

# IUT RÉSEAUX & TÉLÉCOMS

SAE 23 / 24



# Sommaire

1 Projet PHOTO\_ATB

2 Raspberry Pi + Pico

3 Caméra

4 Fiche Anomalie

5 Plan Validation (part 1)

6

# Projet Système Prise Photos Banc Avionique

1 Contexte

2 Objectif / Entrées et Sorties

3 Compétences mises en œuvre

4 Entrées & Livrables

5 Organisation

6 Semestre 2

# CONTEXTE

## /// Contexte

- / Un banc avionique est composé de plusieurs équipements reliés par des dizaines de câbles eux même composés de centaines de fils
- / Pour valider une avionique, il est nécessaire d'investiguer / vérifier des comportements physiques en ajoutant des équipements de mesure (oscilloscope, multimètre, boîtes éclatées,...) et/ou de corriger des câbles mal définis ou réalisés
- / Ces interventions / modifications ne sont pas forcément tracés (car ponctuel) mais elles agissent sur la configuration du banc et donc sur les résultats obtenus

## /// Situation initiale

- / Aucune indication sur la configuration réel du banc au niveau du matériel n'est conservé dans le cas de manipulation temporaire



# EXEMPLE D'UN BANC AVIONIQUE



Boîtes  
éclatées

Bouchons



Espion  
externe

# • OBJECTIF

## /// Objectif

- / Mettre en place un système qui prenne des photos du banc avant chaque exécution de test ou sur demande
- / Le système devra gérer la prise de photo automatique en journée et durant la nuit, l'archivage des photos avec un site web permettant de parcourir l'historique de toutes les photos disponibles
- / Le système devra gérer un système d'éclairage en cas de besoin
- / Le système devra gérer les utilisateurs autorisés

## /// Situation finale

- / Photo de l'état du banc au démarrage d'un test (de manière automatique) ou sur demande (en journée et la nuit)
- / Conservation des photos
- / Présentation des photos à l'utilisateur
- / Suppression des photos
- / Gestion des utilisateurs

# BESOIN FONCTIONNEL

## /// Besoin Fonctionnel

- / Site web local (pas de connexion internet)
  - / Se connecter au site avec login et mot de passe
  - / De manière autonome (au bout de 24h si aucune autre photo prise), prendre une photo et lui donner un nom
  - / Sur demande suite à un appel en ligne de commande, prendre une photo et lui donner un nom
  - / 1 seul utilisateur accède au site à la fois (pas d'accès concurrent)
- 
- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>/ En tant qu'utilisateur<ul style="list-style-type: none"><li>▪ Modifier son mot de passe</li><li>▪ Prendre une photo et lui donner un nom</li><li>▪ Pouvoir identifier des photos à supprimer</li><li>▪ Parcourir les photos enregistrer</li></ul></li></ul> | <ul style="list-style-type: none"><li>/ En tant qu'administrateur<ul style="list-style-type: none"><li>▪ Créer ou supprimer un compte utilisateur ou administrateur</li><li>▪ Modifier le mot de passe d'un utilisateur</li><li>▪ Débloquer le compte d'un utilisateur</li><li>▪ Modifier le nom du programme sur lequel le système est installé</li><li>▪ Supprimer définitivement une photo</li><li>▪ Parcourir les log de l'application</li></ul></li></ul> |
|---|--|

# ENTRÉES & LIVRABLES

## /// Entrées

- / Identification Projet = PHOTO\_ATB
- / Description du besoin fonctionnel
- / Exigences pour la gestion des mots de passe
- / Liste des contraintes
  - Doit fonctionner sur un système linux
  - Utilisation de logiciels / librairies libres de droit pour les entreprises
  - Utilisation github en mode privé



## /// Matériel à disposition

- / 1 raspberry PI 3 + 1 caméra + 1 détecteur niveau luminosité + 1 LED + 1 raspberry PICO WH
  - PI 3 → site web, base de données, caméra
  - Pico → détecteur niveau luminosité + LED



## /// Livrables

- / Source des programmes informatiques avec commentaires
- / Site web + programme python opérationnels
- / Document décrivant le design de la solution avec les exigences
- / Plan de validation / rapport des essais effectués





# • PÉRIMÈTRES & OBJECTIFS DES SAE

## /// SAE15

→ Semestre 1

### / Spécification du produit

- Exigences du produit
- Architecture logicielle
- Organigrammes (algorithmes)
- Maquette site web

### / Projet git mis en place et à jour

## /// SAE23

→ Semestre 2

### / Site web sécurisé et « écolo » (tout élément est utile et n'entraîne pas une surconsommation inutile)

### / Base de Données

### / Projet git mis à jour

## /// SAE24

→ Semestre 2

### / Application complète opérationnelle et validée

- Site web + Base de Données + programme python

### / Documentation complète

### / Projet git mis à jour

# • SEMESTRE 2

## / Amphi S2:

- Présentation Matériel

## / 28/02

- Plan de Validation
- Travail à faire:
  - Mise à jour de l'organisation de l'équipe (qui fait quoi) → sur github
  - Planification des activités du S2 → sur github
  - Mise à jour schéma Base de Données → sur github
  - Canevas Fiche d'Anomalie → sur github
  - Configuration Raspberry + Pico

## / 21/03 – TD SAE24

- Retour sur les rapports du semestre 1

## / 04/04 + 25/04 + 16/05 + 06/06 – TD SAE24

- Avancement des activités

## / 11/06: Remise des rapports pour SAE23 + SAE24

## / 19/06 : Soutenance pour notation S2 pour SAE23 + SAE24

# • SEMESTRE 2

## ///Rapport pour SAE23 + SAE24

/ Format: Word

/ Nom fichier: <nom\_equipe>\_SAE23\_24\_rapport.docx

/ Contenu

- Programme informatique: commentaires, nom des variables et des fonctions en anglais
- Description de l'installation + configuration du raspberry PI 3 et PICO WH (système + applications + configuration)
- Schéma électrique du montage
- Décrire les activités à faire pour configurer le raspberry sur une adresse IP particulière pour un sous-réseau spécifique
- Plan de Validation, Procédures de test, Rapports de test et fiches d'anomalies
- Décrire échanges entre pages du site web (schéma avec les infos qui transitent et explications)
- Gestion de Configuration des logiciels
- Gestion de projet (statut des risques, RACI, planification, Retour d'Expérience de l'équipe)

