
FLIRC remote

Autor:

Data de publicació: 10-06-2014

FLIRC - A review of the ultimate media centre companion

What is FLIRC?

Remote controls intended for computer use, on the whole, include a cheap IR remote and a USB receiver which interprets button presses from the remote into input events the operating system can use. These products are fairly cheap, and more or less get the job done, but they aren't perfect.

For one, you are left with yet another remote in your surely growing collection, made even worse by the fact that this one is probably of significantly lower quality than whatever came with your TV or audio receiver. The other, more practical, problem is how you utilize the remote itself. When you buy a remote and receiver combination, the functions of all the buttons on the remote are predetermined and generally can't be changed. If you wanted to do something special, like assign a particular button on the remote to an Alt+ key combination, you would need third party software that could translate whatever static event the receiver spits out for that button into a keyboard event.

Enter FLIRC, a product designed to change the way people use IR remotes to control their computers. FLIRC doesn't include a remote control, instead, it uses whatever remotes you already have. This immediately remedies the first problem, as you can simply program FLIRC against whatever remote you are already using to control the existing A/V equipment. But even better than that, FLIRC gives you incredible control over what happens on specific button presses. This is really the greatest strength of FLIRC, as it allows the user to create advanced control layouts that would not otherwise be possible.

For example, in my home theater the surround sound receiver's remote includes a full number pad, even though in my particular setup it is non functional. With FLIRC, I can program these otherwise useless buttons to command various functions on my HTPC, turning what was once a half useless remote into something I can use for two separate devices without them interfering with each other.

How FLIRC Works

FLIRC is fundamentally different than other products on the market, as the mapping of IR input to keypresses is not static, but user programmable. While it is possible to get this type of functionality with standard USB IR receivers, it requires intermediary software to be constantly running: adding another layer of complication and potential failure. By saving configuration on the FLIRC device itself, the user no longer has to constantly keep software running to interpret remote commands into their desired keypresses. What's more, since FLIRC stores its configuration internally rather than on the computer, you can move FLIRC between computers and devices without having to reconfigure it.

The key is that FLIRC reports itself to the operating system as a standard USB Human Interface Device (HID), so any modern OS will be able to read it as if it was a keyboard. Even if the device or OS you want to use FLIRC on has no included support for remote controls, you can program FLIRC to "press" keys that the system will understand. For example, I was able to program FLIRC to interpret the directional pad on my remote to standard keyboard arrow keys; I then plugged FLIRC into my Android device and was able to navigate through the applications with my remote.

This allows for incredible flexibility. Being able to program FLIRC from a different device than what you end up using it on and the fact that it emulates a USB keyboard means you can use FLIRC where other products would be useless.

FLIRC Hardware

The FLIRC hardware is a small and unassuming little clear box that plugs into your USB port, about half the length of your average USB flash drive. The clear casing is pretty cool as you can see the internal workings of the device, though I was somewhat surprised there were no LEDs to indicate IR activity like in the other IR receivers I've used. It

seems logical enough to have a little SMD LED on the PCB if the case is going to be clear, but on the other hand, a random blinking LED might look a little odd in your home theater setup. Still, it would be handy to have a visual verification that the FLIRC dongle is receiving the IR signals from your remote, especially since the nature of FLIRC means you will be throwing random remotes at it and hoping FLIRC is able to understand them.

Of course, that being said, I had no problem programming FLIRC against any IR remote I threw at it. The FLIRC documentation suggests some remotes may not be compatible, but I certainly couldn't find any that didn't work exactly as expected.

Build quality feels very good on the FLIRC; the plastic case is made thick enough that you're not going to break it under normal use, and there is no wiggle at all where the PCB lays into it. I also liked the new packaging, which is simple to open and doesn't waste a lot of materials. It's interesting to note that the packaging isn't strictly necessary at this point in FLIRC's development (as there are no retail stores that sell it yet), but is included simply to make the product look more professional.

Between the construction and presentation, it's clear that the FLIRC team is committed to delivering a polished and professional device, which is definitely something to be commended for such a small group. It would be easier and cheaper to have made the FLIRC out of cheaper materials and ship it to you in a plastic bag, but they've clearly taken the high road throughout the production of the physical device.

FLIRC Software

The hardware is just one part of the FLIRC equation, the arguably more important component is the software used to program it. FLIRC's configuration software is written in Qt, and is available on Linux, Windows, and Mac OS. The Windows and Mac OS versions of the software are distributed as stand-alone installers, while for Linux an Ubuntu repository is maintained for the 32 and 64 bit versions. The repository is a nice touch for distributions which use .DEB packages, but it would be nice to see the stand-alone package in other formats such as RPM or just a plain TAR.GZ. As it stands, user's are left to their own for converting the DEB to a format usable on their system, which is obviously less than ideal.

Configuring FLIRC through the software is very simple. You have the option of using an automated setup process which will flash different buttons on the on-screen display of the remote control and ask you to press buttons on the physical remote to correspond with them. You can also choose to go through this process manually if, for example, you didn't intend on programming all the buttons or you are using an unconventional button layout.

