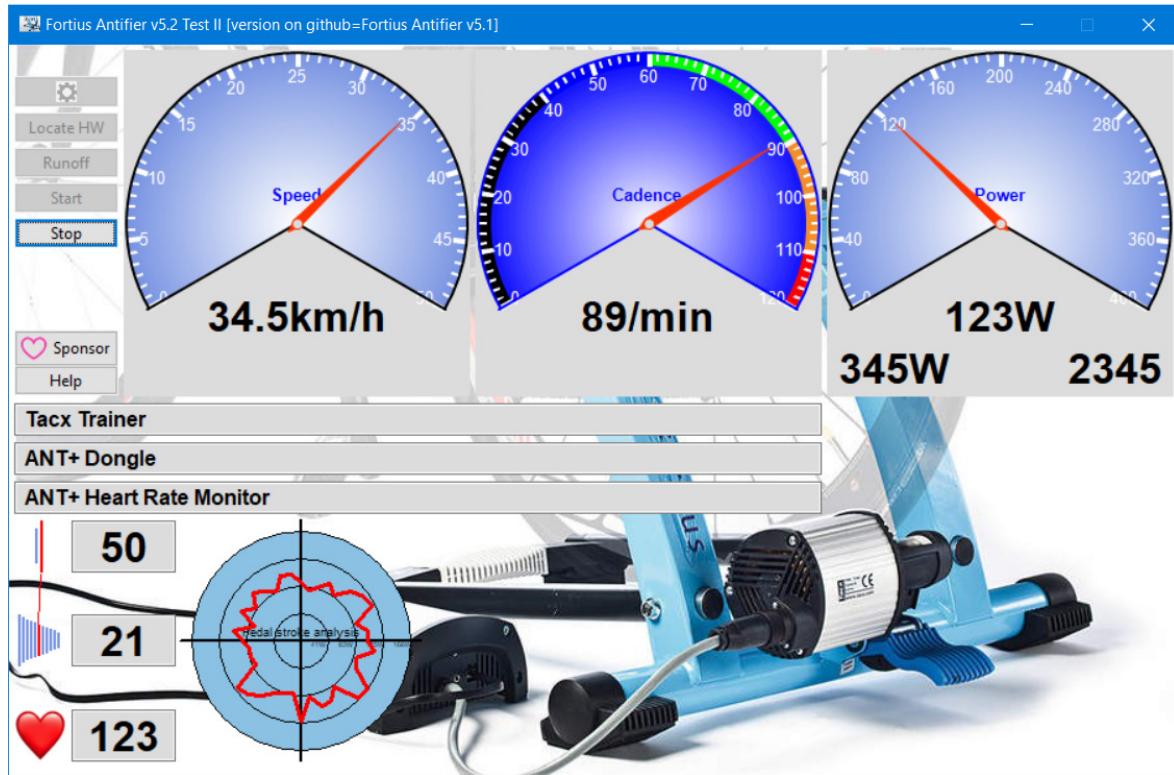




FortiusAnt

User Manual

Author and copyright: Wouter Dubbeldam
© 2020, 2021





Contents

1.	Introduction.....	5
1.1	Summary.....	5
1.2	Thanks	5
1.3	The structure of this manual	6
1.4	The story in a picture:.....	6
1.5	References, abbreviations, and terminology	7
1.6	Revision History	8
2.	Background information.....	10
2.1	FortiusAnt Sensor- and Monitor devices.....	10
2.2	Description	13
2.2.1	Resistance	13
2.2.2	Power- or Ergo-mode	13
2.2.3	Resistance mode	13
2.3	Tacx trainers	14
2.3.1	Tacx Catalogue 2012.....	14
2.3.2	General	14
2.3.3	Tacx proprietary ANT trainers.....	15
2.4	Bluetooth Low-Energy (BLE)	16
2.4.1	Introduction	16
2.4.2	Design.....	16
2.5	Calculations.....	17
2.5.1	Power mode.....	17
2.5.2	Grade mode	17
2.6	Tacx Trainer Power ranges.....	18
3.	Installation instructions	19
3.1	Introduction	19
3.2	Requirements.....	19
3.2.1	You, the Tacx Athlete	19
3.2.2	Access rights and firewall	19
3.2.3	ANT+ or Bluetooth Low Energy?	19
3.2.4	Hardware	20
3.3	Windows – QuickStart, using executable version.....	20
3.4	Download FortiusAnt from github.....	21
3.5	Install Python.....	21
3.6	Install USB-driver	22
3.6.1	Windows – prepackaged drivers.....	22
3.6.2	Windows – from libusb.....	22
3.6.3	MacOS	23



3.6.4	Linux – General.....	23
3.6.5	Linux Ubuntu 20.04.....	23
3.7	Install ANTdongle.....	24
3.8	Using Bluetooth Low Energy (BLE) Bluetooth is a short-range wireless communication technology standard.....	24
3.8.1	Supported Hardware.....	24
3.8.2	Installation on macOS.....	24
3.8.3	Installation on Windows.....	24
3.8.4	Run FortiusANT with BLE support.....	25
3.9	Start FortiusAnt	25
3.10	Settings in the JSON-file	26
3.11	Check FortiusAnt.....	26
3.12	Raspberry Pi.....	28
3.12.1	Required hardware	28
3.12.2	Installation decisions.....	29
3.12.3	Operating system.....	29
3.12.4	Start and stop the Raspberry.....	30
3.12.5	Install software.....	32
3.12.6	Connect LEDs and a button to the Raspberry Pi.....	34
3.12.7	Connect TFT screen to raspberry.....	35
3.12.8	Now pair with a CTP	36
4.	Operating instructions.....	37
4.1	The main functions of FortiusAnt and the head unit.....	37
4.1.1	Settings	37
4.1.2	Locate HW	37
4.1.3	Runoff test.....	37
4.1.4	Start	38
4.1.5	Stop.....	38
4.1.6	Buttons on the Tacx head unit.....	38
4.1.7	ANT+ Remote Control	39
4.2	Command line	39
4.3	Modify settings	41
4.4	Locate HW	42
4.5	The FortiusAnt display in power mode.....	43
4.6	The FortiusAnt display in grade mode	44
4.7	Debugging FortiusAnt	45
5.	FAQ and Troubleshooting.....	46
5.1	Low cadence on Fortius	46
5.2	Flat resistance on magnetic brake at low speed (#223)	46
5.3	Power tuning with Golden Cheetah (#231).....	46



5.4	Zwift speed does not match Garmin	47
5.5	Average speed in Trainer Road	47
5.6	Can TTS4 and FortiusAnt coexist?	48
5.7	TTS and FortiusANT on Dedicated USB-ports	49
5.8	Tacx head unit with firmware to be loaded	49
5.8.1	"Unable to load firmware" on 1942 head unit.....	50
5.9	Fortius without cadence sensor	50
5.10	Two ANTdongles – disturbed communication.....	50
5.11	Tacx returns insufficient data	51
5.12	Sudden drop of requested power	51
5.13	Zwift on Android does not pair with FortiusAnt.....	51
5.14	Detailed notes on Tacx Genius	51
5.15	Detailed notes on Tacx Bushido.....	52
5.16	Bluetooth (BLE) needs lib_programname, bleno, express (#229)	52
5.17	Bluetooth (BLE) Node.js installation does not install tools.....	52
5.18	Odd behavior in cold garage (#248).....	53
6.	Power curve validation.....	54
6.1	General	54
6.2	Test for Tacx Fortius (T1932)	54
6.3	Test for i-Magic (T1901-T1902)	55
6.4	PowerCurve for i-Flow (T1901-T1932)	57



1. Introduction

1.1 Summary

Tacx created trainers and provided software (Tacx Training Software, TTS) to enable users to do structured training or ride in a virtual world. Trainer and TTS were sold in a package and the interface was not available to other manufacturers.

Based upon these initial products, open standards were defined. ANT+ defines how CTP's (Cycling Training Programs) and FE-C's (Controllable Fitness Equipment) communicate with each other. This open standard enables that software (CTP) and hardware (FE-C) can be created by different manufacturers.

Known CTP's are Zwift, Trainer Road, Rouvy, Sufferfest, Golden Cheetah, Road Grand Tour, BigRingVR, TTS and many others. Programs have their own specialty domain: Zwift provides the possibility to train together in a virtual world, Trainer Road has structured training programs and Rouvy allows to ride in augmented reality – and of course each product also provides functionality in the competitive arena.

Trainers are provided by hardware manufacturers, like Tacx, Wahoo, Elite and others. The open ANT+ standard allows the CTP's to communicate with FE-C's.

BUT: old Tacx trainers (without inbuilt ANT+ or Bluetooth) have been left behind – the proprietary Tacx-interface is not supported by the modern CTP's and we understand that the current version of TTS does not support the old trainers anymore.

This is where FortiusAnt comes in: **FortiusAnt enables users of old Tacx Trainers (like Fortius, Magic, Flow, Vortex (earlier non-Smart version), Bushido, Genius and others) to use the latest Cycling Training Programs, [CTP] (like Zwift, Trainer Road, Rouvy, Sufferfest, Golden Cheetah, Road Grand Tour, BigRingVR, TTS and many others as mentioned above). Important: only one computer (laptop, desktop or Raspberry Pi) is required to run FortiusAnt and a [CTP].**

1.2 Thanks

FortiusAnt is based upon the methods and architecture from **[Antifier]** and uses the interface description from **[TotalReverse]** and without their work, FortiusAnt would not have existed. The same is true for Golden Cheetah for i-Magic's resistance formulas.

Thanks to **@darkpotpot** and **@iepuzaaur** for testing the i-Vortex, **@yegorvin** for testing the iMagic power curve, **@ElDonad** for cracking the CYCPLUS dongle issue and **@mattipee** for enthusiasm on programming, correctness and testing as well as valuable github- and vsc-lessons. Thanks to **@switchabl**, **@cyclingflow**, **@bikebeppe64**, **@mikeherriman**, **@mk2mark** and **@jurgen-iflow** for developing and testing the Flow/Magic trainer on T1932, **@cyclingflow** for parameterizing the runoff procedure and **@MarcoVeeneman** for the Bluetooth support. 2021 starts with the release of Bushido and Genius support, thanks to the efforts of **@switchabl**. And in 2021 **@martin-vi**, **@wtal710174**, **@decodais** and **@Meanhat** joined in making the Raspberry Pi accessible for all. And of course many thanks to all who have reacted and added to FortiusAnt improvements.

If you use FortiusAnt, let me know and add yourself to the map!

FortiusAnt has matured in the 2020's corona virus era and continues to expand in 2021 with ANT+ and Bluetooth support. It shows where github code-sharing and joint effort can bring us; a new and inspiring experience for myself.

If you use FortiusAnt, let me know and you add stars on the map! See github [issue #14](#).





1.3 The structure of this manual

After the introduction you are now reading; the manual has the following chapters

- Background information: explaining concepts
- Installation instructions: how to install FortiusAnt
- Operating instructions: how to run FortiusAnt
- FAQ and : to describe anything else
- The section on Power curve validation is moved to the end of the manual

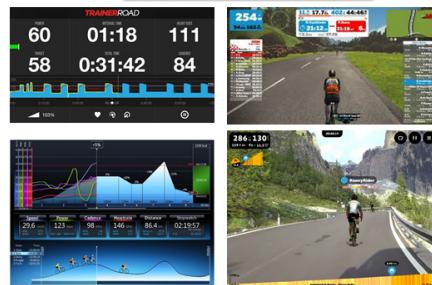
I hope the manual helps in operating FortiusAnt and I'm always happy to hear from you @github!

1.4 The story in a picture:



FortiusANT
is the communication bridge between
Tacx Fortius, Magic, Flow, Vortex
and
Zwift, Trainer Road, Rouvy

Important: only one computer (laptop or desktop)
is required to run FortiusAnt and a [CTP].





1.5 References, abbreviations, and terminology

Term	Explanation	See also
[Antifier]	The predecessor of FortiusAnt, created by "John".	https://github.com/john-38787364/antifier
[ANT+]	ANT+ is a wireless technology that allows devices to talk to each other. The following documents are most interesting to study when digging into the Python code: <ul style="list-style-type: none">• D00000652_ANT_Message_Protocol_and_Usage_Rev_5.1.pdf• D00001198_-_ANT+_Common_Data_Pages_Rev_3.1.pdf• D000001231_-_ANT+Device_Profile-Fitness_Equipment-Rev_5.0(6).pdf• D00000693 - ANT+Device Profile- Heart Rate Rev 2.1.pdf	www.thisisant.com
[BLE]	Bluetooth Low Energy	
[CTP]	Cycling Training Program, such as Zwift, Trainer Road, Rouvy, Sufferfest, Golden Cheetah, Road Grand Tour, BigRingVR, TTS and many others	
[FE]	Fitness Equipment, like legacy Tacx trainers	
[FE-C]	Controllable Fitness Equipment, the ANT+ name for an indoor trainer. A [FE] + [FortiusAnt] becomes an FE-C.	
[FortiusAnt]	FortiusAnt enables a usb-connected Tacx trainer to communicate with [CTP]s through ANT.	https://github.com/WouterJD/FortiusAnt
[HRM]	ANT+ Heart Rate Monitor profile	See [ANT+]
[PWR]	ANT+ Power Meter	See [ANT+]
[Python]	Python is a high-level programming language	www.python.com
[Raspberry]	Raspberry Pi is a series of small single-board computers	www.raspberrypi.org/downloads magpi.raspberrypi.org/books
[SCS]	ANT+ Speed and Cadence Sensor	See [ANT+]
[TotalReverse]	Invaluable source of information regarding Tacx USB interfaces.	https://github.com/totalreverse/ttyT1941
[TTS]	Tacx Training Software; proprietary [CTP] connecting to Tacx trainers only.	
[VNC]	Refer to and www.realvnc.com and www.raspberrypi.org/documentation/remote-access/vnc/ for information regarding remote access on Raspberry Pi.	



1.6 Revision History

Date of this revision: April 22nd, 2021

Version: v6.2 Published

Version	Revision Date	Summary of Changes
6.2	April 22 nd , 2021	<ul style="list-style-type: none">- Added: -I -L and -O command-line options.- -O adds a small display to the raspberry and provides additional information on the process of FortiusAnt, including differences between the different trainer types.
		<ul style="list-style-type: none">- Added: 3.2.2 "Access rights and firewall" and 3.2.3 "ANT+ or Bluetooth Low Energy?"- Leds are also implemented in the graphical user interface
6.0.e	March 17 th , 2021	<ul style="list-style-type: none">- Raspberry implementation released
6.0.d	March 13 th , 2021	<ul style="list-style-type: none">- Review @Meanhat implemented- Added: sections 2.3.1 "Tacx Catalogue 2012" and 2.6 "Tacx Trainer Power ranges".
6.0.c	March 10 th , 2021	<ul style="list-style-type: none">- Minor modifications, regarding Raspberry Pi.- Raspberry Pi installation scripts modified, Pi0 support.- Added 5.7 "TTS and FortiusANT on Dedicated USB-ports"
6.0	March 4 th , 2021	<ul style="list-style-type: none">- Section 3.12 "Raspberry Pi" added
5.2	February 11 th , 2021	<ul style="list-style-type: none">- Section 3.3 "Windows – QuickStart, using executable version." added to summarize that installing can be easy.- homeTrainer mode added
5.1	February 3 rd , 2021	<ul style="list-style-type: none">- Revision of the manual- User-interface changes effectuated- Documentation-suggestions from github- Section 3.9 "Start FortiusAnt" extended with some details on starting FortiusAnt and the location of logfiles etc.- Bluetooth added
5.1	January 8 th , 2021	<ul style="list-style-type: none">- Typo in description of -p command-line parameter.- Note added to install libusb as Administrator, if so required.- Gear switching is extended to front/rear- Power can be adjusted in ERGmode- Settings can be modified interactively and saved
5.0	January 4 th , 2021	<ul style="list-style-type: none">- Tacx Bushido and Genius support added <p>Note that the -t option for i-Vortex is changed into -t Vortex. Detailed remarks for Bushido and Genius are written in the appendices 5.14 "Detailed notes on Tacx Genius" and 5.15 "Detailed notes on Tacx Bushido".</p>
4.1.a	December 27 th , 2020	<ul style="list-style-type: none">- Start-up sequence added to 4.1 "The main functions of FortiusAnt and the head unit."
4.1	December 22 nd , 2020	<ul style="list-style-type: none">- Bluetooth Low Energy supported (-b).- ANT remote control button implemented (-C).- Main program can only closed from the GUI, so that it cannot be stopped from the head unit by mistake.- Minor functional improvements.
3.i 3.h 4.0	December 11 th , 2020	<ul style="list-style-type: none">- Supported tacx trainertypes described in more detail (see 2.3)- Power calculations implemented for magnetic brakes- Documentation improvements based upon some issues- Command line parameter -r (resistance) introduced; to send TargetPower directly to brake for test-purpose- Command line parameters -D (antDeviceID)- -G (ModifyGrade) and -c CalibrateRR introduced.- -H described with some examples- -u uphill removed (replaced by -G/0)
3.h	November 15 th , 2020	Debugging section added



Version	Revision Date	Summary of Changes
3.g	November 5 th , 2020	In manual power/grade mode, a tcx file is created. Sections added: <ul style="list-style-type: none">- 6 Power curve validation- 5.9 Fortius without cadence sensor- 5.10 Two ANTdongles – disturbed communication
3.f	October 21 st , 2020	i-Vortex and -P option added
3.e	October 16 th , 2020	Command line parameter -u uphill added Section added: 5.8 Tacx head unit with firmware to be loaded. Minor textual modifications
3.d	October 5 th , 2020	Section Requirements added
3.c	October 2 nd , 2020	Only ONE computer is required to run FortiusAnt and a [CTP], two computers are drawn in the pictures to explain the concept.
3.b	October 1 st , 2020	First version published
3.a	June, 17 th 2020	First version



2. Background information

2.1 FortiusAnt Sensor- and Monitor devices

ANT+ Master devices typically include Heartrate monitor, Powermeter, Speed- and Cadence sensor and [FE-C]'s. "Master" could be replaced by "Sensor": they measure and transmit the results through ANT+. These sensors now often communicate using BLE as well as indications suggest that BLE is becoming the more prevalent communication type. In this section (2.1) ANT+ functionality will typically also apply to BLE (but see 2.5 for more information).