# • SEMESTRE 2

## ///Soutenance

/ Format: powerpoint

/ Nom fichier: <nom\_equipe>\_SAE23\_24\_presentation.docx

/ Contenu

- xxx

# RASPBERRY

1 A quoi tu sers ?

2 Y a quoi dedans ?

3 Une famille qui s'agrandit ...

4 Parlons logiciel

5 Installation

6 PICO



# MUSIC PLAYER



# MUSIC BOX & SYNTHÉTISEUR

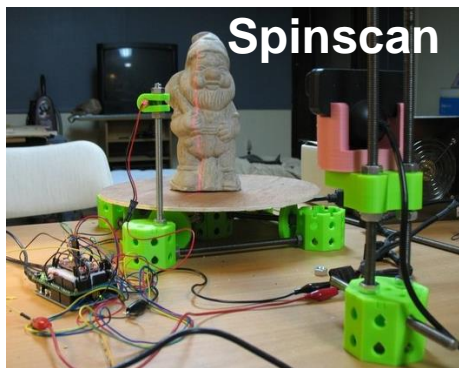
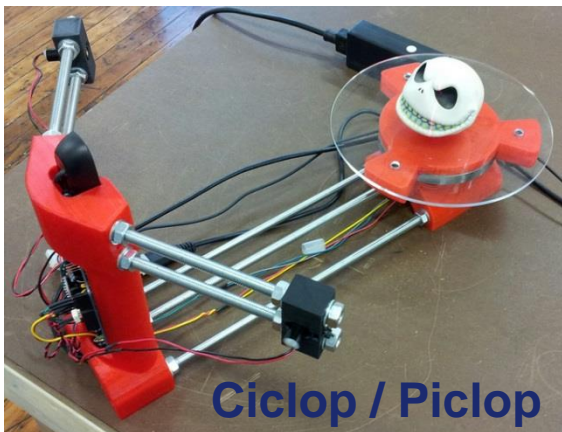


<http://zynthian.org/#section-home>

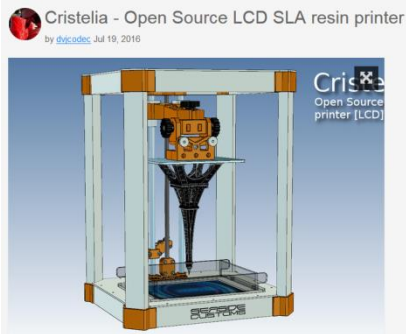
# RETRO GAMING



# SCANNER 3D



# IMPRIMANTE 3D



<https://www.thingiverse.com/thing:1680172>

# UV LCD DLP 3D PRINTER

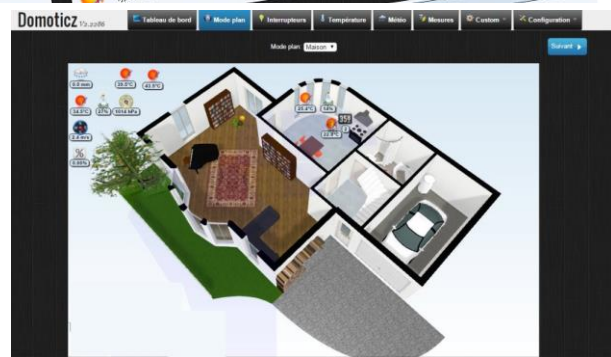


UV LCD SLA 3D Printer - presentation

<https://www.youtube.com/watch?v=UkXdtqxKLL0>

# DOMOTIQUE

<https://domoticz.com/>

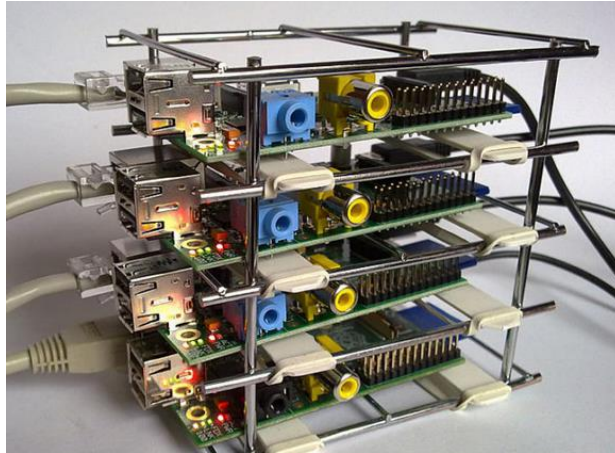


<https://www.thingiverse.com/thing:9972>

# CLUSTERING

Mise en parallèle de plusieurs Raspberry de 3 à plus de 100 ...

→ Pour des traitements massivement parallèles



<https://www.raspberrypi.org/magpi/cluster-computer-raspberry-pi-3/>

<https://www.framboise314.fr/un-cluster-de-raspberry-pi-avec-kubernetes/>

<https://raspberrypi.fr/docker-swarm-raspberry-pi/>





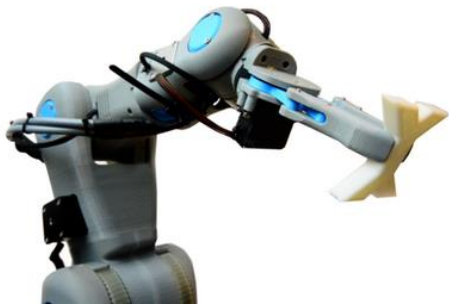
# ROBOTIQUE (R.O.S.)

