



Projet 4A : UniPi

Etat d'avancement

Présentation, le 26/01/2022

Présenté par :

AYOUB EL ASRI

LILIAN BESSONNEAU

Encadré par :

M. NICOLAS DELANOUE

01

...

Présentation

Les différents approches pour la réalisation du projet.

02

...

Configuration du Raspberry Pi

Système d'exploitation
Liaison avec **UniPi 1.1**

03

...

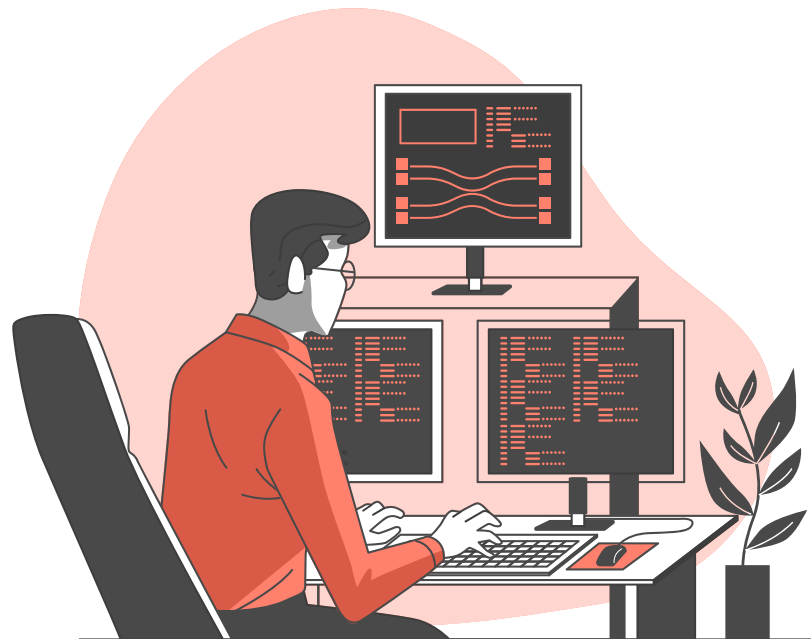
Programmation du Raspberry Pi

Mervis IDE
Langage FBD

04

...

Application & Perspectives





01

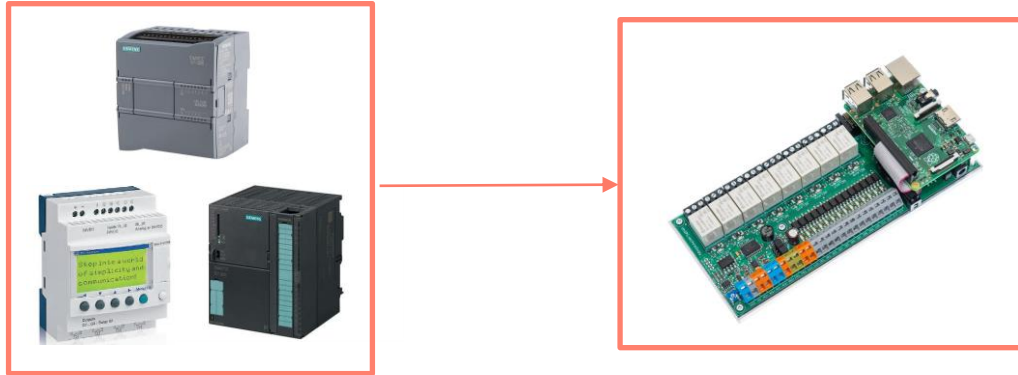
Présentation



Présentation

Objectif

Remplacer l'**API** par l'ensemble d'un **Raspberry Pi** et **UniPi**.



Pourquoi ?

- Prix
- Performances

Présentation

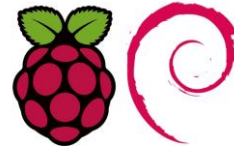
Méthodes possibles

- Fonctionnement en utilisant un OS compatible avec la carte UniPi pour le Raspberry Pi.

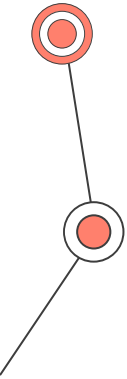
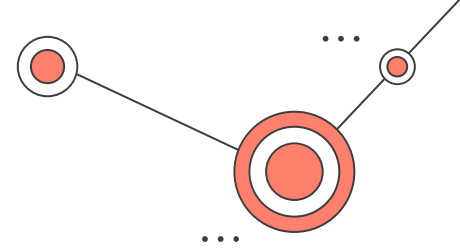
mervis

Neuron Mervis OS

- Fonctionnement en utilisant l'OS standard du Raspberry Pi, le Raspbian.

