

PC Control of the Gewa Prog Programmable Infrared Transmitter

By Tom Butler

Hello, I'm a C4/5 quadriplegic and as such cannot move my legs and have very limited use of my arms. I am unable to control my environment without the help of speciality devices for the disabled. One such device is the Gewa Prog, which functions like a universal remote control. It can learn infrared codes from other remote controls (TV, stereo, air-conditioner etc), it can control X-10 devices (X-10 is a system that works through a home's electrical wiring for controlling appliances, lights and other things) and it can control infrared accessible telephones (like the TS Phone sold by Technical Solutions). It can be accessed by lots of different types of switches, meaning severely disabled people can use the Prog. It also can be accessed by computer, which has been the easiest way for me to use it, and what I will be covering in this document.

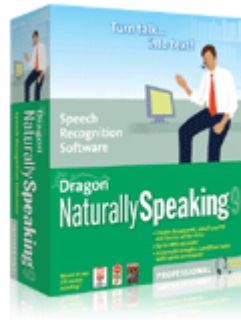


The Prog works by learning infrared codes from other infrared devices and storing them. Codes are programmed into each individual button on the device, with numerous "levels" of storage. For example button 1 on level 1 could turn a heater on, whereas button 1 on level 2 could turn a TV on. This allows for many codes to be stored, but makes them difficult to access as the locations either have to be memorised or written down somewhere. I found the constant clicking that is needed for switch access to be frustratingly slow. Setting up the Prog is quite simple, but it is a fairly boring job to initially program in lots of codes. Recording which code was stored on which button and which level is essential!

The Prog can be connected to a computer using the PC linkup kit accessory, which includes software that gives an on-screen representation of the device. This allows functions to be accessed by mouse, but has the same difficulty as previously mentioned of having to remember levels and buttons that codes are stored on.



Voice Control using Dragon NaturallySpeaking Pro




As an alternative to the previous methods, the Prog can be controlled by voice when connected to a computer that has Dragon NaturallySpeaking Pro installed. This setup can be more intuitive and user-friendly for those that spend large amounts of time at their computer.

Every function that is programmed into the Prog can have a simple voice command created for it in NaturallySpeaking. Macros and scripts that can perform more complicated functions can also be created. As is the case in initially programming codes into the Prog, creating voice commands for every function is a slow process, but it only needs to be done once and is worth it in the long run.

Things needed:

- Gewa Control Prog/Prog III with PC linkup kit
- send_ir.exe program downloadable from Gewa web site
- Dragon NaturallySpeaking Pro voice recognition software

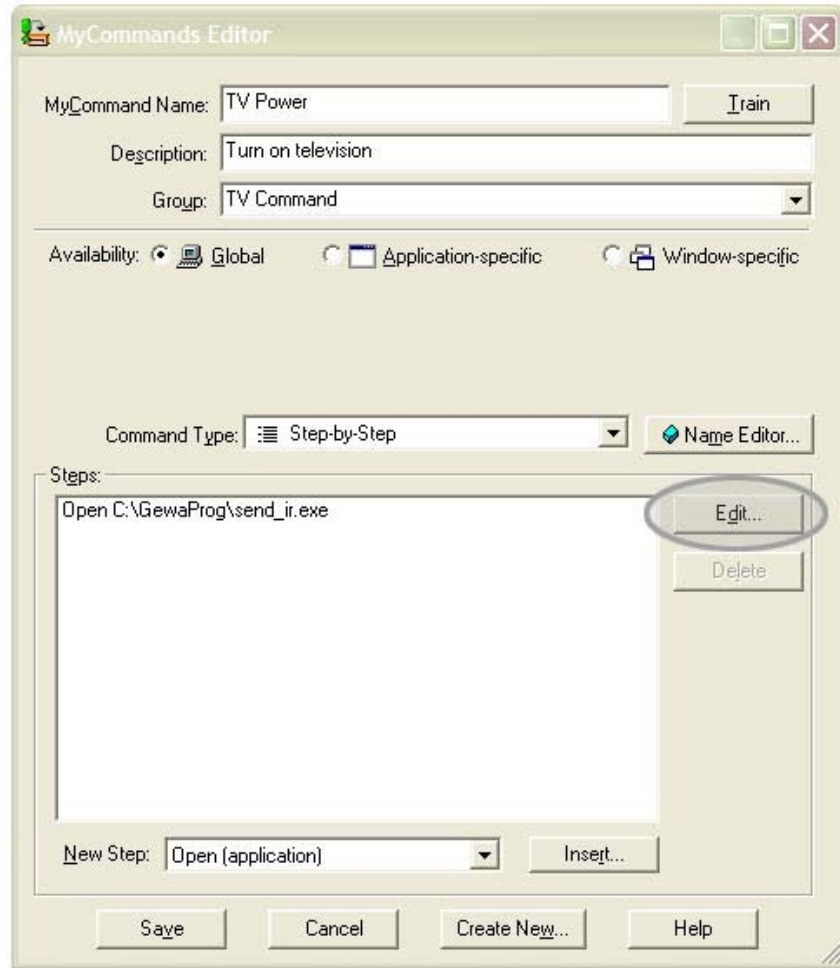
How to create a Prog voice command in NaturallySpeaking

