Illuminate Bright Pi Automatically

The Raspberry Pi Camera Module is a fantastic piece of kit for any Raspberry enthusiast. It provides HD quality images easily and can be used in a wide variety of projects. Those of you who have followed the guides on the Maker Zone may have already seen my previous post about "Adding Push Notifications to MotionEyeOS using Pushover"- this is a great example of using the Camera Module as part of a home CCTV system.

The thing is, CCTV systems are only useful if you can actually see what your camera picks up, so if it is dark then you're going to struggle and might as well just not bother. Enter the Raspberry Pi NoIR Camera Module. This is essentially the same as the standard camera module, except it has had the infra-red sensor (IR) removed, meaning the camera will be able to detect IR light and this is useful for seeing in the dark.

In addition to this, the guys at Pi-Supply have created a nifty little add-on called Bright Pi which allows you to supplement your Camera Module with its own light source, including infra-red. The Bright Pi module itself is quite simple to put together although it does involve soldering 12 LED's in place, but if you follow Pi-Supply's assembly guide, you will have it up and running in no time.

You can also find some code examples that will get your Bright Pi working (hopefully!), but if you want something a little more automated, then read on!

Now, my scripts are as basic as you can get. They are simple bash scripts and use the code Pi-Supply created, but I have put them into an easier to use format that can be ran very simply.

The idea behind my scripts is that I wanted to be able to turn on the Bright Pi when the extra illumination is required - for example, when it is dark! Now I am already thinking of a more advanced way of doing this, such as with the inclusion of a photocell, but for starters I thought that it would be easier just using the scheduling tool, crontab.

First, you will need to download my scripts and you can do this by grabbing them from Git Hub as follows:

This should clone my code into a new folder called "brightpi-control". In order to use the scripts, we now need to make them executable:

Now that the scripts are executable, you can run them as follows:

If you have run them in that order, you should have noticed that the white LED's come on, the IR

LED's come on (although you may not see them) and then all the LED's will go off.

The next step is to create a crontab to schedule these scripts at your desired time. Crontab is very useful, but at first the commands (or expressions) look confusing. To make things easier, I stumbled across a website called crontab.guru that provides an interactive way of creating your schedule. It is definitely worth a look if you are new to using this.

So let's say we want the IR LED's to come on when it starts getting dark (which obviously will depend on the time of year, so a little trial and error will come in handy here) and then turns off when it starts getting light. For the purpose of this example, I have said that it starts getting dark at 4pm and starts getting light at 8am so to create a crontab, do this:

If this is the first time you have used crontab, you'll be asked which editor to use - stay with Nano by typing 2 and then enter.

You'll need to scroll down to the bottom of the text file that opens, and then add the following two lines:

Save by pressing Ctrl+O and then exit out of the editor by pressing Ctrl+X.

If everything has gone to plan, your IR LEd's on your Bright Pi should turn on at 4pm and off at 8am the following day. You may wish to test this out using the white LED's first, to ensure that the scripts are working, so all you would need to do is to change the "IR-on.sh" to "white-on.sh".

Anyway, I hope this simple approach helps you get started with your new Bright Pi! Please feel free to improve, comment and share your own suggestions to this