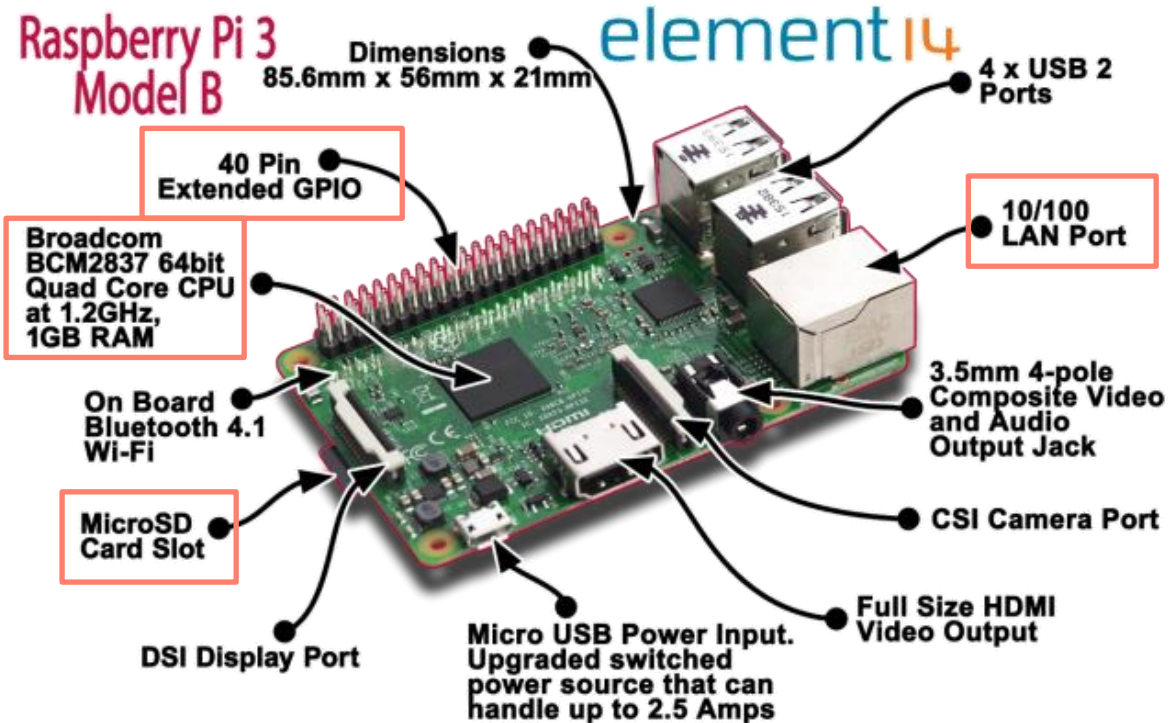


Raspbian



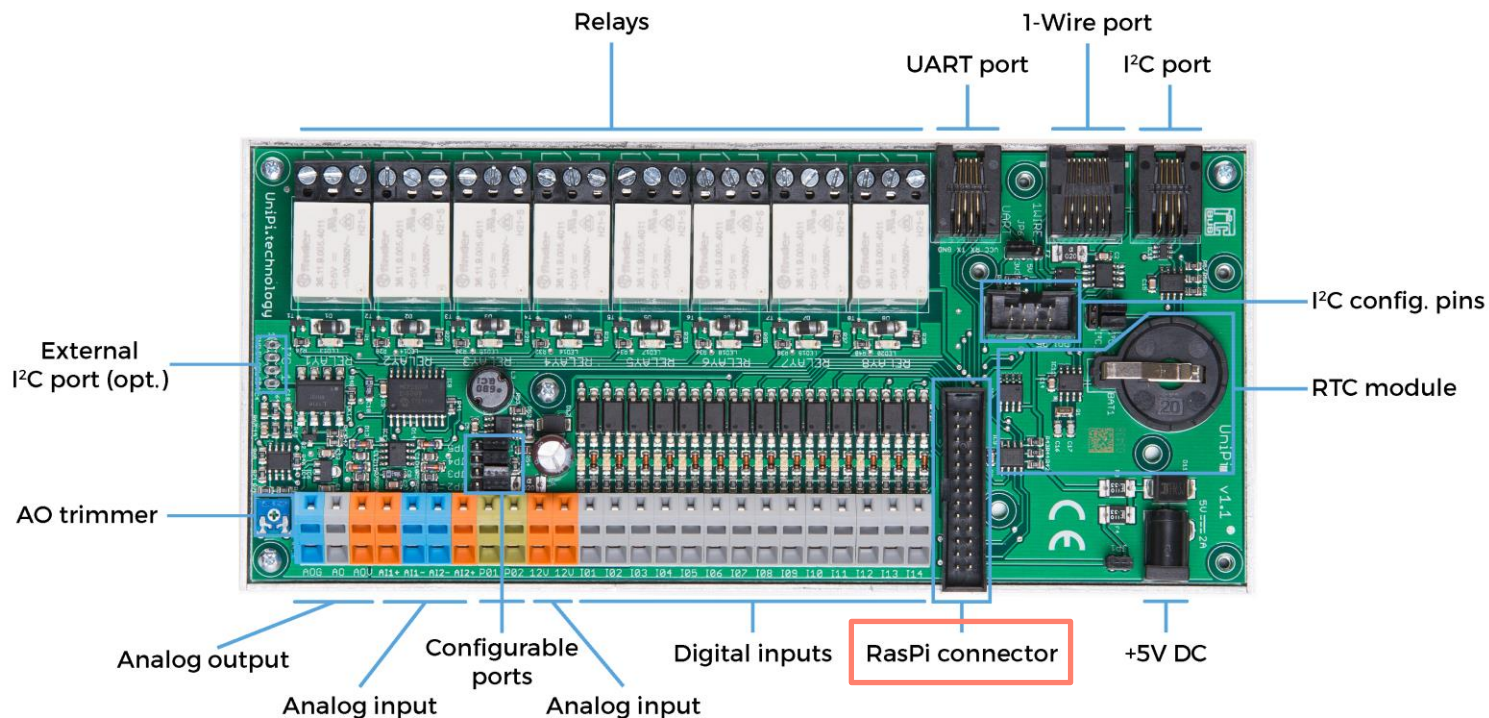
Présentation

Caractéristiques du Raspberry Pi 3 Model B



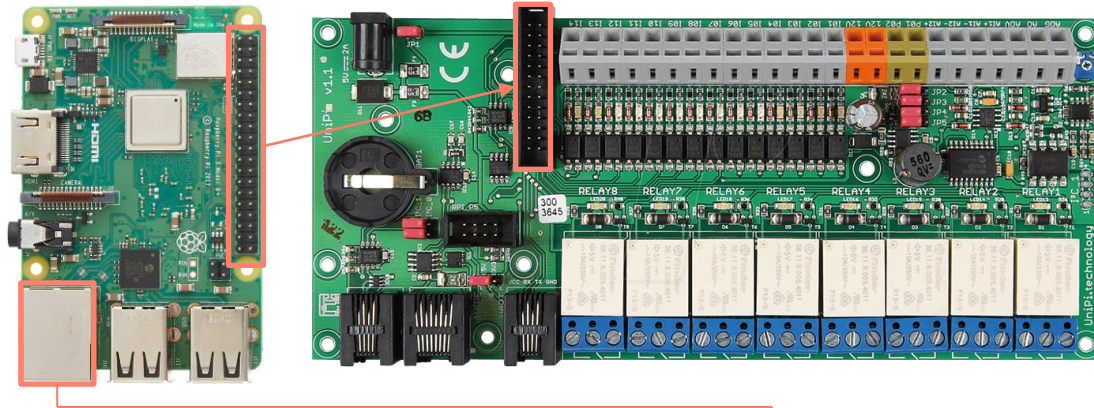
Présentation

Caractéristiques de l'UniPi 1.1




Présentation

Schéma du montage




mervis IDE

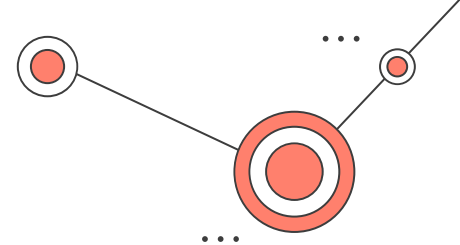


02

Configuration du Raspberry Pi



Configuration du Raspberry Pi



Etape 1 :

- Installation de l'OS sur le Raspberry Pi

mervis
Neuron Mervis OS

Mervis OS

mervis

Up-to-date Mervis OS image for Neuron units:

[neuron-mervis-os_image-2.4.1.0.zip](#)

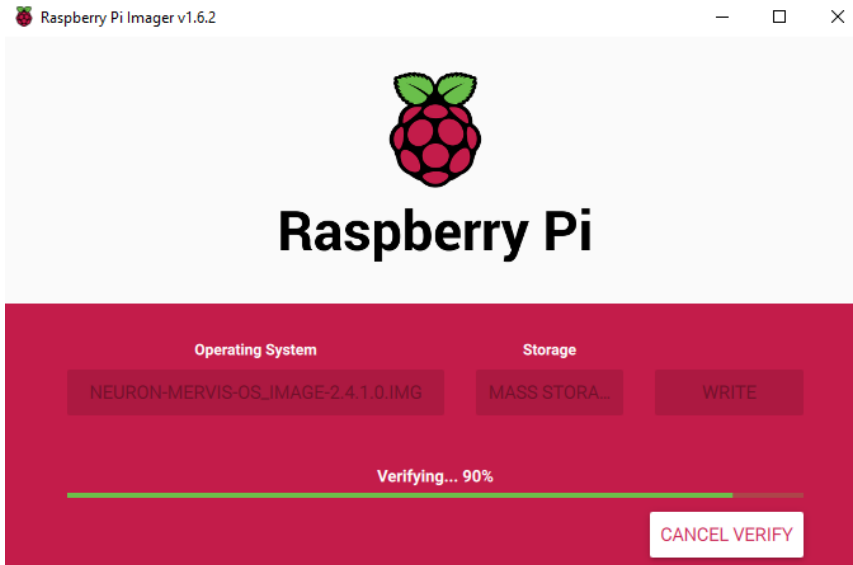
To install the Mervis OS into these units, continue with tutorial: [Mervis on Unipi Neuron](#).