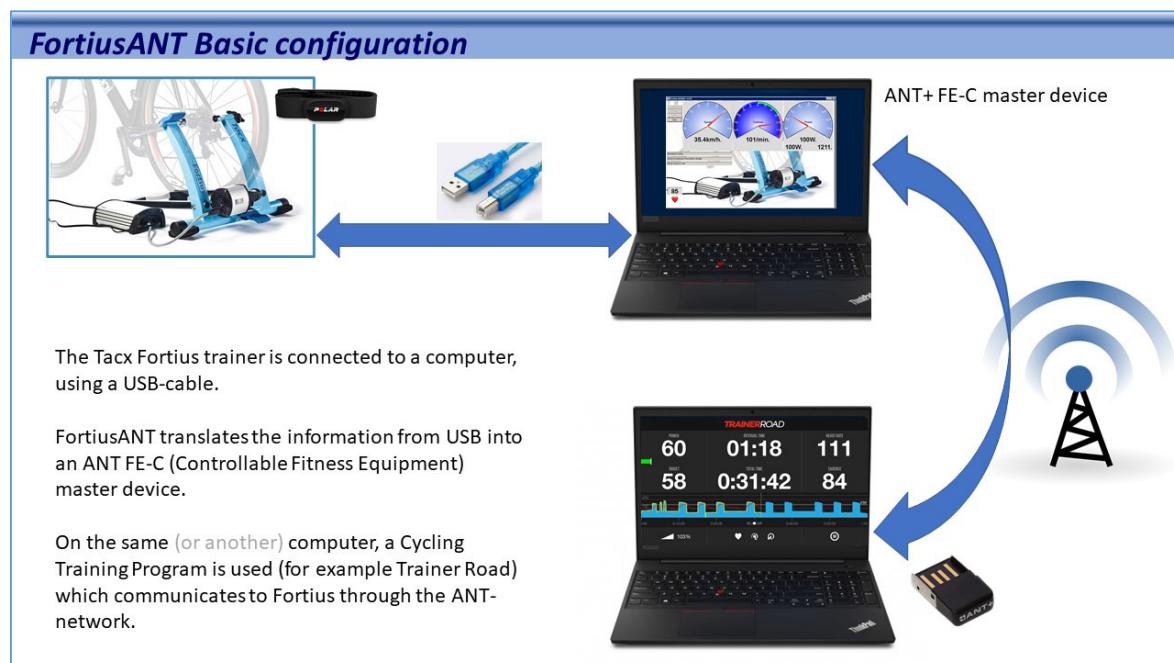
ANT+ Slave devices receive the transmitted signals and display the results, like a speed/cadence/power display on a bicycle; they may also send commands to the sensor – for example a [CTP] sends commands to a [FE-C]. "Slave" could be replaced by "Display": the display/monitor/control what the sensor does.

FortiusAnt follows ANT+ terminology also when the terms Master and Slave are replaced in future.

To understand "what we are doing"; the following pictures explain at a high level what happens "under the hood".

Important: only one computer (laptop or desktop) is required to run FortiusAnt and a [CTP]. Two computers are shown to explain the concept. One computer with three USB-connections will do the job: (1) the USB-cable to the Tacx Trainer, (2) the ANT+ dongle for FortiusAnt and (3) the ANT+ dongle for the [CTP]. And indeed: the two programs on one computer (FortiusAnt and [CTP]) communicate with each other externally and wirelessly, using two ANT+ dongles.

See also section 3.2 "Requirements".





ANT master and slave devices



ANT+ FE-C master device

ANT recognizes master- and slave-devices.
The Tacx Fortius is a master-device (broadcasting it's information),
The Cycling Training Program (listening to the master and displaying it's information) is a slave-device.

The Tacx Fortius sends four types of data on it's cable: Speed, Cadence, Power and Heartrate. All this data is broadcasted on the ANT FE-C channel.



ANT+ FE-C slave device



FortiusANT creates two ANT+ master devices (FE-C and HRM)



ANT+ FE-C master device
ANT+ HRM master device

Tacx Fortius is from before the ANT-period and integrates the heartrate monitor. This is supported by the ANT FE-C definition, but Cycling Training Programs like Trainer Road and Zwift expect independent HRM's.

For that reason, FortiusANT creates two ANT+ master-devices: FE-C and HRM.



ANT+ FE-C and HRM slave device





FortiusANT can also listen to an independent ANT+ HRM strap



Although the basic purpose of FortiusANT is to translate Fortius to ANT+ FE-C, the following option is implemented. A little gadget in fact.

Imagine, you use an ANT+ coded Heartrate monitor; the Tacx Fortius does not recognize the HRM and no data can be displayed or broadcasted.

Therefore the `-H` flag tells FortiusANT to listen to a master HRM-device and display the heartrate. In that case, no master ANT+ HRM device is created.



Zwift works slightly differently



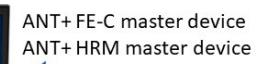
As explained, FortiusANT creates two master devices



ZWIFT shows the following devices that can be paired individually:

- ANT+ Heartrate monitor
- ANT+ Controller
- ANT+ Cadence sensor
- ANT+ Power meter

The latter three can all be paired to the FortiusANT FE-C device, but if your bicycle has an own powermeter, you could choose to connect to that device instead.





2.2 Description

A [CTP] (Cycling Training Program) send commands to the [FE-C] (Controllable Fitness Equipment) through [ANT+] (or [BLE]). There are two modes: Power- or ERG-mode and Resistance- or Slope-mode.

2.2.1 Resistance

It is important to understand that an [FE-C] only understands 'resistance' which is the torque required to turn the brake/motor-brake. From physics we know that power (Watt) is proportional to torque (Nm) * rotational speed (rpm). For a given resistance, the power required is linear with the speed (=cadence), provided you do not change gears. Also, for a given resistance and cadence, the power required is linear with the gear-ratio.

PS. This is especially valid for the older trainers; Tacx i-Vortex (T1960/T1961) can be set natively to a defined Power. (Note that there are also Vortex Smart Trainers, such as T2180).

2.2.2 Power- or Ergo-mode

In Powermode the [CTP] sends the required power to the [FE-C] and regardless gear or cadence, the requested power is constant. FortiusAnt calculates the resistance = power/speed (with some constants applied). Note that, if you change gears and/or cadence, the required power will remain equal because the resistance is adjusted.

2.2.3 Resistance mode

In resistance mode, FortiusAnt receives the required grade from [CTP]. Grade may vary from -20% (downhill) to +20% (uphill). FortiusAnt calculates the resistance to be sent to the [FE-C].

The power required to ride up a hill is based upon the given grade with a mass of 85 kg (rider + bike) at a given speed'. Input parameters are grade and mass (from [CTP]) and speed (as measured by [FE-C]). The result is Power and conversion to resistance is described above.



2.3 Tacx trainers

2.3.1 Tacx Catalogue 2012

In the catalogue of Tacx we find the following classes and trainers:

			FortiusAnt
Cycle trainers			Use magnetic brakes that can be adjusted mechanically and therefore it is not possible to connect with software.
Booster	Magnetic brake with 8 permanent magnets.		No
Satori	Magnetic brake with 1 powerful Neodymium magnet.		No
Sirius	Magnetic brake with 6 permanent magnets.		No
Ergo trainers			Use a brake that can be adjusted from a head unit.
Bushido	Electro brake without motor for descents; wireless		-t Bushido
i-Vortex	Electro brake with 8 electromagnets; wireless		-t Vortex
Vortex	Electro brake with 8 electromagnets		USB
Flow	Electro brake with 6 electromagnets		USB
Magic	Electro brake with 6 electromagnets (legacy head unit)		USB
Virtual reality trainers MP=Multiplayer	Genius MP	Motor brake (driven for descents); wireless	-t Genius
	Fortius MP	Motor brake (driven for descents)	USB
	Flow MP	Electro brake with 6 electromagnets	USB

Ergo trainers were standalone trainers and MultiPlayer trainers were sold with a software package to be installed on a PC (ANT-dongle where required). Ergo could be upgraded to MultiPlayer. Hardware-wise there is no difference.

2.3.2 General

We already saw in the previous section that Tacx has created a large variety of trainers with commercial names like Magic, Flow, Fortius, Vortex, Bushido, Genius, with or without i-. Technically, you will find three numbers: one for the configuration, one for the head unit, one for the brake and then for other optional equipment parts (such as the steering unit). For more information see [[TotalReverse](#)], “Tacx product number overview” where the T-numbers describe the part they represent.

FortiusAnt is developed for the configuration I own: a **T1930** Tacx Fortius Multiplayer bundle with **T1941** brake, **T1932** PC-head unit and **T1905** steering unit. Then FortiusAnt is extended with support for the other models. This explains the name of the product.

FortiusAnt converts the USB-interface to an ANT+ or BLE interface. In this USB-mode, FortiusAnt only “knows” the head unit, not what “commercial name” is used. The head unit comes in two flavors: T1902 head unit (the so-called USB legacy interface) and all other USB head units. FortiusAnt also knows the type of brake used: magnetic or motor and adjusts automatically to the required characteristics.

Bundle	Brake	Head unit	Remarks
T1930 Tacx Fortius	T1941 Motor	T1932	USB interface tested, see 6.2
T1900 Tacx i-Magic	T1901 Magnetic	T1902	Legacy USB interface tested, see 6.3
T2250 Tacx Flow	T1901 Magnetic	T1932	Magnetic Brake interface tested, see 6.4. Issues: #102 #128 #143 #153

When directed to do so (using the -t flag), FortiusAnt does not look for a USB-trainer, but tries to pair with a proprietary ANT+ Tacx trainer. FortiusAnt then communicates with the brake and optionally with the



head unit. Compatibility between Tacx trainers and head units can be found in the product overview referenced above.

FortiusAnt supports the following trainers, connecting to the **USB head unit**:

- Head unit T1902: Old "solid green" iMagic head unit (with or without firmware)
This head unit uses a so-called legacy-USB protocol, the others the New-USB-protocol.
- Head unit T1904: New "white, green" iMagic head unit (firmware inside)
- Head unit T1932: New "white, blue" Fortius head unit (firmware inside)
- Head unit T1942: Old "solid blue" Fortius (firmware inside)
- Head unit 0xe6be: Old "solid blue" Fortius (without firmware)
This head unit requires software to be loaded when FortiusAnt is started.
See also section 5.8 "Tacx head unit with firmware to be loaded".

FortiusAnt supports the following trainers (-t flag), connecting to an **ANT+ brake**:

- -t Vortex: T1961 brake and T2172 head unit are supported (see issue #46) (not to be confused with the T2180 Vortex Smart Trainer)
- -t Bushido: Head unit T1982 is required (see issue #117)
- -t Genius: Head unit T2020 is optional (see issue #101)

2.3.3 Tacx proprietary ANT trainers

The ANT-interface was initially used to connect a heartrate monitor to a watch and from there developed into a low-energy wireless protocol in the sport-environment. For more info, refer to www.thisisant.com.

When the whole world went wireless, the USB-connection became obsolete. ANT+ enabled trainers to connect with a variety of computers: Windows, tablet, smartphone, etc. Tacx followed and built trainers with an ANT+ connection but decided to keep the interface between trainer and software proprietary.

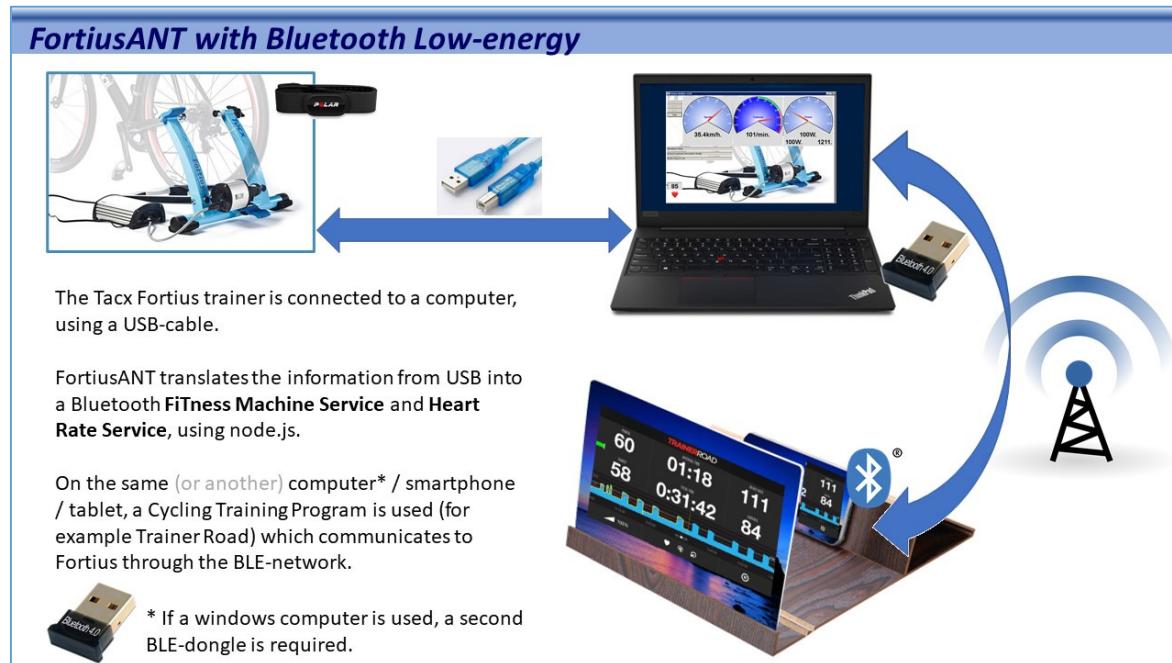
And hence Bushido, Genius and Vortex were created, communicating with an ANT+ protocol, but not following the standard regarding transmission details and data. (And perhaps, giving some credit to Tacx, the standard followed the proprietary Tacx-developments).

FortiusAnt can communicate with those trainers. Functionally you can see that instead of a USB-cable, the ANT-dongle is used. Otherwise there are no functional differences.





2.4 Bluetooth Low-Energy (BLE)



2.4.1 Introduction

ANT+ is designed as low-energy protocol so that sensors can communicate with a sports-watch. “Bluetooth Low Energy” [BLE] can simply be seen as an alternative protocol. Many smartphones support BLE but not ANT and ANT-dongles are often hard to find and tend to be more expensive than BLE-dongles. Sensors (power-meters and FE-C’s) tend to support both protocols and therefore there are many reasons to use Bluetooth Low Energy.

Unfortunately, there is no direct connection available between FortiusAnt and an installed BLEdongle; and a Node.js server is required to make the connection.

FortiusANT was originally designed to provide an ANT+ interface to a Tacx Fortius. This requires the user to use ANT+ dongles in order to connect to applications such as Zwift, Trainer Road, Rouvy, Sufferfest, Golden Cheetah, Road Grand Tour, BigRingVR, TTS and many others.

At the request of several users of FortiusANT support for BLE (Bluetooth Low Energy) has been added. When using the BLE interface the ANT+ dongle is not required anymore so long as you have supported BLE hardware.

If you have a tacx-proprietary ANT trainer (Bushido, Genius, Vortex), want to pair with an ANT+ heartrate monitor or transmit power data to a Garmin Edge (power- or speed/cadence sensor) then the ANT-dongle is still required.

2.4.2 Design

The BLE support for FortiusANT is implemented in Node.js, unlike FortiusANT itself which is written in Python. The implementation makes use of the well established Bluetooth LE library see abandonware/bleno on <https://github.com/abandonware/bleno>.

Using this library FortiusANT advertises the following services which can be discovered:

- FTMS (Fitness Machine Service)
- HRS (Heart Rate Service)

Communication between FortiusANT and the BLE server happens internally via a local http server where FortiusANT acts as the client and the BLE server as the server.



2.5 Calculations

2.5.1 Power mode

When the [CTP] is in power-mode, a required number of Watts is sent to [FE-C], in our case FortiusAnt. The [FE] requires a resistance to be set and hence a function TargetPower2Resistance(Power, Speed) is used to convert this. The function is different for legacy- and new-USB trainers.

So, if you want to ride with a power of 100Watt and the bicycle wheel runs at 10km/h, the [FE] needs to receive a different resistance than when the wheel is rotating at 40km/h.

Similarly, the [FE] returns the current actual resistance and a function is used to calculate the corresponding Actual Power.

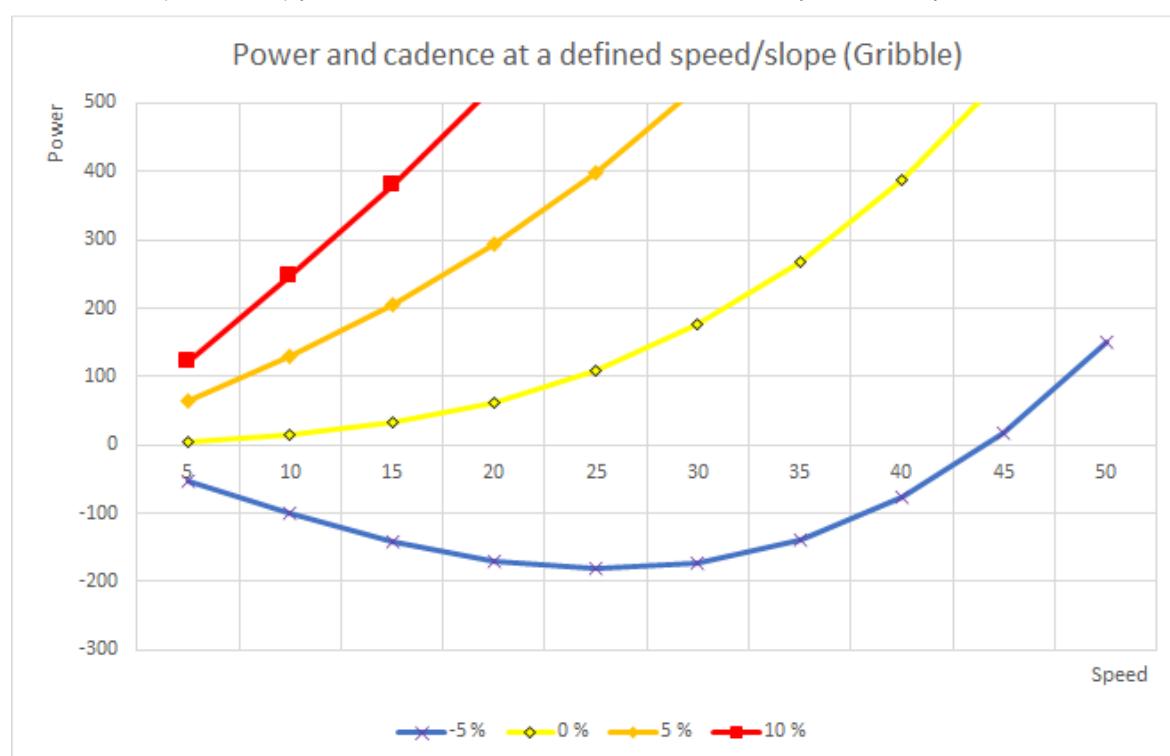
2.5.2 Grade mode

In Grade mode, the [CTP] communicates the slope where you are riding: flat = 0%, up hill (e.g. 10%) or downhill (e.g. -10%).

An additional step is required, using function Grade2Power(Grade, Speed, UserAndBikeWeight). First the Grade is converted to power and then the same applies as described in the previous paragraph.

The function also depends on RollingResistance, WindResistance, WindSpeed and DraftingFactor. Constants are used if [CTP] does not explicitly specify these parameters. The default value for UserAndBikeWeight is 85kg.

