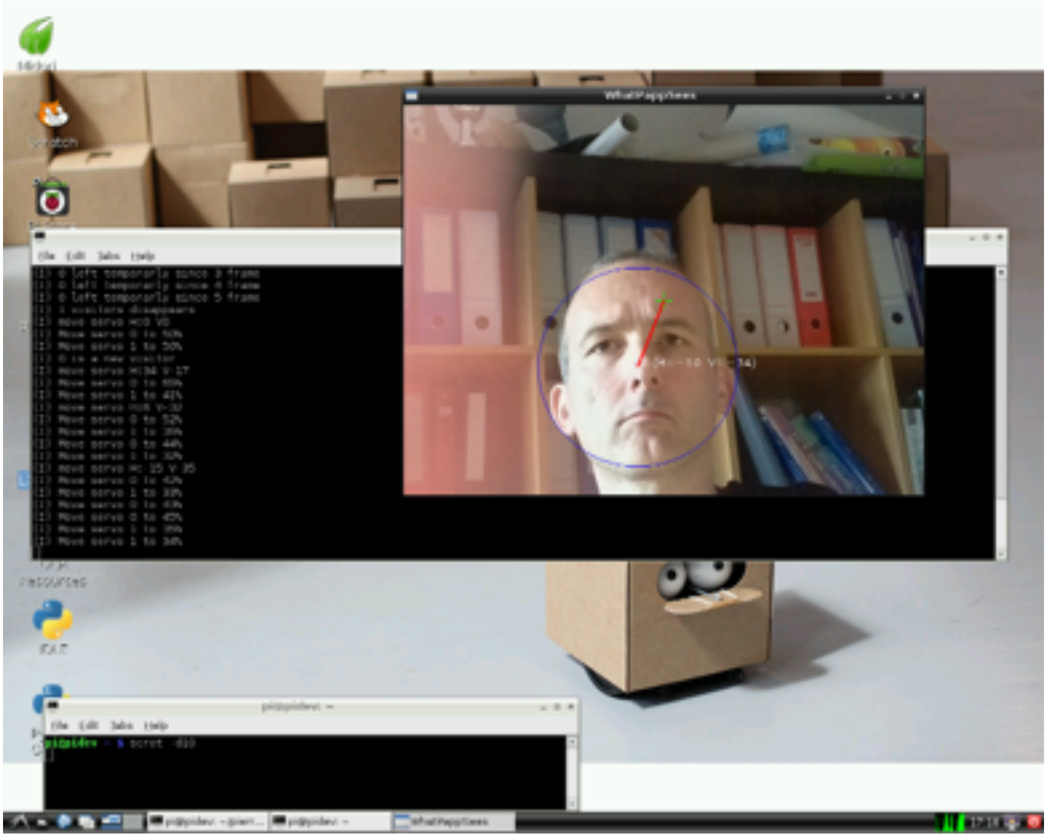
Rapsberry Pi Software for Tordal PAPP Portrait
How To Manual

Pierre RAUFAST - 2014
Version 3.0 - April 16th



```
[I] 0 left temporarly since 3 frame
[I] 0 left temporarly since 4 frame
[I] 0 left temporarly since 5 frame
[I] 1 visitors disappears
[I] move servo H:0 V0
[I] Move servo 0 to 50%
[I] Move servo 1 to 50%
[I] 0 is a new visitor
[I] move servo H:36 V-17
[I] Move servo 0 to 69%
[I] Move servo 1 to 41%
[I] move servo H:6 V-32
[I] Move servo 0 to 52%
[I] Move servo 1 to 35%
[I] Move servo 0 to 44%
[I] Move servo 1 to 32%
[I] move servo H:-15 V 35
[I] Move servo 0 to 42%
[I] Move servo 1 to 33%
[I] Move servo 0 to 43%
[I] Move servo 0 to 45%
[I] Move servo 1 to 35%
[I] Move servo 1 to 34%
```

```
pi@pidev ~ $ scrot -d10
```

How to use the software manually

Software is installed under `/home/pi/pierre/papp`
To go here : open an LXTerminal window (on background)
enlarge size of window (length) (for a better confort)
`cd pierre/papp`

To start software. It doesn’t work, since you need to add parameters. But it shows you the syntax.
`./papp`

```
Usage : papp trace(0/1/2)  ToleranceFramesOff (5=2s) distMinToMoveEye(20)
nbFramesMaxLooking@sameguy (5) MaxServo_X (95) MinServo_X (5) MaxServo_Y
(95) MinServo_Y (5) reverseServo_H (0) reverseServo_V (0)
```

- **Trace :** level of trace. 0 = no trace, no window. 1 = debug trace + window. 2 = only window

- **ToleranceFramesOff** = number of frames without seeing a visitor already seen, before considering the visitor left. Due to the webcam framerate, value of 5 = 2 seconds.

