

# ELI Screen Blanking During Boot

## Summary:

This document describes the process of creating a simple script that will blank the screen of an ELI unit attached to a Raspberry Pi during the boot process. However, this process can be applied to any system to prevent the screen from displaying any output until the EDID file has been read properly.

## Target Device(s):

- Raspberry Pi 3 Model B ..... [Website](#)

**Contents**

<b>1. Blank using Resistor in Parallel .....</b>	<b>3</b>
a. ELI43-CP and ELI43-CR .....	3
b. ELI70-CR .....	4
c. ELI70-IPHW, ELI70-IRHW and ELI70-INHW .....	6
d. ELI101-CPW .....	7
e. ELI101-IPHW .....	8
f. ELI121-CPW .....	9
<b>2. Website and Support .....</b>	<b>11</b>

## 1. Blank using Resistor in Parallel

This method uses a resistor connected from the signal line to ground to pull down the signal until the device has booted. After the device has booted, the signal line is brought high and the ELI backlight will turn on. However, ELI has a pull up resistor already in place to pull the signal high when nothing is connected and allow the screen to operate with no backlight control. Therefore, we need to add a resistor that has enough pull to bring the signal below the threshold voltage. The pull up resistor's value is different depending on the ELI unit, see sections below for your ELI unit.

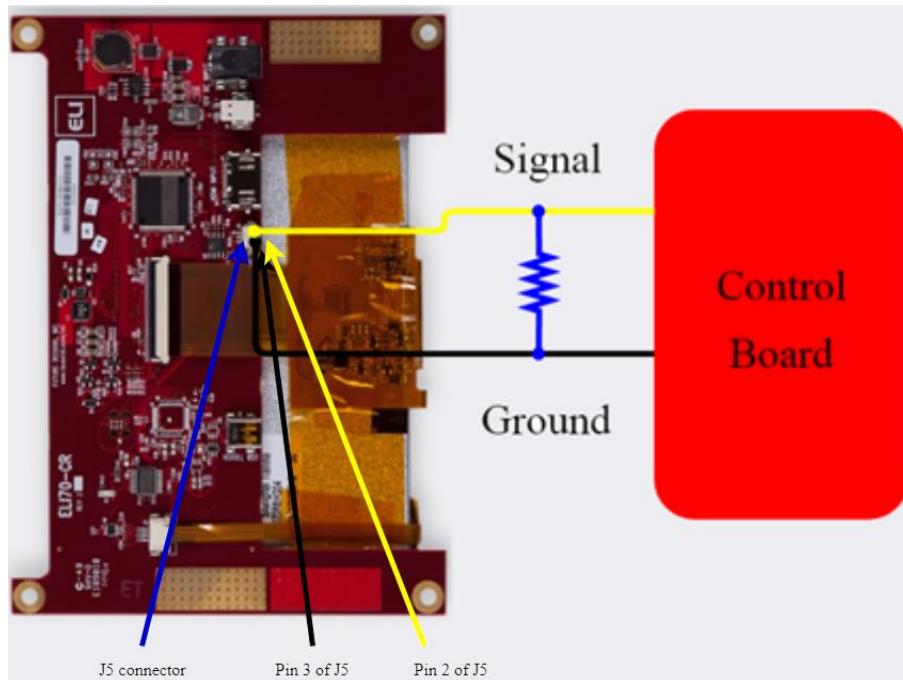


Figure 1. General Diagram

### a. ELI43-CP and ELI43-CR

The recommended resistance for the ELI43-CP and ELI43-CR is between 1,375 and 1,000 ohms to achieve a voltage below the threshold of 0.4V. Anything more than this will cause the screen to still be visible, though very dim, during boot. The wiring diagram for the blanking circuit using a Raspberry Pi can be seen below in Figure 2 and more information on controlling the ELI backlight can be found in the user's manual available on the FDI website at <https://www.teamfdi.com/product-details/eli43-cr#documentation>.

