



Raspberry Pi – User Guide

Description of how to use AMF products with the raspberry pi

Version 01.00

Advanced Microfluidics SA

Emilie Collot
11/10/2017

Document Purpose

This document gathers all the information regarding occlusion detection with the pump intended for insulin use.

Document Status and Revision History

<i>Version</i>	<i>Author</i>	<i>Release date</i>	<i>Modifications</i>	<i>Revised by</i>	<i>Approved by</i>
V01.00	Emilie Collot	11.10.17	First release version		

Reference and applicable documents

Applicable documents

<i>Reference</i>	<i>Document description</i>
AD01	
AD02	

Reference documents

<i>Reference</i>	<i>Document description</i>
RD01	
RD02	

Table of content

1.	Introduction.....	1
1.1.	Setting up the computer for cloud connection.....	1
1.2.	Setting up the raspberry pi for cloud connection.....	1
2.	Writing a python script.....	5
3.	Launching the python script.....	6

1. Introduction

This guide gives you all the necessary information to use AMF products with the raspberry pi. Everything is explained for use with the **raspberry pi 3 model B**. If you use another model, some things may need to be adapted.

To use your raspberry pi as a computer, you can either:

- Plug a screen, keyboard and mouse directly into your raspberry pi
- Setup Real VNC and connect to your raspberry pi through VNC viewer from any other computer (both the raspberry pi and the computer need to be connected to the internet or to the same local network).

1.1. Setting up the cloud connection

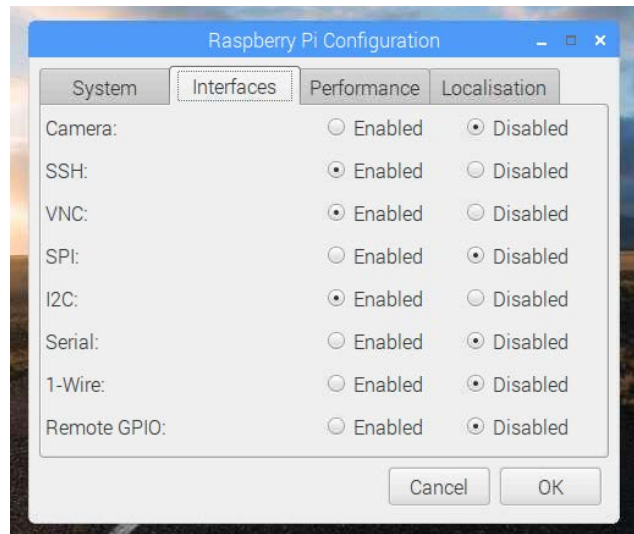
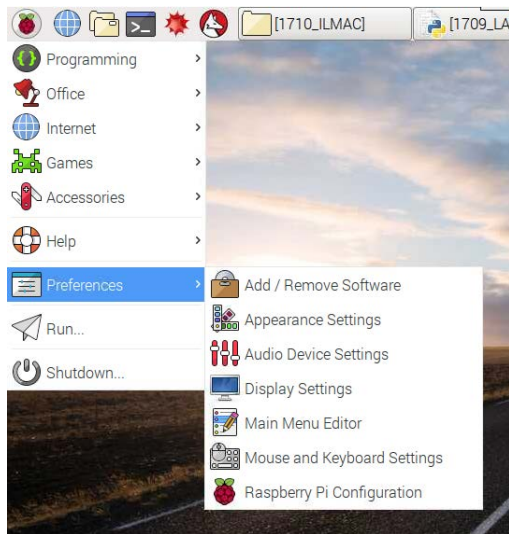
1.1.1. Setting up the computer for cloud connection

1. Go to <https://www.realvnc.com/en/>, click on *Sign in* in the top right corner and create an account.
2. Go to <https://www.realvnc.com/en/connect/download/viewer/>, download VNC Viewer for your operating system and install it.
3. Open VNC Viewer on your computer.
4. Connect to the account you just created. The *Sign in* button is on the top right corner.