- **distMinToMoveEye** = max distance of a guy's move (in pixels) to be ignored by servo (avoid micro move) : to be tuned, value of 20 (pixels) seems fine

- **nbFramesMaxLooking@sameguy** = number of frames looking at the same visitor before changing. To be tested, but 10 is around 4 seconds. Looks fine.

- **MaxServo_H** = Max Amplitude for Servo Horizontal (in percent) : to use to set amplitude of your horizontal servo. If you want to limit horizontal move of your eye : for safety reason, I recommand to limit to 95 (= 95% of maximum).
- **MinServo_H** = Min Amplitude for Servo Horizontal (in percent) (no less than 5%)
- **MaxServo_V** = Max Amplitude for Servo Vertical (in percent) (no more than 95%)
- **MinServo_V** = Min Amplitude for Servo Vertical (in percent) (no less than 5%)

- **ReverseServo_H** : 0=No , 1=Yes : if you want to put your servo in reverse mode —> 100% far left instead of far right.
- **ReverseServo_V** : 0=No , 1=Yes. To inverse up/bottom.
- **ReverseServo_B** : 0=No , 1=Yes. To inverse up/bottom. **2=NO BLINK SERVO** (Eye Blink and Nap are disactivated)
- **MinServo_B** = Min Amplitude for Servo Blinking Eye (in percent) (no less than 5%)
- **MaxServo_B** = Max Amplitude for Servo Blinking Eye (in percent) (no more than 95%)
- **BlinkPeriod** : period in seconds between two blink eye. (30 seconds seems fine). A +/- 3 seconds is added randomly
- **HalfBlinkDuration** = time to close eye (milliseconds: 500 —> it takes 2x500 = 1 second to close and open eye again = 1 blink)
- **napPeriod** = time between 2 naps (seconds) : 600 = 10 minutes
- **napDuration** = duration of a nap (seconds) : 30 seconds seems ok. : eyes are closed during nap. Eyes are closing 3 times slower than **HalfBlinkDuration**
- **servoTrace** = display trace for all servos movement (0=no/1=yes)
- **BehaviorTrace** = when nobody is here, frequency of special eyes behavior (btw 0=(never) and 1000 (all the times, each cycle), value of 10 = around 1 per minute)

exemple :

```
./papp 1 5 20 5 65 35 62 38 0 0 0 60 40 30 500 600 30 0 10
```

Play with parameters to understand them. Read debug info to learn more about how it works.

To stop : `Ctrl-C` inside the window

<u>To recompile the software</u>

if you want to change code (good luck !!)

1. Modify `pappRobot.cpp` source code. This is a cpp code.
2. `cd /home/pi/pierre/papp`
3. Compile : just type : `make`

if no error ([120%] Build target papp) , `papp` software is generated.

<u>eyelid random blinking event</u>

Sometimes, the Pi take a nap. : close eyelid
Play with parameters **napPeriod** and **napDuration** to adjust period and duration

Sometimes Pi blinks eyes. (move of eyelid)
play with parameter **BlinkPeriod** and **HalfBlinkduration** to adjust frequency and blink duration.

eyelid random special event

When nobody is found, the software could have randomly a special eye move.

- looking all around a circle
- looking from left to right
- looking from up to down

Set parameter **behaviorFreq** to adjust the frequency. 10 = 10/1000 of cycles = around 1 time per minute randomly.

Automatically of Manually Start the software

To start Automatically

If the software is not launched automatically after a boot (the redlight of the webcam is switch off and servos are not moving)

Open a LXTerminal window

```
cd pierre/papp
./pappAfterBoot start
sudo reboot
```

After this reboot, the software will be start automatically (no window, no trace). No need to have a screen or keyboard.