Contains Mervis tools and default ModbusTCP & SysFS drivers.



To install this OS, it is necessary to have an SD card with a capacity of at least 2GB in the device.

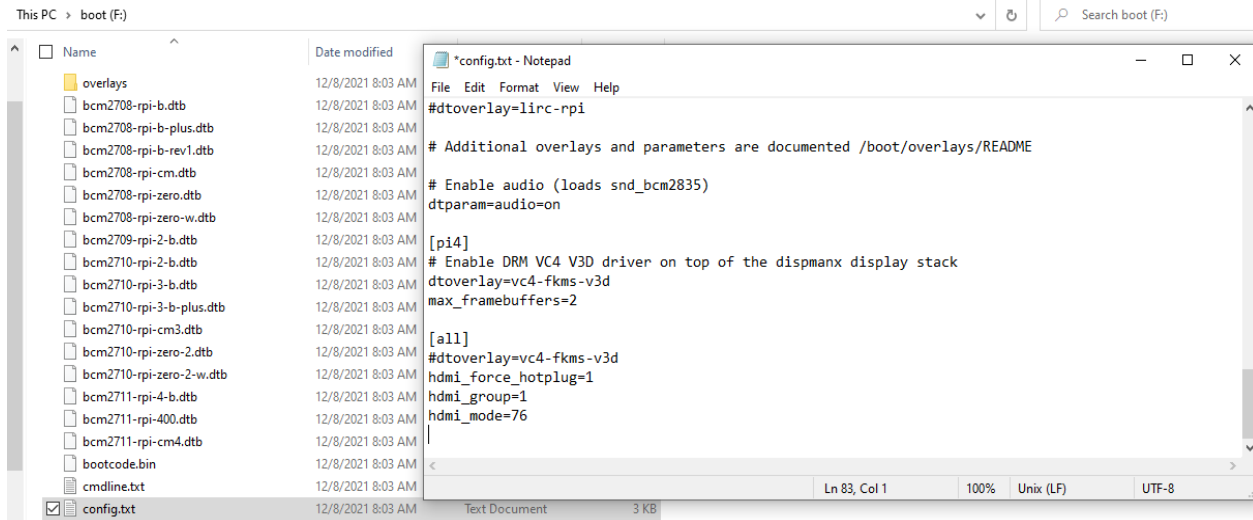
As of image version Mervis OS v2.4.0, Neuron units with **Raspberry Pi 4** are also supported.



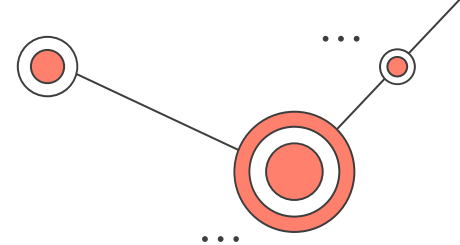
Configuration du Raspberry Pi

Etape 2:

- Avant d'éjecter la carte SD, il faudra ajouter dans config.txt, hdmi_force_hotplug, hdmi-group et hdmi_mode.



Configuration du Raspberry Pi

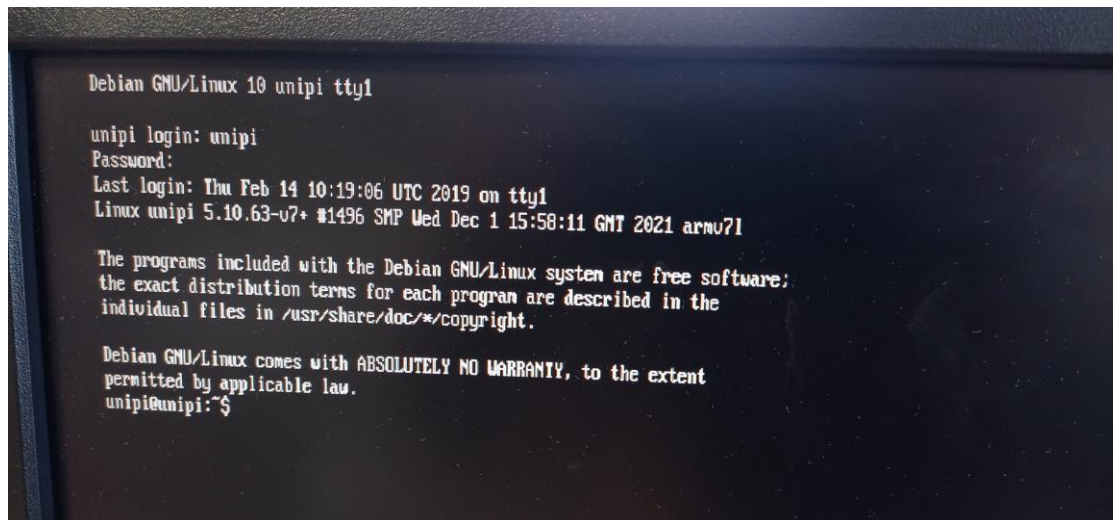


Etape 3:

- Allumons le Raspi, on peut s'authentifier en utilisant :

Login : unipi

Password : unipi.technology

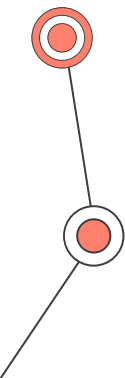


```
Debian GNU/Linux 10 unipi tty1

unipi login: unipi
Password:
Last login: Thu Feb 14 10:19:06 UTC 2019 on tty1
Linux unipi 5.10.63-v7+ #1496 SMP Wed Dec 1 15:58:11 GMT 2021 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
unipi@unipi:~$
```



Configuration du Raspberry Pi

Etape 4:

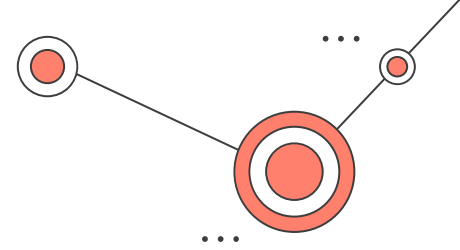
- On fait une liaison entre le Raspberry Pi et l'ordinateur avec le câble Ethernet, et on récupère l'adresse IP.

```
unipi@unipi:~$ /sbin/ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> ntu 1500
      inet 169.254.191.134 netmask 255.255.0.0 broadcast 169.254.255.255
      inet6 fe80::3115:721b:9234:f518 prefixlen 64 scopeid 0x20<link>
      ether b8:27:eb:6b:6b:01 txqueuelen 1000 (Ethernet)
      RX packets 138 bytes 15308 (14.9 KiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 29 bytes 4859 (4.7 KiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> ntu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
      loop txqueuelen 1000 (Local Loopback)
      RX packets 775 bytes 55071 (53.7 KiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 775 bytes 55071 (53.7 KiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

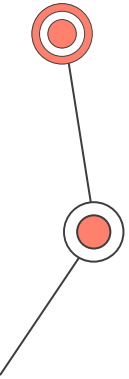
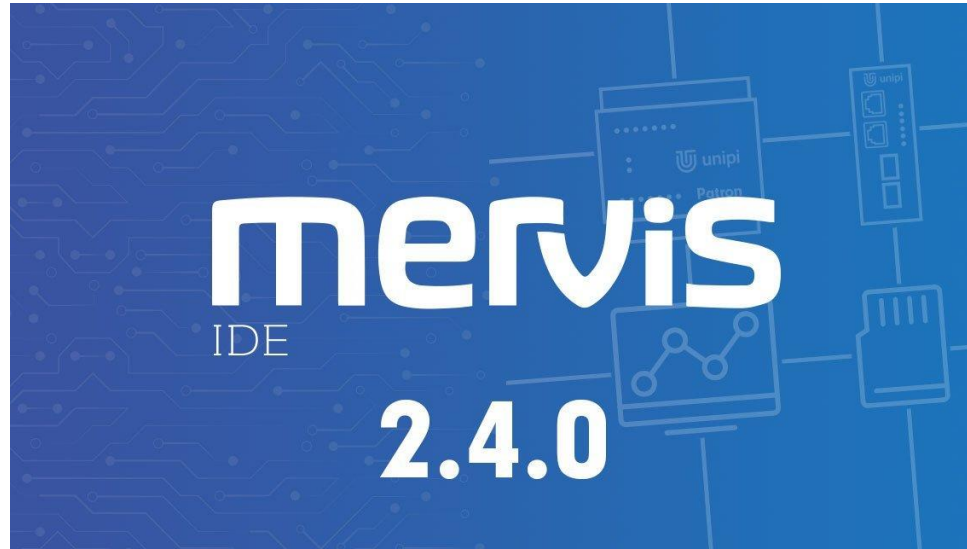
unipi@unipi:~$
```

Configuration du Raspberry Pi



Etape 5:

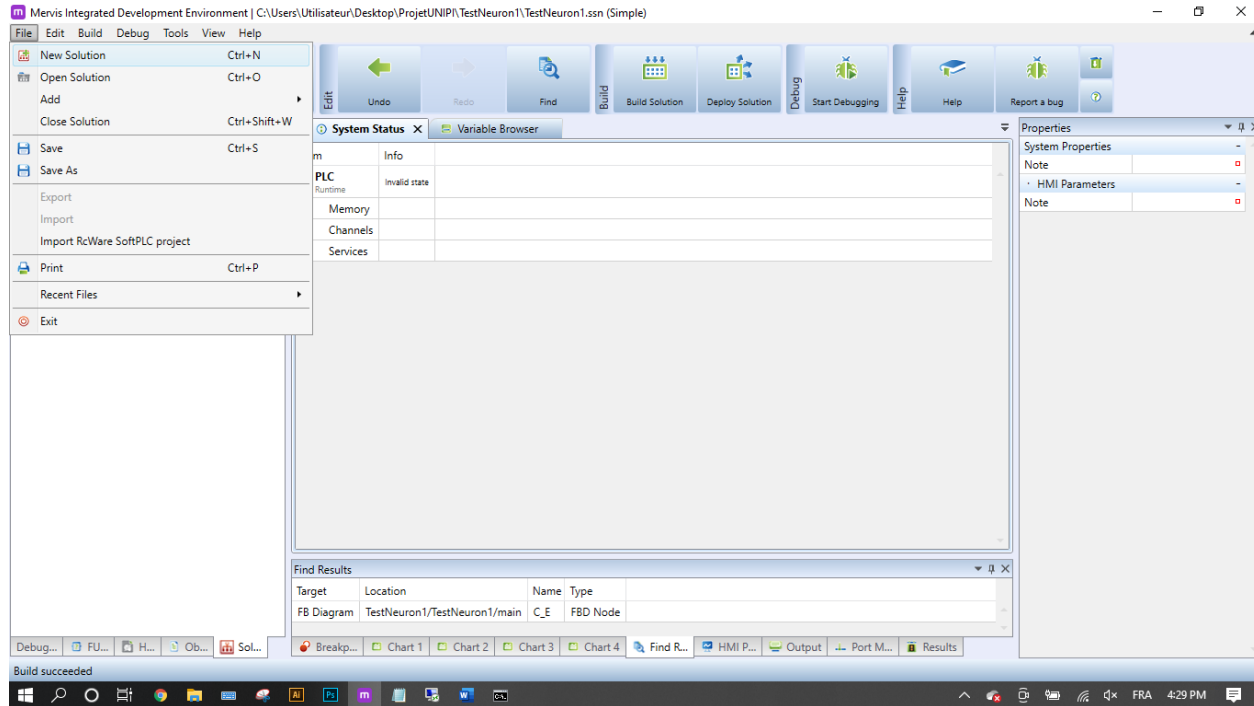
- Au niveau de l'ordinateur on va installer Mevris IDE.



Configuration du Raspberry Pi

Etape 6:

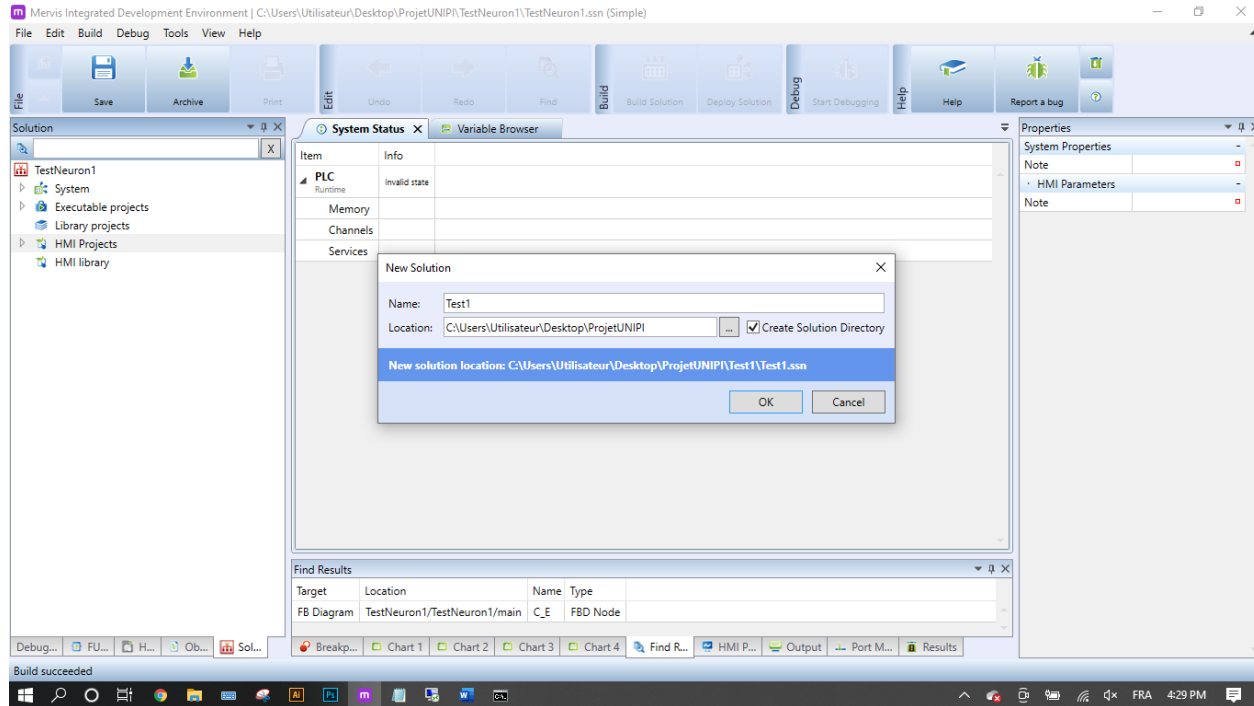
- On commence par créer une nouvelle solution