1.1.2. Setting up the raspberry pi for cloud connection

You will need a mouse, keyboard and screen directly connected to your raspberry pi the first time you setup the cloud connection.

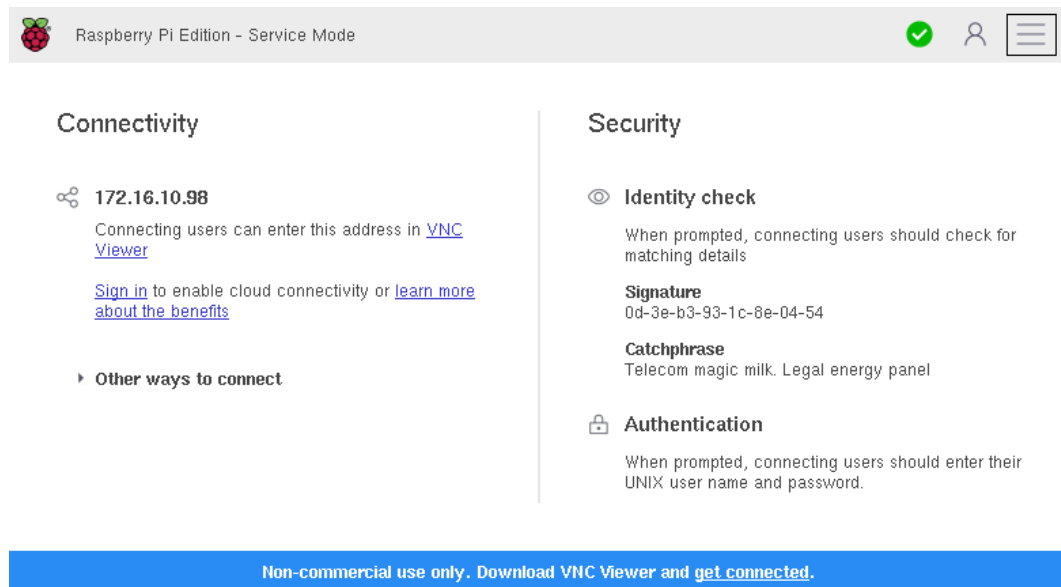
1. Open terminal and execute the series of instructions below (to make sure everything is up to date)
 - a. `sudo apt-get install rpi-update`
 - b. `sudo rpi-update`
 - c. `sudo apt-get update`
 - d. `sudo apt-get upgrade`
 - e. `sudo apt-get dist-upgrade`
 - f. `sudo apt-get install realvnc-vnc-viewer realvnc-vnc-server`
2. Go to Main Menu > Preferences > Raspberry Pi Configurations > Interfaces and check that VNC is enabled.



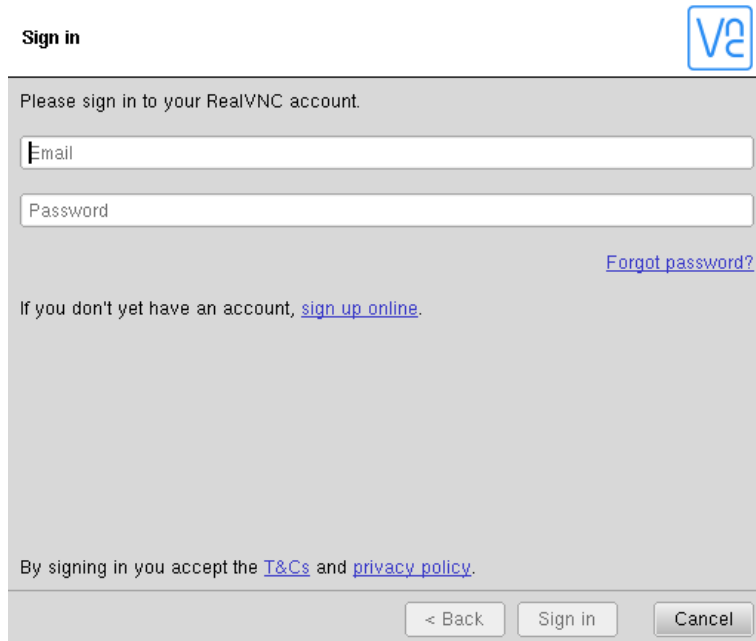
- Open the VNC logo in the top bar (may need to reboot once it is enabled)




- Connect to VNC Server with the login that you created on your other computer. Click on *Sign in* in the middle of the left hand side.



