

START MAIN.PY AT BOOT

To create a Python program that runs at the startup of the Raspberry Pi, you need to create a new bash file in the same directory as the main program. This results in a structure like the following:

```
<yourMainProgramPath>
├── main.py
└── launcher.sh
```

Write the following lines in your '*launcher.sh*' file:

```
#!/bin/sh
cd / # Go to the base folder
cd home/<yourMainProgramPath> # Navigate to the folder where your main program
is located
python3 main.py # Command line to execute the main file
cd /
```

Make sure not to forget to write the path in the third line starting from the base folder, so it looks like this:

```
cd /home/<userName>/<lisaFolder>/
```

After finishing this, you need to make the '*launcher.sh*' file executable. Open the terminal (Ctrl+Alt+T), navigate to the folder where the file is located (cd <lisaFolder>), and ensure that the file is there by typing 'ls', which provides a list of the items in the current folder. Check that '*launcher.sh*' is in that folder; if not, you may have made a mistake.

To make the '*launcher.sh*' file executable, type:

```
chmod 755 launcher.sh / chmod u+x launcher.sh
```

Now your bash file is executable. To verify that everything is working, type:

```
./launcher.sh
```

Your '*main.py*' should run. To set this script to start at the Raspberry Pi's startup, go back to the home folder by typing 'cd'. Create a folder for the logs by typing:

```
mkdir logs
```

Now you need to modify the crontab. Type:

```
sudo crontab -e
```

Enter your password if requested, and the crontab file should appear. Go to the last line and add:

```
@reboot sh /home/<userName>/<lisaFolder>/launcher.sh >/home/<userName>/logs/cronlog 2>&1
```

To exit crontab, press 'Ctrl+X', 'y', then 'Enter'. Your '*main.py*' should now start at the Raspberry Pi's startup.