Configuration du Raspberry Pi

Etape 6:

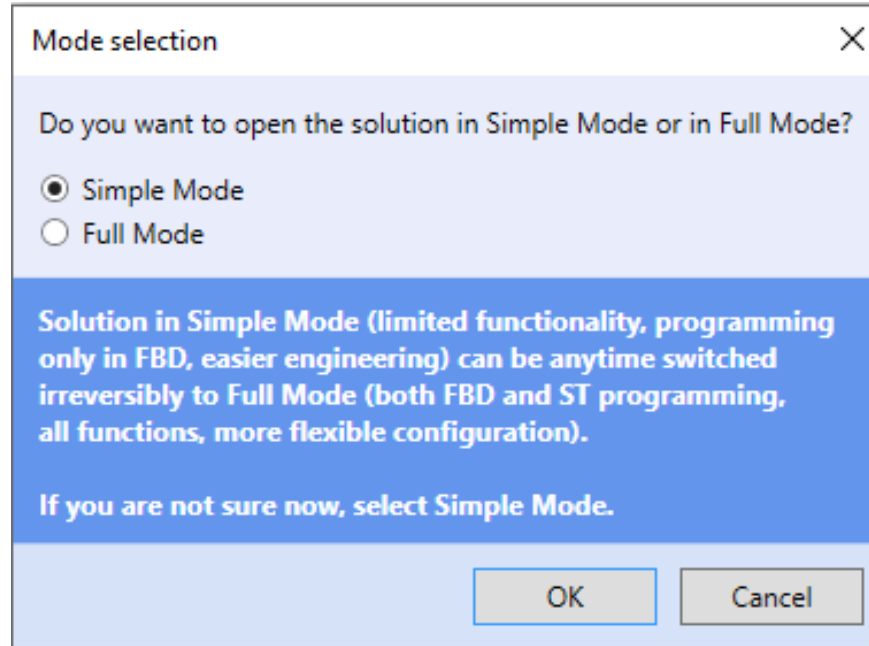
- On commence par créer une nouvelle solution



Configuration du Raspberry Pi

Etape 6:

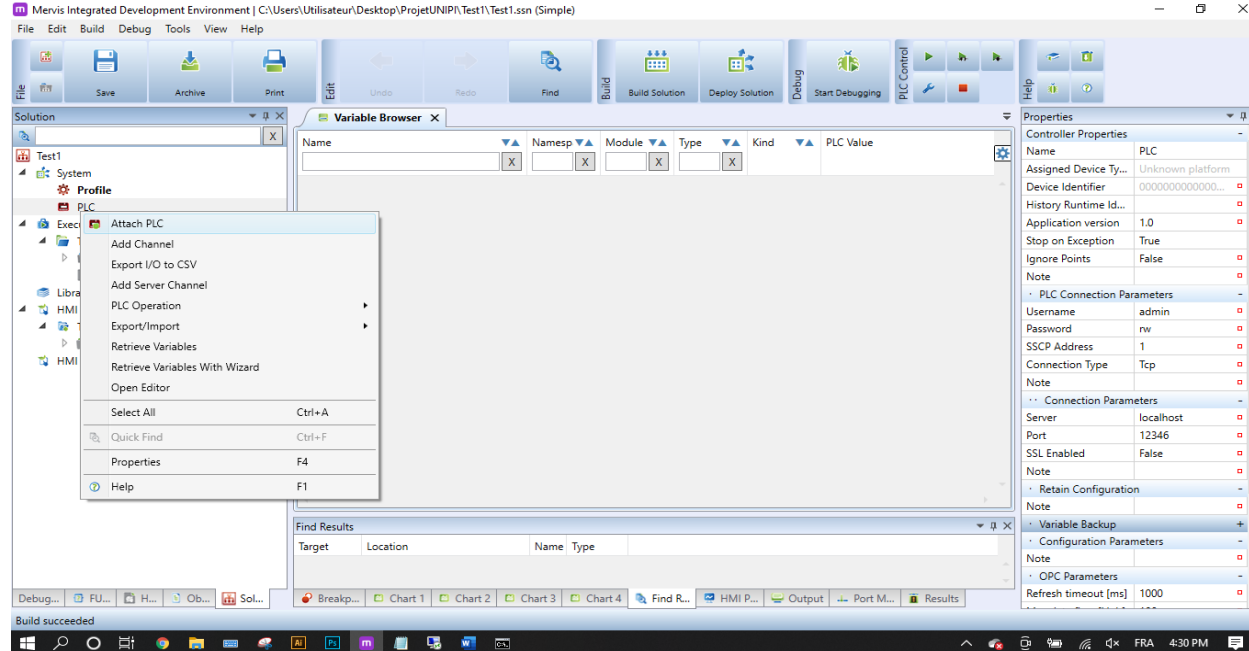
- On commence par créer une nouvelle solution



Configuration du Raspberry Pi

Etape 7:

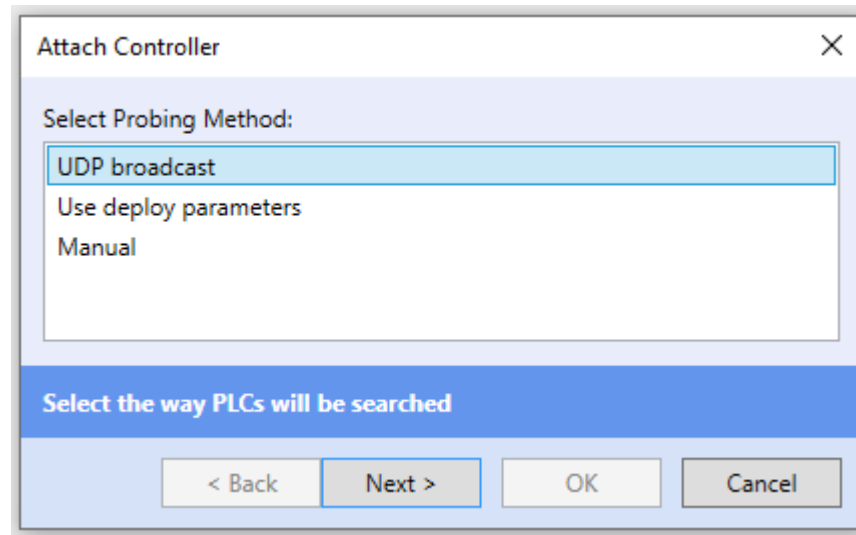
- Après la création, il faudra que l'IDE détecte le Raspberry Pi qui joue le rôle d'un PLC.



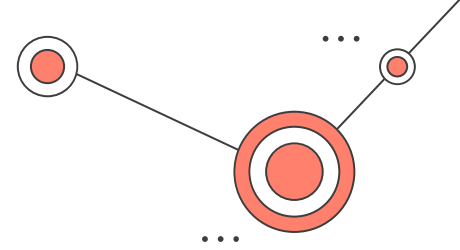
Configuration du Raspberry Pi

Etape 7:

- Après la création, il faudra que l'IDE détecte le Raspberry Pi qui joue le rôle d'un PLC.



