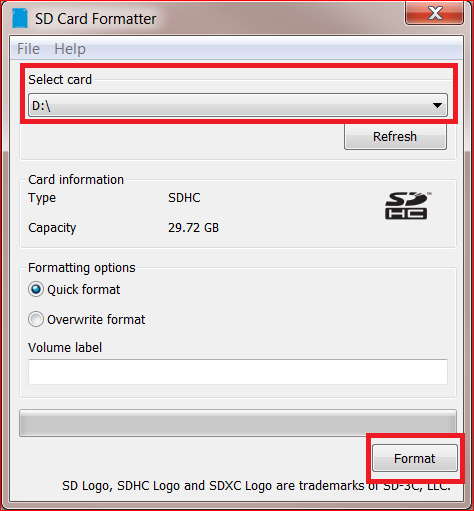
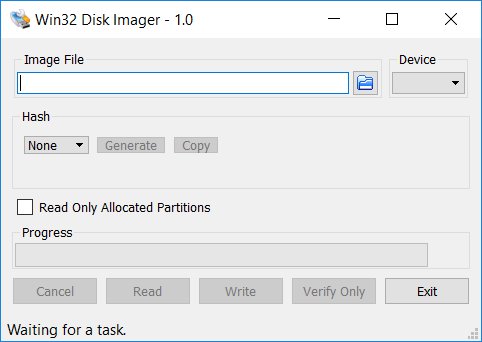
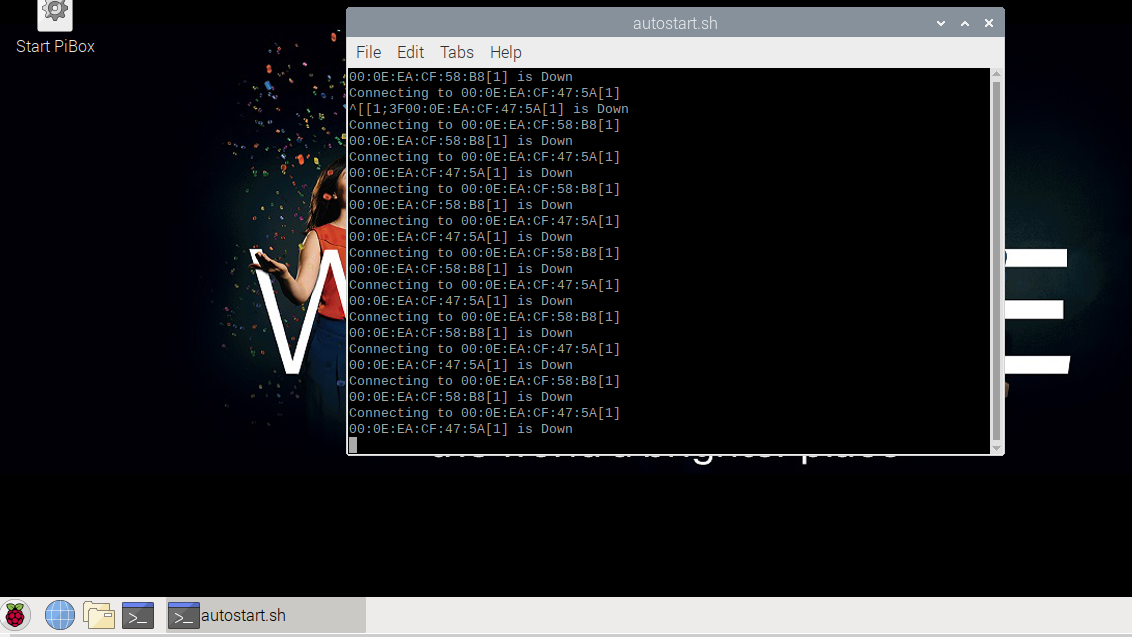
Installation of EasyCasting PiBox Mock