<https://www.ros.org/>  
<https://robots.ros.org/>

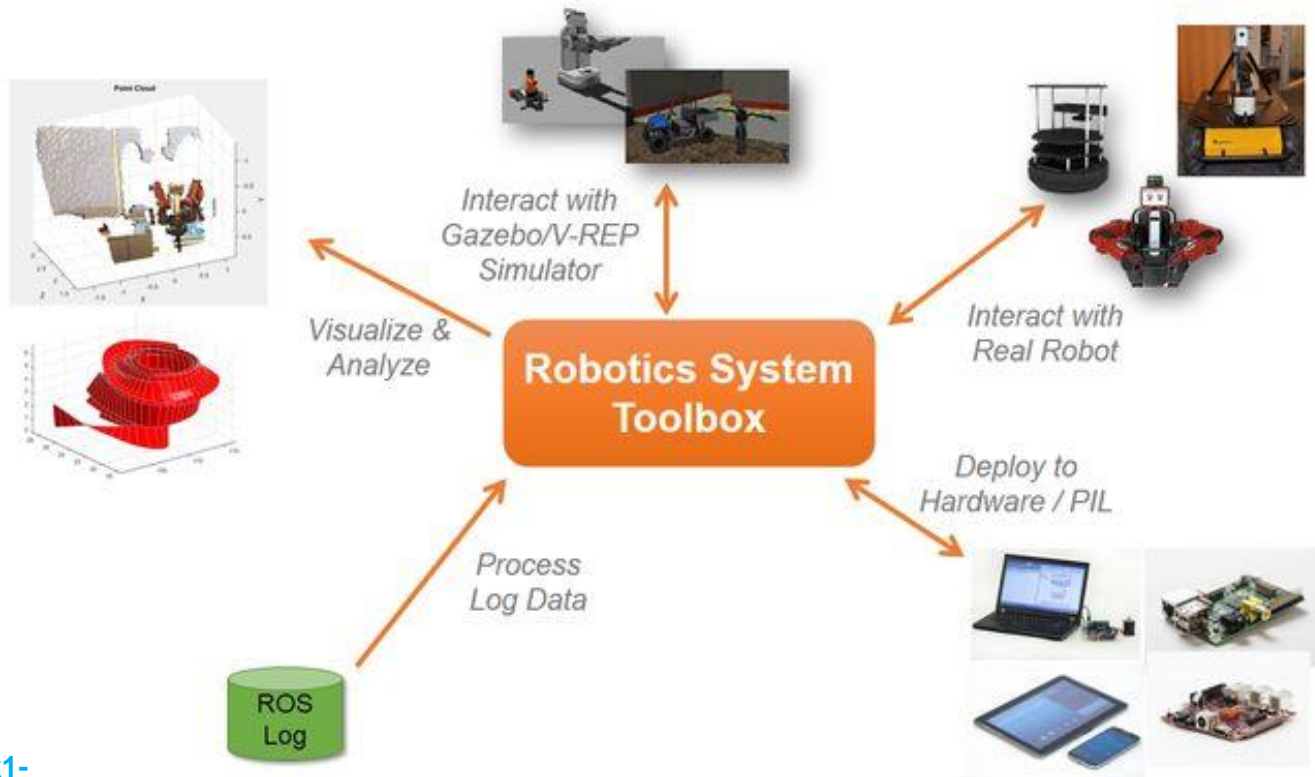
ROS



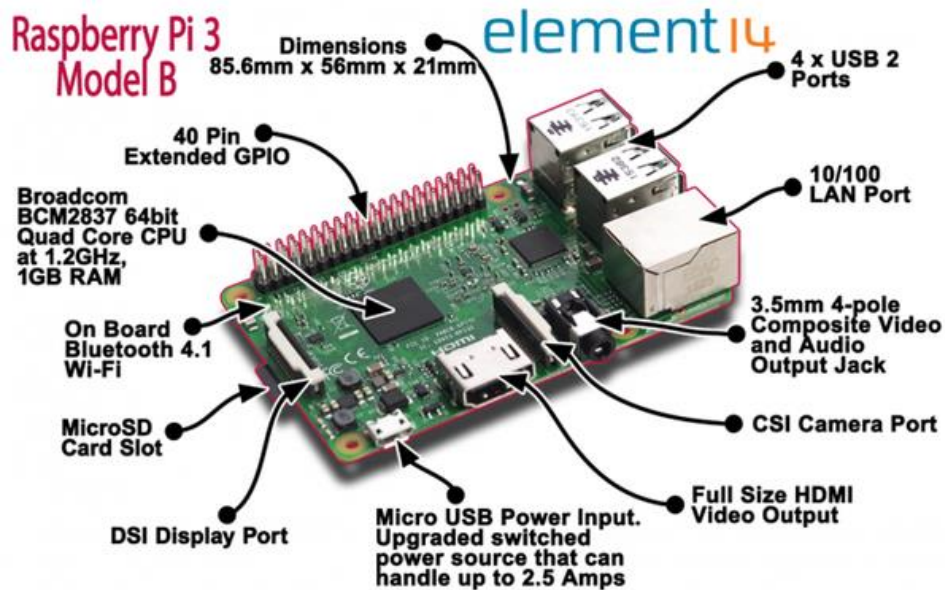
<https://niryo.com/fr/>



<https://roboteurs.com/products/rbx1-remix-3d-printed-6-axis-robot-arm-kit>



# Y A QUOI DEDANS ?



Un ordinateur **complet** sur une seule carte de la taille d'une carte de crédit !

Aussi appelé nano-ordinateur en raison de sa taille

## Composants :

- Processeur ARM & RAM
- Carte vidéo
- Différents ports d'interface
- « Disque dur »
- Connectivité sans fil

## Alimentation:

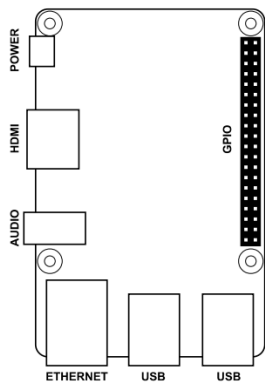
5 V





# General Purpose Input/Output

- Ajouter des extensions
- Chaque broche à un rôle spécifique (entrée, sortie, alimentation, masse, gestion de bus de données, ...)