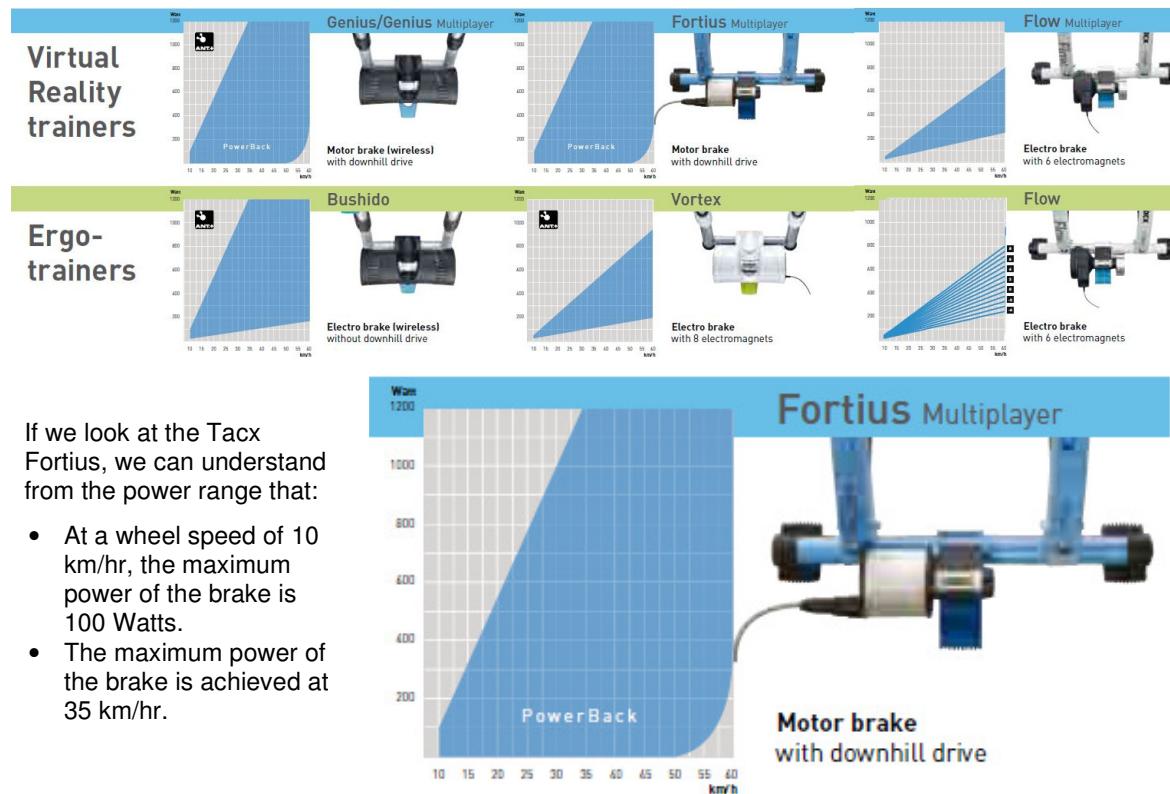
The formula results in the power curve shown below. Note that on a flat surface (yellow curve) you need 100Watt to ride at 25km/hr. On a hill of 5% (orange curve) you need 300Watts at 20km/hr. And if you ride downhill -5% (blue curve) you would ride at around 44 km/hr without any additional power.





2.6 Tacx Trainer Power ranges.

In the Tacx documentation, we find that the different trainers have different operational zones:



If we look at the Tacx Fortius, we can understand from the power range that:

- At a wheel speed of 10 km/hr, the maximum power of the brake is 100 Watts.
- The maximum power of the brake is achieved at 35 km/hr.

Where the Tacx trainers were used for Ergo-training, this was not a problem. The requested power was adjusted on the head unit, the speed realized with the bike and the two factors (speed and power) were not related.

In Virtual Reality (e.g., using Zwift or Rouvy), there is an issue however:

- If you ride uphill 10% at 10km/hr, 250Watt is required. (see graph on previous page)
- But Tacx Fortius cannot realize a required power of 250Watt at that wheel-speed!

Tacx Fortius will not run smooth and does not provide an optimal experience. As rider you have no options here:

- If you change gears to a lighter ratio, the speed will go down and the issue will get worse.
- If you ride faster to increase wheel-speed, perhaps the Tacx Fortius will get into a more workable range – but also the required power goes up (you need 380 Watts at 15km/hr)

Conclusion: With a calculated relation between Grade and Wheel-speed, at 10 km/hr and max 100 Watts, Tacx Fortius can do a slope of max 4%.

FortiusAnt provides the following solutions for this issue:

- Use the -G50 option; when a CTP requires 10%, a slope of 5% is used for the required power calculation. This is a static setting, however. (see 4.2 "Command line")
- Use the digital gearbox: use the head unit to "change gears" (changing ratio); where the default is 50*19 and you change to 34*19, a slope of 10% is reduced to 7%. (see 4.1.6 "Buttons on the Tacx head unit").

The different brakes have different characteristics, the story remains the same: try to ride at a high wheel-speed and reduce required power using the "digital gearbox". Fortius will adjust the displayed speed accordingly.

Note that CTP's in GradeMode, usually calculate the speed from the Grade/Power, using an inversed calculation as used in the previous section. Their speed will differ from the displayed speed in FortiusAnt.



3. Installation instructions

3.1 Introduction

FortiusAnt is written in **[Python]** and can be run on any computer for which a Python runtime environment exists. For Windows, FortiusAnt is also available as an executable, containing the Python runtime environment. Windows users can therefore decide which version to use. Implementation on a Raspberry Pi is shown separately in 3.12.

FortiusAnt communicates with the Tacx Trainer through a USB-interface, which needs some special care. And in the other direction, FortiusAnt communicates with a **[CTP]** using an ANTdongle or BLE.

3.2 Requirements

3.2.1 You, the Tacx Athlete

Getting the Tacx connected to Zwift, Trainer Road, Rouvy, Sufferfest, Golden Cheetah, Road Grand Tour, BigRingVR, TTS and many others – **[CTP]** in short – needs some basic knowledge. You should be able to:

- Download documentation and software from <https://github.com/WouterJD/FortiusAnt/>
- Connect Tacx and ANT+/BLE dongles.
- Install drivers on your computer.
- Start an executable
- Understand basic concepts, this manual only explains the FortiusAnt specifics.

If this is complex ask a friend to assist! However, don't feel too intimidated and read section 3.3 "Windows – QuickStart, using executable version." first.

3.2.2 Access rights and firewall

To be able to install, you will usually need administrative rights.

Also mind the firewall blocking access until you grant it; FortiusAnt, NodeJs and your CTP's will access devices, the firewall may control.

3.2.3 ANT+ or Bluetooth Low Energy?

The following considerations must be taken into account to choose for ANT or BLE:

- If you use a USB-connected Tacx trainer, you can use ANT or BLE to communicate with the **[CTP]** of your choice. First check whether ANT and/or BLE is supported before you buy.
- If you use an ANT-connected Tacx Trainer, you need at least an ANT-dongle. If your **[CTP]** supports ANT+ an ANT-dongle may be the obvious choice.
- It seems that the market for trainers is moving from ANT to BLE, but currently usually both are supported. Some **[CTP]**s support BLE before ANT, but eventually support both.
- And finally: ANT-dongles are harder to get and more expensive than BLE-dongles.



3.2.4 Hardware

Before installing, decide what hardware you want to use:

Tacx trainer	Hardware	Connections	Software	Description
USB connected: Fortius, Flow, Magic...	1 One computer ¹⁾	USB 1a - Tacx Trainer	FortiusAnt	FortiusAnt communicates with Tacx Trainer (USB 1a) and with [CTP] (USB 1b ↔ USB 1c)
		USB 1b - ANT dongle 1		
		USB 1c - ANT dongle 2	[CTP]	
	2 Two computers ¹⁾	USB 1a - Tacx Trainer	FortiusAnt	FortiusAnt communicates with Tacx Trainer (USB 1a) and with [CTP] (USB 1b ↔ USB 2a)
		USB 1b - ANT dongle 1		
	3 One computer ¹⁾ One smartphone ²⁾	USB 2a - ANT dongle 2	[CTP]	FortiusAnt communicates with Tacx Trainer (USB 1a) and with [CTP] (USB 1b ↔ ANT interface)
		USB 1a - Tacx Trainer	FortiusAnt	
		USB 1b - ANT dongle 1	[CTP]	
ANT+ connected: Bushido, Genius or Vortex	4 One computer ¹⁾ ³⁾ ⁴⁾ One smartphone ²⁾	ANT interface	[CTP]	FortiusAnt communicates with Tacx Trainer (USB 1a) and with [CTP] (USB 1b ↔ BLE interface)
		USB 1a - Tacx Trainer	FortiusAnt	
		USB 1b - BLE dongle 1		
	5 One computer ¹⁾	BLE interface	[CTP]	FortiusAnt communicates with Tacx Trainer (USB 1a) and with [CTP] (USB 1a ↔ USB 1b)
		USB 1a - ANT dongle 1	FortiusAnt	
		USB 1b - ANT dongle 2	[CTP]	
	6 Two computers ¹⁾	USB 1a - ANT dongle 1	FortiusAnt	FortiusAnt communicates with Tacx Trainer (USB 1a) and with [CTP] (USB 1a ↔ USB 2a)
		USB 2a - ANT dongle 2	[CTP]	
	7 One computer ¹⁾ One smartphone ²⁾	USB 1a - ANT dongle 1	FortiusAnt	FortiusAnt communicates with Tacx Trainer (USB 1a) and with [CTP] (USB 1a ↔ ANT interface)
		ANT interface	[CTP]	
	8 One computer ¹⁾ ³⁾ ⁴⁾ One smartphone ²⁾	USB 1a - ANT dongle 1	FortiusAnt	FortiusAnt communicates with Tacx Trainer (USB 1a) and with [CTP] (USB 1b ↔ BLE interface)
		USB 1b - BLE dongle 1		
		BLE interface	[CTP]	

¹⁾ Computer: May be running Windows, Linux, MacOS or Raspberry.

²⁾ Smartphone: or tablet.

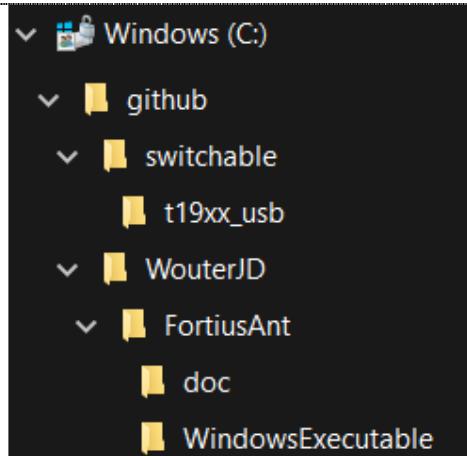
³⁾ FortiusAnt cannot receive heartrate from HRM since BLE/HRM is not supported.

⁴⁾ If the computer is a **Raspberry Pi**, the BLE-interface is onboard and no dongle is required.

3.3 Windows – QuickStart, using executable version.

This section describes a quick-start option, explaining the basic steps required to set-up a Windows-environment.

1. On your computer, create the following folders:



2. Download three files:

The manual: <https://github.com/WouterJD/FortiusANT/blob/master/doc/FortiusANTUserManual.pdf>

USB drivers: https://github.com/switchabl/t19xx_usb/releases/download/v1/libusb-win32_T19xx.exe

Executable: <https://github.com/WouterJD/FortiusANT/blob/master/WindowsExecutable/FortiusANT.exe>

and move these files into the folders doc, t19xx_usb and WindowsExecutable respectively.



3. Install the USB drivers (only if your Tacx trainer has a USB-head unit)	Execute libusb-win32_T19xx.exe as Administrator, click "Next" and then "Finish".
4. Insert the ANT-dongle in a USB-port	Install the drivers as requested by the ANT dongle
5. Start FortiusAnt	Double-click FortiusAnt.exe

FortiusAnt is now ready for use; proceed with section 3.9 “Start FortiusAnt”.

More detailed installation instructions are in the next sections. But you can choose to skip them.

3.4 Download FortiusAnt from github

Goto <https://github.com/WouterJD/FortiusAnt/>

In section <code> click on [↓Code] and download the code as a zipfile.

Create a folder on your computer called C:\Github\FortiusAnt (windows) or .../Github/FortiusAnt (unix). From now on, this folder will be referred to <**the FortiusAnt folder**>.

The downloaded file contains a folder called **FortiusAnt-master**. Copy the contents into <**the FortiusAnt folder**>. <**the FortiusAnt folder**> now contains sub-folders **pythoncode**, **WindowsExecutable**, etc.

3.5 Install Python

Note: Python is NOT required when you use the Windows executable.

- Goto <https://www.python.org/downloads/>
- Follow the installation instructions for your system. It's impossible to handle all operating systems specific instructions. For Windows it's easiest NOT to install for "All users".
- Noted by @msjnaessens: "If you installed Python 3.9, you cannot install numpy. I chose to install Python version 3.7.9 instead."

Note that, Python version 3.x.x is required. If you have multiple instances of Python installed, mind the PATH settings!

If the Python path is NOT specified there, next commands may give the error “command not found”. On Windows the PATH settings can be checked using Computer, Properties, Advanced, Environment variables.

After installation, the Python version can be checked with the following commands:

```
python --version  
pip --version
```

You can check whether the most recent pip is installed with the command:

```
pip install --upgrade pip
```

As soon as Python is installed correctly, the modules that are required for FortiusAnt can be installed as follows:

```
pip install -r requirements.txt
```

or, if you do not intend to use the Graphical user interface (e.g. on Raspberry Pi0):

```
pip install -r requirementsNoGUI.txt
```

the requirements file is available in <**the FortiusAnt folder**>/pythoncode.



3.6 Install USB-driver

3.6.1 Windows – prepackaged drivers

The installation from libusb is explained in the next paragraph. For FortiusAnt a prepackaged installation is available:

Goto https://github.com/switchabl/t19xx_usb

And follow the installation steps as explained there:

- Download the all-in-one package from https://github.com/switchabl/t19xx_usb/releases/download/v1/libusb-win32_T19xx.exe
- Run libusb-win32_T19xx.exe as Administrator.
- Click "Next", then "Finish".

3.6.2 Windows – from libusb

You can skip this step if you have installed the prepackaged drivers from @switchabl as explained in the previous section.

On the system where FortiusAnt is running, it is recommended that TTS is not installed since the two programs require different USB-drivers which may conflict. There are studies where both programs are installed but, for sake of simplicity of this installation instruction, I will describe a FortiusAnt-only installation. **Refer to** 5.6 “Can TTS4 and FortiusAnt coexist?” for more information!

You have to (re)install your trainer as a libusb-win32 device.

Download the libusb driver

- Download software from <https://sourceforge.net/projects/libusb-win32/>
- Read the wiki, sections download and Device Driver Installation
- Download libusb-win32-bin-1.2.6.0.zip from files/libusb-win32-releases/1.2.6.0/
- Unzip the file
- Note: when Windows does not allow to install, installing the libusb driver as Administrator (rightclick, Run as Administrator when adding) may help.

Uninstall TTS-driver (refer to **section** 5.6 “Can TTS4 and FortiusAnt coexist?” first)

- Open Device Manager.
- Right click on the device and click "Uninstall". It may be listed as a "Jungo" device (see <http://www.tacxdata.com/files/support/Windows10driverissues.pdf> - DO NOT RUN TacxDriversSetup.exe!)
- Unplug the trainer, wait 5 seconds, and plug it back in again

Install option 1

- Find it again (usually under other devices>VR-interface)
- Right click and select "update driver software"
- Select "Browse my computer for driver software"
- Select "Let me select from a list of device drivers on my computer"
- Select libusb-win32 devices
- Select ANT USB Stick 2, then OK in the warning, then close Your USB-trainer is now installed as "ANT USB Stick 2" which works, but perhaps is not a very clear name.

Install option 2

- Go to the ..\libusb-win32-bin-1.2.6.0\bin folder
- Start inf-wizard.exe, click next
- Select "VR-Interface" (which is the USB-device you plugged in again), click next
- Specify the name for Manufacturer (Tacx) and Device (VR-Interface), click next
- Store the results in the same folder; this creates a file like VR-Interface.inf
- Complete the installation



Now the USB-trainer is installed as "VR-Interface" (you could have chosen for "Fortius Virtual Trainer").

Technically, it is the same as option 1, but especially if you have two ANT sticks and one Tacx Trainer a named device is clearer. The names will only show up in Device Manager and ExplorANT and have little significance for the end-user.

3.6.3 MacOS

Hints:

- libusb can be installed using brew install libusb.
- Get brew if you don't have it already: <https://brew.sh/>.

3.6.4 Linux – General

Hints:

- Root required
- Refer to AntBridge installation instructions for hints (<https://github.com/pepelkod/AntBridge>).
- Instructions supplied by FortiusAnt users, since I have no Linux environment to test.

3.6.5 Linux Ubuntu 20.04

As provided by [@msjnaessens](#); thanks.

Fresh install of Ubuntu 20.04; installed in Oracle VM VirtualBox 6.1; 2048 MB ram; 4 cpu cores (i7 8750H); 16 MB video memory; installed on MSI GV62 8RC laptop.

```
sudo apt install git
git clone https://github.com/WouterJD/FortiusAnt
sudo apt upgrade python3
sudo apt install python3-pip
pip3 install --upgrade pip
sudo apt-get install python3-pygame
```

```
sudo apt install make gcc libgtk-3-dev libgstreamer-glib1.0-0 freeglut3 freeglut3-dev python3-gst-1.0 libglib2.0-dev ubuntu-restricted-extras libgstreamer-plugins-base1.0-dev ubuntu-dev-tools
```

```
sudo apt install python3-wxgtk4.0
pip3 install -r ./FortiusAnt/pythononcode/requirements.txt
pip3 install --upgrade wxpython
```

```
git clone https://github.com/pepelkod/AntBridge
LOC=/lib/modules/`uname -r`/kernel/drivers/usb/serial/
sudo mv $LOC/usb-serial-simple.ko ~/Documents
sudo mv $LOC/usbserial.ko ~/Documents
sudo rmmod usb_serial_simple usbserial
sudo apt install libusb-dev
sudo apt install libgoogle-glog-dev
sudo apt install libusb-1.0-0-dev
sudo apt install pkg-config
```

Open terminal in AntBridge folder

make; make;

```
sudo make install
```

```
git clone https://github.com/Tigge/openant
git clone https://github.com/Tigge/antfs-cli
```



```
open terminal in openant folder:  
sudo python3 setup.py install  
open terminal in antfs-cli folder  
sudo python3 setup.py install
```

Now run FortiusAnt:

```
sudo python3 ./FortiusAnt/pythoncode/FortiusAnt.py
```

Done!

3.7 Install ANTdongle

ANTdongles are a lot easier to use than the Tacx-USB-interface, since they are plug&play; insert the dongle in your computer and the required drivers will be installed automatically.

Dongles from manufacturer=CYCPLUS are reported not to work well with FortiusAnt; refer to github FortiusAnt issues (#61, #45 and #65).

As you are using ANT+ rather than Bluetooth please proceed to section 3.9.

3.8 Using Bluetooth Low Energy (BLE)

Bluetooth is a short-range wireless communication technology standard

3.8.1 Supported Hardware

Since BLE support in FortiusANT depends on the bleno library, hardware support is also limited to what bleno supports.

On macOS, on-board Bluetooth is used, no need for an external dongle.

On Windows a Bluetooth dongle is required; there is a limited set of supported hardware. It is important that your Bluetooth dongle has one of the supported chipsets. See node-bluetooth-hci-socket on <https://github.com/noble/node-bluetooth-hci-socket#windows>.

3.8.2 Installation on macOS

- Install Xcode: App Store see <https://apps.apple.com/nl/app/xcode/id497799835?l=en&mt=12>
- Install NodeJS: brew install node
- Install dependencies: cd <the FortiusANT folder>\node and npm install

3.8.3 Installation on Windows

- Install Git for Windows (<https://git-scm.com/downloads>); only needed if not installed yet.
- Install NodeJS LTS version (<https://nodejs.org>)
 - During installation: **check the box** which installs the necessary tools for native modules.
 - After NodeJS installation completes, a command prompt will appear which will install the necessary tools (Python, Visual Studio and - Build Tools, Chocolatey and some Windows upgrades with a total size of approx. 3Gb). This will take a while, grab a drink in the meantime. Note that, manual installation of the components is complex and therefore not recommended.
- Install Zadig (<https://zadig.akeo.ie>)
- Insert the Bluetooth dongle and replace the driver for your Bluetooth dongle using Zadig.