5. Enter your login details, from the Real VNC account that you created previously.



Sign in 

Please sign in to your RealVNC account.

Email

Password

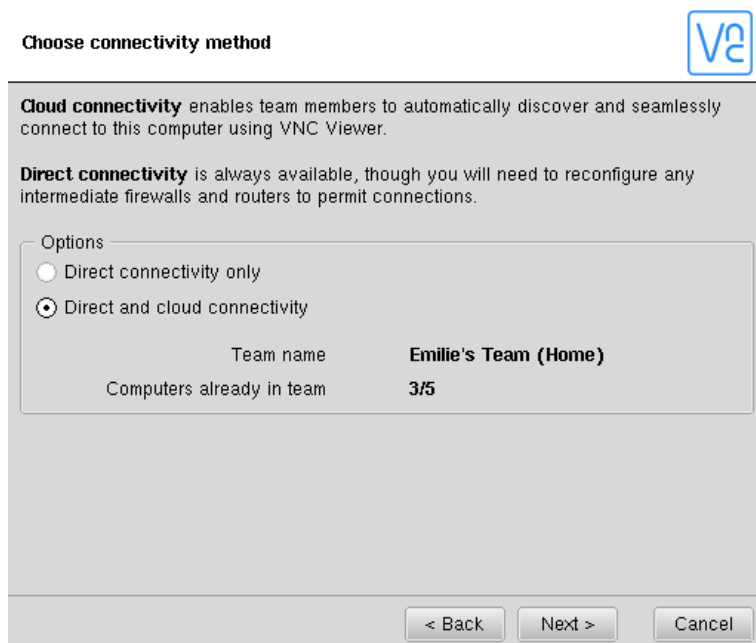
[Forgot password?](#)


If you don't yet have an account, [sign up online](#).

By signing in you accept the [T&Cs](#) and [privacy policy](#).

< Back Sign in Cancel

6. Choose *Direct and cloud connectivity*



Choose connectivity method 

Cloud connectivity enables team members to automatically discover and seamlessly connect to this computer using VNC Viewer.

Direct connectivity is always available, though you will need to reconfigure any intermediate firewalls and routers to permit connections.

Options

☐ Direct connectivity only

☒ Direct and cloud connectivity

Team name	Emilie's Team (Home)
Computers already in team	3/5

< Back Next > Cancel

7. Validate the rest of the steps

Apply changes V2

Please confirm that you want to apply these changes to this computer.

Subscription type	Home
Connectivity	Direct and Cloud
Team name	Emilie's Team (Home)
Computers already in team	3/5


Complete operation V2

Thank you. Your subscription has been successfully applied to this computer.

Team name	Emilie's Team (Home)
Computers now in team	4/5


This computer is now ready to receive connections from VNC Viewer.

8. You're now up and running. You should see in which team this device is. In this case, it's *Emilie's Team*.


Home subscription - Service Mode

✓
⦿
☰

Connectivity


 **rpi-3**

Belonging to **Emilie's Team (Home)**

Connecting users see this information when they sign in to [VNC Viewer](#)

▶ **Other ways to connect**


Security

 **Identity check**

When prompted, connecting users should check for matching details

Signature
0d-3e-b3-93-1c-8e-04-54

Catchphrase
Telecom magic milk. Legal energy panel

 **Authentication**

When prompted, connecting users should enter their UNIX user name and password.

Non-commercial use only. Download [VNC Viewer](#) and [get connected](#).

5. You can now view the raspberry in VNC Viewer on the other computer. (You may need to click on *Refresh* in VNC Viewer).