Configuration du Raspberry Pi



Etape 7:

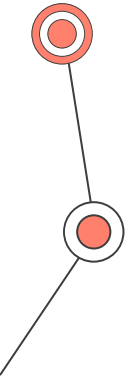
- Après la création, il faudra que l'IDE détecte le Raspberry Pi qui joue le rôle d'un PLC.

Attach Controller

Username: admin

Password: ••

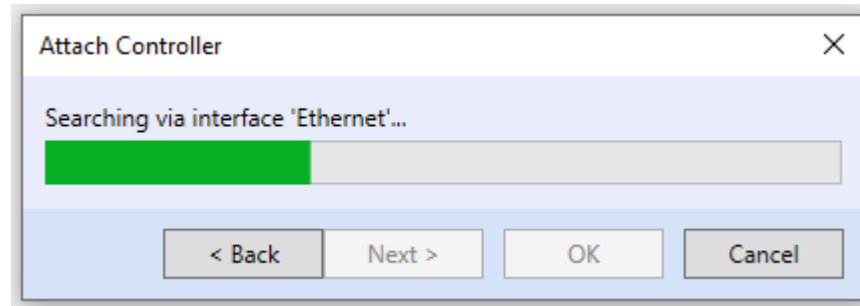
< Back Next > OK Cancel



Configuration du Raspberry Pi

Etape 7:

- Après la création, il faudra que l'IDE détecte le Raspberry Pi qui joue le rôle d'un PLC.



Configuration du Raspberry Pi

Etape 7:

- Après la création, il faudra que l'IDE détecte le Raspberry Pi qui joue le rôle d'un PLC.

Attach Controller ×

Found PLCs:

Address	Slave	Serial Number	Name	License	Runtime Version	Platform
169.254.191.134	1	0000B827EB686B01	PLC	Not licensed	2.4.2611.125194	Unipi 1.1 / RPi 2 rev. 1.0 System version: 2.4.1.0f

Attach Controller ×

☒ Download Configuration

Select if user can download configuration from selected PLC

< Back Next > OK Cancel

< Back Next > OK Cancel

- Notre Raspberry Pi est bien détecté



03

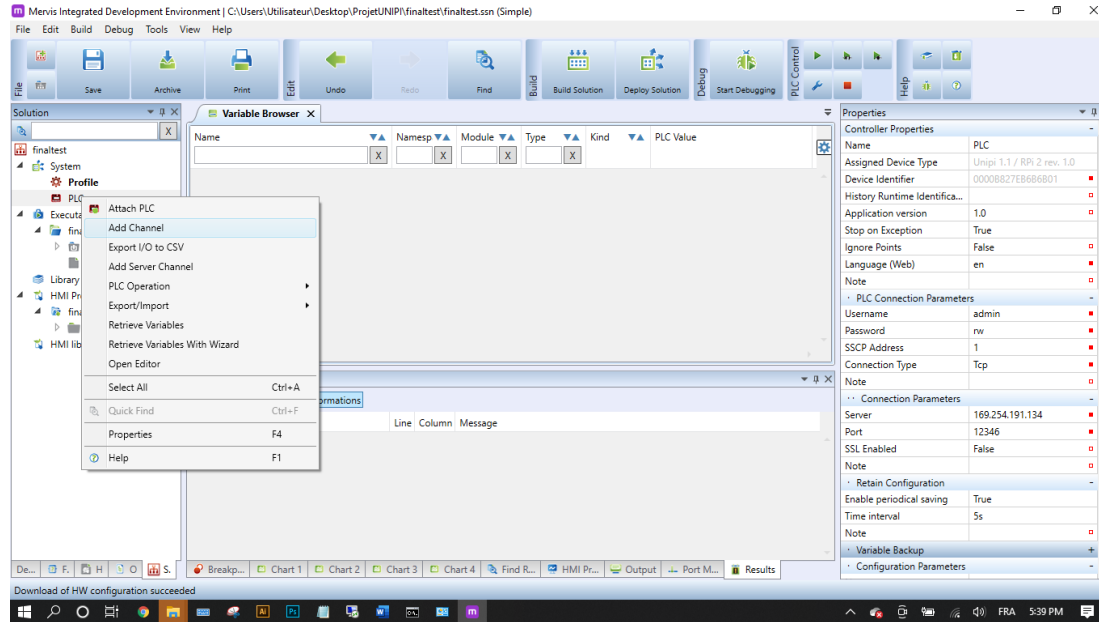
Programmation du Raspberry Pi



Programmation du Raspberry Pi

Etape 1:

- Ajoutons un chaine pour la configuration du Modbus TCP



Programmation du Raspberry Pi

Etape 1:

- Ajoutons un chaine pour la configuration du Modbus TCP

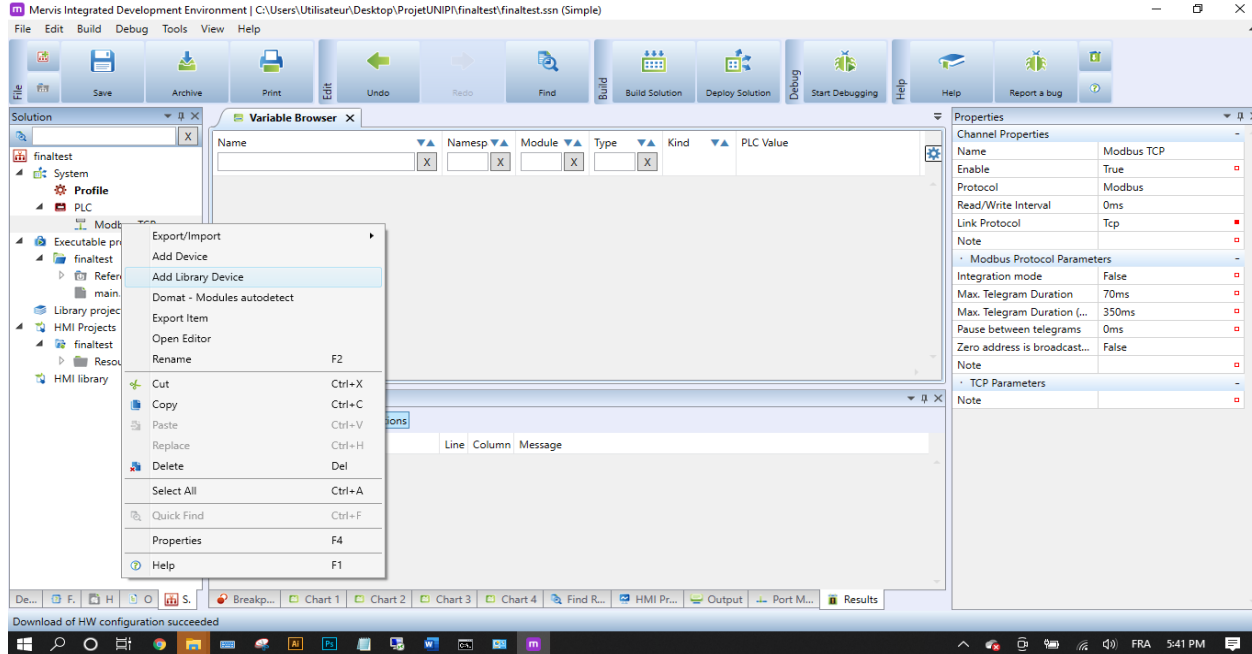
The screenshot displays the Siemens SIMATIC Manager interface. On the left, the 'Solution' tree shows a project named 'finaltest' with a 'Modbus TCP' channel highlighted under the 'PLC' section. The central pane shows the 'Name' field for the channel. On the right, the 'Properties' window is open, showing the 'Channel Properties' for 'Modbus TCP'.