Note that you cannot use the Bluetooth dongle for Windows itself when you perform this step. Using the exact same steps as mentioned below you can restore the old driver if you want.

1. Start Zadig
2. Select options, list all devices



3. Select the Bluetooth dongle; **double-check that the correct driver is chosen.**
Note: It may be difficult to know which device is the correct BLE dongle in case your machine also has BLE on-board. Disable the on-board BLE device before inserting the BLE dongle so Zadig will see only one.
 4. Remember the current driver, in case you want to restore the driver later on.
 5. Check that WinUSB driver is set as target driver, this should be the default. (Choose the old driver when reverting)
 6. press Replace Driver.
- Install dependencies
 1. Start the Windows command prompt and type: cmd <enter>
 2. Go to the FortiusANT folder and type: cd <the FortiusANT folder>\node <enter>
 3. Install node.js and type: npm install <enter>

3.8.4 Run FortiusANT with BLE support

To start FortiusAnt with BLE support add the ` -b` option.

When [Start] is pressed the BLE interface will be started until [Stop] is pressed. FortiusANT will start advertising as 'FortiusANT Trainer' on Windows and Linux systems. On macOS, it will start advertising as your computer name.

A [CTP] can then connect to FortiusAnt in the usual way, selecting the Bluetooth device.

3.9 Start FortiusAnt

Now FortiusAnt is ready for use. To start (or run) FortiusAnt:

- Double-click FortiusAnt.exe from Windows explorer (compiled version)
- Double-click FortiusAnt.py from Windows explorer (Python version)
- Execute the FortiusAnt.exe or FortiusAnt.py command from the command-prompt
- Create a script, command-file, shortcut or menu-entry to do the same
- Use any of the sample-scripts in <the FortiusAnt folder>/StartUp
Note that these are called xxx.bat for usage in the Windows-environment.
When made executable, the same scripts can be used in MacOS or Linux environment (after making them executable).

Windows, Linux, MacOS all use the same concept of giving a command, but in slightly different ways. Further explanation is beyond the scope of this manual.

If you start FortiusAnt without additional command-line options, FortiusAnt will use the default (best-practice) settings.

After starting FortiusAnt using one of the described methods, a "console" is opened and then the FortiusAnt user-interface is started. The pause command in the sample scripts is present, so you can see messages in the "console" in case of an abnormal end.

See section 4.2 "Command line" for information on parameters that can be passed on the command-line.

Note that logfiles, training-files and data-files (.log,.GUI.log, .tcx or .json) are created in the 'current directory' where the FortiusAnt is started (not where it is located!).

Therefore, mind the difference between

```
cd <the FortiusAnt folder>/pythoncode  
FortiusAnt.py -a -g
```

and

```
<the FortiusAnt folder>/StartUp  
./pythoncode/FortiusAnt.py -a -g
```

In the first example, the files are created/expected in the pythoncode directory, in the second example those files are created/expected in the StartUp directory.



3.10 Settings in the JSON-file

Instead of using a command-line to start FortiusAnt, settings can also be stored in a JSON-file; a sample is available in folder “<the FortiusAnt folder>/StartUp Json”.

The settings in FortiusAntSettings.json may be modified with an editor or using FortiusAnt (click the settings button).

3.11 Check FortiusAnt

When FortiusAnt is started without additional command-line parameters, the following text is displayed in the console:

```
Hello!  
You have started FortiusAnt without command-line parameters.
```

```
Therefore, we start with a best practice setting:  
FortiusAnt.exe -a -g -H0 -A
```

```
If you want to start without the graphical user interface:  
FortiusAnt.exe -a
```

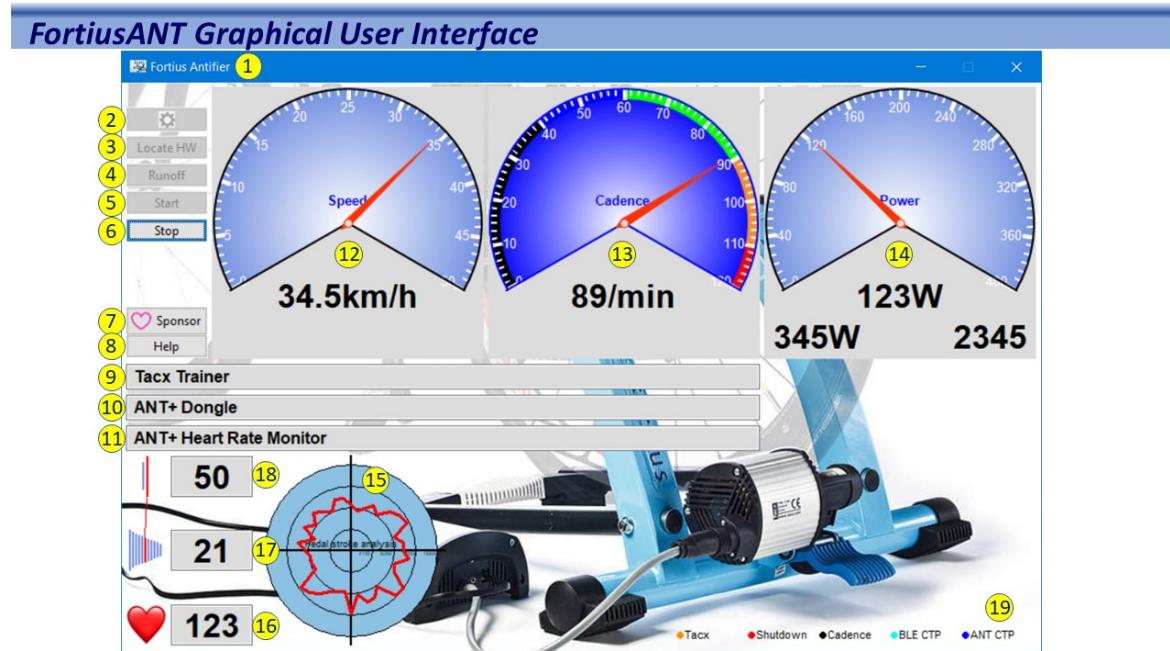
```
For more info, please refer to the wiki on github.  
Success!
```

```
FortiusAnt is open source and can be used freely.
```

```
Just for the fun of knowing where you all are training,  
put yourself on the FortiusANT map by making yourself known  
by leaving a message with name/location/trainer on  
https://github.com/WouterJD/FortiusANT/issues/14
```

```
or visit the sponsoring page https://github.com/sponsors/WouterJD
```

And then the graphical user interface appears:



When the GUI is shown you know that FortiusAnt is installed correctly.



The user interface contains the following elements:

1. The window title displays name and version of the software you run and the actual version as available on github (excluded from the images in the manual).
2. <Settings> is a button to open a dialogue to modify and/or save the FortiusAnt settings.
3. <Locate HW> is a button and when pressed FortiusAnt will search for a Tacx USB-device and an ANT-dongle. When found, the result is displayed in (5) and (6), the button is disabled and <Runoff> and <Start> are enabled.
4. <Runoff> activates the user-driven calibration
5. <Start> activates FortiusAnt to bridge USB data to ANT+ and vice-versa; then <Start> is disabled and <Stop> enabled.
6. <Stop> stops the FortiusAnt bridge.
7. <Sponsor> opens the sponsoring page on github.
8. <Help> opens the manual on github.
9. Shows what USB-trainer is found
10. Shows what ANT-dongle is found (or Bluetooth if using BLE)
11. Shows what heartrate is used
12. Displays the speed of the bicycle wheel (returned by the Tacx trainer)
13. Displays the cadence of the pedals (returned by the Tacx trainer). Note that the calibration progress is also shown here during calibration.
14. Displays the power as returned by the Tacx trainer.
Also, the target is displayed as requested by the CTP.
15. Displays the Pedal Stroke Analysis, as calculated by FortiusAnt.
16. Displays heartrate.
17. and 18 show the virtual gearbox.
- 18.
19. The status LEDs:
 - **Led Tacx** blinks when information is received from the Tacx trainer. Blinking does not mean that the Tacx is working well; bad connections cannot be detected through the head unit.
 - **Led Shutdown** is used by the shutdown button (raspberry only).
 - **Led Cadence** blinks when the pedal passes the magnet (USB trainers only).
 - **Led BLE** blinks when a Bluetooth message is received from a **[CTP]**.
 - **Led ANT** blinks when an ANT message is received from a **[CTP]**.
 - **Led Tacx + ANT + BLE + Cadence** blink together when FortiusAnt is waiting for a pedal-kick to start calibrating.
 - **Led Tacx + ANT + BLE** blink together when FortiusAnt is calibrating.



3.12 Raspberry Pi

FortiusANT on Raspberry with Bluetooth Low-energy

The Tacx Fortius trainer is connected to a Raspberry, using a USB-cable.

FortiusANT translates the information from USB into a Bluetooth **Fitness Machine Service** and **Heart Rate Service**, using node.js.

On another computer* / smartphone / tablet, a Cycling Training Program is used (for example Trainer Road) which communicates to Fortius through the BLE-network.

* If a windows computer is used, a second BLE-dongle is required.

Many of us are familiar with Windows and Apple devices and the installation instructions are described in the previous sections. www.raspberrypi.org explains that:

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing and to learn how to program in languages like Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.



What's more ... it can run FortiusAnt!

Instead of adding (two) ANT-dongles to run a CTP on a computer (which ties that computer to running FortiusAnt), a Raspberry Pi can be used instead with one ANT dongle or built-in Bluetooth and then the CTP can be run from a smartphone or tablet. If required, the smartphone can then be connected to or cast to a TV or Monitor.

This section explains how to order and install a Raspberry.

This section is made possible by **@martin-vi** and **@wtal710174** for the detailed installation instructions and **@switchabl** for the valuable suggestions, **@decodais** for programming the LEDs and the feedback from **@MeanHat**.

3.12.1 Required hardware

I have started with below configuration.

- Raspberry Pi 4 Model B / 4GB.
- 3A USB-C power supply.
- SanDisk Ultra Micro SDHC 32GB 80mb/s.
- Oval Case with fan and Copper Heatsink.

Additionally, you need: USB keyboard and mouse and HDMI display.



Notes:

- FortiusAnt is reported running on Raspberry Pi0W by [@Meanhat](#) and Raspberry Pi3 by [@Decodais!](#)
- It may be convenient to buy a so-called starters-kit, especially because it helps you during the ordering process.
- PI4 is today's standard and can be bought for the same price as PI3.
- When running FortiusAnt only, 1Gb memory is (more than) enough. I have bought 4Gb for two reasons: it enables future usage (for other purposes) and less memory was not available. Task manager reports a memory usage of 200MB
- 8Gb disk (SD card) is enough, take the fastest available you can afford. I have taken 32Gb to enable future use as well.
- You can order the disk with or without operating system, it may be comfortable to order with Raspbian already installed. (see also next section).
- 3A power supply is also to be prepared for future expansion.
- On Pi4 Cooling is recommended. When running FortiusAnt the fan does not seem necessary (it is never switched on); if you do not mind the noise permanent on keeps your Raspberry Pi cooler.
- I use a micro-HDMI to VGA cable because I had such a monitor available; the Raspberry Pi has micro-HDMI connectors so select a cable that is suitable for you.

3.12.2 Installation decisions

The installation description is assuming that the Raspberry Pi is dedicated for the Tacx and FortiusAnt and will describe a straight-forward FortiusAnt-dedicated installation which is good for the novice and a reference for others. Therefore:

- I use standard root-user Pi as operational user, although a non-root user would be nicer from security point of view.
- I do not use a virtual environment, to simplify for most users. If multiple applications used on the same Raspberry Pi, Python's virtual environment is recommended.
- The button to shutdown is projected on GPIO3 because that is the default power-on pin for Raspberry itself.
- The installation is described for Raspberry Pi3 and Pi4. Raspberry Pi0W would be a cheaper solution but has another architecture (ARMV6l). Installing wxPython on a Pi0 is complex and extremely time-consuming task, so a headless configuration is recommended. Note that the RPi0W and RPi0WH models have inbuilt Bluetooth while the standard RPi0 does not.
- The microSD-card is mounted in the onboard slot; it is also possible to use a USB-3 mounted microSD-card which is faster and makes switching of systems easier. See:
<https://www.raspberrypi.org/documentation/hardware/raspberrypi/usb/README.md>

3.12.3 Operating system

If you ordered an empty microSD-card, you must format and add an operating system to it.

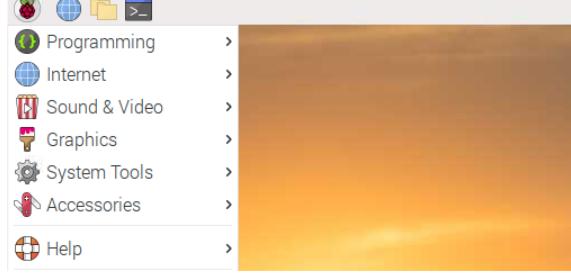
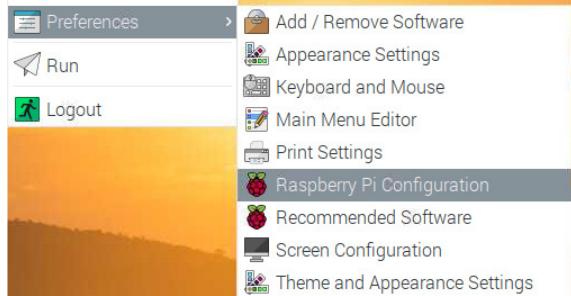
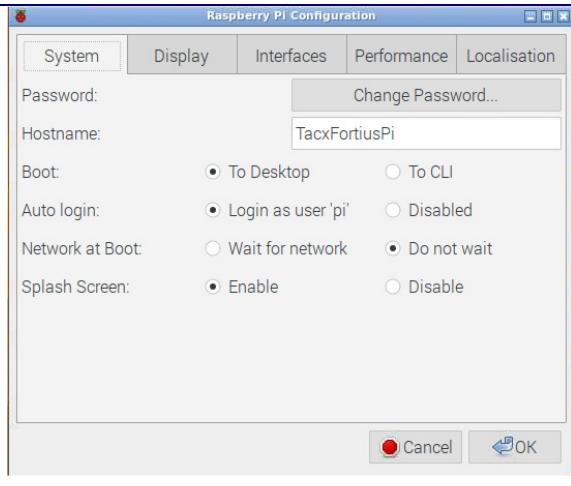
Use another computer to format the microSD-card.

- Mount the microSD-card in a SD-card slot or a USB slot, using an adapter (see examples).
- go to www.raspberrypi.org/downloads.
- Install and start Raspberry Pi Imager.
- Select Raspberry Pi OS.
- Select the SD-card (verify to select the correct one and do not overwrite a system-disk).
- Write SD-card.
- Remove the SD-card from the computer.





3.12.4 Start and stop the Raspberry.

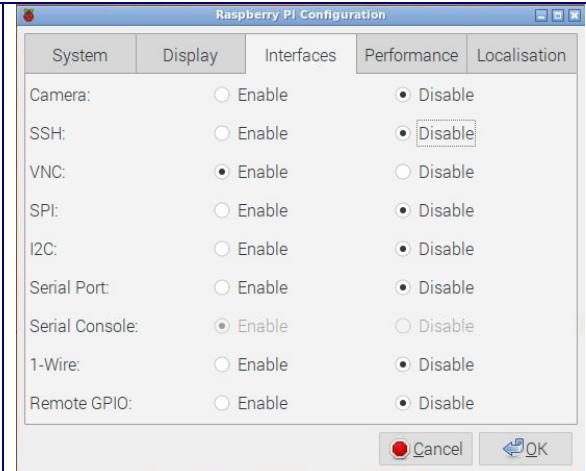
Insert the microSD card into the slot on the RPi, connect keyboard, mouse and monitor and power on by putting the power supply in the mains. A Raspberry Pi does not have a power switch.	
The first time	Execute some simple installation instructions such as: <ul style="list-style-type: none">• Define your language and geography• Connect to Wifi (or LAN) when asked.• Note that if updating the software fails, don't worry as we will do it later on. Your Raspberry Pi is now ready for use.
Normally, starting-up will take half a minute or so, and then you will see a desktop like other computers you are probably used to. In the taskbar at the top you will see three icons which we will use in the next sections:  The applications menu  The file manager  Terminal	
Let's have a short look at the Raspberry Pi configuration. Select "Raspberry Pi Configuration" as shown aside:	
The Raspberry Pi has started with the default user pi and here you can change the password you defined during the first set-up. Pi is the root user (administrator). As you see I have renamed the Raspberry Pi as "TaczFortiusPi". The next settings are default: Raspberry Pi starts with the desktop and automatically logs on as user pi. Do not change these settings unless you know how to manage them.	



If you intend to connect with VNC from your laptop, you have to allow this.

Enable VNC on tab Interfaces.

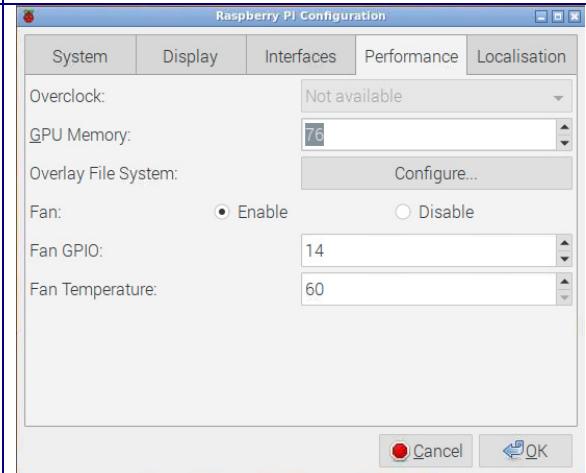
Refer to [\[VNC\]](#) for more information.



If you ordered a Raspberry Pi with a fan this fan is probably connected to a 5V/GND on pin 4/6 and running permanently.

You may choose to connect the fan to pin 8/6 which corresponds to GPIO 14. The fan will now run above 60° and be silent below that temperature.

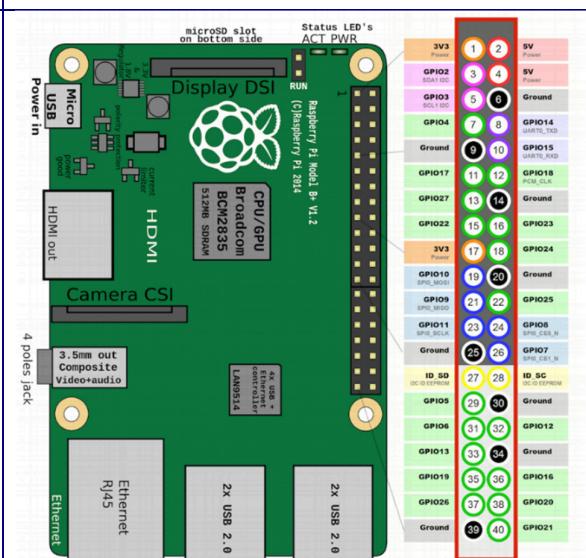
GPIO = General Purpose IO.



The PIN-layout of the Raspberry Pi is displayed here.

Note that the pins are numbered 1, 2...40 and that the functional numbering GPIO## is different!

