The driver HIDUSBF and its associated tools, developed by Sweetlow, allow you to overclock the polling of your GCC to USB adapter by your PC to 1000Hz. By default, the PC polls it at 125Hz (every 8 milliseconds).

This should result in a much closer experience to console play. It considerably improves the latency stability, which directly impacts execution, and also reduces input lag - significantly so, in the case of the Mayflash (~3.5ms for the official adapter, 14ms for the Mayflash adapter).

Additionally, as surprising as it may sound, not all ports, ports combinations of the adapters (and not all adapters) work equally well. See end of document.

All PC Melee players should overclock their adapter and follow the port-related advice so as to have the best possible experience.

Adapter overclocking

Windows

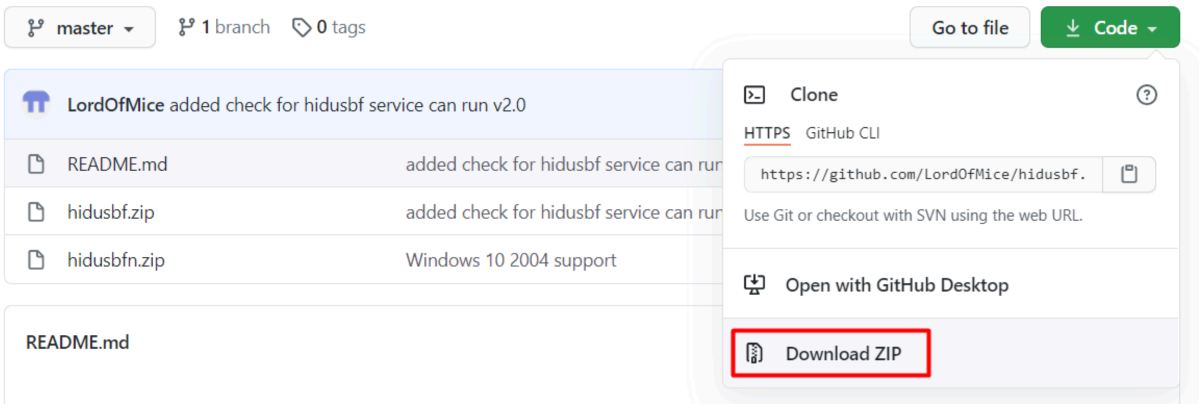
Have your GCC to USB adapter (in WiiU/Switch mode if Mayflash) (aka the WUP-028) plugged in for this process.

Ensure you don't have a Dolphin/Slippi open.

Check out the Troubleshooting section of this document if you encounter a problem.

What you have to interact with is highlighted with a red box in the following images.

Head to https://github.com/LordOfMice/hidusbf



LordOfMice added check for hidusbf service can run v2.0

README.md added check for hidusbf service can run

hidusbf.zip added check for hidusbf service can run

hidusbfn.zip Windows 10 2004 support

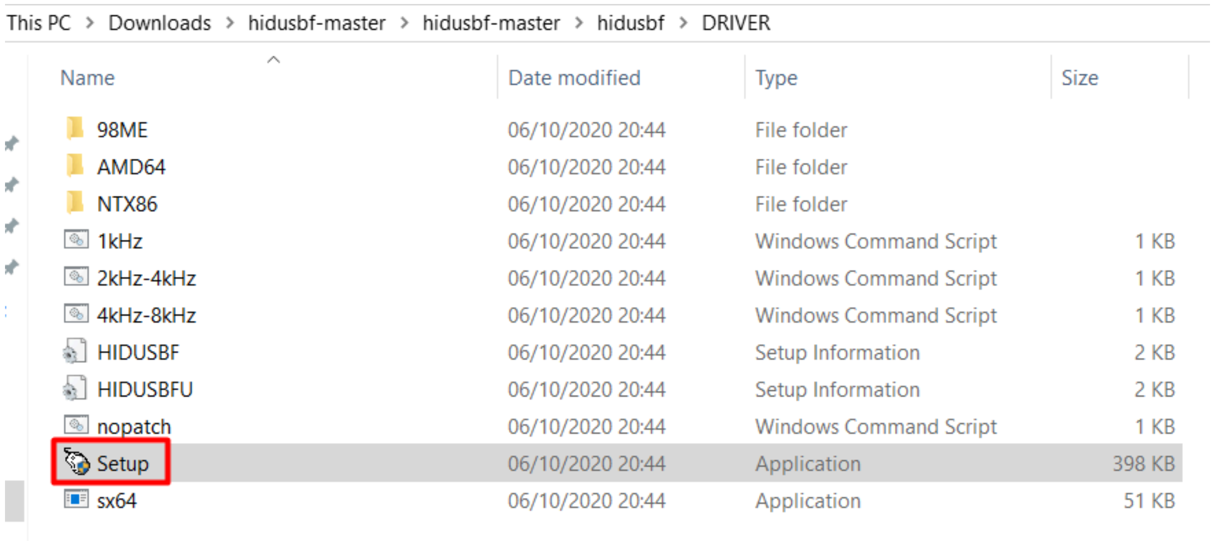
https://github.com/LordOfMice/hidusbf.

Use Git or checkout with SVN using the web URL.

Download the project as a zip file.

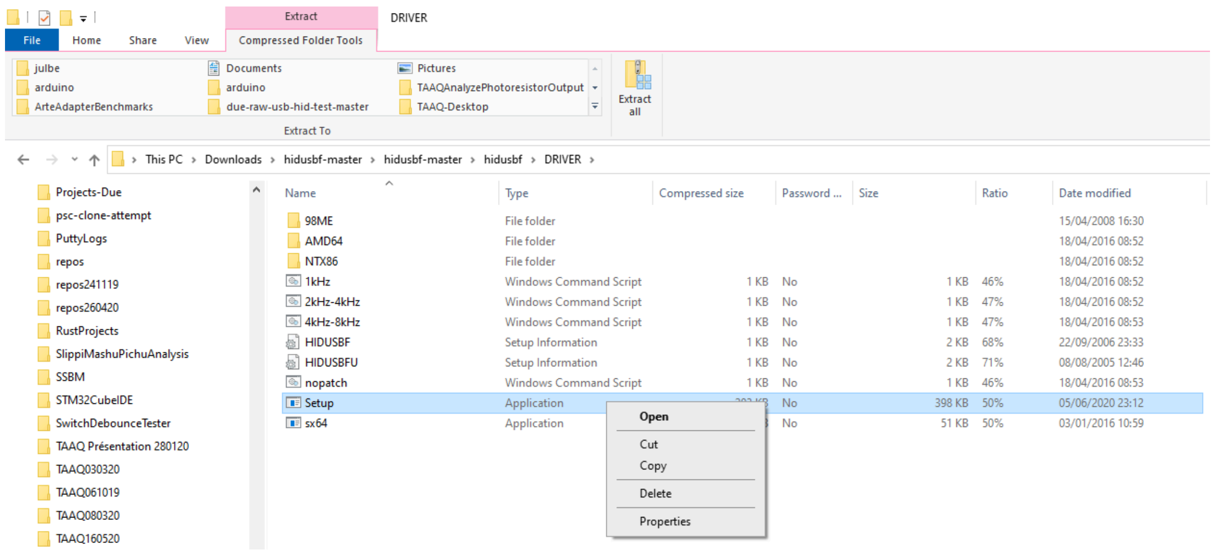
Extract the hidusbf-master compressed folder. Inside it, you should find two compressed folders, hidusbf and hidusbfn. Extract the hidusbf folder.

