Unicorn Hybrid Black

Version Number: 1.18.00

Copyright © 2019 g.tec neurotechnology GmbH Austria

[www.unicorn-bi.com](http://www.unicorn-bi.com) | [hello@unicorn-bi.com](mailto:hello@unicorn-bi.com)



## CONTENTS

1.	SAFETY NOTICE .....	3
2.	INTRODUCTION .....	7
2.1.	Highlights.....	7
2.2.	Intended Use .....	7
2.3.	Release Notes.....	7
2.4.	Conditions of Use.....	7
2.4.1.	Operation and storage .....	7
3.	UNICORN HYBRID BLACK .....	8
3.1.	Unicorn Brain interface Hybrid Black.....	9
3.1.1.	Status LED.....	9
3.1.2.	Push Button .....	10
3.1.3.	LED bow .....	10
3.1.4.	Magnetic docking station .....	10
3.1.5.	Symbols.....	10
3.2.	Unicorn Hybrid EEG Electrodes .....	11
3.3.	Assembling/Disassembling.....	11
3.3.1.	Insert/Remove Unicorn Hybrid EEG Electrodes.....	12
3.3.2.	Connect/Disconnect Unicorn Hybrid EEG Electrodes .....	12
3.3.3.	Attach/Detach Unicorn sticky electrodes .....	13
3.3.4.	Attach/Detach Unicorn Brain Interface on the cap.....	13
3.4.	Charging the Unicorn Brain Interface .....	14
3.5.	Safe operation.....	15
3.6.	Setup the unicorn brain interface to record data .....	16
3.6.1.	Put the Unicorn Brain Interface on the subject's head .....	16
3.6.2.	Prepare the Unicorn Hybrid EEG Electrodes .....	16
3.6.3.	Avoid electrode and cable movements .....	18
3.6.4.	Avoid electrostatic charges .....	18
3.6.5.	Grounding .....	18
3.6.6.	Switch off .....	18
3.6.7.	Cleaning .....	19
3.6.8.	Packing and storage .....	19
3.7.	Cleaning the Unicorn Brain interface .....	19
3.7.1.	Cleaning the Unicorn Brain Interface and Unicorn hybrid EEG electrode cables.....	20



3.7.2. Cleaning the Unicorn cap and Unicorn hybrid EEG electrodes .....	20
3.7.3. Drying Unicorn caps and Unicorn hybrid EEG electrodes .....	21
4. GENERAL NOTES .....	22
4.1. Classification .....	22
4.2. Transportation and storage conditions .....	22
4.3. Location details.....	22
4.4. Waste disposal details .....	22
4.5. Warranty.....	22
5. DECLARATION OF CONFORMITY.....	23
6. UNICORN BRAIN INTERFACE PACKAGING .....	24
7. TECHNICAL SPECIFICATIONS.....	25
7.1. Unicorn Brain Interface .....	25
7.2. Unicorn Brain Interface Settings .....	25
7.2.1. Analog-Digital-Converter (ADC) .....	26
7.3. Motion tracking .....	26
7.4. RF module.....	26
7.4.1. FCC RF Radiation Exposure Statement.....	27
7.4.2. IC Statements .....	27
8. ELECTROMAGNETIC COMPATIBILITY (EMC) .....	28
9. SOFTWARE SYSTEM REQUIREMENTS .....	29
10. INSTALLATION.....	30
10.1. Install Unicorn Suite Hybrid Black.....	30
10.2. Uninstall Unicorn Suite Hybrid Black .....	30
11. SOFTWARE PREREQUISITES.....	31
11.1. Bluetooth configuration.....	31
11.1.1. Switch Bluetooth dongle .....	31
11.1.2. Disable Bluetooth power saving mode .....	32



11.2.	Optimize the Power Plan .....	33
12.	UNICORN SUITE HYBRID BLACK.....	35
12.1.	Apps.....	35
12.2.	Development Tools.....	36
12.3.	Licensing .....	37
12.4.	My Unicorn .....	40
13.	UNICORN C API .....	42
13.1.	Requirements.....	42
13.2.	Files on your computer .....	42
13.3.	Using the Unicorn C API.....	42
13.4.	Setting up a project using Visual Studio 2015 .....	42
13.5.	Command Order .....	46
13.6.	Unicorn C API – C Reference .....	47
13.6.1.	Constants .....	47
13.6.2.	Error Codes.....	48
13.6.3.	Type definitions .....	49
13.6.4.	Structures .....	50
13.6.5.	Functions.....	52
14.	UNICORN .NET API .....	59
14.1.	Requirements.....	59
14.2.	Files on your computer .....	59
14.3.	Setting up a project using Visual Studio 2015 .....	60
14.4.	Command Order .....	63
14.5.	Unicorn .NET API – .NET Reference .....	64
14.5.1.	Class DeviceException .....	64
14.5.2.	Class Unicorn .....	64
15.	UNICORN PYTHON API.....	75
15.1.	Requirements.....	75



15.2.	Files on your computer .....	75
15.3.	Using the Unicorn Python API .....	76
15.3.1.	Activate License .....	76
15.3.2.	Set library path.....	76
15.4.	Setting up a project using Visual Studio 2015 .....	77
15.5.	Command Order .....	78
15.6.	Unicorn Python API – Reference.....	80
15.6.1.	Constants .....	80
15.6.2.	Exceptions.....	82
15.6.3.	Structures.....	82
15.6.4.	Static Functions.....	84
15.6.5.	Class Unicorn .....	85
16.	UNICORN SIMULINK INTERFACE.....	90
16.1.	Requirements.....	90
16.2.	Files on your computer .....	90
16.3.	Using the Unicorn Simulink Interface .....	91
16.3.1.	Activate License .....	91
16.3.2.	Set library path.....	91
16.4.	Unicorn Simulink Interface .....	92
16.4.1.	Unicorn Block.....	93
16.4.2.	Examples .....	95
17.	UNICORN RECORDER .....	97
17.1.	Starting Unicorn Recorder from the Unicorn Suite .....	97
1.	Control Bar .....	99
1.1.	Start/Stop Acquisition.....	99
1.2.	Record .....	99
1.3.	Settings .....	99
1.4.	Help.....	100
2.	Processing and display settings .....	101
2.1.	Amplitude Range.....	101
2.2.	Bandpass Filter .....	101
2.3.	Notch Filter .....	101
3.	Data Viewer.....	102
4.	Signal Quality Scope .....	102
17.2.	Recordings .....	103
17.2.1.	Matlab Tools .....	103



18.	UNICORN SPELLER.....	105
18.1.	Requirements for Text-To-Speech Output.....	105
18.2.	Files on your computer .....	106
18.3.	Terms .....	107
18.4.	Mounting Unicorn Hybrid EEG Electrodes.....	108
18.5.	Starting Unicorn Speller .....	109
18.5.1.	Starting Unicorn Speller from the Unicorn Suite.....	109
18.5.2.	Starting Unicorn Speller via command line.....	110
18.5.3.	Behavior of Unicorn Speller on startup .....	112
18.6.	Unicorn Speller .....	112
18.6.1.	Status indicators .....	113
18.7.	Establish a connection to the Unicorn Brain Interface .....	115
18.7.1.	connection via Bluetooth.....	115
18.7.2.	Loss of connection.....	118
18.7.3.	No Connection between Unicorn Speller and Unicorn Brain Interface.....	118
18.8.	Check signal quality.....	118
18.8.1.	The data viewer.....	118
18.8.2.	Integrated signal quality criteria .....	122
18.9.	Calibrate Unicorn Speller.....	123
18.10.	Write with Unicorn Speller .....	126
18.11.	Items with specially linked actions in the StandardBoard.ibc and StandardFaceBoard.ibc file....	128
18.11.1.	Configuring item: Select completion suggestion.....	130
18.11.2.	Configuring item: Save .....	130
18.11.3.	Configuring item: Speech output .....	131
18.12.	Specify the settings for flashing .....	133
18.13.	Specify the settings for item selection .....	134
18.13.1.	Selection method: Standard .....	135
18.13.2.	Selection method: Dynamic.....	136
18.14.	Customizing the board and configuring the timing of flashing .....	136
18.14.1.	Overall configuration of the board and timing .....	138
18.14.2.	Editing the content of the board .....	139
18.15.	Word prediction.....	141
18.15.1.	Using word prediction.....	141
18.15.2.	Configuring word prediction.....	142
18.15.3.	Creating custom dictionaries with the Unicorn Speller Dictionary Builder .....	144



18.16.	Configuring network output.....	146
18.17.	Programming applications using Unicorn Speller's network output.....	151
19.	UNICORN SPHERO.....	159
19.1.	Pair with a Sphero SPRK+.....	159
19.2.	Configuring Unicorn Sphero .....	160
19.3.	Configure Unicorn Speller.....	161
1.1.1.	Load the Unicorn Sphero board into Unicorn Speller .....	161
1.1.2.	Enable Unicorn Speller network output.....	162
19.4.	Board items.....	163
20.	UNICORN PAINTING .....	170
20.1.	Configuring Unicorn Painting.....	170
20.2.	Configure Unicorn Speller .....	171
1.1.3.	Load the Unicorn Painting board into Unicorn Speller .....	171
1.1.4.	Enable Unicorn Speller network output.....	172
20.3.	Board items.....	173



## GLOSSARY

Wording	Explanation
Unicorn Brain Interface	The amplifier (device) including battery and electrode connectors.
Unicorn Hybrid EEG Electrodes	The rubber electrodes of the Unicorn.
Unicorn Sticky Electrodes	Disposable electrodes for R and L clip connectors.
Unicorn Bluetooth Dongle	The Bluetooth dongle delivered with the Unicorn Brain Interface.
Unicorn USB Charging Cable	The USB cable delivered with the Unicorn Brain Interface.
Unicorn Cap	The EEG cap delivered with the Unicorn Brain Interface.
Unicorn Suite Hybrid Black	The Unicorn software environment.
Unicorn Hybrid Black	The Unicorn Brain Interface bundle referring to the version Hybrid Black. This bundle includes Unicorn Brain Interface Hybrid Black, 8 Unicorn Hybrid EEG Electrodes, 50 pieces Unicorn Sticky Electrodes, Unicorn Cap Size M, Unicorn USB Charging Cable, Unicorn Bluetooth dongle and the Unicorn Suite Hybrid Black.
R	The electrode clip for the Unicorn Sticky Electrode on the right mastoid.
L	The electrode clip for the Unicorn Sticky Electrode on the left mastoid.
Unicorn Gel	The conductive electrode gel for Unicorn Hybrid EEG Electrodes.
Unicorn .NET API	The Unicorn C# library.
Unicorn Python API	The Unicorn Python library.
Unicorn Simulink Interface	The Unicorn Simulink Interface application.
Unicorn block	The MATLAB Simulink block for the Unicorn.
Unicorn C API	The Unicorn C library.
Unicorn Speller	The P300 Spelling application.
Unicorn Sphero	The P300 based Sphero control application.



---

Unicorn Painting

The P300 based painting application.

---

Unicorn Recorder

The recording application for Unicorn Brain Interface.

---

Unicorn Hybrid Black User Manual

The user manual for the Unicorn Brain Interface version Hybrid Black.

---



## 1. SAFETY NOTICE

In order to use this product safely and fully understand all its functions, read this manual before using the product. Follow the instructions for use for the used PC and the connected devices for allowed environmental conditions. The used PC must not go to sleep, hibernate, turn off, or turn on the screensaver during a measurement.



### **WARNING:**

Conductive parts of all Unicorn Hybrid EEG Electrodes must not have contact with the earth or other conductive parts.



### **WARNING:**

Avoid electrostatic discharge impulses when handling the device or touching the Unicorn Hybrid EEG Electrodes.



### **WARNING:**

Pay attention to the precautions regarding electromagnetic compatibility.



### **WARNING:**

The operation of the device can be compromised within shielded rooms. In case of problems, relocate the receiving device or consult technical support.



### **WARNING:**

The operator must be familiar with the operation of the device and must operate the device according to the instructions for use.



### **WARNING:**

The device and its accessories must not be exposed to increased mechanical stress.



### **WARNING:**

Each time you use the device, you must first check the device and its accessories for possible damage to connectors, sockets and cables. Check the Unicorn Hybrid EEG electrode cable connections with special care and ensure that the electrode cables have no breaks or cracks. Any cables, connectors, accessories, or other parts of the equipment must be replaced immediately if damaged or not working correctly.

**WARNING:**

The Unicorn Brain Interface is not protected against electrical defibrillation - before defibrillation, the Unicorn Hybrid EEG Electrodes must be removed from the subject!

**WARNING:**

Only use accessories identified for use with this device.

**WARNING:**

The device is powered internally via a lithium-ion polymer accumulator. The accumulator must only be replaced by the manufacturer.

**NOTE:**

The Unicorn Brain Interface uses special lightweight, thin and highly flexible cables for the Unicorn Hybrid EEG Electrodes to provide high comfort and easy cap mounting, especially for multi-channel recording. These cables are sensitive and need to be treated with special care. Following some basic guidelines will prolong the lifetime of Unicorn Hybrid EEG Electrodes:

- ⌚ Never pull on Unicorn Hybrid EEG Electrodes cables.
- ⌚ Avoid knots in cables.
- ⌚ Do not soak cables and Unicorn Hybrid EEG Electrodes for more than 30 minutes.
- ⌚ Avoid exposure to direct sunlight or chemical agents.
- ⌚ Make sure that no Unicorn Gel remains on Unicorn Hybrid EEG Electrodes or cables after cleaning.
- ⌚ Protect the amplifier unit from contamination with Unicorn Gel, water or disinfectant.
- ⌚ Always make sure that Unicorn Hybrid EEG Electrodes, cables and Unicorn caps are completely dry before storing.
- ⌚ Don't cut, kink or pinch electrode cables; light bending is safe.

**NOTE:**

Unicorn Hybrid EEG Electrode and cable lifetime depends on proper usage, careful treatment and cleaning, and appropriate storage. The manufacturer will provide warranty replacement only if there is no visible physical damage to the parts, such as: damaged, broken or pinched cables; or damaged housings or connectors.

**WARNING:**

- ⌚ Do not use any detergent other than detergents mentioned in this manual!
- ⌚ Do not perform automated reprocessing in Washer Disinfectors (WD) or Endoscope Washer Disinfectors (EWD)!
- ⌚ Do not machine-wash!
- ⌚ Do not use a laundry dryer or other hot air devices!
- ⌚ Do not put into an ultrasonic bath!
- ⌚ Do not autoclave Unicorn caps or Unicorn Hybrid EEG Electrodes!

**NOTE:**

The manufacturer is responsible for the safety, performance and reliability of the Unicorn Brain Interface as supplied to the customer at the time of delivery. This responsibility expires if the Unicorn Brain Interface is changed. Please note the following:

- ⌚ Changes to the Unicorn Brain Interface must be performed by the manufacturer only, and service and repair must be performed by corresponding qualified personnel only.
- ⌚ The Unicorn Brain Interface must be used according to the instructions for use.

**NOTE:**

The Unicorn Brain Interface and its components have been tested and comply with the electromagnetic compliance limits for the Directive 2014/53/EU (radio equipment directive, RED). See the chapter on Electromagnetic compatibility. The equipment, if not installed and used in accordance with the instructions, may cause interference with other devices in the vicinity. If this equipment does interfere with other devices, which can be determined by turning the equipment off and on, try to correct the interference through one or more of the following measures:

- ⌚ Reorient or relocate the receiving device.
- ⌚ Increase the separation between the equipment.
- ⌚ Consult Unicorn technical support.

**WARNING:**

The Unicorn Brain Interface must not be used in dangerous conditions such as wet rooms or explosive environments. The relative humidity must be between 25 % and 80 %. The Unicorn Brain Interface must not be used in combination with any other high-frequency device. Using a high frequency device with the Unicorn Brain Interface can cause burning under the Unicorn Hybrid EEG Electrodes and could damage the Unicorn Brain Interface.

**WARNING:**

The Unicorn Brain Interface must not be used in humans with pace-makers or electrical stimulators.



**NOTE:**

The Unicorn Brain Interface uses the 2.4 GHz band for wireless transmission. Ensure that enough transmission bandwidth is available in your environment, since other devices might also use the same band (e.g. WiFi or other Bluetooth devices).



## 2. INTRODUCTION

The Unicorn Brain Interface is a consumer grade biosignal amplifier kit. It allows developers, artists and makers to integrate signals from the human body within their projects – ranging from simple display of the signals to designing and controlling attached devices and interacting with artistic installations, toys, computer programs or apps and more. The Unicorn Brain Interface acquires the EEG from eight Unicorn Hybrid EEG Electrodes. The Unicorn Brain Interface consists of the Unicorn Brain Interface Hybrid Black, Unicorn C Size M, Unicorn Hybrid EEG Electrodes, Unicorn USB Charging Cable and a Unicorn Bluetooth dongle to acquire data on a computer. The Unicorn Suite is the software environment, consisting of standalone applications and APIs to interface the Unicorn Brain Interface, acquire and process data and to perform BCI paradigms.

### 2.1. HIGHLIGHTS

- ⌚ EEG recordings without cable connection via radio signal
- ⌚ Bluetooth 2.1 interface
- ⌚ Hybrid electrodes for wet and dry measurements
- ⌚ 8 DC-coupled analog input channels with 24 Bit resolution
- ⌚ sampling rate of 250 Hz per channel
- ⌚ oversampling to achieve a high signal-to-noise ratio
- ⌚ input sensitivity of  $\pm 750$  mV
- ⌚ 3-axis accelerometer
- ⌚ 3-axis gyroscope

### 2.2. INTENDED USE

The Unicorn Brain Interface is intended for use in non-medical environment for non-medical applications. The Unicorn Brain Interface is used by developers, artists, makers and gamers in the user's environment.

### 2.3. RELEASE NOTES

Version Name	Version Number	Date	Changes
Unicorn Hybrid Black	1.18.00	6/7/19	Initial Release

### 2.4. CONDITIONS OF USE

#### 2.4.1. OPERATION AND STORAGE

- ⌚ Temperature: +5 to +40 °C
- ⌚ Relative humidity: 25 to 80 %, non-condensing
- ⌚ Atmospheric pressure: 700 to 1060 hPa



### 3. UNICORN HYBRID BLACK

Bundle components	Quantity
Unicorn Brain Interface Hybrid Black	1
Unicorn USB Charging Cable	1
Unicorn Hybrid EEG Electrodes	8
Unicorn Sticky Electrodes	50
Unicorn Cap Size M	1
Unicorn Bluetooth Dongle	1
Unicorn Box	1
Unicorn Suite Hybrid Black	1





### 3.1. UNICORN BRAIN INTERFACE HYBRID BLACK



#### 3.1.1. STATUS LED

The color of the Status LED shows the actual battery voltage measured by the Unicorn Brain Interface.

Status LED	Battery Status
Cyan	Battery OK
Yellow	Battery Low
Red	Empty
Green	Charging

The Status LED shows the actual operation mode of the Unicorn Brain Interface.

Status LED	Operation Mode
Flashing	Bluetooth connection closed
Slow Blinking	Bluetooth connection established
Continuous	Acquisition running
Off	Device off and/or charging completed



### 3.1.2. PUSH BUTTON

- 💡 Pressing the pushbutton for more than 2 seconds turns the Unicorn Brain Interface ON if it is OFF.
- 💡 Pressing the pushbutton for more than 2 seconds turns the Unicorn Brain Interface OFF if it is ON.



#### NOTE:

If the Status LED is RED, the Unicorn Brain Interface does not turn on.

### 3.1.3. LED BOW

The LED bow comprises 8 LEDs that can be controlled with Unicorn APIs:

- 💡 Setting the digital output high turns the LED on
- 💡 Setting the digital output low turns the LED off

### 3.1.4. MAGNETIC DOCKING STATION

The Unicorn Brain Interface is attached to the Unicorn Cap by a magnetic docking station. To disassemble the Unicorn Brain Interface from the Unicorn cap, just lift it off the docking station. Be careful not to pull on the electrode leads.

For reassembly, just bring the Unicorn Brain Interface in the vicinity of the docking station and it will attach and center by itself. It is not possible to attach it upside down.

### 3.1.5. SYMBOLS

Symbol	Description
	CE mark
	Do not dispose with domestic waste. Dispose of it via the separate collection system for electrical and electronic equipment.
	Avoid electrostatic discharge
	Follow instructions for use
	Manufacturer



## S1

S1: permanent operation

**SN UN - 2018.12.01**

Serial number in the format:  
UN-YearOfProduction.Month.Number

FCC ID: QOQWT12

Contains FCC ID: QOQWT12

IC: 5123A-BGTWT12A

Contains IC: 5123A-BGTWT12A

### 3.2. UNICORN HYBRID EEG ELECTRODES

The Unicorn Hybrid EEG Electrodes are made of a conductive rubber that allow their operation as dry or wet Unicorn Hybrid EEG Electrodes.



Dry Unicorn Hybrid EEG Electrodes have the advantage of fast and easy preparation, but sometimes suffer from poor signal quality. If you encounter poor signal quality, the Unicorn Hybrid EEG Electrode can be used with special Unicorn Gel to reach the performance of a wet electrode.

To connect or disconnect the clip connector to/from the Unicorn Hybrid EEG Electrode, just slide it on or off.

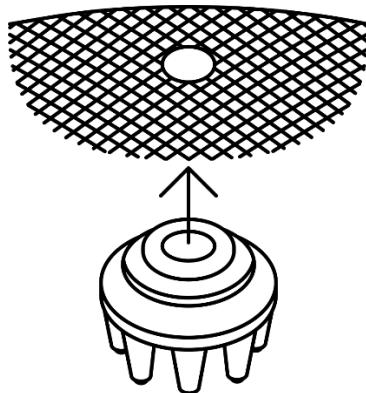
### 3.3. ASSEMBLING/DISASSEMBLING

The Unicorn Brain Interface is delivered disassembled. The system must be assembled properly to measure EEG. The following section describes how to assemble or disassemble the Unicorn.



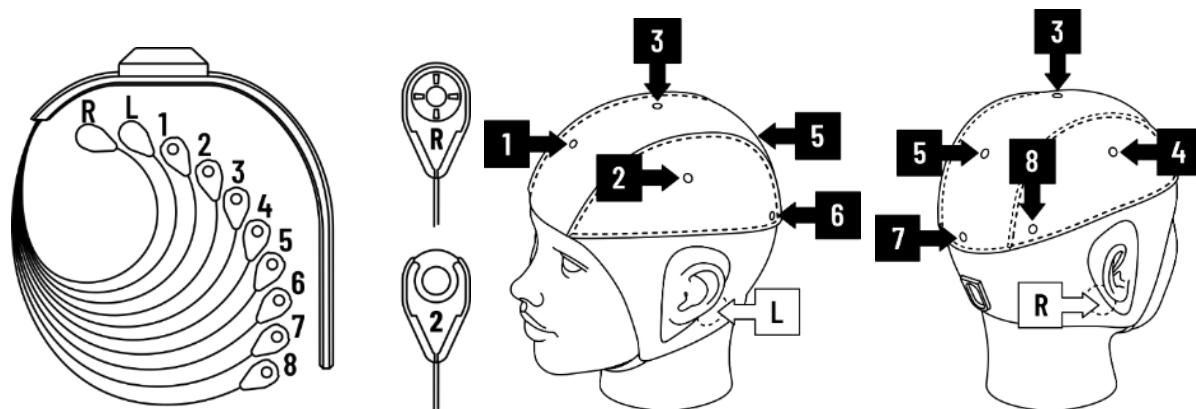
### 3.3.1. INSERT/REMOVE UNICORN HYBRID EEG ELECTRODES

The Unicorn cap features eight holes for holding eight Unicorn Hybrid EEG Electrodes. The first groove of the electrode is used to attach the electrode connector clip. The second groove is provided to insert and hold the electrode within a Unicorn cap. Stretch one of the predefined electrode holes of the Unicorn cap and insert the electrode until the second groove of the electrode is surrounded by fabric. Insert all eight Unicorn Hybrid EEG Electrodes into the predefined positions of the Unicorn cap.

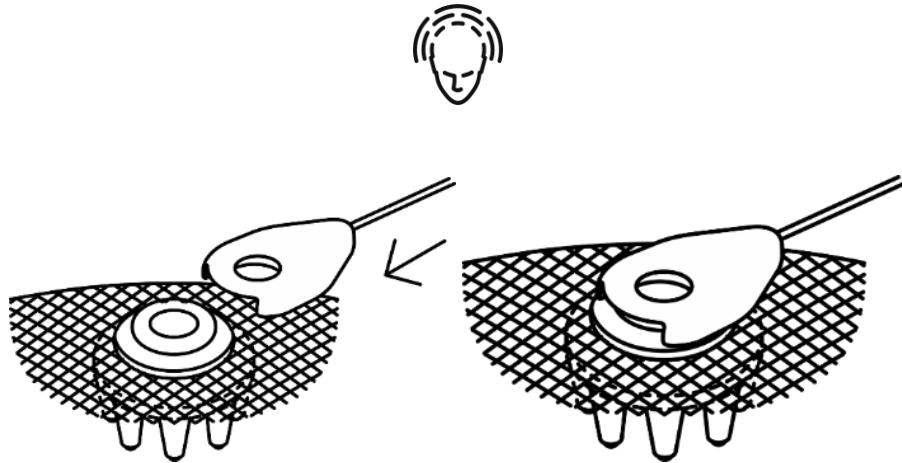


### 3.3.2. CONNECT/DISCONNECT UNICORN HYBRID EEG ELECTRODES

The bottom of each clip connector has a number or letter with the correct position. Attach the clip connectors numbered 1-8 by sliding them onto the first notch of the corresponding mounted Unicorn Hybrid EEG Electrodes on the cap. The following pictures show the correct positions of the electrode clips on the cap. The predefined electrode positions of the cap are Fz, C3, Cz, C4, Pz, PO7, Oz and PO8 according to the 10/20 system. The positions marked as L (for left) and R (for right) are for the Unicorn Sticky Electrodes and will be described in the following chapter.

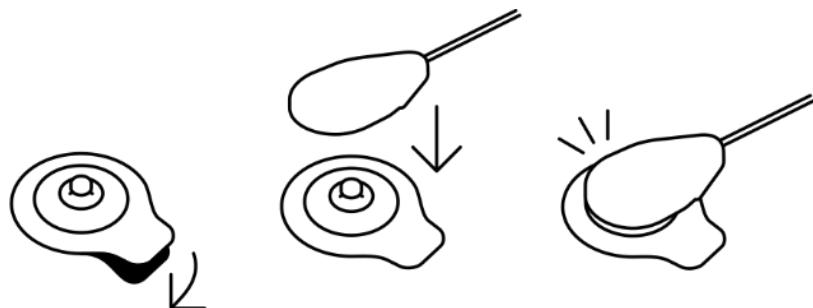


To connect or disconnect the clip connector to/from the Unicorn Hybrid EEG Electrode, just slide the clip connector on or off. Make sure that the clip surrounds the first groove of the electrode and that the holes of the electrode and electrode clip overlap.



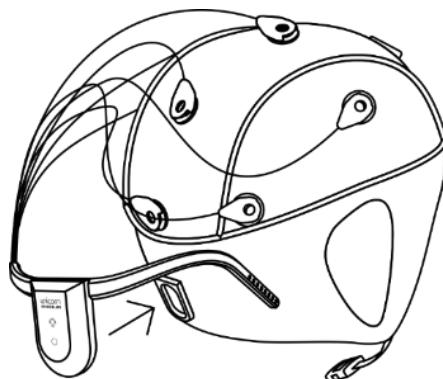
### 3.3.3. ATTACH/DETACH UNICORN STICKY ELECTRODES

R and L electrode feature a clip connector to attach or detach disposable Unicorn Sticky Electrodes. Put a Unicorn Sticky Electrode on each of the mastoid bones behind your ears (L and R positions seen in the figure above). Click the L and R electrode clips onto the corresponding Unicorn Sticky Electrodes. Remove the disposable Unicorn Sticky Electrodes by pulling the R and L clip connectors until the electrode is detached from the clip connector.



### 3.3.4. ATTACH/DETACH UNICORN BRAIN INTERFACE ON THE CAP

The Unicorn Brain Interface is attached to the Unicorn Cap by a magnetic docking station. To disassemble the Unicorn Brain Interface from the Unicorn cap, just lift it off the docking station. Be careful not to pull on the electrode leads. For reassembly, just bring the Unicorn Brain Interface in the vicinity of the docking station and it will attach and center by itself. It is not possible to attach it upside down.





### 3.4. CHARGING THE UNICORN BRAIN INTERFACE

For charging, lift off the Unicorn Brain Interface from the docking station to reach the USB port. Connect the provided Unicorn USB Charging Cable to the USB port of the Unicorn Brain Interface and to a free USB port of a PC, Notebook or appropriate wall adapter. Charging starts automatically. The accumulator cannot be overcharged, so the Unicorn Brain Interface can be connected to the supply mains without being damaged. When charging the Unicorn Brain Interface, the Status LED shines green. Charging is completed when the Status LED goes off.



**NOTE:**

The Unicorn has a built-in Lithium-Polymer accumulator. The accumulator may be partly discharged when the system is delivered.



**NOTE:**

As long as the USB port of the Unicorn Brain Interface is connected to a working USB port of a PC, Notebook or appropriate wall adapter, the Unicorn Brain Interface can't be turned on.



**NOTE:**

Charging an empty Unicorn battery will take about 3 hours.



**WARNING:**

Do not over-discharge the Unicorn battery. Over-discharging can damage the Unicorn battery or reduce the performance and life-time.



**WARNING:**

The Unicorn cell/battery would be at an over-discharged state by its self-discharge characteristics if the cell is not used for long time. Over long storage periods, batteries should be cycled every 90 days.



**WARNING:**

Store in a 50 % charged state. Do not store at fully charged state (4.2V) for a long period of time.



**WARNING:**

Expected Unicorn battery life cycle: The capacity after 300 cycles is expected to be equal to or more than 80 % of the rated capacity. The capacity after 500 cycles is expected to be equal to or more than 60 % of the rated capacity.



### **WARNING:**

Operating Temperature

- ⌚ Charging: 0 °C to 45 °C
- ⌚ Discharging: -20 °C to 60 °C

Storage Temperature

- ⌚ 1 year at -10 °C to 25 °C



### **WARNING:**

Electrical Specifications for charging

- ⌚ Voltage nominal input: 5 V DC
- ⌚ Voltage input min/max: 4.25 V - 7 V
- ⌚ Power input: 5 W

## **3.5. SAFE OPERATION**



### **WARNING:**

Use on healthy, intact skin only. Make sure no force is applied onto the Unicorn Hybrid EEG Electrodes to avoid any harm to the user.



### **WARNING:**

Do not use on or near open wounds, bruised or weakened skin, whether due to injury and/or other medical conditions of the subject.



### **WARNING:**

Do not use on subjects with a history of skin allergies or sensitivity to cosmetics and lotions.



### **WARNING:**

If rash, redness, itching, swelling, or abnormality appears on skin, remove the Unicorn cap and wash any residual Unicorn Gel off immediately.



### **WARNING:**

Instruct subjects to communicate any persistent redness, soreness or swelling at the Unicorn Hybrid EEG Electrodes sites.



### **WARNING:**

The Unicorn Cap should not be left in place for longer than three hours for any single recording session.



#### **WARNING:**

Avoid eye contact with Unicorn Gel, as it may cause mild transient irritation. Wash eyes with appropriate 0.9 % saline solution, commercial eye wash solution or water to flush out residual particles. Avoid rubbing eyes.



#### **WARNING:**

Unicorn Gel is essentially non-toxic; however, ingestion could result in nausea, vomiting and/or diarrhea. Drink 1 to 2 glasses of water if ingested.



#### **WARNING:**

Electrostatic discharge (ESD) events can harm electronic components inside your Unicorn Brain Interface. Under certain conditions, electric charge may build up on your body or an object, such as a peripheral, and then discharge into another object, such as your Unicorn Brain Interface. To prevent ESD damage, you should discharge static electricity from your body before you interact with any of your devices. You can protect against ESD and discharge static electricity from your body by touching a metal grounded object.

## **3.6. SETUP THE UNICORN BRAIN INTERFACE TO RECORD DATA**

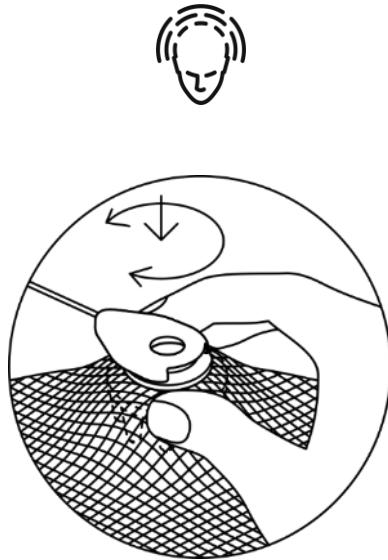
### **3.6.1. PUT THE UNICORN BRAIN INTERFACE ON THE SUBJECT'S HEAD**

Place the Unicorn Cap with the Unicorn Brain Interface on the subject's head to position the Unicorn Hybrid EEG Electrodes according to the electrode positions. Use a Unicorn Cap (S, M or L) that provides gentle pressure to all Unicorn Hybrid EEG Electrodes on the skin without causing pain to the subject. To properly mount the Unicorn Cap, measure the distance between the nasion and inion and the distance between the left and right preauricular points of the subject. The position at the middle of these two locations is the vertex position, position 3. Put on the Unicorn Cap and align this measured position with the position 3 of the Unicorn Cap. Now, the Unicorn Cap is in the correct position. Fix the Unicorn Cap with the chin strap to keep it in place.

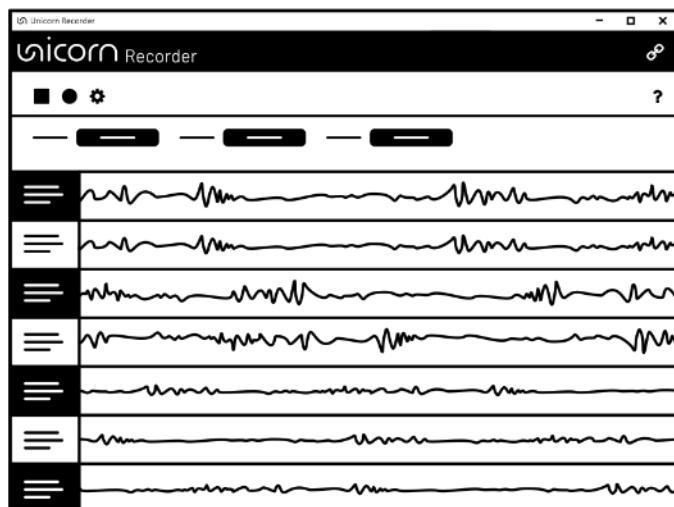
### **3.6.2. PREPARE THE UNICORN HYBRID EEG ELECTRODES**

The Unicorn Hybrid EEG Electrodes are attached to the Unicorn Cap. Before attaching the Unicorn Sticky Electrodes to the left (L) and right (R) bones behind the ears (processus mastoideus), the skin behind the ears should be cleaned with a soft cloth and a bit of medical alcohol (70 %). Connect the R (right mastoid) and the L (left mastoid) connectors to the Unicorn Sticky Electrodes.

After putting on and positioning the Unicorn Cap, each Unicorn Hybrid EEG Electrode needs to be twisted clockwise and counterclockwise a few times with slight pressure on the skin. Just grab the Unicorn Hybrid EEG Electrodes with two fingers from outside the Unicorn Cap to do so. This allows the pins to go through the hair and get in good contact with the skin.



After preparation of all Unicorn Hybrid EEG Electrodes, it might take 2-3 minutes until all channels display stable signals. If some Unicorn Hybrid EEG Electrodes do not show stable signals after a few minutes, you may need to repeat the twisting and/or check the environment and the correct pressure of all Unicorn Hybrid EEG Electrodes on the skin. To see clear EEG signals, it is recommended to use a bandpass filter (0.5 – 30 Hz or 2 – 30 Hz) and a 50/60 Hz notch filter.



#### **NOTE:**

EEG recordings in general are quite sensitive to a number of disturbing influences such as movements of the subject, the cables and the Unicorn Hybrid EEG Electrodes, or coupling of electromagnetic and electrostatic fields near the recording environment. For dry recordings especially, movements of the Unicorn Hybrid EEG Electrodes and electrostatic charges can cause significant artefacts in the signals.

Consider the following actions to avoid artefacts.



### 3.6.3. AVOID ELECTRODE AND CABLE MOVEMENTS

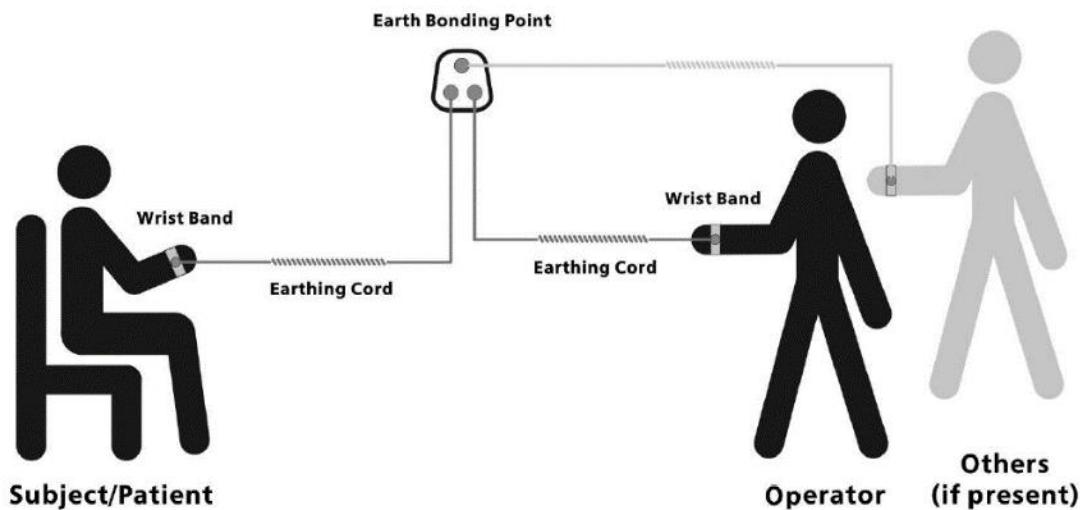
- ⌚ Ensure the subject is in a comfortable and relaxed position during recording.
- ⌚ Instruct the subject to stay in a relaxed position and to avoid movements.
- ⌚ Position electrode wires to avoid tension and direct contact with moving parts of the body or any other moving components, such as devices near the head.

### 3.6.4. AVOID ELECTROSTATIC CHARGES

- ⌚ Choose an antistatic environment/lab for high quality recordings.
- ⌚ Use a room with antistatic carpet or floor covering or wooden/stone floor instead of normal carpets or artificial parquet floors.
- ⌚ Furniture made of wood, metal and leather is preferable to any plastic materials and synthetic textiles.
- ⌚ Cotton clothing is preferred to wool and synthetics.
- ⌚ Avoid shoes with synthetic soles.

### 3.6.5. GROUNDING

- ⌚ Grounding of any additional equipment (using a potential equalization conductor) and metal-made furniture to a central earth/ground contact (lab ground) is recommended.
- ⌚ The subject as well as the operator(s) may be grounded to avoid electrostatic charges (e.g. using an antistatic wrist band connected to the lab ground).



### 3.6.6. SWITCH OFF

After stopping the data acquisition, you can turn off the Unicorn Brain Interface by again pressing the power button for at least 2 seconds. The LED should turn off and remain off.



### 3.6.7. CLEANING

Clean the Unicorn Cap and the Unicorn Hybrid EEG Electrodes immediately after use. Read cleaning instructions in the following chapter.

### 3.6.8. PACKING AND STORAGE

Never pack the system while any of the components remain wet or moist! Always allow all components to dry completely before packing or storage to avoid damage to the system. When packing the Unicorn Brain Interface in the protective case, make sure that no wires are being pinched or kinked.

## 3.7. CLEANING THE UNICORN BRAIN INTERFACE

#### NOTE:

When the EEG recording is finished, remove the Unicorn Cap and L and R clip connectors from the Unicorn Sticky Electrodes. Wipe the subject's forehead and mastoids with a tissue or cloth towel, and lightly wipe the hair. Brushing or combing the hair will remove all visual evidence of the Unicorn Gel.

#### NOTE:

The Unicorn Caps do not have an expiration date, aside from elasticity. Elasticity of the material dictates the life time of the Unicorn Cap. When a Unicorn Cap becomes stretched, it loses its elasticity. It must then be discarded and replaced, as the Unicorn Hybrid EEG Electrode placements will be inaccurate and artifacts may occur more often.

#### NOTE:

Keep the Unicorn Brain Interface away from any liquids! Liquids must not enter the Unicorn Brain Interface!

#### NOTE:

During cleaning and disinfection, the USB cable must be disconnected from the Unicorn Brain Interface.

#### NOTE:

Please note that devices that blow warm or hot air, such as hand or hair dryers, must not be used to dry Unicorn Caps, because it weakens the elastic material; the Unicorn Cap lifetime will be greatly shortened. Hanging the Unicorn Cap in front of a fan will accelerate drying.

#### NOTE:

Take care that the Unicorn Hybrid EEG Electrodes leads are not damaged during cleaning!



**NOTE:**

Do not use excessive force while cleaning.



**NOTE:**

The Unicorn Brain Interface must not come in contact with germicide.



**NOTE:**

It is the user's responsibility to qualify any deviations from the recommended method of processing. Users should note appropriate disclaimers if there are deviations. The manufacturer assumes no responsibility for damage to its products caused by the use of products or processes not recommended in this policy.



**NOTE:**

If questions arise, or more information is needed, please contact the manufacturer.

### **3.7.1. CLEANING THE UNICORN BRAIN INTERFACE AND UNICORN HYBRID EEG ELECTRODE CABLES**

Whenever the Unicorn Brain Interface after usage gets in contact with Unicorn Gel or other substances, it should be cleaned immediately. Use a damp cloth to clean the Unicorn Brain Interface. Do not submerge the Unicorn Brain Interface in water and prevent any liquids from entering the Unicorn Brain Interface! For disinfection, use cleaning and disinfection wipes (e.g. Mikrozid® sensitive wipes #70000807, Mikrozid® AF wipes #109203 from Schülke & Mayr GmbH, [www.schuelke.com](http://www.schuelke.com), or similar products).



**NOTE:**

The Unicorn Brain Interface and all cables coming with the system must not be submerged in liquids! Use disinfection wipes only to clean and disinfect these parts of the system.

### **3.7.2. CLEANING THE UNICORN CAP AND UNICORN HYBRID EEG ELECTRODES**

When used for dry EEG recordings, the Unicorn Hybrid EEG Electrodes can be cleaned without removing them from the Unicorn cap. For this purpose, use a soft cloth and 70 % isopropyl alcohol or disinfection wipes, as mentioned above. Clean the pins of all Unicorn Hybrid EEG Electrodes by wiping from different directions over the tips of the pins. When used as a wet electrode system, clean all Unicorn Hybrid EEG Electrodes inside and outside with a brush, until gel is removed.

For cleaning and disinfection of the Unicorn Cap, the Unicorn Hybrid EEG Electrodes may remain in the Unicorn Cap. The Unicorn Cap itself can be cleaned in warm water (30 – 35 °C) with some shampoo, soap



or multistage enzymatic cleaner (e.g. Sekusept® MultiEnzyme P #3046670 from ECOLAB, [www.ecolab.com](http://www.ecolab.com), or a comparable product).

Do not soak the Unicorn Cap for more than 10 minutes, and rinse it carefully with clear water after cleaning. For disinfection, use a low-level disinfectant based on Glucoprotamin (e.g. Sekusept® PLUS #3011100 from ECOLAB, [www.ecolab.com](http://www.ecolab.com), or a similar product). Follow the instructions from the manufacturer of the specific product you use regarding concentration and duration of the treatment. Use the lowest possible concentration and the shortest treatment time to keep the Unicorn cap in a good condition for a long time.



**NOTE:**

The clip connectors should not be submerged in disinfectant or other liquids. Use disinfection wipes to disinfect these parts of the system!

### **3.7.3. DRYING UNICORN CAPS AND UNICORN HYBRID EEG ELECTRODES**

Rinse the Unicorn Cap carefully with clear water after cleaning. Then, put one towel inside the Unicorn Cap and another one around the outside of the Unicorn Cap. The towels will soak up most of the liquid and allow the Unicorn Cap and Unicorn Hybrid EEG Electrodes to dry quickly. Put the Unicorn Cap in a well-ventilated place to dry.

Follow these rules to keep the Unicorn Cap in good condition for a long time:

- ⌚ Do not dry the Unicorn Cap in a stretched position, as it might lose elasticity.
- ⌚ Do not use hot air (e.g. from a hair dryer) to speed up drying.
- ⌚ Do not put the Unicorn Cap in a laundry dryer.
- ⌚ Do not iron the Unicorn Cap.
- ⌚ Never pack and store a Unicorn Cap before it completely dries.



**NOTE:**

The Unicorn Caps may bleach out a bit after frequent cleaning and disinfection. As long as the fabric remains elastic and the signs on the Unicorn cap are readable, replacement is not necessary. However, Unicorn Caps and Unicorn Hybrid EEG Electrodes are consumables and need to be replaced from time to time. Careful treatment and cleaning will prolong the life-time of your Unicorn Caps and Unicorn Hybrid EEG Electrodes significantly.



## 4. GENERAL NOTES

### 4.1. CLASSIFICATION

- ⌚ Protection against mechanical distortion and liquids: IP40
- ⌚ Operation mode: S1 (Permanent operation)

### 4.2. TRANSPORTATION AND STORAGE CONDITIONS

The Unicorn Brain Interface can be stored at temperatures between  $-20^{\circ}$  to  $+45^{\circ}$  Celsius. The relative humidity must be between 25 % and 80 %. If there is any condensed water, wait until it disappears before use (wait at least 1 h in a heated room).

### 4.3. LOCATION DETAILS

Do not use the Unicorn Brain Interface near a heating system or directly in the sun. During operation, the outside temperature should be between  $+5^{\circ}$  Celsius and  $+35^{\circ}$  Celsius and the air pressure between 700 and 1060 hPa.

### 4.4. WASTE DISPOSAL DETAILS

Bring the Unicorn Brain Interface to a recycling center or sent it back to the manufacturer.

### 4.5. WARRANTY

Warranty in the EU is 6 month and 30 days in other countries for the Unicorn Brain Interface. The Unicorn Hybrid EEG Electrodes, the Unicorn Gel and the disposable Unicorn Sticky Electrodes are consumables. Only use parts from g.tec to operate the Unicorn Brain Interface. Warranty is invalidated if anyone except a g.tec employee opens or disassembles any components of the Unicorn. Warranty only applies for properly used devices. Please note that any damage resulting from improper treatment of the system will not be covered by the warranty. This may include broken, kinked or damaged wires and cables, damaged isolators and enclosures.