When programming, the GPIO## number is always used.



Always stop the system before disconnecting from the power to avoid damage to the (data on the) SD-card. This is done easily as shown or alternatively type "poweroff" from the terminal.



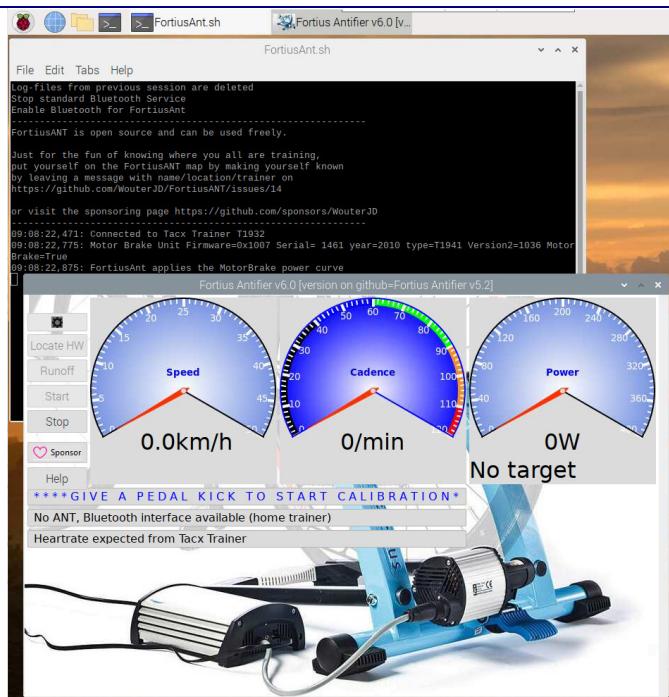


3.12.5 Install software.

The procedure in the table below will take approximately 15 minutes.

If you want to use an ANT-dongle to communicate with [CTP] insert that dongle now.	
Start the terminal	Click
Download GIT to be able to install FortiusAnt	<code>sudo apt install git</code>
Download FortiusAnt	<code>git clone https://github.com/WouterJD/FortiusANT.git</code>
Close the terminal	<code>exit</code>
These are the only "command line" commands you have to give.	
Now open the file manager	Click
The file manager will open in /home/pi which is the home-directory for user pi. Of course, the installation-scripts can also be executed from the command-line. Note that, on the command-line typing the whole name is not required type ./1<tab><enter> instead.	
Upgrade the system software.	Double click <code>1_UpgradeSystem.sh</code> You get a pop-up: Click "Execute in terminal" Press Enter after the message "System is upgraded, press Enter to continue"
Proceed likewise with the following steps:	
Install required packages; execute BLE / ANT or both. Note that for ANT a suitable ANT dongle is required (read issue #61 on github first).	<code>2_InstallPackagesBLE.sh</code> <code>2_InstallPackagesANT.sh</code> (under development)
Install NodeJs for Bluetooth support (version 14.15.3). (Note that this is not required if you only intend to use ANT)	<code>3_InstallNodeJs.sh</code>



If you want to use the GUI, a modified wxPython must be installed. Otherwise, you may skip this step. Skip for RPi0.	4_InstallWxPython.sh
Install FortiusAnt and dependencies	5_GetFortiusAnt_Dependencies.sh
Install the NodeJs parts required for FortiusAnt (if BLE is used).	6_SetupNodeJs_FortiusAnt.sh
Optional: Configure FortiusAnt to be started when Raspberry Pi is started (and user pi is logged in).	7_RunFortiusAntAtStartup.sh
Optional: Make home-directory of user Pi accessible through share PiShare.	8_Share_UserPi.sh
The last step is to grant access to USB to user pi. To enable this, the system must be rebooted. Therefore, this is the last step in the installation procedure.	9_GrantAccessToUSB_withReboot.sh
<p>The Raspberry Pi is now restarted and if everything is well, FortiusAnt is running!</p> <p>In the raspberry directory, you see the script FortiusAnt.sh which is started at poweron/login.</p> <p>This script contains two lines:</p> <pre># autostart, bluetooth, gui, led, no ANT ~/FortiusANT/raspberry/FortiusAntMain.sh -a -b -g -l -D-1</pre>	
<p>Use other options:</p> <p>If you want to start-up with other parameters, you can modify FortiusAnt.sh. (See section 4.2 "Command line").</p> <p>Run headless:</p> <p>To run headless remove “-g”</p> <p>Use ANT-dongle:</p> <p>To use ANT remove “-b” and “-D-1”.</p>	
If you want to start-up with other parameters, you can modify FortiusAnt.sh.	Right-click and start editor: 



3.12.6 Connect LEDs and a button to the Raspberry Pi.

The fun of a Raspberry Pi is that you have an easily accessible GPIO bus (General Purpose IO bus).

FortiusAnt supports connection to five LEDs and a button as follows:

Usage	GPIO pin		Usage	GPIO bus			
Shutdown Button (also powerup)	5	6	Cooling fan as customized on GPIO14	GPIO3 SCL1 I2C	5	6	Ground
		8		GPIO4	7	8	GPIO14 UART0_TXD
	9			Ground	9	10	GPIO15 UART0_RXD
Led Tacx	29			GPIO5	29	30	Ground
Led Shutdown	31			GPIO6	31	32	GPIO12
Led Cadence	33			GPIO13	33	34	Ground
Led BLE	35			GPIO19	35	36	GPIO16
Led ANT	37			GPIO26	37	38	GPIO20
Ground	39			Ground	39	40	GPIO21

When -I is specified on the command-line, the LEDs and button are activated. If -I is not specified, FortiusAnt does not use GPIO-pins.

Using the -L command-line variable, the GPIO-pins can be redefined.

Usage of the LEDs

- Refer to page 27 item 19 "The status LEDs:" for the meaning of the LEDs.
- **Led Tacx + Shutdown** light faintly during Raspberry startup, until FortiusAnt is started. This is the default behavior of the used ports (pull up). The other LEDs are off (default pull down).

Usage of the BUTTON

As you have seen earlier, a Raspberry Pi does not have a power-button. Before pulling the plug, you have to shutdown the system. The question is how to do that if no monitor is attached.

While FortiusAnt is running, press the button. The **Shutdown Led** will blink 5 times and then all LEDs will blink one by one. If you keep the button pressed, then FortiusAnt will light all LEDs, close down normally and shutdown the Raspberry.

Because all the LEDs go off you can see that the procedure is completed and you can disconnect the power.

Whenever the power is disconnected you can (re)start the Raspberry Pi with the same button.



3.12.7 Connect TFT screen to raspberry.

It's also possible to use a mini-TFT screen on the raspberry, -O display/180 tells FortiusAnt to use the ST7789 interface and rotate the image by 180 degrees. The display will show the following texts:

After start-up the initial screen is displayed and after pressing the [Locate HW] button, the screen will be one of the four below. In case of autostart, FortiusAnt proceeds to the next screen immediately.	FortiusAnt started Press [Locate HW] to continue FortiusAnt stopped	
For USB-trainers: Where Txxxx is the head-unit connected, <device> is ANT, BLE or both. And "Give pedal kick" and "calibrating" are only displayed when applicable (depends on trainer and -n flag)	FortiusAnt started Tacx Txxxx trainer Give pedal kick Calibrating... Activate <device> Ready for training Trainer stopped <device> stopped FortiusAnt stopped	The trainer type is displayed and then the high-lighted message shows the state of the trainer. USB-trainers: When [Start] is pressed, calibration is activated (if applicable).
For Vortex trainers: <device> is ANT, BLE or both.	FortiusAnt started Tacx Vortex trainer Activate <device> Waiting for Vortex Ready for training Trainer stopped <device> stopped FortiusAnt stopped	After calibration, the ANT+/BLE interface is activated, and a CTP will see FortiusAnt. The 'ready for training' message will quickly be replaced by the info-screen below.
For Genius trainers: <device> is ANT, BLE or both.	FortiusAnt started Tacx Genius trainer Activate <device> Waiting for Genius Nudge wheel forward Calibrating... Ready for training Trainer stopped <device> stopped FortiusAnt stopped	ANT+ trainers: When [Start] is pressed, the ANT+/BLE interface is activated, and a CTP will see FortiusAnt. Then FortiusAnt waits for the trainer to pair and when paired, will get into the 'Ready for training' state.
For Bushido trainers: <device> is ANT, BLE or both.	FortiusAnt started Tacx Bushido trainer Activate <device> Waiting for Bushido Ready for training Trainer stopped <device> stopped FortiusAnt stopped	The Genius will do calibration after pairing and then get into the 'Ready for training state'.
When the 'ready for training' state is entered, the display will show the actual values	Speed 0 km/hr Cadence 0 /min Power 0 Watt Target 0 Watt Gears 50 : 19	
When the [Stop] button pressed, the ANT+/BLE interfaces are disabled and the 'FortiusAnt stopped' message is highlighted.	... FortiusAnt stopped	
When the shutdown button is pressed the raspberry is stopped and power can be disconnected	... Power can be disconnected	



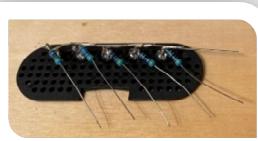
Example of Raspberry Pi with LEDs and button



Starting off with the standard Raspberry Pi casing ...



Some marking on the removable front and soldering the leds



And the Raspberry is operational.

For more suggestions, refer to
<https://github.com/WouterJD/FortiusANT/wiki/Raspberry-Pi-Hall-of-Fame>

I hope these pictures encourage to get creative!

Post your pictures in issue 14 (Put yourself on the FortiusANT map!) and they will be added to this page.



3.12.8 Now pair with a CTP

Zwift, Trainer Road, Rouvy, Sufferfest, Golden Cheetah, Road Grand Tour, BigRingVR, TTS and many others can now pair with the Bluetooth trainer “FortiusANT Trainer” on a device that supports BLE. A Smartphone (Apple or Android) is probably the most generally available device to use and from there the screen can be easily streamed to a TV or monitor. Of course, you can also use a PC with a Bluetooth dongle.

I think that this is the recommended configuration because BLE is native on Raspberry Pi, a Bluetooth dongle is usually cheaper than an ANT-dongle and because ANT-dongles are hard to get (see issue #61). Also, it seems that the market of CTP's is moving from ANT to BLE; usually Bluetooth is preferred and/or supported before ANT.



4. Operating instructions

4.1 The main functions of FortiusAnt and the head unit.

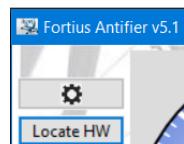
After FortiusAnt is started (see 3.9 “Start FortiusAnt”) you will see the user interface with the buttons **Settings**, **Locate HW**, **Runoff**, **Start** and **Stop**.

The correct start-up sequence is:

- Setup environment (trainer, computer, bicycle, connections) and connect heartrate monitor.
- Start FortiusAnt.
- Press Locate HW to connect to USB-devices (Tacx trainer and/or ANTdongle)
- Execute runoff-procedure (magnetic brake trainers)
- Press Start
 - complete Calibration – motor brake trainers and Genius only
 - wait until Bushido, Genius or Vortex trainer are paired to FortiusAnt.
- Start [**CTP**] and pair with FortiusAnt.
- Start Training or Virtual Ride.

On a raspberry, a small display is connected showing the FortiusAnt process; refer to 3.12.7 “Connect TFT screen to raspberry.” for more info. This section also provides additional information on the process of FortiusAnt, including differences between the different trainer types.

4.1.1 Settings



From version 5.1 onwards settings can also be modified interactively by clicking the “settings” button above “Locate HW”. This button is enabled when FortiusAnt is not active.

See section 4.3 “Modify settings” for more info.

4.1.2 Locate HW

Checks for the presence of USB-trainer and ANT-dongle or Bluetooth. If successful, results are displayed, and the button is disabled. For more info read section 4.4 “Locate HW”. Note that, when successful, FortiusAnt is not yet discoverable by a [**CTP**]; see section 4.1.4 **Start**.

4.1.3 Runoff test

To ensure comparable and accurate training sessions, the trainer should provide the same relative resistance each time

1. Aim for about 7 bar (100psi) in the rear tire when cold
2. Warm up for 2-3 minutes to warm tire
3. Increase speed and exceed 40 km/hr then stop pedaling and let wheel slow down on its own
4. Ideally, the wheel should stop after 7 seconds from 40km/h

If the rundown time is too short, the rolling resistance may be too high and if the rundown is too long, the rolling resistance may be too low. In that case, adjust the roll using the wheel behind the trainer motor/brake and retry the rundown test.

Using the -R command-line parameter, the procedure can be customized:

- maxSpeed = the speed that must be reached, default = 40 km/hr (allowed = 20...50)
- dip = the speed under maxSpeed when the runoff timer starts, default = 2 km/hr (range allowed = 0...5)
- minSpeed = when the runoff-timer stops, default = 1 km/hr (range allowed = 0...10)
- targetTime = runoff target time, default = 7.2 seconds (range allowed = 0..10)
- power = the power during the runoff, default 100W (range allowed = 0...500)



4.1.4 Start

FortiusAnt starts to calibrate the trainer (if supported by the trainer and -n is not specified). Calibration means that the brake rotates the wheel at 20 km/hr and returns the resistance found. As soon as the resistance is constant, the calibration stops. The calibration time is at least 30 seconds (warming up the tire) and stops when the resistance value is constant.

Note that the calibration starts when you turn the pedal as if starting to cycle, which is the only physical action to take. Note that, starting the motor automatically would be a risk for physical injury, therefore the confirmation with a pedal-kick is required.

When calibration is started, **do not pedal**, the process completes automatically.

After calibration, the Fortius is ready for training and will listen to the **[FE]** (Tacx USB-trainer) and **[CTP]** and exchange info between them. Also, information is received from **[HRM]** and **[CTRL]** and information is broadcast as **[PWR]** and **[SCS]**.

4.1.5 Stop

Pressing this button stops any running process (runoff, calibration or operational mode).

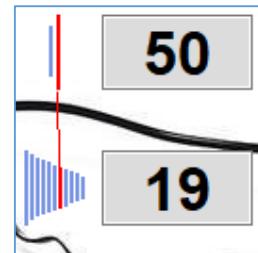
4.1.6 Buttons on the Tacx head unit



There are four buttons: Cancel, Enter, Up, Down.

- If not in an active mode, **Up/Down** navigates through the menu, **Enter** activates the selected button. You cannot use **Cancel** anymore to stop FortiusAnt; the "Stop"-button must be used.
- In runoff or manual mode, **Up/Down** modifies the required power by ±10Watt. **OK *** resets the power to the initial value of 100Watt. **Cancel** stops the runoff.
- In manual grade mode, **Up/Down** modifies the slope by ±1degree. **OK *** resets the slope to the initial value of 0 degrees.
- In homeTrainer mode, **Up/Down** modifies the required power by ±10%. **OK *** resets the power to the initial value of 100Watts. The minimum value is 10Watts.
- Note that Manual power and manual grade mode are intended for test-purposes, while homeTrainer mode is intended to be used for exercising.
- In resistance and ERG mode, **Up/Down** and **Cancel** modify the resistance of the Fortius using the virtual gear; **Up/Down** controls the cassette and **Cancel** the front gear. **OK *** resets the resistance to the initial value of 100%. The standard transmission = double 50-34 x 11 speed 11-34. The virtual gear is displayed to represent the reduction.

* Tacx has a variety of trainers with different head units and not all head-units have an OKbutton.





4.1.7 ANT+ Remote Control



It may be useful to use an ANT+ Remote control instead of the head unit. Not all Tacx configurations have a head unit; sometimes the head unit does not fit the bike or the buttons are impractical.

Any standard ANT+ Remote Control can be used to command FortiusAnt acting as Up / OK / Down as described in the previous paragraph.

No configuration required; you can pair the control with FortiusAnt as described in the related manual.

4.2 Command line

FortiusAnt is started with a command (see 3.9 “Start FortiusAnt”). In addition to the examples shown there, parameters can be passed by using the command-line:

```
usage: FortiusAnt.py [-h] [-a] [-A] [-b] [-B 0...65535] [-c 0...100] [-d 0...255] [-D USB-DeviceID] [-e] [-g]
                     [-G % / % / %] [-H ANT+DeviceID] [-i] [-l] [-L value] [-m] [-M] [-n] [-p %] [-P] [-r] [-R see text]
                     [-s] [-T see text] [-t {Bushido,Genius,Vortex,Magneticbrake,Motorbrake,i-Vortex}] [-x]
```

Basic arguments:

- h show this help message and exit
- l shows speed in imperial units (mph)
- a Automatically start; “Locate HW” and “Start” if the required devices were found.
- b Advertise FortiusAnt as “FortiusAnt Trainer” on a Bluetooth Low Energy dongle.
- e Run in homeTrainer mode (-e stands for Exercise bike)
- g Run with graphical user interface (GUI).
- t value Specify Tacx Type; if not specified, USB-trainers will be detected automatically. Allowed values are: Bushido, Genius, Vortex, Magneticbrake, Motorbrake.

Raspberry:

- l When -l is specified the RPi LEDs and button are activated. If -l is not specified, FortiusAnt does not use the RPi GPIO-pins.
- L value The value specifies the GPIO-pin for Button(3), TacxLed(5), ShutdownLed(6), CadenceLed(13), BLE-Led(19) and ANT-Led(26), separated by /. The default values are mentioned between (and). If -O is specified, the button is moved to GPIO-16.
- O value The value specifies what OutputDisplay is to be used. Allowed values are Display (default ST7789 TFT display) or Console. For displays, also a rotation can be defined, separated by / (0,90,180,270 degrees).

Power curve adjustment:

- c value Calibrate the rolling resistance for magnetic brake. See section 6.4 “PowerCurve for i-Flow (T1901-T1932)”. Default is 15. If the power, displayed by FortiusAnt is higher than your power-meter, specify -c 14 to reduce the RollingResistance by 1 Newton.
- G value Modify the requested grade with a factor/factorDownhill; see section 6.4 “PowerCurve for i-Flow (T1901-T1932)”. Note that GradeAdjust is not active in manual mode.
FortiusAnt -a -g → the default value: targetGrade = 100% of requested Grade
FortiusAnt -a -g -G50 → recommended when using Rouvy: targetGrade = 50%
FortiusAnt -a -g -G50/50 → targetGrade = 50% (uphill) and 25% downhill (50% * 50%)
FortiusAnt -a -g -G100/0 → targetGrade = uphill only (replaces former -u flag)
FortiusAnt -a -g -G0/0 → targetGrade = always flat
- p value Adjust target Power by multiplying by this factor for static calibration.
If the power, displayed by FortiusAnt is always 10% higher than your power-meter, specify -p90 to reduce resistance by 10%.