Warning : when started automatically, parameters are found on `/etc/init.d/startPapp` file.
To modify then (before doing the previous action)
- open a LXTerminal window.
- edit the file using nano editor :

```
sudo nano /etc/init.d/startPapp
```

- modify the parameters

WARNING : be sure that the line is finished by a &

- Save the file : `Ctrl-O` and `Ctrl-X`

To start Manually

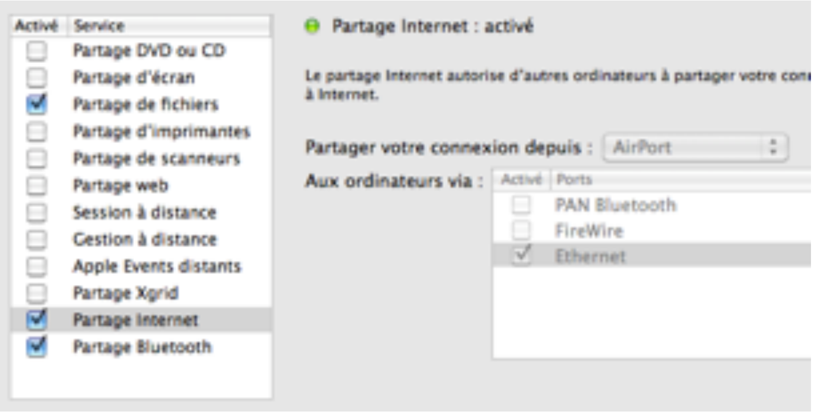
If the software is started automatically after a boot and you want to stop it.
Open a LXTerminal window

```
cd pierre/papp
./pappAfterBoot stop
sudo reboot
```

After this reboot, the software will no start. To start it, follow instructions « How to use the software manually »

Network

The Pi doesn't need the network to work.
If you need to connect it, do it thru your mac. Directly connect them with a ethernet cable. And configure your mac like this :



Activé	Service
□	Partage DVD ou CD
□	Partage d'écran
☑	Partage de fichiers
□	Partage d'imprimantes
□	Partage de scanneurs
□	Partage web
□	Session à distance
□	Gestion à distance
□	Apple Events distants
□	Partage Xgrid
☑	Partage Internet
☑	Partage Bluetooth

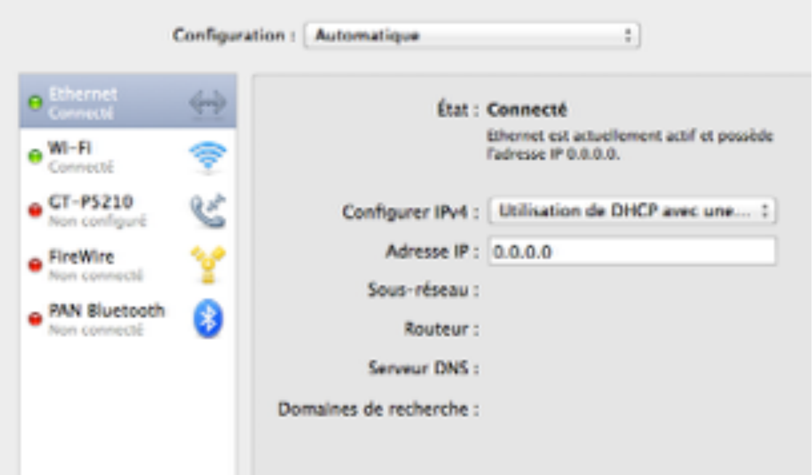
● Partage Internet : activé

Le partage Internet autorise d'autres ordinateurs à partager votre con
à Internet.

Partager votre connexion depuis : AirPort

Aux ordinateurs via :

Activé	Ports
□	PAN Bluetooth
□	FireWire
☑	Ethernet



Configuration : Automatique

- ● Ethernet Connecté
- ● Wi-Fi Connecté
- ● GT-P5210 Non configuré
- ● FireWire Non connecté
- ● PAN Bluetooth Non connecté

État : Connecté

Ethernet est actuellement actif et possède l'adresse IP 0.0.0.0.

Configurer IPv4 : Utilisation de DHCP avec une...

Adresse IP : 0.0.0.0

Sous-réseau :

Routeur :

Serveur DNS :

Domaines de recherche :

if you need to connect to the Pi thru network

```
ssh 192.168.2.2 -l pi
```

password : raspberry
(check IP adress first)

Start and Stop the Pi

Since there is no hardrive (physical flush issue), you can stop/start your Raspberry without big risk.
There is always a risk to crash your sd card, but it's quite low.
To prevent it, always have a backup of your SD Card.