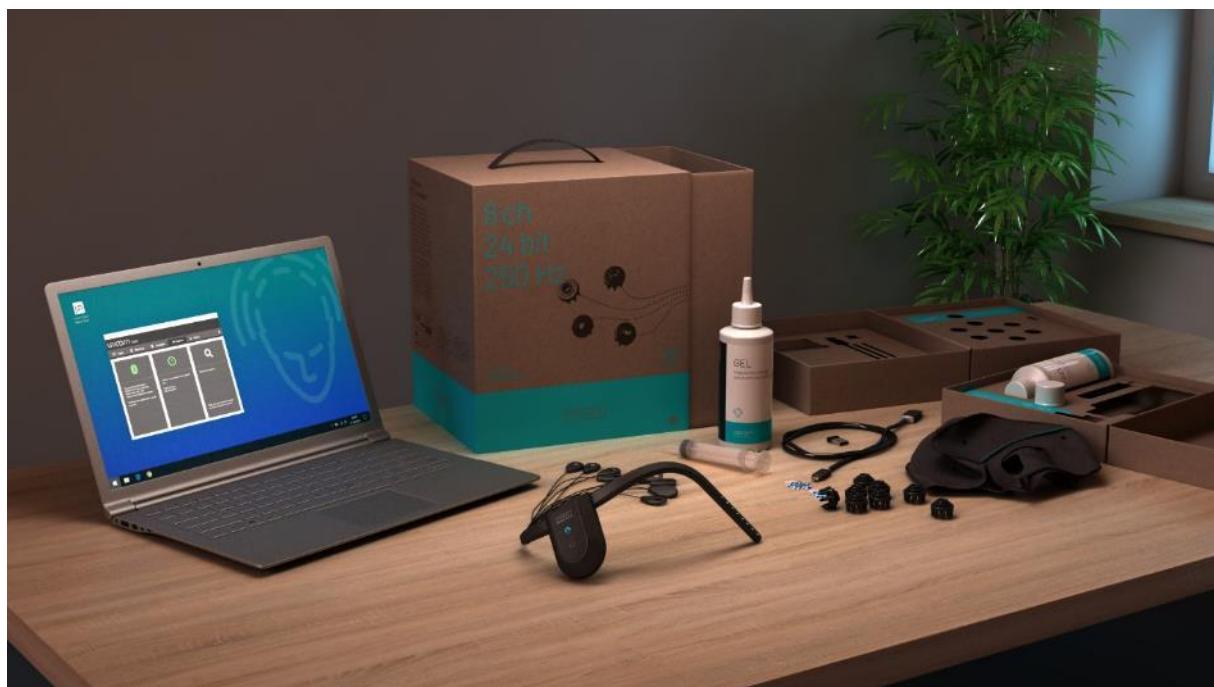


## 5. DECLARATION OF CONFORMITY

The declaration of conformity is available on request.



## 6. UNICORN BRAIN INTERFACE PACKAGING





## 7. TECHNICAL SPECIFICATIONS

### 7.1. UNICORN BRAIN INTERFACE

Technical Specifications	
Model	Unicorn Hybrid Black
Type	8-channel amplifier
Battery	LP-422339-PACK, 350 mAh, IEC 62133 (ed.2)
Rated power consumption	0.3W
Rated DC voltage	3.7 V
Rated current of fuse	Little fuse 0467001.NR (1.0 A)
Rated voltage of fuse	32 V
Manufacturer	g.tec neurotechnology GmbH Sierningstrasse 14 4521 Schiedlberg, Austria

### 7.2. UNICORN BRAIN INTERFACE SETTINGS

Technical Specifications	
Channels	1 to 8 and R channel
Sensitivity	± 750 mV
Highpass	0 Hz
Lowpass	10.23 kHz
Input impedance	>100 MΩ



### 7.2.1. ANALOG-DIGITAL-CONVERTER (ADC)

Technical Specifications	
Resolution	24 Bit
Sampling frequency	250 Hz
Number of ADCs	8

## 7.3. MOTION TRACKING

Technical Specifications	
Acceleration range	± 8 g setting in x, y and z directions
Acceleration bandwidth	44.8 Hz
Gyroscope range	± 1000 °/s setting in x, y and z directions
Gyroscope bandwidth	41 Hz

## 7.4. RF MODULE

Technical Specifications	
Operating Frequency range	2400.0 ... 2483.5 MHz (ISM Band)
Transmit power Max	+3 dBm
Receiver sensitivity	-86 dBm
Compliance	Bluetooth specification, version 2.1 + EDR
Marking	CE, FCC, IC, Japan and South-Korea
FCC ID	QOQWT12
IC	5123A-BGTWT12A
Japan	R 209- J00036



This Unicorn Brain Interface complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this Unicorn Brain Interface may not cause harmful interference, and (2) this Unicorn Brain Interface must accept any interference received, including interference that may cause undesired operation.

#### **7.4.1. FCC RF RADIATION EXPOSURE STATEMENT**

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. End users must follow the specific operating instructions for satisfying RF exposure compliance. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

#### **7.4.2. IC STATEMENTS**

This Unicorn Brain Interface complies with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions: (1) this Unicorn Brain Interface may not cause interference, and (2) this Unicorn Brain Interface must accept any interference, including interference that may cause undesired operation of the Unicorn Brain Interface.

Under Industry Canada regulations, this radio transmitter may only operate using an antenna of a type and maximum (or lesser) gain approved for the transmitter by Industry Canada. To reduce potential radio interference to other users, the antenna type and its gain should be chosen so that the equivalent isotropically radiated power (e.i.r.p.) is not more than that necessary for successful communication.



##### **NOTE:**

The Unicorn Brain Interface uses the 2.4 GHz band for wireless transmission. Ensure that enough transmission bandwidth is available in your environment, since other devices might also use the same band (e.g. WiFi or other Bluetooth devices). Use wireless screening tools to ensure the availability of the necessary transmission channel.



## 8. ELECTROMAGNETIC COMPATIBILITY (EMC)

The EMC declaration is available on request



## 9. SOFTWARE SYSTEM REQUIREMENTS

Minimum hardware requirements to run the Unicorn Suite.

Hardware	Properties
CPU	2 GHz or faster processor
Hard disk	20-30 GB
RAM	4 GB
Bluetooth	Bluetooth Adapter with Bluetooth 2.1 + EDR support

Minimum software requirements to run the Unicorn Suite.

Software	Properties
Operating System	Windows 10 Pro
	English
	64-bit
PDF Reader	Acrobat Reader DC 2015



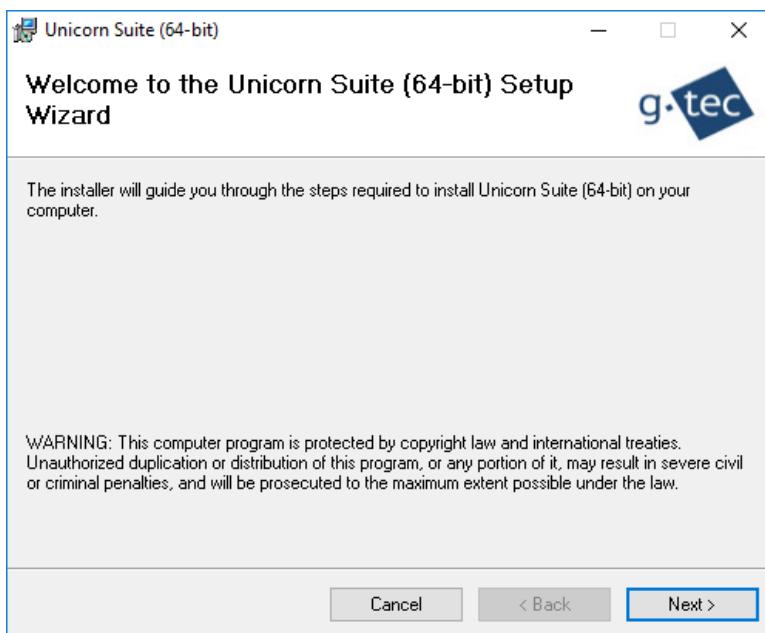
## 10. INSTALLATION

The following section describes how to install Unicorn Suite.

### 10.1. INSTALL UNICORN SUITE HYBRID BLACK

Perform the following steps for installation:

1. If there is an old version of the Unicorn Suite package on the computer, please uninstall it.
2. Close all running applications.
3. Open the Unicorn Suite directory, select the correct directory for the architecture of the PC (Win64). To install the Unicorn Suite, run setup.exe. If setup asks you for installation of the .NET Framework, confirm the dialog by clicking the Accept button. The installer will install the .NET Framework.
4. Follow the instructions on the screen. If User Account Control is turned on, additional dialogs may ask for permission. Confirm the dialogs to allow installation of the Unicorn Suite software through User Account Control.



5. Choose the installation folder (default is C:\Program Files\gtec\ ) where the install routine copies all necessary driver files and press Next.
6. Follow the instructions on the screen. When the following window informs you about completion of the installation, click Close to complete.

### 10.2. UNINSTALL UNICORN SUITE HYBRID BLACK

Remove Unicorn Suite using the standard uninstall process of Windows accessed via the Control Panel.

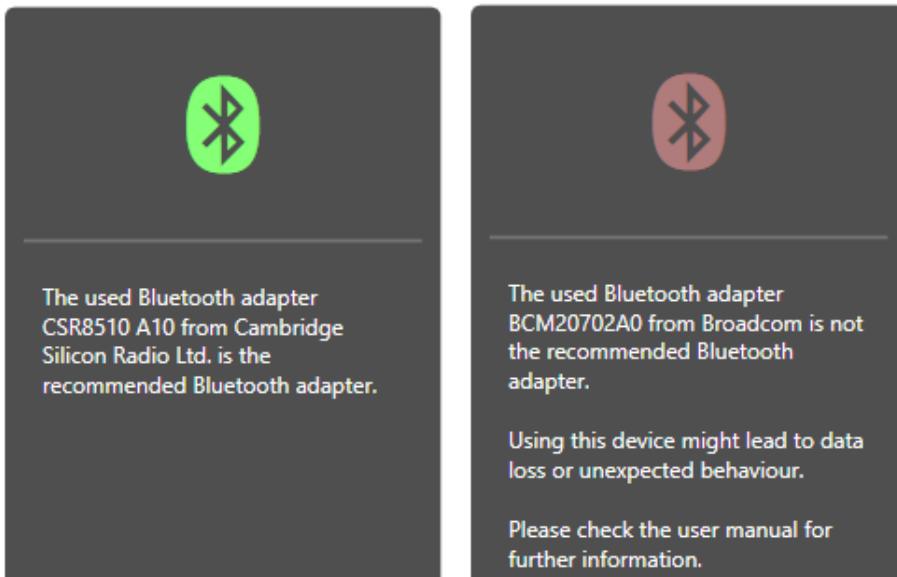


## 11. SOFTWARE PREREQUISITES

### 11.1. BLUETOOTH CONFIGURATION

Most computers are delivered with an internal Bluetooth adapter. The Unicorn Brain Interface is tested and delivered with a recommended Unicorn Bluetooth adapter. To avoid data loss and unexpected behavior, the Unicorn Brain Interface should be used with the Unicorn Bluetooth dongle. Insert the Unicorn Bluetooth dongle into an USB slot and open the Unicorn Suite. Open the "My Unicorn" tab.

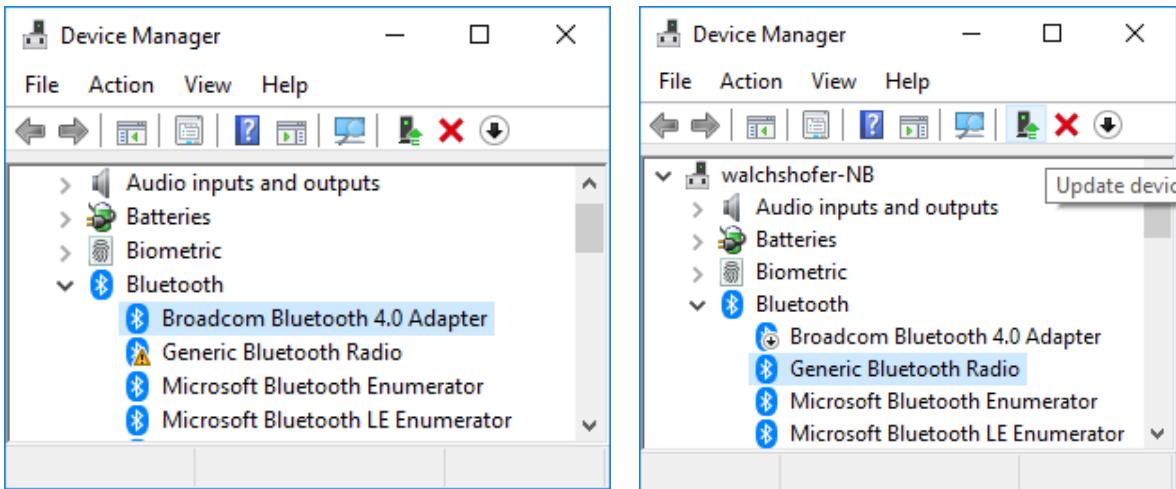
- ⌚ If the Bluetooth symbol is colored green, the recommended Bluetooth adapter is in use and configured properly.
- ⌚ If the Bluetooth symbol is blinking red, either the wrong Bluetooth adapter is used or it is not configured correctly.
- ⌚ The delivered Bluetooth dongle should be named "CSR8510 A10".



#### 11.1.1. SWITCH BLUETOOTH DONGLE

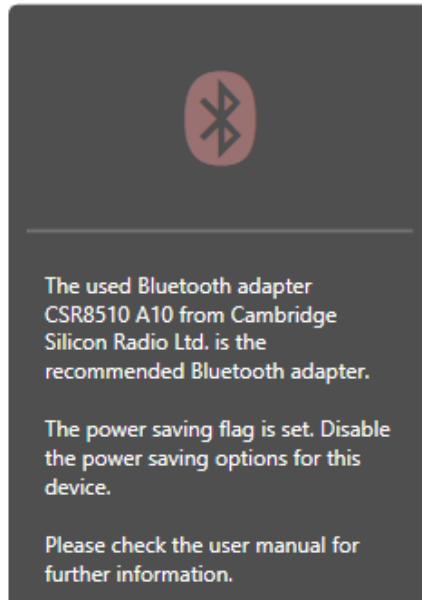
Open the "Device Manager" and go to the Bluetooth section. The Unicorn Bluetooth dongle should be listed as "Generic Bluetooth Radio". If the Generic Bluetooth Radio symbol features a small warning sign, the Unicorn Brain Interface is not in use or not working properly.

If there are multiple Bluetooth devices, the computer has an internal Bluetooth adapter, which is currently in use. Disable the internal Bluetooth adapter by right clicking the device and selecting "Disable device". Reinject the Unicorn Bluetooth dongle. The Unicorn Bluetooth dongle should be used now.

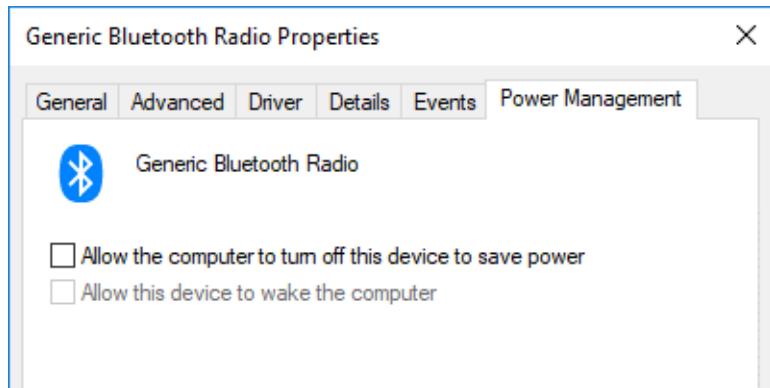


### 11.1.2. DISABLE BLUETOOTH POWER SAVING MODE

Open the Unicorn Suite and go to the "My Unicorn" tab if the delivered Bluetooth dongle is in use. The delivered Bluetooth dongle should be named "CSR8510 A10". If the delivered dongle is in use but still blinking red, the power saving mode is enabled.



Open the "Device Manager" and go to the Bluetooth section. Open the Bluetooth properties of the "Generic Bluetooth Radio". Go to the "Power Management" tab. Deselect the "Allow the computer to turn off this device to save power" option to disable the power saving mode.

**NOTE:**

Do not use any other Bluetooth dongle than the Unicorn Bluetooth dongle.

## 11.2. OPTIMIZE THE POWER PLAN

To ensure that the computer does not go to sleep or hibernate while acquiring data or performing a paradigm, the "Power Options" should be optimized. Modify the power plan settings as following:

The screenshot displays two windows titled "Power Options" side-by-side, illustrating how to modify power plan settings.

**Left Window (Processor Power Management):**

- Section: Advanced settings
- Text: Select the power plan that you want to customize, and then choose settings that reflect how you want your computer to manage power.
- Power plan: High performance [Active]
- Processor power management:
  - Minimum processor state
    - On battery: 100%
    - Plugged in: 100%
  - System cooling policy
  - Maximum processor state
    - On battery: 100%
    - Plugged in: 100%
  - Display
  - Multimedia settings
  - Battery
- Buttons: Restore plan defaults, OK, Cancel, Apply

**Right Window (Wireless Adapter Settings):**

- Section: Advanced settings
- Text: Select the power plan that you want to customize, and then choose settings that reflect how you want your computer to manage power.
- Power plan: High performance [Active]
- Wireless Adapter Settings:
  - Power Saving Mode
    - On battery: Maximum Performance
    - Plugged in: Maximum Performance
  - Sleep
  - USB settings
  - Intel(R) Graphics Settings
  - Power buttons and lid
  - PCI Express
  - Processor power management
  - Display
- Buttons: Restore plan defaults, OK, Cancel, Apply



### Modify the hard disk power settings

The screenshot shows the 'Power Options' dialog with 'Advanced settings' selected. Under the 'High performance [Active]' plan, the 'Hard disk' section is expanded, showing 'Turn off hard disk after' settings for both battery and AC power. Other collapsed sections include Internet Explorer, Desktop background settings, Wireless Adapter Settings, Sleep, USB settings, and Intel(R) Graphics Settings.

### Modify the sleep settings

The screenshot shows the 'Power Options' dialog with 'Advanced settings' selected. Under the 'High performance [Active]' plan, the 'Sleep' section is expanded, showing 'Sleep after' settings for both battery and AC power. Other collapsed sections include Allow hybrid sleep, Hibernate after, and Allow wake timers.

### Modify the USB settings

The screenshot shows the 'Power Options' dialog with 'Advanced settings' selected. Under the 'High performance [Active]' plan, the 'USB settings' section is expanded, showing 'USB selective suspend setting' for both battery and AC power. Other collapsed sections include Internet Explorer, Desktop background settings, Wireless Adapter Settings, Sleep, and Intel(R) Graphics Settings.

### Modify display settings

The screenshot shows the 'Power Options' dialog with 'Advanced settings' selected. Under the 'High performance [Active]' plan, the 'Display' section is expanded, showing 'Turn off display after' settings for both battery and AC power. Other collapsed sections include Processor power management, Multimedia settings, and Battery.

Make sure that the screen saver is turned off.



## 12. UNICORN SUITE HYBRID BLACK

### 12.1. APPS

The Apps tab features available standalone applications for the Unicorn Brain Interface. Applications can be used if a valid license is present. See chapter 12.3 Licensing for more information about the licensing system. Applications can be started with a paired Unicorn Brain Interface and will establish the connection to the device automatically. See chapter 12.4 My Unicorn for more information about pairing with Unicorn Brain Interfaces and setting up the system environment.

The screenshot shows the 'Unicorn Suite Hybrid Black' application window. The title bar displays the application name. The menu bar includes 'File', 'Apps', 'DevTools', 'Licenses', 'My Unicorn', 'About', and a question mark icon. The main content area has a dark background. On the left, there is a sidebar with four items: 'Recorder' (selected), 'Speller', 'Sphero', and 'Painting'. The 'Recorder' item is highlighted with a light blue background. To the right of the sidebar, the 'Unicorn Recorder' application is displayed. The title 'Unicorn Recorder' is at the top, followed by a descriptive text block: 'The Unicorn Recorder allows to visualize and record brain waves from 8 positions on the head (Fz, C3, Cz, C4, Pz, P07, Oz, P08). The data can be filtered to suppress artifacts or to extract certain brain frequencies like alpha and beta. Data are stored in CSV format and can be read by self-made programs or can be loaded with the off-line processing toolbox g.BSanalyze from g.tec.' At the bottom of the main window, there is a status bar with the text 'UN-2018.12.01' and a dropdown arrow, and a large 'Open' button.



## 12.2. DEVELOPMENT TOOLS

The DevTools tab features application programming interfaces (API) for the Unicorn. Free APIs can be opened directly by clicking the "Open" button. The API library folder will be opened automatically. Licensed APIs require a license to be unlocked. Follow section 12.3 Licensing to get more information about the licensing system.

A screenshot of the Unicorn Suite software interface. The window title is "Unicorn Suite". The main menu bar includes "Unicorn Suite", "Apps", "DevTools" (which is selected and highlighted in blue), "Licenses", "My Unicorn", "About", and "?". On the left, a sidebar lists "C API", ".NET API", "Python API", and "Simulink Interface". The main content area is titled "Unicorn C API" and contains the text: "The Unicorn comes with application programming interfaces that allow you to create your own applications with the device. Use the C API to create your own C/C++ applications." At the bottom right of the content area is a black "Open" button.

Unicorn Suite

Unicorn Suite

Apps DevTools Licenses My Unicorn About ?

C API .NET API Python API Simulink Interface

Unicorn C API

The Unicorn comes with application programming interfaces that allow you to create your own applications with the device. Use the C API to create your own C/C++ applications.

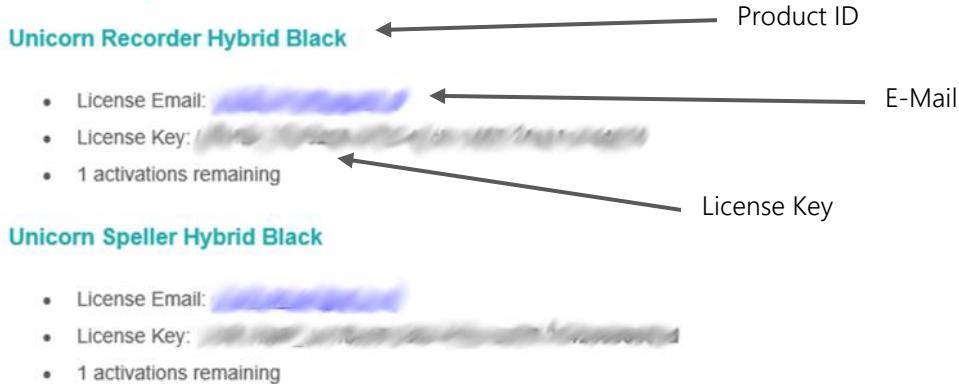
Open



### 12.3. LICENSING

The "Licenses" tab manages licenses for applications and development tools. Licenses can be purchased at [www.unicorn-bi.com](http://www.unicorn-bi.com). It is possible to activate, deactivate and check the status of licenses within the "Licenses" dialog. After purchase, licenses are delivered via e-mail. All information required to activate a license is delivered with the e-mail.

#### License Keys



Click "Add license" to activate a license key. Use "Product Id", "License key" and "License Email" from the received email to complete the dialog. Click "Activate" to enable the license. If the activation succeeded, the license is listed as installed license. Check your internet connection and license information if the license key could not be activated.

If the license server is reachable, licenses are refreshed whenever the Unicorn Suite is started. Applications and development tools can also be used offline if the license was refreshed within one week. Afterwards, an online license refresh is required. The License expiration date determines the expiration date of the license for a specific product.



Unicorn Suite

# unicorn Suite

Apps DevTools Licenses My Unicorn About ?

Show licenses Add license

After purchasing the software license please enter the information below that you received per email to activate the software.

Product Id:

License Key:

Email:

Activate

To deactivate a license, select the product to deactivate from the installed licenses list and click "Deactivate". The license will be removed from the installed licenses list if the deactivation succeeded.



Unicorn Suite Hybrid Black

# unicorn Suite Hybrid Black

Apps DevTools Licenses My Unicorn About ?

Show licenses Add license

### Installed Licenses

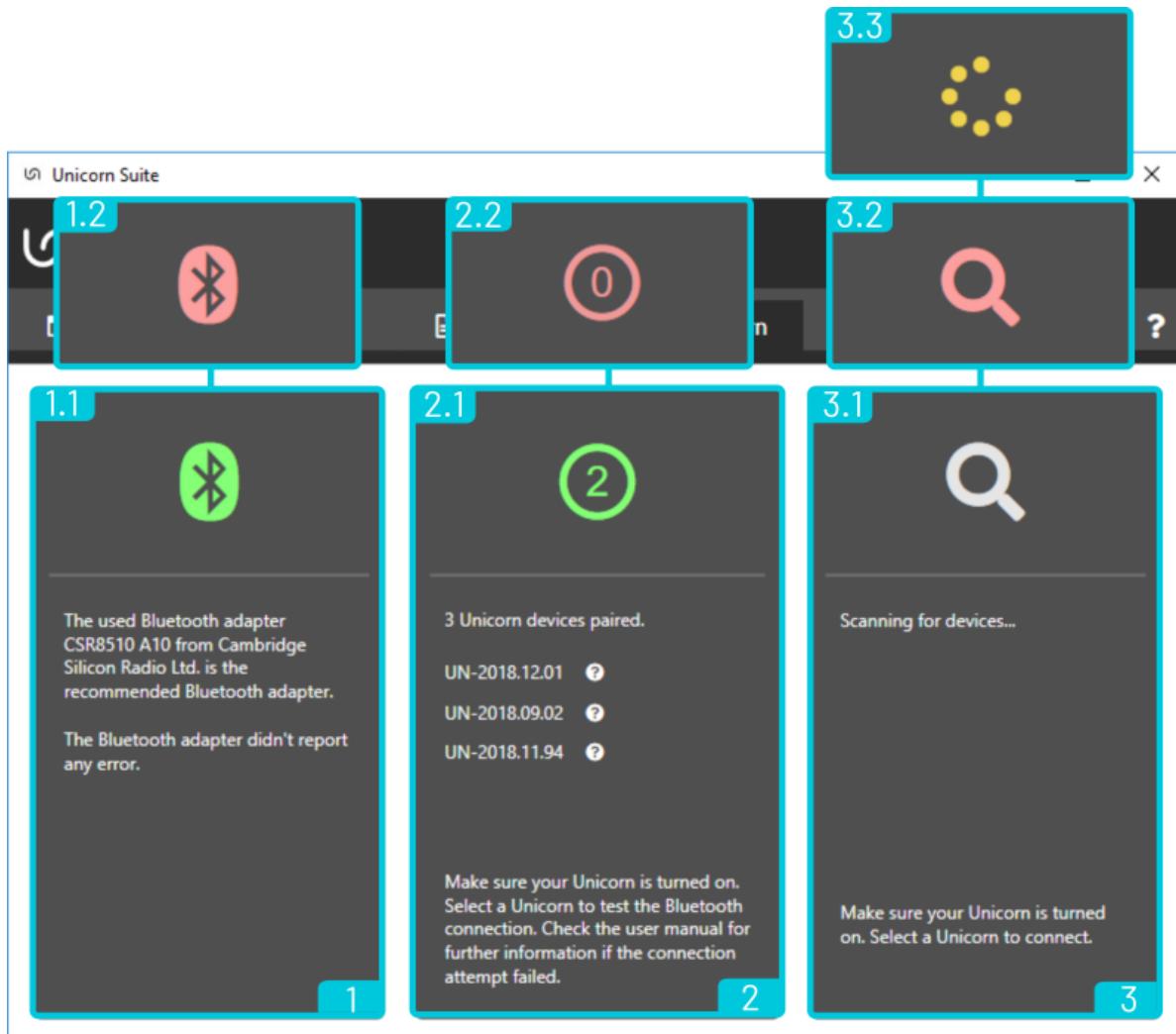
Product	Offline usage expiration date	License expiration date	License state
Unicorn Recorder Hybrid Black	Perpetual	Perpetual	Valid
Unicorn Simulink Hybrid Black	Perpetual	Perpetual	Valid
Unicorn Python Hybrid Black	Perpetual	Perpetual	Valid
Unicorn Speller Hybrid Black	Perpetual	Perpetual	Valid

Add... Deactivate

✓ Licenses queried.



## 12.4. MY UNICORN



- 1 This panel displays the Bluetooth adapter information of the Bluetooth device that is currently in use. The Unicorn Brain Interface features a Bluetooth module for data communication. Therefore, you must install and use the Bluetooth dongle delivered with the Unicorn Brain Interface. Please insert the Bluetooth dongle and wait until the driver of the Unicorn Brain Interface is installed.
  - 1.1 A green Bluetooth symbol indicates that the delivered Bluetooth adapter is in use. Using the delivered Bluetooth adapter is required to ensure a proper interaction between Unicorn hard- and software.
  - 1.2 A red blinking Bluetooth symbol indicates that there is either an invalid or no Bluetooth adapter available for use. Using an invalid Bluetooth adapter might lead to data loss or



unexpected behavior. Please check if the delivered Bluetooth dongle is inserted, installed and in use. If you are using a PC with an internal Bluetooth interface, it is very likely that the computer is still using the internal Bluetooth adapter instead of the recommended Unicorn Bluetooth adapter. You have to disable the internal Bluetooth adapter before the delivered Bluetooth adapter is inserted. Modifications of the Bluetooth adapter can be performed in Windows "Device Manager".

- 2 This panel displays all paired Unicorn brain interfaces.
  - 2.1 The green circle indicates that there are paired Unicorn Brain Interfaces ready for use. The number in the circle represents the number of paired Unicorn Brain Interfaces.
  - 2.2 A red blinking symbol indicates that no Unicorn Brain Interface is paired and ready for use. Unicorn Brain Interfaces can be paired in the Unicorn Suite directly or in the Windows "Bluetooth and other devices settings".
- 3 This panel is displaying unpaired Unicorn Brain Interfaces which can be paired.
  - 3.1 A grey blinking loupe symbol indicates that the Unicorn Suite is searching for available Unicorn Brain Interfaces. Available Unicorn Brain Interfaces will be listed in the panel and can be selected to be paired.
  - 3.2 A red blinking loupe symbol indicates that the Unicorn Brain Interface discovery failed. Please check if the delivered Bluetooth dongle is inserted, installed and in use.
  - 3.3 A yellow blinking symbol appears after a discovered Unicorn brain interface was selected for pairing. Please follow the Windows instructions that will pop up and allow the Unicorn Brain Interface to get paired.



## 13. UNICORN C API

The Unicorn C API is a C/C++ application programming interface (API) enabling the communication with Unicorn brain interfaces from C/C++ applications. The Unicorn C API allows users to acquire data from Unicorn Brain Interfaces easily without having to take care of low-level data acquisition issues. The raw binary data stream is converted into numerical values such that the user receives data ready to analyze.

### 13.1. REQUIREMENTS

Software	Properties
Visual Studio	Microsoft Visual Studio 2015 Desktop development for the C++ programming language

### 13.2. FILES ON YOUR COMPUTER

By default, the Unicorn C API library is installed into the Documents folder.	
C:\Users\<username>\Documents\gtec\Unicorn Suite\Hybrid Black\Unicorn CAPI	Standard installation folder for the Unicorn C API library
Within this directory, subdirectories are generated containing all installed files.	
.\\Lib	Contains the Unicorn C API for Windows 64-bit
.\\Examples	Contains application examples for the Unicorn C API

### 13.3. USING THE UNICORN C API

There are several possibilities to refer to the Unicorn C API files from your individual project.

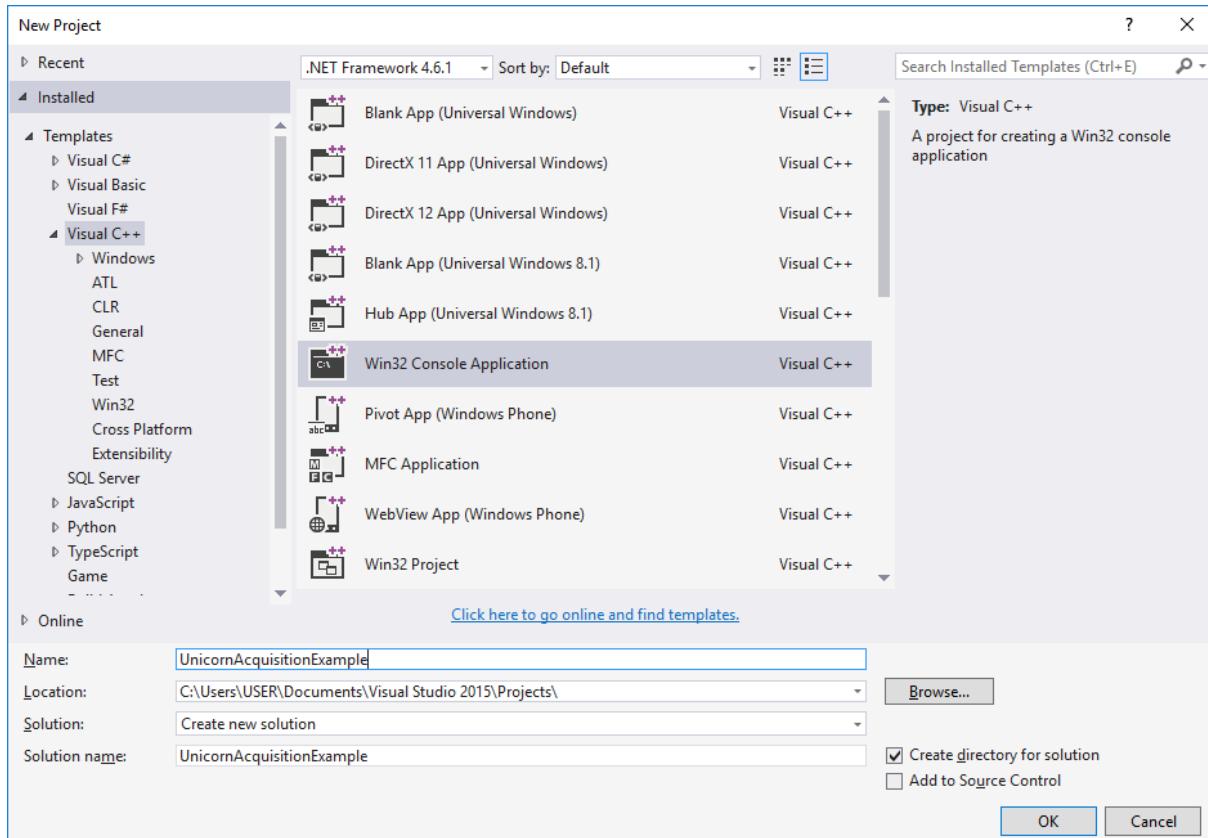
- ⌚ Copy the Unicorn.lib and Unicorn.h to your project folder and add both files to your project.
- ⌚ Add the library folder (by default C:\Users\<username>\Documents\gtec\Unicorn Suite\Hybrid Black\Unicorn CAPI\Examples) to the search path in your IDE and add Unicorn.lib and Unicorn.h to your project.

### 13.4. SETTING UP A PROJECT USING VISUAL STUDIO 2015

1. Open Visual Studio 2015.



2. Create a new C++ Win32 Console application (File → New → Project → Visual C++).



3. Ensure that 64-bit is selected as target platform for the project.
4. Open the project properties dialog (Project → Properties).
5. Open the C/C++ Settings (Configuration Properties → C/C++ → General).
6. Add the path of the Unicorn C API (by default C:\Users\<username>\Documents\gtc\Unicorn Suite\Unicorn CAPI\Hybrid Black\Lib) to the "Additional Include Directories".



UnicornAcquisitionExample Property Pages

Configuration: All Configurations Platform: x64 Configuration Manager...

Configuration Properties	Additional Include Directories	C:\Users\<username>\Documents\gtc\Unicorn Suite\Hybrid Black\Unicorn CAPI\Lib
General	Additional #using Directories	<different options>
Debugging	Debug Information Format	
VC++ Directories	Common Language RunTime Support	
C/C++	Consume Windows Runtime Extension	
General	Suppress Startup Banner	Yes (/nologo)
Optimization	Warning Level	Level3 (/W3)
Preprocessor	Treat Warnings As Errors	No (/WX-)
Code Generation	Warning Version	
Language	SDL checks	
Precompiled Headers	Multi-processor Compilation	
Output Files		
Browse Information		
Advanced		
All Options		
Command Line		
Linker		
Manifest Tool		
XML Document Generator		
Browse Information		
Build Events		
Custom Build Step		
Code Analysis		

Additional Include Directories  
Specifies one or more directories to add to the include path; separate with semi-colons if more than one. (/I[path])

OK Cancel Apply

7. Open the Linker Settings (Configuration Properties → C/C++ → General).
8. Add the path of the Unicorn C API (by default C:\Users\<username>\Documents\gtc\Unicorn Suite\Hybrid Black\Unicorn CAPI \Lib) to the "Additional Library Directories".

UnicornAcquisitionExample Property Pages

Configuration: All Configurations Platform: x64 Configuration Manager...

Configuration Properties	Output File	\$(OutDir)\$(TargetName)\$(TargetExt)
General	Show Progress	Not Set
Debugging	Version	
VC++ Directories	Enable Incremental Linking	<different options>
C/C++	Suppress Startup Banner	Yes (/NOLOGO)
Linker	Ignore Import Library	No
General	Register Output	No
Input	Per-user Redirection	No
Manifest File	Additional Library Directories	C:\Users\<username>\Documents\gtc\Unicorn Suite\Hybrid Black\Unicorn CAPI\
Debugging	Link Library Dependencies	Yes
System	Use Library Dependency Inputs	No
Optimization	Link Status	
Embedded IDL	Prevent DLL Binding	
Windows Metadata	Treat Linker Warning As Errors	
Advanced	Force File Output	
All Options	Create Hot Patchable Image	
Command Line	Specify Section Attributes	
Manifest Tool		
XML Document Generator		
Browse Information		
Build Events		
Custom Build Step		
Code Analysis		

Additional Library Directories  
Allows the user to override the environmental library path. (/LIBPATH:folder)

OK Cancel Apply

9. Add the header file and the library file of the Unicorn C API as follows:



```
#include "stdafx.h"

// Include unicorn header-file.
#include "unicorn.h"

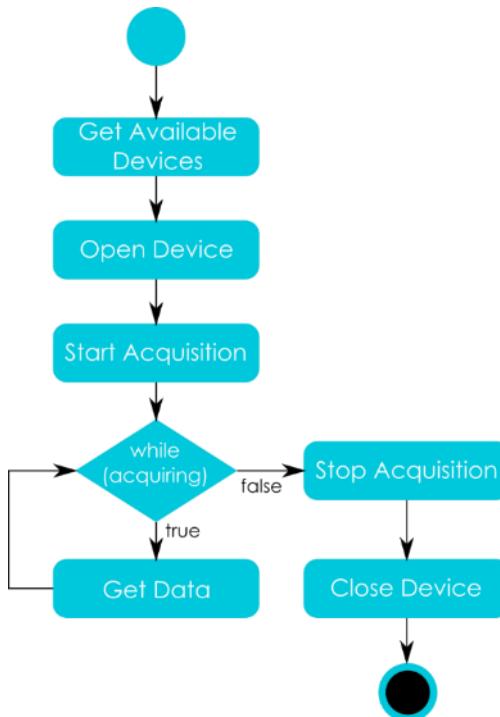
// Include unicorn lib.
#pragma comment(lib, "Unicorn.lib")

int main()
{
    // Insert your code here.
    return 0;
}
```



## 13.5. COMMAND ORDER

To perform a data acquisition using the Unicorn C API, a defined command execution order is required.



1. Before connecting to a Unicorn Brain Interface, it is possible to check the operating environment of Unicorn Brain Interfaces and to discover available Unicorn Brain Interfaces.
2. A connection has to be established to communicate with the Unicorn brain interface. This can be performed by calling [Open Device](#). If a Unicorn Brain Interface handle unequal to [Null](#) is received, the connection attempt was executed successfully. After connecting to a Unicorn Brain Interface, it is possible to interact with the Unicorn Brain Interface and call all functions that require a [UNICORN HANDLE](#). For example, it is possible to read the current configuration of the Unicorn Brain Interface, set a new configuration or start data acquisition.
3. To start data acquisition, [Start Acquisition](#) must be called. After calling [Start Acquisition](#), the Unicorn Brain Interface is set into acquisition mode and is continuously sending data.
4. Therefore, it is required to read the incoming data stream continuously by calling [Get Data](#) within an acquisition loop. Other API calls are not allowed while data acquisition is running.
5. To stop data acquisition, [Stop Acquisition](#) must be called. The Unicorn Brain Interface will terminate the data stream. The Unicorn Brain Interface is still connected. It is possible to interact with the Unicorn Brain Interface and call all functions that require a [UNICORN HANDLE](#).
6. To disconnect from a Unicorn Brain Interface, [Close Device](#) must be called. Afterwards, it is not possible to interact with the Unicorn Brain Interface anymore. The Unicorn Brain Interface has to be opened again for interaction.



## 13.6. UNICORN C API – C REFERENCE

### 13.6.1. CONSTANTS

UNICORN\_ACCELEROMETER\_CHANNELS\_COUNT

The number of available accelerometer channels.

UNICORN\_ACCELEROMETER\_CONFIG\_INDEX

Index of the first accelerometer UNICORN\_AMPLIFIER\_CHANNEL in the UNICORN\_AMPLIFIER\_CONFIGURATION channels array.

UNICORN\_BATTERY\_CONFIG\_INDEX

Index of the battery level UNICORN\_AMPLIFIER\_CHANNEL in the UNICORN\_AMPLIFIER\_CONFIGURATION channels array.

UNICORN\_COUNTER\_CONFIG\_INDEX

Index of the counter UNICORN\_AMPLIFIER\_CHANNEL in the UNICORN\_AMPLIFIER\_CONFIGURATION channels array.

UNICORN\_DEVICE\_VERSION\_LENGTH\_MAX

The maximum length of the Unicorn Brain Interface version.

UNICORN\_EEG\_CHANNELS\_COUNT

The number of available EEG channels.

UNICORN\_EEG\_CONFIG\_INDEX

Index of the first EEG UNICORN\_AMPLIFIER\_CHANNEL in the UNICORN\_AMPLIFIER\_CONFIGURATION channels array.

UNICORN\_FIRMWARE\_VERSION\_LENGTH\_MAX

The maximum length of the firmware version.

UNICORN\_GYROSCOPE\_CHANNELS\_COUNT

The number of available gyroscope channels.

UNICORN\_GYROSCOPE\_CONFIG\_INDEX

Index of the first gyroscope UNICORN\_AMPLIFIER\_CHANNEL in the UNICORN\_AMPLIFIER\_CONFIGURATION channels array.

UNICORN\_NUMBER\_OF\_DIGITAL\_OUTPUTS

The number of digital output channels.



`UNICORN_RECOMMENDED_BLUETOOTH_DEVICE_MANUFACTURER`

The manufacturer of the recommended (delivered) Bluetooth adapter.

`UNICORN_RECOMMENDED_BLUETOOTH_DEVICE_NAME`

The Unicorn Brain Interface name of the recommended (delivered) Bluetooth adapter.

`UNICORN_SAMPLING_RATE`

The sampling rate of the Unicorn Brain Interface.

`UNICORN_SERIAL_LENGTH_MAX`

The maximum length of the serial number.

`UNICORN_STRING_LENGTH_MAX`

The maximum string length.

`UNICORN_SUPPORTED_DEVICE_VERSION`

The Unicorn Brain Interface version which is valid for this API.

`UNICORN_TOTAL_CHANNELS_COUNT`

The total number of available channels.

`UNICORN_VALIDATION_CONFIG_INDEX`

Index of the validation indicator `UNICORN_AMPLIFIER_CHANNEL` in the `UNICORN_AMPLIFIER_CONFIGURATION` channels array.

### **13.6.2. ERROR CODES**

`UNICORN_ERROR_BLUETOOTH_INIT_FAILED`

The initialization of the Bluetooth adapter failed.

`UNICORN_ERROR_BLUETOOTH_SOCKET_FAILED`

The operation could not be performed because the Bluetooth socket failed.

`UNICORN_ERROR_BUFFER_OVERFLOW`

The acquisition buffer is full.

`UNICORN_ERROR_BUFFER_UNDERFLOW`

The acquisition buffer is empty.



#### UNICORN\_ERROR\_CONNECTION\_PROBLEM

The operation could not complete because of connection problems.

#### UNICORN\_ERROR\_GENERAL\_ERROR

An unspecified error occurred.

#### UNICORN\_ERROR\_INVALID\_CONFIGURATION

The configuration is invalid.

#### UNICORN\_ERROR\_INVALID\_HANDLE

The specified Unicorn handle is invalid.

#### UNICORN\_ERROR\_INVALID\_PARAMETER

One of the specified parameters does not contain a valid value.

#### UNICORN\_ERROR\_OPEN\_DEVICE\_FAILED

The Unicorn Brain Interface could not be opened.

#### UNICORN\_ERROR\_OPERATION\_NOT\_ALLOWED

The operation is not allowed during acquisition or non-acquisition.

#### UNICORN\_ERROR\_SUCCESS

The operation completed successfully. No error occurred.

#### UNICORN\_ERROR\_UNSUPPORTED\_DEVICE

The Unicorn Brain Interface is not supported with this API (UNICORN\_SUPPORTED\_DEVICE\_VERSION).

### 13.6.3. TYPE DEFINITIONS

#### BOOL

The Boolean data type, whose values can be TRUE or FALSE.

#### FALSE

The FALSE value for the BOOL type.

#### TRUE

The TRUE value for the BOOL type.



NULL

The null pointer.

UNICORN\_DEVICE\_SERIAL

The type that holds Unicorn Brain Interface serial.

UNICORN\_DEVICE\_VERSION

The type that holds Unicorn Brain Interface version.

UNICORN\_FIRMWARE\_VERSION

The type that holds firmware version.

UNICORN\_HANDLE

The type that holds the Unicorn Brain Interface handle associated with a device.

### 13.6.4. STRUCTURES

UNICORN\_AMPLIFIER\_CHANNEL

The type containing information about a single channel of the Unicorn Brain Interface.

#### PUBLIC ATTRIBUTES

- char name[32]  
The channel's name.
- char unit[32]  
The channel's unit.
- float range[2]  
The channel's input range as float array. First entry is min value; second is max value.
- BOOL enabled  
The channel's enabled flag. [TRUE](#) to enable the channel; [FALSE](#) to disable the channel.

UNICORN\_AMPLIFIER\_CONFIGURATION

The type holding an Unicorn Brain Interface configuration.

#### PUBLIC ATTRIBUTES

- UNICORN\_AMPLIFIER\_CHANNEL Channels[UNICORN\_TOTAL\_CHANNELS\_COUNT]  
The array holding a configuration for each available UNICORN\_AMPLIFIER\_CHANNEL.

UNICORN\_BLUETOOTH\_ADAPTER\_INFO



The type that holds information about the Bluetooth adapter.

#### PUBLIC ATTRIBUTES

- char name[[UNICORN\\_STRING\\_LENGTH\\_MAX](#)]  
The name of the Bluetooth adapter used.
- char manufacturer[[UNICORN\\_STRING\\_LENGTH\\_MAX](#)]  
The manufacturer of the Bluetooth adapter.
- BOOL isRecommendedDevice  
Indicates whether the used Bluetooth adapter is a recommended (delivered) device. [FALSE](#) if the adapter is not a recommended Bluetooth device. [TRUE](#) if the adapter is a recommended device.
- BOOL hasProblem  
Indicates whether the Bluetooth adapter reports a problem. [FALSE](#) if the adapter behaves as supposed. [TRUE](#) if the adapter reports a problem.

## UNICORN\_DEVICE\_INFORMATION

Type that holds additional information about the Unicorn Brain Interface.

#### PUBLIC ATTRIBUTES

- uint16\_t numberOfEegChannels  
The number of EEG channels.
- UNICORN\_DEVICE\_SERIAL serial  
The serial number of the Unicorn Brain Interface.
- UNICORN\_FIRMWARE\_VERSION firmwareVersion  
The firmware version number.
- UNICORN\_DEVICE\_VERSION deviceVersion  
The Unicorn Brain Interface version number.
- uint8\_t pcbVersion[4]  
The PCB version number.
- uint8\_t enclosureVersion[4]  
The enclosure version number.