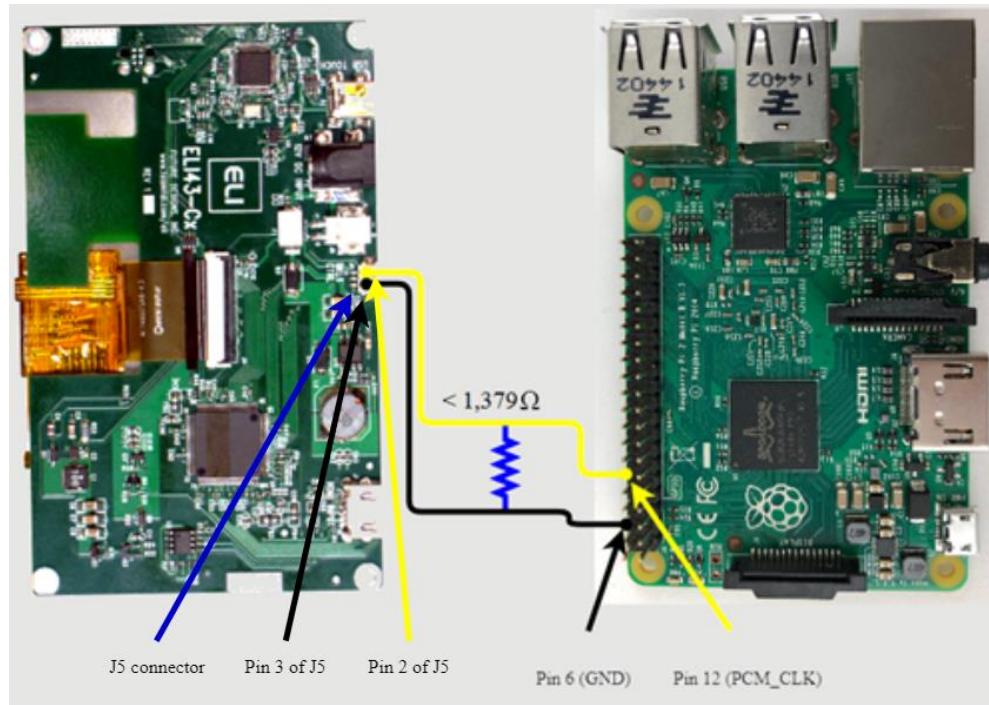


Figure 2. Raspberry Pi connected to ELI43-Cx

Using the Raspberry Pi, blanking the screen is achieved through a simple bash script that runs on boot. Create a file with the .sh extension, i.e. “screenblanking.sh”, and place it wherever you would like. Now edit the script and put the code seen below in Figure 3 in the script file. Save and exit. Next run the command “sudo chmod +x screenblanking.sh”, replacing “screenblanking.sh” with your file name, this will ensure the script is executable. Finally, we need to add an entry to Cron. Run the command “sudo crontab -e” to edit crontab, if it is the first time this command has run, it will ask for a text editor to use, select whichever option you prefer, then add the line “@reboot sudo /path/to/screenblanking.sh” at the end of the file. Again, replacing “screenblanking.sh” with your file name. Now whenever the Raspberry Pi boots up, it should blank the display until the desktop screen has loaded.

```
sudo echo 18 > /sys/class/gpio/export
sudo echo out > /sys/class/gpio/gpio18/direction
sleep 7
sudo echo 1 > /sys/class/gpio/gpio18/value
```

Figure 3. Bash Script for Blanking Display

### b. ELI70-CR

The recommended resistance for the ELI70-CR is between 93 and 75 ohms to achieve a voltage below the threshold of 0.4V. Anything more than this will cause the screen to still be visible, though very dim, during boot. The wiring diagram for the blanking circuit using a Raspberry Pi can be seen below in Figure 4 and more information on controlling the ELI backlight can be

found in the user's manual available on the FDI website at <https://www.teamfdi.com/product-details/eli70-cr#documentation>.