Advanced arguments:

- A Pedal Stroke Analysis.
- B value ANT DeviceNumber range Base, making multiple FortiusAnt sessions unique, which is useful if you have multiple FortiusAnt sessions simultaneously in the same room.
- D value Select one specific antDongle (perhaps with a non-standard deviceID). The following types are known: 4104 = Suunto, 4105 = Garmin, 4100 = Older. -1 means: no ANT.
- H value Pair this Heart Rate Monitor (0: any, -1: none). Tacx HRM is used if not specified.
 - FortiusAnt -a -g → will use Tacx heartrate monitor (analog HRM paired to Tacx)
 - FortiusAnt -a -g -H-1 → no HRM is used at all
 - FortiusAnt -a -g -H0 → FortiusAnt will pair with first HRM found
 - FortiusAnt -a -g -H1234 → FortiusAnt will pair with HRM with ID=1234
- m Run manual power (ignore target from ANT+ Dongle).
- M Run manual grade (ignore target from ANT+ Dongle).
 - When -m or -M is specified, a .tcx file will be created for every exercise.
- n No calibration (do not calibrate before start).
- P Power mode has preference over Resistance mode (for 30 seconds).
 - Run Zwift or Rouvy to ride a route, the target is transmitted as a grade and you see where you ride. In parallel, run Trainer Road (TR) to do a structured training and the target is transmitted as power. FortiusAnt “listens” to TR and transmits power and cadence to both. Now you can do a structured TR-training in the virtual world of Zwift or Rouvy.
- R value The runoff procedure can be customized: maxSpeed/dip/minSpeed/targetTime/power. Refer to 4.1.3 "Runoff test".
- T value Transmission, default value = 34-50*x34-30-27-25-23-21-19*-17-15-13-11
- x Export TCX file to upload into Strava, Sporttracks, Training peaks.
 - x is implicit in manual (grade) mode.

Developer arguments:

- d value Create logfile with debugging data (see section 4.7 “Debugging FortiusAnt”). Value is a binary flag list that defines what to write to the logfile0=nothing, 127=everything. The values below can be added together.
 - 0 = No logfile
 - 1 = Application
 - 2 = Function
 - 4 = antDongle
 - 8 = usbTrainer
 - 16 = multiProcessing
 - 32 = jsonFile
- r Target Resistance = Target Power (to create power curve)
- s Simulate trainer to test ANT+ or BLE connectivity.

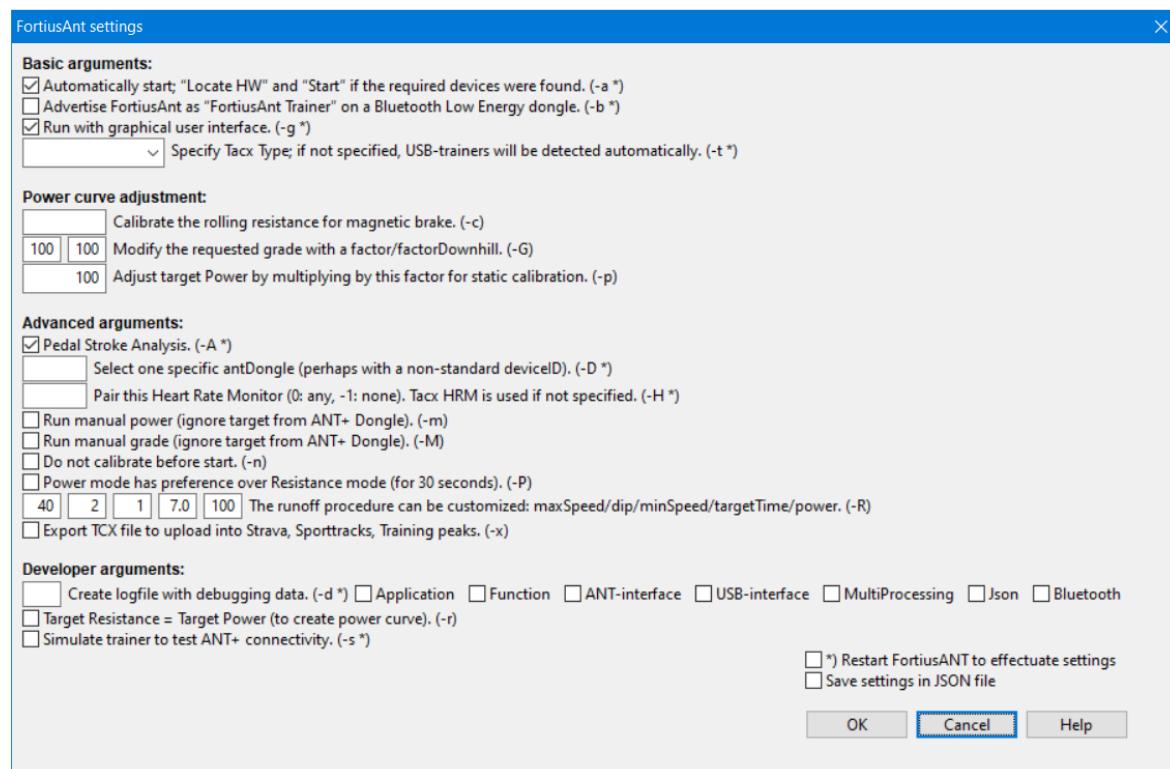
Examples:

FortiusAnt.py	FortiusAnt is started without user-interface, -g -a -A -H0 are assumed.
FortiusAnt.py -g -a	FortiusAnt is started with user-interface and starts automatically.
FortiusAnt.py -g -m	FortiusAnt is started with user-interface. No [CTP] is required, power can be set using the console. Although intended for interface testing, you could do a manual ride this way.
FortiusAnt.py -g -M	Same as -m but now the slope-grade can be adjusted.
FortiusAnt.py -g -s	FortiusAnt is started with user-interface. No [FE] is required, automatic response to [CTP] is generated. This is intended for interface testing.
FortiusANT.py -a -g -H0 -t Bushido	
FortiusANT.py -a -g -H0 -t Genius	
FortiusANT.py -a -g -H0 -t Vortex	



4.3 Modify settings

When pressed, the following dialogue appears. The dialogue contains the settings that are currently active.



All settings are described in the previous section but please note the following:

- All settings can be saved in a JSON-file, which is stored in “the current directory”.
- “The current directory” is the directory that is active when the .py or .exe file is started.
- Settings marked with *) will cause a restart of FortiusAnt after clicking the OK button. The restart will be done automatically.
- The restart can be suppressed by de-selecting the appropriate checkbox above the OK-button, but then only the options that are not marked with *) are in effect. The marked options will be in effect during the following session if saved to the JSON file.

The precedence of settings is: **(1) JSON file; (2) command line; (3) default values**.

The reason for the latter is that, if you change the settings, save in the JSON-file and execute the same command, the settings in the JSON-file are used.

Recommendations:

- if you use a JSON-file do not use command-line options.
- If you want to use a command-line then, note that the JSON-file takes precedence.

Because of the variety of systems used (Windows, Linux, Raspberry, ...) no special rules are (yet) defined for system-standard locations of a JSON-file.



4.4 Locate HW

When the “Locate HW” button is pressed the following happens:

Find ANT+ dongle

A check is done whether an ANT+ device with DeviceID 4100, 4104 or 4105 is found. If found, an attempt is made to use the dongle; if in use another dongle will be searched for.

FortiusAnt always needs an ANT-dongle, unless -m or -M is specified in that case you can set power or slope with the trainer's head unit buttons.

The following messages can be displayed:"

No (free) ANT-dongle found

Using <manufacturer> dongle

or messages indicating what interface-error occurred.

Find Tacx Trainer

Then a check is done whether a Tacx device is used with one of the DeviceID's as listed in section 2.3 “Tacx trainers”.

The following messages can be displayed:"

No Tacx trainer found

Connected to Tacx Trainer T<DeviceID>

or messages indicating what interface-error occurred.

Note that, when the -s command-line parameter is specified, the following message is displayed:

Simulated Tacx Trainer to test ANT-interface

Note that, when the -t i-Vortex command-line parameter is specified, the following messages are displayed:

Pair with Tacx i-Vortex and Head unit (pairing can take a minute)

Tacx i-Vortex paired: %s, Head unit: %s

Heartrate monitor

Old Tacx trainers paired with a heartrate monitor (HRM) and passed the heartrate through the USB-interface to **[TTS]**. Even though this option is supported on the **[FE-C]** ANT+ interface, it is not used by **[CTP]** since this software pairs with a HRM itself.

The FortiusAnt display shows the heartrate and therefore the following options exist:

- No command-line option: use the heartrate from the Tacx trainer
- -H0: pair with an ANT+ HRM, use the first HRM that is found
- -Hnnnnn: pair with the ANT+ HRM with DeviceID=nnnnn
- -H-1: no HRM.

The following messages can be displayed:"

Heartrate expected from Tacx Trainer

Heartrate expected from ANT+ HRM

Heart Rate Monitor paired: <DeviceID>



4.5 The FortiusAnt display in power mode

FortiusANT Graphical User Interface – Operating in Power mode



This display means that, the [CTP] requires 345Watts, which results in a resistance of 2345.

The athlete currently cycles at 89 revs/minute, resulting in 123Watts at a wheel speed of 34.5 km/hr.

The cycling athlete can modify the requested grade (→ power and resistance) using the virtual gearbox; up/down changes the rear gear and the cancel button changes the front gear.

(Note that all figures in this graphic are fictitious and do not represent actual training).

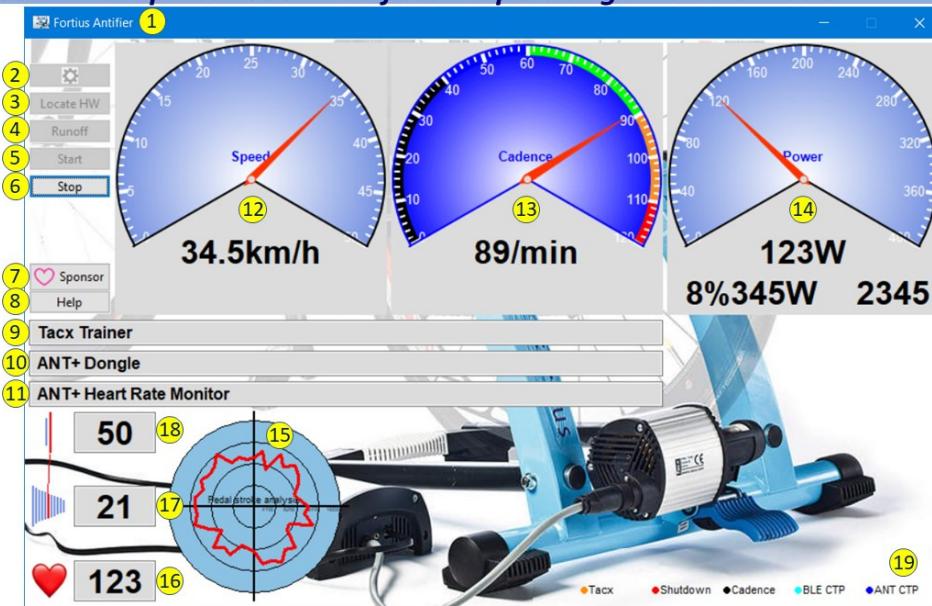
The elements on the screen are explained in detail in section 3.9 “Start FortiusAnt”.

In addition to the standard display, under the power gauge (11) the actual power (99W), the requested power (100W) and the resistance set in the trainer (2345) are displayed.



4.6 The FortiusAnt display in grade mode

FortiusANT Graphical User Interface – Operating in Grade mode



This display means that, the [CTP] requires a slope of 8% which (using speed and user/bike weight) results in a requested power of 345Watt and a resistance of 2345.

The athlete currently cycles at 89 revs/minute, resulting in 123Watts with a wheel speed of 34.5 km/hr.

The cycling athlete can modify the requested grade (→ power and resistance) using the virtual gearbox; up/down change the rear gear and the cancel button changes the front gear.

(Note that all figures in this graphic are fictitious and do not represent actual training).

The elements on the screen are explained in detail in section 3.9 “Start FortiusAnt”.

In addition to the standard display, under the power gauge (11) the actual power (257W), the requested grade (0%) and resulting required power (274W) and the resistance set in the trainer (2345) are displayed.

Here you also see the “digital gearbox”. In addition to the gears on your bicycle imagine you have a second drive train with a 15x15 ratio. So, the displayed situation is neutral. When you press Up/down the ratio is changed with ±10%, increasing or decreasing the required resistance on the bike. The digital gearbox shows the number of teeth on your digital cassette.

If your [CTP] requires you to ride uphill with a slope of +10% and you have a high wheel-speed (because that is how the Fortius works well: see section 5.1 “Low cadence on Fortius”) you would need a high power. If that required power is higher than you can produce, you can reduce the required power using the down-button without reducing speed. If you are using Zwift you will see that the difference in the speed shown in Zwift (based upon power) and the speed displayed by FortiusAnt will increase.

The digital gearbox is not active in power-mode because if 100W is required you have to produce 100W. At a high wheel speed, the resistance is already calculated accordingly.

Thanks to Erik OT for the magnificent suggestion!



4.7 Debugging FortiusAnt

FortiusAnt has a debugging flag -d specifying what output must be written to the logfile.

The options are defined as a decimal number, being the sum of the options desired:

No logfile	= 0x00	= 0
Application	= 0x01	= 1
Function	= 0x02	= 2
antDongle	= 0x04	= 4
usbTrainer	= 0x08	= 8
Multiprocessing	= 0x10	= 16
JSON	= 0x20	= 32
Bluetooth	= 0x40	= 64

For example, If you want all options activated use, -d127.

To log the ANT+ calls: use -d4 and if you want ANT+ and the USBtrainer to be logged use -d12.

If -d32 is specified, a JSON file will be created with the relevant data from FortiusAnt. This may be helpful for further analysis.

Although books could be written to explain the content of logfile and JSON file, I leave it to the user's creativity to understand the content. If you want to interpret the JSON file, "FortiusAnt JSON Analysis.xlsx" in the support files section on github may be helpful.

Note that when the GUI is running in a separate thread (e.g. because Pedal Stroke Analysis is active) a .GUI.log file is also created.



5. FAQ and Troubleshooting

5.1 Low cadence on Fortius

One of the limitations of the Fortius is the resistance at low wheel-speed, which implies a low rotational speed of the brake (combined with a high torque).

If you have a high wheel-speed the Fortius can realize a resistance of up to 1000Watts. This works fine and is good to train your power output.

Doing an "Uphill training at 20%" requires some thinking however. In real world, you would change gears and reduce speed to be able to ride uphill. If you do that on a Tacx Fortius, you will quickly touch the "low wheel-speed limitation".

On the Tacx Fortius, I always use the highest gear on my bike, and therefore realize the highest possible wheel-speed. All I do is **vary cadence** – and don't use the bikes shifters. When looking at the wheel-speed, you will ride uphill with an unrealistic speed. But hey, note that modern direct-drive trainers don't even have a rear wheel. The message is: ignore the wheel-speed.

In the example of the 20% uphill, even at 50 rpm, probably the required power will be too high to realize. At that moment use the "virtual gear box" shifting with the head-units up/down buttons to reduce/increase speed (internally: reduce required resistance).

Summarized: use highest gear on the bike, vary cadence and the virtual gears in FortiusAnt.

Refer to section 2.6 "Tacx Trainer Power ranges".

5.2 Flat resistance on magnetic brake at low speed (#223)

If you do a virtual tour on Rouvy and tend to be in a relative high gear it may mean that you simulate a ride uphill of 10% at a speed of 30 km/hr. This will result in a high resistance and you need a high FTP!

Most likely you will not enjoy the ride, your speed will decrease and you will be training at low cadence and when you switch to a lower gear and/or ride at a lower speed the brake does not feel good.

See also previous topic.

You can tune FortiusAnt in two ways:

- Use the virtual gear and shift to a lower gearing
- Specify -G50 on the FortiusAnt command-line to reduce the slope.

For more info refer to issue #223.

5.3 Power tuning with Golden Cheetah (#231)

This is not an issue but I wanted to recognize the excellent work you have done and provide a setup that others may find useful.

I'm running on Windows 10 and the hardware in use is as follows: Tacx Trainer T1932 (Motor Brake Unit Firmware=0x1007 Serial= 6639 year=2010 type=T1941 Version2=3086 MotorBrake=True) and using the development version of Golden Cheetah and selecting the FEC trainer presented by FortiusAnt

One of the issues I had was adjusting the power (which you kindly solved in #210). With the new feature in 5.1 of the virtual gears I now have a working setup where I can fine tune the power reflected by the brake as it changes during the session (this is something that the brake I have has always done...power changes through the session, sometimes quite a lot).

Basically, I use the virtual gearing to do the fine adjustment during the session and I leave Golden Cheetah to adjust the actual power rates for the session.

I start FortiusAnt with the following options (the HRM is really optional -H0 works just as well). The power difference between the value recorded from the brake and the powertap hub in use is adjusted at a macro level using '-p135' and I have tightened up the mid-range virtual gearing so that I have fine granularity



using changes between 17 and 21 teeth on the rear cassette. This is adjusted using the '-T34-50x30-27-25-23-21-20-19-18-17-15-13' option.

Details of the shortcut (call the downloaded .exe whatever you want):

```
...\\FortiusANT.exe -n -g -a -H10030 -p135 -T34-50x30-27-25-23-21-20-19-18-17-15-13
```

This now means that I can finely adjust the power difference to stay within a few watts of the Golden Cheetah (GC) profile throughout the ride. While this is not a show stopper for folk using GC it does mean you don't get the GC interface colouring your power output because you are higher/lower than the specified rate.

Absolutely fantastic job and it's a huge help in me maintaining my fitness for a return to Brazilian Jiu Jitsu in the coming months. And it has given the Fortius a huge new lease of life :-)

Dankjewel.

Paul C

5.4 Swift speed does not match Garmin

If you ride your bicycle on your trainer the speed that Swift displays does not match Garmin's speed.

Your Garmin receives the speed from the bicycle and displays the real speed of the wheel on the trainer. (Probably you realize, with the same effort, you would not achieve that speed in the real world)

Swift receives the realized power from the trainer and uses its own algorithms to conclude what speed you are riding: If riding on a flat surface, without headwind, the simple formula would be: speed=power/resistance. But when you ride uphill and you take air-resistance into account, the formula is far more complicated.

Interested in the power you need to ride?

- <https://www.fiets.nl/2016/05/02/de-natuurkunde-van-het-fietsen/>
- https://www.gribble.org/cycling/power_v_speed.html

The gribble formulas are used by FortiusAnt to convert grade to power - see also section 2.5.2 "Grade mode".

5.5 Average speed in Trainer Road

Jerome uses TrainerRoad and notices that during a ride his average speed is 20km/hr and his friend is going at 40 km/hr. How is that possible?

Imagine that TrainerRoad requires you to ride with a Power of 200Watts, the two riders have the same bike and the same trainer and have selected the same gear-ratio.

If Jerome chooses to ride with 50 rpm, his speed will be low, and the trainer will increase the resistance so that the required power of 200Watts is realized.

If his friend rides at 100 rpm, his speed will be higher than yours and the trainer will decrease the resistance so that the required power of 200Watts is realized.

Note that power = resistance * speed! Jerome rides at half speed of his friend, but with a higher resistance and hence both athletes produce the same power at a different speed and different distance. Note therefore that, on a [FE] speed and distance are irrelevant, time, power and cadence are the deciding factors.

For you it's a pity that your Strava statistics for the end-of-year applause are adversely affected; but having read the explanation that would be a lesser issue.

The situation in real world is different: if you go for a ride together time, speed and distance will be the same and therefore you will come home with the same average power. Difference choice in gears will change the resistance and cadence with less impact on the Strava results.

Why would you ride at 50rpm or 100rpm?



When TrainerRoad demands 200Watt it may instruct you (through the displayed texts) to ride at a high or a low cadence, depending what purpose is intended. It is not correct to say that the speed is not relevant: it may be that either you or your friend did not do the intended training (high power or high force).

5.6 Can TTS4 and FortiusAnt coexist?

[**TTS**] is Tacx' own training software which uses its own USB-driver and/or ANT+ interface. The installation instruction suggests to de-install (for simplicity), see 3.6 "Install USB-driver".