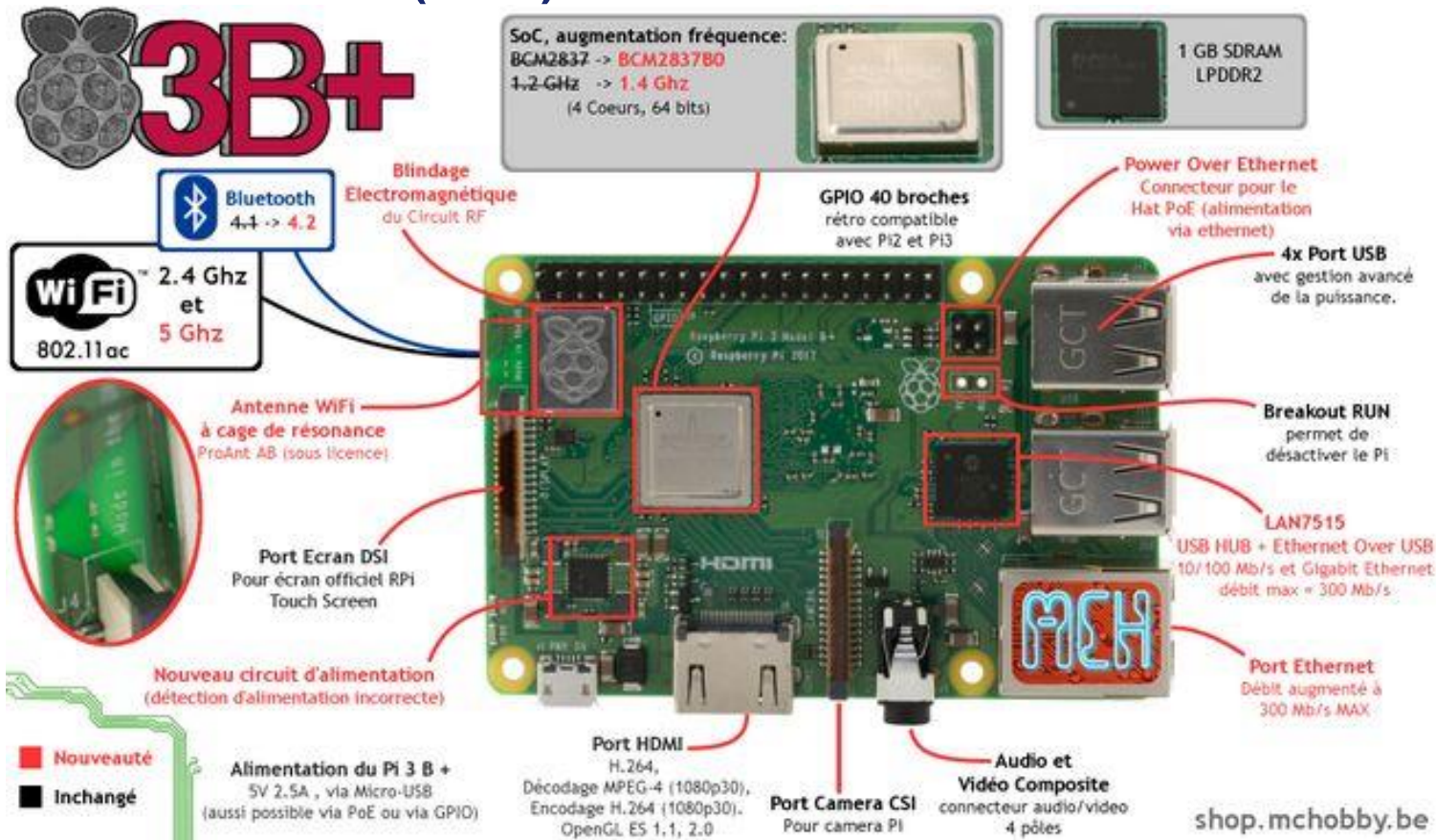


Power 3V3	1	2	Power 5V
HotKey J1	3	4	Power 5V
HotKey J2	5	6	GND
Haut J1	7	8	Ⓛ / Btn3 J1
GND	9	10	Ⓧ / Btn2 J1
Bas J1	11	12	Ⓛ / Btn1 J1
Gauche J1	13	14	GND
Droite J1	15	16	Ⓡ / Btn6 J1
Power 3V3	17	18	Ⓛ / Btn5 J1
Start J1	19	20	GND
Select J1	21	22	Ⓡ / Btn4 J1
Haut J2	23	24	Ⓛ / Btn3 J2
GND	25	26	Ⓧ / Btn2 J2
Non utilisé	27	28	Non utilisé
Bas J2	29	30	GND
Gauche J2	31	32	Ⓛ / Btn1 J2
Droite J2	33	34	GND
Start J2	35	36	Ⓡ / Btn6 J2
Select J2	37	38	Ⓛ / Btn5 J2
GND	39	40	Ⓡ / Btn4 J2











Peripherals	GPIO		Particle	Pin #		Pin #	Particle	GPIO	Peripherals
3.3V				1	X	X	2	5V	
I2C	GPIO2	SDA	3	X	X	4	5V		
	GPIO3	SCL	5	X	X	6	GND		
Digital I/O	GPIO4	D0	7	X	X	8	TX	GPIO14	UART
GND			9	X	X	10	RX	GPIO15	Serial 1
Digital I/O	GPIO17	D1	11	X	X	12	D9/A0	GPIO18	PWM 1
Digital I/O	GPIO27	D2	13	X	X	14	GND		
Digital I/O	GPIO22	D3	15	X	X	16	D10/A1	GPIO23	Digital I/O
3.3V			17	X	X	18	D11/A2	GPIO24	Digital I/O
SPI	GPIO10	MOSI	19	X	X	20	GND		
	GPIO9	MISO	21	X	X	22	D12/A3	GPIO25	Digital I/O
	GPIO11	SCK	23	X	X	24	CE0	GPIO8	SPI
GND			25	X	X	26	CE1	GPIO7	(chip enable)
DO NOT USE	ID_SD	DO NOT USE	27	X	X	28	DO NOT USE	ID_SC	DO NOT USE
Digital I/O	GPIO5	D4	29	X	X	30	GND		
Digital I/O	GPIO6	D5	31	X	X	32	D13/A4	GPIO12	Digital I/O
PWM 2	GPIO13	D6	33	X	X	34	GND		
PWM 2	GPIO19	D7	35	X	X	36	D14/A5	GPIO16	PWM 1
Digital I/O	GPIO26	D8	37	X	X	38	D15/A6	GPIO20	Digital I/O
GND			39	X	X	40	D16/A7	GPIO21	Digital I/O