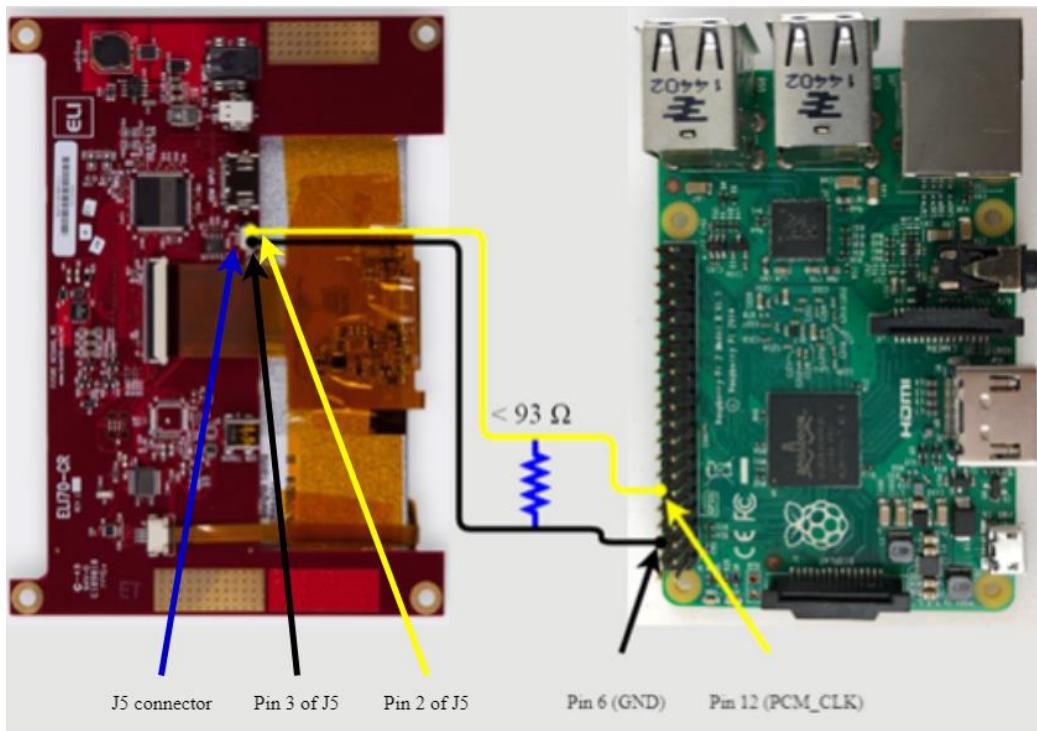


Figure 4. Raspberry Pi connected to ELI70-CR

Using the Raspberry Pi, blanking the screen is achieved through a simple bash script that runs on boot. Create a file with the .sh extension, i.e. "screenblanking.sh", and place it wherever you would like. Now edit the script and put the code seen below in Figure 5 in the script file. Save and exit. Next run the command "sudo chmod +x screenblanking.sh", replacing "screenblanking.sh" with your file name, this will ensure the script is executable. Finally, we need to add an entry to Cron. Run the command "sudo crontab -e" to edit crontab, if it is the first time this command has run, it will ask for a text editor to use, select whichever option you prefer, then add the line "@reboot sudo /path/to/screenblanking.sh" at the end of the file. Again, replacing "screenblanking.sh" with your file name. Now whenever the Raspberry Pi boots up, it should blank the display until the desktop screen has loaded.

```
sudo echo 18 > /sys/class/gpio/export
sudo echo out > /sys/class/gpio/gpio18/direction
sleep 7
sudo echo 1 > /sys/class/gpio/gpio18/value
```

Figure 5. Bash Script for Blanking Display

### c. ELI70-IPHW, ELI70-IRHW and ELI70-INHW

The recommended resistance for the ELI70-IxHW is between 545 and 500 ohms connected to 3.3V. Anything more than this will cause the screen to still be visible, though very dim, during boot. This is a pull up resistor rather than a pull-down resistor due to the ELI70-IxHW signal being backwards to the other ELI units. The wiring diagram for the blanking circuit using a Raspberry Pi can be seen below in Figure 6 and more information on controlling the ELI backlight can be found in the user's manual available on the FDI website at <https://www.teamfdi.com/product-details/eli70-irhw#documentation>.