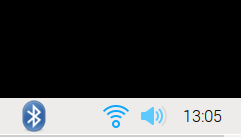
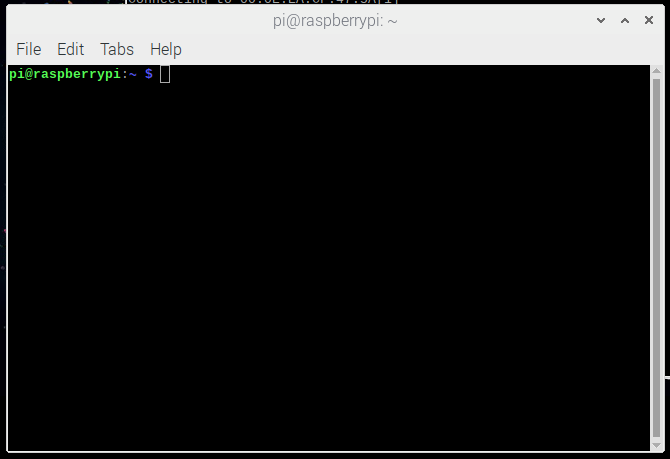
1. Create the SDCard

* Download and install SDCard Formatter : <https://www.sdcard.org/downloads/formatter/>
* Format a 32Go micro SDCard (with adapter) with SDCard Formatter with the option Overwrite Format
* 
* Download the Raspbian Pibox image here : www.cyrilvincent.com/download/raspbian-pibox-mock.rar
* Dezip the file to obtain a .img
* Download and install win32diskimager: <https://sourceforge.net/projects/win32diskimager/>
* 
* Put the image file on the SDCard with win32diskimager
* Eject properly de SDCard
* Sometimes this step doesn't work, it's due to the different size of the card
* We can create manually the Raspian OS look at the before last chapter : Install the Raspberry from scratch

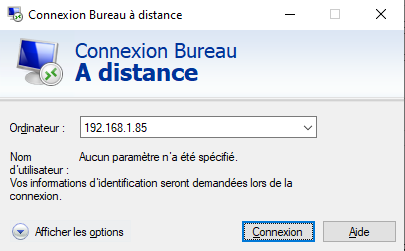
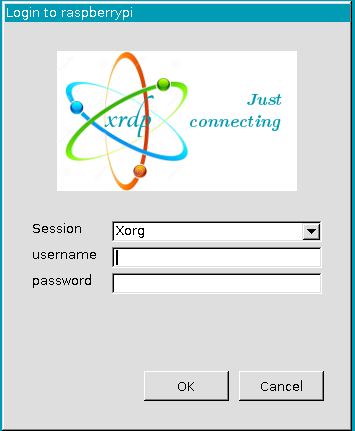
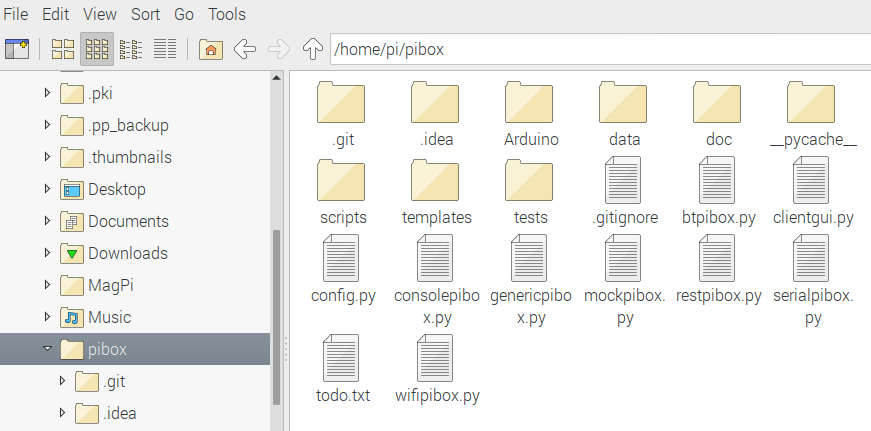
1. First boot of the Raspberry