# RASPBERRY PI 3 (2016)



## PERFORMANCES DES RASPBERRY

<http://socialcompare.com/fr/comparison/raspberrypi-models-comparison>

	Raspberry Pi 3 B+	Raspberry Pi Zero WH	Raspberry Pi Zero W	Raspberry Pi 3	Raspberry Pi Zero	Raspberry Pi 2	Raspberry Pi A+	Raspberry Pi B
Image								
Date de sortie	14 mars 2018	12 janv. 2018	28 févr. 2017	29 févr. 2016	30 nov. 2015	1 févr. 2015	10 nov. 2014	15 févr. 2012
Description		Même carte que Raspberry Pi Zero W mais avec un connecteur GPIO déjà soudé						model B mostly w/some model A detail
Caractéristiques								
Prix	35,00 \$US	15,00 \$US	10,00 \$US	35,00 \$US	5,00 \$US	35,00 \$US	20,00 \$US	
SOC								
SOC Type	Broadcom BCM2837B0	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2837	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2835	Broadcom BCM2835
Core Type	Cortex-A53 64-bit	ARM1176JZF-S	ARM1176JZF-S	Cortex-A53 64-bit	ARM1176JZF-S	Cortex-A7	ARM1176JZF-S	ARM1176JZF-S
No. Of Cores	4	1	1	4	1	4	1	1
GPU	VideoCore IV	VideoCore IV	VideoCore IV	VideoCore IV 1080p@30	VideoCore IV	VideoCore IV	VideoCore IV	VideoCore IV 1080p@30
CPU Clock	1,4 GHz	1 GHz	1 GHz	1,2 GHz	1 GHz	900 MHz	700 MHz	700 MHz
Mémoire vive	1 GB	512 MB	512 MB	1 GB DDR2	512 MB	1 GB	256 MB	512 MB
Wired Connectivity								
USB Ports	✔ 4xUSB 2.0	✔ micro & micro OTG	✔ micro & micro OTG	✔ 4	✔ micro + micro OTG	✔ 4	✔ 1	✖ 2
Ethernet	✔ Gigabit - Over USB 2.0	✖	✖	✔	✖	✔ 10/100M	✖	✔
SATA Ports	✖	✖	✖	✖	✖	✖	✖	✖
Port HDMI	✔ 1	✔ mini	✔ mini	✔	✔ mini	✔	✔	✔ 1.3
Analog Video Out	✔ shared with audio jack	✔ via unpopulated pin	✔ via unpopulated pin	✔ shared with audio jack	✔ via unpopulated pin	✔ shared with audio jack	✔ shared with audio jack	✔ Composite
Analog Audio Out	✔ 3.5mm jack	✖ HDMI audio	✖ HDMI audio	✔	✖ HDMI audio	✔	✔	✔
Analog Audio In	✖	✖	✖	✖	✖	✖	✖	✖ USB mic or sound-card could be added
SPI	✔	✔	✔	✔	✔	✔	✔	✔
I2C	✔	✔	✔	✔	✔	✔	✔	✔
GPIO	✔ 40-pin	✔	✔	✔ 40-pin	✔	✔	✔	✖ 26-pins
LCD Panel	✔	✖	✖	✔	✖	✔	✔	✔ DSI
Camera	✔	✔	✔	✔	✔ latest version include a camera connector	✔	✔	✔ DSI
SD/MMC	✔ micro-SD	✔ microSD	✔ microSD	✔ microSD	✔ microSD	✔ microSD	✔ microSD	✖ SD
Serial	✔ RX/TX UART	✖	✖	✖	✖	✖	✖	✖ Through Expansion Connector, needs level shifting
Wireless Connectivity (On-Board)								
Wi-Fi	✔ 2.4GHz and 5GHz 802.11 b/g/n/ac	✔ 802.11n	✔ 802.11n	✔ 802.11n	✖	✖	✖	✖
Bluetooth®	✔ 4.2, BLE	✔ 4.1	✔ 4.1	✔ 4.1 LE	✖	✖	✖	✖

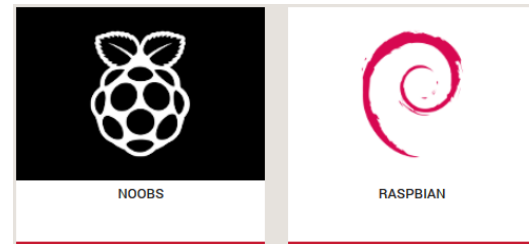


/// Le Raspberry PI est un ordinateur personnel qui a besoin

- D'un système d'exploitation
- De pour pouvoir installer d'autres applications

/// Originellement développé dans l'esprit logiciel libre → solution LINUX

- ! Le premier OS : RASPBERRY DEBIAN → RASPBIAN
- ! Pour démarrer facilement: NOOBS qui comporte plusieurs choix au démarrage