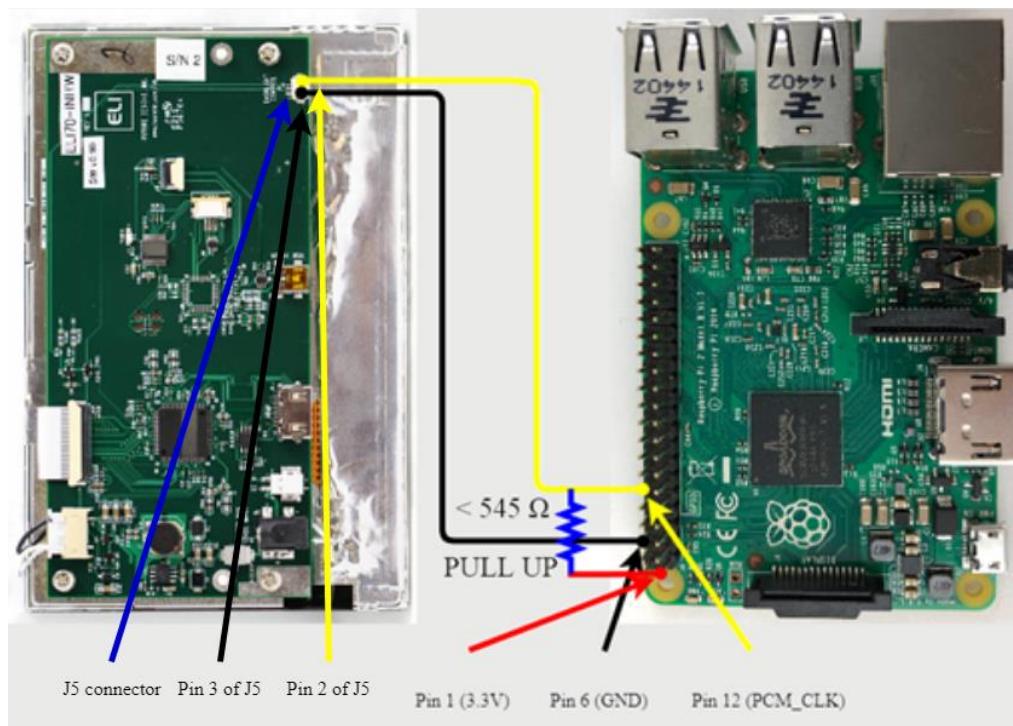


Figure 6. Raspberry Pi connected to ELI70-INHW

NOTE: Pull-up resistor not pull-down resistor.

Using the Raspberry Pi, blanking the screen is achieved through a simple bash script that runs on boot. Create a file with the .sh extension, i.e. "screenblanking.sh", and place it wherever you would like. Now edit the script and put the code seen below in Figure 7 in the script file. Save and exit. Next run the command "sudo chmod +x screenblanking.sh", replacing "screenblanking.sh" with your file name, this will ensure the script is executable. Finally, we need to add an entry to Cron. Run the command "sudo crontab -e" to edit crontab, if it is the first time this command has run, it will ask for a text editor to use, select whichever option you prefer, then add the line "@reboot sudo /path/to/screenblanking.sh" at the end of the file. Again, replacing "screenblanking.sh" with your file name. Now whenever the Raspberry Pi boots up, it should blank the display until the desktop screen has loaded.

```

sudo echo 18 > /sys/class/gpio/export
sleep 7
sudo echo out > /sys/class/gpio/gpio18/direction

```

Figure 7. Bash Script for Blanking Display

#### d. ELI101-CPW

The recommended resistance for the ELI101-CPW is between 93 and 75 ohms to achieve a voltage below the threshold of 0.4V. Anything more than this will cause the screen to still be visible, though very dim, during boot. The wiring diagram for the blanking circuit using a Raspberry Pi can be seen below in Figure 8 and more information on controlling the ELI backlight can be found in the user's manual available on the FDI website at <https://www.teamfdi.com/product-details/eli101-cpw#documentation>.