Backup and restore your SD Card on a Mac

To do a backup (image) on Mac :
Open a Terminal window on your Mac
To list all devices, type :

```
df -h
```

(should be disk1 or rdisk1)
WARNING : don't backup your harddrive !!! Check the Size column to verify that the filesystem is 8 Gb large and no more !!!

to do a backup :

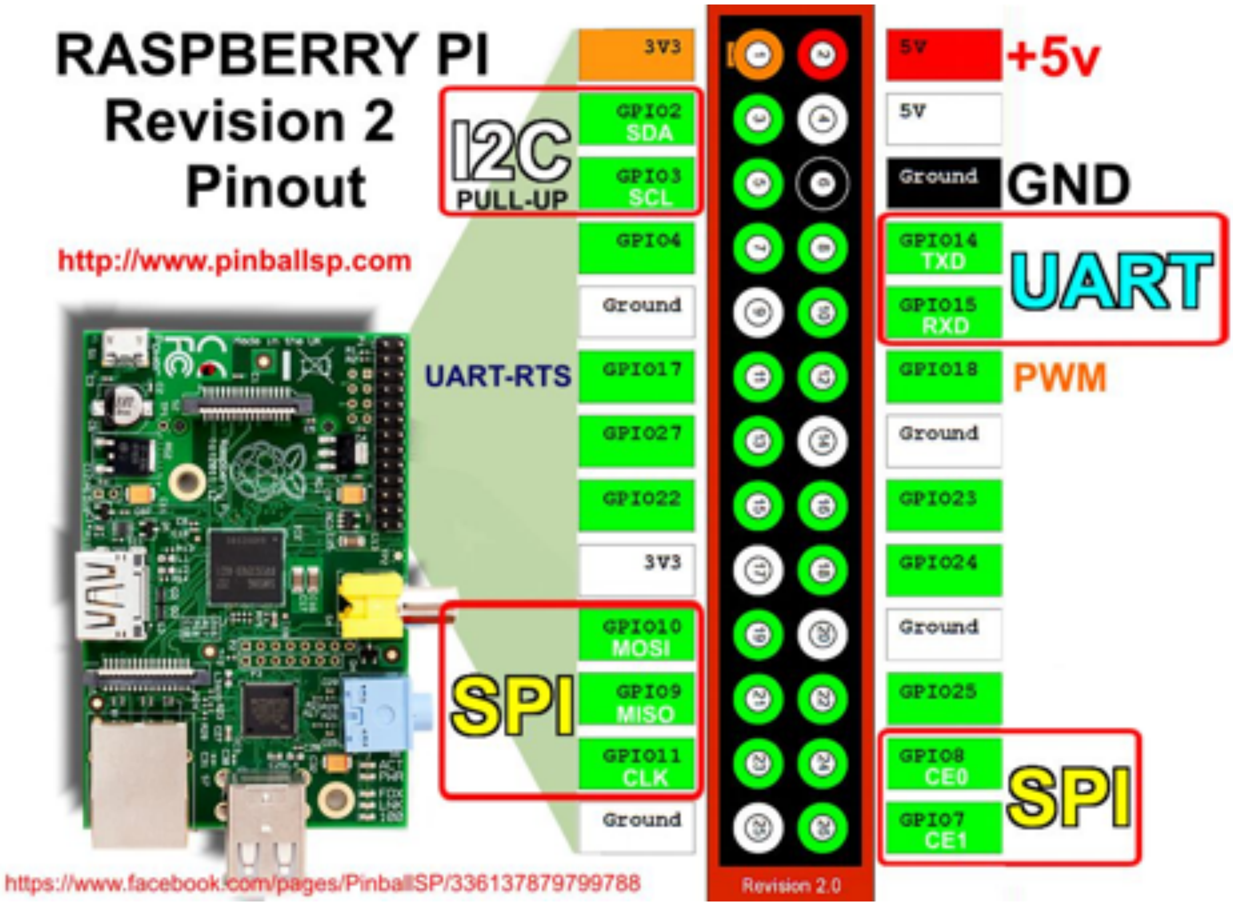
```
sudo dd bs=1m if=/dev/rdisk1 of=/Users/pierre/dev/Raspberry/backup.img
```

(after `of`, put the directory where to put the img file. Reminder, on Terminal, type `pwd` to know where you are) to restore a backup :

```
sudo dd bs=1m of=/dev/rdisk1 if=/Users/pierre/dev/Raspberry/
backup.img
```

After if = path where your image file is.

GPIO and Servo connexion



Servo 0 - Pin GPIO 4 = pin 7 —> X axis
Servo 1 - Pin GPIO 17 = pin 11 —> Y axis
Servo 2 (for eyeblind blinking if needed)- Pin GPIO 18 = pin 12

GND pin is connected to gnd servo power

The code defaults to driving 8 servos, the control signals of which should be connected to P1 header pins as follows:

Servo number	GPIO number	Pin in P1 header
0	4	P1-7
1	17	P1-11
2	18	P1-12

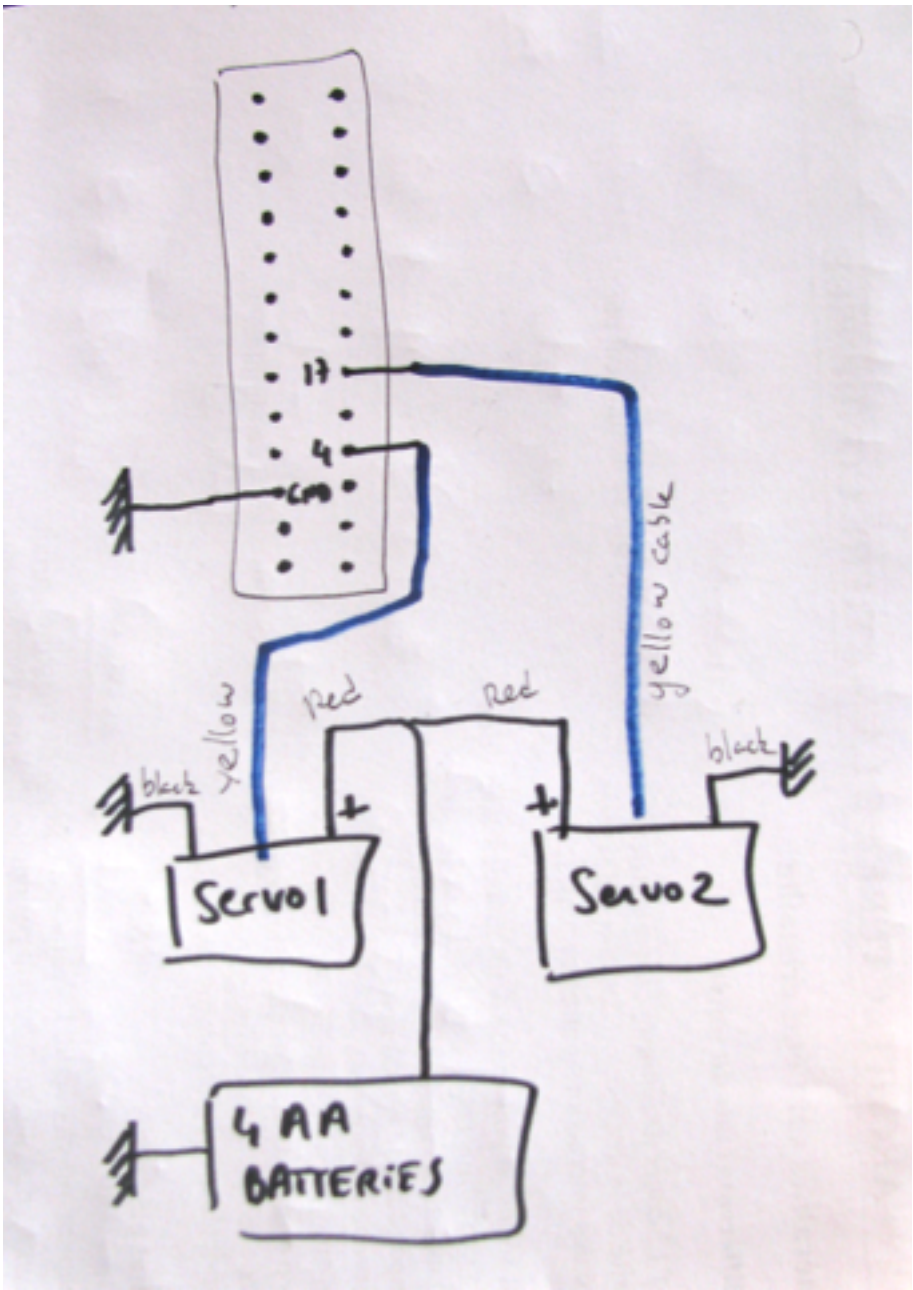
3	21/27	P1-13
4	22	P1-15
5	23	P1-16
6	24	P1-18
7	25	P1-22

to test servo

echo ServoNumber=Percentage > /dev/servoblaster

```
echo 0=10% > /dev/servoblaster
echo 1=70% > /dev/servoblaster
```

Warning : the servo are powered with additionnal batteries (4 AA) and ground are connected together



Synopsis Pierre Raufast

proper title : `Pierre Raufast : Software Magician`
I like this title, is more fun than just « code » or « Raspberry Code ».

Synopsis (difficult exercice, I don’t know how many lines and spirit of this synopsis, feel free to cut information).
Pierre Raufast is a software magician, working for many years on various computers. He lives in France with his wife and two lovely daughters. He also wrote two management books and one fiction book.
In this portrait, he wrote the software running on a Raspberry Pi computer, that controls eyes movement.

Picture is attached in the email.