* Put the SDCard into the Raspberry
* Plug a keyboard and a mouse on the USB
* Plug a screen on the micro-HDMI port
* Plug the Raspberry with a power supply USB – C 5.1V 2A
* 

1. Network

* Network is mandatory to access and debug the Raspberry from a PC
* Internet is mandatory to install drivers while the development process
* Connect to the WIFI with the Wifi icon at the bottom right of the screen
* 
* Test Internet with the Chromium browser (first icon after de raspberry icon)
* 
* If you have only an Ethernet connection plug it but it will be more difficult to share file
* Open a terminal with the terminal icon 
* 
* Type ifconfig
* Note your ip address in the section wlan0 / inet
* 
* In this example the ip address is 192.168.1.104

1. Remote access with Remote Desktop

* On Windows open a remote desktop
* 
* Enter raspberrypi, if it does not work enter the raspberry ip address
* The login is pi, the password is raspberry
* 
* You can now unplug the screen and the keyboard from the raspberry
* The code is into the pibox directory
* 
* This documentation is into pibox/doc directory
* Never change a file into the pibox directory, except config.py

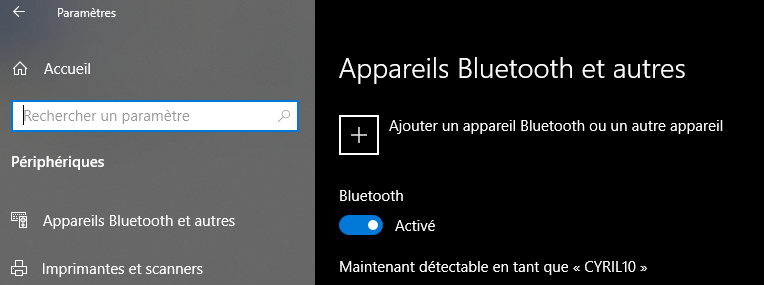
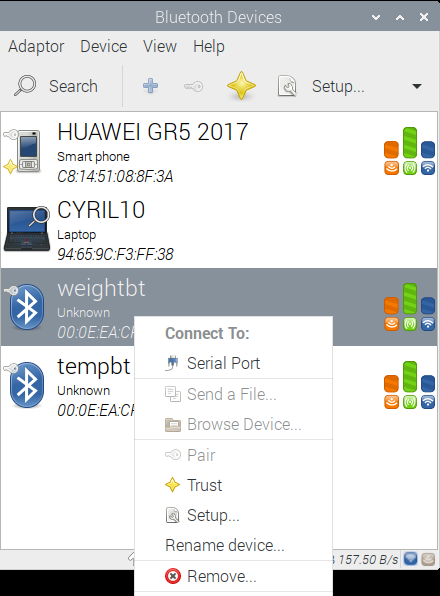
1. Remote access with Samba

* You can have a remote access to a shared directory with the Samba protocol
* On your PC open a file explorer
* Type [\\raspberrypi](file:///\\raspberrypi) or ip address
* You should see the directory of the pi user
* This documentation is into pibox/doc directory
* Never change a file into the pibox directory, except config.py
* Make a copy of the pibox directory on your PC