It should look like this:


This PC › Downloads › hidusbf-master › hidusbf-master › hidusbf › DRIVER

Name	Date modified	Type	Size
98ME	06/10/2020 20:44	File folder	
AMD64	06/10/2020 20:44	File folder	
NTX86	06/10/2020 20:44	File folder	
1kHz	06/10/2020 20:44	Windows Command Script	1 KB
2kHz-4kHz	06/10/2020 20:44	Windows Command Script	1 KB
4kHz-8kHz	06/10/2020 20:44	Windows Command Script	1 KB
HIDUSBF	06/10/2020 20:44	Setup Information	2 KB
HIDUSBFU	06/10/2020 20:44	Setup Information	2 KB
nopatch	06/10/2020 20:44	Windows Command Script	1 KB
Setup	06/10/2020 20:44	Application	398 KB
sx64	06/10/2020 20:44	Application	51 KB

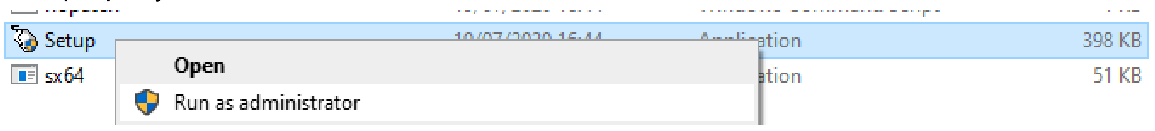
NOT like this:



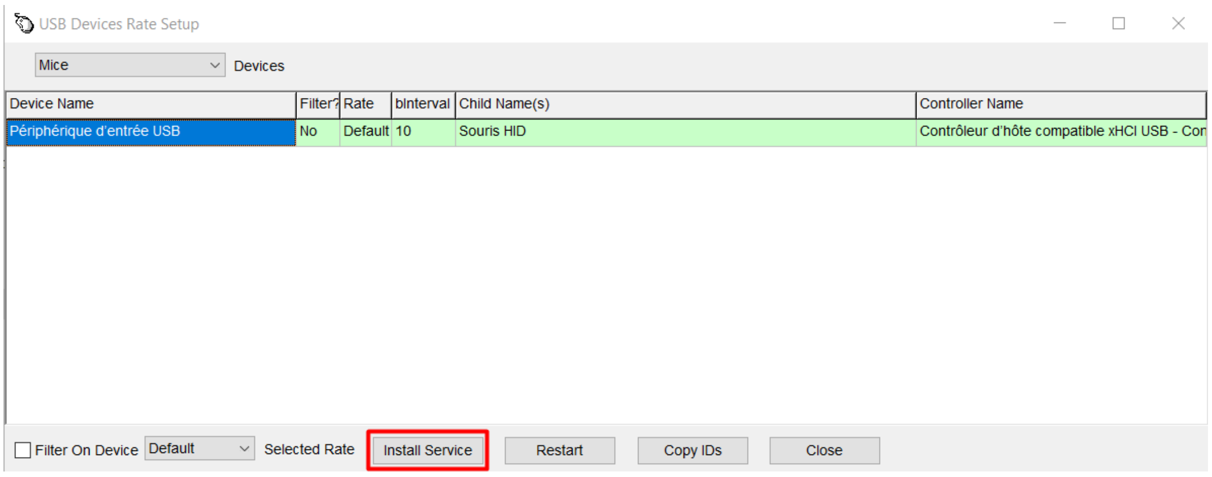
This PC › Downloads › hidusbf-master › hidusbf-master › hidusbf › DRIVER ›

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
98ME	File folder					15/04/2008 16:30
AMD64	File folder					18/04/2016 08:52
NTX86	File folder					18/04/2016 08:52
1kHz	Windows Command Script	1 KB	No	1 KB	46%	18/04/2016 08:52
2kHz-4kHz	Windows Command Script	1 KB	No	1 KB	47%	18/04/2016 08:52
4kHz-8kHz	Windows Command Script	1 KB	No	1 KB	47%	18/04/2016 08:53
HIDUSBF	Setup Information	1 KB	No	2 KB	68%	22/09/2006 23:33
HIDUSBFU	Setup Information	1 KB	No	2 KB	71%	08/08/2005 12:46
nopatch	Windows Command Script	1 KB	No	1 KB	46%	18/04/2016 08:53
Setup	Application	[illegible]	No	398 KB	50%	05/06/2020 23:12
sx64	Application	[illegible]	No	51 KB	50%	03/01/2016 10:59

Follow the path as indicated in the image (hidusbf-master/hidusbf/DRIVER), right click Setup.exe, select “Run as administrator”. If the option doesn’t appear, you didn’t extract the folder properly.



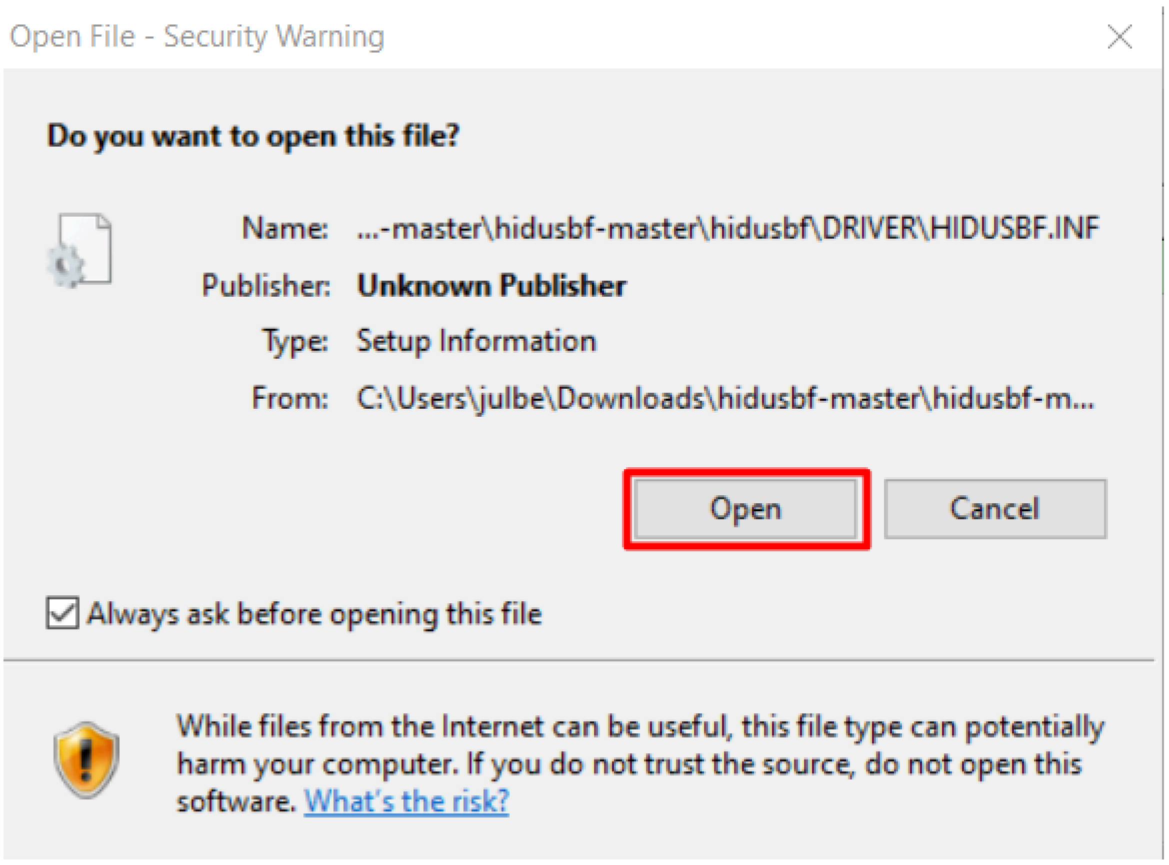
.