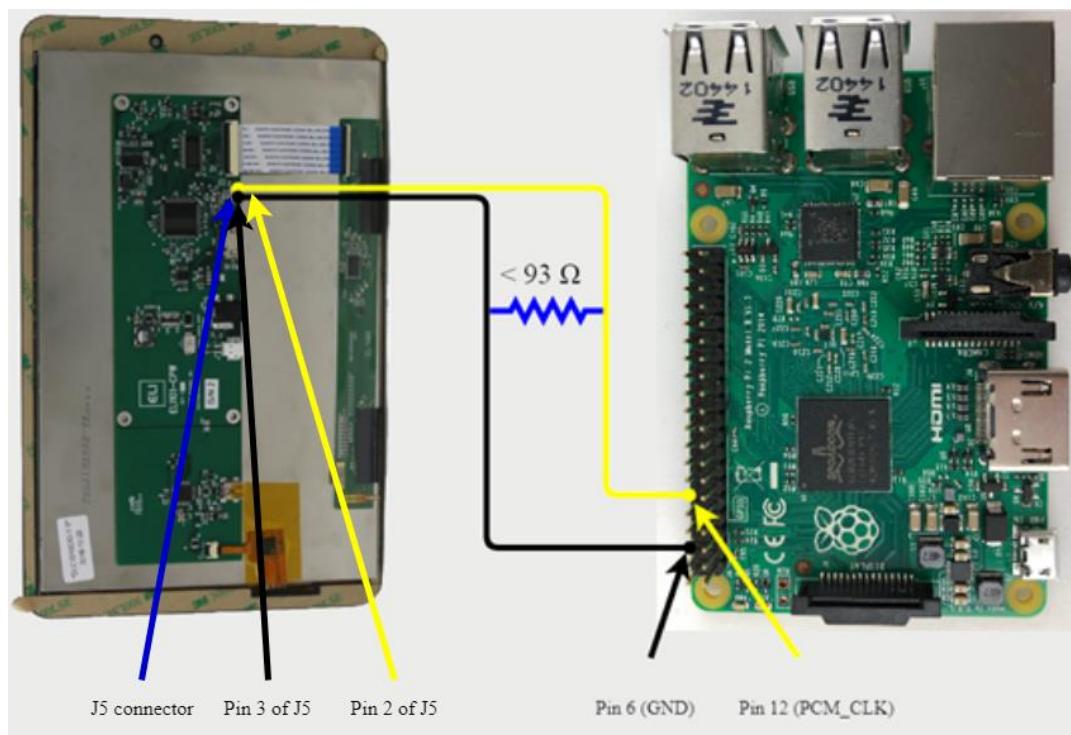


Figure 8. Raspberry Pi connected to ELI101-CPW

Using the Raspberry Pi, blanking the screen is achieved through a simple bash script that runs on boot. Create a file with the .sh extension, i.e. "screenblanking.sh", and place it wherever you would like. Now edit the script and put the code seen below in Figure 9 in the script file. Save and exit. Next run the command "sudo chmod +x screenblanking.sh", replacing "screenblanking.sh" with your file name, this will ensure the script is executable. Finally, we

need to add an entry to Cron. Run the command “sudo crontab -e” to edit crontab, if it is the first time this command has run, it will ask for a text editor to use, select whichever option you prefer, then add the line “@reboot sudo /path/to/screenblanking.sh” at the end of the file. Again, replacing “screenblanking.sh” with your file name. Now whenever the Raspberry Pi boots up, it should blank the display until the desktop screen has loaded.

```
sudo echo 18 > /sys/class/gpio/export
sudo echo out > /sys/class/gpio/gpio18/direction
sleep 7
sudo echo 1 > /sys/class/gpio/gpio18/value
```

Figure 9. Bash Script for Blanking Display

#### e. ELI101-IPHW

The recommended resistance for the ELI101-IPHW is between 500 and 333 ohms to achieve a voltage below the threshold of 0.67V. Anything more than this will cause the screen to still be visible, though very dim, during boot. The wiring diagram for the blanking circuit using a Raspberry Pi can be seen below in Figure 10 and more information on controlling the ELI backlight can be found in the user’s manual available on the FDI website at <https://www.teamfdi.com/product-details/eli101-iphw#documentation>.

