Method of Setting the boot self-starting program

There are many methods to set up the boot applet from the Raspberry Pi. Below we use the new **.desktop** file to implement the Raspberry Pi program boot self-starting program.

1. Making test scripts

The effect of the script file is to create a new **hello.c** file in the pi directory and add the "hello word!" string to the hello.c file.

1.1 Enter the command shown below at the command terminal to create a new testStart.sh script file

Nano testStart.sh

1.2 Enter the following in the testStart.sh file:

#!/bin/sh
touch /home/pi/hello.c
sudo chmod 777 /home/pi/hello.c
echo "hello word!">>/home/pi/hello.c

```
#!/bin/sh
touch /home/pi/hello.c
sudo chmod 777 /home/pi/hello.c
echo "hello word!">>/home/pi/hello.c
```

1.3 After the enter is completed, press **ctrl+x**, "**y**", "**enter**" to save and exit the file

1.4 Enter the command shown below at the command terminal to Add execute permission to the script

sudo chmod 777 testStart.sh

```
pi@raspberrypi:~ $ sudo chmod 777 testStart.sh
pi@raspberrypi:~ $
```

1.5 Test script function

./testStart.sh

We can see that the **hello.c** file has been generated.

```
pi@raspberrypi:~ $ ls
Desktop    Downloads  MagPi  Pictures  Templates     Videos
Documents  LCD-show   Music  Public    testStart.sh  wiringPi
pi@raspberrypi:~ $ ./testStart.sh
pi@raspberrypi:~ $ ls
Desktop    Downloads  LCD-show  Music     Public     testStart.sh  wiringPi
Documents  hello.c    MagPi     Pictures  Templates  Videos
pi@raspberrypi:~ $
```

1.6 Look at the hello.c file to see that there is a hello word! String.

cat hello.c

```
pi@raspberrypi:~ $ cat hello.c
hello word!
```

This test tutorial has been completed.

In order to verify the effect, we can first delete the generated hello.c.

Enter the command shown below:

rm hello.c

```
pi@raspberrypi:~ $ rm hello.c
pi@raspberrypi:~ $ ls
Desktop    Downloads  MagPi   Pictures  Templates     Videos
Documents  LCD-show   Music   Public    testStart.sh  wiringPi
```

If you don't delete it, every time you run the script, it will add a sentence "hello word!" on the next line.

As shown below.

```
pi@raspberrypi:~ $ cat hello.c
hello word!
hello word!
```

2. Create a new .desktop file

Input the command shown below to enter ./config file:

cd /home/pi/./config

Create a new autostart folder, ignore this step if you already have it.

mkdir autostart

Enter autostart file:

cd autostart

Create self-starting shortcut:

nano start.desktop

Then enter the following:

[Desktop Entry]
Type=Application
Exec=/home/pi/testStart.sh

```
  GNU nano 3.2      start.desktop

[Desktop Entry]
Type=Application
Exec=/home/pi/testStart.sh
```

After the modification is completed, press **ctrl+x**, "**y**", "**enter**" to save and exit the file.

Exec = the path to the startup script.

We can restart the Raspberry Pi and see the actual effect:
Input the command shown below:
sudo reboot

！Note: This method uses the Raspberry Pi to enter the desktop and then automatically start the program to achieve automatic startup, so you need to wait for the desktop to load before you can start, waiting for a relatively long time.

If the Raspberry Pi is not connected to the monitor, there may be a problem that the boot will not start automatically after adding the boot file.
In this case, you need to modify the /boot/config.txt file.

Input the command shown below:
sudo nano /boot/config.txt

```
pi@raspberrypi:~ $ sudo nano /boot/config.txt
```

Find the **hdmi_force_hotplug=1** line and delete the "#", as shown below:

```
# uncomment to force a console size. By default it will be displ
# overscan.
#framebuffer_width=1280
#framebuffer_height=720

# uncomment if hdmi display is not detected and composite is bei
hdmi_force_hotplug=1

# uncomment to force a specific HDMI mode (this will force VGA)
#hdmi_group=1
#hdmi_mode=1
```

Finally, we need to press **Ctrl+O** to save, and press **Ctrl+X** to quit.