USB Devices Rate Setup

Mice Devices

Device Name	Filter?	Rate	bInterval	Child Name(s)	Controller Name
Périphérique d'entrée USB	No	Default	10	Souris HID	Contrôleur d'hôte compatible xHCI USB - Con

Click “Install Service”.
The service installation is unrelated to what device you currently have selected, you only need to do it once. Don’t worry over the devices listed on your screen differing from that of this example image (i.e a mouse).

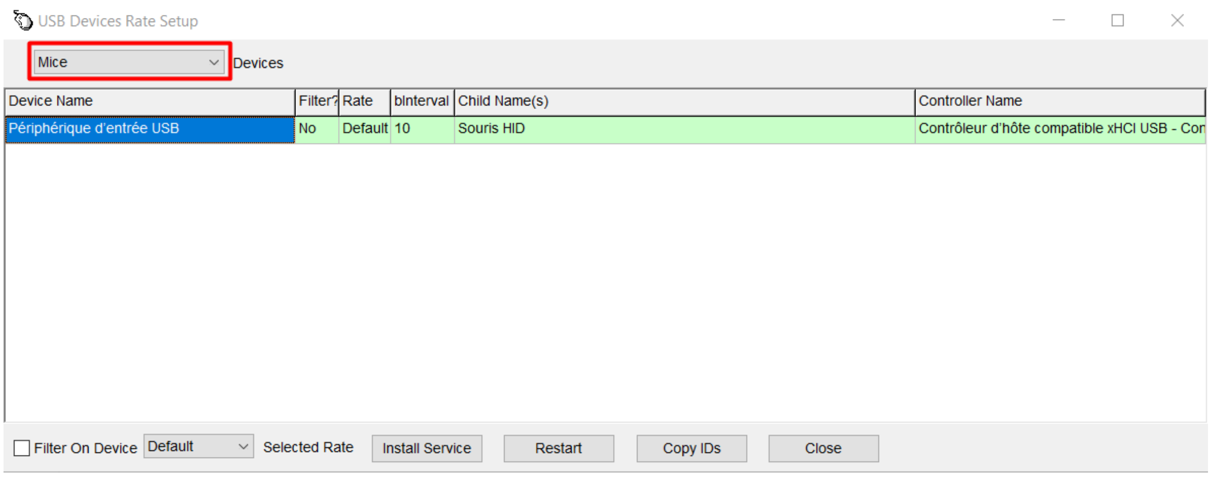

Open File - Security Warning

Do you want to open this file?

Name: ...-master\hidusbf-master\hidusbf\DRIVER\HIDUSBF.INF
Publisher: Unknown Publisher
Type: Setup Information
From: C:\Users\julbe\Downloads\hidusbf-master\hidusbf-m...

While files from the Internet can be useful, this file type can potentially harm your computer. If you do not trust the source, do not open this software. What's the risk?

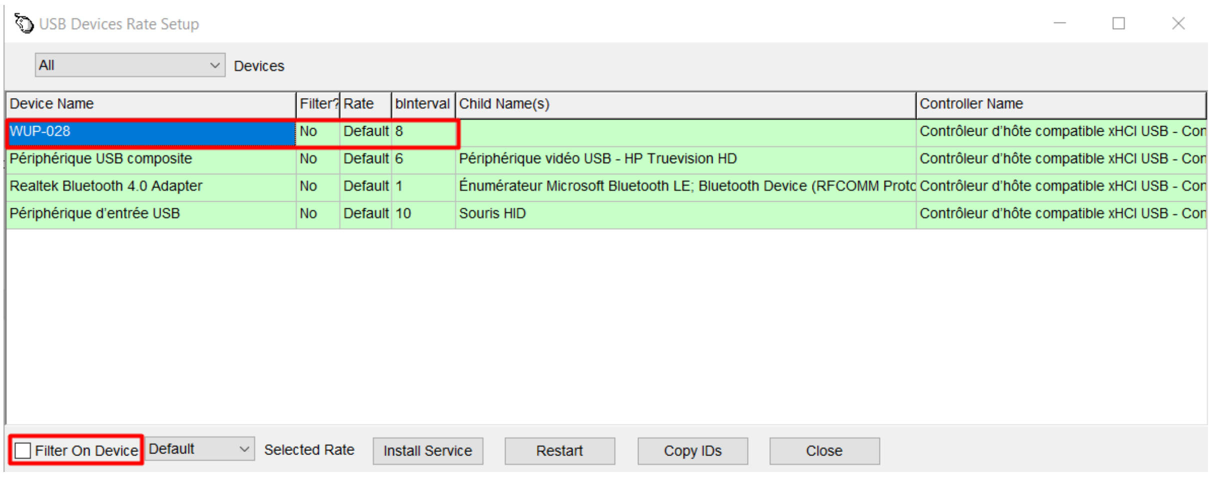
Open the file when asked if you’re asked to. (Some people aren’t, in that case, just move on)


USB Devices Rate Setup

Mice Devices

Device Name	Filter?	Rate	bInterval	Child Name(s)	Controller Name
Périphérique d'entrée USB	No	Default	10	Souris HID	Contrôleur d'hôte compatible xHCI USB - Con