Channel Properties	
Name	Modbus TCP
Enable	True
Protocol	Modbus
Read/Write Interval	0ms
Link Protocol	Tcp
Note	
Modbus Protocol Parameters	
Integration mode	False
Max. Telegram Duration	70ms
Max. Telegram Duration (...)	350ms
Pause between telegrams	0ms
Zero address is broadcast...	False
Note	
TCP Parameters	
Note	

Programmation du Raspberry Pi

Etape 2:

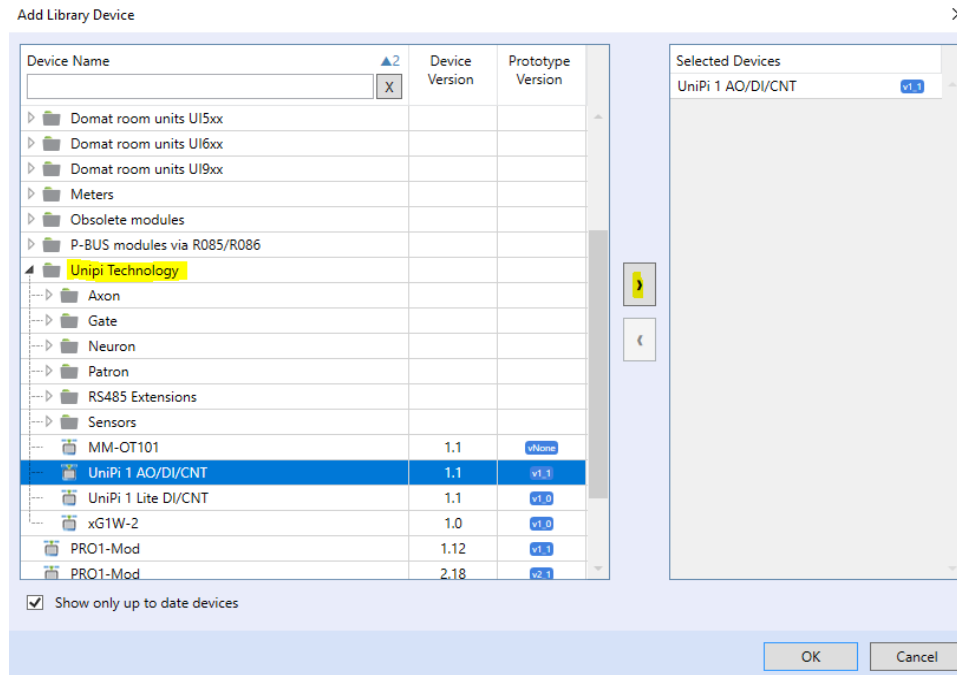
- Au niveau du chaine de Modbus TCP, on ajoute la bibliothèque des composants correspondants de l'UniPi.



Programmation du Raspberry Pi

Etape 2:

- Au niveau du chaine de Modbus TCP, on ajoute la bibliothèque des composants correspondants de l'UniPi.



Programmation du Raspberry Pi

Etape 3:

- On crée une autre chaine I2C.

Mervis Integrated Development Environment | C:\Users\Utilisateur\Desktop\ProjetUNIPi\finaltest\finaltest.ssn (Simple)

File Edit Build Debug Tools View Help

Archive (Ctrl+Shift+U) Print Edit Undo Redo Find Build Build Solution Deploy Solution Debug Start Debugging Help Report a bug

Solution

finaltest

- System
 - Profile
 - PLC
 - I2C
 - Modbus TCP
 - UniPi1-AO-DI-CNT
- Executable projects
 - finaltest
 - References
 - Globals
 - main.Program.fbd
 - Library projects

Variable Browser

Name	Namesp	Module	Type	Kind	PLC Value
UniPi1-AO-DI-CNT_UnipiSetCnt_12_comm	hw	finaltest	bool	Global	no value
UniPi1-AO-DI-CNT_UnipiSetCnt_12_timest	hw	finaltest	dt	Global	no value
UniPi1-AO-DI-CNT_UnipiSetCnt_12_comm	hw	finaltest	bool	Global	no value
UniPi1-AO-DI-CNT_UnipiSetCnt_11_comm	hw	finaltest	bool	Global	no value
UniPi1-AO-DI-CNT_UnipiSetCnt_11_timest	hw	finaltest	dt	Global	no value
UniPi1-AO-DI-CNT_UnipiSetCnt_11_comm	hw	finaltest	bool	Global	no value
UniPi1-AO-DI-CNT_UnipiSetCnt_10_comm	hw	finaltest	bool	Global	no value
UniPi1-AO-DI-CNT_UnipiSetCnt_10_timest	hw	finaltest	dt	Global	no value
UniPi1-AO-DI-CNT_UnipiSetCnt_10_comm	hw	finaltest	bool	Global	no value
UniPi1-AO-DI-CNT_UnipiSetCnt_09_comm	hw	finaltest	bool	Global	no value

Properties

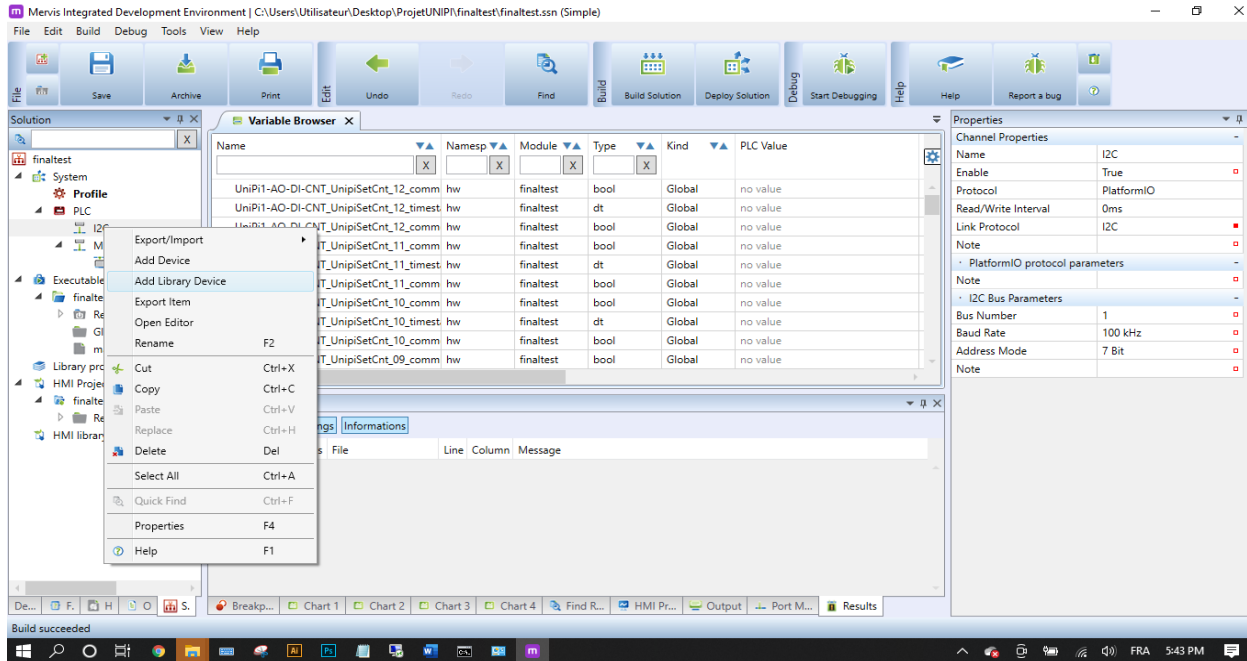
Channel Properties

Name	I2C
Enable	True
Protocol	PlatformIO
Read/Write Interval	0ms
Link Protocol	I2C
Note	
PlatformIO protocol parameters	
Note	
I2C Bus Parameters	
Bus Number	1
Baud Rate	100 kHz
Address Mode	7 Bit
Note	