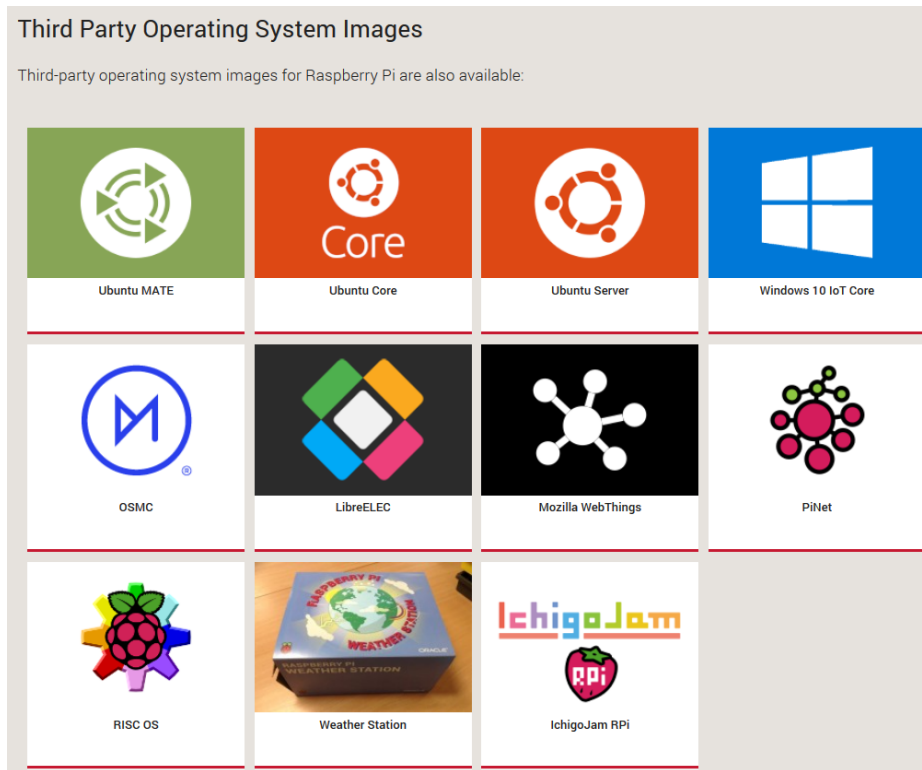


/// Pour télécharger les OS:

<https://www.raspberrypi.org/downloads/>

/// Connection internet recommandée pour mise à jour de l'OS juste après installation

# PLUSIEURS AUTRES DISTRIBUTIONS POSSIBLES





# INSTALLATION SYSTÈME

/// Installer un système exploitation sur la carte Micro-SD :

- ! Télécharger l'image de l'OS que vous voulez installer
- ! Télécharger & installer le logiciel SDFormater
- ! Télécharger & installer le logiciel Win32DiskManager

/// Connecter la carte SD à l'ordinateur

/// Formater la carte avec SDFormater

/// Installer l'image avec Win32DiskManager



# CONFIGURATION RASPBERRY

/// Installer la carte SD dans la raspberry

/// Connecter écran + clavier + souris

/// Brancher l'alimentation

/// Lancer un terminal et exécuter la commande : `sudo raspi-config`

■ Passer le clavier QWERTY en AZERTY

■ Passer le Raspberry en français

■ Changer le mot de passe si nécessaire

■ Configurer le wifi

■ Activer les interfaces nécessaire pour votre projet

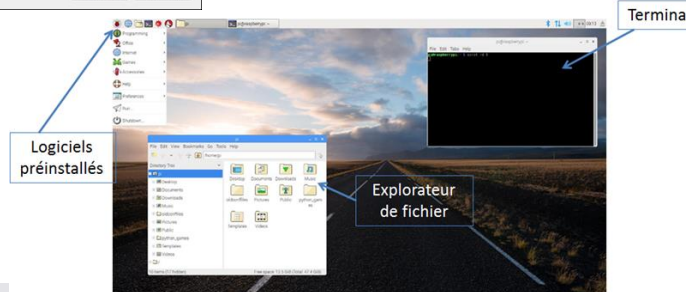
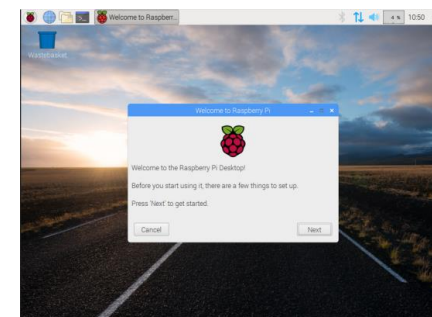
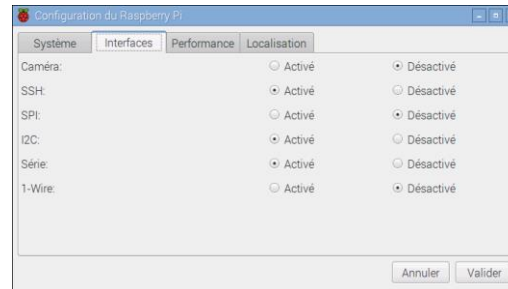
/// Mise à jour du système

■ Exécution des commandes: `sudo apt update` puis `sudo apt upgrade`

■ Installer package gpio: `sudo apt install rpi.gpio`

/// Installation des librairies et logiciels dont vous avez besoin

■ Commande `sudo apt install xxx`



# UTILISATION RASPBERRY

## /// En direct

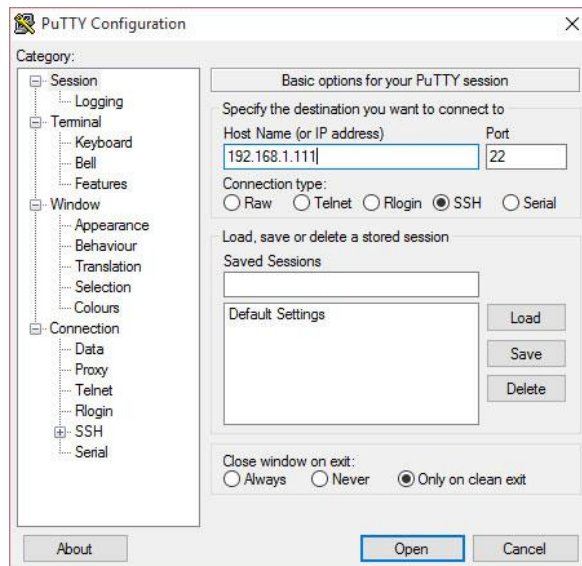
- Connecter écran + clavier + souris

## /// Déport d'écran

- Avec VNC (<https://raspberry-pi.fr/vnc-raspberry-pi/>)