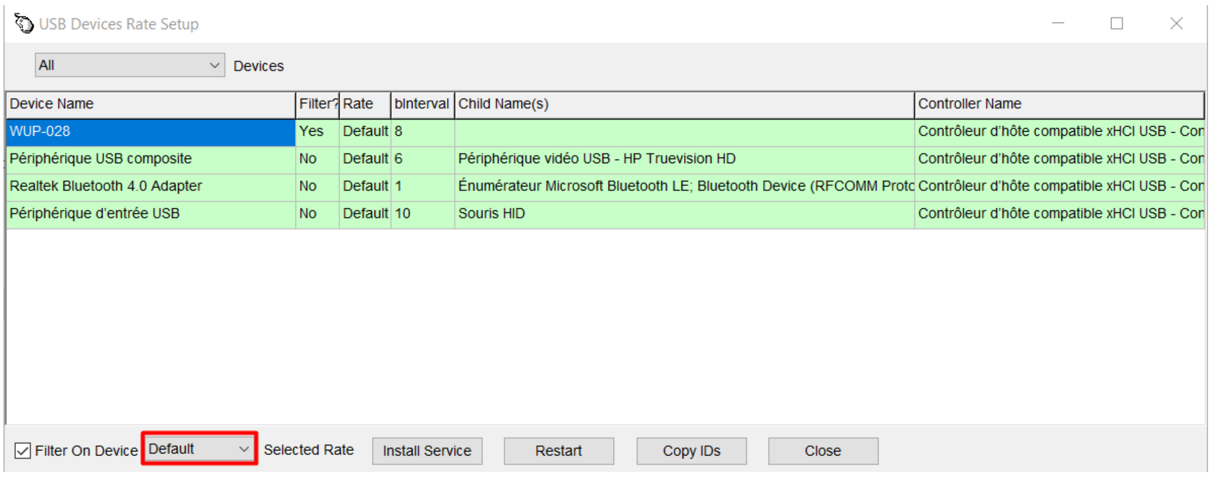
Select “All” in the Devices drop-down menu.


USB Devices Rate Setup

All Devices

Device Name	Filter?	Rate	bInterval	Child Name(s)	Controller Name
WUP-028	No	Default	8		Contrôleur d'hôte compatible xHCI USB - Con
Périphérique USB composite	No	Default	6	Périphérique vidéo USB - HP Truevision HD	Contrôleur d'hôte compatible xHCI USB - Con
Realtek Bluetooth 4.0 Adapter	No	Default	1	Énumérateur Microsoft Bluetooth LE; Bluetooth Device (RFCOMM Protc	Contrôleur d'hôte compatible xHCI USB - Con
Périphérique d'entrée USB	No	Default	10	Souris HID	Contrôleur d'hôte compatible xHCI USB - Con

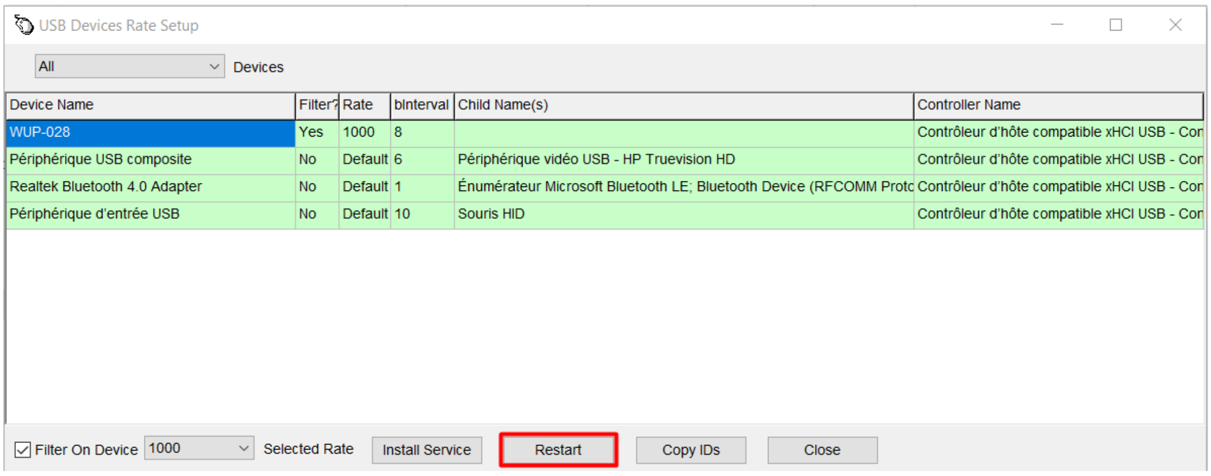
Select WUP-028, then toggle “Filter on Device”.


USB Devices Rate Setup

All Devices

Device Name	Filter?	Rate	bInterval	Child Name(s)	Controller Name
WUP-028	Yes	Default	8		Contrôleur d'hôte compatible xHCI USB - Con
Périphérique USB composite	No	Default	6	Périphérique vidéo USB - HP Truevision HD	Contrôleur d'hôte compatible xHCI USB - Con
Realtek Bluetooth 4.0 Adapter	No	Default	1	Énumérateur Microsoft Bluetooth LE; Bluetooth Device (RFCOMM Protc	Contrôleur d'hôte compatible xHCI USB - Con
Périphérique d'entrée USB	No	Default	10	Souris HID	Contrôleur d'hôte compatible xHCI USB - Con

Then, choose “1000” in the Selected Rate drop-down menu. If you’re using a Hitbox adapter, pick “250”.

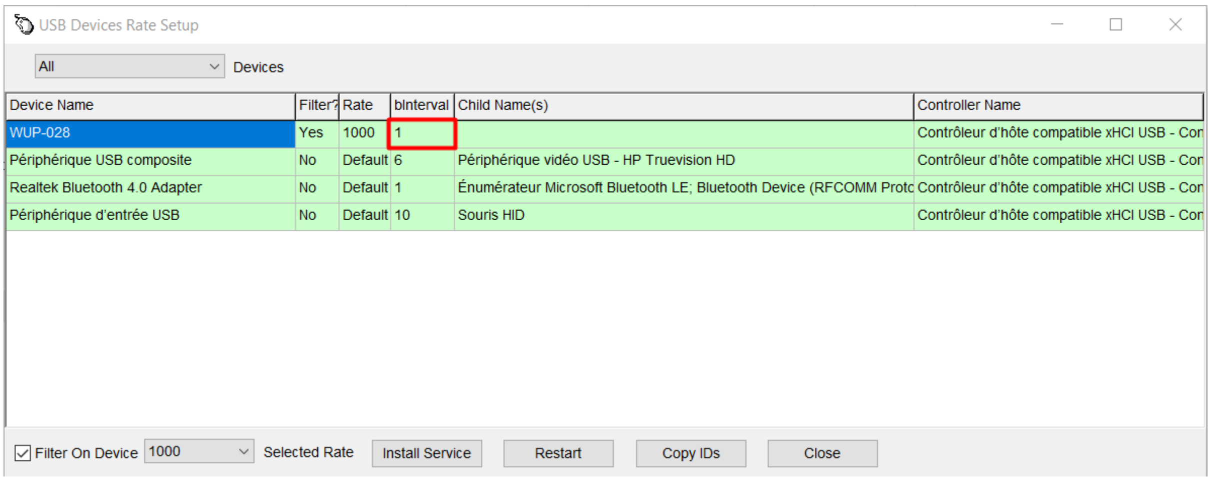

USB Devices Rate Setup

All ▾ Devices

Device Name	Filter?	Rate	bInterval	Child Name(s)	Controller Name
WUP-028	Yes	1000	8		Contrôleur d'hôte compatible xHCI USB - Cor
Périphérique USB composite	No	Default	6	Périphérique vidéo USB - HP Truevision HD	Contrôleur d'hôte compatible xHCI USB - Cor
Realtek Bluetooth 4.0 Adapter	No	Default	1	Énumérateur Microsoft Bluetooth LE; Bluetooth Device (RFCOMM Protc	Contrôleur d'hôte compatible xHCI USB - Cor
Périphérique d'entrée USB	No	Default	10	Souris HID	Contrôleur d'hôte compatible xHCI USB - Cor