Programmation du Raspberry Pi

Etape 4:

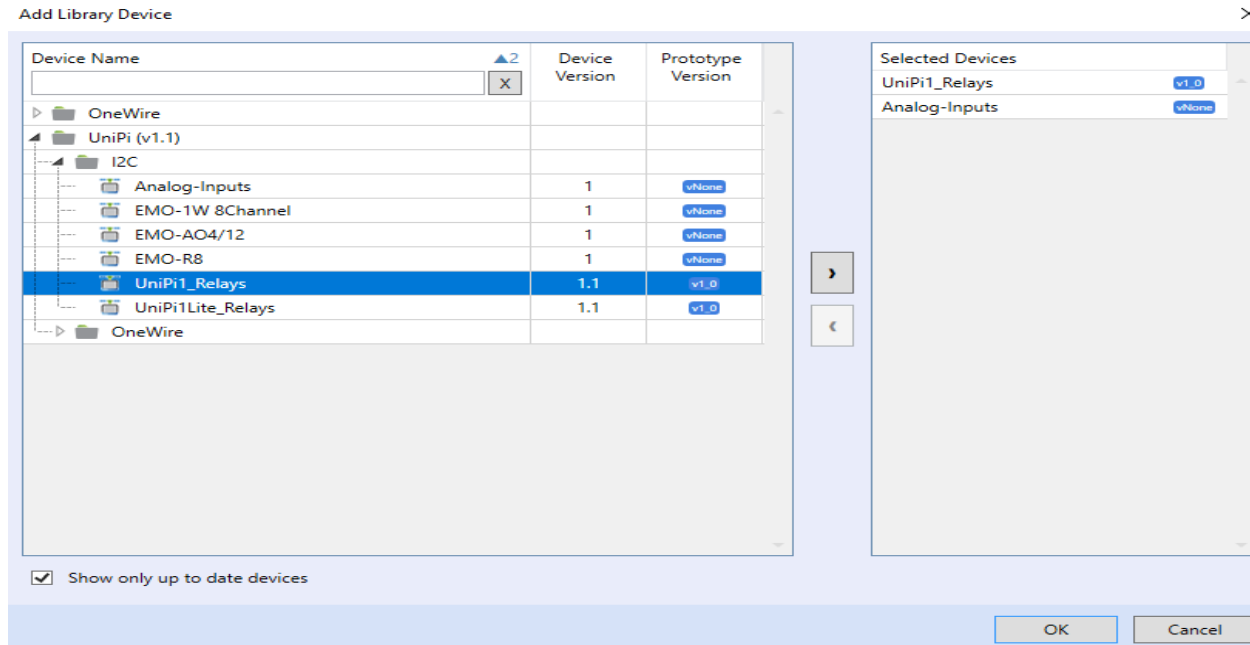
- Au niveau du chaine de I2C, on ajoute la bibliothèque des composants correspondants de l'UniPi.

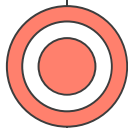


Programmation du Raspberry Pi

Etape 4:

- Au niveau du chaine de I2C, on ajoute la bibliothèque des composants correspondants de l'UniPi.





Programmation du Raspberry Pi

Etape 5:

- On bascule vers le main pour créer notre premier programme.

The screenshot displays the Mervis IDE interface. The title bar reads 'Mervis Integrated Development Environment | C:\Users\Utilisateur\Desktop\ProjetUNIP\finaltest\finaltest.ssn (Simple)'. The menu bar includes File, Edit, Build, Debug, Tools, View, and Help. The toolbar contains icons for File, Save, Archive, Print, Edit, Undo, Redo, Find, Build, Build Solution, Deploy Solution, Debug, Start Debugging, Fit to Width, Help, and Report a bug. The Solution Explorer on the left shows a project tree with 'finaltest' as the root, containing 'System', 'Profile', 'PLC', 'I2C', 'UniAI.Mcp-Analog', 'UniPi1_Relays.Uni...', 'Modbus TCP', 'UniPi1-AO-DI-CN...', 'Executable projects', 'finaltest' (with sub-items 'References', 'Globals', and 'main.Program.fbd'), 'Library projects', and 'HMI Projects'. The main workspace shows a ladder logic diagram with a 'Button' input connected to two outputs: 'UniPi1_Relays_Rela y7' and 'UniPi1_Relays_Rela y1'. The Properties window on the right displays 'Code Unit Properties' for the 'main' program, with details such as 'Code Kind: Program', 'Implementation Language: fbd', 'Namespace: main', 'Resulting Declaration: main', 'Document File Name: main_4c592603-87b2-4b4...', 'Include in Build: True', and 'Status: Development'.

Code Unit Properties	
Name	main
Code Kind	Program
Implementation Language	fbd
Namespace	
Resulting Declaration	main
Document File Name	main_4c592603-87b2-4b4...
Include in Build	True
Note	
-	
Abbreviation	
Image	
Help Text	
Help Path	
Show in Explorer	False
Display Name	
Category Path	
Status	Development

Programmation du Raspberry Pi

Etape 5:

- On bascule vers le main pour créer notre premier programme.

- On fait le build de notre programme, puis on effectue le déploiement.


New state of PLC controller

Run PLC as after	Select run mode
<input checked="" type="radio"/> Warm restart	<input checked="" type="radio"/> Full run
<input type="radio"/> Cold restart	<input type="radio"/> Evaluation only
<input type="checkbox"/> Clear flash variables	<input type="radio"/> Communication only
<input type="checkbox"/> Clear OEM parameters	<input type="radio"/> Commissioning mode
Boot project location:	
<input type="radio"/> None <input type="radio"/> Memory Area 1 <input checked="" type="radio"/> Memory Area 1+2	
<input type="checkbox"/> Upload variable collection	
<div></div>	
<input type="checkbox"/> Remember the last used	
<div>OK</div>	



04

Application & Perspectives



Application & Perspectives

Perspectives:

- 1- L'Interface Homme Machine existe au niveau de l'IDE, comment peut-on utiliser ce mécanisme ?
- 2- Est-ce qu'on peut programmer en langage C ou Python sans passer par l'IDE ?
- 3- Comment peut-on court-circuiter l'OS d'UniPi et utiliser directement le Raspbian ?



Application & Perspectives

Références :

Informations générales et études de cas :

<https://www.unipi.technology/products/unipi-1-1-1-1-lite-19>

Datasheets, OS, liens de téléchargement :

<https://kb.unipi.technology/en:hw:03-unipi11>

Tutoriel d'UniPi :

<https://www.youtube.com/watch?v=mgUas3lxH2w>

**Merci pour votre
Attention**