## /// En mode remote

- Avec mobaXterm ou putty



# PICO WH

## /// Description

- Microcontrôleur : Basé sur la puce RP2040 double cœur ARM Cortex-M0+ cadencé à 133 MHz
- Mémoire : 264 Ko de mémoire RAM et 2 Mo de mémoire flash intégrée
- Connectivité : Intégration du Wi-Fi 802.11n
- Entrées et sorties : 26 broches GPIO multifonction dont certaines peuvent être utilisées pour I2C, SPI, UART, PWM, et ADC
- Modules supplémentaires : Utilisation possible avec des modules et capteurs externes
- Ports et connecteurs : Micro-USB pour alimentation et transfert de données, connecteur de débogage SWD pour la programmation
- Dimension : 51 mm x 21 mm
- Pas cher (< 15 €)

## /// Type de projet

- Projets IoT grâce à sa connectivité Wi-Fi intégrée

## /// Utilisation

- Apprendre à programmer en MicroPython avec la carte Raspberry Pi Pico W = site qui explique tout de A à Z !
  - [Raspberry Pi Pico W](#)
  - <https://raspberrypi.developpez.com/cours-tutoriels/raspberry-pi-pico-w/micropython/debuter/>

# • OÙ TROUVER DES INFOS

/// Le site officiel <https://www.raspberrypi.org/>



[Blog](#) [Downloads](#) [Community](#) [Help](#) [Forums](#) [Education](#)

/// Presque le même en français <http://www.raspberrypi-france.fr/>



/// Le Magazine gratuit <https://www.raspberrypi.org/magpi/>



/// Et des milliers d'autres sites

<https://raspi.tv/category/raspberry-pi>

<https://www.framboise314.fr/>

...

# MATÉRIEL: MODULE CAMÉRA UVS 5MP OV5648 AUTOFOCUS USB

## /// site internet

- <https://www.arducam.com/product/arducam-ov5648-auto-focus-usb-camera-ub0238-6/>

## /// Identification

- Arducam 5MP Auto Focus Mini USB Camera Board for Computer, 1/4" CMOS OV5648 UVC USB2.0 Video Webcam

## /// Item description

- 5MP, UVC compliant, USB 2.0 camera, based on the 1/4" Sony OV5648 image sensor

## /// Feature

- 5 Megapixel USB camera High Definition 2592\*1944 resolution webcam with autofocus lens is able to ensure your videos are always sharp and crystal clear
- 1/4" CMOS OV5640 Image sensor Webcams. Picture format MJPEG or YUY2 optional, cameras with high resolution can be used in any video system for personal or industrial
- Super small and Mini Size for your embedded applications. Compact camera for all kinds of machine vision
- Just plug the USB cable into your PC's USB port, plug and play, there is no need to download or install complicated driving software
- USB webcam widely works with WinXP/Vista/Windows 7/8, Windows 10 Linux with UVC. Webcam for MAC, PC, desktop computer, laptop USB





# MATÉRIEL: MODULE CAMÉRA UVS 5MP OV5648 AUTOFOCUS USB

## /// Specifications

### ■ Camera

- Sensor: 1/4" OV5648
- Resolution: 2592(H)x1944(V)
- Frame Rate: MJPG 15fps@2592 x 1944/1600 x 1200; 30fps@1080P/1280 x 960/1280 x 720/1024 x 768/800 x 600/640 x 480; YUY2 2fps@2592 x 1944; 3fps@2048 x 1536; 5fps@1920 x 1080/1600 x 1200/1280 x 960; 10fps@1280 x 720/1024 x 768; 15fps@800 x 600; 20fps@640 x 480.

### / Lens

- Field of View (FOV): 70°(D)
- Lens Mount: M6.5
- IR Sensitivity: Integral 650 IR filter, only visible light

## /// Functionality and Compatibility

- Adjustable parameter: Saturation, Contrast, Acutance, White balance, Exposure.
- System Compatibility: Windows XP (SP2, SP3)/Vista/7/8/10, Linux, and Mac with UVC

# MATÉRIEL: MODULE CAMÉRA UVS 5MP OV5648 AUTOFOCUS USB

## /// Power

- Power supply: USB powered 5V
- Working Current: MAX 250mA

## /// Physical

- Operating Temp: -4°F~158°F (-20°C~+70°C).
- Dimension : 62mmx9mmx5mm
- Cable Length: 3.3ft (1M)

# UTILISATION RAPIDE AVEC RASPBERRY PI + FSWEBCAM

/// Site: <https://raspberrypi-guide.github.io/electronics/using-usb-webcams>

## /// Installation

- `sudo apt install fswebcam`

## /// Utilisation

- `fswebcam -r 1280x720 --no-banner /images/image1.jpg`

# UTILISATION RAPIDE AVEC RASPBERRY PI + MJPG\_STREAMER

1. Connect USB camera to Raspberry
2. Check whether the camera is connected properly and recognized: `ls /dev`
3. If Video0 is listed with the command, it means that the camera is recognized  
If you cannot find the device Video0, please re-plug the camera and try again
4. Install necessary libraries  
`sudo apt-get install libjpeg8-dev`  
`sudo apt-get install libv4l-dev`
5. Download the software mjpeg-streamer (<https://krystof.io/mjpg-streamer-on-a-raspberry-pi-zero-w-with-a-usb-webcam-streaming-setup/>)
  1. `mkdir ~/mjpg-streamer`
  2. `cd ~/mjpg-streamer`
  3. `git clone https://github.com/jacksonliam/mjpg-streamer.git`
  4. `cd mjpg-streamer/mjpg-streamer-experimental`
  5. `make`
  6. `sudo make install`
6. Execute the command  
`./mjpg_streamer -i “./input_uvc.so -r 800x600” -o “output_http.so -w ./www”`
7. Open browser and enter the IP address of your Raspberry Pi (ex: 192.168.1.63:8080)  
You need to change “192.168.1.63” to the exact IP address of your Raspberry Pi  
Then you could see that the Stream will display the image captured by camera