The following information is of interest for everybody who would like to work with [**TTS**] and other [**CTP**]'s.

lepuzaur: I am running on Windows10 TTS4 and FortiusAnt, at first I also thought I have to uninstall the Jungo driver TTS uses, but then I noticed you can have both drivers installed and there is no issue on my system.

Of course, you cannot use them concurrently, if you start TTS4 it will use the Jungo driver, while when you start FortiusAnt the libusb drive will be selected.

So as long as you do not intent to use them at the same moment, the drivers shall not conflict (at least this is my case). (Thanks to @lepuzaur, june 2020).

Note however: If your TTS4 software is already installed and registered then that should work. If it isn't, it won't as the registration servers have been switched off. The only software that works without registration is Fortius v2.02.

Never uninstall TTS4 if you have a working registered copy as there is no way to reinstall it and re-register it. (Thanks to Shaun Murray, @aegisdesign, sept 2020).

frensk8: I have been using FortiusAnt for 2 months now in combination with the Tacx Desktop App (Windows 10). It works fine! The Tacx Desktop App connects via ANT with ANT-ID 57591 (identified as Tacx Neo2T!).

Incidentally, I also own the 4iii Viiiva HRM with ANT to BT bridge, but I expect the bridge to work in one direction only, so not from BT to ANT. So, using BT via the 4iii HRM is unlikely to provide a solution, I failed. However, this is not necessary, because FortiusAnt can work directly via ANT with the Tacx Desktop App.

By the way: I'm using FortiusAnt and Tacx App on two different laptops, because I sometimes want to use Tacx TTS4 (installed on the same laptop as Tacx Desktop App). I failed to get FortiusAnt and TTS4 to work on the same laptop.

Note that, TTS4 and Tacx Desktop App are 2 different applications. TTS 4 is outdated software, but like the Tacx Desktop App. A few years ago, Tacx replaced TTS4 with the Desktop App and they changed the revenue model: with the Desktop App they introduced the subscription model (monthly / annual fee). TTS4 uses real life videos that had to be purchased separately.

FortiusAnt also works with TTS4 (with 2 laptops), but that is not necessary, because TTS 4 also works via the USB connection of the Fortius trainer (with the Jungo USB driver that is installed by TTS4). It is **tricky to uninstall the jungo driver** (to try to run TTS4 and FortiusAnt on 1 laptop), because Tacx has ended the support of TTS4 and reinstallation of TTS4 is therefore no longer easily possible (because Tacx servers for authentication are out of operation).
(Thanks to @frenkse8, August 2020)

Github: read issues #99, #168, #180

TTS/ANT+: TTS can successfully pair with FortiusAnt and then the Tacx USB driver is not needed anymore. Of course, then two ANTdongles are required.



5.7 TTS and FortiusANT on Dedicated USB-ports

@Barry01 wrote: <https://github.com/WouterJD/FortiusANT/issues/256#issuecomment-792636230>

The way that was successful for me, is the following:

- I use TTS4 while the trainer is connected to USB port 1. So, check that the trainer is connected to the USB port you want to use with TTS.
- Switch the trainer to the USB port you want to use with FortiusANT (I use USB port 3), the device manager should automatically assign the Jungo driver by default
- Download the libusb driver as described in chapter 3.6.2. of the manual. Do not uninstall the TTS-driver.
- Install the libusb driver (by executing the 64 bit version in case of Windows 10) and follow install option 1 in chapter 3.6.2 of the manual so that the driver is now assigned to the trainer.
- -Then, I put my Suunto ANT+ dongle in USB port 1 and I start FortiusAnt with parameters -a -g -A.
- Calibrate as described in 4.1.4. of the manual.
- After calibration, my heart rate is shown in the GUI and I put my second ANT+ dongle (brand: Anself) in USB port 2.
- Now I am ready and I start Zwift.

For me this works perfect as Windows remembers which driver to choose for which USB port. So I don't have to fiddle with the drivers anymore. I just marked the USB ports so it is easy to see where to put in the trainer for which use case ;)

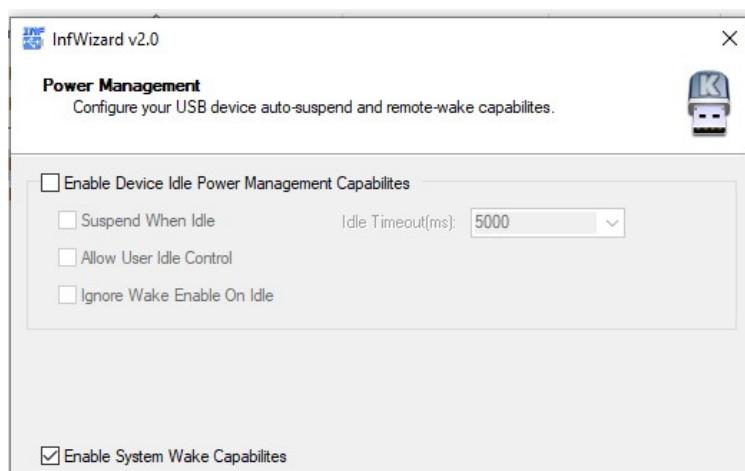
5.8 Tacx head unit with firmware to be loaded

Some head units (0xe6be, Old "solid blue" Fortius) do not have firmware; the firmware must be loaded.

When FortiusAnt discovers a 0xe6be head unit, the firmware is loaded automatically and (after a wait time of 5 seconds to let the head unit reboot) a 1942 head unit is expected to be present. The following messages will be displayed, before/after loading the firmware:

```
....Stop CPU  
....Start CPU
```

Some users have reported that the head unit is not found after the reboot and state that this can be resolved with the following settings in the Power Management Options of the libusbK inf wizard:





5.8.1 **Unable to load firmware" on 1942 head unit**

In other cases, the head unit was not found because old TTS-drivers were still available. It seems that the head unit (without software) is detected by the libusb-driver but not after the reboot (#158)!

@BrRoBo reports through issue #155 see also #101:

I seem to have had the exact same issue as @bradyap, but found a quick solution to my case. Brady's screenshots of his issue are identical to what I was seeing and we had the same head unit... T1942. The install is on a clean version of Windows 10 64bit, no prior TTS or other Tacx software installed.

What I think happens is that we do the lib32 USB driver install and successfully see the Fortius unit listed under libusb-win32 devices along with how-ever-many Ant USB stick 2 are connected. Then when we run the FortiusAnt.exe application, it seems to be looking for a different device or registration, causes the Fortius to unload from the list of "libusb-win32 devices" and then there is a new "Tacx Fortius" listed under "Other devices". This unregistration, re-registration cycle is reproducible over and over by disconnecting the Fortius USB, then plugging it in to load the lib32 driver and seeing it properly listed; then using the FortiusAnt "Locate HW" will again cause the Fortius to get re-registered under "Other devices".

The simple solution at this point is to open the properties for the "Tacx Fortius" listed under "Other devices", then "Update drivers", select "libusb-win32 devices", then "Tacx", then select whatever name you used ("VR-Interface", "Fortius Virtual Trainer", etc. FWIW, I used the latter "Fortius Virtual Trainer" as my device name). Bingo, the Fortius head unit is reloaded into the correct list of "libusb-win32 devices". Issue fixed for this USB port. If the Fortius USB plug is moved to a different port, this library/device driver fix must be performed once for each port to correct the issue and then will no longer occur if the Fortius USB cable is moved to different ports.

5.9 Fortius without cadence sensor

Although it appears that the cadence sensor is for information purpose only the Fortius (and perhaps other Tacx trainers as well) does not work if no cadence sensor is connected and/or no cadence is measured. Note that the magnet must be correctly positioned relative to the sensor.

In a downhill ride, the brake will drive your bike (negative power) provided that you keep pedaling. If you stop pedaling, the motor will stop. Also in normal mode, usually after 4 seconds or so, if the unit detects no cadence, it will "stop" the motor.

If you do NOT have a cadence sensor, you will experience a high resistance – because the brake is in stop-mode. Therefore, operation without the standard cadence sensor is not possible.

5.10 Two ANTdongles – disturbed communication

Two laptops are used:

- The left laptop is connected to Tacx Fortius, is running FortiusAnt and has the ANTdongle at the left side.
- The right laptop is running Zwift and has the ANTdongle at the right side.

Sometimes the connection is lost and FortiusAnt does not display the required grade as transmitted by Zwift or Zwift appears not to receive power data from FortiusAnt.

After moving the dongles so that these are closer to each other the transmission errors were gone.

Perhaps the presence of two computers between the two dongles disturbs the ANT-transmission; note that ANT is low-power.

Therefore, in case of transmission errors check whether there may be interference by placing the dongles closer together with no other electronic devices between them.



5.11 Tacx returns insufficient data

This error can be displayed and has different reasons:

- FortiusAnt reads too often from the head unit and therefore the head unit is not ready to provide the next buffer
- Loose or faulty cabling
- Other hardware errors

The following message is given

Tacx returns insufficient data, len=XX

To resolve, try to run without Pedal Stroke Analysis (PSA)

When pedal stroke analysis is active (-A flag), FortiusAnt reads more often from the head unit and this MAY cause the error.

Or Tacx returns insufficient data, len=XX

To resolve, check all cabling for loose contacts

FortiusAnt is developed using a 1932 head unit on a Tacx Fortius. In that environment, the message is given very seldomly and the built-in retry (4 times) appears to work. Some systems, even without Pedal Stroke Analysis, show the short message error more often which needs further investigation.

5.12 Sudden drop of requested power

If you are experimenting with multiple CTP's (for example Zwift on one system and Trainer Road on another) it is possible that multiple CTP's send commands to FortiusAnt. FortiusAnt listens to both and changes behavior as requested. If one CTP commands 100Watts and the other sends Grade=-20% the required power will bounce between the two requests.

This can be intended, for example you ride a structured training in Trainer Road and entertain yourself with a virtual ride in Zwift. In that case specify -P so that the ERG mode commands from Trainer Road take precedence over the grade-mode requests in Zwift.

Of course, if two CTP's both send ERGmode requests (100Watts and 200Watts) then you'd better switch off one of the CTP's 😊.

5.13 Zwift on Android does not pair with FortiusAnt

Check Zwift whether your Android/ANT hardware is supported; that is not always the case.

Reference: <https://forums.zwift.com/t/ant-update/195323/119>

Please understand I do not try to maintain a Zwift (or other CTP) hardware compatibility list.

5.14 Detailed notes on Tacx Genius

- Installing t19xx USB drivers is not required
- Turn Genius brake on
- Run FortiusAnt with '-a -g -H0 -t Genius'
- Pairing to the Genius is automatic (may take a minute)
- Calibration is initiated automatically after pairing
- When requested, nudge the wheel forward by hand (or give the pedals a light kick)
- The brake spins up the wheel automatically, DO NOT PEDAL DURING CALIBRATION
- After calibration is complete, a calibration value will be shown
- A value between -15 and 15 is good
- A value lower than -15 indicates that the roller pressure is too low, should increase using knob on the trainer
- A value lower than +15 indicates that the roller pressure is too high, should decrease using knob on the trainer



- To re-do calibration, turn the brake off and on again, then restart FortiusAnt
- For consistent results, warm up for ~5 minutes before starting FortiusAnt is recommended
- Calibration (and warm-up) can be skipped by adding the '-n' command-line option
- The brake remembers the last calibration value
- This should be fine if the last calibration is recent and nothing was changed/tyre pressure is the same
- FortiusAnt will show 'UNCALIBRATED' to remind you when the calibration is not fresh

Known issues:

- Cadence estimation by the brake is not always reliable and often seems to read 3-5rpm high
- In ERG mode the wheel sometimes takes a while to stop spinning after you stop pedaling
- Downhill simulation slows down after a few seconds if you stop pedaling completely (built-in safety limitation)
- The Genius remote control (T2022) is not supported (yet)
- Virtual shifting is currently not implemented for the Genius

5.15 Detailed notes on Tacx Bushido

- Installing t19xx USB drivers is not required
- Bushido support is experimental (consider leaving feedback in #117)
- The Bushido head unit (T1982) is (currently) required
- Head unit needs to be turned on and paired to the brake (see Tacx manual)
- Calibration is done on the head unit (see Tacx manual)
- Run FortiusAnt with '-a -g -H0 -t Bushido'
- Pairing to the Bushido is automatic (may take a minute)
- Head unit buttons can be used to control FortiusAnt (keymap is the same as Vortex)

Known issues:

- If power is not transmitted properly, it may help to reset the head unit:
- Settings -> Info -> 10x right arrow (suggested by @tinusvee)
- Virtual shifting is currently not implemented for the Bushido

5.16 Bluetooth (BLE) needs lib_programname, bleno, express (#229)

Reported in issue 229: "I had problems running the latest version did not find it in the documentation. When I ran FortiusAnt version 5.1 on Windows 10 I thought my firewall was blocking the connection to node as it could not connect to server."

I ran the server manually and got the error about bleno and express."

The following commands solved my issue:

```
python -m pip install lib_programname  
npm install bleno  
npm install express
```

After installing the missing packages it worked.

5.17 Bluetooth (BLE) Node.js installation does not install tools.

After Installing Node.js, even though the checkbox to "install the necessary tools for native modules" is selected, no post-installation is done.

No clear reason is found, but after switching off IPv6 on the network controller, installation proceeded.



5.18 Odd behavior in cold garage (#248)

@oldgerian wrote: I suspect the answer lies in your very last point of your 'freezing cold garage'. I am training in a freezing cold cabin in the garden and it took me about a week, 2 new connection cables (usb and motor) 3 or 4 uninstalls and re-installs of FortiusAnt software etc before I realised that the T1932 headset just doesn't want to work after a night in sub zero temperatures.

I now treat it with the respect it deserves and bring it inside at the end of each workout to warm up, along with laptops and any other electrical equipment which doesn't like very cold weather.



6. Power curve validation

6.1 General

As explained [CTP]'s require Power or Grade data and either requirement is translated into a required resistance as explained in the previous section. The calculations were validated during development/testing but it may be useful to check for your own trainer whether the calculations are correct.

“The Power Curve” is the relation between requested Power and the resistance set on the trainer.

The formula basically is: Resistance = $c * \text{Power} / \text{Speed}$; the resistance of the brake goes up with increasing power and reduces with increasing speed (of the bicycle wheel).

Overall conclusion from the tests is that the algorithm does what it should do, which is confirmed by users who are happy with how their trainer works. Individual situations are reported however (e.g. issue 102) that the power curve is not satisfactory. This section can be used to validate settings for your own trainer.

6.2 Test for Tacx Fortius (T1932)

The algorithm is tested on a Tacx Fortius with a T1932 head unit, using a bicycle with power-meter. FortiusAnt is started in manual mode (-m flag) so that power can be selected.

Test protocol.

Select gears for 10, 20, 30, 40, 50 km/hr at reasonable cadence (e.g. 90 rpm).

For each speed perform the following test:

Manually select power 50Watts (up/down button on head unit).

Ride until reading from FortiusAnt and Power Meter is stable.

Write down power from Power Meter (46 in table).

Increase power on head unit 100, 150, 200, 250, 300 Watts and repeat test.

Test Results (Target Power in column header, resultant power in the table)

	50W	100W	150W	200W	250W	300W
10 km/hr	46	97	145	194	245	285
20 km/hr	50	100	145	197	245	290
30 km/hr	63	102	154	196	245	295
40 km/hr	105	120	160	210	260	305
50 km/hr	123	130	165	210	250	310

With the assumptions:

- Attempts to improve the algorithm may be useless, since it would not be more exact than the Tacx Fortius (was designed for).
- Changing the algorithm still requires an empirical result since the technical specs from Tacx are not available to validate

The conclusions are:

- 50Watts at 50km/hr gives odd readings but that is not too strange.
- Overall measured power corresponds with Target Power.
- Multiple measurements give different results within 5%

Tests were undertaken at different times and hence have different brake and tire temperatures.



6.3 Test for i-Magic (T1901-T1902)

@yegorvin has performed a performance calibration test using his iMagic with T1901 magnetic brake, T1902 head unit and a power meter. Note that the T1902 head unit is different from (all) other head units [TotalReverse].

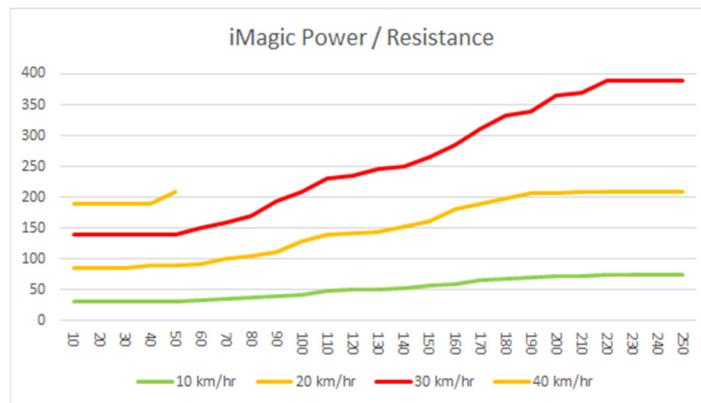
The test is performed as follows:

- FortiusAnt is operated in manual mode (-m flag), modifying the power from 10...250 in 10 Watt steps.
- The power as selected is transferred directly to the iMagic (Resistance = TargetPower) without intermediate formula. (The Python code is changed for this)

Results of the test are:

Res	10km/hr	20km/hr	30km/hr	40km/hr
10	32	85	140	190
20	32	85	140	190
30	32	85	140	190
40	32	89	140	190
50	32	89	140	210
60	34	93	150	
70	35	100	160	
80	38	105	170	
90	40	112	193	
100	42	128	210	310
110	48	139	230	
120	50	142	235	
130	50	145	245	
140	52	152	250	
150	58	162	265	~410
160	60	180	285	
170	65	190	310	
180	68	198	332	
190	70	207	340	
200	72	207	365	~550
210	72	210	370	
220	75	210	390	
230	75	210	390	
240	75	210	390	
250	75	210	390	

The power curve of the iMagic is as shown below, where the range R=50...200 seems quite linear.





Convert Resistance to Power formulae

The assumption is that $P = a * R + b$, when using the target resistance of 50W we have three formulae, one for each speed:

$$\begin{aligned} \text{Speed} = 10: 32 &= 50 a + b \\ 72 &= 200 a + b \implies P_{10} = R * 0.27 + 19 \\ \text{Speed} = 20: 89 &= 50 a + b \\ 207 &= 200 a + b \implies P_{20} = R * 0.78 + 50 \\ \text{Speed} = 30: 140 &= 50 a + b \\ 365 &= 200 a + b \implies P_{30} = R * 1.50 + 65 \end{aligned}$$

And with this, we could create the function as follows:

```
Function Resistance2PowerLegacy(Resistance, Speed) As Double
If Speed = 30 Then
    Resistance2PowerLegacy = Resistance * 1.50 + 65
ElseIf Speed = 20 Then
    Resistance2PowerLegacy = Resistance * 0.78 + 50
ElseIf Speed = 10 Then
    Resistance2PowerLegacy = Resistance * 0.27 + 19
End If
End Function
```

But of course, this would work for three distinct speeds only.

A good formula for the constants 65, 50 and 19 is $2.2 * Speed$ and validation shows that it's good enough.