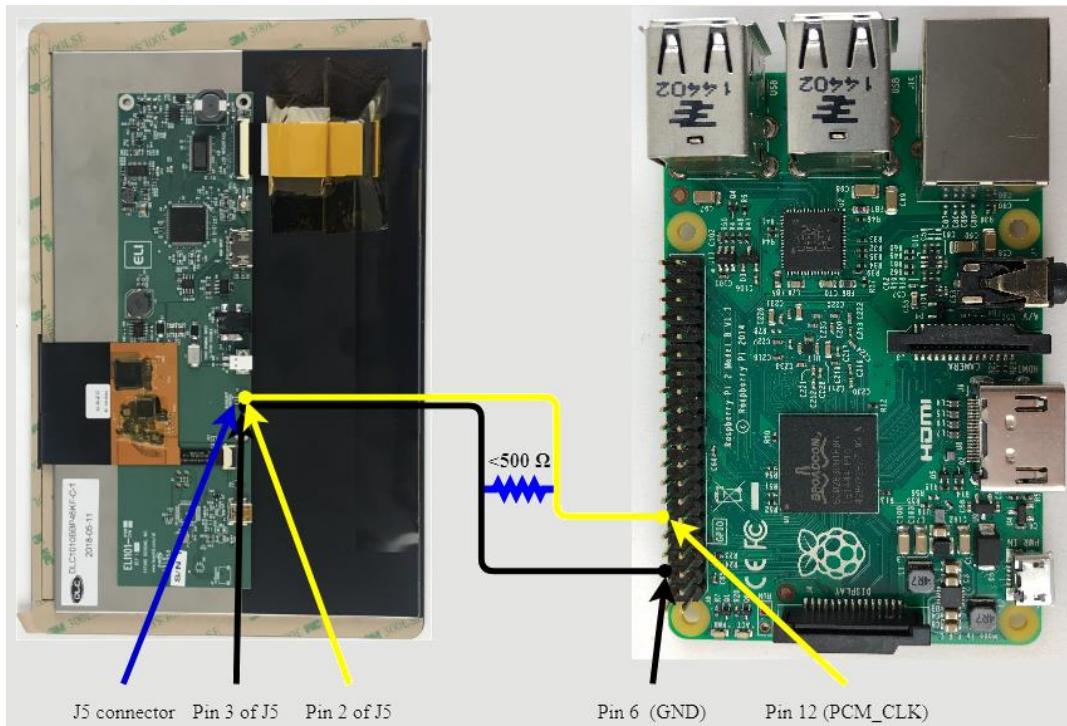


Figure 10. Raspberry Pi connected to ELI101-IPHW

Using the Raspberry Pi, blanking the screen is achieved through a simple bash script that runs on boot. Create a file with the .sh extension, i.e. “screenblanking.sh”, and place it wherever you would like. Now edit the script and put the code seen below in Figure 11 in the script file. Save and exit. Next run the command “sudo chmod +x screenblanking.sh”, replacing “screenblanking.sh” with your file name, this will ensure the script is executable. Finally, we

need to add an entry to Cron. Run the command “sudo crontab -e” to edit crontab, if it is the first time this command has run, it will ask for a text editor to use, select whichever option you prefer, then add the line “@reboot sudo /path/to/screenblanking.sh” at the end of the file. Again, replacing “screenblanking.sh” with your file name. Now whenever the Raspberry Pi boots up, it should blank the display until the desktop screen has loaded.

```
sudo echo 18 > /sys/class/gpio/export
sudo echo out > /sys/class/gpio/gpio18/direction
sleep 7
sudo echo 1 > /sys/class/gpio/gpio18/value
```

Figure 11. Bash Script for Blanking Display

#### f. ELI121-CPW

The recommended resistance for the ELI121-CPW is between 475 and 400 ohms to achieve a voltage below the threshold of 0.15V. Anything more than this will cause the screen to still be visible, though very dim, during boot. The wiring diagram for the blanking circuit using a Raspberry Pi can be seen below in Figure 12 and more information on controlling the ELI backlight and the J5 connector can be found in the user’s manual available on the FDI website at <https://www.teamfdi.com/product-details/eli121-crw#documentation>.