## 13.6.5. FUNCTIONS

### 13.6.5.1. Close Device

```
int UNICORN_CloseDevice(UNICORN_HANDLE *hDevice)
```

Closes a Unicorn Brain Interface.

Disconnects from a Unicorn Brain Interface by a given Unicorn handle.

#### PARAMETERS

hDevice	A pointer to the handle associated with the session.
---------	--

#### RETURNS

An error code is returned as an integer if the disconnection attempt fails.

### 13.6.5.2. Get API Version

```
float UNICORN_GetApiVersion()
```

Returns the current API version.

#### RETURNS

The current API version.

### 13.6.5.3. Get Available Devices

```
int UNICORN_GetAvailableDevices(UNICORN\_DEVICE\_SERIAL *availableDevices, uint32_t  
*availableDevicesCount, BOOL onlyPaired)
```

Scans for available Unicorn Brain Interfaces.

Discovers available paired or unpaired Unicorn Brain Interfaces. Estimates the number of available paired or unpaired Unicorn Brain Interfaces and returns information about discovered Unicorn Brain Interfaces.

#### PARAMETERS

availableDevices	A pointer to the beginning of an array of <a href="#">UNICORN DEVICE SERIAL</a> which receives available Unicorn Brain Interfaces when the method returns. If NULL is passed, the number of available devices is returned only to determine the amount of memory to allocate.
availableDevicesCount	A pointer to a variable that receives the number of available devices.
onlyPaired	Defines whether only paired devices or only unpaired devices should be returned. If only unpaired devices should be returned, an extensive device scan is performed. An extensive device scan takes a rather long time. In the meantime, the Bluetooth adapter and the application are blocked. Scanning for paired devices only can be executed faster. If <a href="#">TRUE</a> , only



	paired devices are discovered. If <a href="#">FALSE</a> , only unpaired devices can be discovered.
--	--

#### RETURNS

An error code is returned as integer if scanning for available devices fails.

#### 13.6.5.4. Get Bluetooth Adapter Info

```
int UNICORN_GetBluetoothAdapterInfo(UNICORN_BLUETOOTH_ADAPTER_INFO *bluetoothAdapterInfo)
```

Retrieves information about the used Bluetooth Adapter.

Evaluates which Bluetooth adapter is currently in use and whether it is the recommended (delivered) Bluetooth adapter.

#### PARAMETERS

bluetoothAdapterInfo	A pointer to a <a href="#">UNICORN BLUETOOTH ADAPTER INFO</a> structure that receives information about the used Bluetooth adapter.
----------------------	---

#### RETURNS

An error code is returned as integer if the Bluetooth adapter information could not be acquired.

#### 13.6.5.5. Get Channel Index

```
int UNICORN_GetChannelIndex(UNICORN_HANDLE hDevice, const char *name, uint32_t *channelIndex)
```

Determines the index of the requested channel within an acquired scan.

Uses the currently set [UNICORN AMPLIFIER CONFIGURATION](#) to get the index of the requested channel within an acquired scan.

The default names are:

- EEG 1|2|3|4|5|6|7|8
- Accelerometer X|Y|Z
- Gyroscope X|Y|Z
- Counter
- Battery Level
- Validation Indicator

#### PARAMETERS

hDevice	The UNICORN_HANDLE associated with the session.
name	The name of the requested channel.
channelIndex	A pointer to a variable that receives the zero-based channel index.

#### RETURNS

An error code is returned as integer if the index could not be determined.



#### 13.6.5.6. Get Configuration

```
int UNICORN_GetConfiguration(UNICORN_HANDLE hDevice, UNICORN_AMPLIFIER_CONFIGURATION *configuration)
```

Gets the current Unicorn Brain Interface configuration.

Retrieves the current Unicorn Brain Interface configuration from the device as UNICORN\_AMPLIFIER\_CONFIGURATION.

##### PARAMETERS:

hDevice	The UNICORN_HANDLE associated with the session.
configuration	A pointer to a UNICORN_AMPLIFIER_CONFIGURATION that receives the configuration of the Unicorn Brain Interface.

##### RETURNS

An error code is returned as an integer if the configuration could not be read.

#### 13.6.5.7. Get Data

```
int UNICORN_GetData(UNICORN HANDLE hDevice, uint32_t numberOfScans, float *destinationBuffer, uint32_t destinationBufferLength)
```

Reads a specific number of scans into the specified destination buffer of known length. Checks whether the destination buffer is big enough to hold the requested number of scans.

##### PARAMETERS

hDevice	The UNICORN_HANDLE associated with the session.
numberOfScans	The number of scans to read. The number of scans must be greater than zero. A scan consists of one 32-bit floating point number for each currently acquired channel.
destinationBuffer	A pointer to the destination buffer that receives the acquired data. The destination buffer must provide enough memory to hold the requested number of scans multiplied by the number of acquired channels.  Call UNICORN_GetNumberOfAcquiredChannels to determine the number of acquired channels. Call UNICORN_GetChannelIndex to determine the index of a channel within a scan. Example: The sample of the battery level channel in the n-th scan is:  <code>n*UNICORN_GetNumberOfAcquiredChannels() + UNICORN_GetChannelIndex("Battery Level")</code>
destinationBufferLength	Number of floats fitting into the destination buffer.

##### RETURNS

An error code is returned as integer if data could not be read.



#### 13.6.5.8. Get Device Information

```
int UNICORN_GetDeviceInformation(UNICORN_HANDLE hDevice, UNICORN_DEVICE_INFORMATION *deviceInformation)
```

Reads the device information by a given UNICORN\_HANDLE.

##### PARAMETERS

hDevice	The UNICORN_HANDLE associated with the session.
deviceInformation	A pointer to a UNICORN_DEVICE_INFORMATION that receives information about the device.

##### RETURNS

An error code is returned as an integer if the device information could not be read.

#### 13.6.5.9. Get Digital Outputs

```
int UNICORN_GetDigitalOutputs(UNICORN_HANDLE hDevice, uint8_t *digitalOutputs)
```

Reads the current state of the digital outputs.

##### PARAMETERS

hDevice	The UNICORN_HANDLE associated with the session.
digitalOutputs	A pointer to a variable that receives the states of the digital output channels. Each bit represents one digital output channel. If a bit is set, the corresponding digital output channel's value is set to high. If a bit is cleared, the corresponding digital output channel's value is set to low.  Examples (the binary representation of each decimal value is shown in parentheses): 0 (0000 0000 <sub>b</sub> ) → all digital outputs set to low. 170 (1010 1010 <sub>b</sub> ) → digital outputs 2,4,6,8 are set to high. 255 (1111 1111 <sub>b</sub> ) → all digital outputs set to high.

##### RETURNS

An error code is returned as an integer if the state of the digital output channels could not be read.

#### 13.6.5.10. Get Last Error Text

```
const char* UNICORN_GetLastErrorText()
```

Returns the description of the last error occurred.

##### RETURNS

The description of the last error occurred.



#### 13.6.5.11. Get Number of Acquired Channels

```
int UNICORN_GetNumberOfAcquiredChannels(UNICORN HANDLE hDevice, uint32_t *numberOfAcquiredChannels)
```

Determines the number of acquired channels.

Uses the currently set UNICORN\_AMPLIFIER\_CONFIGURATION to get the number of acquired channels.

##### PARAMETERS

hDevice	The UNICORN_HANDLE associated with the session.
numberOfAcquiredChannels	A pointer to a variable that receives the number of acquired channels.

##### RETURNS

An error code is returned as an integer if the number of acquired channels could not be determined.

#### 13.6.5.12. Open Device

```
int UNICORN_OpenDevice(const char *serial, UNICORN HANDLE *hDevice)
```

Connects to a certain Unicorn device and assigns a Unicorn handle if the connection attempt succeeded.

##### PARAMETERS

serial	The serial number of the device to connect to.
hDevice	A pointer to a <a href="#">UNICORN HANDLE</a> that receives the handle associated with the current session if the device could be opened successfully.

##### RETURNS

An error code is returned as an integer if the device could not be opened.

#### 13.6.5.13. Set Configuration

```
int UNICORN_SetConfiguration(UNICORN_HANDLE hDevice, UNICORN_AMPLIFIER_CONFIGURATION *configuration)
```

Sets an UNICORN\_AMPLIFIER\_CONFIGURATION.

##### PARAMETERS

hDevice	The UNICORN_HANDLE associated with the session.
configuration	A pointer to the UNICORN_AMPLIFIER_CONFIGURATION to set.

##### RETURNS

An error code is returned as an integer if configuration is invalid or could not be set.



#### 13.6.5.14. Set Digital Outputs

```
int UNICORN_SetDigitalOutputs(UNICORN HANDLE hDevice, uint8_t digitalOutputs)
```

Sets the digital outputs to high or low.

##### PARAMETERS

hDevice	The UNICORN_HANDLE associated with the session.
digitalOutputs	<p>The state of the digital output channels to set in bits. Each bit represents one digital output channel. Set a bit to set the corresponding digital output channel's value to high. Clear a bit to set the corresponding digital output channel's value to low.</p> <p>Examples (the binary representation of each decimal value is shown in parentheses):</p> <ul style="list-style-type: none"><li>0 (0000 0000<sub>b</sub>) → all digital outputs set to low.</li><li>170 (1010 1010<sub>b</sub>) → digital outputs 2,4,6,8 are set to high.</li><li>255 (1111 1111<sub>b</sub>) → all digital outputs set to high.</li></ul>

##### RETURNS

An error code is returned as an integer if the state of the digital output channels could not be set.

#### 13.6.5.15. Start Acquisition

```
int UNICORN_StartAcquisition(UNICORN HANDLE hDevice, BOOL testSignalEnabled)
```

Starts data acquisition in test signal or measurement mode.

##### PARAMETERS

hDevice	The UNICORN_HANDLE associated with the session.
testSignalEnabled	Enables or disables the test signal mode. <a href="#">TRUE</a> to start the data acquisition in test signal mode; <a href="#">FALSE</a> to start the data acquisition in measurement mode.

##### RETURNS

An error code is returned as an integer if data acquisition could not be started.

#### 13.6.5.16. Stop Acquisition

```
int UNICORN_StopAcquisition(UNICORN HANDLE hDevice)
```

Stops a currently running data acquisition session.

##### PARAMETERS

hDevice	The UNICORN_HANDLE associated with the session.
---------	---



## RETURNS

An error code is returned as an integer if the acquisition could not be terminated.



## 14. UNICORN .NET API

The Unicorn .NET API is a .NET application programming interface (API) that enables C# and other .NET applications to communicate with Unicorn brain interfaces. The Unicorn .NET API allows users to acquire data from Unicorn devices easily without having to take care of low-level data acquisition issues. The raw binary data stream is converted into numerical values such that the user receives data ready to analyze.

### 14.1. REQUIREMENTS

Software	Properties
.NET Framework	.NET Framework 4.7.1
Visual Studio	Microsoft Visual Studio 2015

### 14.2. FILES ON YOUR COMPUTER

By default, the Unicorn.NET API library is installed to the Documents folder.

C:\Users\<username>\Documents\gtc\Unicorn Suite\Hybrid Black\Unicorn DotNet	Standard installation folder for the Unicorn .NET API library
---	---

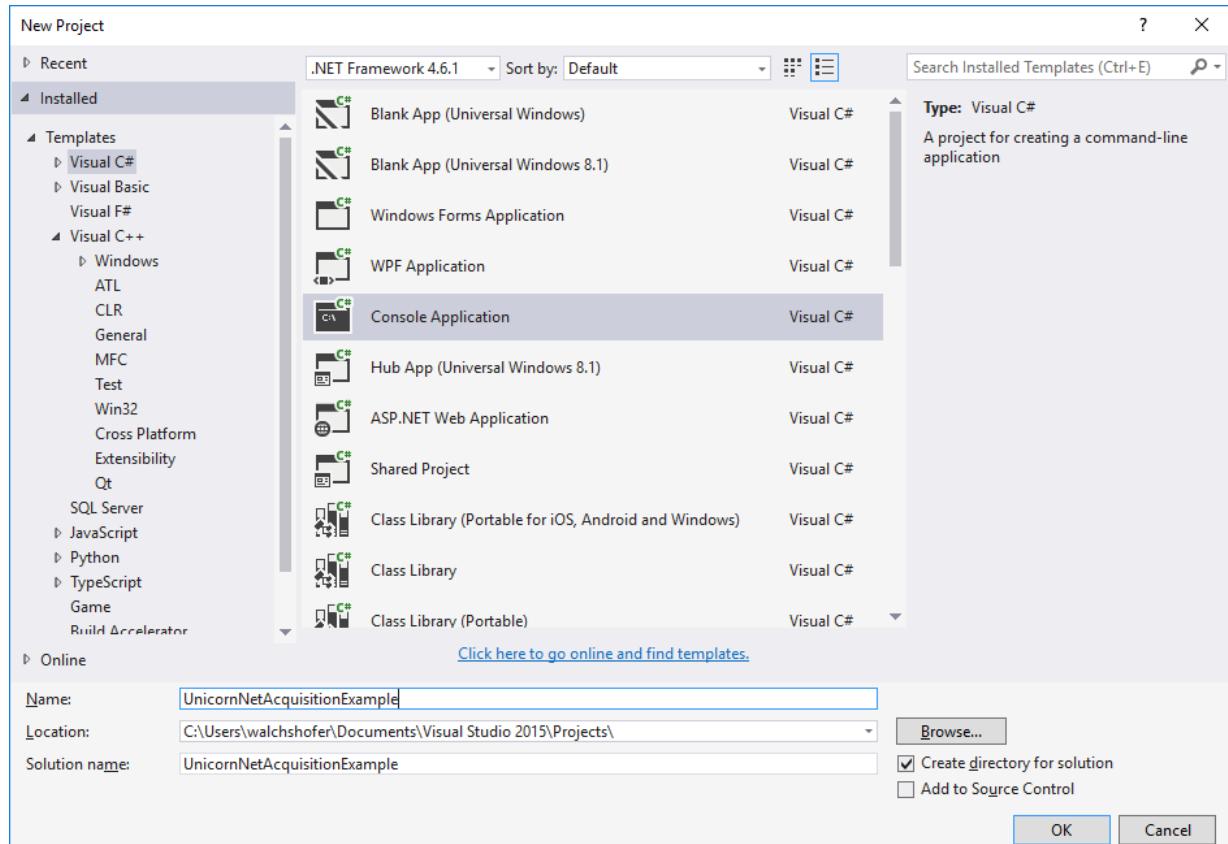
Within this directory, subdirectories are generated containing all installed files.

.\Lib	Contains the Unicorn .NET API for Windows 64-bit
.\Examples	Contains application examples for the Unicorn .NET API

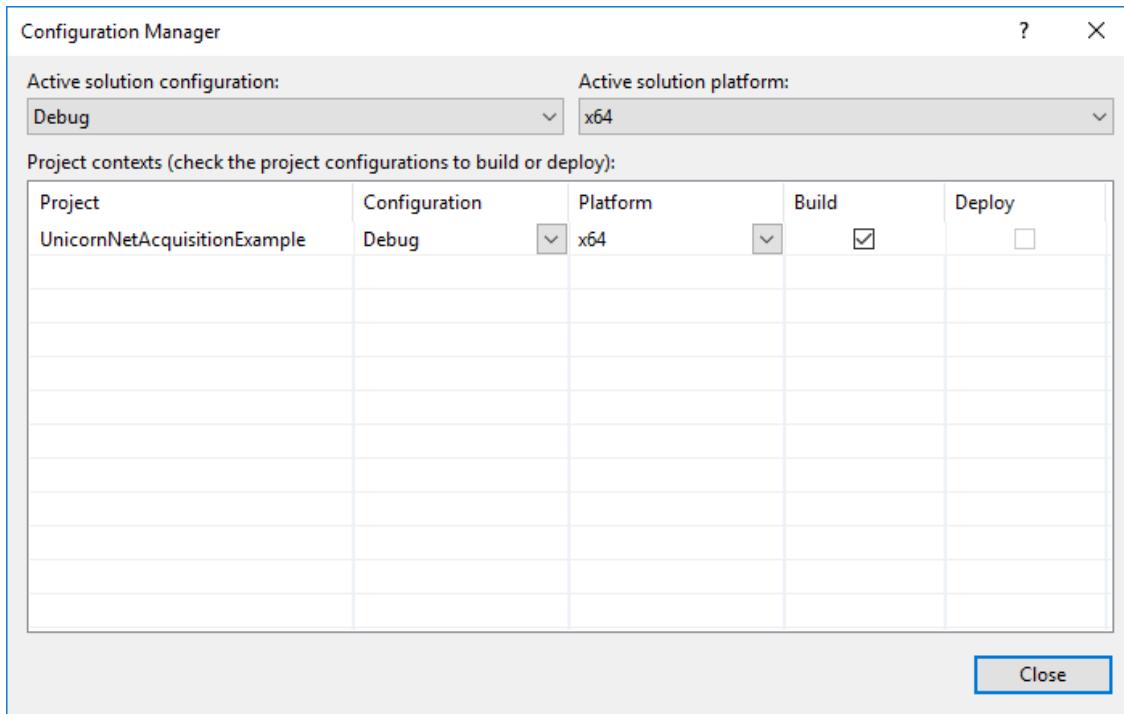


### 14.3. SETTING UP A PROJECT USING VISUAL STUDIO 2015

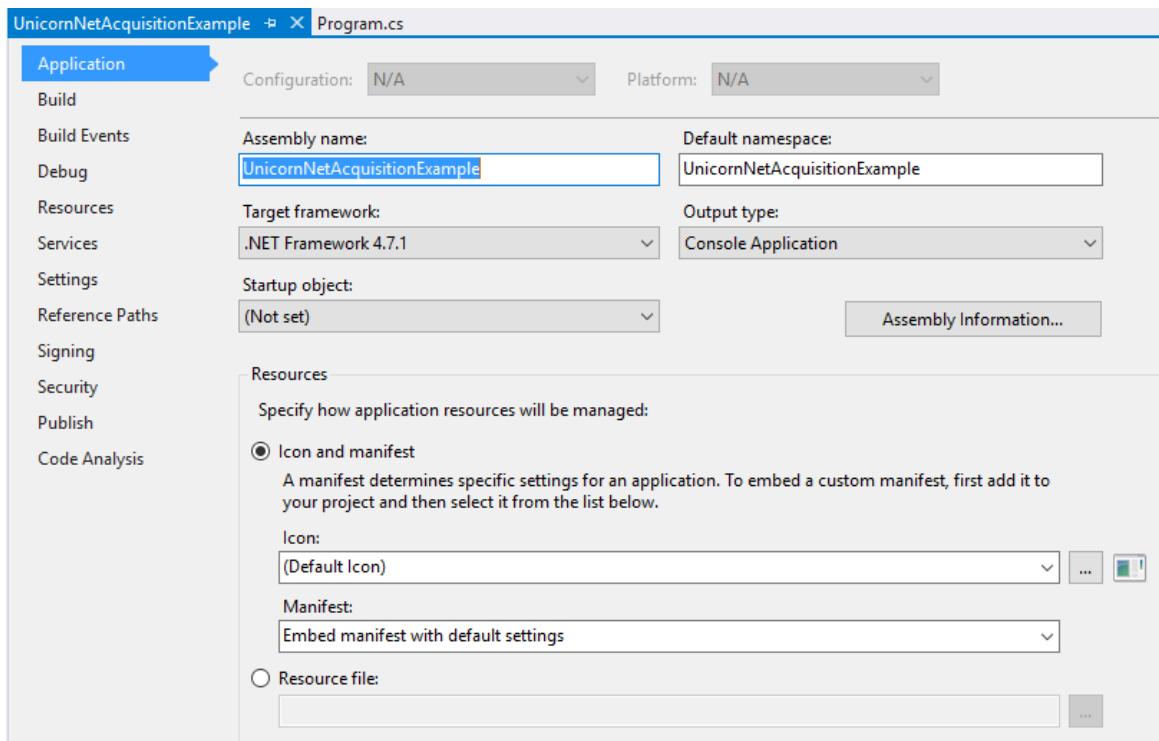
1. Open Visual Studio 2015.
2. Create a new C# Console application (File → New → Project → Visual C#).



3. Open the Configuration Manager and set the "Active solution platform" and project "Platform" to "x64".



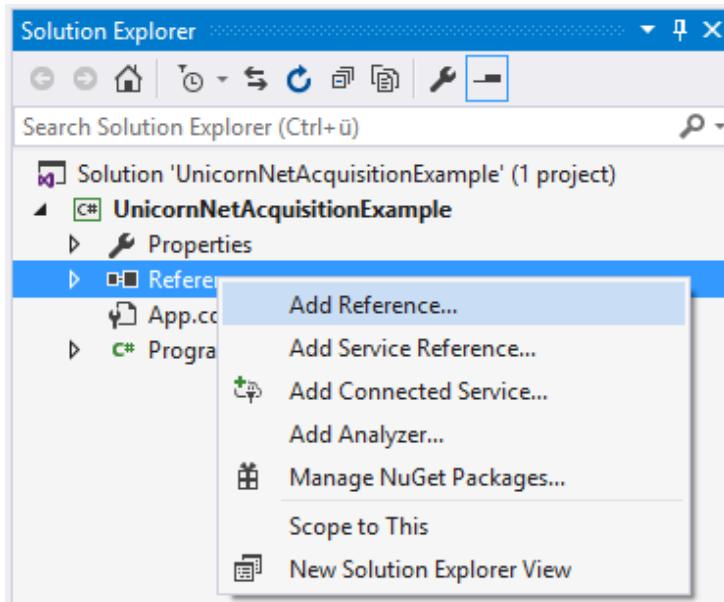
4. Open the project properties dialog (Project → Properties).



5. Set the target framework to ".NET Framework 4.7.1".
6. Select "Add Reference" by right clicking "References" in the Solution Explorer



7. Browse to the library path of the Unicorn .NET API (by default C:\Users\<username>\Documents\gtc\Unicorn Suite\Hybrid Black\Unicorn DotNet\Lib) and add "UnicornDotNet.dll" as reference.



8. Add "using Gtec.Unicorn" to the top of each file using the Unicorn .NET API.

```
using Gtec.Unicorn;

namespace UnicornNetAcquisitionExample
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("API Version: " + Unicorn.GetApiVersion());
            Console.ReadLine();
        }
    }
}
```

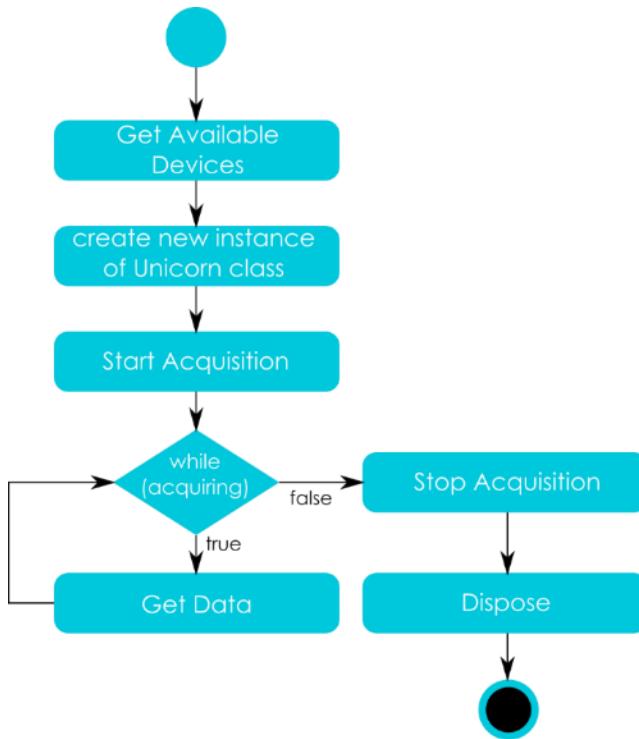
9. You must copy the native Unicorn library (by default C:\Users\<username>\Documents\gtc\Unicorn Suite\Hybrid Black\Unicorn DotNet\Lib\Unicorn.dll) to the execution folder of the C# application. This can be done manually or within the "Build Events" by adding the following lines to the Pre- or Post-build event command Lines.

```
echo Copying Unicorn libraries...
copy "C:\Users\<username>\Documents\gtc\Unicorn Suite\Hybrid Black\Unicorn
DotNet\Lib\Unicorn.dll" "$(TargetDir)"
```



## 14.4. COMMAND ORDER

To perform a data acquisition using the Unicorn .NET API, a defined command execution order is required.



1. Before connecting to a Unicorn device, it is possible to check the operating environment of Unicorn devices and to discover available devices.
2. A connection must be established to communicate with the Unicorn brain interface. To do so, create an instance of the [Unicorn](#) class. If the instance could be created, the connection attempt was executed successfully. After connecting to a device, it is possible to interact with the device and to call all available public member functions of the [Unicorn](#) class. For example, it is possible to read the current configuration of the device, set a new configuration or start data acquisition.
3. To start data acquisition, [Start Acquisition](#) must be called. After calling [Start Acquisition](#), the Unicorn device is set into acquisition mode and is continuously sending data.
4. Therefore, it is necessary to read the incoming data stream continuously by calling [Get Data](#) within an acquisition loop. Other API calls (except [Stop Acquisition](#) and [Set Digital Outputs](#)) are not allowed while data acquisition is running.
5. To stop data acquisition, [Stop Acquisition](#) must be called. The Unicorn device will terminate the data stream. The device is still connected. It is possible to interact with the device as long as the corresponding class instance is kept alive.
6. The device gets disconnected as soon as the garbage collector finalizes the corresponding class instance, or [Dispose](#) is called. Afterwards, it is not possible to interact with the device anymore. The device must be opened again for interaction.



## 14.5. UNICORN .NET API – .NET REFERENCE

### 14.5.1. CLASS DEVICEEXCEPTION

14.5.1.1. Properties

Unicorn.ErrorCodes ErrorCode

Gets the error code of the device exception.

14.5.1.2. Constructors

DeviceException()

Initializes a new instance of the DeviceException class.

DeviceException(Unicorn.ErrorCodes errorCode)

Initializes a new instance of the [DeviceException](#) class with a specified error code.

DeviceException(string message)

Initializes a new instance of the [DeviceException](#) class with a specified error message.

DeviceException(string message, Exception innerException)

Initializes a new instance of the [DeviceException](#) class with a specified error message and a reference to the inner exception that is the cause of this exception.

DeviceException(Unicorn.ErrorCodes errorCode, string message)

Initializes a new instance of the [DeviceException](#) class with a specified error code and error message.

DeviceException(Unicorn.ErrorCodes errorCode, string message, Exception innerException)

Initializes a new instance of the [DeviceException](#) class with a specified error code and error message and a reference to the inner exception that is the cause of this exception.

### 14.5.2. CLASS UNICORN

14.5.2.1. Constants

DllName

The name of the native Unicorn API DLL file that the .NET API uses underneath.

SupportedDeviceVersion

The Unicorn device version that is valid for this API.



RecommendedDeviceName

The device name of the recommended (delivered) Bluetooth adapter.

RecommendedDeviceManufacturer

The manufacturer of the recommended (delivered) Bluetooth adapter.

SerialLengthMax

The maximum length of the serial number, including the terminating null character.

DeviceVersionLengthMax

The maximum length of the device version, including the terminating null character.

FirmwareVersionLengthMax

The maximum length of the firmware version, including the terminating null character.

StringLengthMax

The maximum string length.

SamplingRate

The sampling rate of the Unicorn Brain Interface.

EEGChannelsCount

The number of available EEG channels.

AccelerometerChannelsCount

The number of available accelerometer channel.

GyroscopeChannelsCount

The number of available gyroscope channel.

TotalChannelsCount

The total number of available channels.

EEGConfigIndex

The index of the first EEG AmplifierChannel in AmplifierConfiguration.Channels.

AccelerometerConfigIndex

The index of the first accelerometer AmplifierChannel in AmplifierConfiguration.Channels.



#### GyroscopeConfigIndex

The index of the first gyroscope AmplifierChannel in AmplifierConfiguration.Channels.

#### BatteryConfigIndex

The index of the Battery AmplifierChannel in AmplifierConfiguration.Channels.

#### CounterConfigIndex

The index of the Counter AmplifierChannel in AmplifierConfiguration.Channels.

#### ValidationConfigIndex

The index of the Validation Indicator AmplifierChannel in AmplifierConfiguration.Channels.

#### NumberOfDigitalOutputs

The number of digital output channels.

### 14.5.2.2. Enumerations

#### ErrorCodes

The error codes that the Unicorn API can return.

Success	The operation completed successfully. No error occurred.
InvalidParameter	One of the specified parameters does not contain a valid value.
BluetoothInitFailed	The initialization of the Bluetooth adapter failed.
BluetoothSocketFailed	The operation could not be performed because the Bluetooth socket failed.
OpenDeviceFailed	The device could not be opened.
InvalidConfiguration	The configuration is invalid.
BufferOverflow	The acquisition buffer is full.
BufferUnderflow	The acquisition buffer is empty.
OperationNotAllowed	The operation is not allowed during acquisition or non-acquisition.
ConnectionProblem	The operation could not complete because of connection problems.
UnsupportedDevice	The device is not supported with this API's SupportedDeviceVersion.
InvalidHandle	The specified Unicorn handle is invalid.
UnknownError	An unspecified error occurred.



#### 14.5.2.3. Structures

##### AmplifierChannel

The structure containing information about a single channel of the Unicorn Brain Interface.

###### PUBLIC ATTRIBUTES

- string Name  
The channel's name.
- string Unit  
The channel's unit.
- float[] Range  
The channel's input range as float array. First entry is min value; second is max value.
- bool Enabled  
The channel's enabled flag. true to enable channel; false to disable channel.

##### AmplifierConfiguration

The structure containing the Unicorn Brain Interface configuration.

###### PUBLIC ATTRIBUTES

- [AmplifierChannel](#)[] Channels  
An array of [AmplifierChannel](#) representing all channels.

##### BluetoothAdapterInfo

The structure that holds information about the Bluetooth adapter.

###### PUBLIC ATTRIBUTES

- string Name  
The name of the Bluetooth adapter used.
- string Manufacturer  
The manufacturer of the Bluetooth adapter.
- bool IsRecommendedDevice  
Indicates whether the used Bluetooth adapter is a recommended (delivered) device. True if the adapter is a recommended device; false if the adapter is not a recommended device.
- bool HasProblem  
Indicates whether the Bluetooth adapter reports a problem or not. False if the adapter behaves as expected; true if the adapter reports a problem.



## DeviceInformation

The structure that holds additional information about the device.

### PUBLIC ATTRIBUTES

- ushort NumberOfEegChannels  
The number of EEG channels.
- string Serial  
The serial number of the device.
- string FwVersion  
The firmware version number.
- string DeviceVersion  
The device version number.
- string PcbVersion  
The PCB version number.
- string EnclosureVersion  
The enclosure version number.

#### 14.5.2.4. Constructors

##### Unicorn(string serial)

Initializes a new instance of the [Unicorn](#) class. Connects to the [Unicorn](#) device with the specified serial number. Unicorn class instances can be finalized explicitly by calling [Dispose\(\)](#) to disconnect from devices and deallocate memory. Otherwise, the device gets disconnected as soon as the garbage collector finalizes the corresponding class instance.

##### PARAMETERS

serial	The serial number of the device to connect to.
--------	--

##### EXCEPTIONS

<a href="#">DeviceException</a>	The device could not be opened.
---------------------------------	---------------------------------

#### 14.5.2.5. Static Functions

##### 14.5.2.5.1. Get API Version

float GetApiVersion()

Returns the current API version.



#### RETURNS

The current API version.

#### 14.5.2.5.2. Get Available Devices

`IList<string> GetAvailableDevices(bool discoverPairedDevicesOnly)`

Discovers available paired or unpaired devices.

#### PARAMETERS

<code>discoverPairedDevicesOnly</code>	Defines whether only paired devices or only unpaired devices should be returned. If only unpaired devices should be returned, an extensive device scan is performed. An extensive device scan takes a rather long time. In the meantime, the Bluetooth adapter and the application are blocked. Scanning for paired devices only can be executed faster. If true, only paired devices are discovered. If false, only unpaired devices can be discovered.
--	--

#### RETURNS

A list holding available devices.

#### EXCEPTIONS

<u><a href="#">DeviceException</a></u>	Scanning for devices failed.
--	------------------------------

#### 14.5.2.5.3. Get Bluetooth Adapter Info

`BluetoothAdapterInfo GetBluetoothAdapterInfo()`

Evaluates which Bluetooth adapter is currently in use and whether it is the recommended (delivered) Bluetooth adapter.

#### RETURNS

Information about the used Bluetooth adapter.

Exceptions:

<u><a href="#">DeviceException</a></u>	Bluetooth adapter information could not be acquired.
--	--

#### 14.5.2.5.4. Is Device Library Loadable

`bool IsDeviceLibraryLoadable()`

Determines whether all libraries required for device communication are installed.



#### RETURNS:

True if all required native libraries are installed and can be loaded; otherwise false.

#### 14.5.2.6. Dispose

```
void Dispose()
```

Disconnects from a device and releases allocated memory.

#### 14.5.2.7. Get Channel Index

```
uint GetChannelIndex(string name)
```

Uses the currently set [AmplifierConfiguration](#) to get the index of the requested channel within an acquired scan.

#### PARAMETERS

name	The name of the requested channel.  The default names are: EEG 1 2 3 4 5 6 7 8 Accelerometer X Y Z Gyroscope X Y Z Counter Battery Level Validation Indicator
------	---

#### RETURNS

The zero-based index of the requested channel in an acquired scan.

#### EXCEPTIONS

<a href="#">DeviceException</a>	The channel index could not be determined.
---------------------------------	--

#### 14.5.2.7.1. Get Configuration

```
AmplifierConfiguration GetConfiguration()
```

Retrieves the current Unicorn Brain Interface configuration from the device.

#### RETURNS

The configuration of the Unicorn Brain Interface.

#### EXCEPTIONS

<a href="#">DeviceException</a>	The current Unicorn Brain Interface configuration could not be read.
---------------------------------	--



#### 14.5.2.7.2. Get Data

```
void GetData(uint numberOfScans, IntPtr destinationBuffer, uint destinationBufferLength)
```

Reads a specific number of scans into the specified destination buffer of known length. Checks whether the destination buffer is big enough to hold the requested number of scans.

#### PARAMETERS

numberOfScans	The number of scans to read. The number of scans must be greater than zero. A scan consists of one 32-bit floating point number for each currently acquired channel.
destinationBuffer	A pointer to the native block of memory that receives the acquired data. The destination buffer must provide enough memory to hold the requested number of scans multiplied by the number of acquired channels.  Call GetNumberOfAcquiredChannels() to determine the number of acquired channels. Call GetChannelIndex(string) to determine the index of a channel within a scan. Example: The sample of the battery level channel in the n-th scan is:  $n * \text{GetNumberOfAcquiredChannels}() + \text{GetChannelIndex}(\text{"Battery Level"})$
destinationBufferLength	The number of floats fitting into the destination buffer.

#### EXCEPTIONS

<u>DeviceException</u>	Data could not be read.
------------------------	-------------------------

#### REMARKS

The native buffer memory can be allocated with Marshal.AllocHGlobal(int), or call GCHandle.Alloc(object, GCHandleType) with GCHandleType.Pinned on an allocated managed float array of size destinationBufferLength and use GCHandle.AddrOfPinnedObject() to retrieve the pointer to the pinned block of memory that can be passed to destinationBuffer.

It is recommended to allocate buffer memory only once before acquisition starts and reuse it within the acquisition loop. Don't forget to release the allocated memory with Marshal.FreeHGlobal(IntPtr) if memory was allocated with Marshal.AllocHGlobal(int) or GCHandle.Free() if memory was allocated with GCHandle.Alloc(object, GCHandleType) when it is not needed anymore (e.g. after acquisition has been stopped).

#### 14.5.2.7.3. Get Device Information

```
DeviceInformation GetDeviceInformation()
```

Reads the device information.



#### RETURNS

Information about the device that this instance belongs to.

#### EXCEPTIONS:

<a href="#">DeviceException</a>	Device information could not be read.
---------------------------------	---------------------------------------

#### 14.5.2.7.4. Get Digital Outputs

byte GetDigitalOutputs()

Reads the current state of the digital outputs.

#### RETURNS

The states of the digital output channels in bits. Each bit represents one digital output channel. If a bit is set, the corresponding digital output channel's value is set to high. If a bit is cleared, the corresponding digital output channel's value is set to low.

Examples (the binary representation of each decimal value is shown in parentheses):

- 💡 0 (0000 0000<sub>b</sub>) → all digital outputs set to low.
- 💡 170 (1010 1010<sub>b</sub>) → digital outputs 2,4,6,8 are set to high.
- 💡 255 (1111 1111<sub>b</sub>) → all digital outputs set to high.

#### EXCEPTIONS

<a href="#">DeviceException</a>	The state of the digital output channels could not be read.
---------------------------------	---

#### 14.5.2.7.5. Get Number of Acquired Channels

uint GetNumberOfAcquiredChannels()

Gets the number of acquired channels according to the currently set Unicorn Brain Interface configuration.

#### RETURNS

The number of acquired channels.

#### EXCEPTIONS

<a href="#">DeviceException</a>	The number of acquired channels could not be determined.
---------------------------------	--

#### 14.5.2.7.6. Set Configuration

void SetConfiguration(ref [AmplifierConfiguration](#) configuration)

Sets the configuration of the device.



#### PARAMETERS

configuration	The device configuration to set.
---------------	----------------------------------

#### EXCEPTIONS

<a href="#">DeviceException</a>	The device configuration is invalid or could not be set.
---------------------------------	--

#### 14.5.2.7.7. Set Digital Outputs

void SetDigitalOutputs(byte digitalOutputs)

Sets the digital outputs to high or low.

#### PARAMETERS

digitalOutputs	<p>The state of the digital output channels to set in bits. Each bit represents one digital output channel. Set a bit to set the corresponding digital output channel's value to high. Clear a bit to set the corresponding digital output channel's value to low.</p> <ul style="list-style-type: none"><li>💡 Examples (the binary representation of each decimal value is shown in parentheses):<ul style="list-style-type: none"><li>💡 0 (0000 0000<sub>b</sub>) → all digital outputs set to low.</li><li>💡 170 (1010 1010<sub>b</sub>) → digital outputs 2,4,6,8 are set to high.</li><li>💡 255 (1111 1111<sub>b</sub>) → all digital outputs set to high.</li></ul></li></ul>
----------------	---

#### EXCEPTIONS

<a href="#">DeviceException</a>	The state of the digital output channels could not be set.
---------------------------------	--

#### 14.5.2.7.8. Start Acquisition

void StartAcquisition(bool testsignalEnabled)

Starts data acquisition in test signal or measurement mode.

#### PARAMETERS

testsignalEnabled	Enables or disables the test signal mode. If true, testsignal mode is enabled. If false measurement mode is enabled.
-------------------	--

#### EXCEPTIONS

<a href="#">DeviceException</a>	Data acquisition could not be started.
---------------------------------	--



#### 14.5.2.7.9. Stop Acquisition

```
void StopAcquisition()
```

Stops a currently running data acquisition session.

##### EXCEPTIONS:

<a href="#">DeviceException</a>	Data acquisition could not be stopped.
---------------------------------	--



## 15. UNICORN PYTHON API

The Unicorn Python API is a Python application programming interface (API) enabling the communication with Unicorn brain interfaces from Python. The Unicorn Python API allows users to acquire data from Unicorn devices easily without having to take care of low-level data acquisition issues. The raw binary data stream is converted into numerical values such that the user receives data ready to analyze.

### 15.1. REQUIREMENTS

Software	Properties
Python	> Python 3.3 – x64 NumPy
Visual Studio	Microsoft Visual Studio 2015 Python Development Tools

### 15.2. FILES ON YOUR COMPUTER

By default, the Unicorn Python API library is installed to the Documents folder.

C:\Users\<username>\Documents\gtec\Unicorn Suite\Hybrid Black\Unicorn Python	Standard installation folder for the Unicorn Python API library
--	---

Within this directory, subdirectories are generated containing all installed files.

.\Lib	Contains the Unicorn Python API for Windows 64-bit
.\Examples	Contains application examples for the Unicorn Python API library in Python



## 15.3. USING THE UNICORN PYTHON API

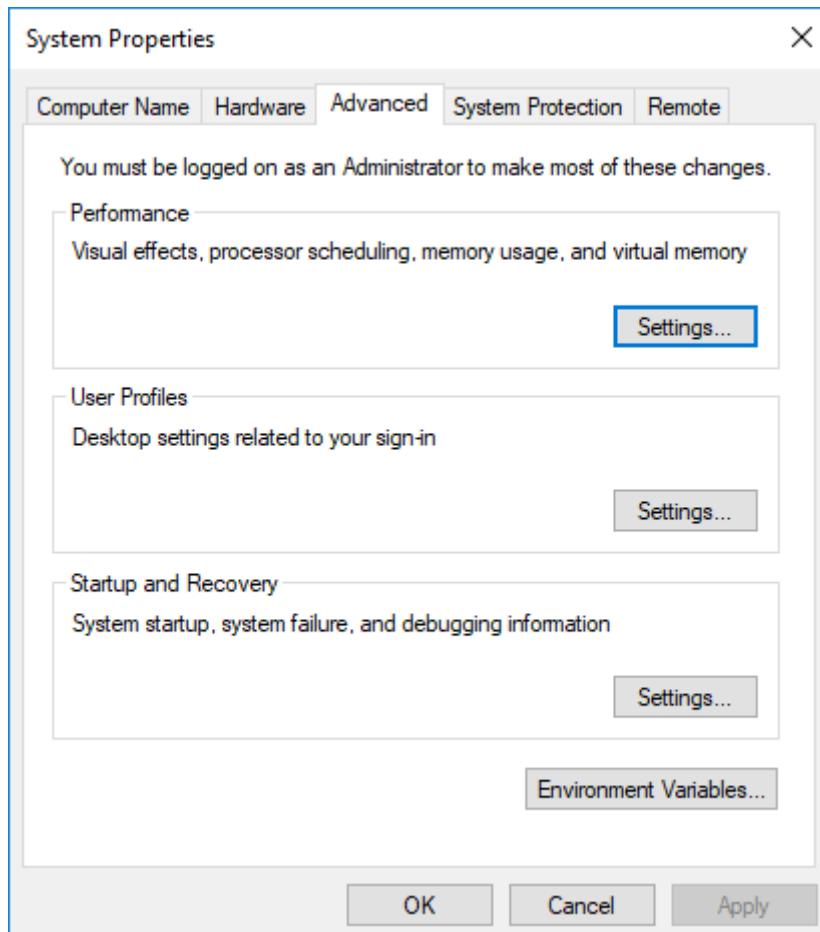
### 15.3.1. ACTIVATE LICENSE

The Unicorn Python library requires a license key to load. Unlock the Unicorn Python API by following the instructions from section 12.3 Licensing.

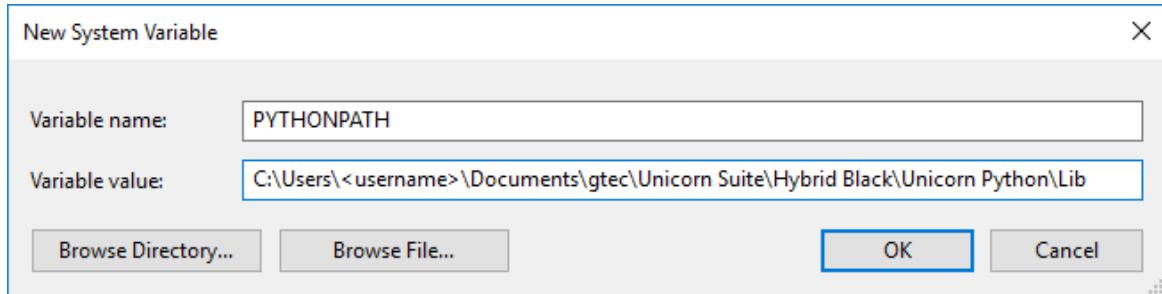
### 15.3.2. SET LIBRARY PATH

You must add the Unicorn Python library paths to the system environment variable PYTHONPATH to be able to use the library in Python applications.

1. Open the "System Properties" and go to the "Advanced" tab.



2. Click "Environment Variables" and add a new "System variable" called PYTHONPATH pointing to the Unicorn Python library folder.



3. It should be possible to load the library by typing "import UnicornPy" in Python after adding the library folder to the python path.