# GESTION DES ANOMALIES

## /// Processus de gestion des Anomalies

- / Tout écart du produit par rapport aux spécifications doit être tracé
- / Tout écart du produit par rapport à une demande d'essai doit être tracé
- / Tout écart doit être identifié de manière unique
- / Une fiche standard doit être définie pour décrire le comportement anormal observé ainsi que les actions qui seront prises jusqu'à la clôture de l'anomalie
- / Un processus spécifique doit être mis en place pour gérer les anomalies avec un automate spécifique

# PROCESSUS DE VÉRIFICATION & VALIDATION

/// Processus de V&V doit être appliqué à chaque étape du développement

## /// Objectifs

- Montrer que le système n'a pas de défauts
- S'assurer que le système construit est utile et utilisable en pratique
- S'assurer que le système répond à la demande du client et à l'utilisation qu'il veut en faire

## /// Vérification

- Apporter la preuve que le produit remplit ses spécifications (exigences techniques)
- Spécifications sont précisées dans le document (réalisé par le fournisseur) spécifiant la solution
- Permet de répondre à la question: avons-nous bien construit le produit ?

## /// Validation

- Apporter la preuve que le produit satisfait aux besoins du client dans l'environnement opérationnel
- Besoins sont spécifiés comme des exigences incluses dans le cahier des charges fourni par le client
- Matrice de conformité (réalisée par le fournisseur) spécifie l'engagement du fournisseur vis-à-vis des exigences du client
- Permet de répondre à la question: avons-nous construit le bon produit ?



# PROCESSUS DE VÉRIFICATION & VALIDATION

## /// Intérêt pour le client

- S'assurer que le produit développé par le fournisseur est ce qu'il attendait
- Les travaux de vérification et de validation apportent la preuve que le produit final fonctionnera correctement

## /// Intérêt pour le fournisseur

- Vérifier qu'il fournit la bonne solution au client
- Les travaux de vérification et de validation apportent la preuve que le produit est conforme à sa spécification

## /// Intérêt Commun

- Matrice de conformité établit formellement l'accord entre le client et le fournisseur sur les résultats des travaux de vérification et de validation

## /// Y-a-t-il une chronologie à respecter ?

- Vérification est réalisée avant la Validation
- Tout changement peut induire la nécessité d'effectuer de nouvelles vérifications et validations

# POURQUOI SÉPARER LES RÔLES ?

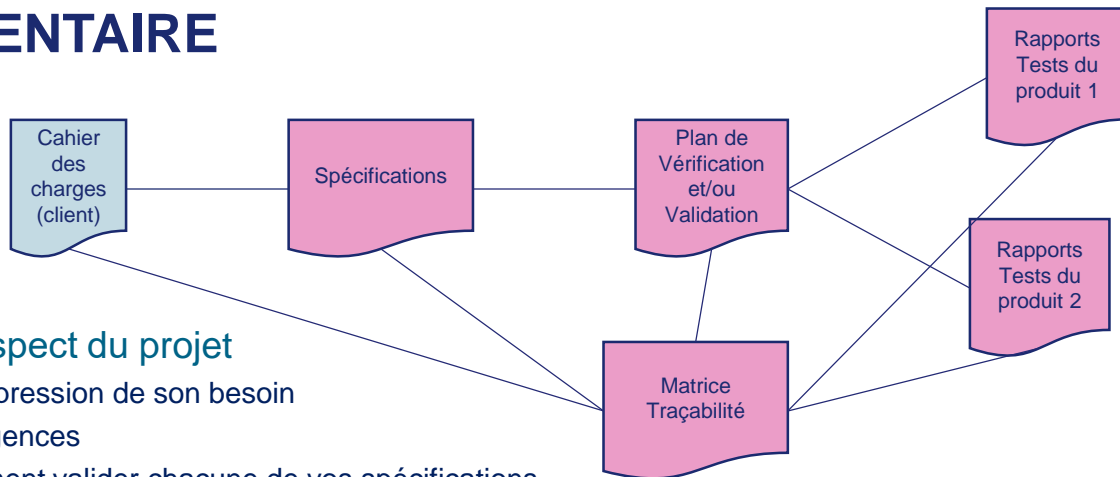
## /// 3 Rôles

- / “concepteur” = personne qui conçoit les spécifications du produit, réalise les plans, définit les fonctionnalités, rédige les procédures de tests
- / “développeur” = personne qui construit le produit en suivant les spécifications
- / “testeur” = personne qui déroule les procédures de tests pour s’assurer que le produit fonctionne tel qu’il a été spécifié

## /// Répartition des rôles dans l’équipe du projet

- / Risque si 1 personne tient les 3 rôles
  - La personne connaît parfaitement le produit. Si elle le teste, elle pourrait, même inconsciemment, “adapter” la procédure pour faire en sorte que le test soit correct. Ou, elle aura plus de mal à le mettre en défaut. Or c’est l’objectif de la campagne de tests.
- / Idéalement : 1 personne pour chaque rôle
  - Une personne ne connaissant pas le produit suivra la procédure à la lettre et sera extrêmement efficace pour dénicher tous les défauts de la procédure, comme du produit

# ORGANISATION DOCUMENTAIRE



## /// Chaque document traite d'un seul aspect du projet

- Cahier des charges fournit par le client = expression de son besoin
- Spécifications = votre solution avec vos exigences
- Plan de Vérification et/ou Validation = comment valider chacune de vos spécifications
- Rapports de tests = comment chacun de vos produits se comporte face à ces tests

## /// Avantages

- Documents sont plus courts
- Documents plus facilement maintenables
- Documents peuvent évoluer chacun indépendamment les uns des autres

## /// Matrice de traçabilité

- Permet de s'assurer qu'il y a au moins un test qui couvre chacune des spécifications
- Si ce n'est pas le cas, il faut écrire davantage de procédures pour tester toutes les spécifications