Default username: pi

Default password: raspberry

For more information, please view <https://www.realvnc.com/en/raspberrypi/>

2. Writing a python script

Before even writing a python script, please ensure that your raspberry pi is up to date.

To communicate with our devices, you will need to install the pyserial library:

[sudo apt-get install python-serial](#)

1. Create the file
 - a. Go the folder where you'd like your script to be saved
 - b. Create a text file, and give it the name that you would like with the extension ".py" at the end (e.g. raspberryPiDemo.py)
2. Prepare the file (a detailed example will be shown after this section)
 - a. At the very top of your file, before you write anything else (even before the description of the file), you must add these 2 lines:
 - i. `#!/usr/bin/env python`
 - ii. `# -*- coding: UTF8 -*-`
 - b. Just under the description, you must include the libraries that you need to communicate with our systems:
 - i. `import sys`
 - ii. `import serial`
 - iii. `import time`
 - c. At the end of the file, add the text `sys.exit(0)`
3. Communicate with the devices
 - a. To connect to a device, you need to use the serial library and specify the communication port and the baudrate (the rest is right by default):
 - i. The usual COM port found in Windows OS is not quite defined in the same way. You rather access your device by specifying on which USB port you are connected:
`/dev/ttyUSB1`
 1. To check that your devices are correctly plugged in, you can use the instruction
[ls /dev/serial/by-id](#)
 2. If you still can't connect, you may just need to change the port number, e.g.
`/dev/ttyUSB0`
 - ii. Baudrate = 9600
 - iii. `deviceName = serial.Serial('/dev/ttyUSB0',9600)`
 - b. When you send an instruction to a device (or several devices), you need to wait a certain amount of time until the instruction is finished. You do so by adding the `time.sleep(X)` command, where X is the number of seconds that you would like to wait.
 - c. The commands for each instruction is found in the user manuals for each device. Please go to our downloads page to check out the latest versions:
<https://amf.ch/new/downloads>
Password: downloads@AMF

```

1  #!/usr/bin/env python
2  # -*- coding: UTF8 -*-
3
4  """
5  raspberryPiDemo.py
6
7  ~~~~~
8  This snippet is an example to show how to use our products with a raspberry pi.
9  You will need to install the pyserial library
10
11  To execute the code:
12  1. Open Terminal.
13  2. Go to the directory in which the file is saved using the command cd
14  (e.g. cd /home/pi/Documents/Demos/)
15  3. Launch the file with the command python
16  (e.g. python raspberryPiDemo.py)
17
18  For any other information, please refer to the document "RaspberryPi - UserGuide".
19  If you still can't find what you are looking for, don't hesitate to contact us at
20  info@amf.ch
21
22  """
23
24  import sys
25  import serial
26  import time
27
28  # SPM initialisation
29  spm = serial.Serial('/dev/ttyUSB1',9600,timeout=1000)
30  print('SPM',spm.name)
31  spm.write(b"/1ZR\r") # initialise SPM
32
33  # RVM initialisation
34  rvm = serial.Serial('/dev/ttyUSB0',9600)
35  print('RVM',rvm.name)
36  rvm.write(b"/1ZR\r") # initialise RVM
37  time.sleep(30) #leave some time for both devices to finish initialising
38  print("Initialised")
39
40
41  # Example for a priming sequence
42  lsp.write(b"/1B2V600M1000A2000R\r")
43  # b"/1R\r" is the "standard code" for sending a command
44  # B2 - move to port 2 with the shortest path
45  # M1000 - wait 1000 milliseconds
46  # A2000 - go to absolute position 1000 (resolution 3000 by default. Use command N to change resolution)
47
48  time.sleep(9) # leave some time to execute command on top
49
50  lsp.write(b"/1B1V1000M1000A0R\r") # empty syringe through port 1
51  time.sleep(6)
52
53  sys.exit(0)

```

3. Launching the python script

1. Open Terminal
2. Go to the directory in which the file is saved using the command `cd`
(e.g. `cd /home/pi/Documents/Demos/`)
3. Launch the file with the command `python` (e.g. `python raspberryPiDemo.py`)