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import UnicornPy
>>> help(UnicornPy)
Help on module UnicornPy:

NAME
    UnicornPy - A Python interface for Unicorn brain interfaces

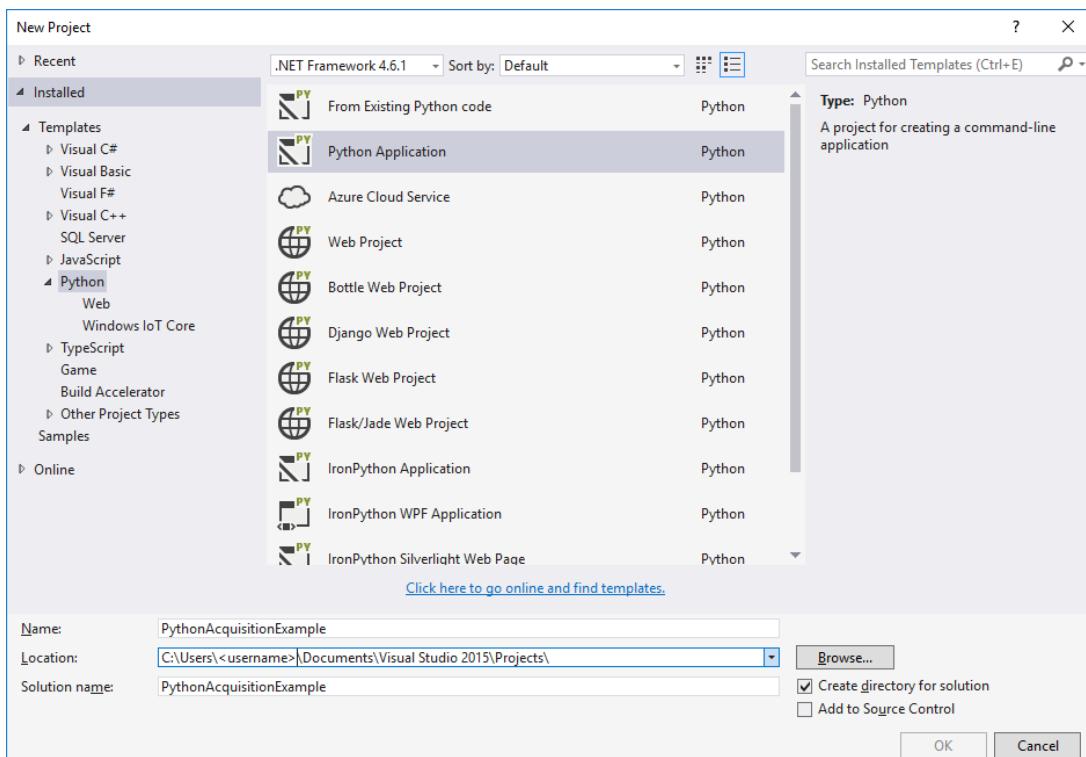
CLASSES
    builtins.Exception(builtins.BaseException)
        DeviceException
    builtins.object
        AmplifierChannel
        AmplifierConfiguration
        BluetoothAdapterInfo
        DeviceInformation
        Unicorn

    class AmplifierChannel(builtins.object)
        | The structure containing information about a single channel of the amplifier.
        |

Ln: 350 Col: 4
```

## 15.4. SETTING UP A PROJECT USING VISUAL STUDIO 2015

1. Open Visual Studio 2015.
2. Create a new Python application (File → New → Project → Python).
3. Make sure that the library path is set as described in 15.3.2 Set library path.

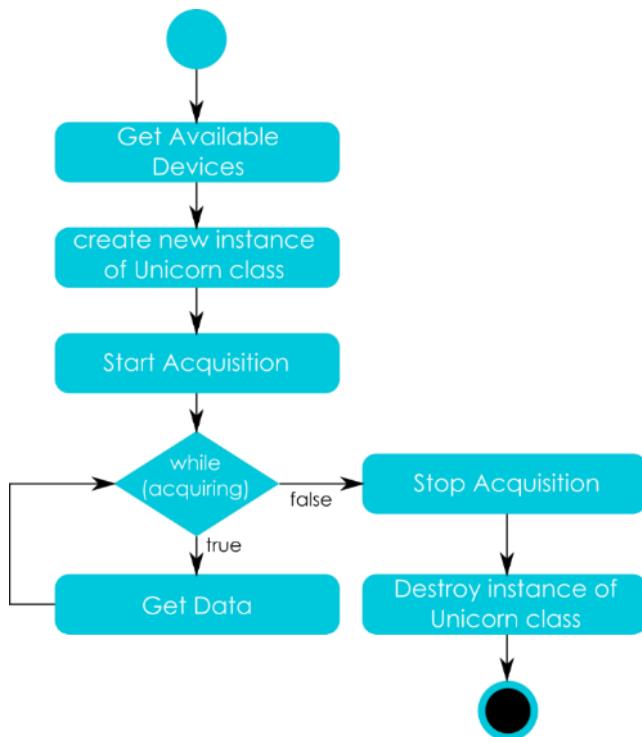


4. Import the Unicorn Python API by importing the library with "import UnicornPy"

```
import UnicornPy
help(UnicornPy)
print("Api Version: ", UnicornPy.GetApiVersion())
```

## 15.5. COMMAND ORDER

To perform a data acquisition using the Unicorn Python API, a defined command execution order is required.



1. Before connecting to a Unicorn device, it is possible to check the operating environment of Unicorn devices and to discover available devices.
2. A connection has to be established to communicate with the Unicorn brain interface. This can be performed by creating an instance of the [Unicorn](#) class. If the instance could be created, the connection attempt was executed successfully. After connecting to a device, it is possible to interact with the device and to call all available public member functions of the [Unicorn](#) class. For example, it is possible to read the current configuration of the device, set a new configuration or start data acquisition.
3. To start data acquisition, [Start Acquisition](#) must be called. After calling [Start Acquisition](#) the Unicorn device is set into acquisition mode and is continuously sending data.
4. Therefore, it is required to read the incoming data stream continuously by calling [Get Data](#) within an acquisition loop. Other API calls (except [Stop Acquisition](#) and [Set Digital Outputs](#)) are not allowed while data acquisition is running.
5. To stop data acquisition, [Stop Acquisition](#) must be called. The Unicorn device will terminate the data stream. The device is still connected. It is possible to interact with the device as long as the corresponding class instance is kept alive.
6. The device gets disconnected as soon as the garbage collector finalizes the corresponding class instance. Delete the Unicorn object to disconnect from the Unicorn and free allocated memory.



## 15.6. UNICORN PYTHON API – REFERENCE

### 15.6.1. CONSTANTS

SupportedDeviceVersion

The Unicorn device version that is valid for this API.

RecommendedDeviceName

The device name of the recommended (delivered) Bluetooth adapter.

RecommendedDeviceManufacturer

The manufacturer of the recommended (delivered) Bluetooth adapter.

SerialLengthMax

The maximum length of the serial number, including the terminating null character.

DeviceVersionLengthMax

The maximum length of the device version, including the terminating null character.

FirmwareVersionLengthMax

The maximum length of the firmware version, including the terminating null character.

StringLengthMax

The maximum string length.

SamplingRate

The sampling rate of the Unicorn Brain Interface.

EEGChannelsCount

The number of available EEG channels.

AccelerometerChannelsCount

The number of available accelerometer channel.

GyroscopeChannelsCount

The number of available gyroscope channel.

TotalChannelsCount

The total number of available channels.



### EEGConfigIndex

The index of the first EEG AmplifierChannel in AmplifierConfiguration.Channels.

### AccelerometerConfigIndex

The index of the first accelerometer AmplifierChannel in AmplifierConfiguration.Channels.

### GyroscopeConfigIndex

The index of the first gyroscope AmplifierChannel in AmplifierConfiguration.Channels.

### BatteryConfigIndex

The index of the Battery AmplifierChannel in AmplifierConfiguration.Channels.

### CounterConfigIndex

The index of the Counter AmplifierChannel in AmplifierConfiguration.Channels.

### ValidationConfigIndex

The index of the Validation Indicator AmplifierChannel in AmplifierConfiguration.Channels.

### NumberOfDigitalOutputs

The number of digital output channels.

### ErrorSuccess

The operation completed successfully. No error occurred.

### ErrorInvalidParameter

One of the specified parameters does not contain a valid value.

### ErrorBluetoothInitFailed

The initialization of the Bluetooth adapter failed.

### ErrorBluetoothSocketFailed

The operation could not be performed because the Bluetooth socket failed.

### ErrorOpenDeviceFailed

The device could not be opened.

### ErrorInvalidConfiguration

The configuration is invalid.



#### ErrorBufferOverflow

The acquisition buffer is full.

#### ErrorBufferUnderflow

The acquisition buffer is empty.

#### ErrorOperationNotAllowed

The operation is not allowed during acquisition or non-acquisition.

#### ErrorConnectionProblem

The operation could not complete because of connection problems.

#### ErrorUnsupportedDevice

The device is not supported with this API's SupportedDeviceVersion.

#### ErrorInvalidHandle

The specified Unicorn handle is invalid.

#### ErrorUnknownError

An unspecified error occurred.

## 15.6.2. EXCEPTIONS

### DeviceException

Exception that returns an error code and a message.

## 15.6.3. STRUCTURES

### AmplifierChannel

The structure containing information about a single channel of the Unicorn Brain Interface.

#### PUBLIC ATTRIBUTES

- string Name  
The channel's name.
- string Unit  
The channel's unit.
- list(float) Range  
The channel's input range as float array. First entry is min value; second is max value.
- bool Enabled



The channel's enabled flag. True to enable channel; false to disable channel.

#### AmplifierConfiguration

The structure containing the Unicorn Brain Interface configuration.

#### PUBLIC ATTRIBUTES

- [list\(AMPLIFIERCHANNEL\)](#) Channels  
A list of [AMPLIFIERCHANNEL](#) representing all channels.

#### BluetoothAdapterInfo

The structure that holds information about the Bluetooth adapter.

#### PUBLIC ATTRIBUTES

- string Name  
The name of the Bluetooth adapter used.
- string Manufacturer  
The manufacturer of the Bluetooth adapter.
- bool IsRecommendedDevice  
Indicates whether the used Bluetooth adapter is a recommended (delivered) device. True if the adapter is a recommended device; false if the adapter is not a recommended device.
- bool HasProblem  
Indicates whether the Bluetooth adapter reports a problem or not. False if the adapter behaves as expected; true if the adapter reports a problem.

#### DeviceInformation

The structure that holds additional information about the device.

#### PUBLIC ATTRIBUTES

- ushort NumberOfEegChannels  
The number of EEG channels.
- string Serial  
The serial number of the device.
- string FwVersion  
The firmware version number.
- string DeviceVersion  
The device version number.
- string PcbVersion  
The PCB version number.
- string EnclosureVersion



The enclosure version number.

#### 15.6.4. STATIC FUNCTIONS

##### 15.6.4.1. Get API Version

double GetApiVersion()

Returns the current API version.

**RETURNS:**

The current API version.

##### 15.6.4.2. Get Available Devices

list<string> GetAvailableDevices(bool discoverPairedDevicesOnly)

Discovers available paired or unpaired devices.

**PARAMETERS:**

discoverPairedDevicesOnly	Defines whether only paired devices or only unpaired devices should be returned. If only unpaired devices should be returned, an extensive device scan is performed. An extensive device scan takes a rather long time. In the meantime, the Bluetooth adapter and the application are blocked. Scanning for paired devices only can be executed faster. If true, only paired devices are discovered. If false, only unpaired devices can be discovered.
---------------------------	--

**RETURNS:**

A list holding available devices.

**EXCEPTIONS:**

<a href="#">DeviceException</a>	Scanning for devices failed.
---------------------------------	------------------------------

##### 15.6.4.2.1. Get Bluetooth Adapter Info

BluetoothAdapterInfo GetBluetoothAdapterInfo()

Evaluates which Bluetooth adapter is currently in use and whether it is the recommended (delivered) Bluetooth adapter.

**RETURNS:**

Information about the used Bluetooth adapter.



Exceptions:

<a href="#">DeviceException</a>	Bluetooth adapter information could not be acquired.
---------------------------------	--

#### 15.6.4.2.2. Is Device Library Loadable

bool IsDeviceLibraryLoadable()

Determines whether all libraries required for device communication are installed.

**RETURNS:**

True if all required native libraries are installed and can be loaded; otherwise false.

## 15.6.5. CLASS UNICORN

### 15.6.5.1. Constructors

Unicorn(string serial)

Initializes a new instance of the [Unicorn](#) class. Connects to the [Unicorn](#) device with the specified serial number. The device gets disconnected as soon as the garbage collector finalizes the corresponding class instance.

**PARAMETERS:**

serial	The serial number of the device to connect to.
--------	--

**EXCEPTIONS:**

<a href="#">DeviceException</a>	The device could not be opened.
---------------------------------	---------------------------------

### 15.6.5.2. Functions

#### 15.6.5.2.1. Get Channel Index

int GetChannelIndex(string *name*)

Uses the currently set [AmplifierConfiguration](#) to get the index of the requested channel within an acquired scan.

**PARAMETERS:**

name	The name of the requested channel.  The default names are: EEG 1 2 3 4 5 6 7 8 Accelerometer X Y Z
------	--



	Gyroscope X Y Z Counter Battery Level Validation Indicator
--	---

**RETURNS:**

The zero-based index of the requested channel in an acquired scan.

**EXCEPTIONS:**

<a href="#">DeviceException</a>	The channel index could not be determined.
---------------------------------	--

#### 15.6.5.2.2. Get Configuration

AmplifierConfiguration GetConfiguration()

Retrieves the current Unicorn Brain Interface configuration from the device.

**RETURNS:**

The configuration of the Unicorn Brain Interface.

**EXCEPTIONS:**

<a href="#">DeviceException</a>	The current Unicorn Brain Interface configuration could not be read.
---------------------------------	--

#### 15.6.5.2.3. Get Data

void GetData(int numberOfScans, bytearray destinationBuffer, int destinationBufferLength)

Reads a specific number of scans into the specified destination buffer of known length. Checks whether the destination buffer is big enough to hold the requested number of scans.

**PARAMETERS:**

numberOfScans	The number of scans to read. The number of scans must be greater than zero. A scan consists of one 32-bit floating point number for each currently acquired channel.
destinationBuffer	The destination bytearray. The destination buffer must provide enough memory to hold the requested number of scans multiplied by the number of acquired channels.  Call <a href="#">GetNumberOfAcquiredChannels()</a> to determine the number of acquired channels. Call <a href="#">GetChannelIndex(string)</a> to determine the index of a channel within a scan. Example: The sample of the battery level channel in the n-th scan is:



	n*GetNumberOfAcquiredChannels() + GetChannelIndex("Battery Level")
destinationBufferLength	The number of floats fitting into the destination buffer.

**EXCEPTIONS:**

<a href="#">DeviceException</a>	Data could not be read.
---------------------------------	-------------------------

**REMARKS:**

The native buffer memory can be allocated as bytearray. It is recommended to allocate buffer memory only once before acquisition starts and reuse it within the acquisition loop. Don't forget to release the allocated memory it is not needed anymore (e.g. after acquisition has been stopped).

#### 15.6.5.2.4. Get Device Information

DeviceInformation GetDeviceInformation()

Reads the device information.

**RETURNS:**

Information about the device that this instance belongs to.

**EXCEPTIONS:**

<a href="#">DeviceException</a>	Device information could not be read.
---------------------------------	---------------------------------------

#### 15.6.5.2.5. Get Digital Outputs

byte GetDigitalOutputs()

Reads the current state of the digital outputs.

**RETURNS:**

The states of the digital output channels in bits. Each bit represents one digital output channel. If a bit is set, the corresponding digital output channel's value is set to high. If a bit is cleared, the corresponding digital output channel's value is set to low.

Examples (the binary representation of each decimal value is shown in parentheses):

0 (0000 0000<sub>b</sub>) → all digital outputs set to low.

170 (1010 1010<sub>b</sub>) → digital outputs 2,4,6,8 are set to high.

255 (1111 1111<sub>b</sub>) → all digital outputs set to high.

**EXCEPTIONS:**

<a href="#">DeviceException</a>	The state of the digital output channels could not be read.
---------------------------------	---



#### 15.6.5.2.6. Get Number of Acquired Channels

```
int GetNumberOfAcquiredChannels()
```

Gets the number of acquired channels according to the currently set Unicorn Brain Interface configuration.

**RETURNS:**

The number of acquired channels.

**EXCEPTIONS:**

<a href="#">DeviceException</a>	The number of acquired channels could not be determined.
---------------------------------	--

#### 15.6.5.2.7. Set Configuration

```
void SetConfiguration(AmplifierConfiguration configuration)
```

Sets the configuration of the device.

**PARAMETERS:**

configuration	The device configuration to set.
---------------	----------------------------------

**EXCEPTIONS:**

<a href="#">DeviceException</a>	The device configuration is invalid or could not be set.
---------------------------------	--

#### 15.6.5.2.8. Set Digital Outputs

```
void SetDigitalOutputs(byte digitalOutputs)
```

Sets the digital outputs to high or low.

**PARAMETERS:**

digitalOutputs	The state of the digital output channels to set in bits. Each bit represents one digital output channel. Set a bit to set the corresponding digital output channel's value to high. Clear a bit to set the corresponding digital output channel's value to low.  Examples (the binary representation of each decimal value is shown in parentheses): 0 (0000 0000 <sub>b</sub> ) → all digital outputs set to low. 170 (1010 1010 <sub>b</sub> ) → digital outputs 2,4,6,8 are set to high. 255 (1111 1111 <sub>b</sub> ) → all digital outputs set to high.
----------------	---



**EXCEPTIONS:**

<a href="#">DeviceException</a>	The state of the digital output channels could not be set.
---------------------------------	--

15.6.5.2.9. Start Acquisition

void StartAcquisition(bool testsignalEnabled)

Starts data acquisition in test signal or measurement mode.

**PARAMETERS:**

testsignalEnabled	Enables or disables the test signal mode. If true, testsignal mode is enabled. If false measurement mode is enabled.
-------------------	--

**EXCEPTIONS:**

<a href="#">DeviceException</a>	Data acquisition could not be started.
---------------------------------	--

15.6.5.2.10. Stop Acquisition

void StopAcquisition()

Stops a currently running data acquisition session.

**EXCEPTIONS:**

<a href="#">DeviceException</a>	Data acquisition could not be stopped.
---------------------------------	--



## 16. UNICORN SIMULINK INTERFACE

The Unicorn Simulink Interface is an interface that enables MATLAB Simulink applications to communicate with Unicorn brain interfaces. The Unicorn Simulink Interface allows users to acquire data from Unicorn devices easily without having to take care of low-level data acquisition issues. The raw binary data stream is converted into numerical values such that the user receives data ready to analyze.

### 16.1. REQUIREMENTS

Software	Properties
MATLAB Simulink	2017a
Signal Processing Toolbox	

### 16.2. FILES ON YOUR COMPUTER

By default, the Unicorn Simulink Interface library is installed to the Documents folder.

C:\Users\<username>\Documents\gtc\Unicorn Suite\Hybrid Black\Unicorn Simulink	Standard installation folder for the Unicorn Simulink Interface library
---	---

Within this directory, subdirectories are generated containing all installed files.

.\Lib	Contains the Unicorn Simulink Interface library for Windows 64-bit
.\Lib\Help	Contains documentation of the Unicorn Simulink Interface library
.\Examples	Contains application examples for the Unicorn Simulink Interface library in MATLAB Simulink



## 16.3. USING THE UNICORN SIMULINK INTERFACE

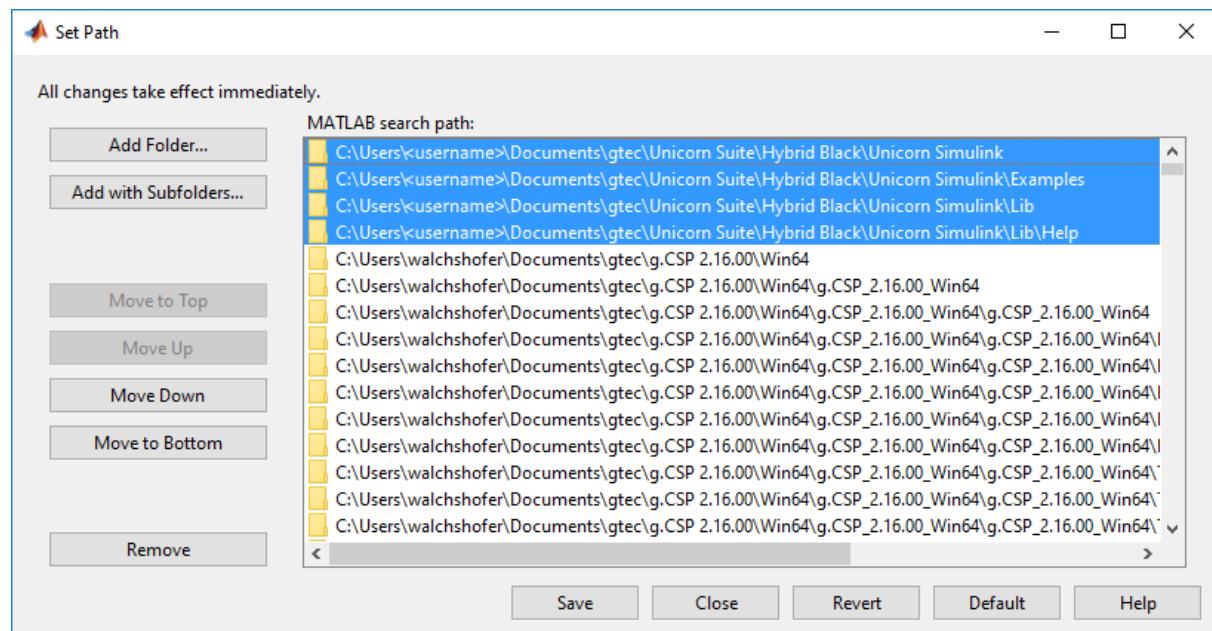
### 16.3.1. ACTIVATE LICENSE

The Unicorn Python library requires a license key to load. Unlock the Unicorn Python API by following the instructions from 12.3 Licensing.

### 16.3.2. SET LIBRARY PATH

You must add the Unicorn Simulink Interface library paths to the MATLAB path to use the library in MATLAB Simulink applications.

1. Open MATLAB 2017a
2. Select "Set Path" in the Home tab to modify the MATLAB path
3. Select "Add with Subfolders" and add the Unicorn Simulink Interface library folders to the path (by default C:\Users\<username>\Documents\gtc\Unicorn Suite\Hybrid Black\Unicorn Simulink)

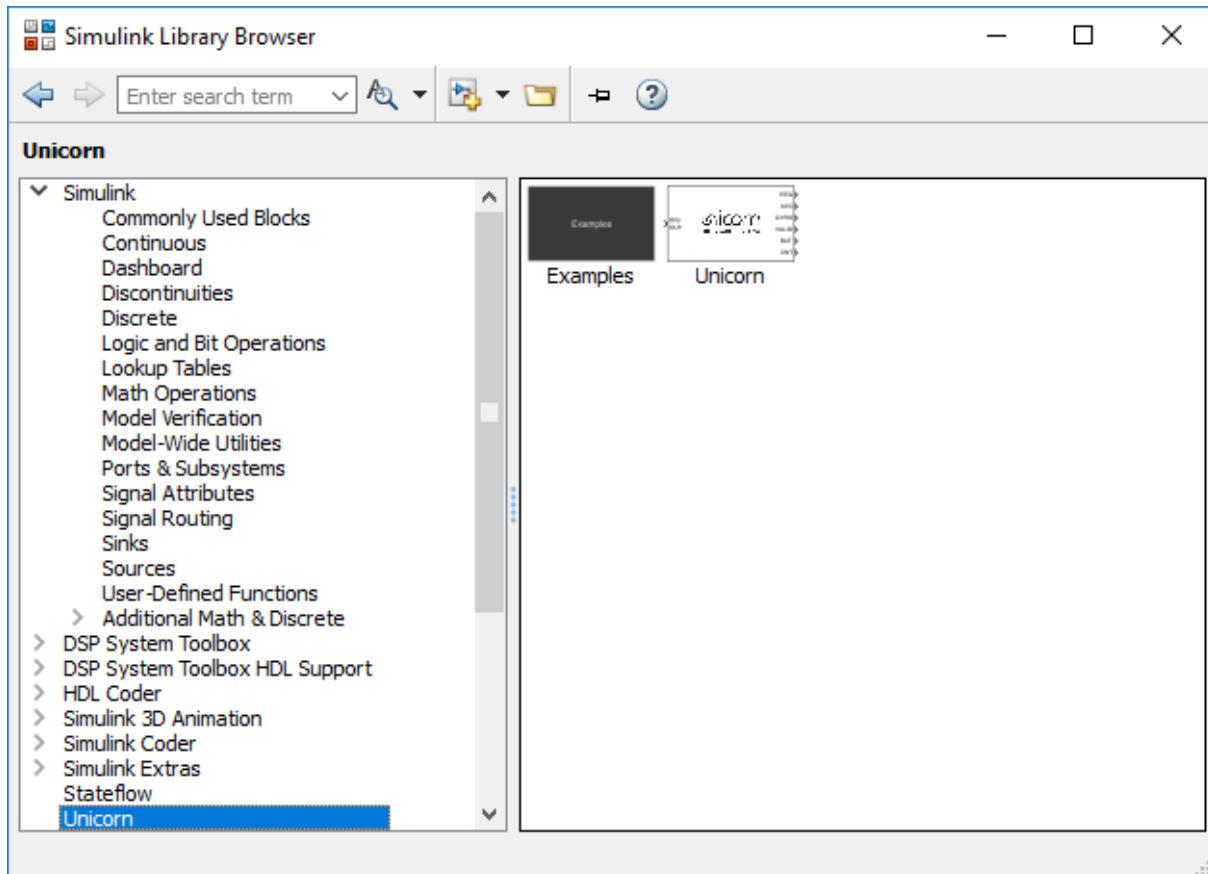


4. Click Save and Close to save the MATLAB path settings.



## 16.4. UNICORN SIMULINK INTERFACE

After the MATLAB Path configuration and license activation, the Unicorn Simulink Interface is ready for use. The Unicorn Simulink Interface is listed in the "Simulink Library Browser". The Unicorn Simulink Interface consists of a Unicorn Block and application examples.





### 16.4.1. UNICORN BLOCK



The Unicorn Block provides a graphical interface to the Unicorn Brain Interface, which can be used in combination with MATLAB Simulink to configure the device and to acquire the data.

#### Inputs

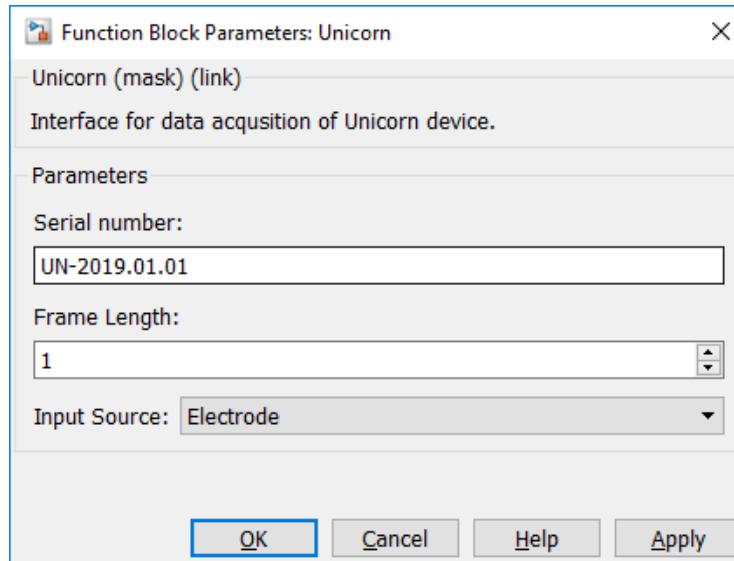
DIG OUT	The input DIG OUT can be used to activate the digital output of the Unicorn brain interface. A value greater or equal to 1 will be treated as TRUE and the output will be set to a high. A value equal to zero is treated as FALSE and the output is set to low.
---------	--

#### Outputs

EEG	The EEG output provides the signal of the 8 analog EEG channels.
ACC	The ACC output provides the signal of the three-axis accelerometer.
GYRO	The GYRO output provides the signal of the three-axis gyroscope.
VALID	The VALID output indicates if samples are lost during the data acquisition.
BAT	The BAT output provides the battery level in percent.
CNT	The CNT output provides the system counter, which is incremented with every received sample during data acquisition.



## Dialog Box



---

Serial Number      Specify the serial number of the Unicorn brain interface used.

---

Frame Length      Select the number of samples in between 1 and 25 acquired per acquisition cycle.

---

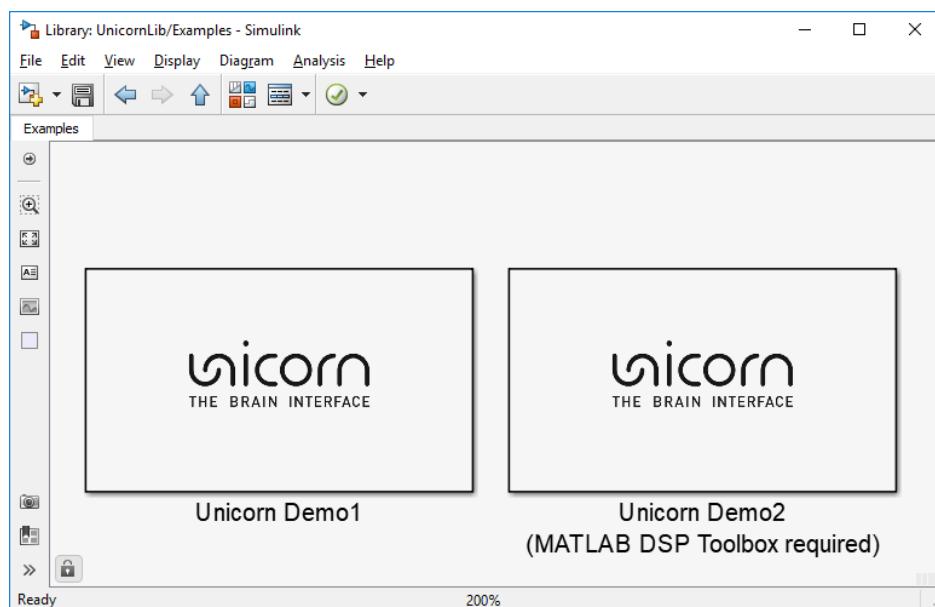
Input Source      Switch between testsignal and EEG (Electrode) as input source for the acquisition.



## 16.4.2. EXAMPLES

The Unicorn Simulink Interface library is delivered with example MATLAB Simulink models.

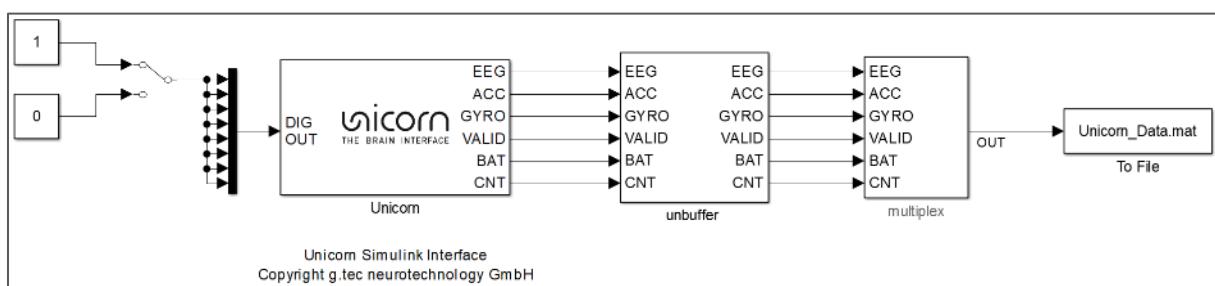
1. Open MATLAB 2017a and the "Simulink Library Browser".
2. Select the "Unicorn" library from the dropdown menu.
3. Double click the "Examples" block.
4. Select one of the example applications.



### 16.4.2.1. Demo 1

This application acquires all available signal channels and writes them to a file.

1. Double click the Unicorn block to configure the device.
2. Change the "Serial Number" to the serial of the device used.
3. Turn on the Unicorn brain interface.
4. Click "Play" to start the MATLAB Simulink model.

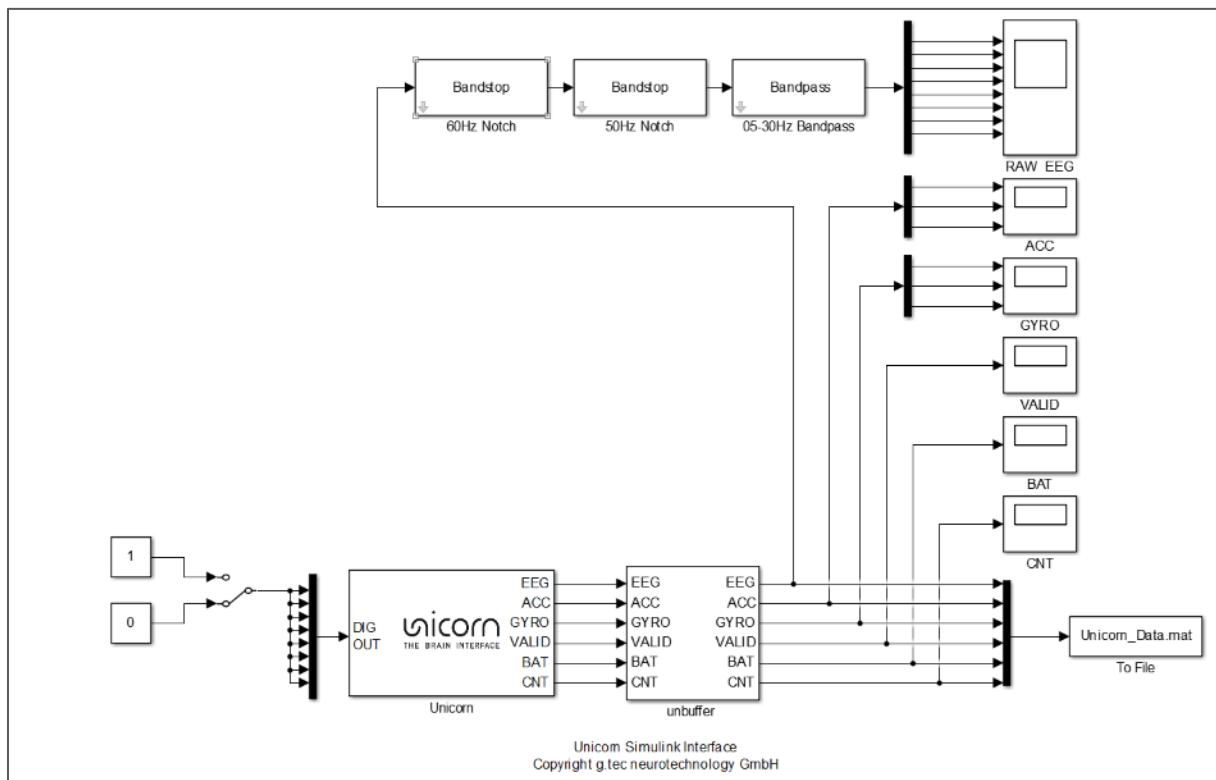




#### 16.4.2.2. Demo 2

This application acquires all available signal channels and writes them to a file. All channels can be visualized with a data viewer. EEG data visualized in the EEG scope is filtered with a 50Hz Notch filter, a 60Hz Notch filter and a 0.5 to 30Hz Bandpass filter.

1. Double click the Unicorn block to configure the device.
2. Change the "Serial Number" to the serial of the device used.
3. Turn on the Unicorn brain interface.
4. Click "Play" to start the MATLAB Simulink model.



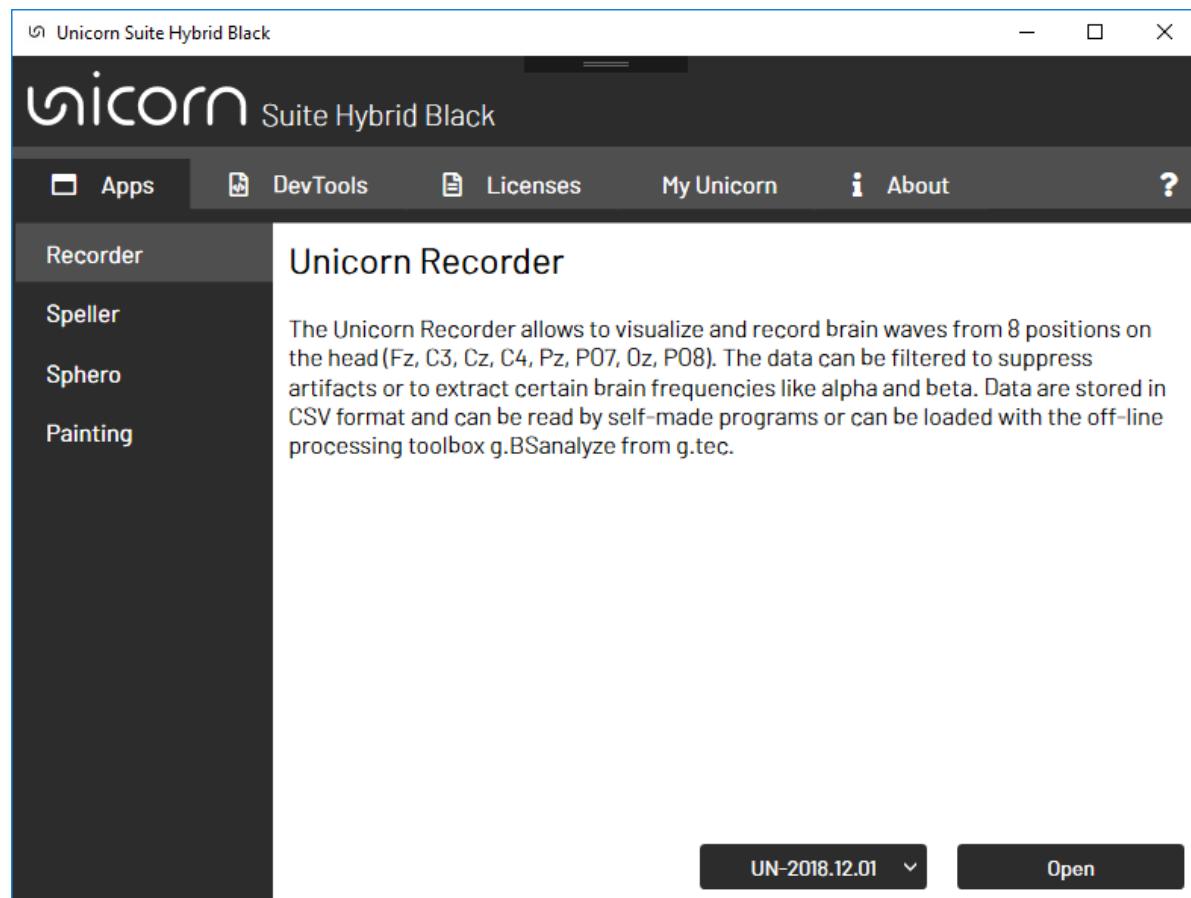


## 17. UNICORN RECORDER

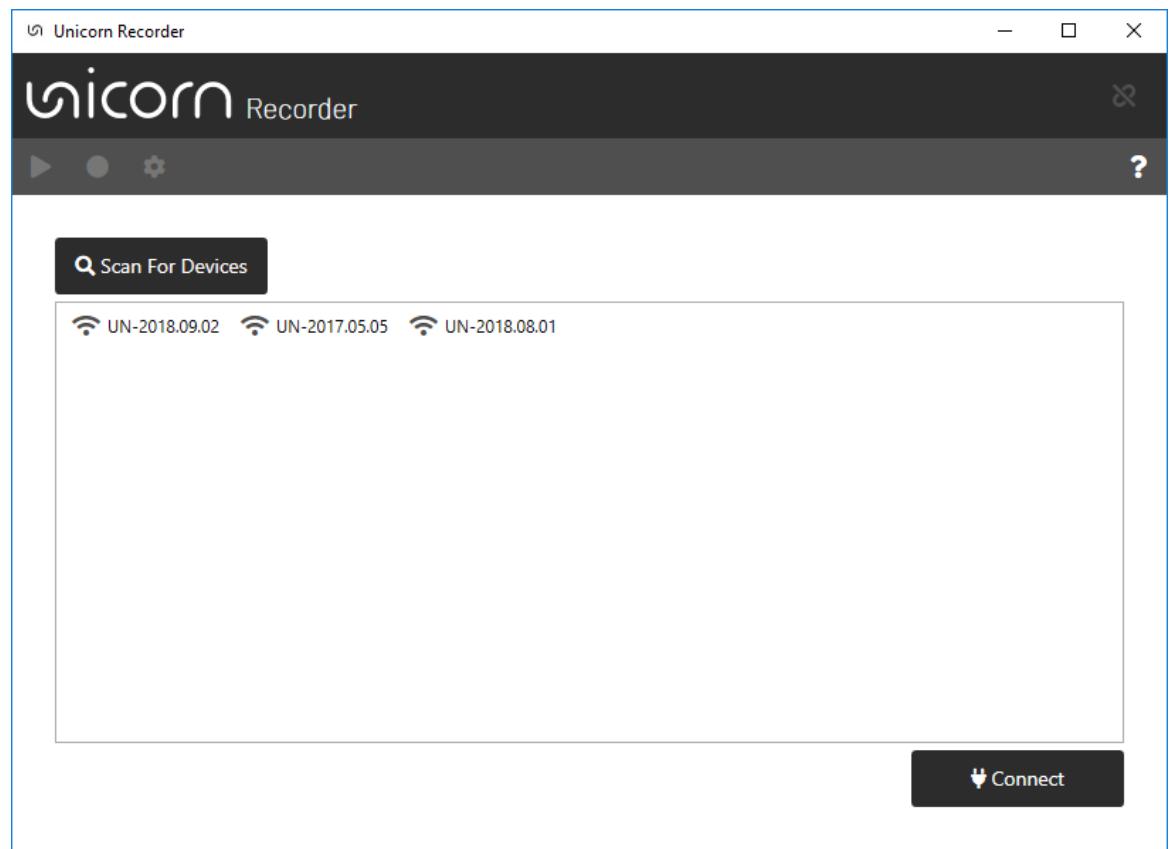
Unicorn Recorder is an application used to acquire, visualize and record data from Unicorn devices. The Unicorn Recorder allows you to process raw EEG with pre-defined Notch and Bandpass filters. Recorded data can be stored in a CSV file.

### 17.1. STARTING UNICORN RECORDER FROM THE UNICORN SUITE

The Unicorn Recorder can be started from the Unicorn Suite. The Unicorn Recorder is listed in the Unicorn Suite in the "Apps" section. Before being able to start Unicorn Recorder, the application must be unlocked using the license manager. The Unicorn must be paired before using it with the Unicorn Recorder. If the Unicorn is paired, the serial number of the device should be listed in the drop-down box in the Unicorn Suite. Select the serial number and press open, start the Unicorn Recorder with the selected device.



This dialog is displayed if the connection attempt failed. Make sure that the device is turned on and paired before using the Unicorn Recorder.





## 1. CONTROL BAR



### 1.1. START/STOP ACQUISITION

This button starts or terminates a data acquisition. The play-button (▶) is displayed if the device is connected and the data acquisition is not running. Press the play-button to start a data acquisition. Data are only displayed if the play button is pressed. The record-button (●) must be pressed to start a recording session. The stop-button (■) is displayed if a data acquisition or recording session is already running. A running data acquisition or recording session can be terminated by pressing the stop-button.

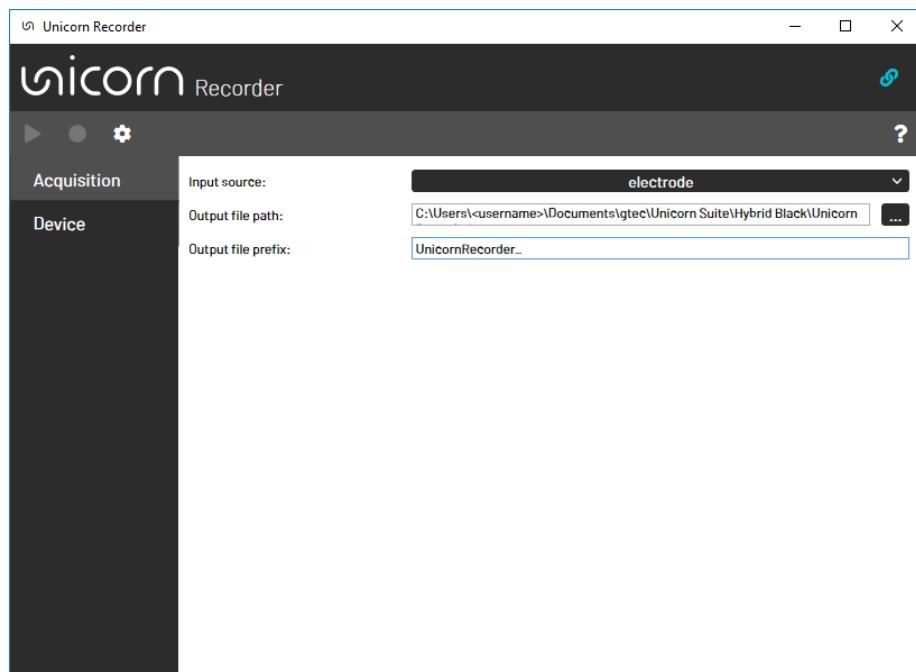
### 1.2. RECORD

The record-button (●) is used to start or stop a recording session. Pressing the record button if no acquisition is running initiates an acquisition and recording. Displayed data are written to a \*.csv file simultaneously. The data logger is added to an acquisition if the record-button is pressed during a running data acquisition. The recording state is indicated by the color of the record-button. A recording session is running if the color of the record-button is cyan. Data are not recorded if the record-button is white. Pressing the record-button during a running recording session causes the recording session to stop. The signal display is not going to be stopped. Pressing the stop-button (■) during a running acquisition causes signal-display and recording to stop. By default, recorded data files are stored at C:\Users\<username>\Documents\gtec\Unicorn Suite\Hybrid Black\Unicorn Recorder\ in a csv file format. All sensor inputs (EEG, accelerometer, gyroscope, battery level, counter and validation indicator) are stored in this file. A timestamp representing the beginning of the recording session is added to the filename of the recorded data file for each new recording session.

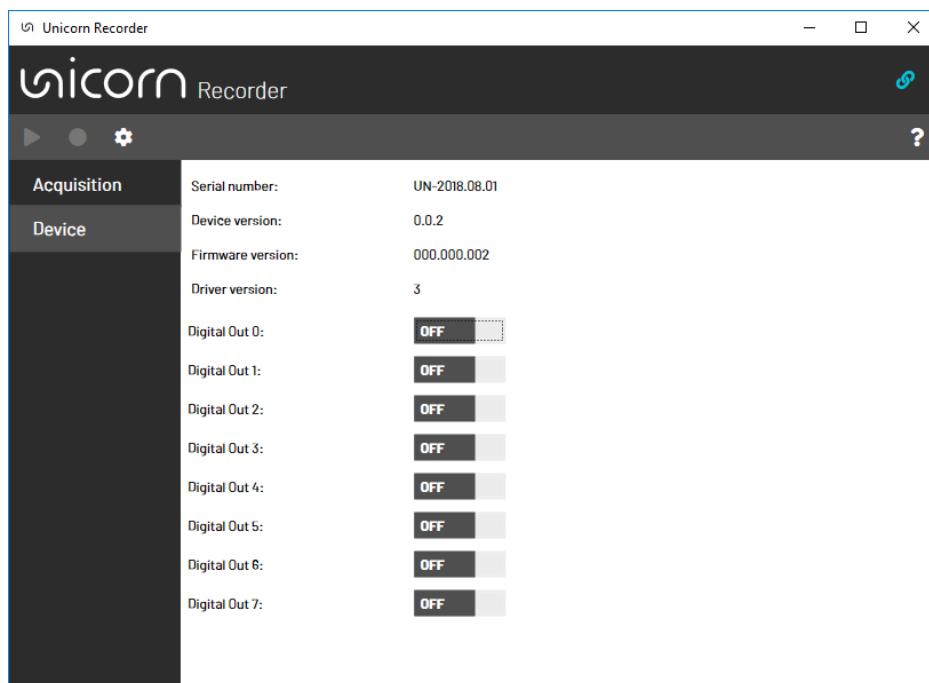
### 1.3. SETTINGS

The settings-button (⚙) opens or closes the settings dialog. The settings dialog is only accessible if no data acquisition is running.

Acquisition parameters can be modified in the acquisition dialog. It is possible to switch the unicorn input source between "electrode" and "test signal". A rectangular test-signal will be acquired if "test signal" is selected. The input of the eight Unicorn Hybrid EEG Electrodes will be acquired if "electrode" is selected. The output file path is the file path where recordings are stored. Make sure that user has write permission to the specified path. Otherwise, the data acquisition can't be started. The output file prefix represents the filename of a recording. A timestamp is added to the prefix for each new recording session.



Device information can be read in the device dialog. The serial number and software versions of the Unicorn Brain Interface currently used are displayed in this dialog. Additionally, the digital outputs can be controlled within this dialog.



## 1.4. HELP

The help-button (?) opens the user manual.



## 2. PROCESSING AND DISPLAY SETTINGS

Scaling:	2.1	$\pm 50 \mu\text{V}$	Bandpass:	2.2	0.5-30 Hz	Notch:	2.3	50 Hz	2
----------	-----	----------------------	-----------	-----	-----------	--------	-----	-------	---

### 2.1. AMPLITUDE RANGE

The amplitude range changes the displayed amplitude range of all EEG channels in the data viewer.

### 2.2. BANDPASS FILTER

The bandpass box allows you to apply predefined IIR bandpass filters to the raw data to extract specified frequency bands and remove artifacts. Predefined bandpass filter settings are:

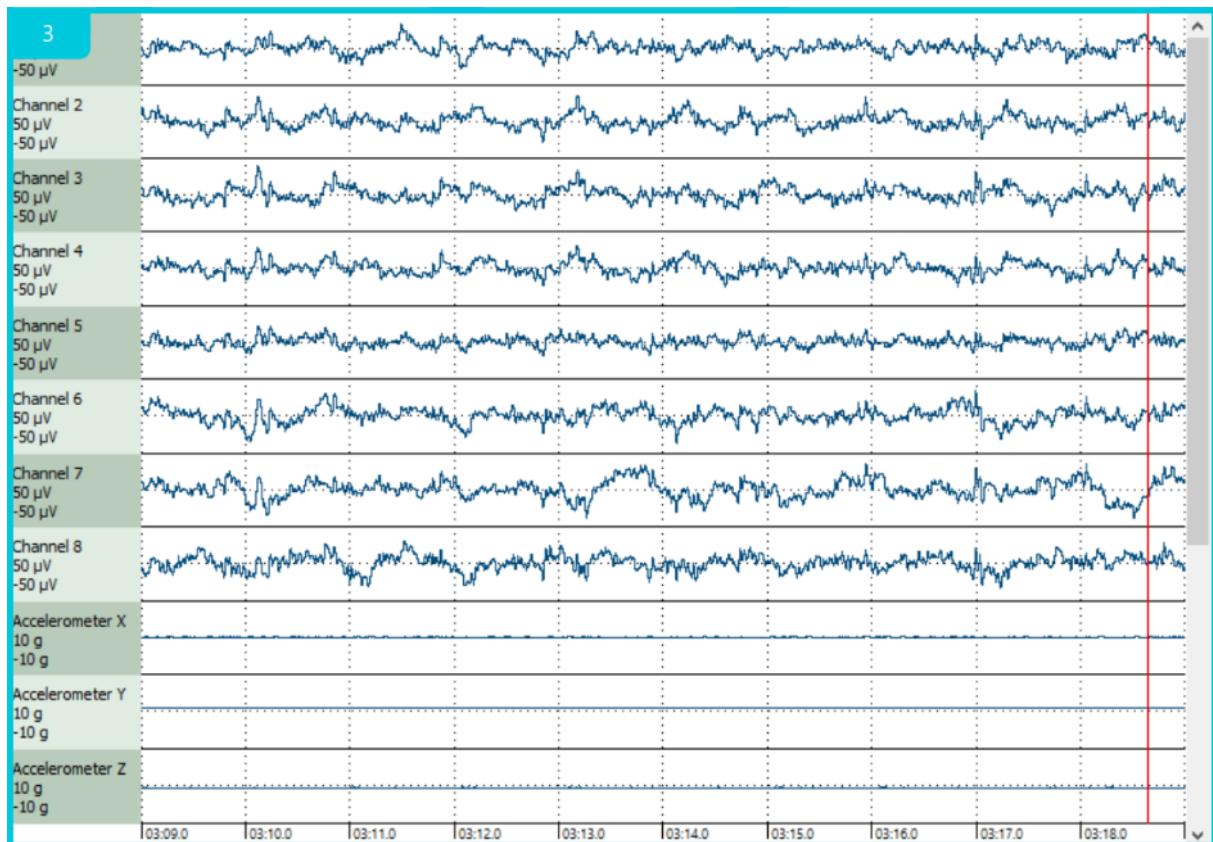
- 0.1 – 30 Hz
- 0.5 – 30 Hz
- 1 – 30 Hz
- 2 – 30 Hz
- 0.1 – 60 Hz
- 0.5 – 60 Hz
- 1 – 60 Hz
- 2 – 60 Hz

### 2.3. NOTCH FILTER

The notch box allows you to apply predefined IIR notch filters to suppress the power line noise. Depending on the country, power line interference may occur around 50 Hz or 60 Hz.

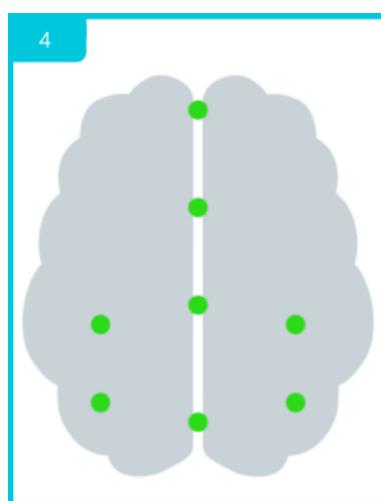


### 3. DATA VIEWER



The Data Viewer displays incoming data in real-time. All channels provided by the Unicorn are listed underneath. The data viewer is limited to a time range of 10 seconds.

### 4. SIGNAL QUALITY SCOPE





The signal quality scope provides feedback about the signal quality. Therefore, the raw EEG is filtered to a certain frequency range where amplitude variations are observed. It takes about 30 seconds until the filters have stabilized and the signal quality scope is reliable. If Unicorn Hybrid EEG Electrodes in the signal quality scope turn yellow, the default amplitude range of EEG signals was exceeded too often, because of artifacts or bad preparation. If Unicorn Hybrid EEG Electrodes in the signal quality scope turn red, the amplitude of the according electrodes is not changing at all. All Unicorn Hybrid EEG Electrodes should turn green if the EEG amplitude stays in a proper range.

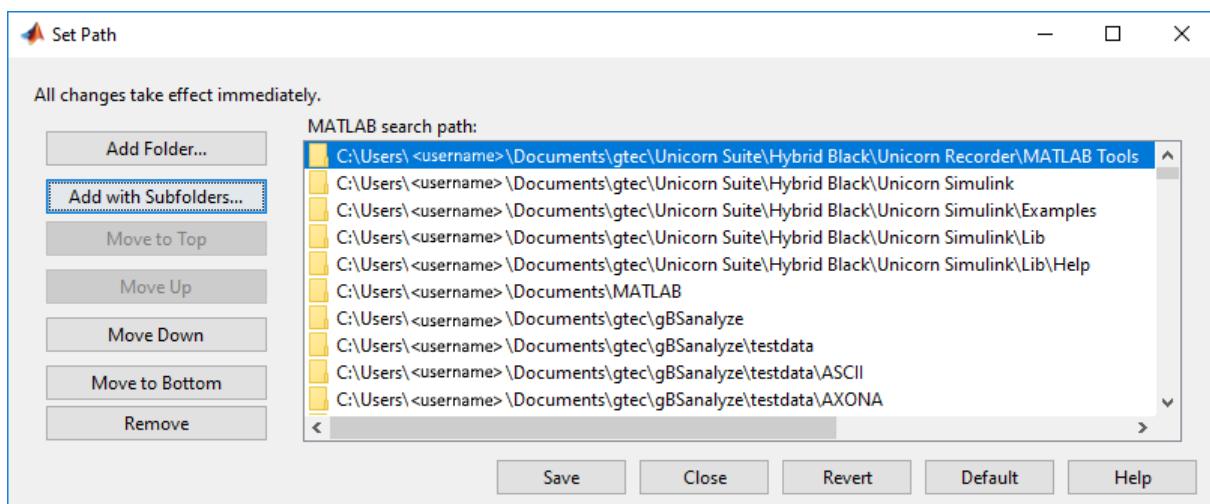
## 17.2. RECORDINGS

Recorded data is stored in a .csv file. Values in the recorded data file are separated by commas. Data are arranged in the file as following:

EEG 1	EEG 2	EEG 3	EEG 4	EEG 5	EEG 6
EEG 7	EEG 8	Accelerometer X	Accelerometer Y	Accelerometer Z	Gyroscope X
Gyroscope Y	Gyroscope Z	Battery Level	Counter	Validation Indicator	

### 17.2.1. MATLAB TOOLS

An import script is provided to read the Unicorn Recorder file into Matlab. Open Matlab and select "Set Path". Add the folder "C:\Users\<username>\Documents\gtec\Unicorn Suite\Hybrid Black\Unicorn Recorder\MATLAB Tools" to the Matlab search path and click save.