☑ Filter On Device 1000 ▾ Selected Rate

Click “Restart”. (It restarts device communication, not your computer)
If you get an “Error[...] 577” message, or the WUP-028 line turns red at this step, check https://docs.google.com/document/d/1cQ3pbKZm_yUtcLK9ZIXyPzVbTJkvnfxKIyvuFMwzWe0/edit#bookmark=id.pgm54yf22iz

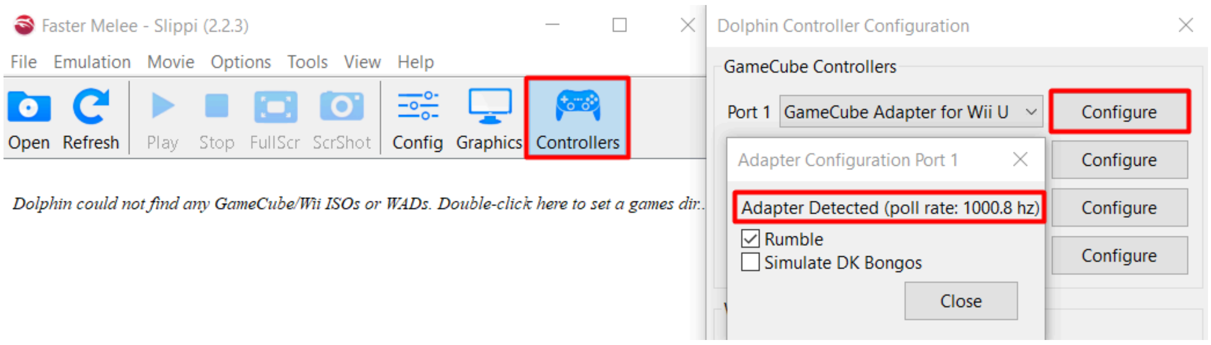

USB Devices Rate Setup

All ▾ Devices

Device Name	Filter?	Rate	bInterval	Child Name(s)	Controller Name
WUP-028	Yes	1000	1		Contrôleur d'hôte compatible xHCI USB - Cor
Périphérique USB composite	No	Default	6	Périphérique vidéo USB - HP Truevision HD	Contrôleur d'hôte compatible xHCI USB - Cor
Realtek Bluetooth 4.0 Adapter	No	Default	1	Énumérateur Microsoft Bluetooth LE; Bluetooth Device (RFCOMM Protc	Contrôleur d'hôte compatible xHCI USB - Cor
Périphérique d'entrée USB	No	Default	10	Souris HID	Contrôleur d'hôte compatible xHCI USB - Cor

☑ Filter On Device 1000 ▾ Selected Rate

You should see the “bInterval” on the line of the WUP-028 set to 1.


Faster Melee - Slippi (2.2.3)

File Emulation Movie Options Tools View Help
Open Refresh Play Stop FullScr ScrShot Config Graphics Controllers

Dolphin could not find any GameCube/Wii ISOs or WADs. Double-click here to set a games dir.

Dolphin Controller Configuration

GameCube Controllers

Port 1 GameCube Adapter for Wii U ▾

Adapter Configuration Port 1

Adapter Detected (poll rate: 1000.8 hz)

☑ Rumble
☐ Simulate DK Bongos

Configure
Configure
Configure
Configure
Close

Start Slippi, open the Controllers panel, click Configure (assuming you’ve selected GameCube Adapter for Wii U beforehand) and check the poll rate. If you’re reading 1000±100 hz i.e a varying rate between 900 and 1100, you’re good. The variance is expected. If it’s still ~125Hz, something went wrong.

It is fine to have some 500Hz readings in this panel. You’re expected to have at least 17% ~500Hz readings if you are using the official adapter with a single port in use. If you have

two, you should have few but having some doesn't necessarily mean things are going wrong, especially if your CPU is busy with other things.
With the Mayflash adapter you're expected to have at least 8% ~500Hz readings.
For other adapters, see "Averaging something stable between 125Hz and 1000Hz" in Troubleshooting.

Credits / some technical information:

HIDUSBF has been developed by **Sweetlow** since 2004, originally for the overclocking of mice or keyboards and was enriched in 2016 with the experimental ability to overclock non-HID devices (a specific kind of USB devices whom mice or keyboards belong to - but not GCC to USB WiiU/Switch mode adapters). All the technical credit goes to him.

Note that HIDUSBF is a filter driver. It complements the main driver (WinUSB), it doesn't replace it. So the Zadig procedure about installing WinUSB for the WUP-028 is unrelated, and still necessary.
Finally, I felt like I should clarify that none of this is illegal / dangerous for your device / undefined behaviour or anything of the sort. "Overclocking" a USB device is specification compliant. USB devices say to the PC "I would like to be polled at least every X ms when I'm in use." Polling less often than that is not specification compliant. Polling more is completely fine. So it should be "overclocking the communication frequency of your PC with the adapter" (which is too long) and "overclocking your adapter" is somewhat of a misleading name: we're not overclocking anything inside the device, nothing changes inside the device, we're just polling it faster than it asked for. Operating systems by default poll devices as rarely as they're allowed to so as to preserve USB bandwidth. Then depending on the philosophy of the OS, the difficulty of explaining to the OS you don't want to poll as seldom as possible differs.

On Windows it is very not straight-forward and I haven't heard of a way to do it besides Sweetlow's.

For Linux and MacOS, see the respective parts.

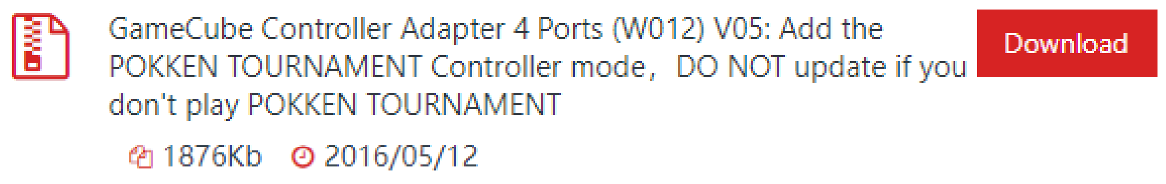
Troubleshooting

500Hz with the Mayflash 4 ports

The Mayflash 4 ports adapters have 2 common firmwares in the wild: v5 and v6.
v5 is better than v6, making the adapter poll the controllers at ~375Hz instead of ~250Hz, so v5 is more stable latency-wise (and very likely also has less latency).

You can upgrade/downgrade firmwares using installers from the Mayflash website. In our case we want to downgrade to v5 in case you're v6 (or upgrade if you're in the rare case of having firmware < v5). Attempting to downgrade to v5 will let you know whether you already had v5 - in that case, nothing to do.

Mayflash download page: [Download]


GameCube Controller Adapter 4 Ports (W012) V05: Add the POKKEN TOURNAMENT Controller mode, DO NOT update if you don't play POKKEN TOURNAMENT

v5 firmware direct download
Mirror

You need to be on Windows, and put the adapter in PC mode for firmware changes.