 First download the send_ir executable from the Gewa website, located here: www.gewa.se/svensk/laddahem/program/send_ir.exe and save it to a suitable location, such as the default C:\GewaProg directory.

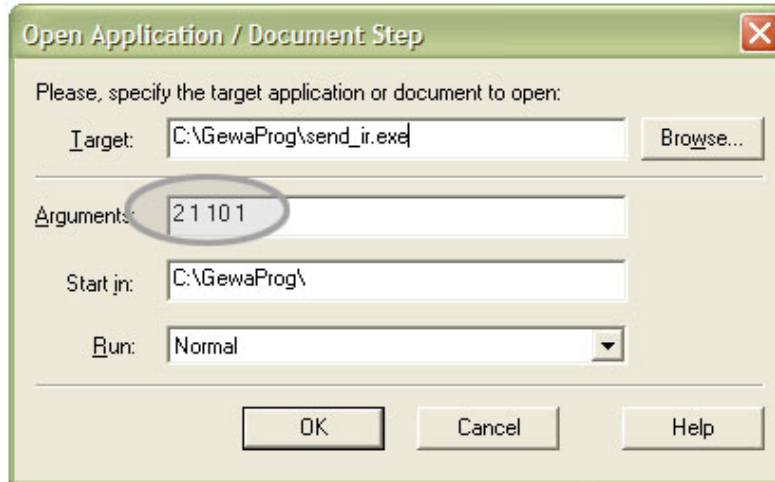
Send_ir is a small program that can be run with different parameters (or "arguments") to tell the Prog to emit different stored infrared codes. Each voice command calls send_ir with the parameters that correspond to a specific level and button on the Prog.

Next, open the Command Browser in NaturallySpeaking (in the Tools menu) and go to the "Script" panel, where custom commands can be created. Click the "New" button to open the MyCommands Editor and create a new command.

In the new command window, enter a name for the voice command such as "TV Power" or "Lights Off". This will be the phrase to run the command. You can also enter a description of the command as well as create groups for different types of Prog voice commands.



Leave "Availability" on the Global radio button. Change the "Command Type" to "Step by Step". These commands only need one step, which is to run the send_ir program once. To do this, select "Open (application)" from the "New Step" drop-down box at the bottom of the screen and click "insert". That step now needs to be told which program to run, which can be done by clicking "edit".



Browse to where the send_ir program was saved on the computer and select it to set it as the target. The arguments field determines which code that is stored in the Prog gets emitted. The format that the arguments must be in is four numbers separated by spaces, with the sequence being "level", "button", "duration" and "com port". In the example pictured above, the arguments would tell the Prog to emit the code that is stored at level 2, button 1, for a duration of 10 milliseconds if the device was connected to com port 1.

Click okay, then save the command. The voice command should now activate the correct code on the Prog. New commands can then be created to give voice control to any other desired Prog functions. For example you can program in TV commands so you can say "TV Power" followed by "TV ABC", then "TV volume up" (for this last command you would have to play with the duration argument until you get an adequate volume change). If an infrared controlled phone, like the TS phone, is present then that can also be accessed by commands like "dial 1", "dial 2", "dial 3", "start call". Home automation devices like X-10 modules can also be controlled with commands like "select heater", then "device on", or "select lights", then "device on". The X-10 module selection and on or off command can also be combined into the one voice command by using multiple "steps" in the MyCommands Editor.