```
[datastruct] = unicornrecorder_read(filename)
```

Imports a Unicorn Recorder data file and returns data, as well as recording information.



**PARAMETER:**

filename	String containing the name of file to import.
----------	---

**RETURN:**

Matlab structure containing all information stored in the Unicorn Recorder data file.

**REMARKS:**

Type "help unicornrecorder\_read" into the Matlab command window for further information.

Example usage:

```
data = unicornrecorder_read('UnicornRecorder_20190122_220912.csv');
```

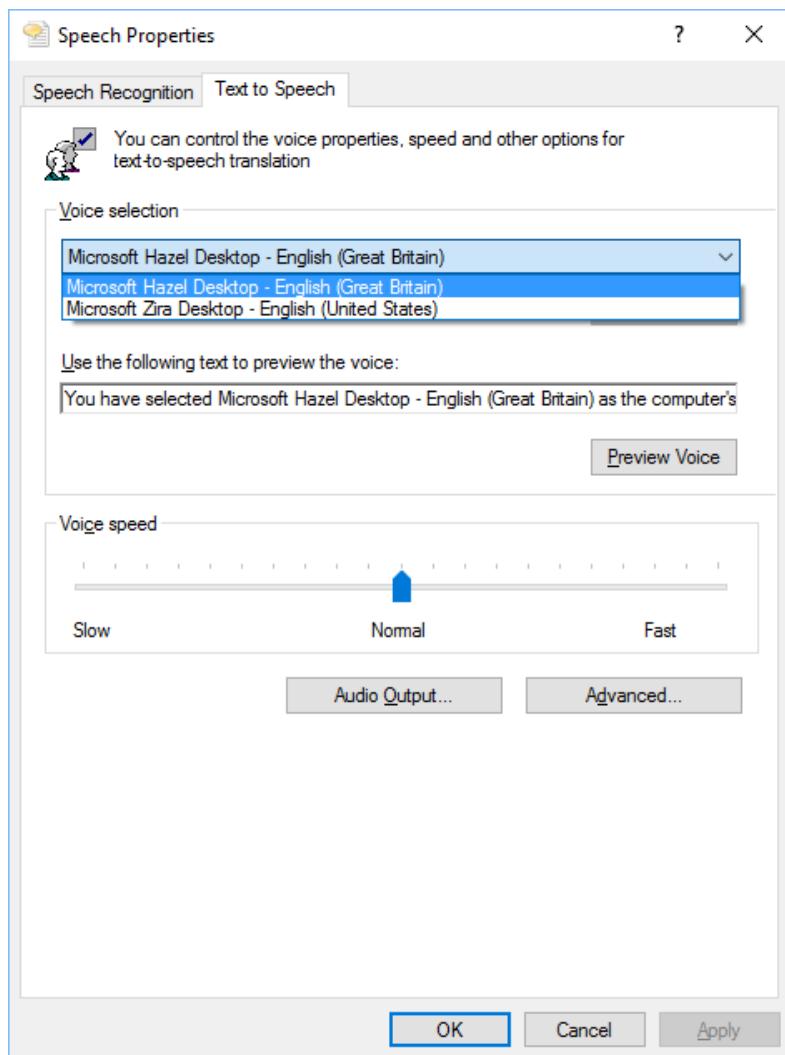


## 18. UNICORN SPELLER

Unicorn Speller is a spelling system that uses the P300 complex. Numerous items such as letters flash the screen. The user must silently count each time the item he/she wants to select flashes, and ignore all other flashes.

### 18.1. REQUIREMENTS FOR TEXT-TO-SPEECH OUTPUT

Unicorn Speller uses the text-to-speech engines and voices that are supported by the operating system for text-to-speech output. A voice must be displayed and working in the Text to Speech link of the control panel's Speech Recognition configuration page before it can be used with Unicorn Speller. When obtaining voices of different styles and languages from third-party vendors, ensure that these voices are compatible with the current version of your operating system.





Select Speech Recognition from the operating system's control panel and click the Text to Speech item on the left to open the Speech Properties window. A voice must be listed under 'Voice selection' before it can be used with Unicorn Speller.



### **ATTENTION:**

A voice that is not compatible with the currently used operating system version will not work and can cause the program to crash or work improperly!

## **18.2. FILES ON YOUR COMPUTER**

Unicorn Speller files – are stored under (assuming that the default path setting is used):

C:\Program Files\gtec\Unicorn Suite\Hybrid Black\Unicorn Speller

Unicorn Speller user directory – is created under:

C:\Users\<username>\Documents\gtec\Unicorn Suite\Hybrid Black\Unicorn Speller

Predefined board configurations – are created on each start of the program in the following subfolder of the Unicorn Speller user directory:

C:\Users\<username>\Documents\gtec\Unicorn Suite\Hybrid Black\Unicorn Speller\Boards

Calibration files – will be stored on calibration in the following subfolder of the Unicorn Speller user directory (if no explicit path is specified on calibration):

C:\Users\<username>\Documents\gtec\Unicorn Suite\Hybrid Black\Unicorn Speller\Classifiers

Word prediction dictionaries – are installed into the following subfolder of the Unicorn Speller user directory:

C:\Users\<username>\Documents\gtec\Unicorn Suite\Hybrid Black\Unicorn Speller\Dictionarys



### 18.3. TERMS

The table below lists the terms used throughout the document along with a short description:

Term	Description
Dark time	The duration where no item flashes directly after a previous item has flashed.
Flash time	The duration the item(s) flash when they're flashed on.
Flashing	The actual mode of operation of Unicorn Speller, where the items on the board are flashing and Unicorn Speller tries to select the item that the user is silently counting based on the user's EEG signals.
Calibration item	The item that the user wants to select. This item should be selected by Unicorn Speller after some flash cycles. For calibration, calibration items must be predefined in the calibration items panel to provide Unicorn Speller information if the measured EEG currently belongs to a calibration or a non-calibration item. Based upon this information, Unicorn Speller learns to separate signals belonging to calibration items from signals belonging to non-calibration items.
Non-calibration item	Each item in the board that is currently not a calibration item.
Randomized patterns mode	During flashing, a randomized selection of items (a.k.a. pattern) distributed across the whole board flash at the same time. The patterns flash consecutively. One single flash cycle is complete when each item on the board has flashed exactly once.
Row/column mode	During flashing, all items in a whole row or column flash at the same time. The rows and columns flash consecutively. One single flash cycle is complete when each item on the board has flashed exactly once.
Single character mode	During flashing, only one item on the board flash at any time. Each item flashes consecutively. One single flash cycle is complete when each single item of the board has flashed exactly once.
Flash cycle	A group of flashes such that each item of the board has flashed exactly once.
Free-spelling mode	Unicorn Speller is calibrated and selects items based on the user's EEG signals during flashing.



---

#### Copy-spelling mode

Before the flashing begins, the user can specify the desired items he or she is going to focus consecutively during the following flashing session by adding them to the calibration items panel. Specifying calibration items has no influence on the mode of operation during flashing. Unicorn Speller behaves just as in the free-spelling mode. The calibration items serve just as an orientation aid for the user. This mode is used to verify a successful calibration.

---

#### Calibration

To calibrate Unicorn Speller, the user has to preselect several calibration items he or she is going to focus consecutively during the following flashing cycles by adding them to the calibration items panel. After the flashing ends, Unicorn Speller learns to separate the EEG signals of the EEG segments that belonged to calibration items from those that belonged to the non-calibration items.

---

## 18.4. MOUNTING UNICORN HYBRID EEG ELECTRODES

Unicorn Speller works with real-time EEG data measured and transmitted by the Unicorn Brain Interface. Unicorn Speller is intended to be used with eight EEG channels using unipolar derivation. The table below lists the electrode assignment.

Electrode position	Color	Description
Left mastoid	Yellow	L
Right mastoid	Blue	R
Fz	Red	Channel 1
C3	Red	Channel 2
Cz	Red	Channel 3
C4	Red	Channel 4
Pz	Red	Channel 5
PO7	Red	Channel 6
Oz	Red	Channel 7



---

PO8

Red

Channel 8

---

## 18.5. STARTING UNICORN SPELLER

After finishing the installation, Unicorn Speller can be started with the Unicorn Brain Interface. In order to use Unicorn Speller, several Unicorn Hybrid EEG Electrodes (see section 18.4) must be attached the user's head and connected to the Unicorn Brain Interface.

### 18.5.1. STARTING UNICORN SPELLER FROM THE UNICORN SUITE

The Unicorn Speller can be started from the Unicorn Suite. The Unicorn Speller is listed in the Unicorn Suite in the "Apps" section. Before being able to start Unicorn Speller, the application must be unlocked using the license manager. The Unicorn Brain Interface must be paired before using it with the Unicorn Speller. If the Unicorn Brain Interface is paired, the serial number of the Unicorn Brain Interface should be listed in the drop-down box in the Unicorn Suite. Select the serial number and press open, start the Unicorn Speller with the selected Unicorn Brain Interface.

The screenshot shows the Unicorn Suite application window. The title bar reads "Unicorn Suite Hybrid Black". The main menu bar includes "Apps", "DevTools", "Licenses", "My Unicorn", "About", and a question mark icon. On the left, a sidebar lists "Recorder", "Speller" (which is currently selected), "Sphero", and "Painting". The main content area is titled "Unicorn Speller". It describes the app as a spelling system using the P300 component, featuring a predictive speller and dynamic stopping. At the bottom, there are buttons for "Build dictionary", "Test network output", a dropdown menu showing "UN-2018.12.01", and "Open".



This dialog is displayed if the connection attempt failed. Make sure that the Unicorn Brain Interface is turned on and paired before using the Unicorn Recorder.

### 18.5.2. STARTING UNICORN SPELLER VIA COMMAND LINE

Change your directory in the command line window to the folder where Unicorn Speller has been installed. You can start Unicorn Speller now by using the following command line form:

```
intendiX.exe [-device <deviceType>] [-deviceSerial <deviceSerialNumber>] [-ch <numChannels>] [-board <filename>] [-record <filename>] [-testmode]
```

The table below lists a description of the supported command line options:

Option	Description
-device <deviceType>	Specifies the type of Unicorn Brain Interface that is used with the application. Supported values for <deviceType> are <code>Unicorn</code> . The default value is <code>Unicorn</code> .
-deviceSerial <deviceSerialNumber>	The serial number of the Unicorn Brain Interface that is used as default. If not specified, the most recently used device is used.
-ch <numChannels>	Specifies the number of channels that shall be used. Valid values range from 1 to 8 for Unicorn Brain Interfaces. If the specified number of channels is less than available number channels, the application uses only the channels from 1 to <numChannels>. Attention: A previous calibration performed with a different number of channels will not work.
-board <filename>	The filename of the board to load as default. If not specified, the most recently loaded board is used. If a relative path or only the filename is specified, the path is relative to the default predefined board configurations directory (see section 18.2).
-record <filename>	All EEG data received from the Unicorn Brain Interface will be written to an ASCII text file specified by <filename>. The application writes permanently to the file as long as the application is running. Reading from the file is permitted from other programs while the application is running and writing to the specified file.



---

If a relative path or only the filename is specified, the path is relative to the default Unicorn Speller user directory (see section 18.2).

---

-testmode

If specified, the Unicorn Brain Interface sends a test signal to the PC, ignoring the input signals of the connected Unicorn Hybrid EEG Electrodes. This mode is for testing connection and proper working of the Unicorn Brain Interface only! Using this test mode causes Unicorn Speller to work improperly because it never receives real EEG signals in this mode!

---

#### 18.5.2.1. File format of the recorded samples using the –record option

When starting Unicorn Speller via command line with the `-record` option specified, the created ASCII text file is built up in the following way, each line representing one sample (assuming the used number of channels is `c` and there are `n` columns in the whole file separated by the tabulator):

Column (separated by tabulator)	Content
1	Measuring time of the sample in seconds
2...n-2	Sample values of channels 1 to <code>ch</code> in microvolt ( $\mu$ V), each column representing one channel
$n-1$	The one-based indices of all items that were flashing while this sample has been measured, separated by commas. A value of zero indicates that no items were flashing at the time of measuring this sample. The index of the top-left item of the board is 1, index of an item in the column <code>col</code> (1-based) and the row <code>row</code> (1-based) will be calculated as: $index = (row-1) * \text{number of columns} + col$ .
$n$	A value of 0 or 1 indicates that the flashing item or row/column with index specified in column <code>n-1</code> at sample time was a calibration item/row/column (1) or not (0). If no information about calibration items is available, a 0 is written as well.

On closing the application, a header will be inserted in the first few rows of the file to make the file compatible with g.BSanalyze, a toolbox from g.tec medical engineering GmbH created for MATLAB and SIMULINK. Details on this header can be found in the user manual of g.BSanalyze.



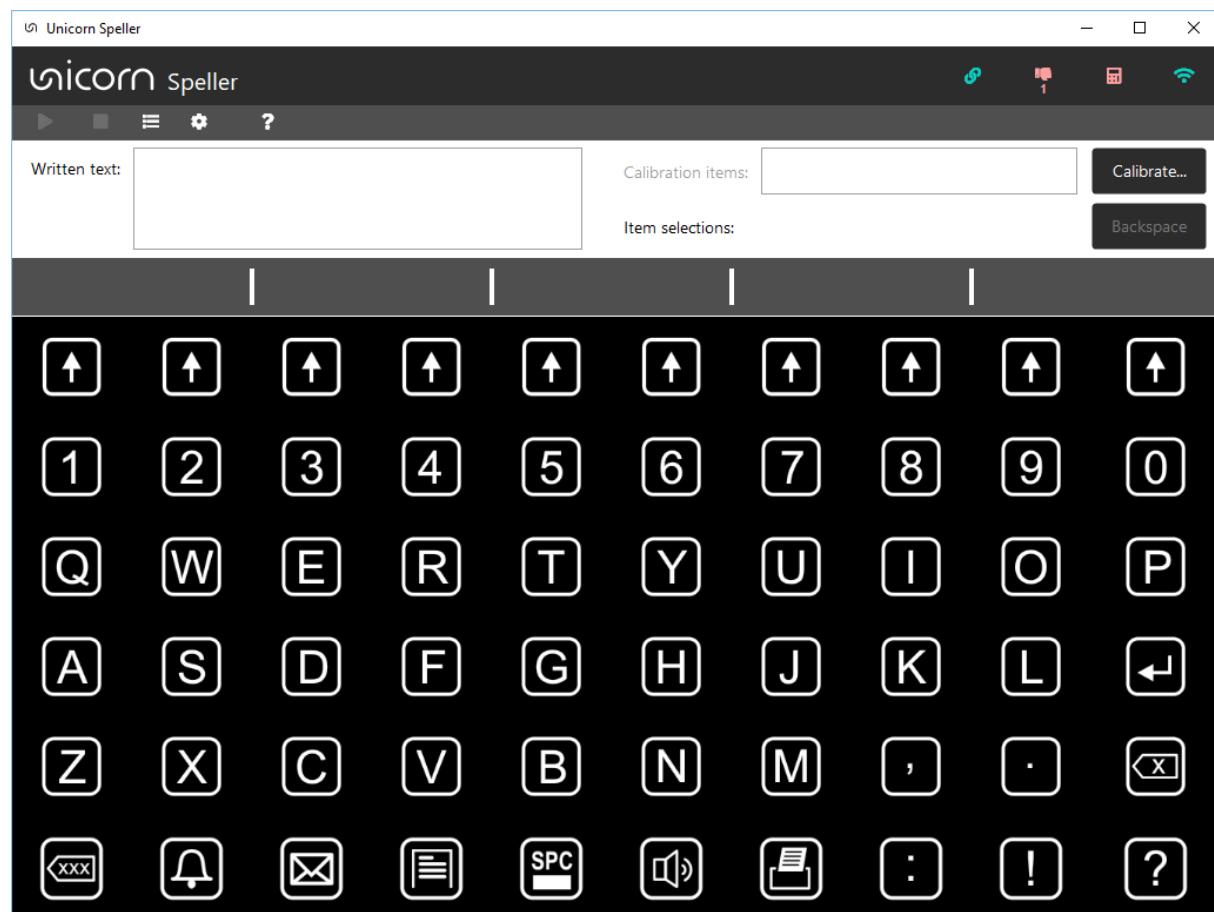
### 18.5.3. BEHAVIOR OF UNICORN SPELLER ON STARTUP

The predefined boards with filename StandardBoard.ibc, and SimpleBoard.ibc and EmptyBoard.ibc will be restored at each startup of Unicorn Speller in the default directory for predefined board configurations (see section 18.2). Therefore, changes made to these files are not persistent.

On startup, Unicorn Speller connects the most recently used Unicorn Brain Interface with the most recently used settings and the main window is shown. If Unicorn Speller can load the most recent calibration, it immediately starts the free-spelling mode with the last used configuration. Otherwise, the user must manually prepare and start spelling.

## 18.6. UNICORN SPELLER

When starting Unicorn Speller, the main window of the application is shown.



The main window consists of 5 main elements:

1. The **status bar** at the top on the right of the main window indicates the current state of the application.
2. The **menu bar** below the status bar lets you access all components of the application.



3. The Written text box displays the corresponding **Output Text** of a selected item. Right click on that area to open a context menu where size and style of the font of the written text can be changed.
  - a. The **Calibration items** panel displays the selected calibration items during calibration or in copy-spelling mode. When no calibration items are selected, this panel is grayed out. With the **Backspace** button, items can be removed from the calibration items panel.
  - b. The Item selections panel displays previously selected items. In copy-spelling mode, all selected items are displayed. In free-spelling mode, only the last three selected items are displayed.
4. The completion suggestions panel (between the Written text box and the board area) displays probable word completion suggestions based on the currently written text. A completion suggestion can be selected and inserted into the **Written text** by selection of one of the corresponding up-arrow items right below in the first row of the board.
  - a. The **completion suggestions** panel is not shown if word prediction is disabled. In this case, selection of an up-arrow item would have no effect. See sections 18.15 and 18.11.1 for details on word prediction.
5. The board area in the middle of the main window displays the set of possible selectable items. Double-clicking on an item in standby mode will add it to the calibration items panel.



**NOTE:**

You can resize the main window according to your preferences and needs.



**NOTE:**

All user specific settings (such as connection, timing,...) and used files (such as calibration file, board configuration file,...) of the application will be stored to a default configuration file when Unicorn Speller is closed. This default configuration is used next time you start Unicorn Speller.

### 18.6.1. STATUS INDICATORS

The status bar contains four status indicators:

Status indicator	Possible states	Description
Connection	 Data acquisition running	The application is connected to the Unicorn Brain Interface and receives EEG data.
	 Data acquisition stopped	The application can't receive data from a Unicorn Brain Interface. Please ensure that the Unicorn Brain Interface is turned on and on standby, then check



Status indicator	Possible states	Description
		connection settings in menu  → <b>Connection...</b>
	Connecting to device...	The application tries to establish a connection to the selected Unicorn Brain Interface. To change the Unicorn Brain Interface, go to menu  → <b>Connection...</b>
	Not connected	The application is not connected to a Unicorn Brain Interface and can't receive data. Please ensure that the Unicorn Brain Interface is turned on and on standby, then check connection settings in menu  → <b>Connection...</b>
Signal quality	Good signal quality Poor signal quality	The specified signal quality criteria are met. Here the signal quality is rated to be good.  The specified signal quality criteria are not met for at least one channel (the name of the first channel that doesn't meet the criteria is displayed directly below, and the name of all channels that do not meet the criteria is displayed when the mouse hovers over). Signal quality is said to be poor. You can inspect the signals in the data viewer (.
Calibration state	Calibrated Not calibrated	The last used calibration could be loaded and is ready to operate.  <b>ATTENTION:</b> A calibration file that could be loaded but wasn't created with the currently used number of channels will not work even if the status indicator says it is okay.
Network output state	Network output enabled	Unicorn Speller must be calibrated, or an existing calibration file must be selected (menu  → <b>Selection method...</b> ).
		Network output is enabled and no error occurred on sending the most recently selected item over the network.



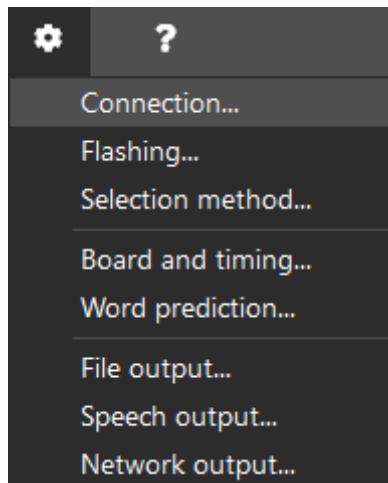
Status indicator	Possible states	Description
	Couldn't send item over network	If network output is enabled (menu <b>⚙️ → Network output...</b> ) and an item couldn't be sent on selection, this status message will be presented. Please ensure that your firewall allows outgoing connections for Unicorn Speller on the specified port.
	Network output disabled	Network output is disabled.

## 18.7. ESTABLISH A CONNECTION TO THE UNICORN BRAIN INTERFACE

### 18.7.1. CONNECTION VIA BLUETOOTH

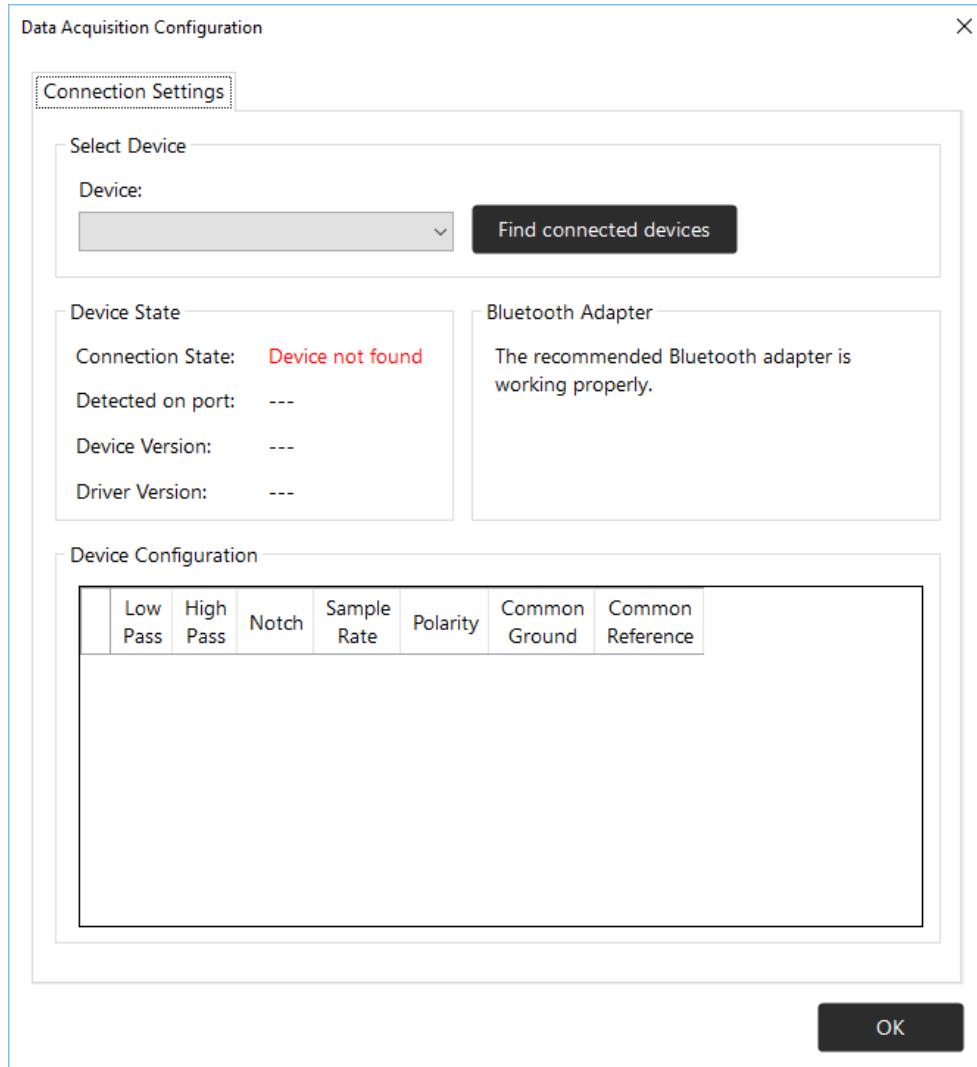
Before you can establish a Bluetooth connection between your Unicorn and your PC for the first time, you first have to pair your Unicorn with your PC. Unicorn Speller can connect to the Unicorn once it is paired with the PC, turned on and in standby mode.

Open the **Data Acquisition Configuration** dialog to configure the settings of the connection by selecting **⚙️ → Connection...** from the menu bar as illustrated.



Select **⚙️ → Connection...** from the menu to edit connection settings of Unicorn Speller.

The **Data Acquisition Configuration** dialog will open as shown below.



If your Unicorn Brain Interface is not listed in the drop-down menu but it is connected to your PC, click the **Find connected devices** button. Unicorn Speller now searches for paired Unicorn Brain Interfaces and lists the serial numbers of the found devices in the drop-down list afterwards. While the application tries to establish a connection to the first found Unicorn Brain Interface, the **Connection State** indicator changes its state to **Connecting....** This may take several seconds.



#### ATTENTION:

If the **Data Acquisition Configuration** dialog states that an unrecommended Bluetooth adapter or no Bluetooth adapter is detected, please connect the recommended Bluetooth adapter that was shipped together with your Unicorn and disconnect all other Bluetooth adapters. Also, disable a possibly built-in Bluetooth adapter. Otherwise, the Unicorn might not work properly. Search again for connected Unicorns when the recommended Bluetooth adapter is connected.



You can select a different Unicorn Brain Interface by selecting the desired serial number from the drop-down list. When a connection could be established, the **Connection State** indicator changes its state to Device found and reads the configuration of the Unicorn Brain Interface (see Error! Reference source not found.).

Data Acquisition Configuration

Connection Settings

Select Device

Device:

UN-2017.08.30

Find connected devices

Device State

Connection State: Device found

Detected on port: Bluetooth SPP

Device Version: 0.0.1

Driver Version: 1

Bluetooth Adapter

The recommended Bluetooth adapter is working properly.

Device Configuration

	Low Pass	High Pass	Notch	Sample Rate	Polarity	Common Ground	Common Reference
► Channel 1	none	none	none	250 Hz	unipolar	no	no
Channel 2	none	none	none	250 Hz	unipolar	no	no
Channel 3	none	none	none	250 Hz	unipolar	no	no
Channel 4	none	none	none	250 Hz	unipolar	no	no
Channel 5	none	none	none	250 Hz	unipolar	no	no
Channel 6	none	none	none	250 Hz	unipolar	no	no
Channel 7	none	none	none	250 Hz	unipolar	no	no
Channel 8	none	none	none	250 Hz	unipolar	no	no

OK

**NOTE:**

The dialog displays the filter settings of the hardware (i.e. the Unicorn Brain Interface) only. Unicorn Speller additionally uses software filters for signal processing, but their settings are not displayed in the **Data Acquisition Configuration** dialog.

**NOTE:**

When a serial number is selected in the drop-down list but the **Connection State** keeps indicating that the Unicorn Brain Interface is not found, ensure that the Unicorn Brain Interface is turned on and paired.

Click **OK** to establish a connection with the selected Unicorn, configure it with the chosen options and start data acquisition.

### 18.7.2. LOSS OF CONNECTION

When Unicorn Speller is currently not connected to an Unicorn Brain Interface or has lost its Bluetooth connection, it periodically tries to reconnect to the Unicorn Brain Interface every few seconds.

### 18.7.3. NO CONNECTION BETWEEN UNICORN SPELLER AND UNICORN BRAIN INTERFACE

In certain cases, Unicorn Speller either can't establish a connection with the Unicorn Brain Interface or an existing connection gets lost. This could result if the battery is too low, the Unicorn Brain Interface is turned off or a subject leaves the Bluetooth transmission range.

When Unicorn Speller is not able to reconnect to the Unicorn Brain Interface, please turn off the Unicorn Brain Interface and turn it on again afterwards. The Unicorn Brain Interface will return to standby mode and Unicorn Speller can reconnect to the Unicorn Brain Interface.

## 18.8. CHECK SIGNAL QUALITY

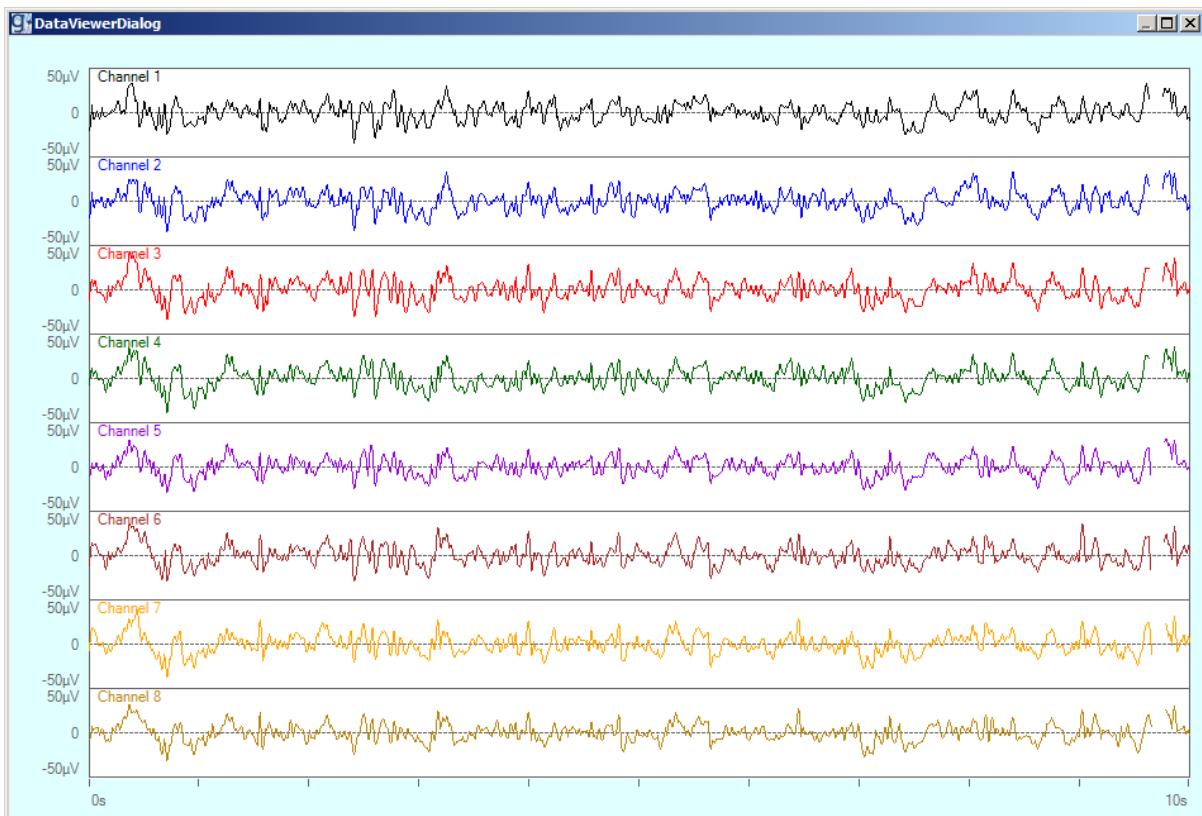
Before you start working with Unicorn Speller, you should ensure that the incoming signals are high quality, so that Unicorn Speller will operate under optimal conditions. You can either investigate the incoming signals using the built-in data viewer or you can obtain information about the signal quality of each channel by defining threshold values of the integrated quality criteria. Both options are described in this section.

### 18.8.1. THE DATA VIEWER

Open the Data Viewer by selecting from the menu bar. The data viewer will open.



Select from the menu bar to open the data viewer window.



The data viewer shows the signal of each channel in a regular scaling range between -50  $\mu$ V and +50  $\mu$ V ( $\mu$ V = microvolts) in a separate color for a period of 10 seconds. Data are overwritten from left to right. If signal amplitudes exceed the scaling range, the signals will overlap the area of channels below or above. The signals will only be truncated at the top and the bottom of the whole drawing area, respectively. Corresponding channel numbers and signals are plotted in the same color.

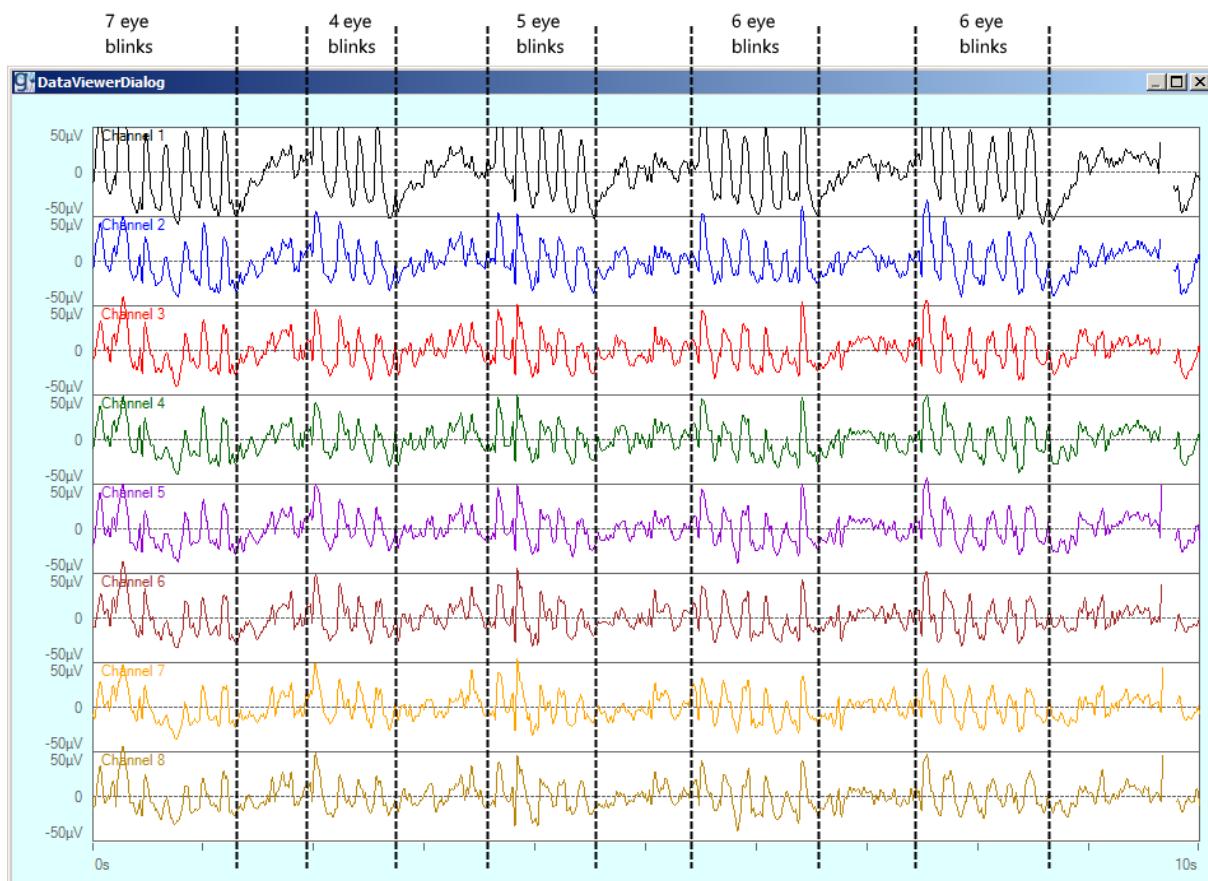


#### NOTE:

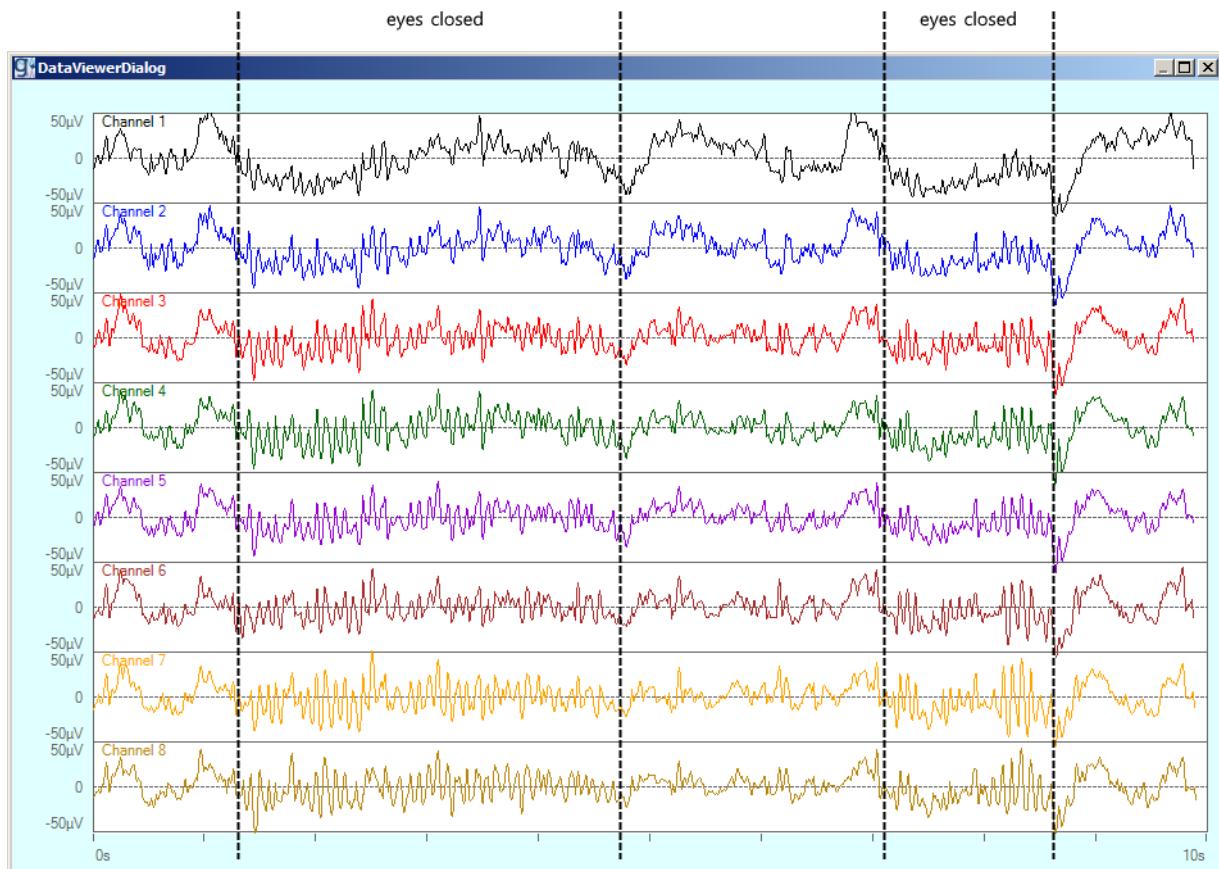
Spontaneous EEG signal amplitudes are typically within the range of  $\pm 50 \mu$ V (using monopolar derivation techniques). If the signals are much smaller or much bigger, please check your electrode settings.

You can quickly check the signal quality by advising the user to perform some simple activities listed in the following steps:

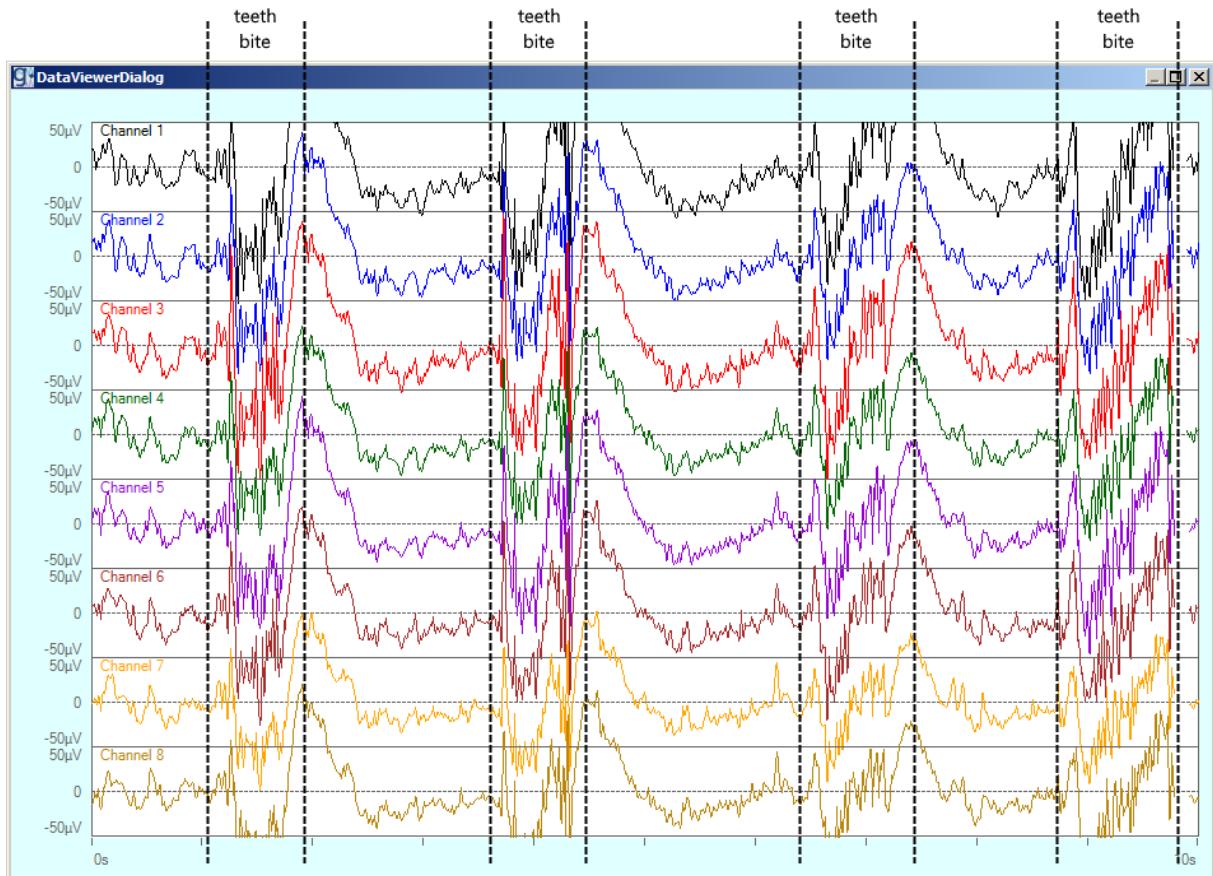
1. The quiescent EEG of a person who is not moving should look like in the following pictures. As you can see, all channels look quite similar and range within  $\pm 50 \mu$ V.



2. Advise the user to blink several times and watch the corresponding EEG signal. For each eye blink, the EEG signal should show a clear deflection, as shown in the picture above.



3. Advise the user to close his or her eyes and relax for several seconds. The EEG signal should show a rhythmic oscillation Error! Reference source not found.reflecting increased alpha activity.



4. Advise the user to clench his or her teeth for a short time. This should temporarily cause strong artifact in the EEG signals, as shown in the picture above.

When you get similar results for those tests, the signal quality should be good.

After finishing inspection, close the **Data Viewer** dialog. The dialog will also be closed when clicking somewhere outside of the dialog.

### 18.8.2. INTEGRATED SIGNAL QUALITY CRITERIA

For each channel, the actual built-in signal quality criteria are evaluated separately and result in an estimation of the signal quality (good or poor). Only the two most recent seconds of the signal will be considered for evaluation. If both criteria indicate good signal quality, the whole channel will show good signal quality. Otherwise, the channel shows poor signal quality.

There are two built-in signal quality criteria that will be checked: the bandpower mean difference (BPMD) and the standard deviation (SD).

- \* The standard deviation criterion: If the standard deviation of the 2 last seconds of the signal of the corresponding channel is within 7 µV and 50 µV, the signal quality is assumed to be Good. If the standard deviation lies outside this range, the signal quality is assumed to be Poor.



- ★ The bandpower mean difference criterion: The signal processing chain of Unicorn Speller applies notch filters that block frequencies around 50 Hz and 60 Hz on the incoming signal from the Unicorn Brain Interface and a bandpass filter with cutoff frequencies of 0.1 Hz and 30 Hz afterwards. First, the mean signal amplitude of the 50 Hz and 60 Hz frequency component of the signal (within a band of  $\pm 2$  Hz) after the notch filtering is calculated. Then the mean signal amplitude within the frequency range of 0.1 – 30 Hz after the bandpass filtering is calculated. If the difference between the calculated mean value after bandpass filtering and the calculated mean value after notch filtering (i.e. bandpass minus notch) is greater than 0.1  $\mu$ V, the signal quality is said to be Good. Otherwise, the signal quality is said to be Poor.

## 18.9. CALIBRATE UNICORN SPELLER

Unicorn Speller must be calibrated before its first use. During calibration, Unicorn Speller learns the difference between the user's typical EEG signals after a calibration item flashes and after all other (non-calibration) items flash. In particular, calibration creates a decision boundary that best separates the recorded signals for calibration items from all other non-calibration items. For a new signal (that comes after calibration), Unicorn Speller then decides whether this signal better matches the recorded signals for a calibration or a non-calibration item from the previous calibration session.

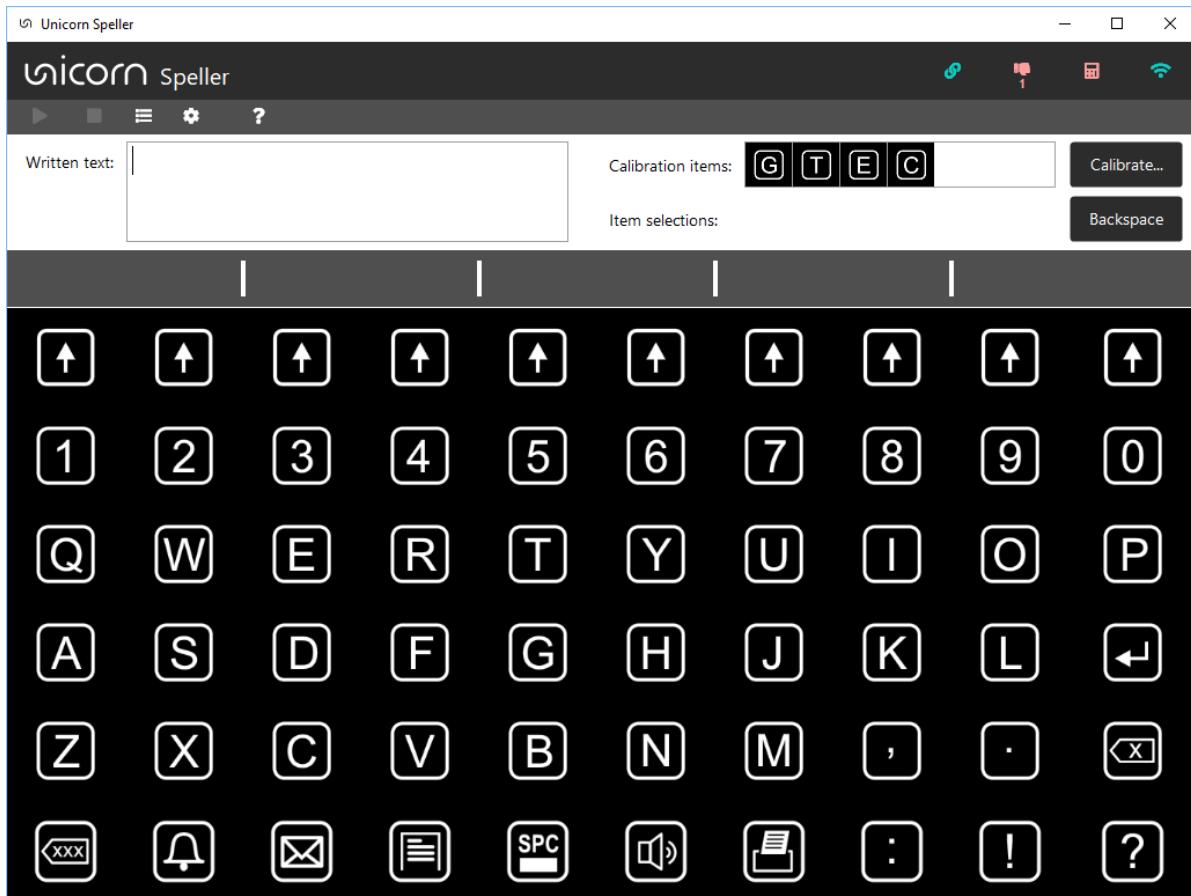
Run through the following steps to successfully calibrate Unicorn Speller:



### NOTE:

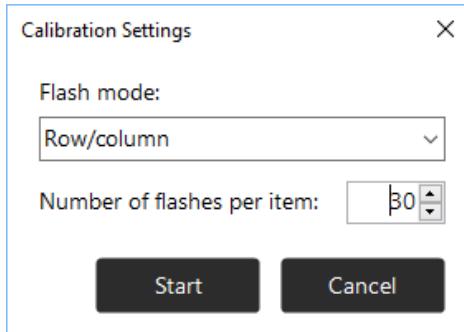
Read through all the steps before actually executing them.

1. Ensure that the Unicorn Hybrid EEG Electrodes are mounted correctly, the Unicorn Brain Interface runs, Unicorn Speller is connected to the Unicorn Brain Interface and receives data from it (status indicator must say that data acquisition is running) and the signals are of a good quality. Also ensure that Unicorn Speller is in standby mode, i.e. flashing must be stopped (■).
2. Add a sequence of items to the calibration items panel by double-clicking the desired items on the board. This is the sequence that the user must sequentially count during calibration. For a reasonable calibration, at least 4 calibration items are recommended. In the figure below, the sequence GTEC has been selected as a calibration item sequence.



3. Adjust the settings for timing (see chapter 18.14.1) to your needs. Note that these settings are strongly user-dependent, so you have to determine the best setting (where the resulting speed and accuracy deliver best results) empirically.
4. At the beginning, recommended settings are:
  - \* for single character mode: **Flash Time** = 60ms, **Dark Time** = 10ms
  - \* for row/column mode: **Flash Time** = 75ms, **Dark Time** = 100ms
  - \* for randomized patterns mode: **Flash Time** = 75ms, **Dark Time** = 100ms
5. Ensure that the user is sitting or lying completely relaxed in front of the screen. The screen should be centered in front of the user, and try to minimize glare or reflections. Recommended distances are:
  - \* approximately 60cm for a 21" screen
  - \* approximately 50cm for a 19" screen
  - \* approximately 30cm for a 10" screen

Click the **Calibrate** button. The **Calibration Settings** dialog opens. Select the flash mode for which Unicorn Speller should be calibrated and the number of flashes that are carried out sequentially for each calibration item in the calibration items panel. At least 30 flashes are recommended.



6. When you are ready, click the **Start** button.

7.

**NOTE:**

It is extremely important that the user has a completely relaxed posture while sitting or lying to obtain good signal quality, and hence good calibration. Any kind of muscle activity can cause artifacts in the EEG signal and substantially impair data quality and effective calibration.

Now, ask the user to focus on the current calibration item and get ready to count it. The current calibration item for the following flashing session will be highlighted for the time specified in the **Wait time** field in the **Flashing Settings** dialog (see section 18.12). Then the flashing starts and the previously specified number of flashes will be executed for the current calibration item.

**NOTE:**

Each time the current calibration item flashes on the board, the user silently count it to enhance the P300 complex in the brain signals. This is a common task to help users pay attention to each flash. The counting isn't necessary. Hence, it doesn't matter if the user loses the count or miscounts. You can even try it without counting.

After the specified number of flashes has been completed for the current calibration item, the next calibration item will be highlighted for the **Wait time** and the procedure repeats for that calibration item.

The procedure will be repeated until all calibration items have been completed this way.

Calibration can be cancelled any time by clicking **■** in the menu bar. In this case, the current flash cycle will be completed and the process immediately proceeds to step 8. Please note that only the executed flashes will be considered for calibration.

8. After the last flash cycle for the last calibration item in the calibration items panel has been completed or calibration has been stopped by the user, the **Save Calibration Set** dialog will appear.

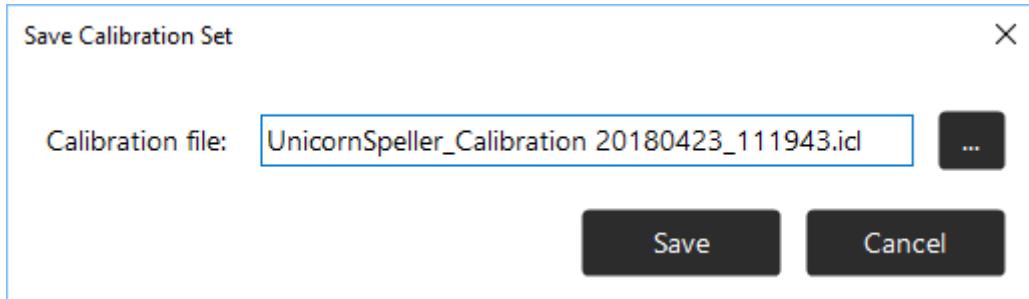
Specify filename (and optionally a path) under which the calibration file should be saved to on the PC. The default filename includes a timestamp of when the calibration was performed. You can use the **...** button



to browse through your file system. If just the filename without a path is specified, the calibration will be stored to the default calibration folder of Unicorn Speller (see section 18.2).

An existing calibration that has been used until now will not be used anymore, but the calibration file will be kept if you want to use it again later (see section 18.13).

To cancel calibration and discard the recorded data, click the **Cancel** button. In this case, the recorded calibration data will be deleted.



In the Save Calibration Set dialog, enter the filename (extension \*.ic1) under which to save the calibration. The default filename includes a timestamp in the format yyyyMMdd\_hhmmss, where yyyy represents the 4-digit year, MM the 2-digit month, dd the 2-digit day, hh the 2-digit hours, mm the 2-digit minutes and ss the 2-digit seconds of the performed calibration.

9. After a short calculation time, a confirmation message will be presented that Unicorn Speller has been calibrated successfully. If you get an error message instead, something went wrong during calibration (maybe the created calibration file couldn't be saved to the specified location or an internal calculation error occurred). Please calibrate again.
10. You can clear the calibration items panel now by removing each item with the **Backspace** button. Your calibration is now ready to be used.

## 18.10. WRITE WITH UNICORN SPELLER

To use Unicorn Speller with an existing calibration, run through the following steps:

1. Ensure that the Unicorn Hybrid EEG Electrodes are mounted correctly, the Unicorn Brain Interface runs, Unicorn Speller is connected to the Unicorn Brain Interface and receives data from it (status indicator must say that data acquisition is running) and the signals are high quality. Also, ensure that Unicorn Speller is in standby mode, i.e., flashing must be stopped (menu ).
2. Ensure that Unicorn Speller is calibrated (see section 18.9).
3. If your chosen selection method requires parameters that are not configured yet, please configure them first (see section 18.13 and following subsections).
4. Now you have the choice to run Unicorn Speller in free-spelling mode (which is the usual case) or in copy-spelling mode (e.g. for testing).
  - a. To run Unicorn Speller in copy-spelling mode, select the desired calibration items first by adding them to the calibration items panel by double-clicking them in the board.



- b. To run Unicorn Speller in free-spelling mode, delete all items from the calibration items panel using the **Backspace** button. The user has to select the current calibration item by himself.



#### NOTE:

Unlike calibration, the calibration items have no influence on the mode of operation now. Unicorn Speller will select the items as it does in free-spelling mode. The only difference is that Unicorn Speller highlights the current calibration item before the flashing for that item start and it stops after the same number of items has been selected. The **Item selections** panel displays all selected items.

5. Ensure that the user is sitting or lying completely relaxed in front of the screen. The screen should be centered in front of the user, and try to minimize glare or reflections. Recommended distances are:
  - \* approximately 60cm for a 21" screen
  - \* approximately 50cm for a 19" screen
  - \* approximately 30cm for a 10" screen
6. Select ► from the menu bar to start the flashing. The flashing will not start if data acquisition is not running or Unicorn Speller is not calibrated.



As during calibration, the user is advised again to focus on the current item (whichever the user wants to write in free-spelling mode). In copy-spelling mode, the current item for the following flashing session will be highlighted for the time specified in the **Wait time** field of the **Flashing Settings** dialog (see section 18.12).

Then, the flashing starts. Depending on the chosen selection method (see section 18.13 and following subsections), several flash cycles are executed until Unicorn Speller makes a decision and selects an item (if it is able to).



#### NOTE:

Each time the current calibration item flashes on the board, the user silently count it to enhance the P300 complex in the brain signals. This is a common task to help users pay attention to each flash. The counting isn't necessary. Hence, it doesn't matter if the user loses the count or miscounts. You can even try it without counting.

Each time an item has been selected, it will appear larger on the screen for the duration specified by **Wait time** (see section 18.12). If double selection is necessary for that item and this item is selected the first time, it will be highlighted with the specified color in the board (see sections 18.14.1, 18.14.2 and 18.11) and has to



be selected again directly after this one. If the item is already preselected and selected a second time or if double selection is not necessary, a possibly linked action is executed (see section 18.11) and the selected item is sent over the network, if enabled (see section 18.16). If word prediction is enabled, the completion suggestion panel is updated with new completion suggestions based on the current content in the **Written text** box in addition (see section 18.15).

After the **Wait time** has elapsed, the first flash cycle for the next calibration item begins immediately.

7. To stop the flashing, click the ■ button from the menu bar.

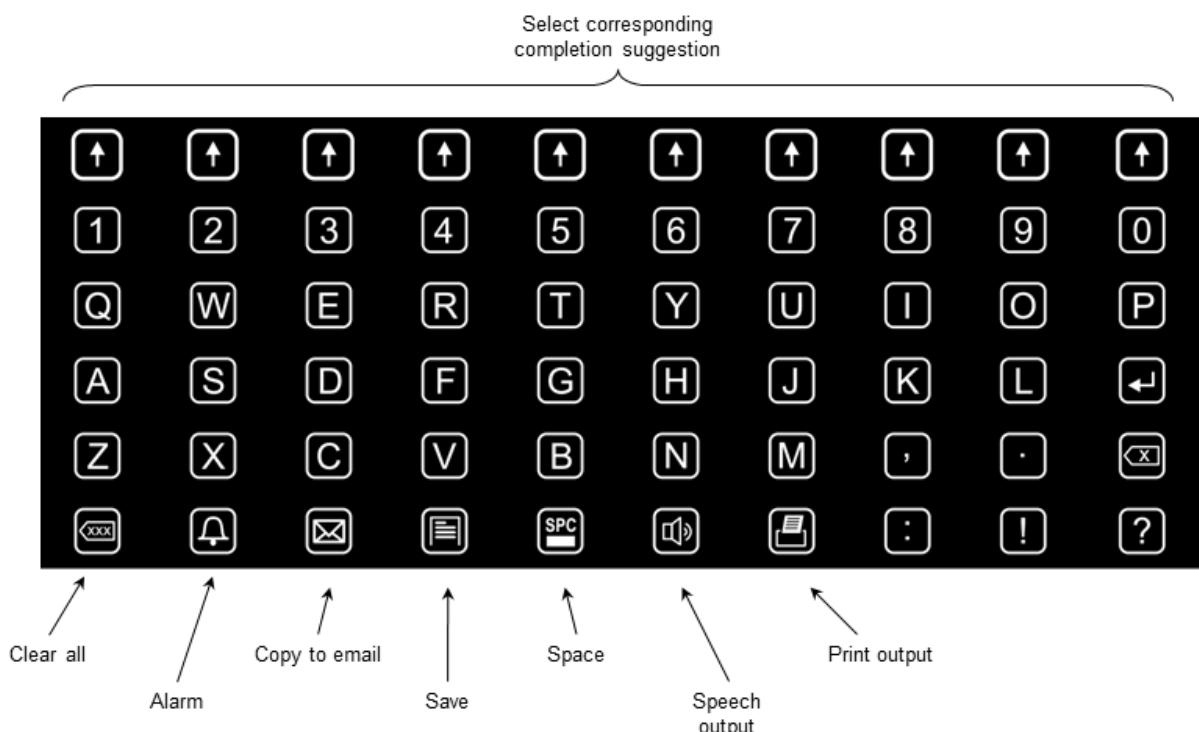


8. To close Unicorn Speller, click the close button at the top-right corner.



## 18.11. ITEMS WITH SPECIALLY LINKED ACTIONS IN THE STANDARDBOARD.IBC AND STANDARDFACEBOARD.IBC FILE

The predefined standard boards (files StandardBoard.ibc and StandardFaceBoard.ibc; see section 18.2) that will automatically be loaded on first startup is presented in the figure below.



The standard board uses **Flash Image** and **Dark Image** for its items instead of **Displayed Text** (see section 18.14.2). Flashing is realized by displaying the item image with inversed color. Items with characters



(0...9, A...Z, enter, space, comma, dot, colon, exclamation mark and question mark) have set their **Output Text** field to the corresponding character such that they write this character to the **Written text** box on selection.

The following items have special functions (for some of the items, double selection is necessary before the action will be executed; see 18.14.2):

Item	Caption	Double selection necessary	Action on selection
	Select completion suggestion	No	<p>Selects the completion suggestion displayed in the corresponding slot of the completion suggestion panel right above the item if word prediction is enabled and completes the currently written word in the <b>Written text</b> box with the selected completion suggestion.</p> <p>If word prediction is disabled or the corresponding slot does not show a completion suggestion, a selection of this item will have no effect.</p> <p>See section 18.11.1 for details on word prediction.</p>
	Backspace	No	<p>Deletes the last written item from the <b>Written text</b> box.</p> <p>If word prediction is enabled and backspace is selected right after a completion suggestion has been selected, the written text will be reverted to its state before the completion suggestion was inserted.</p>
	Clear all	Yes	Clears the whole <b>Written text</b> box.
	Alarm	No	Plays an alarm sound for two seconds.
	Copy to email	Yes	Opens a new email using the configured standard email client (configurable via the operating system)
	Save	Yes	<p>Saves the current content of the <b>Written text</b> box to an ASCII text file. The name of the text file is prespecified and follows the format below:</p> <p>UnicornSpeller_Document yyyyMMdd_hhmmss.txt</p> <p>yyyy are four digits representing the year, MM two digits representing the month, dd two digits representing the day,</p>



Item	Caption	Double selection necessary	Action on selection
------	---------	----------------------------	---------------------

hh two digits for the hour in a 24-hour clock, mm two digits for the minutes and ss two digits for the seconds of the date of saving.

See section 18.11.2 for details on configuring the storage directory.

	Speech output	Yes	Performs text-to-speech output of the whole text in the <b>Written text</b> box with the current settings. See section 18.11.3 for details on configuring the text-to-speech output.
	Print output	Yes	Sends the whole <b>Written text</b> box to the currently set default printer (configurable via the operating system).

### 18.11.1. CONFIGURING ITEM: SELECT COMPLETION SUGGESTION

The **Select completion suggestion** items in the first row are part of the word prediction feature. If word prediction is enabled (i.e. the completion suggestions panel above the board is visible) and one of those items is selected, the current word is completed with the word that is displayed in the corresponding slot of the completion suggestions panel just above the selected item. If that slot does currently not display any word or if word prediction is not enabled and the completion suggestions panel is hidden, selecting the corresponding **Select completion suggestion** item will have no effect.

If the **Backspace** item is selected right after a **Select completion suggestion** item, the completion will be undone. Otherwise, the **Backspace** item deletes the last written item.

See section 18.15 for details on the configuration of the word prediction feature.

### 18.11.2. CONFIGURING ITEM: SAVE

When the **Save** item has been selected twice (as it is necessary because of double selection), the whole **Written text** will be saved to an ASCII text file.

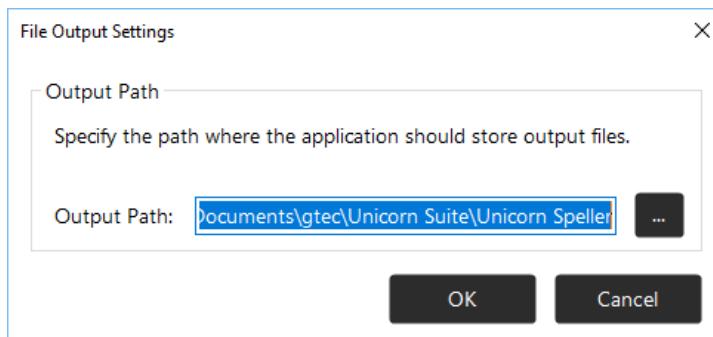
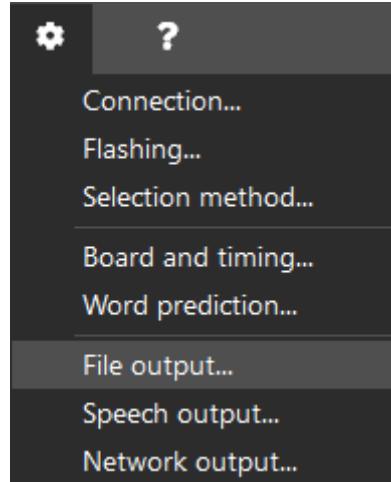
The name of this stored text file is prespecified and of the following format:

UnicornSpeller\_Document yyyyMMdd\_hhmmss.txt

yyyy are four digits representing the year, MM two digits representing the month, dd two digits representing the day, hh two digits for the hour in a 24-hour clock, mm two digits for the minutes and ss two digits for the seconds of the date of saving.



To set or change the directory where the text files will be stored on selecting the **Save** item, select → **File output...** from the menu bar to open the **File Output Settings** dialog.



In the **Output Path** text field, set the directory to store the text files created on selection of the **Save** item. Instead of typing the location, you can click the **...** button to navigate to the directory.



**NOTE:**

Only the directory can be specified, not the filename. The filename will be set by the application as described above, and contains a time stamp of the saving date.

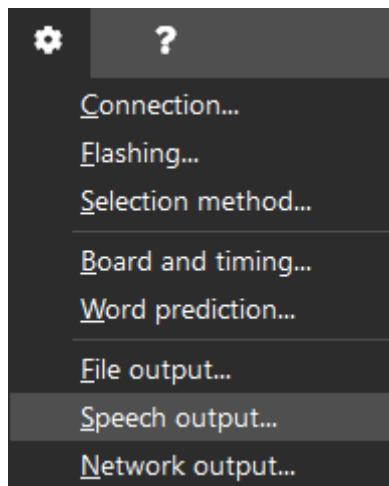
### 18.11.3. CONFIGURING ITEM: SPEECH OUTPUT

When the item **Speech output** has been selected twice (as it is necessary because of double selection) the whole **Written text** will be read aloud by the operating system's text-to-speech engine.

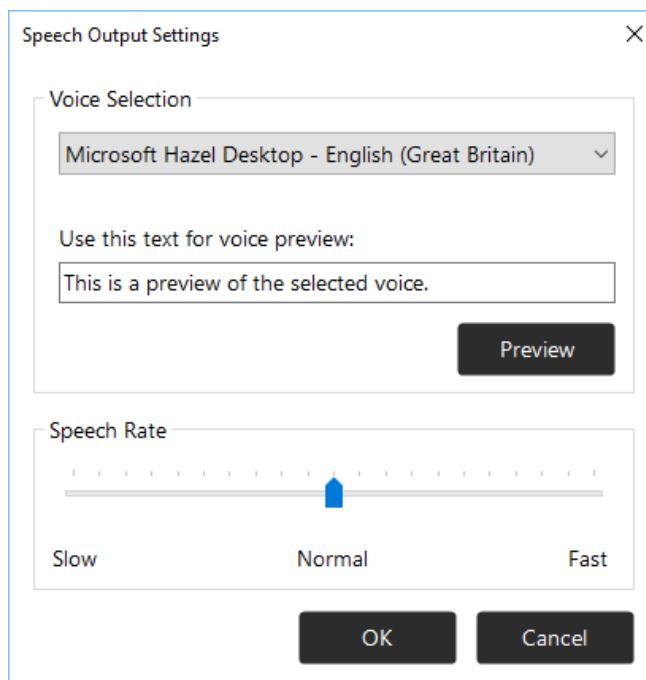
**NOTE:**

For requirements to perform text-to-speech output and installing additional voices and languages, see section 18.1.

To select a specific voice and language and the speed of reading, open the **Speech Output Settings** dialog by selecting → **Speech output...** from the menu bar.



The Speech Output Settings dialog allows selection of the voice to use, adjustment of the speech rate and listening to a preview.



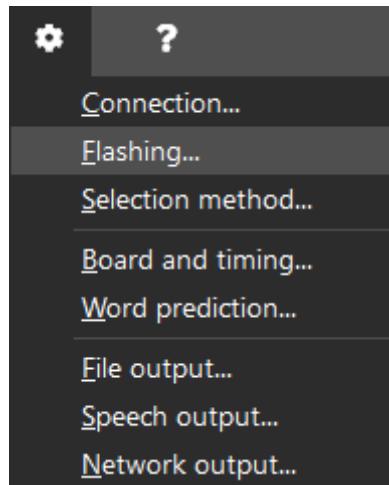


The drop-down list in the **Voice Selection** group lists all currently installed voices for the operating system's text-to-speech engine and lets you select the voice to use. By clicking the **Preview** button, you can listen to a specified preview text read by the selected voice with the current settings. Ensure that your speakers are turned on and the volume is high enough.

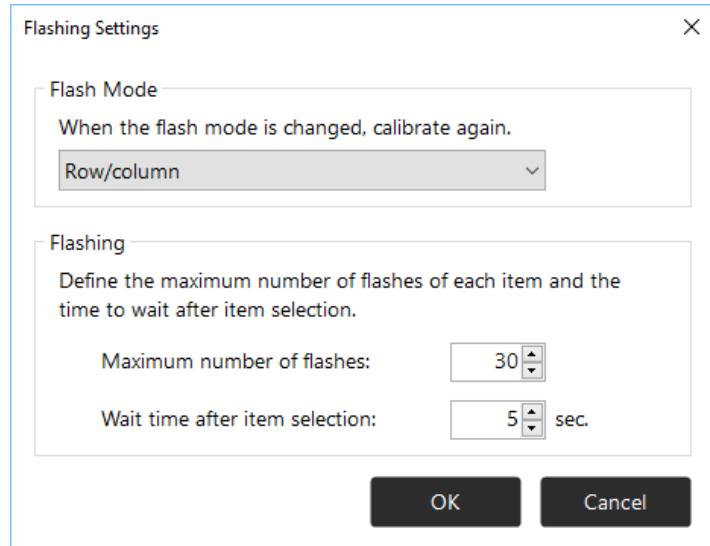
If you don't hear anything in the preview twice, check your speakers and volume settings. If this doesn't help or you get an error message, the voice maybe incompatible with the text-to-speech engine of your operating system. To check compatibility, you can also preview the voice via the control panel of the operating system. See section 18.1 for further details.

## 18.12. SPECIFY THE SETTINGS FOR FLASHING

Open the **Flashing Settings** dialog by selecting → **Flashing...** from the menu bar.



The Flashing Settings dialog allows configuration of the common flashing options.

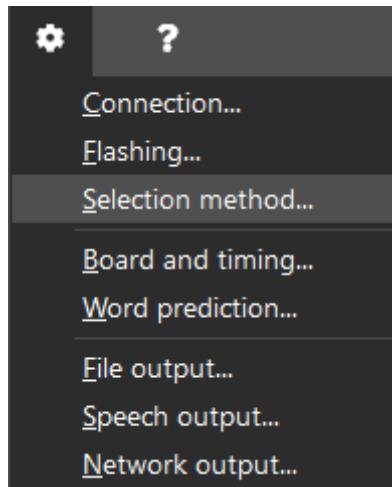


You can configure the following options:

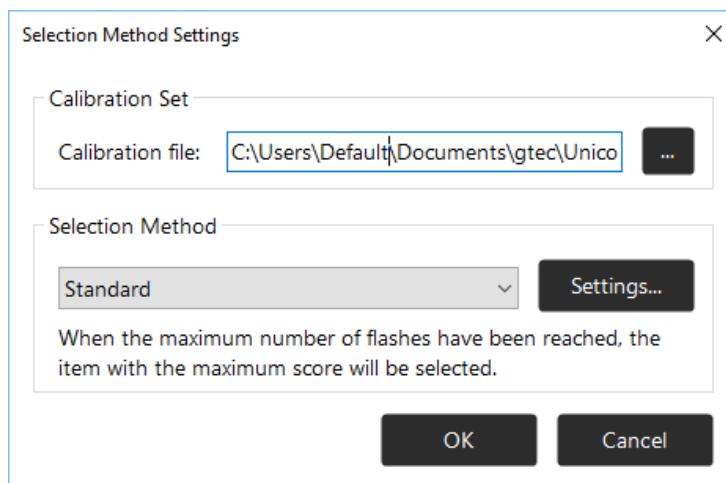
- ✳ **Flash Mode:** Specifies the flash mode for free-spelling or copy-spelling mode.
- ✳ **Maximum number of flashes:** This is the maximum number of the most recent flashes of each item will be considered for item selection. All previous flash cycles older than the specified maximum number of flashes will not be considered for the item selection anymore. This is necessary in **Dynamic** selection modes to quickly recover from long inactive phases where the user doesn't want to select a specific item and doesn't want Unicorn Speller to select anything while flashing.
- ✳ **Wait time between two items:** After an item has been selected, the selected item will be displayed for the time specified by this value. In copy-spelling mode or during calibration, the next calibration item will be highlighted for the same amount of time afterwards. Then, the flashing session for this calibration item begins.

## 18.13. SPECIFY THE SETTINGS FOR ITEM SELECTION

Open the **Selection Method Settings** dialog by selecting → **Selection Method...** from the menu bar. An error message is displayed when Unicorn Speller is not calibrated. In that case, confirm the error message first to show the dialog.



The Selection Method Settings dialog allows to configure item selection.



The currently used calibration file (usually the one from the most recent calibration run) is selected by default.

The selection method to use can be altered after calibration and can be configured by clicking the **Settings** button.



**NOTE:**

During flashing, the user should be in a relaxed position to obtain good results. Muscle activity or eye blinking could cause erroneous selections due to heavy artifacts in the EEG signals.

### 18.13.1. SELECTION METHOD: STANDARD

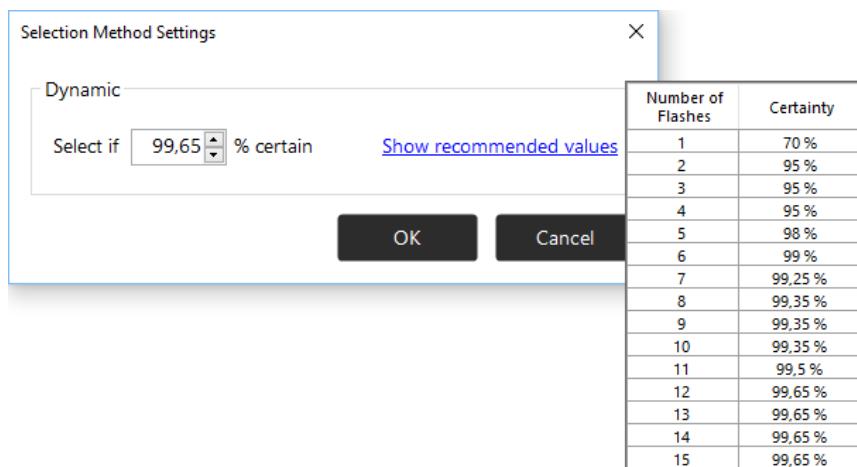
If the **Standard selection method** is selected, an item will only be selected when the **Maximum number of flashes** (see section 18.12) for the current item has been reached, even if the collected data is



not very meaningful (e.g. when the user didn't focus on a specific item). Thus, the number of executed flashes per item is always the same and equals the specified **Maximum number of flashes**, even if fewer flashes might have been sufficient or more flashes would be necessary to reliably select the user's target item.

### 18.13.2. SELECTION METHOD: DYNAMIC

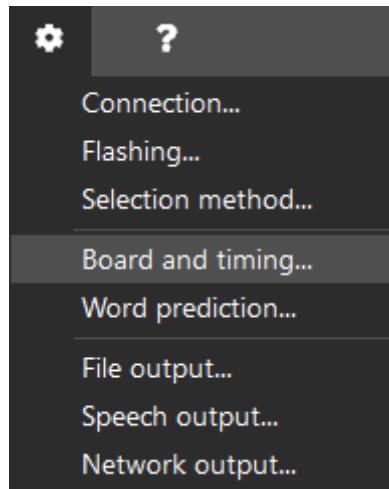
The **Dynamic** selection method attempts to select an item before the **Maximum number of flashes** (see section 18.12) has been reached. In this case, the threshold of how different an item must be to be selected must be specified in percent. This **Certainty** must be configured before the selection method is used. Recommended values of this **Certainty** can be displayed by clicking the **Show recommended values** link in the settings dialog, as shown in the following figureError! Reference source not found..



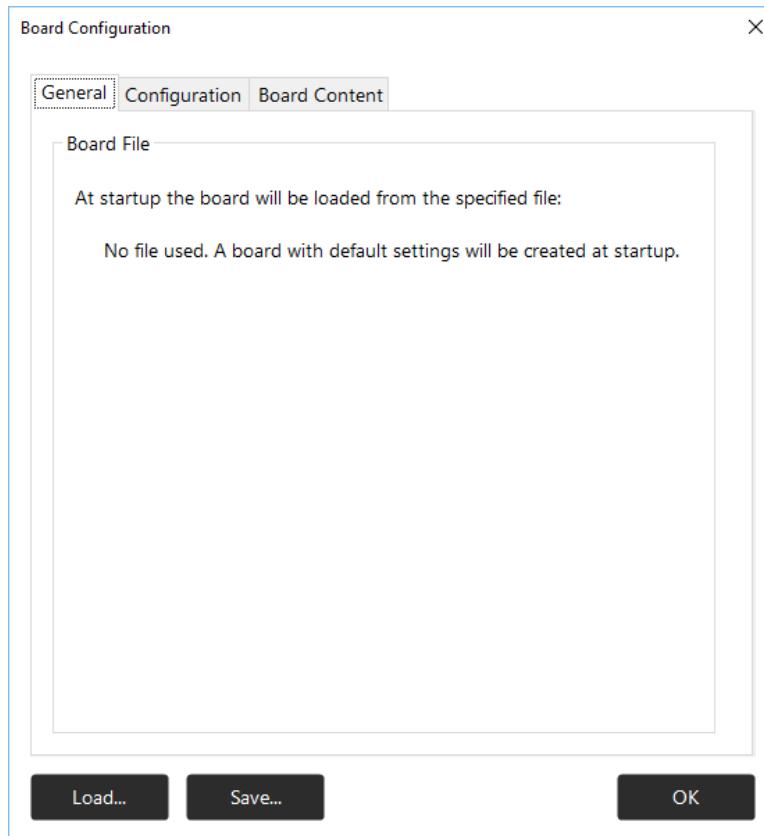
The settings dialog for the Dynamic selection method. Clicking the Show recommended values link will display the recommended values for the Certainty depending on the selected Number of Flashes.

## 18.14. CUSTOMIZING THE BOARD AND CONFIGURING THE TIMING OF FLASHING

Open the **Board Configuration** dialog by selecting → **Board and timing...** from the menu bar to customize the board and configure the timing of flashing.



The General tab of the Board Configuration dialog shows the file that will be loaded at startup. In this example, no file has been specified yet.



The **General** tab of the **Board Configuration** dialog shows you the file that will be reloaded on every startup of Unicorn Speller.

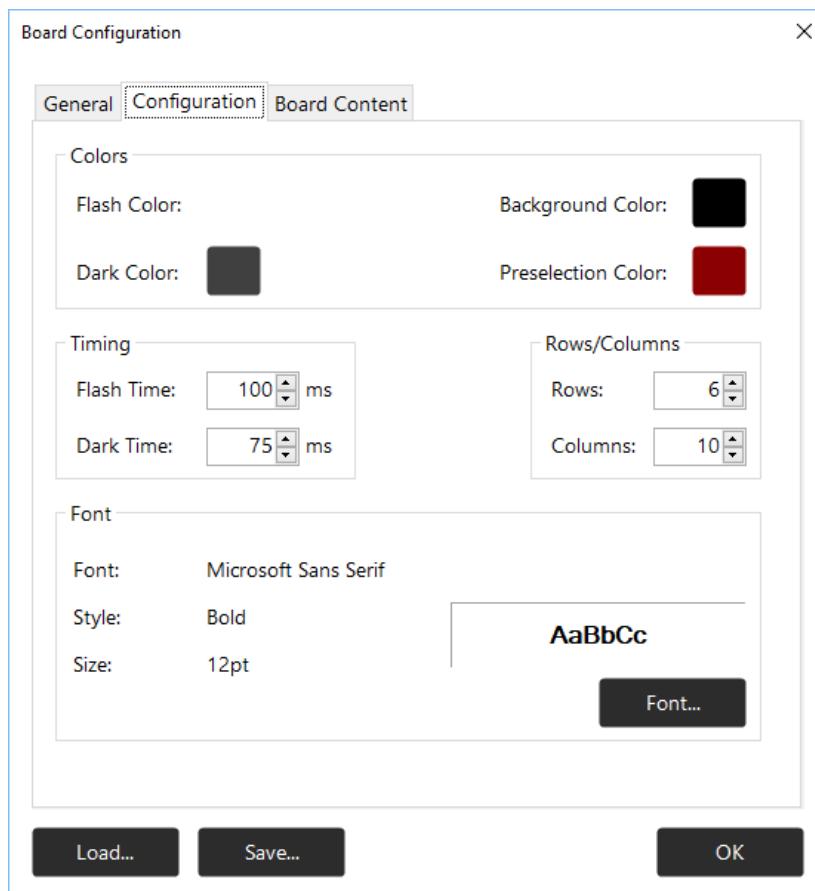
You can save a specific board configuration to a file by clicking the **Save...** button and reload it from this file at each startup by clicking **Load...** and selecting a board configuration file (files with extension \* .ibc).



If no file is specified or the previously specified file couldn't be found or loaded, the predefined file StandardFaceBoard.ibc will be loaded (see section 18.2).

### 18.14.1. OVERALL CONFIGURATION OF THE BOARD AND TIMING

In the **Configuration** tab, the overall settings of the board can be adjusted.



The following colors can be specified and changed by clicking on the colored rectangle next to the corresponding label:

- \* **Flash Color:** When an item of the board has text specified in the **Displayed Text** field (see section 18.14.2), this text will represent the item in the board. When this item gets flashed now, the color of this text in the board will be set to the **Flash Color**. When using **Flash Image** and **Dark Image** only (without **Displayed Text**; see section 18.14.2), this color has no effect.
- \* **Dark Color:** This is the color that will be assigned to items with a set **Displayed Text** field (see section 18.14.2) in the board when they are not flashing currently. When using **Flash Image** and **Dark Image** only (without **Displayed Text**; see section 18.14.2), this color has no effect.
- \* **Background Color:** Sets the background color of the board.
- \* **Preselection Color:** Items that have checked the **Double selection necessary** checkbox must be selected twice before they are actually selected (see section 18.14.2). When such an item is selected the first time, its background will be colored with the **Preselection Color**. If the item is



selected once again, it is actually selected. Otherwise, its background color is reset to the **Background Color** of the board.

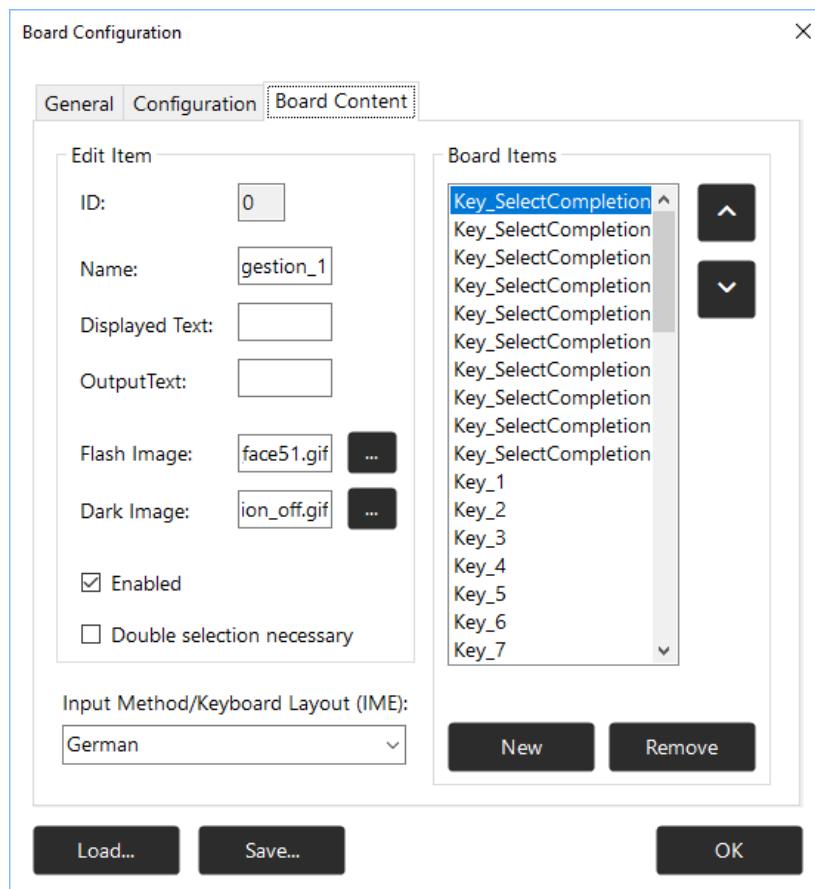
The timing of the flashing can be adjusted in the **Flash Time** and **Dark Time** fields (please see section 0).

The number of displayed rows and columns can be set by the **Rows** and **Columns** field. Please ensure that there are enough cells available to display all contained items.

With the **Font...** button, the font of the **Displayed Text** can be set, if specified.

### 18.14.2. EDITING THE CONTENT OF THE BOARD

The **Board Content** tab of the **Board Configuration** dialog lets you specify the content (items) of the board.



In the **Board Items** list box, the current content/items of the board are listed (the specified **Name** of each item is displayed). The items are arranged from right to left and from top to bottom on the board. The first item in the list represents the top-left item on the board. Items below in the list are arranged on the right of the previous item in the same row. The item will be placed in a new row (below the previous one and on the most left column) if the current row is filled.



To change the position of an item, left click it to select it and use the **▲** and **▼** buttons to move it. To add a new item to the board, click the **New** button. To remove an item from the board, select the desired item and click the **Remove** button.



**NOTE:**

When the file `StandardFaceBoard.ibc` is used, there are some items that cannot be removed because they are linked with internal actions.

Create an empty board by loading the file `EmptyBoard.ibc` to add your own items (see section 18.14 for details). Don't forget to save your changes afterwards to a new file (do NOT save changes to the `EmptyBoard.ibc` file; your changes will get lost on the next startup of Unicorn Speller).

For each item, you can specify the following parameters:

- \* **Name:** The name of the item. This is the name that will be displayed in the **Board Items** list box but not in the board.
- \* **Displayed Text:** You can either specify text that will be displayed in the board to represent the item or a picture (by using **Flash Image** and **Dark Image**). If you choose text, it will be colored with the specified **Flash Color** and **Dark Color** during flashing (see section 18.14.1).
- \* **Output Text:** This is the text that will be written to the **Written text** box if the item has been selected. Note that the **Displayed Text** field just displays the specified text in the board but does not write the **Displayed Text** to the **Written text** box on selection.
- \* **Flash Image/Dark Image:** When not using **Displayed Text**, the item can be represented by images. The **Flash Image** will be displayed when the item is currently flashing. The **Dark Image** will be displayed when the item is currently not flashing. The path to these two images can be set. When creating these images, please set a transparent background! The **Flash Color**, **Dark Color** and **Font** properties will have no effect in this case.
- \* **Enabled:** If checked, the item will be displayed on the board. If unchecked, the item will not be displayed and cannot be selected in the board, but still exists.
- \* **Double selection necessary:** If checked, an item must be selected twice before it is actually selected. When such an item is selected the first time, its background will be colored with the **Preselection Color** (see section 18.14.1). If the item is selected once again, it is actually selected. Otherwise, its background color is reset to the **Background Color** of the board. This provides a chance to correct a mistake before it is executed. For example, deleting the entire text message should not happen by accident. When this item has to be selected twice, mistakes are much less likely.

Character sets of e.g. Asian languages and keyboard layouts must be installed before you can use them. You can add additional keyboard layouts by selecting **Language** from the operating system's **Control Panel**. By clicking the button **Add a language**, you can add the desired language and keyboard layout.



#### NOTE:

You probably have to install the desired keyboard layout before using the installation resources of your operating system.

To use the desired keyboard layout in Unicorn Speller, select one of the recently installed layouts listed in the **Input Method/Keyboard Layout (IME)** drop-down list.

## 18.15. WORD PREDICTION

Word prediction might speed up writing by predicting possible words that the user plans to spell or by completing the currently written text. Selecting one of those completion suggestions requires only one selection rather than a selection for each missing character.

Word prediction requires two things: an Unicorn Speller dictionary file for the desired language (and context, perhaps), and a board that contains the **Select completion suggestion** items like the two standard boards that are installed together with Unicorn Speller (see section 18.11, especially subsection 18.11.1).

#### NOTE:

Unicorn Speller comes with a basic American English dictionary by default. Custom dictionaries for specific contexts or other languages can be created with the Unicorn Speller Dictionary Builder application described in section 18.15.3.

### 18.15.1. USING WORD PREDICTION

If word prediction is enabled, the completion suggestions panel is shown below. The completion suggestion panel is hidden if word prediction is disabled. Word prediction is disabled automatically if the current board does not contain the special **Select completion suggestion** items (the items in the first row showing an up arrow; see also section 18.11.1) or if no valid dictionary file is selected.



Whenever an item is selected that writes something the **Written text** box, the completion suggestion panel below is updated and displays possible candidate words for completion of the current text. A selection of one of the **Select completion suggestion** items completes the text in the **Written text** box with the



candidate word (or completion suggestion) that is displayed right above the selected up-arrow. Multiple **Select completion suggestion** items can be dedicated to the same completion suggestion. Depending on the board's number of columns, a maximum of four or five completion suggestions are shown.

If a slot in the completion suggestion panel does not contain anything, a selection of the corresponding **Select completion suggestion** item will have no effect.

The **Backspace** key usually undoes the previous item selection, i.e. deletes the last character written to the **Written text** box by selection. If the **Backspace** key is selected right after a completion suggestion was selected, all of the completed text is removed.



**NOTE:**

The default dictionary is not case sensitive, since the board does not support case sensitivity. Numbers and punctuation items will be treated as word separators (like whitespaces) or sentence separators.

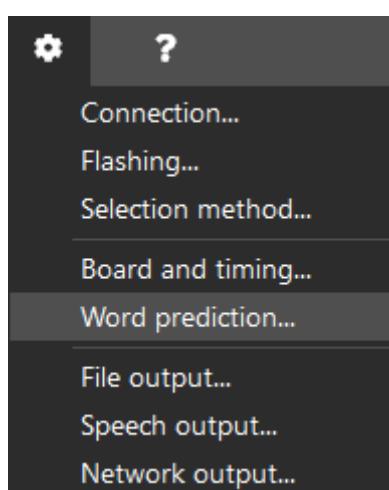


**NOTE:**

The completion suggestions panel is not updated when text is entered into the **Written text** box manually by keyboard.

## 18.15.2. CONFIGURING WORD PREDICTION

Select → **Word prediction...** from the menu bar to open the Word Prediction Settings dialog.



The Word Prediction Settings dialog allows configuring word prediction.



Word Prediction Settings X

**Control**

With learning enabled, the word counts in the dictionary will be updated with each word in the output text field completed by classification. If the default dictionary is used, it can be restored to the installation's default.

Enable Prediction:

Learn:

**Dictionary**

Select the dictionary file to use for word prediction and adjust the corresponding culture setting accordingly. The application must have write access to the directory of the dictionary file.

Dictionary File:  ...

Culture:

**Recovery**

Restoring the default installation dictionary creates a new copy of the originally installed dictionary and sets it as the new dictionary file. Existing dictionary files will not be overwritten.

Restore...

OK Cancel

Word prediction can be enabled or disabled with the **Enable Prediction** checkbox. If word prediction is disabled, the completion suggestions panel is hidden. Word prediction is disabled automatically if the current board does not contain any **Select completion suggestion** items.

If **Learn** is enabled, the word counts in the currently used dictionary get updated with each complete or completed word written in the **Written text** box. It might require several repetitions of learned word sequences until they take any effect in the presented completion suggestions.



#### ATTENTION:

Enabled learning modifies the dictionary file. Only if a pre-installed dictionary file was used and modified, it can be restored to its original state with the **Restore...** button (see below). Custom dictionaries that were not installed with Unicorn Speller cannot be restored to their original state automatically.

The path to the **Dictionary File** to use must be specified in the appropriate text field. Word prediction is only available if the specified path points to a valid Unicorn Speller dictionary file. Unicorn Speller dictionary files have the \*.dic extension.



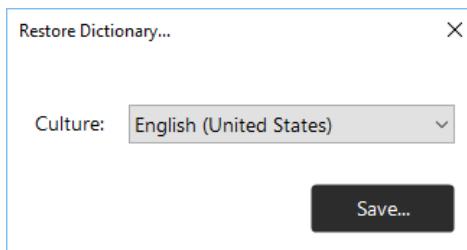
The dictionary file does not contain information about its contained culture or language. Thus, the correct culture of the selected dictionary file must be specified in the **Culture** setting.



**NOTE:**

Pre-installed Unicorn Speller dictionary files contain the ISO language code in the filename. Each language code is made up of a neutral culture code (in lower case) indicating the language of the culture and an optional specific culture code (in upper case) indicating the country represented by the culture, separated by a dash. For example, the language code for American English would be "en-US", while British English would be represented by "en-GB".

Pre-installed dictionary files that were modified by enabled learning (or accidentally deleted) can be restored using the **Restore...** button. The dialog requires you to select the language/culture for which the dictionary file should be restored. Only cultures for which a pre-installed dictionary file exists are listed. Click the **Save...** button to restore the dictionary under a custom filename. Dictionary files should be placed in the designated subfolder of the Unicorn Speller user directory (see section 18.2).



**ATTENTION:**

Overwriting or deleting a modified pre-installed dictionary file results in loss of the modifications like new words or word combinations learned during usage (the original dictionary file can still be restored with the **Restore...** button).

Overwriting or deleting a custom dictionary file results in complete loss of the dictionary. It's the user's responsibility to create backups of modified or custom dictionary files.

### 18.15.3. CREATING CUSTOM DICTIONARIES WITH THE UNICORN SPELLER DICTIONARY BUILDER

Custom Unicorn Speller dictionaries can be created by the user for different languages (and/or different contexts) with the Unicorn Speller Dictionary Builder application that is installed together with Unicorn Speller. It takes plain text files as input and builds a dictionary database based on the occurrences of contained word combinations. Thus, we recommend that you select the text files used for input carefully.



They should cover a wide range of words and word combinations, but should also be tailored to the context in which the dictionary will be used.



**NOTE:**

The better the training text files match the context in which the dictionary will be used, the better the prediction's hit ratio will be. For example, texts from the 19th century would not provide good predictions if the user wants to write sentences in contemporary everyday language, just as a broad spectrum of scripts in everyday language might not be sufficient if communication targets a specific discipline.

Start the Unicorn Speller Dictionary Builder application by clicking the corresponding icon in the start menu. The application's main window will show up as inError! Reference source not found..

---

The **Add...** and **Remove** buttons will let you choose plain text files from connected storage Unicorn Brain Interfaces from which to build up the dictionary's content.

It is recommended to give a hint about the contained **Culture** for proper conversion to lower-case letters when parsing the text files, especially if the language is not based on the basic Latin alphabet. The selected culture information is not stored in the resulting dictionary file.

As long as case sensitivity should not be concerned in prediction, the Lower-case mode checkbox should remain ticked. In lower-case mode, all words in the input text files are converted to lower case before they are added to the dictionary, just as Unicorn Speller converts input words to lower case before it queries the dictionary for predictions. The default boards in Unicorn Speller are not case sensitive, so lower-case mode should be used.

Click the **Train...** button to create the Unicorn Speller dictionary file from the selected training text files. The training process might take a while, depending on the amount of text to parse. Afterwards, the application prompts a dialog to select the location and filename where the resulting dictionary \*.dic file



should be saved to. Unicorn Speller dictionaries should be stored in the designated subfolder of the Unicorn Speller user directory (see section 18.2).



**NOTE:**

It is recommended to include the language or culture and a potentially specialized context in the filename to be able to identify the dictionary's content afterwards.

## 18.16. CONFIGURING NETWORK OUTPUT

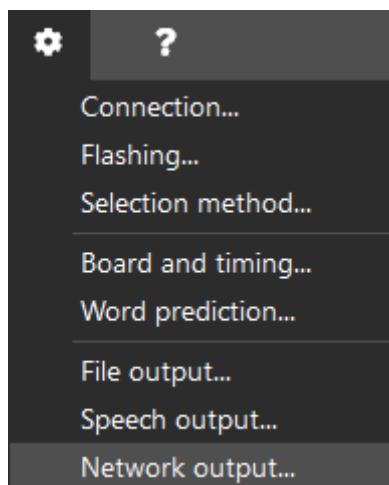
Unicorn Speller can also send selected items over the network. This feature can be useful when other applications should react to a selection and execute actions corresponding to selected items.



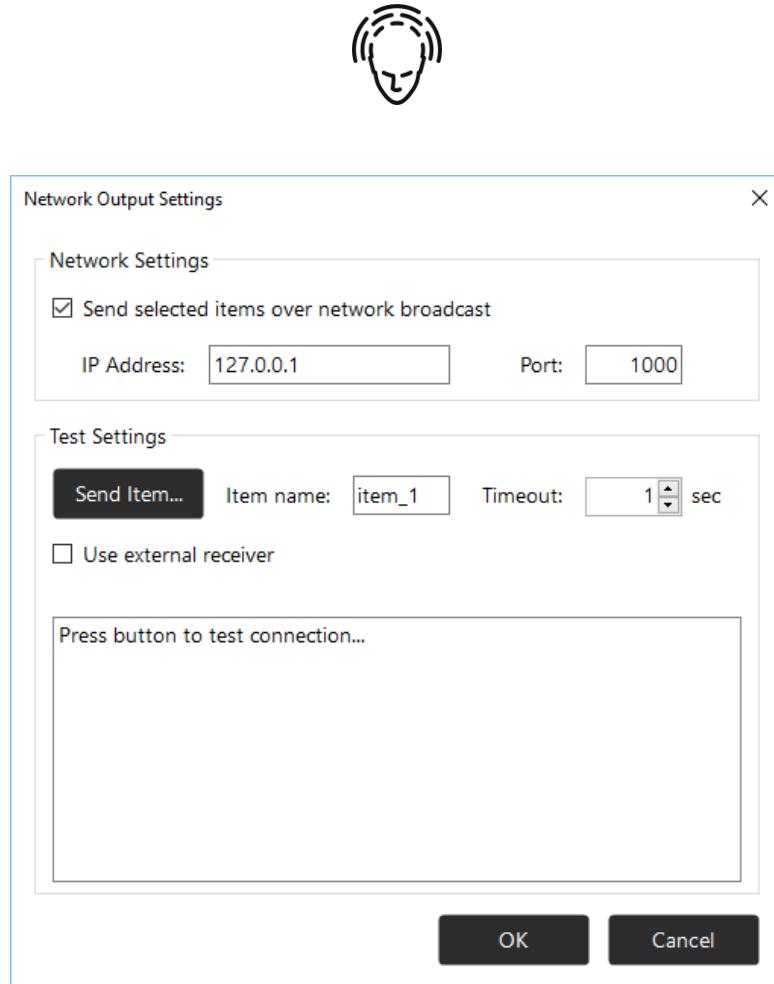
**NOTE:**

Items that have their **Double selection necessary** checkbox checked will only be sent if they are selected twice successively. See section 18.14.2 for details and section 18.11 for items that require double selection.

To activate and configure the network output, open the **Network Output Settings** dialog by selecting **⚙️ → Network output...** from the menu bar.



The Network Output Settings dialog allows enabling and configuring network output of Unicorn Speller.



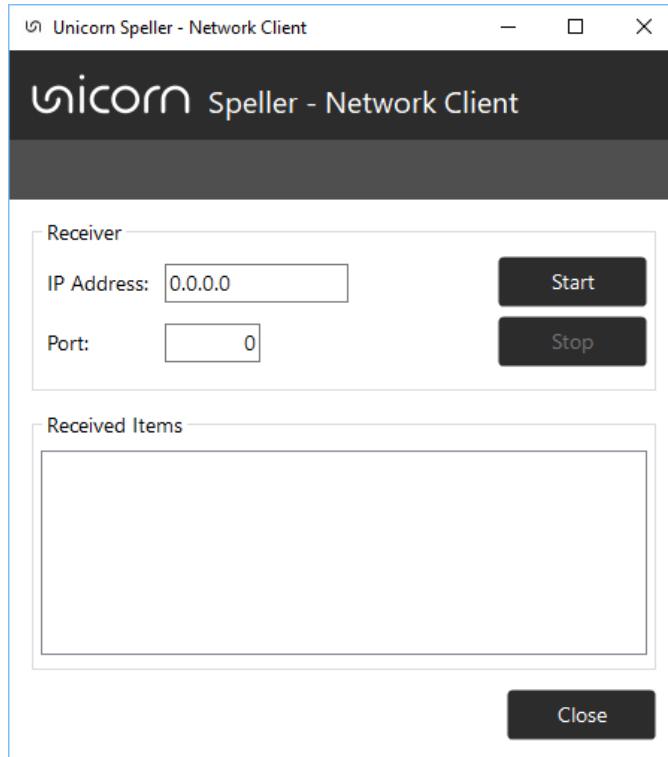
For activating network output, check the checkbox **Send selected items over network broadcast**. Unchecking the checkbox disables network output.

Specify a destination IP address and the port to which Unicorn Speller should send the output. This can also be a broadcast address. Ensure that the firewall on the PC where Unicorn Speller is running permits outgoing connections from Unicorn Speller to the specified destination address. On the PC that runs the receiving application, ensure that this application is permitted to receive incoming connections from Unicorn Speller on the specified port.

To test if the connection is working, you can send a specially generated test item from Unicorn Speller to the specified destination address, pretending this item has been selected previously by the application. The **Name** of the item (see section 18.14.2) can be entered in the **Item name** textbox (in this example it was named `item_1`).

Execute the following steps to test the connection:

1. You should have one PC where Unicorn Speller is running (in the following referred to as `sending PC`) and another PC where your application is used to receive data from the sending PC (in the following referred to as `receiving PC`). Furthermore, the two PCs must be connected by a common network. For the following example, we assume that the receiving PC has the IP address `192.168.100.162` and it should receive data from the sending PC on port `10`.
2. Install Unicorn Speller on the receiving PC. (Unicorn Speller itself is not needed, but the **Network Client** that comes with Unicorn Speller will be used for testing the connection).
3. On the receiving PC, run the **Network Client** Error! Reference source not found..



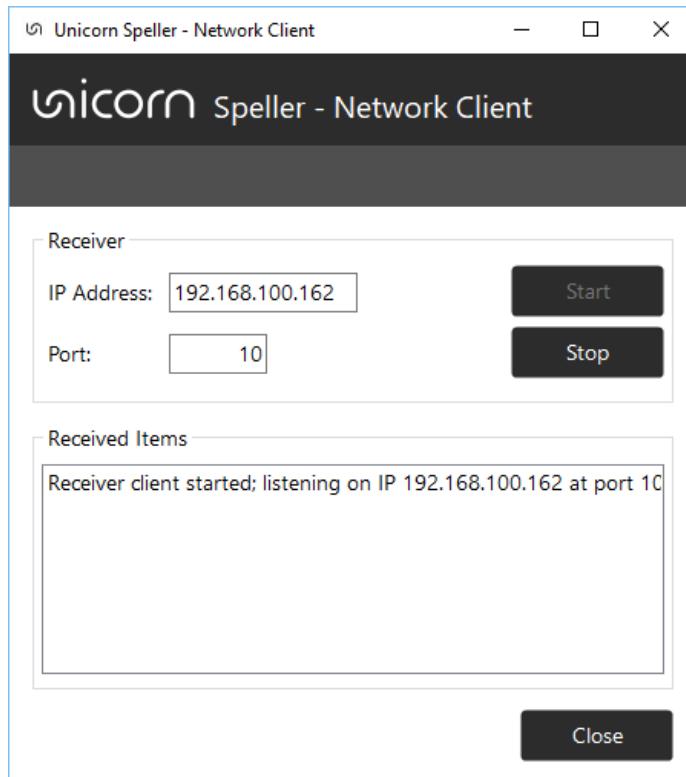
4. In the **Network Client** window on the receiving PC, enter the IP address of the receiving PC and the port where you want to receive the network output from. In the example, we assumed this IP address is 192.168.100.162 and port 10. Then, click the **Start** button.



#### ATTENTION:

Ensure that the firewall on the receiving PC allows incoming connections on the specified port from Unicorn Speller on the sending PC (the sending PC's IP address might be displayed).

The **Network Client** starts listening on the specified network port now and may look like the following figureError! Reference source not found.:

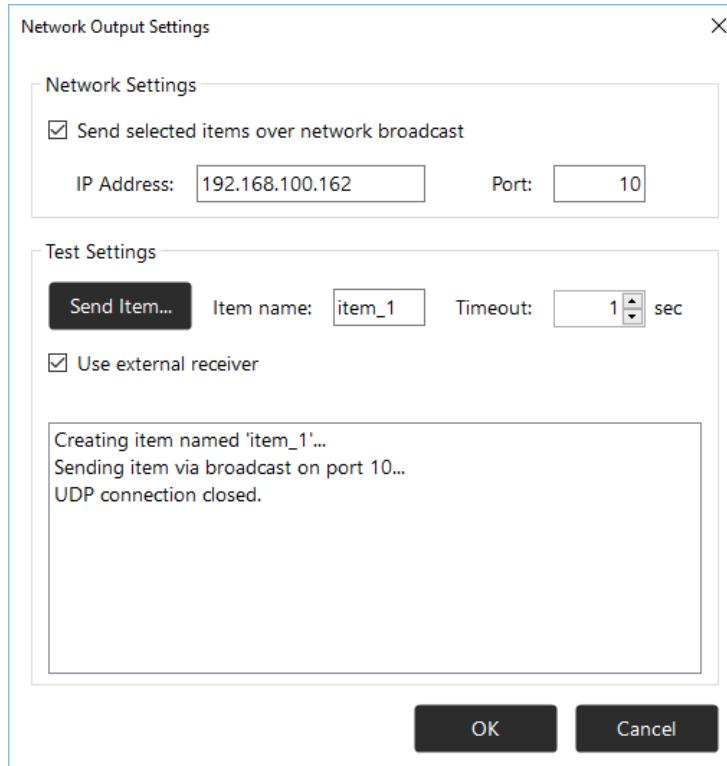


5. In the **Network Output Settings** dialog of Unicorn Speller on the sending PC, enter the IP address of the receiving PC and the ports to which the data should be sent from the sending PC and received from the receiving PC (if disabled, check the **Send selected items over network broadcast** checkbox first). The receiving PC must be listening to the same port, and the IP address is usually the same too.
6. Check the checkbox **Use external receiver** in the Network Output Settings dialog of Unicorn Speller on the sending PC, set the **Timeout** to 1 second and click the **Send Item...** button. A test item with the specified name will be generated and sent to the specified destination address. The dialog might look like in the following figureError! Reference source not found..

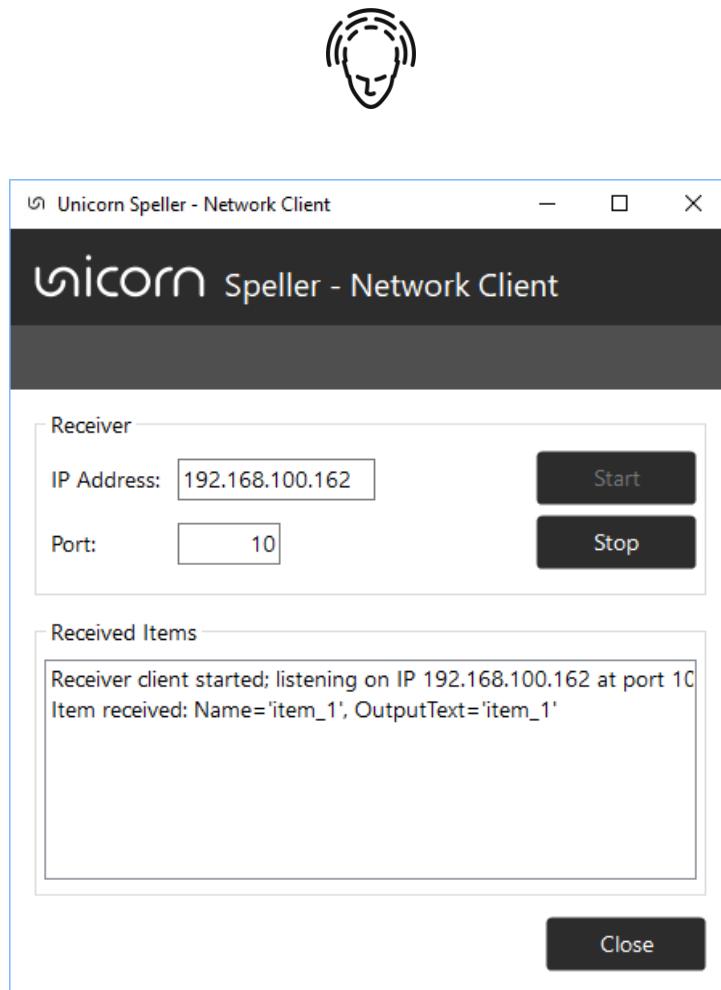


#### ATTENTION:

Ensure that the firewall on the sending PC allows outgoing connections from Unicorn Speller to the specified destination IP address and port.



7. When the transmission was successful, the **Network Client** window should now display a confirmation message about the reception of an item with the same name that has been sent in step 6 (the name was `item_1` in this example).



8. Stop the **Network Client** on the receiving PC listening by clicking the **Stop** button and close it clicking the **Close** button.



**NOTE:**

The connection between Unicorn Speller on the sending PC and the receiving PC is working now. The only thing left is to configure your external application to receive data from Unicorn Speller. Please check the manual of that application for further instructions on how to do this.

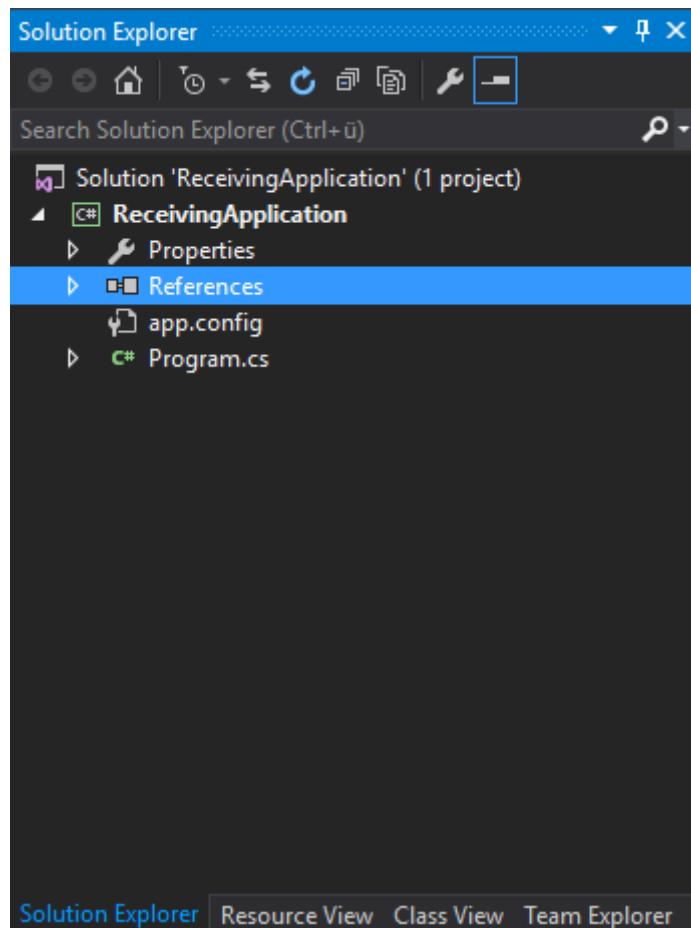
## 18.17. PROGRAMMING APPLICATIONS USING UNICORN SPELLER'S NETWORK OUTPUT

This section describes the steps necessary to receive items sent from Unicorn Speller for developing applications for Microsoft .NET Framework 4.7.1 and the C# programming language with the Microsoft Visual Studio 2015 IDE.

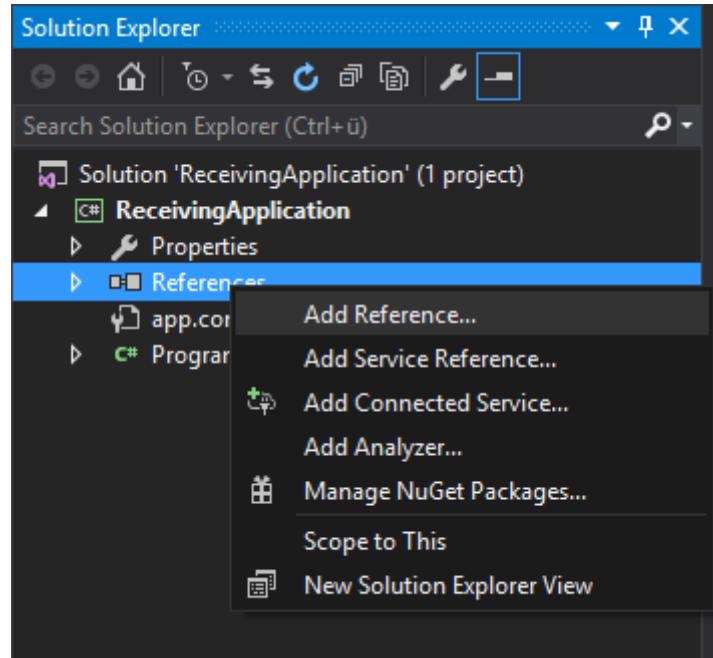
**NOTE:**

For software developers only!

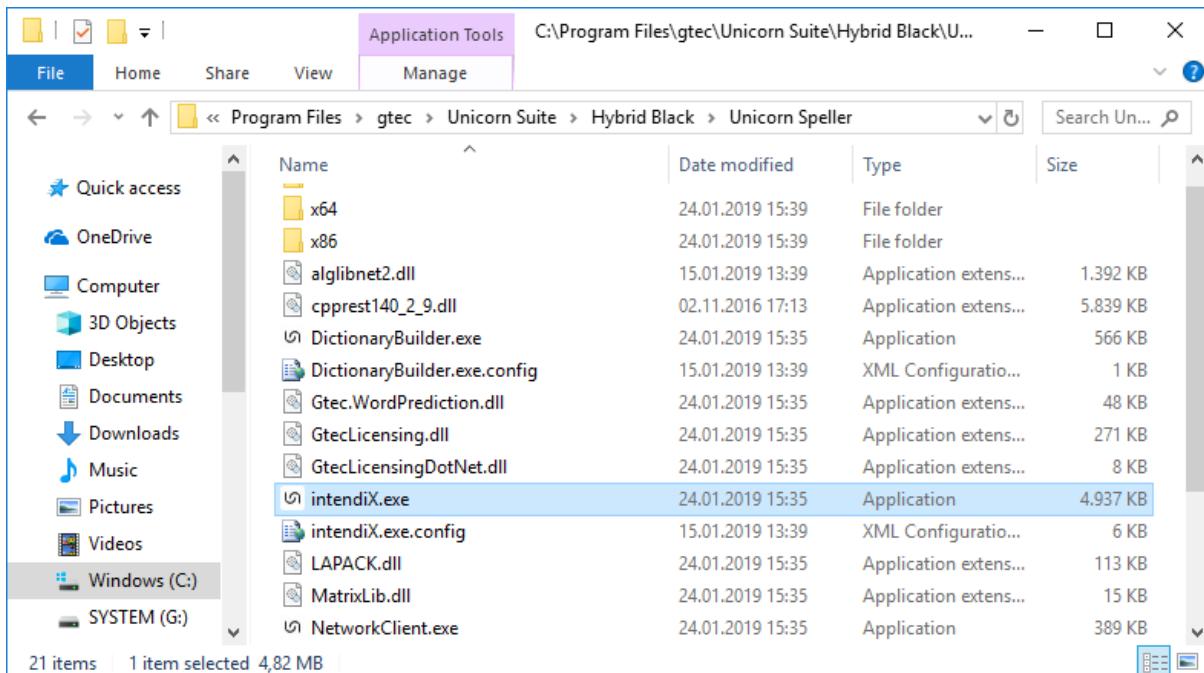
1. Assume you have an existing project opened under Visual Studio. For this example, we use an empty project named ReceivingApplication to demonstrate the steps. The solution explorer gives us an overview about the project's content.



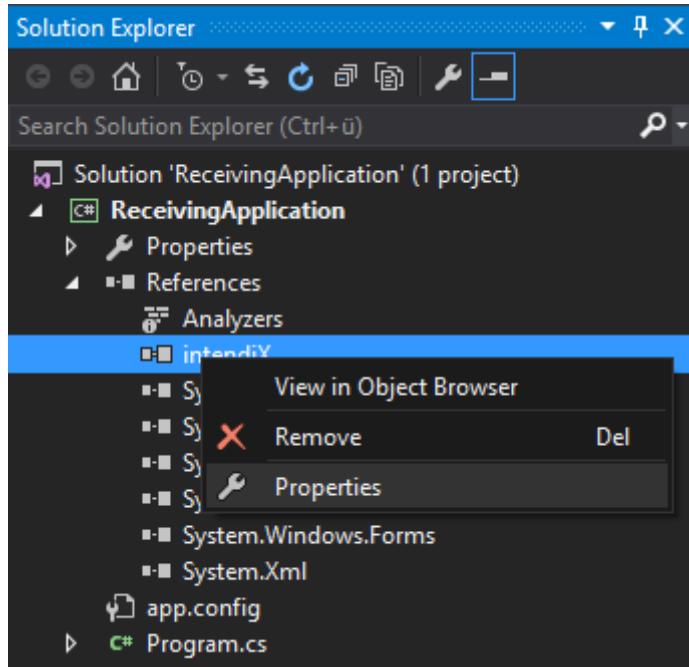
2. The first thing you have to do is to make your application familiar with the class structure that the received items have. This class is named `BoardItem` and is contained in the Unicorn Speller assembly in the namespace `Intendix.Board`. Therefore, Unicorn Speller must be installed on your developing computer.
  - a. In the solution explorer of your project, right click the **References** icon and select **Add Reference...** from the context menu.



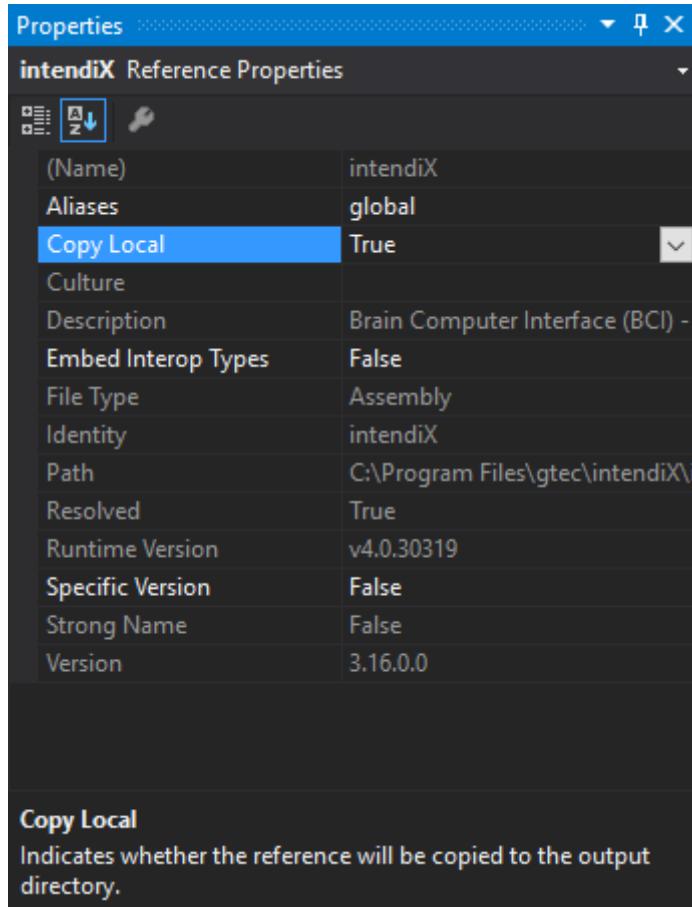
- b. In the opening **Add References** dialog, click the **Browse...** button and navigate to the folder where Unicorn Speller has been installed to. Select the file `intendiX.exe` and click **OK**.



- c. Now you should see the item **Unicorn Speller** in the list under **References** in the solution explorer. Right click the **Unicorn Speller** item in the list under **References** and select **Properties**.



- d. The **Properties** window for the Unicorn Speller reference is displayed. Ensure that the property **Copy Local** is set to **true**. This will copy the file **intendiX.exe** to the output directory of your application. When deploying the project (creating setup files), ensure that the **intendiX.exe** file is copied in the same directory from which your application is supposed to run after installation.



3. Unicorn Speller sends the items and its contents in binary serialized form within a UDP datagram over the network. To receive such an item, you have to set up a UDP connection that listens on the designated destination port.

The following example code shows how to receive an item from the network (it is similar to the content of the file `Program.cs` in the `ReceivingApplication` project). It is assumed that the IP address of the receiving PC, where this code is executed, is `192.168.100.162` and port `10` is used. The `ReceiveItem` method blocks until it receives data on the specified address and port. If you wish to avoid this blocking, the `UdpClient` class provides asynchronous receiving methods (see the MSDN documentation for further information):



#### C# EXAMPLE (PROGRAM.CS)