With v5 and one controller plugged in, you should see most of the time 1000Hz reads in Slippi and sometimes 500Hz. With v6, 500Hz nearly always.

v7 performs as v6, but can't be downgraded from, so be sure not to update to that.

Note that, contrary to the overclocking driver which changes how your PC interacts with the adapter, this modifies the behaviour of the adapter itself (so, a v6->v5 firmware downgrade also most likely saves ~0.67ms of input lag when used on the Switch, whereas the 125Hz->1000Hz overclock saves 3.5ms of input lag on PC but none on the Switch)

If you're confused about, and interested by why you'd see 500Hz/1000Hz in the Slippi Controllers tab when I just said v5 polls at ~375Hz and v6 at ~250Hz, check the FAQ.

Note that v5 and v6 are supposedly aimed at PC mode for Windows/Linux and MacOS respectively (even though it apparently shadow drops a behaviour change for native mode). If you want to use your Mayflash adapter's PC mode and encounter problems, it might be because you're using the firmware for Mac on PC or the opposite, switch the firmware to the right one (v5 Windows/Linux, v6 MacOS).

Averaging <1000Hz >=125Hz

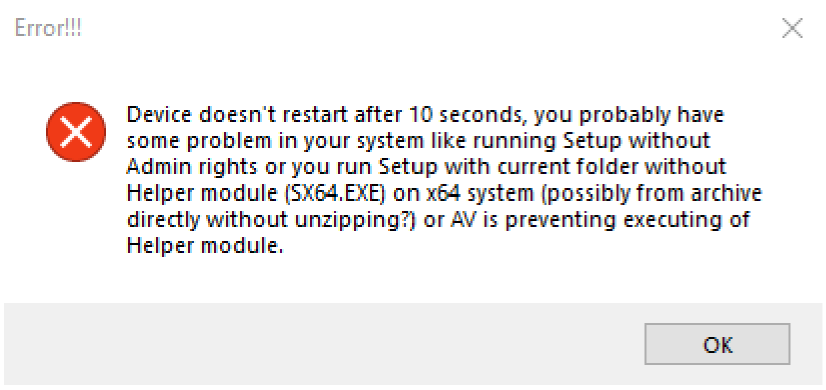
Assuming you're using another adapter than the Mayflash and the official one, getting a perceived poll rate between 125Hz and 1000Hz may be expected.
Overclocking the adapter allows it to provide data at a 1000Hz rate, but if by default it wouldn't be able to, you're going to see a lower rate. That's on the adapter not polling the controllers often enough.

Expected rate:
- Mayflash 2 ports (Wii U compatible): 500Hz
- Smash.gg: 166.7Hz
- Yteam: 125Hz
- pdobq: 125Hz

Need confirmation:
- Yccteam: 125Hz
- CLOUDREAM: 125Hz
- Nyko: 125Hz

Credits to Nikki for testing some of these third party adapters.

“Device doesn’t restart after 10 seconds [...]”


Error!!!

Device doesn't restart after 10 seconds, you probably have some problem in your system like running Setup without Admin rights or you run Setup with current folder without Helper module (SX64.EXE) on x64 system (possibly from archive directly without unzipping?) or AV is preventing executing of Helper module.

Ensure you don’t have any Dolphin/Slippi open, that the folder you’re launching Setup.exe from was extracted and that you launched Setup.exe with administrator rights.
If the “open with admin rights” option doesn’t appear when you right click, you haven’t extracted the folder, or you don’t have admin rights period (in which case you can’t proceed).

If you’re using a USB hub / USB extension card, connect the adapter directly and try again.

“Error [...]: 577”

Download and launch this script, then reboot your computer. Should work on all modern Windows (including 11).

See “FAQ/What does allow_overclock.reg do?” for explanations.

Old method: disabling Secure Boot in the BIOS

Follow the step 1 of solution 2 of the following guide:
https://www.wintips.org/how-to-fix-windows-cannot-verify-the-digital-signature-for-this-file-error-in-windows-8-7-vista/#solution-2
In the case of Full Speed devices, which our adapters belong to, only step 1 i.e disabling Secure Boot should be required.

You can also use solution 1, but when you restart your computer, the overclock will probably stop working. In case things still work (you’re still at 1kHz) after a restart following solution 1, please let me know.

Disabling Secure Boot is done in the BIOS of your computer, which usually requires restarting and mashing a given F[1-12] key. You can look up how to access your BIOS for your given motherboard/laptop brand online if you don’t know how to.

If disabling Secure Boot indeed got it to work, turning it on again will break the driver. It’s necessary to the operation of the driver, not just its installation. If you have an ASUS motherboard/laptop, chances are you won’t be able to disable Secure Boot outright in the BIOS. Check the next troubleshooting entry.

Secure Boot only blocks unsigned drivers from executing - it isn’t a critical security measure.

If you have issues with anti-cheats after using solution 2 part >1, follow the “How to enable Driver Signature Enforcement” part (ignore point 4) to switch back.

Red line when attempting to Filter/restart

A red line means any problem that’s silent for some reason. Your best bet if you’re not using a hub is to follow the procedure for error 577 as described above.

Error 193

Go to the Core isolation page in Windows Security, turn the Memory integrity setting off, then restart the computer.

Source:
https://support.microsoft.com/en-us/windows/a-driver-can-t-load-on-this-device-8eea34e5-ff4b-16ec-870d-61a4a43b3dd5

Can’t disable Secure Boot on ASUS motherboards

ASUS motherboards’ Secure Boot disabling is different - you have to “Clear Secure Boot keys”.

Please follow the instructions there:
https://www.qualityology.com/tech/disable-asus-motherboards-uefi-secure-boot/

“Error [...]: 1060 / No filter driver found after clicking Install Service with no error”

Ensure you extracted the folders first instead of opening the compressed folders.
Right click HIDUSBF.INF in the same folder as Setup.exe -> Install.

USB Hubs

USB hubs may cause issues, consider trying without using one if you’ve encountered problems and have been using one. Note that some hubs are internal, so as a last resort, try different ports.

Rolling back everything

If things stop working and you want to revert everything, reinstall WinUSB as the driver of the WUP-028 through Zadig (yes, even if it shows it’s already using WinUSB). Things should get back to how they are by default.

Polling rate above 1000Hz in the Slippi Controllers tab

A polling rate significantly above 1000Hz (the adapter won’t work in-game) can be caused by unplugs/replugs of the adapter while Slippi is open. Try relaunching Slippi.

Other error codes

If you haven’t disabled Secure Boot yet, try that.

Linux

On Linux overclocking your adapter is unsurprisingly more accessible, but you won’t even need to look up the command lines, because **HannesMan** has made a kernel module for it.

HannesMann/gcadapter-oc-kmod: Kernel module for overclocking the Nintendo Wii U/Mayflash GameCube adapter.

Follow the instructions provided - he even included a video. Note that the latest versions overclock by default to 1000Hz (contrary to what’s shown in the video).