In addition to the standard remote control layout, the FLIRC software also allows you to program your physical remote to various other software profiles. As of this writing, the software includes the following profiles:

- Standard remote
- Full ASCII keyboard
- Boxee
- XBMC
- Windows Media Center

I wouldn't be surprised to see this list grow as time goes on and the community finds more uses for FLIRC. In the meantime, using the full ASCII keyboard should let you program FLIRC for any software or device that doesn't have a specific profile yet.

In addition to the GUI application, there is also a command line version included which lets you program, delete, and backup FLIRC configuration. This could prove very useful for scripting or other system integration.

Developer API

One of the most interesting aspects of FLIRC is that a developer API is planned to better integrate FLIRC into other software or products. As an example, the FLIRC site mentions being able to program the device from within the user interface of applications like Boxee. Unfortunately, this feature hasn't yet been publicly released, and details on how it will work and what it will be licensed under are pretty scarce.

Developers are currently encouraged to contact the FLIRC team directly if they would like a copy of the in-development API and documentation, but until the API is cleaned up, documented, and released publicly released, adoption is going to be seriously hindered.

Open Source No Show?

If there is one serious complaint I have about the FLIRC software (and indeed, the product), it's that no element of the product has been open sourced yet, despite the developer's stated intent. Even the main page of the site says the FLIRC configuration software is "entirely open source", but good luck finding it.

When we contacted FLIRC directly about the issue, we were told that the plan is still to open source the configuration GUI and developer API, but the public release of both have been delayed until the code can be cleaned up and improved. This is a story we've heard many times from many different developers; but in the case of FLIRC, I'm inclined to believe the desire to open source the product is genuine.

FLIRC is still in the early stages of development and the team is very small; while I would have loved to have seen the source get opened on day one, it's difficult to find fault with such a small group taking on the challenges of putting the product into production on their own and finding themselves a bit pressed for time. This is an issue we will certainly be keeping a close eye on as the product matures.

FLIRC's Mission

I would be remiss to end this review without mentioning one of the most important aspects of FLIRC, it's overall mission. When I first heard of FLIRC and checked out the product, the only thing I knew was that it was essentially the project of one man, Jason Kotzin, who was frustrated with the state of IR remote support on Linux and wanted to do something about it:

"I am an avid user of linux, but my biggest gripe was trying to use my Panasonic TV remote control with my ubuntu media center. There is a project called LIRC which converts your IR signal to electrical and has some decent support and a big following. However, my remote wasn't on the supported remote control list, I reluctantly gave up unwilling to make the effort to add my remote. Out of my frustration, F(LIRC) was born."- Jason Kotzin

This seemed like a good enough reason to put a product like this into production (if, perhaps, a serious overreaction), and certainly a good enough reason for us here at The Powerbase to take interest. But as I put in more time researching the product and going through the FLIRC site, I learned more of the incredible story behind its creation and overall mission.

Right after his graduation, Jason Kotzin was diagnosed with stage 3 cancer. He was helped through his surgery and followup chemotherapy treatments by his girlfriend (now wife) Maggie, herself a cancer survivor. That was 5 years ago, and Jason now has a clean bill of health and officially considers himself a cancer survivor. Looking for a way to give back to his oncologist Dr. Heinz Lenz, Jason and Maggie decided to use FLIRC as a way to raise funds for the University of Southern California's Cancer Center; a portion of every sale goes directly to the Cancer Center and the man who Jason feels is responsible for saving his life.

It would have been very easy for Jason to plaster this fact all over the website, remind you on every page about it until you were guilted into purchasing a FLIRC, in the process he could have jacked up the price and likely ended up making more money. But that isn't his style. Aside from the little blue ribbon on the back of the FLIRC package and the footer of the site, you would never have known FLIRC was anything but another product vying for the increasingly popular HTPC market. Jason isn't trying to make himself look like a hero or flaunt his philanthropic goals, FLIRC is simply his personal way of giving back to those who helped him when he needed it the most.

Conclusion

It would be easy to forgive faults with FLIRC given the noble goals of its creator, Jason Kotzin. Poor build quality, shoddy software, all could be given a pass once you realize that your purchase price went to supporting cancer

research. You would have to be a pretty cold person to blast a product for minor issues if one of its goals is to raise money for a good cause.

But the fact of the matter is, with the FLIRC, you don't have to. The hardware and software are top notch, and show the dedication to quality and attention to detail that is a hallmark of a small team working on a labor of love. There is no question that Jason and everyone who is working with him on FLIRC are exceptionally proud of what they have accomplished, and they should be. FLIRC is a new approach to controlling your computer that manages to fix essentially every problem with existing implementations.

Not only is FLIRC a brilliant stand-alone concept, but we would all be better off if this technology was implemented into consumer hardware across the board. Why can't we program our TVs or surround sound receivers to work with remotes of our choice? The hardware is more than capable, it's just missing the software. Perhaps FLIRC is the first step in that idea being a reality.

Review Credits go to Tom Nardi over at thepowerbase.com. Thanks for the fantastic review. Original review available [here](#)