```
using System;
using System.IO;
using System.Net;
using System.Net.Sockets;
using System.Runtime.Serialization.Formatters.Binary;
using Intendix.Board;

namespace ReceivingApplication
{
    static class Program
    {
        /// <summary> The main entry point for the application.</summary>
        static void Main()
        {
            //specify local IP address and port from where to receive data from
            //Unicorn Speller must send the data to this end point
            IPAddress ip = IPAddress.Parse("192.168.100.162");
            int port = 10;

            //block until an item from Unicorn Speller is received
            BoardItem receivedItem = ReceiveItem(ip, port);

            //print the name of the received item
            if (receivedItem != null)
                Console.WriteLine("Item '{0}' received.", receivedItem.Name);
        }

        /// <summary> Receives a <see cref="BoardItem"/> from the network on the provided
        /// <paramref name="portNumber"/>. </summary>
        /// <param name="ipAddress">The local IP address or a broadcast IP address from which you want
        /// to receive items.</param>
        /// <param name="portNumber">The local port number from which you want to receive
        /// items.</param>
        /// <returns> The received <see cref="BoardItem"/>. <b>null</b> if an error occurred.</returns>
        /// <remarks> This method blocks until a valid <see cref="BoardItem"/> is received. If you want to
        /// do the whole thing non-blocking, you can use the <see cref="UdpClient.BeginReceive"/> and
        /// <see cref="UdpClient.EndReceive"/> methods instead.</remarks>
        static public BoardItem ReceiveItem(IPAddress ipAddress, int portNumber)
        {
            BoardItem receivedItem = null;
            UdpClient udpClient = null;
            try
            {
                //create an endpoint from the specified IP address and port number
                IPEndPoint endPoint = new IPEndPoint(ipAddress, portNumber);

                //create a udp client on the specified end point
                udpClient = new UdpClient(endPoint);
```



```
//enable receiving from broadcast addresses
udpClient.EnableBroadcast = true;

//receive data from the UDP client
byte[] receivedBytes = udpClient.Receive(ref endPoint);

//prepare for deserialization of the received data
MemoryStream memoryStream = new MemoryStream(receivedBytes);
BinaryFormatter formatter = new BinaryFormatter();
memoryStream.Position = 0;

//deserialize the byte stream and cast it to type BoardItem
//it is assumed that only data of type BoardItem will be transmitted
//to this end point so any incoming data is supposed to be of type
//BoardItem; if not, an InvalidCastException will be thrown
receivedItem = (BoardItem) formatter.Deserialize(memoryStream);
}

catch (InvalidCastException e)
{
    Console.WriteLine("Received data didn't contain a valid BoardItem.\n{0}", e.ToString());
}
catch (ArgumentOutOfRangeException e)
{
    Console.WriteLine("Invalid port number specified.\n{0}", e.ToString());
}
catch (SocketException e)
{
    Console.WriteLine("An error occurred accessing the underlying socket.\n{0} ", e.ToString());
}
catch (ObjectDisposedException e)
{
    Console.WriteLine("The underlying socket has been closed.\n{0}", e.ToString());
}
finally
{
    try
    {
        //close UDP connection
        if (udpClient != null)
            udpClient.Close();
    }
    catch (SocketException e)
    {
        //an error occurred while accessing the socket
        Console.WriteLine("An error occurred accessing the underlying socket.\n{0}", e.ToString());
    }
}
return receivedItem;
}
```