Steam Deck

Installing the module on Steam Deck involves extra steps. **linuxgamingcentral** made a bash script to ease its installation, which you can find here. Thanks to them !

MacOS

knux has made a MacOS kernel extension for overclocking adapters based on HannesMann’s.
Check: https://github.com/secretkeysio/GCAdapterDriver/releases

Regarding digital controllers

If you’re using a digital controller through the adapter (as opposed to plugging it directly through USB for those that support this), overclocking the adapter may cause problems where you experience random inputs (to be updated with how main digital controllers behave).
In this case, you cannot fully overclock. Revert to Default/125Hz in the HIDUSBF setup tool. You can try incrementally increasing the polling rate (250Hz / 500Hz) until you see random inputs happen and then revert to the highest frequency that works fine.

Not all ports are made equal

As surprising as it may sound, some ports work better than others on the Official and Mayflash adapters. I only know about the ports for these two, that’s not to say other adapters don’t have similar oddities.
Apply the following advice for the best experience. Unless specified otherwise, unplug idle controllers when playing.

Mayflash adapter (4 ports, WiiU/Switch mode)

Note: this document is now several years old and referred to the Mayflash v5 adapter firmware.
We’re now on v10 and v7 and less don’t work on Switch past firmware version v19.0.0 (that initially caused every Mayflash port to control 2 controllers). The comment below may be outdated.

Historically:
Single player: use port 2. Don’t plug anything else in.
Doubles: use ports 1 and 2.

Official adapter

Single player: if you have a spare controller, plug it into port 1. Plug yours into port 2 and use it. I am dead serious.
The official adapter appears to have a special mode that makes it poll the controller more frequently when ports 1 and 2 are plugged in, no less, no more. Under normal operation the official adapter will poll controllers every 1.19971ms. When ports 1&2 are plugged and 3&4 are unplugged, it will poll them every ~0.9995ms (which is better as it's nearly aligned with USB's 1000Hz polling). Setting port 1 as "None" isn't necessary. If you don't, you'll just have a P1 pointer but you can still access any single player mode as P2.

If you don't have a spare controller, use port 1 (or 2, or 3, no difference. Don't use port 4. It malfunctions.)

Doubles: use ports 1 and 2.

Lossless adapter

Doesn't matter.

Which adapter is preferable ?

The Lossless adapter is the best both in latency, latency stability and UX (plug and play, no OC required on Windows, ports work the same). Also, it tests lag. Disclaimer, I made it.

The official adapter has less latency volatility than Mayflash due to polling the controller faster (~1.2ms instead of ~2.65ms). If you have a choice between the two with all other parameters equal, prefer this one. Note that the difference between the two adapters is small compared to the gain provided by 125Hz to 1000Hz overclocking.

The Raphnet is about as good as the official one on Windows, but I lack test data to be able to say for sure on other OSs. I'm not knowledgeable about differences induced by Standard Controller vs Native Control.
In terms of polling it should be as good as the official with double ports trick if not slightly better (i.e the best) however small analog input value differences may occur due to the different protocol. See:
https://docs.google.com/document/d/1Vspg-hGtooDud0zwRkES1RZZeBKEYsC0xzv6R0NWrFg/edit?usp=sharing
If you want to use your Gamecube controller to play other games than Dolphin on PC, the Raphnet adapter is the best one at that task to my knowledge. But, you can't use it on Switch.

Using port 2 in a Netplay lobby

Use the “Assign controller ports” button to set port 2 as a Player. Note you don’t need to do this nowadays as we now use Slippi Online, not Dolphin netplay lobbies.

FAQ

Console only polls at 120Hz, is it too good now ?

No it’s not.

The “125Hz/1000Hz” discussed here and the “120Hz” in the question refer to different things. The console, and thereby the emulator, run the game at 60Hz i.e 60 increments of the game state per second. This means that 60 times per second, the controller “state” is read. (For correctness, 59.94Hz on console and emulator respectively with tauKhan’s polling drift fix and Dan Salvato/tauKhan’s associated gecko code “Polling drift fix + VB”)

In a perfect world, the game engine asks the controllers directly for their state in an instant and gets a response in an instant, whenever they need them. This is not what happens. For technical reasons, it is undesirable to have the game engine perform blocking communications directly with external entities.

Because of this, there is a machinery that comes before the controller states are exposed to the game engine - before from the perspective of the journey of controller data. It handles polling the controllers and fills a data buffer with the latest controller data it knows of, that the game engine can read instantly at leisure.
From an input latency volatility point of view, this buffer and everything that comes with it are a **nuisance**. But, they’re necessary.
Problems happen in proportion to how much this buffer’s content differs from that of the controller. If they are the same but only shifted by a given amount of time, there are no problems (beside that amount of time contributing to input latency).
But, they’re not. That’s because the “controller state” evolves continuously. Your fingers move in real-time, so does the stick position, so does the electrical state mapped to the stick position. They don’t update every millisecond only. On the other hand, the PC/Gamecube’s CPU has other stuff to do than update the controller state every nanosecond (it wouldn’t be able to anyway, a GCC communication takes 350~400us). So, it updates it periodically.

In the case of the console, it updates it on average at 119.88Hz - the “120Hz” you may have heard about.
On PC, the adapter polls at whatever frequency it’s polling at - this is the first stop of controller data - and then, the PC polls the adapter as per the USB protocol. By default it polls it at 125Hz at the demand of the adapter. This overclock tells the PC to poll it at 1000Hz. It’s then stored in a buffer - second stop - and that’s the one Slippi reads. Also, the emulator is smart: you’d expect the 119.88Hz average periodic polling to still be there, but it

has been optimized away. This is the “polling method: on SI Read” that you see when you view the properties of the game (the other option being “Accurate to console”).

So note who’s responsible for what here:
- The “controller->adapter buffer” polling rate is decided by the adapter. You’ll only change it through firmware updates / studying the adapter behavior and finding ports stuff as for the official one.
- The “adapter->host buffer” polling rate is enforced by the host (PC, Switch, etc.) at the demand of the adapter. You need control over the host to change it, and the difficulty of doing so varies with the OS.

Now for questions of “how good is it”, it is less simple than “bigger number of Hz = better”. You have no input latency volatility problems if the intermediate periodic polling frequencies are “aligned” on that of the game engine, i.e, a multiple of it (60Hz here). A periodic polling aligned on the frequency of the one that matters (the game engine’s, 60Hz) at worse delays all inputs - by a given amount, thereby yielding the scenario from earlier (all inputs shifted, no volatility issues)
Being close to such a multiple is still fine but things get bad fast as you stray from it, the associated phenomenon being called “polling drift”.
120Hz periodic polling would be perfect for a 60Hz game engine. 119.88Hz average periodic polling is close enough to still be good. 125Hz is not.
Unfortunately USB periodic polling can’t be made 120Hz, or any multiple of 60Hz for that matter, because USB communication resolves around one millisecond “frames”. You can communicate every 1ms, 2ms, 4ms, 8ms… (for USB 2 full speed, also 0.125ms 0.25ms 0.5ms for “USB 2 high speed”, a more advanced USB mode that our adapters don’t support)
And since 1ms doesn’t divide 16.67ms, we’re fucked. Polling at a multiple of 60Hz is the best option, but since we can’t have it, we have to settle for the second best option: brute force. i.e, polling as fast as possible. If we poll very fast every step of the way, the timeline of the buffer exposed to the game engine will roughly map the timeline of the controller, and keep the latency volatility low. That’s the aim of this overclock and the aim of using a port configuration that makes the adapter’s polling as fast as possible.