But there is no linear solution for 1.50, 0.78 and .27, so let's go for $F = ax^2 + bx + c$ (where $x=Speed$)

$$\begin{aligned} 0.27 &= 100 a + 10 b + c \\
0.78 &= 400 a + 20 b + c \\
1.50 &= 900 a + 30 b + c \implies F = 0.001543 \text{Speed}^2 + 0.0001848 * \text{Speed} + 0.1058 \\
&\quad \implies \text{Speed}^2 / 648 + \text{Speed} / 5411 + 0.1058 \end{aligned}$$

and the final formula then becomes:

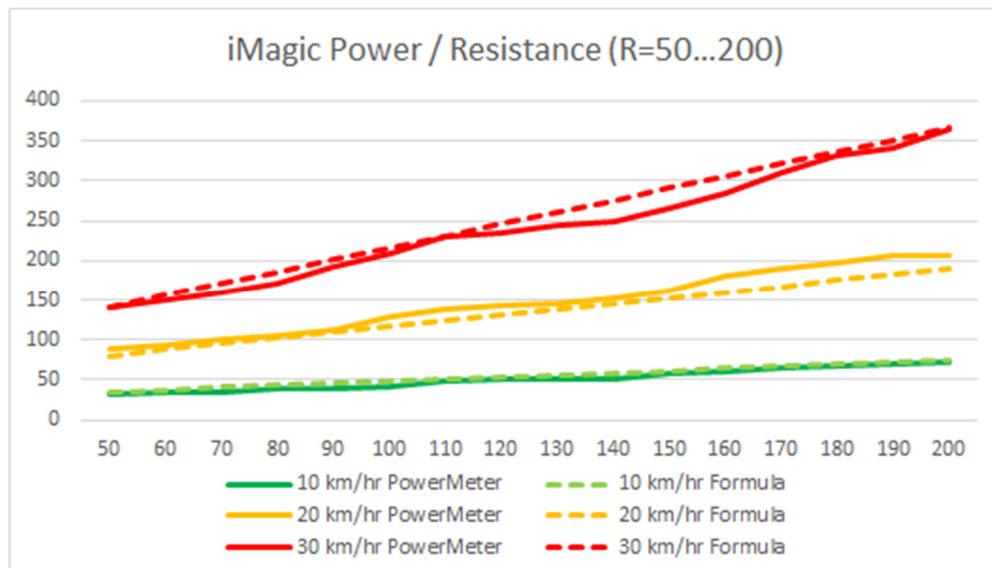
```
Function Resistance2PowerLegacy(Resistance, Speed) As Double
Resistance2PowerLegacy =
    Resistance * (Speed * Speed / 648 + Speed / 5411 + 0.1058) + 2.2 * Speed
End Function
```

To check:

$$\begin{aligned} S = 10, \quad R &= 50; P = 50 * (10 * 10 / 648 + 10 / 5411 + 0.1058) + 2.2 * 10 = 35 \\
R &= 200; P = 200 * (10 * 10 / 648 + 10 / 5411 + 0.1058) + 2.2 * 10 = 74 \\
S = 20, \quad R &= 50; P = 50 * (20 * 20 / 648 + 20 / 5411 + 0.1058) + 2.2 * 20 = 80 \\
R &= 200; P = 200 * (20 * 20 / 648 + 20 / 5411 + 0.1058) + 2.2 * 20 = 190 \\
S = 30, \quad R &= 50; P = 50 * (30 * 30 / 648 + 30 / 5411 + 0.1058) + 2.2 * 30 = 141 \\
R &= 200 * (30 * 30 / 648 + 30 / 5411 + 0.1058) + 2.2 * 30 = 366 \end{aligned}$$



With the Powermeter-measured and Formula-calculated values in one graph the result is as shown as close enough to be used (or at least tested):



Conclusion:

The formula based upon @yegorvin's measurements can be used by the 6 known Tacx i-Magic T1902 users at this moment.

6.4 PowerCurve for i-Flow (T1901-T1932)

At first, FortiusAnt was developed for the Fortius, using the T1932 head unit and the T1941 motor brake. Then the so-called Legacy USB interface was explored (see previous paragraph) using the T1901 magnetic brake and T1902 head unit.

Tacx also sold combinations of the “new” T1932 head unit and the “old” T1901 brake; so that Magnetic brake users could use Virtual Reality CTPs. This combination has a special behavior: instead of a resistance that can be set with a value from 30...226 (magnetic brake) or -5000...+10000 (motor brake) this configuration supports 14 distinct levels.

Experiments show that the following levels are supported (thanks @bikebeppe64 for the tests and @antifier for the table).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
ResistanceLevel	1039	1299	1559	1819	2078	2338	2598	2858	3118	3378	3767	4027	4287	4677
ResistanceSend	1900	2030	2150	2300	2400	2550	2700	2900	3070	3200	3350	3460	3600	3750

The power of the magnetic brake is:

$$\text{Speed} * (\text{ScaleFactor} * \text{Resistance} * \text{Speed} / (\text{Speed} + \text{CriticalSpeed}) + \text{RollingResistance})$$

Where ScaleFactor = 0.013 N, CriticalSpeed = 4.67 m/s and RollingResistance = 15N

Note that:

- **Power = Speed * ScaleFactor * Resistance** is: the braking force due to the magnetic brake: this depends on the resistance setting and the speed. (This is just basic physics; you might even say it is just the definition of power). The ScaleFactor translates the resistance value from the head unit into units of force (unit N = Newton).
- The factor **+ Speed * RollingResistance** is the rolling resistance of the tire, due to the tire deforming continuously as it turns: this would be there even if we removed the magnets in the brake and turned it off. It is largely independent of speed, but can vary a lot with different tire material, size and pressure and the knob setting on the trainer. It is different for everyone and may be different today than it is tomorrow if you forget to pump up your tire.
- The factor **(Speed / (Speed + CriticalSpeed))** the brake cannot deliver the same braking force at all speeds. We assume that as you pedal faster, it increases linearly at first and eventually reaches a constant value (this is an approximation). That is what the factor Speed / (Speed + CriticalSpeed) does. The CriticalSpeed determines where the (gradual) change from linear force to constant force happens. For speeds significantly higher than the critical speed, the factor becomes = 1, so at high speeds the brake force is just ScaleFactor * Resistance value.

These two parameters depend on the geometry of the eddy current brake, the strength and positioning of the magnets,



FortiusAnt - User Manual

the electric current in the coils, the head unit firmware, and so on. We assume they are fixed and more or less the same for all brakes.

This formula has been established by @switchabl and @cyclingflow based upon tests performed by @bikebeppe64, @jürgen-iflow, @mikeherriman and @mk2mark.

Since the Rolling Resistance of the bike depends on external factors (type of bicycle, tire, etc) the Rolling Resistance can be customized with -c CalibrateRR. A figure from 12...18 is probably a reasonable value.

As you know, the environment can operate in PowerMode or GradeMode.

PowerMode

In PowerMode, the requested power is calculated using the formula above. If a required power is known, the resistance is established by searching the appropriate value in the table above. Funny (at least because we do not know why) is that the ResistanceSend is sent to the trainer and then the ResistanceLevel is set.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
ResistanceLevel	1039	1299	1559	1819	2078	2338	2598	2858	3118	3378	3767	4027	4287	4677
ResistanceSend	1900	2030	2150	2300	2400	2550	2700	2900	3070	3200	3350	3460	3600	3750
0 km/hr	0 W	0 W	0 W	0 W	0 W	0 W	0 W	0 W	0 W	0 W	0 W	0 W	0 W	0 W
10 km/hr	57 W	61 W	65 W	69 W	73 W	77 W	81 W	85 W	89 W	93 W	98 W	102 W	106 W	112 W
20 km/hr	129 W	141 W	152 W	164 W	175 W	187 W	198 W	210 W	221 W	233 W	250 W	261 W	273 W	290 W
30 km/hr	207 W	227 W	247 W	268 W	288 W	309 W	329 W	349 W	370 W	390 W	421 W	441 W	461 W	492 W
40 km/hr	286 W	316 W	346 W	376 W	406 W	436 W	466 W	496 W	526 W	556 W	601 W	631 W	661 W	706 W
50 km/hr	368 W	408 W	447 W	487 W	527 W	567 W	607 W	647 W	687 W	726 W	786 W	826 W	866 W	926 W

In the table above, you can see that, when the 2550 is sent to the trainer, the resistance is set to 2338 and the power required is 77, 187, 309, 436, 567 Watt at respectively 10...50 km/hr.

According this table, the trainer can require a resistance of 57W ... 926Watt.

GradeMode

In GradeMode, FortiusAnt first calculates the power that is required to ride the requested grade at the actual speed:

CTP (Swift, Rouvy)	FortiusANT	5 km/hr	10 km/hr	15 km/hr	20 km/hr	25 km/hr	30 km/hr	35 km/hr	40 km/hr	45 km/hr	50 km/hr
-10 %	-10,0 %	-80 W	-195 W	-303 W	-403 W	-491 W	-565 W	-621 W	-657 W	-669 W	-656 W
-7 %	-7,0 %	-43 W	-121 W	-193 W	-256 W	-307 W	-344 W	-363 W	-363 W	-338 W	-288 W
-4 %	-4,0 %	-7 W	-47 W	-82 W	-109 W	-123 W	-123 W	-106 W	-68 W	-7 W	80 W
-1 %	-1,0 %	30 W	26 W	28 W	39 W	61 W	97 W	152 W	226 W	324 W	448 W
2 %	2,0 %	67 W	100 W	138 W	186 W	245 W	318 W	409 W	520 W	655 W	816 W
5 %	5,0 %	104 W	173 W	249 W	333 W	429 W	539 W	667 W	815 W	986 W	1183 W
8 %	8,0 %	140 W	247 W	359 W	480 W	613 W	760 W	924 W	1109 W	1317 W	1551 W
11 %	11,0 %	177 W	320 W	469 W	627 W	796 W	980 W	1182 W	1403 W	1648 W	1919 W
14 %	14,0 %	214 W	394 W	580 W	774 W	980 W	1201 W	1439 W	1698 W	1979 W	2287 W
17 %	17,0 %	251 W	468 W	690 W	921 W	1164 W	1422 W	1697 W	1992 W	2310 W	2655 W
20 %	20,0 %	288 W	541 W	801 W	1069 W	1348 W	1643 W	1954 W	2286 W	2641 W	3023 W

And using the resulting power, the related resistance is calculated (see previous section):

CTP (Swift, Rouvy)	FortiusANT	5 km/hr	10 km/hr	15 km/hr	20 km/hr	25 km/hr	30 km/hr	35 km/hr	40 km/hr	45 km/hr	50 km/hr
-10 %	-10,0 %	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R
-7 %	-7,0 %	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R
-4 %	-4,0 %	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R
-1 %	-1,0 %	2078 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1819 R
2 %	2,0 %	4677 R	4027 R	2858 R	2338 R	2338 R	2598 R	2858 R	3118 R	3767 R	4027 R
5 %	5,0 %	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R
8 %	8,0 %	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R
11 %	11,0 %	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R
14 %	14,0 %	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R
17 %	17,0 %	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R
20 %	20,0 %	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R

Here you can see that, given the nature that negative required power is not possible and that the maximum power is quickly reached, you will get a real feel only between -1% and 5%.

To realize a "real feel" the requested slope can be reduced. By specifying -G50 (Grade = Grade * 50%) the requested grade is reduced to the half and the resistance table becomes:



FortiusAnt - User Manual

CTP (Zwift, Rouvy)	FortiusANT	5 km/hr	10 km/hr	15 km/hr	20 km/hr	25 km/hr	30 km/hr	35 km/hr	40 km/hr	45 km/hr	50 km/hr
-10 %	-5,0 %	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R
-7 %	-3,5 %	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R
-4 %	-2,0 %	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R
-1 %	-0,5 %	3378 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1559 R	2078 R
2 %	1,0 %	4677 R	2338 R	1559 R	1299 R	1559 R	2078 R	2338 R	2858 R	3378 R	
5 %	2,5 %	4677 R	4677 R	3378 R	3118 R	2858 R	3118 R	3378 R	3767 R	4027 R	4677 R
8 %	4,0 %	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R
11 %	5,5 %	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R
14 %	7,0 %	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R
17 %	8,5 %	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R
20 %	10,0 %	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R

And with -G25:

CTP (Zwift, Rouvy)	FortiusANT	5 km/hr	10 km/hr	15 km/hr	20 km/hr	25 km/hr	30 km/hr	35 km/hr	40 km/hr	45 km/hr	50 km/hr
-10 %	-2,5 %	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R
-7 %	-1,8 %	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R
-4 %	-1,0 %	2078 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1819 R	
-1 %	-0,3 %	4027 R	1039 R	1039 R	1039 R	1039 R	1039 R	1039 R	1299 R	1819 R	2338 R
2 %	0,5 %	4677 R	1559 R	1039 R	1039 R	1039 R	1299 R	1559 R	1819 R	2338 R	2858 R
5 %	1,3 %	4677 R	2858 R	1819 R	1559 R	1559 R	1819 R	2078 R	2598 R	3118 R	3378 R
8 %	2,0 %	4677 R	4027 R	2858 R	2338 R	2338 R	2598 R	2858 R	3118 R	3767 R	4027 R
11 %	2,8 %	4677 R	4677 R	3767 R	3378 R	3118 R	3378 R	3767 R	4287 R	4677 R	
14 %	3,5 %	4677 R	4677 R	4677 R	4027 R	4027 R	4027 R	4287 R	4677 R	4677 R	4677 R
17 %	4,3 %	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R
20 %	5,0 %	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R	4677 R

In the examples above, the white area represents the “dynamic area”: at a requested slope of 2% the resistance varies for a speed of 5...50km/hr; at 50km/hr the resistance varies from -10%...+11%.

It is also possible to define an extra reduction for downhill: -G25/50; the grade is then additionally reduced by an extra 50% for grades < zero.

The default is -G100/100 because Zwift has already a standard reduction (similar to -G50, which may be modified in the Zwift settings) and Rouvy has not. It seems that -G50 is a good start when training with Rouvy.



FortiusAnt - User Manual

Tests

When -c 16 (Rolling Resistance) is specified, the curve quite nicely fits the measured figures from [@jurgen-iflow](#):

Power according @switchabl:				Power as measured by @jurgen-iflow			Ratio	
				txcWatts	(Multiplication terms)			
				Avg.tcxWatts	SpeedKm/uur(33,5)			
		Rolling Resistance	16	CurrentResistance	23	33,5	23,0 km/uur	33,5 km/uur
1039		23 km/hr (sw)	158 W	1039	177 Watt	265 Watt	90%	92%
1299		23 km/hr (sw)	173 W	1299	176 Watt	269 Watt	98%	99%
1559		23 km/hr (sw)	187 W	1559	180 Watt	276 Watt	104%	106%
1819		23 km/hr (sw)	201 W	1819	183 Watt	270 Watt	110%	117%
2078		23 km/hr (sw)	215 W	2078	200 Watt	295 Watt	107%	115%
2338		23 km/hr (sw)	229 W	2338	211 Watt	319 Watt	108%	113%
2598		23 km/hr (sw)	243 W	2598	231 Watt	319 Watt	105%	121%
2858		23 km/hr (sw)	257 W	2858	257 Watt	380 Watt	100%	108%
3118		23 km/hr (sw)	271 W	3118	268 Watt	379 Watt	101%	114%
3378		23 km/hr (sw)	285 W	3378	293 Watt	392 Watt	97%	116%
3767		23 km/hr (sw)	306 W	3767	317 Watt		96%	
4027		23 km/hr (sw)	320 W	4027	332 Watt		96%	
4287		23 km/hr (sw)	334 W	4287	347 Watt		96%	
4677		23 km/hr (sw)	355 W	4677	365 Watt		97%	

@switchmcblade

I've been following the thread with interest and as it seemed like you were nearing the end I downloaded this branch 3.9e version to try it out with Zwift.

I don't have a power meter to be sure if it is accurate, but TBH that doesn't bother me as much as being repeatable so I can at least directly compare myself from ride to ride. Previously even a few % change up and down I was having to fiddle with the virtual gearbox to try and get the resistance in an area that I can actually put power down. So I could never be sure from ride to ride if I'd gone up or down the same amount on the virtual gearbox at the same point in the ride.

The new version is so much better, at no point did I feel the need to touch the virtual gearbox. The resistance changes felt natural and I could happily feel like I could manage the resistance changes using the bike gears alone. I'm definitely happy with the experience of installing and riding without worrying about flags or anything else.

Thank you so much to everyone who worked in any way for this improvement, you've transformed my old Flow back into a useful trainer from an ornament in the garage.

And when -c 13 (Rolling Resistance) is specified, the curve quite nicely fits the measured figures from [@mk2mark](#):

Power according @switchabl:				Power as measured by @mk2mark			Ratio	
				txcWatts	SpeedKm/uur			
				Avg.tcxWatts	SpeedKm/uur			
		Rolling Resistance	13	CurrentResistance	12	23	37	12,0 km/uur 23,0 km/uur 37,0 km/uur
1039		12 km/hr (sw)	64 W	1039	65 W	155 W	264 W	100% 90% 91%
1299		12 km/hr (sw)	70 W	1299	62 W	151 W	261 W	113% 102% 103%
1559		12 km/hr (sw)	75 W	1559	64 W	154 W	260 W	116% 108% 114%
1819		12 km/hr (sw)	80 W	1819	68 W	157 W	284 W	118% 115% 114%
2078		12 km/hr (sw)	85 W	2078	70 W	173 W	300 W	122% 113% 117%
2338		12 km/hr (sw)	91 W	2338	71 W	191 W	314 W	127% 110% 120%
2598		12 km/hr (sw)	96 W	2598	81 W	209 W	310 W	118% 107% 130%
2858		12 km/hr (sw)	101 W	2858	86 W	228 W		
3118		12 km/hr (sw)	106 W	3118	92 W	246 W		
3378		12 km/hr (sw)	112 W	3378	94 W	297 W		
3767		12 km/hr (sw)	120 W	3767	107 W	318 W		
4027		12 km/hr (sw)	125 W	4027	116 W	339 W		
4287		12 km/hr (sw)	130 W	4287	121 W	345 W		
4677		12 km/hr (sw)	138 W	4677	120 W	281 W		

@mk2mark

Just to add to the general mood of positivity, and following my successful Zwift Slope mode ride yesterday:-

Today I created a custom workout (ERG mode) testing increasing power levels from 80-450W.

Zwift was driven entirely from FortiusAnt with "-c11.6" set as recommended - power, cadence, speed etc.

I connected bike sensors to Garmin only.

Firstly, the Power output followed the target really well - significantly better than it used to with TTS4 - GREAT RESULT.

The power as measured by Garmin was also close - really good up to about 270W then it started to flatline until I realized that since 200W I had been using a constant gear & cadence and had reached the limit of the brake at that speed. I was surprised at how quickly I hit this limitation of the hardware, so worth keeping in mind that the T1901 brake needs high wheel speed to get the higher power outputs.

Once I upped the wheel speed the power then tracked ok - started to underread from 410W as opposed to over yesterday so I may need to tweak the -c flag a bit. However, in my practical 'working' range of 150-350W it was really good.

I am a very happy user. Thanks so much everyone for what you have achieved.

Conclusion: The purpose of FortiusAnt is to enable "old" Tacx trainers to be used with CTPs like Zwift and Rouvy providing a more or less realistic experience. And for that purpose, +/- 10% is considered acceptable. For more details; check <https://github.com/WouterJD/FortiusAnt/issues/153>