The main drawback of this setup is that the Prog has to be connected to the computer constantly for the voice commands to work. That disadvantage can be reduced though if the person uses a wireless headset for their voice recognition software, then they can potentially have voice activated Prog control away from their computer as well. Another disadvantage is that devices to be controlled must be within line of sight of the Prog, however wireless infrared extenders can solve this problem though.

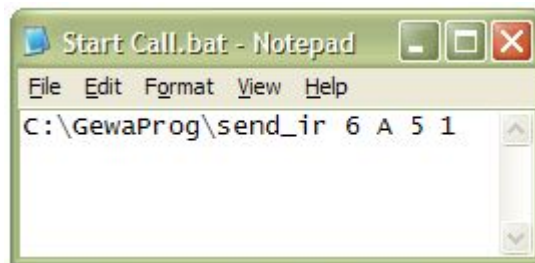
Other Ways to Control a Prog by PC

I use a mouth stick and trackball mouse, in conjunction with voice recognition software, to control my computer. Having voice control of infrared devices around me with

NaturallySpeaking and the Prog is easy and convenient, but sometimes I prefer to control things without having to activate the voice software (handy if there are other people in the room making noise, which makes voice recognition difficult). There are a few ways to do this.

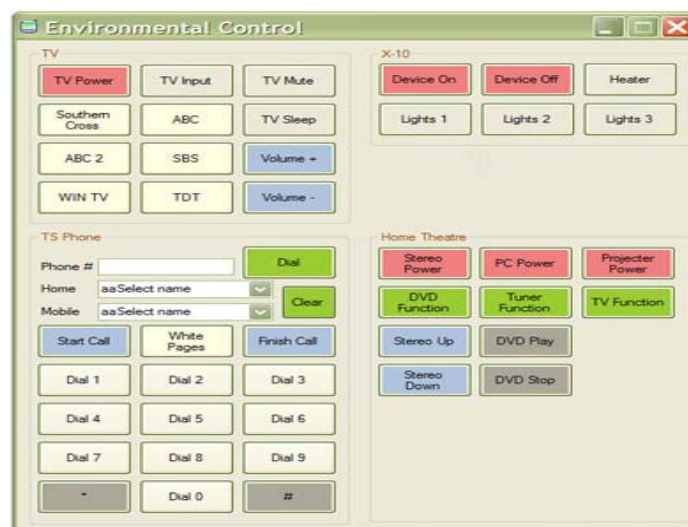
The first way is by using the standard Prog software interface, which I don't like for previously mentioned reasons (there are also some additional reasons which I will mention a bit further down).

The second way is to create "batch" files that are stored on the computer and can be run when needed. I find them useful for commonly used functions like finishing phone calls, powering on or off the TV, turning X-10 devices on or off, or muting the TV. I have a Kensington Expert Mouse Pro that has programmable buttons that can be programmed to run these batch files, giving very easy access.



People without programmable buttons could also achieve something similar by having shortcuts to batch files on the quick launch bar at the bottom of their computer screen. Custom icons can be made easily to help differentiate between the different batch files.

The third way that I control the Prog by computer, and the most involved, is by a custom Visual Basic interface that replaces the standard Prog software.



This was a bit of work, but solved another couple of problems that I had with the standard software:

1. When the software is open my voice commands don't work as I get an error message that the communication port to the prog is already open. I have to close the program for them to work.

2. I can't use the software by voice. It would be nice to not have to use the mouth stick sometimes, and instead just say the number of one of the buttons and for the voice software to click it. If that would work then I could use the on-screen software by voice from bed as well, when I don't have the mouth stick.

Creating a custom interface using Visual Basic overcomes these problems. It doesn't take up the communication port of the Prog, so I can use the interface OR the voice commands at the same time. There is no need to remember which button does what as everything is labelled. And also I only need to say the name of each button to control things because NaturallySpeaking clicks buttons that you say the name of, meaning I don't have to use the mouth stick if I don't want to.

The custom interface also has the flexibility of being able to integrate my phone contact database into it, so now I can just select a name from a list and then click "dial" and the number gets dialed in to the TS phone.

This setup definitely won't suit everyone, but for a high-level quadriplegic like me who spends most of my time at my computer it works great. It's a setup I've been using for years so it wouldn't surprise me if there were better ways of accomplishing a similar thing, so if you know of one please let me know!

Tom Butler

July 2007

tbutler@internode.net.au