Comparing the console’s 120Hz polling to our 1000Hz adapter to PC buffer (which is only the second step) makes no sense.
What makes sense is comparing the damage induced by the console’s 119.88Hz on-average periodic polling straying slightly from 120Hz (assuming you don’t use the tauKhan fix), and the damage induced by (adapter’s controller polling +) 1000Hz adapter polling clearly not being multiples of 60Hz, but being high.
The concerns are the same but the way the damage to latency stability happens differs.
And explaining the maths for that isn’t for today, but I’ll spoil the result: with the adapter overclock only, the console still wins. So this is not “too good”. It’s just slightly worse, when previously, it was waaaay worse.

Why does it drop to 500Hz ? Why does it change slightly ?

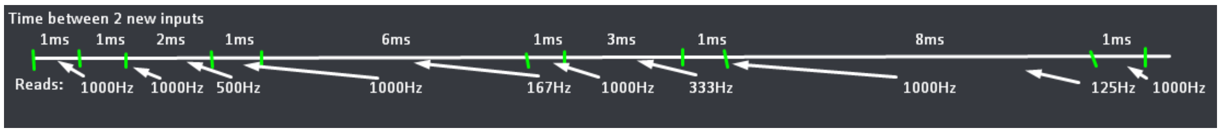
As of Slippi v2.3-beta4, the way the polling rate display in the Controllers tab works is as follows.

There’s a USB polling thread, whose entire operation is roughly the following: { Ask for data from the adapter - wait until data is received - register this new data - repeat }

When informed of a newly completed USB transfer, it writes down the time, and it writes down how long it’s been since last time. What you see in the controllers tab is the result of some external agent showing up every now and then (clearly the value there doesn’t update 1000 times/second), and asking what the value of “how long it’s been since last time” is. They read, say, 1.01 milliseconds, and compute 1/0.00101 = ~990. You see “990Hz”.

First thing to notice here is that even though USB transfers happen precisely on a 1000Hz basis, the thread tasked with handling the result of USB transfers doesn’t wake up precisely on a 1000Hz basis.
On a non-real time operating system, you wake up when you’re given the chance. So the variations spanning 800~1200Hz you witness are just a distorted perception of events truly happening at 1000Hz.

Then, even though you’ve told the PC (by “overclocking”) to ask the adapter for news every 1000Hz, sometimes the adapter just doesn’t have news. The official adapter with only one controller plugged in polls that controller every ~1.2 ms (~833Hz). That means that in 6ms, it has 5 updates. Which means that out of 6ms, there should be one where the adapter just *doesn’t have anything to tell you*. What’s supposed to happen there is that the adapter tells you just that (on the technical level, by sending a “NAK” message). This message doesn’t reach the Dolphin polling thread - it just means that the underlying driver will keep asking, and when it will obtain a real answer, it will inform Dolphin.
In other words, it is expected that one out of 5 times, two subsequent USB transfers will be spaced by 2ms instead of 1ms. If that external “polling rate controlling” agent happens to check the polling rate just after a 2ms spacing, it will report ~500Hz. This is completely expected.



Additionally, sometimes, your PC is just busy. The thread handling USB transfers may simply not be given execution time. In that case, longer USB ‘blackouts’ can be witnessed. 5ms without updates could make a 200Hz polling rate show up. That’s not the end of the world if you see it once - it happens.

Keep the dimensions in mind: 5ms without controller updates can be harmful if you *just* happened to attempt ledgedashing at this very frame, but even then, it’s only 5ms. If you’re experiencing major input problems such as inputs freezing for several frames, it can’t have to do with USB polling. 3 frames is 50ms.

Such micro-blackouts can also happen because of issues with the USB architecture of your setup. Ideally, you want the USB to be plugged as close as possible (architecturally) to the motherboard. If you’re using an external hub, an internal hub, an extension card with USB ports, or anything else of the sort, there’s a possibility that intermediary is limiting the USB communications. If you commonly see drops below 400Hz in the controller tabs, you might want to look into this.

Mayflash: 375Hz or 1000Hz ?

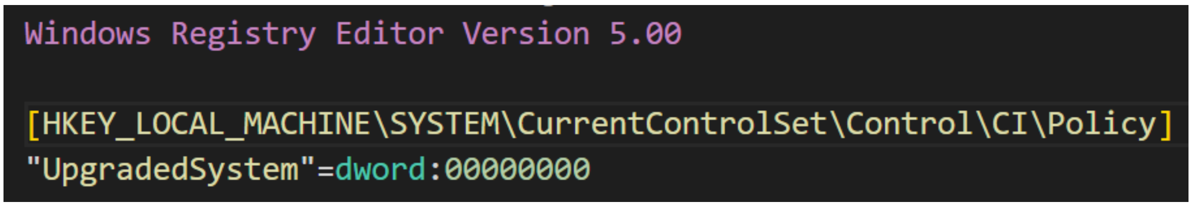
(follows previous entry) Then comes the question of the Mayflash: if it polls controllers at ~375Hz/~250Hz (v5/v6) as opposed to the official's ~833Hz/~1002Hz (without/with double ports trick) - how comes it even ever reports 1000Hz in the Slippi controllers tab ? Because it lies about having new information.

It almost always sends information - but the same as before when it doesn't have new information. So the polling rate seen in the Controllers tab isn't really indicative of the underlying events, it just means your PC is indeed attempting to communicate every millisecond. Somehow, when it has new information every ~2.67ms, the Mayflash adapter claims over USB to have new information every 1ms (sometimes 2), and when it has new information every ~4ms, it claims over USB to have new information every 2ms.

What does allow_overclock.reg do ?

allow_overclock.reg (thanks to Nikki for scripting it) edits the Windows Registry to add an "UpgradedSystem" entry in Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\CI\Policy.

This is the whole script:



```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\CI\Policy]
"UpgradedSystem"=dword:00000000
```

Context:

HIDUSBF was supposed to have been signed, which is why I never got why there were such issues with Secure Boot, nor why it didn't happen with everyone It turns out, there used to be several levels of driver signing, with the first (afaiu) level being "cross-signing", which is, signed by an entity entrusted by Microsoft with signing rights.

HIDUSBF had such a signature, which is why it claims to be signed. But as these external entities didn't protect their signing keys properly, eventually cross-signed malwares happened. Windows therefore decided cross-signature were no longer valid in W10.

However they didn't want to break backwards compatibility even when upgrading from W7 to W10, so they made it so W10s that are upgraded from W7s still accept cross-signed drivers. The method above tells your Windows sets a flag that says it was upgraded from a previous OS, which makes it accept cross-signed drivers.

Contact

Dms open for comments, suggestions or questions, I can be found in most Melee R&D / Gamecube emulation Discords as Arte#9281, or on Twitter.