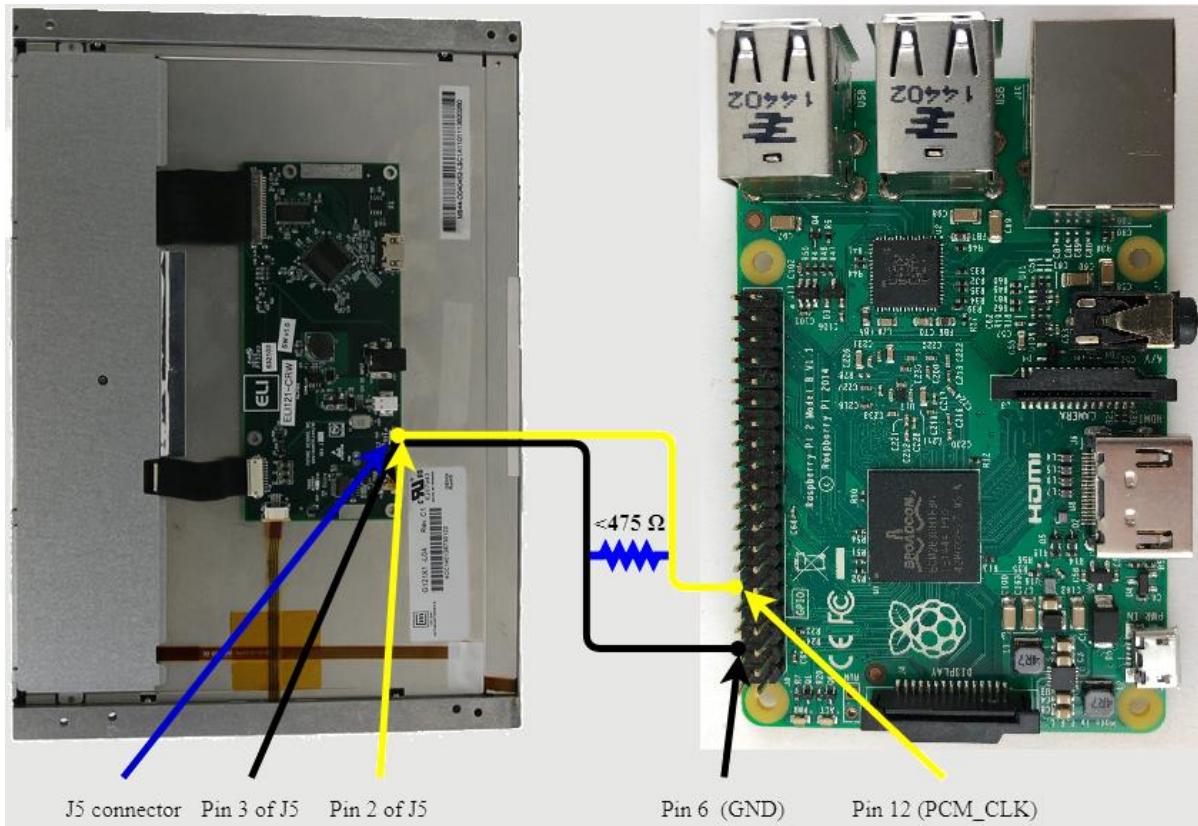


Figure 12. Raspberry Pi connected to ELI121-CPW

Using the Raspberry Pi, blanking the screen is achieved through a simple bash script that runs on boot. Create a file with the .sh extension, i.e. “screenblanking.sh”, and place it wherever you would like. Now edit the script and put the code seen below in Figure 13 in the script file. Save and exit. Next run the command “sudo chmod +x screenblanking.sh”, replacing “screenblanking.sh” with your file name, this will ensure the script is executable. Finally, we need to add an entry to Cron. Run the command “sudo crontab -e” to edit crontab, if it is the first time this command has run, it will ask for a text editor to use, select whichever option you prefer, then add the line “@reboot sudo /path/to/screenblanking.sh” at the end of the file. Again, replacing “screenblanking.sh” with your file name. Now whenever the Raspberry Pi boots up, it should blank the display until the desktop screen has loaded.

```

sudo echo 18 > /sys/class/gpio/export
sudo echo out > /sys/class/gpio/gpio18/direction
sleep 7
sudo echo 1 > /sys/class/gpio/gpio18/value

```

Figure 13. Bash Script for Blanking Display

## 2. ***Website and Support***

Support:

- FDI Support home page: ..... <http://www.teamfdi.com/support>
- FDI Forums: ..... <http://www.teamfdi.com/forum>