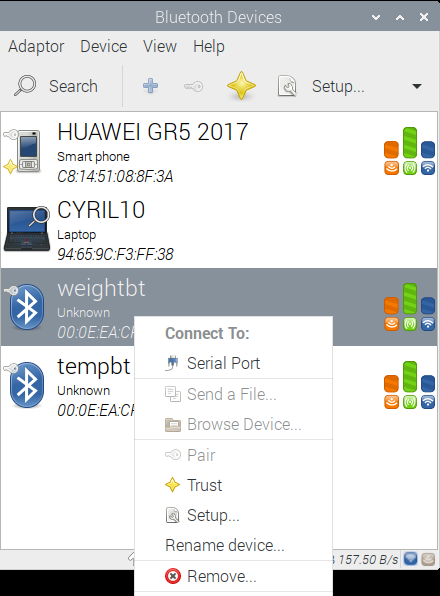
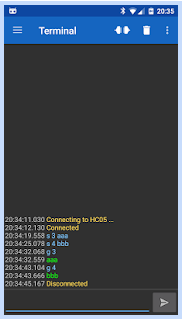
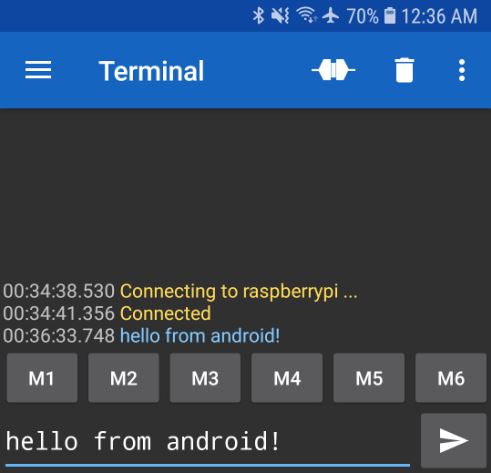
1. The PiBox application

* The Raspberry and Arduino applications and source code are into the pibox folder
* To stop the application just type Enter into the terminal and wait 12 seconds
* To start the application double click on Start PiBox on the desktop
* 

1. Pair the Raspberry with Windows

* On windows 10, type Bluetooth in Cortana and find the Bluetooth Parameter screen
* 
* On Raspbian click on the Bluetooth icon
* 
* If the Bluetooth icon does not appear, unplug the screen from the Raspberry and reboot
* Click on Make Discoverable
* Accept connections
* On the Bluetooth icon on the raspberry click on Devices
* You should see your PC
* 
* Check the little star which means the device is safe (press the big yellow star)

1. Pair the Raspberry with Android

* On Raspbian click on the Bluetooth icon
* Click on Make Discoverable
* 
* On Android, into the bluetooth parameters make the phone visible and search the Raspberrypi device
* On the Bluetooth icon on the raspberry click on Devices
* You should see your phone
* 
* Check the little star which means the device is safe (press the big yellow star)
* Install the Serial Bluetooth Terminal : <https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal>
* Launch de Serial Bluetooth Terminal
* 
* Click on the 3 bars 
* Choose the raspberrypi
* 
* You should see something like this every 2s :
* {pho:-2, temp:20.4, pre:1000, wei:2500, mix:0}
* -4 means device is down, try to reconnect every 10s
* -3 means the device is normally disconnected
* -2 means the device is disconnect, try to connect now
* -1 means the device is connected but no value was sent
* 0 means the device is OK
* > 0 means the last value send by the device, with a timeout of 60s
* Type on your phone 4 + Send 
* Youd should receive {pho:4, temp:-4, pre:-4, wei:-4, mix:100010}
* Data send by Bluetooth respects the JSON standard
* Mix data means
* -4 means device is down, try to reconnect every 10s
* -3 means the device is normally disconnected
* -2 means the device is disconnect, try to connect now
* -1 means the device is connected but no value was sent
* 0 means the device is OK but no data received
* > 1 you should receive a 6 bits binary
* The first bit = 1 means it's ok
* The 5 other bits simulate the 5 mixers's outputs
* Test with other values
* You should configure values in pibox/config.py in the hardwareConfig section and in the pibox/data directory

1. The code

* All the code is under the pibox directory
* The main program is btpibox.py
* All raspbian scripts are into the scripts folder
* Communications between phone and raspberry can be done with Bluetooth 5, Wifi and Rest API + Websocket
* All communications between raspberry and devices can be done with Bluetooth 3 or USB
* The code is updated automatically at every start via Git
* Never modify a file under the pibox directory except the config.py
* Raspberry development : Python 3.7
* Raspberry scripts : bash
* Arduino development : C++ 11 + GCC 11 Arduino 1.8
* Raspberry OS : Raspbian Buster 10 based on Debian Buster 10 ARM32
* IDE Python : Pycharm 2019 CE
* IDE Arduino : Arduino IDE 1.8
* Tested on Raspian 10 and Windows 10
* Tested on Seeeduino board