```
}
```

Please see the API documentation of the `BoardItem` class for details on its provided content. You can use the `BoardItem.Name` property to identify the transmitted item. This property can be set through the **Name** field by the user in the tab **Board Content** of the **Board Configuration** dialog (see section 18.14.2).



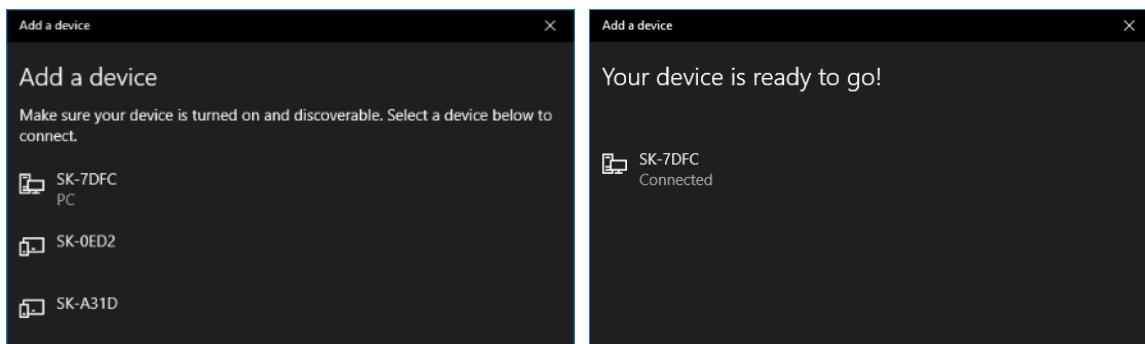
## 19. UNICORN SPHERO

Unicorn Sphero is an application that allows you to control a Sphero SPRK+ using the P300 complex. The Unicorn Sphero receives items that Unicorn Speller sends over the network and controls a Sphero SPRK+ device accordingly.

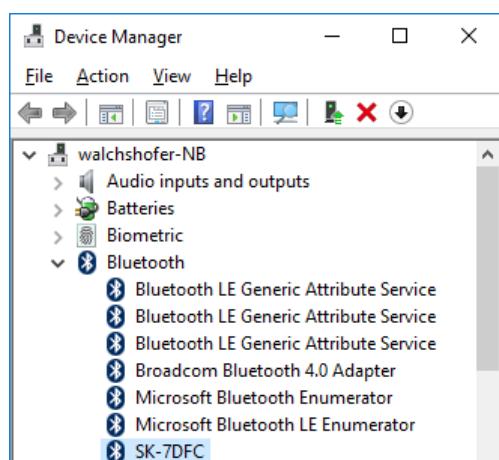
Sphero SPRK+ is a spherical robot toy designed by Sphero (Boulder, CO). It is a white orb wrapped in polycarbonate plastic, capable of rolling and changing colors.

### 19.1. PAIR WITH A SPHERO SPRK+

1. Insert a Bluetooth dongle to the PC or check that the Bluetooth adapter of your computer is enabled.
2. Enter the Bluetooth Settings of your computer.
3. The Sphero SPRK+ device should be listed as discovered Bluetooth device and should not be paired or connected. The name of the Sphero device in the Bluetooth Settings is equal to its serial number starting with SK-xxxx. Select the serial number of the Sphero device to pair with.



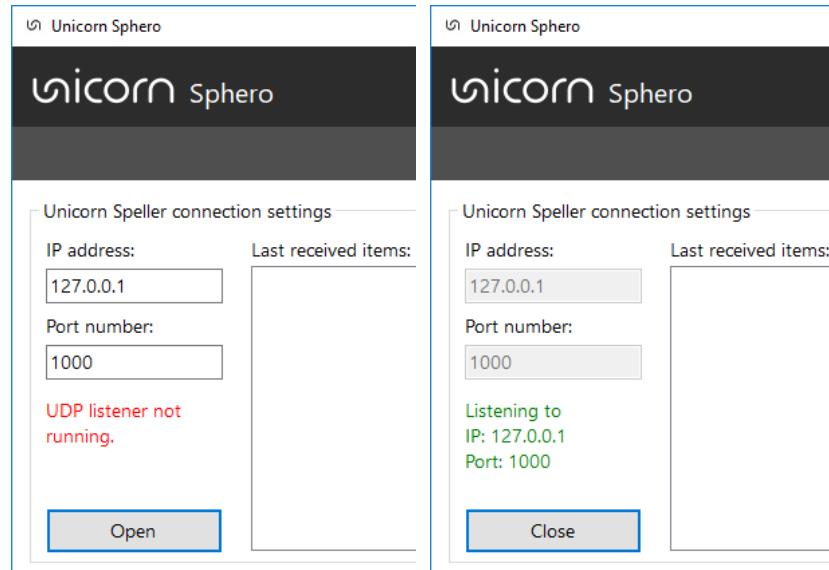
4. The status of the Sphero device should have changed to Paired or Connected. Open the Device Manager and check if the device was added to the Bluetooth devices.
5. The device is ready to be used by the Unicorn Sphero application if all steps could be performed successfully.



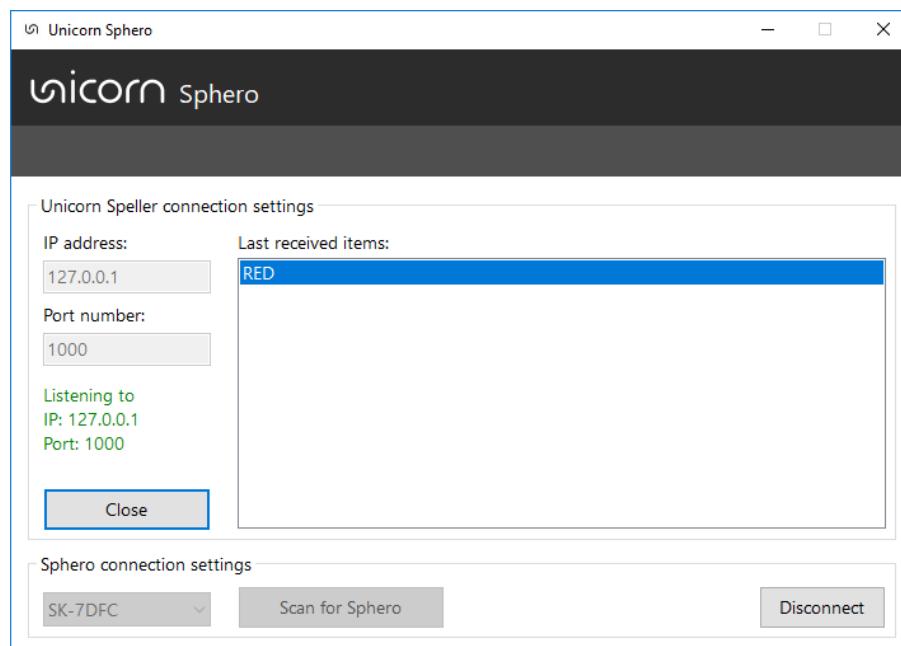


## 19.2. CONFIGURING UNICORN SPHERO

In order to enable Unicorn Sphero to receive the items that Unicorn Speller sends over the network, Unicorn Sphero must be configured with the IP address and port where the items are received (by default IP address 127.0.0.1 and port 1000). Enter the IP address of the machine where you are running Unicorn Sphero. This is the same IP address that must be configured for Unicorn Speller.



To control a Sphero SPRK+ device, the device must be powered and discoverable. The Unicorn Sphero lists all available devices if "Scan for Sphero" is clicked. Select the serial number of the paired Sphero and click Connect. The button text will change to disconnect if the Sphero is connected properly.



From now on, each item that Unicorn Speller classifies will be received by Unicorn Sphero, and the corresponding action is performed.

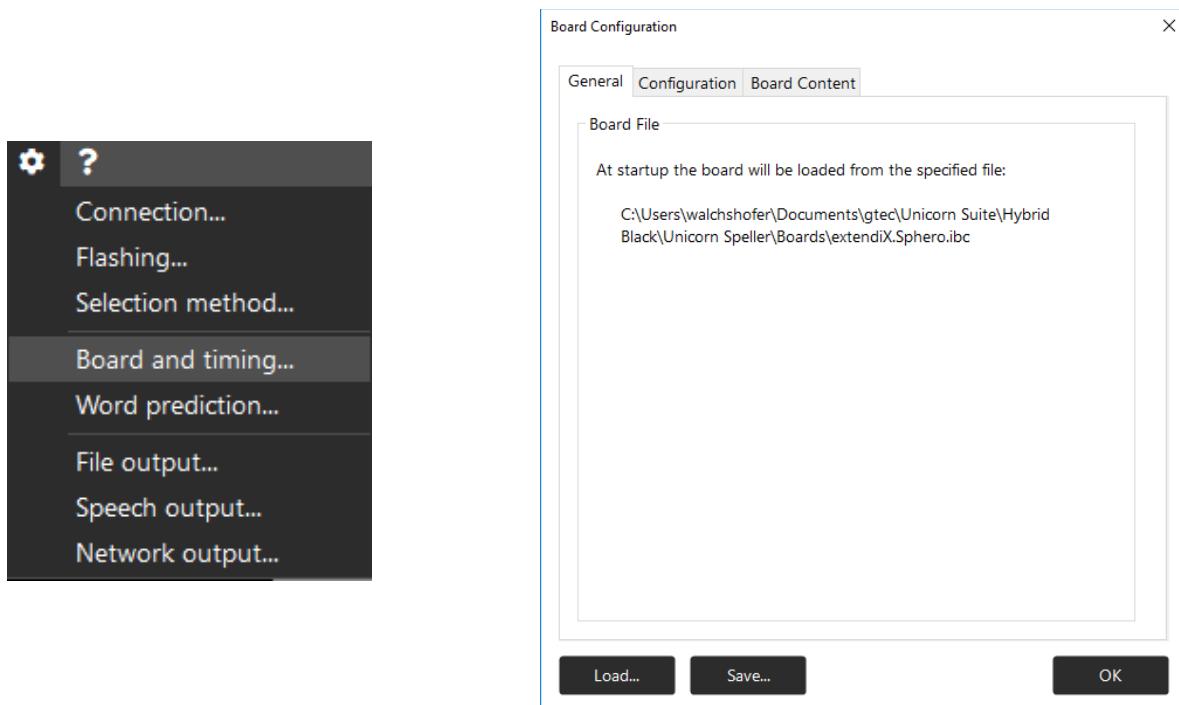


## 19.3. CONFIGURE UNICORN SPELLER

By default, Unicorn Speller should be configured to work with Unicorn Sphero without any user actions. In case Unicorn Speller was reconfigured, the following steps can be performed to establish a communication between Unicorn Speller and Unicorn Sphero

### 1.1.1. LOAD THE UNICORN SPHERO BOARD INTO UNICORN SPELLER

Unicorn Speller loads the last Board that was used by default. To load the Unicorn Sphero board, open the **Board Configuration** dialog by selecting **⚙️** → “Board and timing...” from the menu bar and click the “Load...” button.



Select the Unicorn Sphero board file named “extendiX.Sphero.ibc” from:

C:\Users\<username>\Documents\gtec\ Unicorn Suite\Hybrid Black\Unicorn Speller\Boards\extendiX.Sphero.ibc



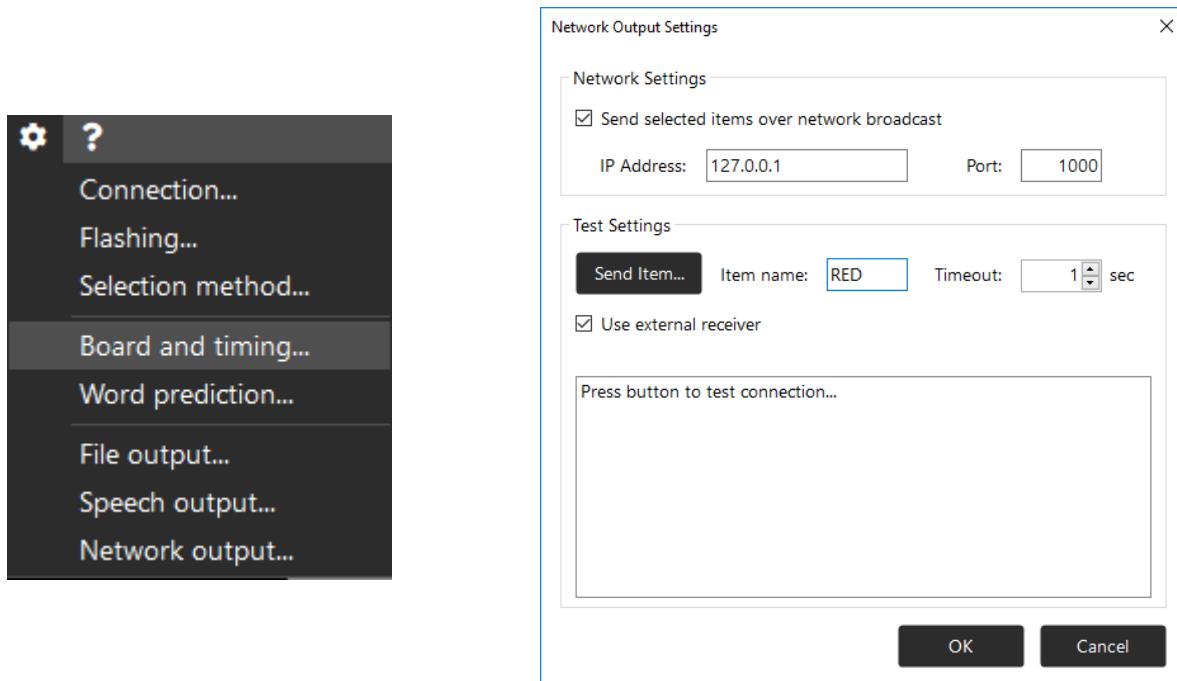
#### ATTENTION:

Never change the content of this board file shown in the Board Content tab of the Board Configuration dialog. Unicorn Sphero uses the item's configuration to identify the selected action and to react accordingly.



### 1.1.2. ENABLE UNICORN SPELLER NETWORK OUTPUT

To enable network output of Unicorn Speller, open the **Network Output Settings** dialog by selecting → “Network Output...” from the menu bar of Unicorn Speller.



Please check the “Send classified items over network broadcast” checkbox and enter the destination IP address and port of the computer where the Unicorn Sphero is supposed to run. Click **OK** to override the settings and close the “**Network Output Settings**” dialog. Now, each time Unicorn Speller classifies an item, this item will also be sent over the network to the specified destination IP address and port.



#### NOTE:

It is recommended to run the Unicorn Sphero on a different computer than Unicorn Speller to ensure optimal performance, and because the Unicorn Sphero could overlap Unicorn Speller. If you still want to run Unicorn Sphero on the same PC as Unicorn Speller, you can enter the local IP address 127.0.0.1.

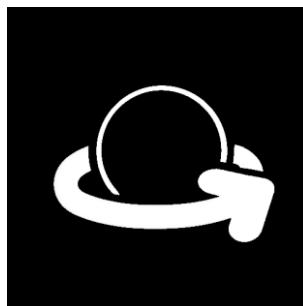


## 19.4. BOARD ITEMS

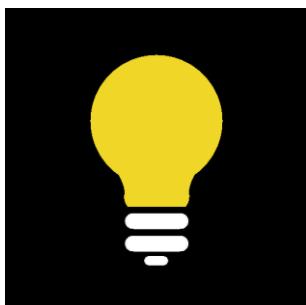
The following section explains the different items of the Unicorn Sphero board.



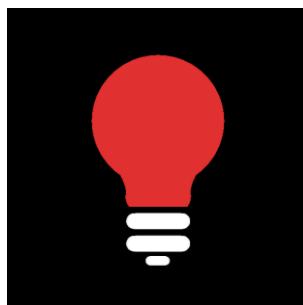
Sphero starts to spin.



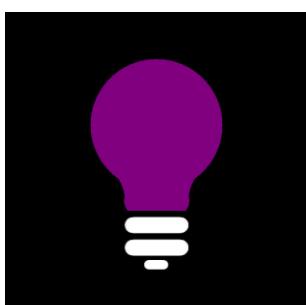
Sphero starts to spin fast.



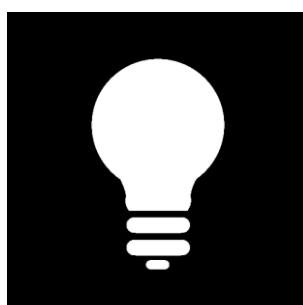
Changes the main LED of Sphero to yellow.



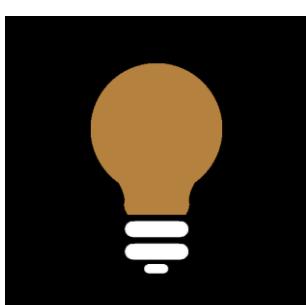
Changes the main LED of Sphero to red.



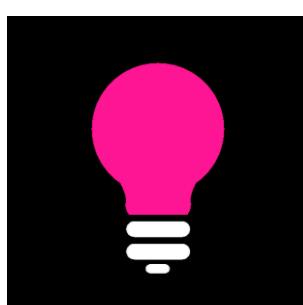
Changes the main LED of Sphero to purple.



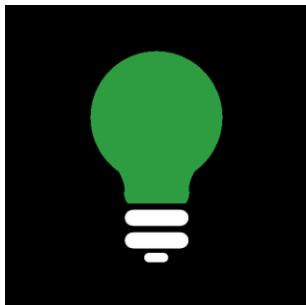
Changes the main LED of Sphero to white.



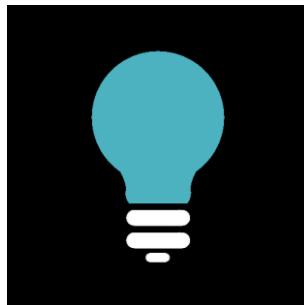
Changes the main LED of Sphero to brown.



Changes the main LED of Sphero to pink.



Changes the main LED of Sphero to green.



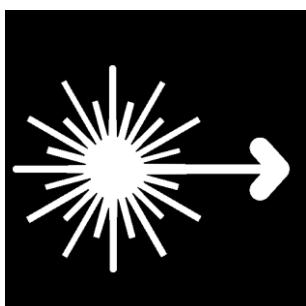
Changes the main LED of Sphero to cyan.



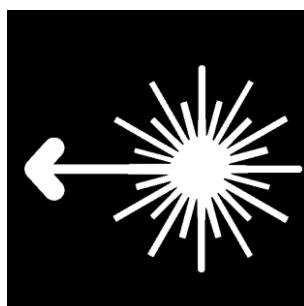
Changes the main LED of Sphero to different colors.



The Sphero starts to blink, indicating the SOS signal.



Enables the front LED of Sphero and turns the device to the right.



Enables the front LED of Sphero and turns the device to the left.



Stops the Sphero if it's moving



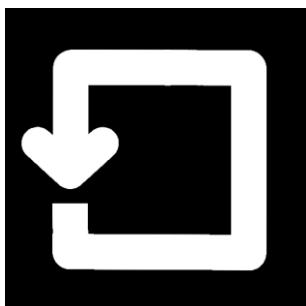
The Sphero moved in a "g" pattern.



The Sphero performs a serpentine line movement.



The Sphero performs a rhombus movement.



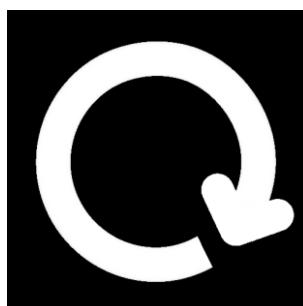
The Sphero performs a square movement.



The Sphero performs a spiral movement.



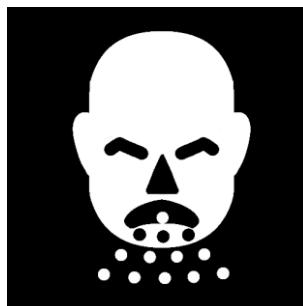
The Sphero performs a triangular movement.



The Sphero performs a circular movement.



Turns off the body light of Sphero to make it "invisible" and turns the body light to white afterwards, signifying a "ghost".



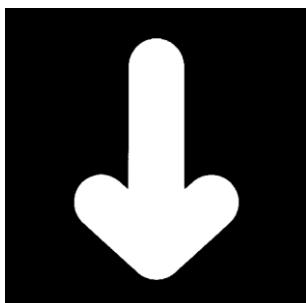
The Sphero performs a fast move and plays a sound signifying a "sneeze".



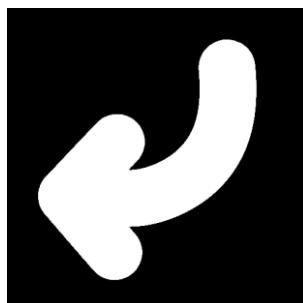
The Sphero starts to move slowly in an uncoordinated way, signifying a "zombie".



The Sphero starts to move in serpentine ways and starts flashing lights, signifying a "dance" movement.



Moves the Sphero backwards.



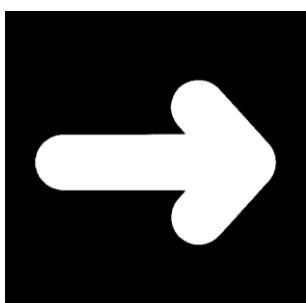
Turns the Sphero to the right.



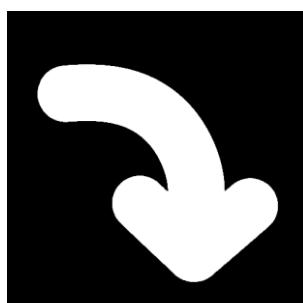
Moves the Sphero forwards.



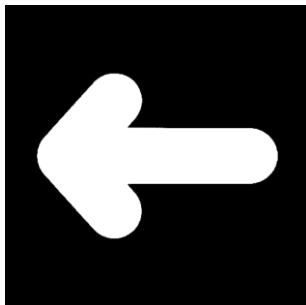
Turns the Sphero to the right.



Turns the Sphero right and then moves right which moves the Sphero to the right.



Turns the Sphero to the right.



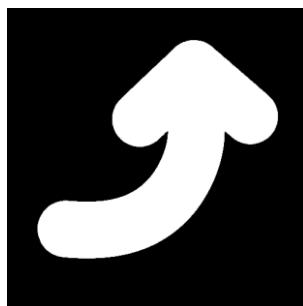
Turns the Sphero left and then moves left, which moves the Sphero to the left.



Turns the Sphero to the right.



Turns the Sphero hard right and then moves right, which moves the Sphero to the right and backward.



Turns the Sphero to the left.



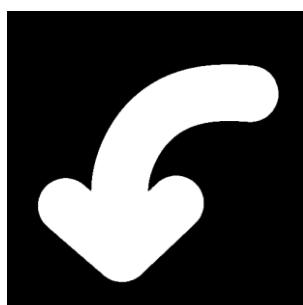
Turns the Sphero slightly left and then moves left, which moves the Sphero to the left and forward.



Turns the Sphero to the left.



Turns the Sphero hard left and then moves left, which moves the Sphero to the left and backward.



Turns the Sphero to the left.



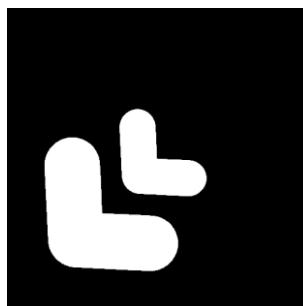
Turns the Sphero slightly right and then moves right, which moves the Sphero to the right and forward.



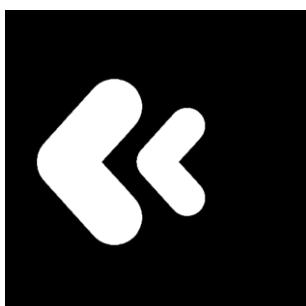
Turns the Sphero to the left.



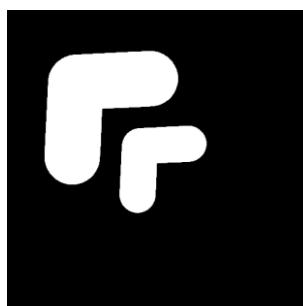
Performs a fast-backward movement.



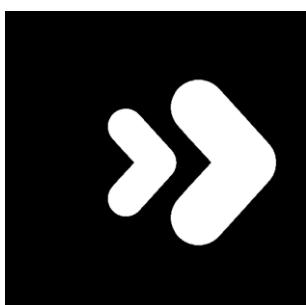
Performs a fast left-down movement.



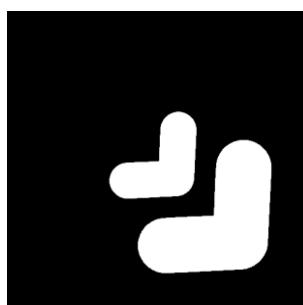
Performs a fast movement to the left.



Performs a fast left-up movement.



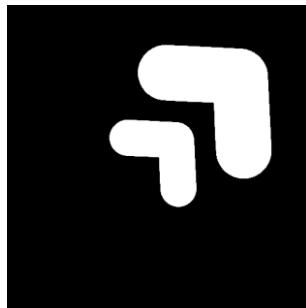
Performs a fast movement to the right.



Performs a fast right-down movement.



Performs a fast-forward movement.

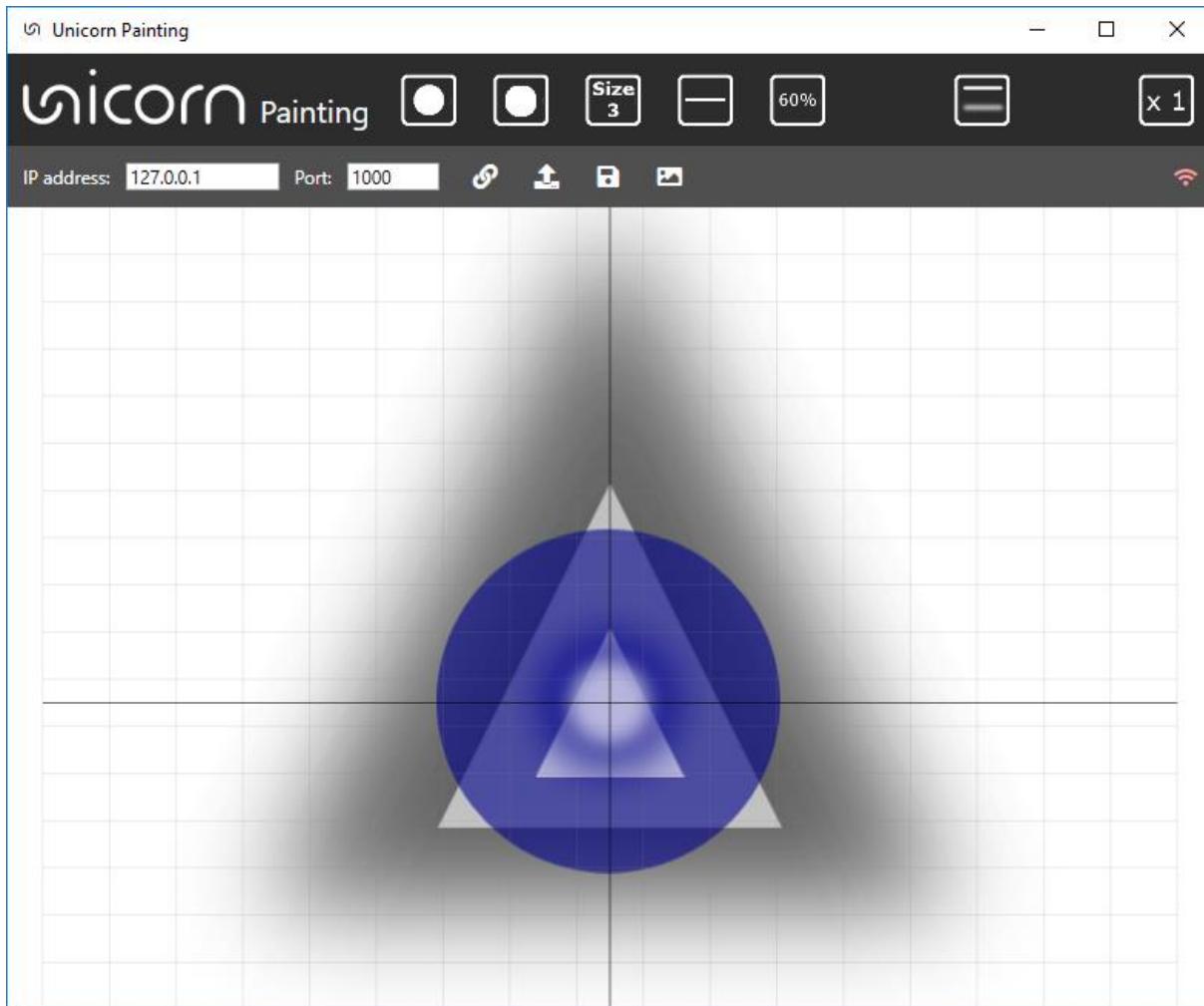


Performs a fast right-up movement.



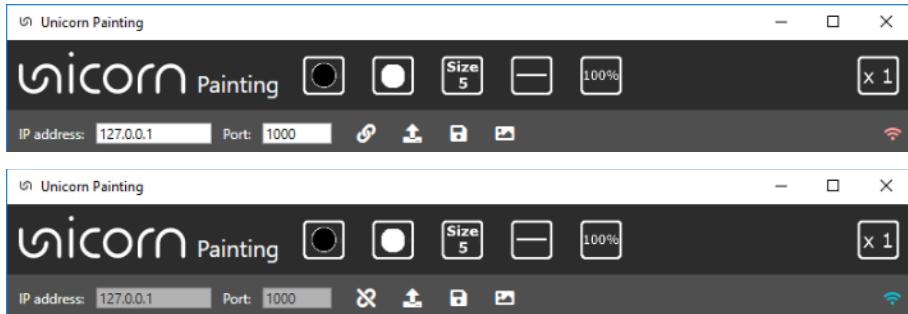
## 20. UNICORN PAINTING

Unicorn Painting is an application that allows you to create paintings using the P300 complex. The Unicorn Painting application receives classified items that Unicorn Speller sends over the network and controls different painting tools to create paintings.



### 20.1. CONFIGURING UNICORN PAINTING

To enable Unicorn Painting to receive the items that Unicorn Speller sends over the network, Unicorn Painting must be configured with the IP address and port where the items are received (by default IP address 127.0.0.1 and port 1000). Enter the IP address of the machine where you are running Unicorn Painting. This is the same IP address that must be configured with Unicorn Speller.



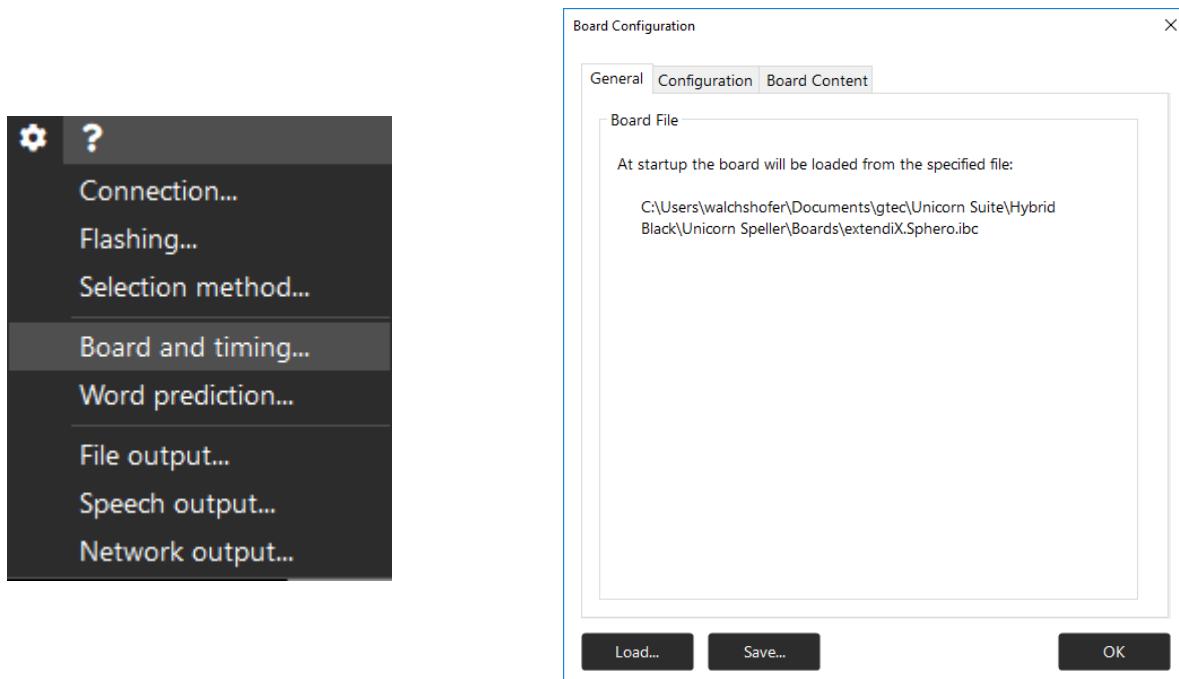
From now on, each item that Unicorn Speller classifies will be received by Unicorn Painting, and the corresponding action is performed.

## 20.2. CONFIGURE UNICORN SPELLER

By default, Unicorn Speller should be configured to work with Unicorn Painting without any user actions. If Unicorn Speller was reconfigured, the following steps can be performed to establish communication between Unicorn Speller and Unicorn Painting.

### 1.1.3. LOAD THE UNICORN PAINTING BOARD INTO UNICORN SPELLER

Unicorn Speller loads the last Board that was used by default. To load the Unicorn Painting board, open the **Board Configuration** dialog by selecting **⚙️** → “Board and timing...” from the menu bar and click the “Load...” button.



Select the Unicorn Painting board file named “UnicornPainting.ibc” from:



C:\Users\<username>\Documents\gtc\ Unicorn Suite\Hybrid Black\Unicorn Speller\Boards\UnicornPainting.ibc

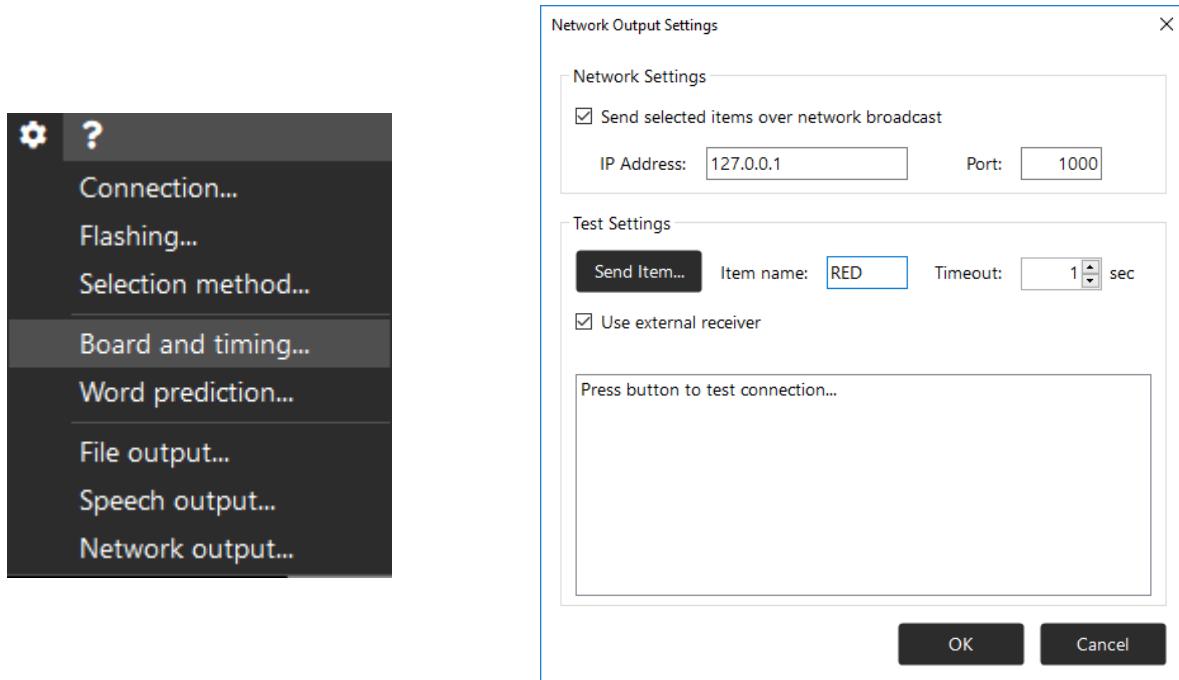


#### ATTENTION:

Never change the content of this board file shown in the Board Content tab of the Board Configuration dialog. Unicorn Painting uses the item's configuration to identify the selected action and to act accordingly.

#### 1.1.4. ENABLE UNICORN SPELLER NETWORK OUTPUT

To enable network output of Unicorn Speller open the **Network Output Settings** dialog by selecting → “Network Output...” from the menu bar of Unicorn Speller.



Please check the “Send classified items over network broadcast” checkbox and enter the destination IP address and port of the computer where the Unicorn Painting is supposed to run. Click **OK** to override the settings and close the “**Network Output Settings**” dialog. Now, each time Unicorn Speller classifies an item, this item will also be sent over the network to the specified destination IP address and port.

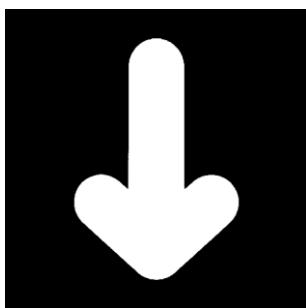


#### NOTE:

It is recommended to run the Unicorn Painting on a different computer than Unicorn Painting to ensure optimal performance, and because the Unicorn Painting could overlap Unicorn Speller. If you still want to run Unicorn Painting on the same PC as Unicorn Speller, you can enter the local IP address 127.0.0.1.

### 20.3. BOARD ITEMS

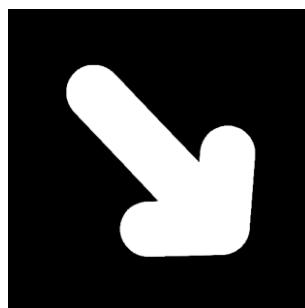
The following section explains the different items of the Unicorn Painting board.



Move the cursor n number of fields down. N is defined by the currently set step-size.

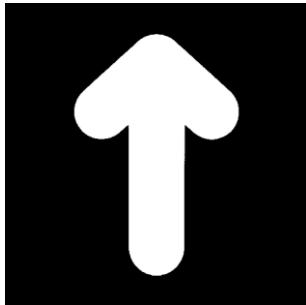
Key: Arrow Down

Numpad: 2



Move the cursor n number of fields to the right and down. N is defined by the currently set step-size.

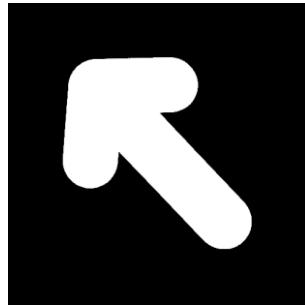
Numpad: 3



Move the cursor n number of fields up. N is defined by the currently set step-size.

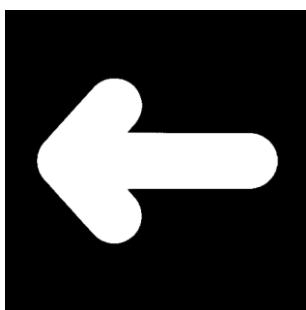
Key: Arrow Up

Numpad: 8



Move the cursor n number of fields to the left and up. N is defined by the currently set step-size.

Numpad: 7



Move the cursor n number of fields to the left. N is defined by the currently set step-size.

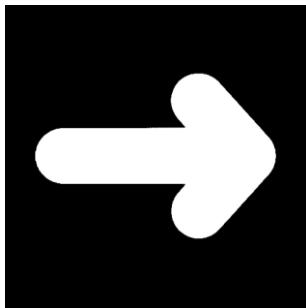
Key: Arrow Left

Numpad: 4



Move the cursor n number of fields to the left and down. N is defined by the currently set step-size.

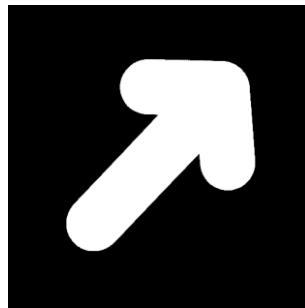
Numpad: 1



Move the cursor n  
number of fields to  
the right. N is  
defined by the  
currently set step-  
size.

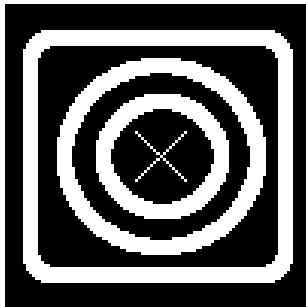
Key: Arrow Right

Numpad: 2



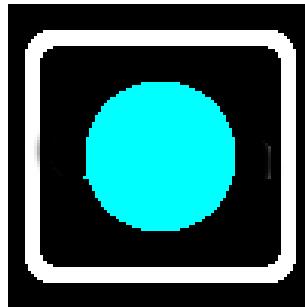
Move the cursor n  
number of fields to  
the right and up. N  
is defined by the  
currently set step-  
size.

Numpad: 9



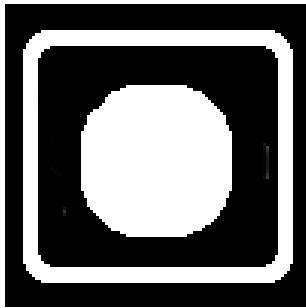
Center the cursor.

Numpad: 5



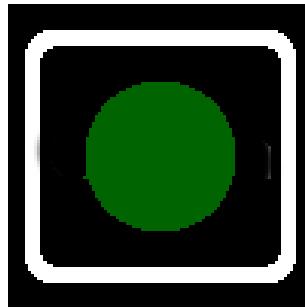
Change the selected  
color to cyan.

Key: C



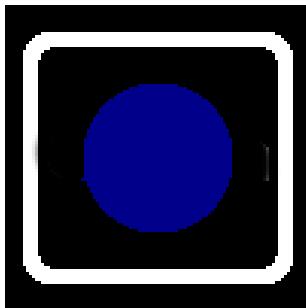
Change the selected  
color to white.

Key: C



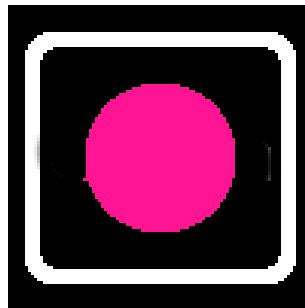
Change the selected  
color to green.

Key: C



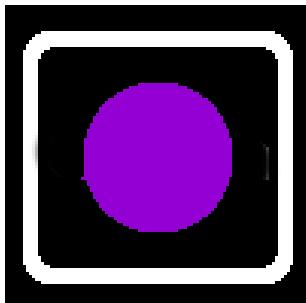
Change the selected  
color to blue.

Key: C



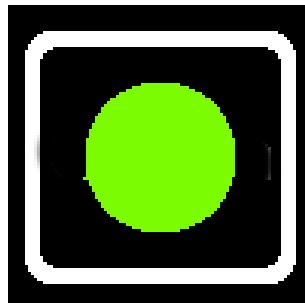
Change the selected  
color to pink.

Key: C



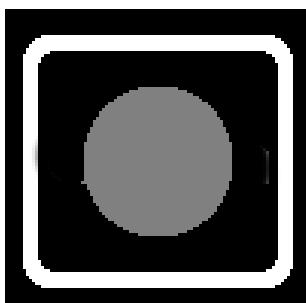
Change the selected color to purple.

Key: C



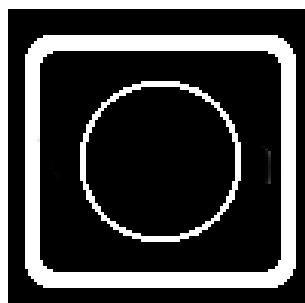
Change the selected color to lime green.

Key: C



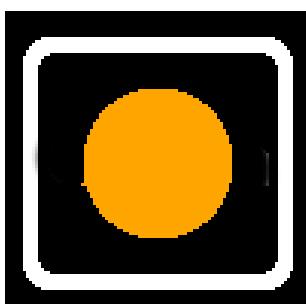
Change the selected color to grey.

Key: C



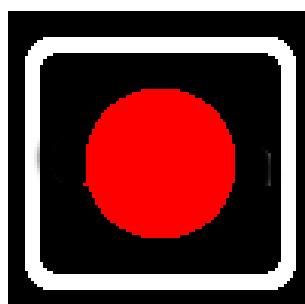
Change the selected color to black.

Key: C



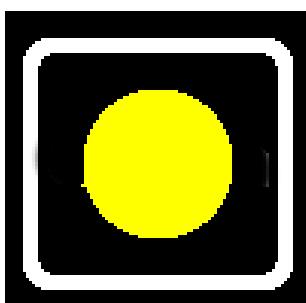
Change the selected color to orange.

Key: C



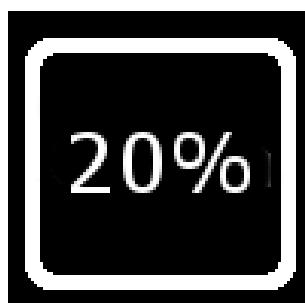
Change the selected color to red.

Key: C



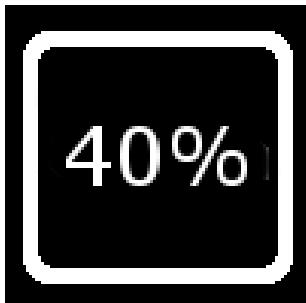
Change the selected color to yellow.

Key: C



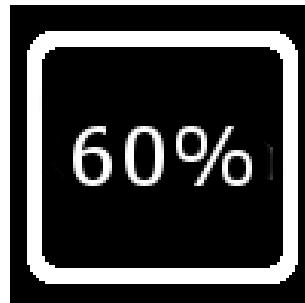
Change the selected opacity to 20%

Key: O



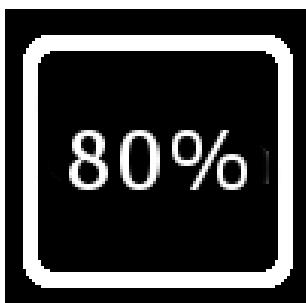
Change the selected opacity to 40%

Key: O



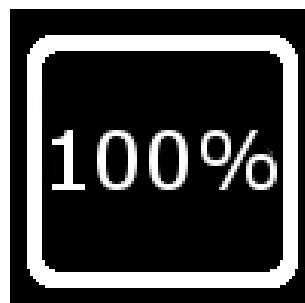
Change the selected opacity to 60%

Key: O



Change the selected opacity to 80%

Key: O



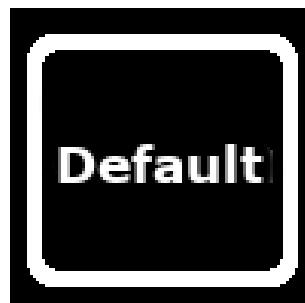
Change the selected opacity to 100%

Key: O



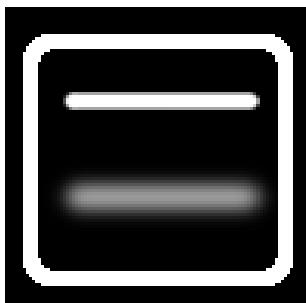
Change the Background color to the currently selected color.

Key: H



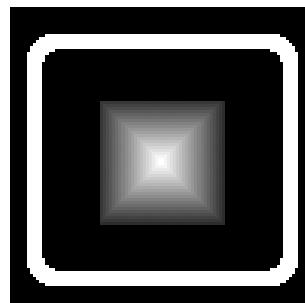
Restores default step-size, shape size, linewidth and shape effect.

Key: D



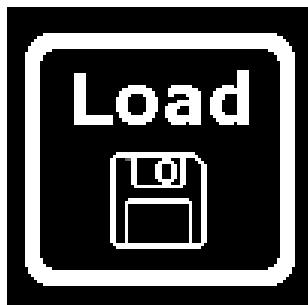
Change the line sharpness if a line tool is selected.

Key: B



Change the shape sharpness if a shape is selected.

Key: V



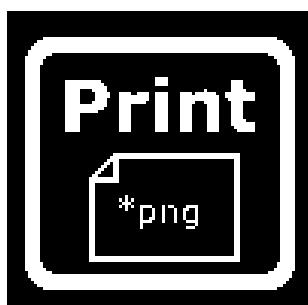
Load image.

Key: F2



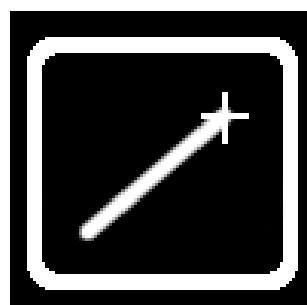
Save current image.

Key: F1



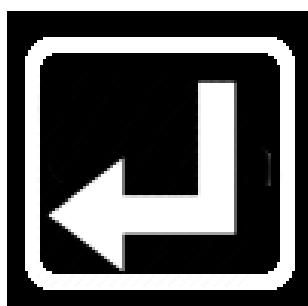
Export image to png file.

Key: F3



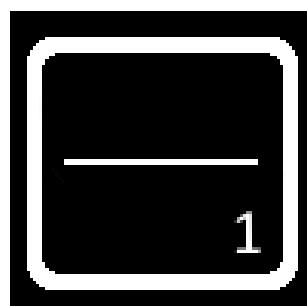
The line tool can be used to draw lines. It is required to select the Enter item to define start and stop of the line.

Key: T



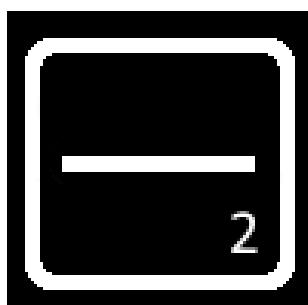
The Enter key must be selected to draw a currently selected shape or to define start and stop of a line.

Key: Enter



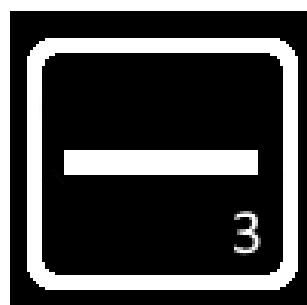
Set linewidth to one.

Key: W



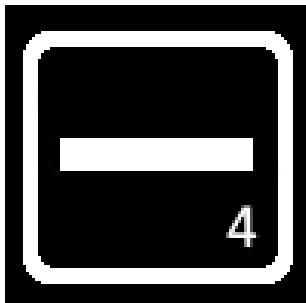
Set linewidth to two.

Key: W



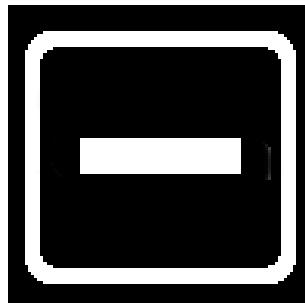
Set linewidth to three.

Key: W



Set linewidth to four.

Key: W



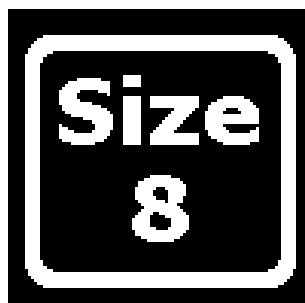
Decrease currently set size by one.  
Shapes will become smaller.

Numpad: -



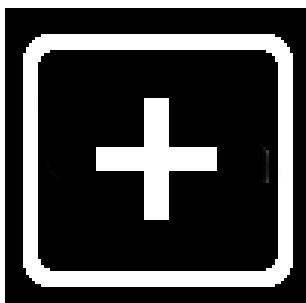
Set the size to three.  
Shapes will be small.

Numpad: + or -



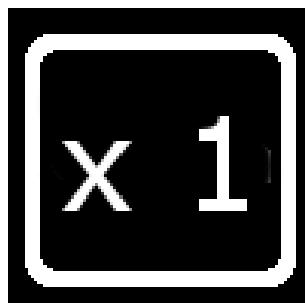
Set the size to eight.  
Shapes will be large.

Numpad: + or -



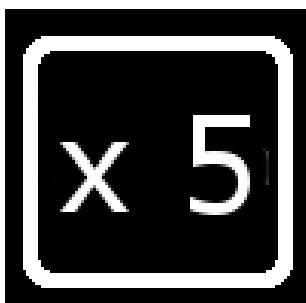
Increase currently set size by one. Shapes will become smaller.

Numpad: +



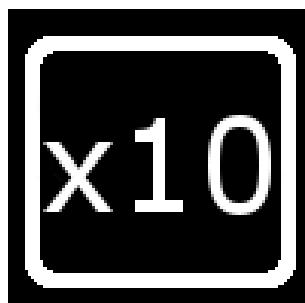
Set the cursor speed to one. The cursor makes one field steps.

Key: S



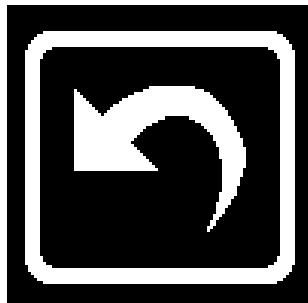
Set the cursor speed to five. The cursor makes five field steps.

Key: S



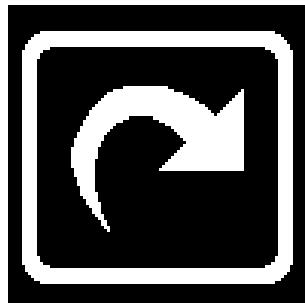
Set the cursor speed to ten. The cursor makes ten field steps.

Key: S



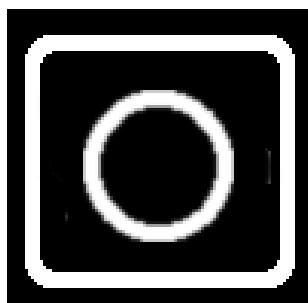
Undo last step.

Key: U



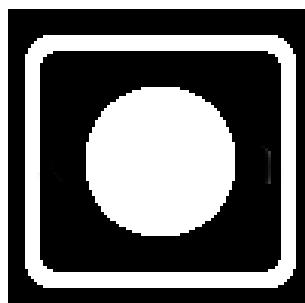
Redo last step.

Key: I



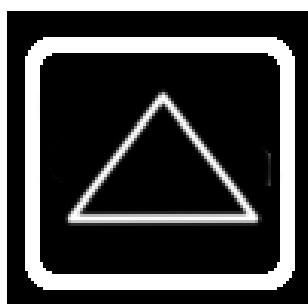
The cycle tool draws  
cycle outlines.

Key: T



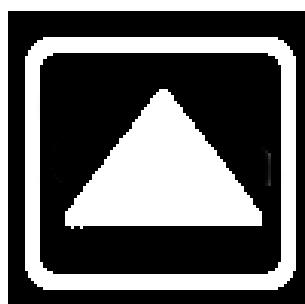
The cycle tool draws  
filled cycles.

Key: T



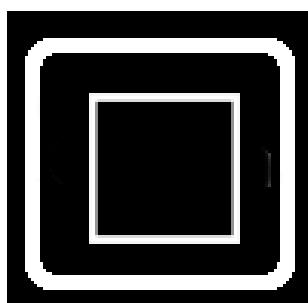
The triangle tool  
draws triangle  
outlines.

Key: T



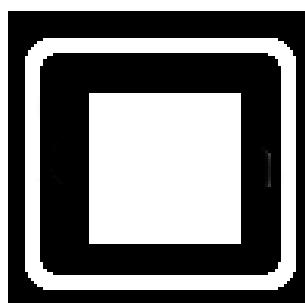
The triangle tool  
draws filled triangles.

Key: T



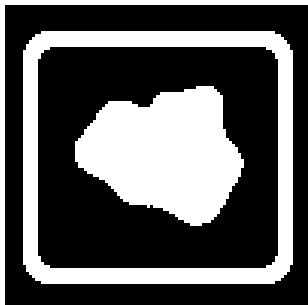
The square tool  
draws square  
outlines.

Key: T



The square tool  
draws filled squares.

Key: T



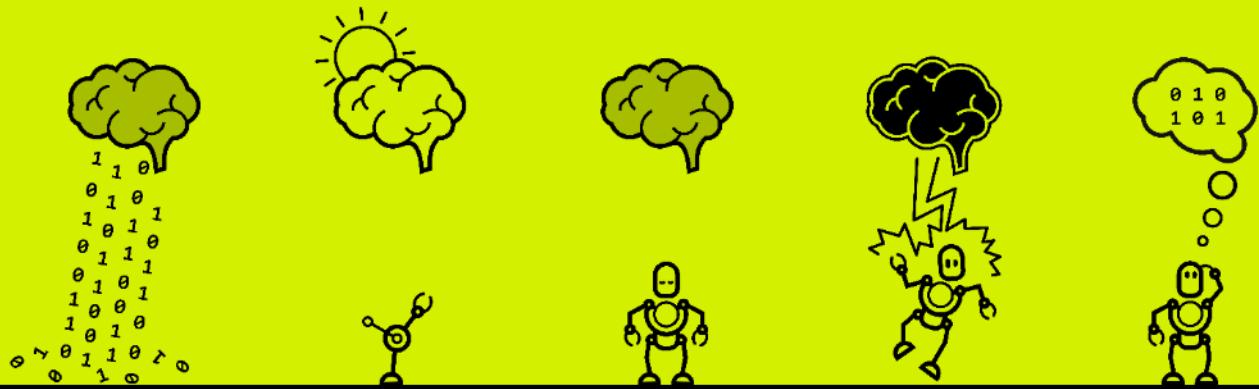
The cloud tool draws filled cloudy shapes.

Key: T



Enter pencil mode.

Key: T



**BR41N.IO**

THE BRAIN-COMPUTER INTERFACE  
DESIGNERS HACKATHON

[WWW.BR41N.IO](http://WWW.BR41N.IO)



unicorn  
THE BRAIN INTERFACE