

**GIPUZKOAKO INGENIARITZA ESKOLA**  
**ESCUELA DE INGENIERÍA DE GUIPUZKOA**



**GRADU AMAIERAKO LANA/ TRABAJO FIN DE GRADO**

---

**Denbora errealeko PLC-aren  
inplementazioa Linux-a duen ARM  
mikrokontrolagailuaren bidez**

---

**DOKUMENTUA: ERABILTZAILEAREN GIDA**

Gradua: Industria Elektronikaren eta Automatikaren Ingeniaritzako Gradua

Ikaslea: Jon Cortajarena Alcorta

Zuzendaria: Mikel Alberro Astarbe

Ikasturtea: 2017-2018

Defentsa eguna: 2018ko maiatzaren 17a

## **ERABILTZAILEAREN GIDAREN AURKIBIDEA**

|      |                               |    |
|------|-------------------------------|----|
| 1.   | SARRERA .....                 | 2  |
| 2.   | MANUAL DE USUARIO .....       | 3  |
| 2.1. | Configuración de la BBB ..... | 3  |
| 2.2. | EtherCAT .....                | 16 |
| 2.3. | PROFINET.....                 | 20 |
| 2.4. | CANopen.....                  | 27 |

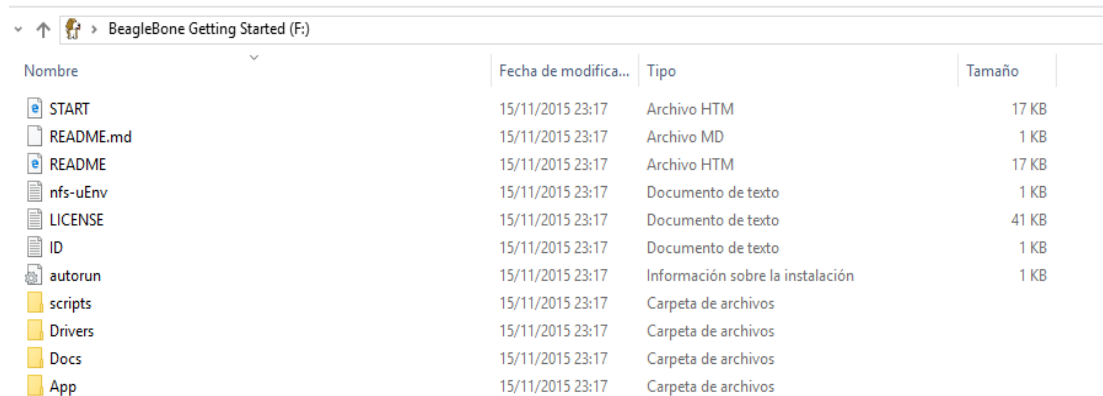
# **1. SARRERA**

Atal honetan proiektua amaitu ondoreneko erabiltzailearen gida azaltzen da. Enpresak prozesuaren azalpen sinple bat nahi zuen, orduan euskaraz egindako prozesua gaztelerara pasaz dokumentu labur bat entregatu da.

## 2. MANUAL DE USUARIO

### 2.1. Configuración de la BBB

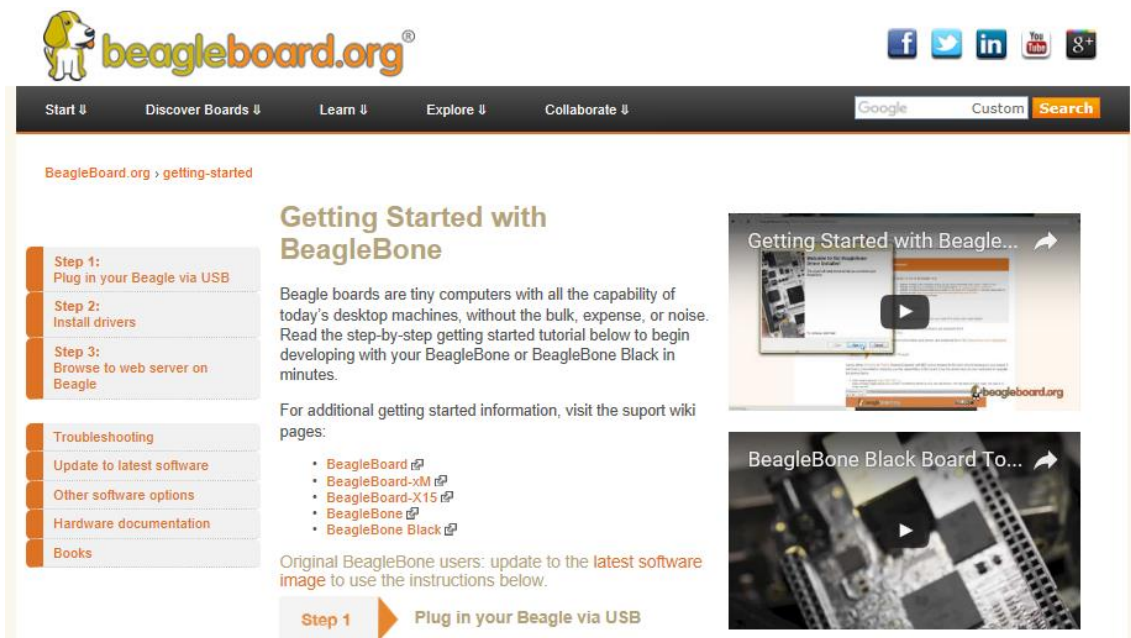
Para comenzar, la BBB se conectará vía USB al ordenador. Aparecerá la siguiente carpeta:



| Nombre    | Fecha de modifica... | Tipo                             | Tamaño |
|-----------|----------------------|----------------------------------|--------|
| START     | 15/11/2015 23:17     | Archivo HTM                      | 17 KB  |
| README.md | 15/11/2015 23:17     | Archivo MD                       | 1 KB   |
| README    | 15/11/2015 23:17     | Archivo HTM                      | 17 KB  |
| nfs-uEnv  | 15/11/2015 23:17     | Documento de texto               | 1 KB   |
| LICENSE   | 15/11/2015 23:17     | Documento de texto               | 41 KB  |
| ID        | 15/11/2015 23:17     | Documento de texto               | 1 KB   |
| autorun   | 15/11/2015 23:17     | Información sobre la instalación | 1 KB   |
| scripts   | 15/11/2015 23:17     | Carpeta de archivos              |        |
| Drivers   | 15/11/2015 23:17     | Carpeta de archivos              |        |
| Docs      | 15/11/2015 23:17     | Carpeta de archivos              |        |
| App       | 15/11/2015 23:17     | Carpeta de archivos              |        |

Figura 1: Fichero de inicio de la BBB.

En esa carpeta elegiremos “START” y nos mandara directo a la página de la BBB:



The screenshot shows the BeagleBoard.org website. At the top, there is a navigation menu with links for 'Start', 'Discover Boards', 'Learn', 'Explore', and 'Collaborate'. A search bar is also present. The main content area is titled 'Getting Started with BeagleBone' and includes a video player and a list of links for additional information. The video player shows a thumbnail of the 'Getting Started with Beagle...' video. Below the video player, there is a list of links for 'BeagleBoard', 'BeagleBoard-xM', 'BeagleBoard-X15', 'BeagleBone', and 'BeagleBone Black'. The page also features a sidebar with a list of links for 'Step 1: Plug in your Beagle via USB', 'Step 2: Install drivers', 'Step 3: Browse to web server on Beagle', 'Troubleshooting', 'Update to latest software', 'Other software options', 'Hardware documentation', and 'Books'.

Figura 2: Página web de la BeagleBoard.

Hay que seguir los 3 pasos que aparecen a la izquierda para poder instalar la BBB. En el primer paso, al reconocer la BBB se pondrá en verde. Entonces, podremos seguir con el segundo paso. En este caso instalaremos los drivers necesarios.

**Step 2** Install drivers

Install the drivers for your operating system to give you network-over-USB access to your Beagle. Additional drivers give you serial access to your board.

| Operating System | USB Drivers       | Comments   |
|------------------|-------------------|--|
| Windows (64-bit) | 64-bit installer  | If in doubt, try the 64-bit installer first. <ul style="list-style-type: none"><li>• <b>Note #1:</b> Windows Driver Certification warning may pop up two or three times. Click "Ignore", "Install" or "Run"</li><li>• <b>Note #2:</b> To check if you're running 32 or 64-bit Windows see this: <a href="https://support.microsoft.com/kb/827218">https://support.microsoft.com/kb/827218</a></li><li>• <b>Note #3:</b> On systems without the latest service release, you may get an error (0xc00007b). In that case, please install the following and retry: <a href="https://www.microsoft.com/en-us/download/confirmation.aspx?id=13523">https://www.microsoft.com/en-us/download/confirmation.aspx?id=13523</a></li><li>• <b>Note #4:</b> You may need to reboot Windows.</li><li>• <b>Note #5:</b> These drivers have been tested to work up to Windows 10</li></ul> |
| Windows (32-bit) | 32-bit installer  |  |
| Mac OS X         | Network<br>Serial | Install both sets of drivers.  |
| Linux            | mkudevrule.sh     | Driver installation isn't required, but you might find a few udev rules helpful.   |

Figura 3: 2. paso. Diferentes drivers.

Elegiremos el driver según el sistema operativo. En mi caso Windows (64-bit). Al instalarlos, aparecerán en verde los dos primeros pasos.

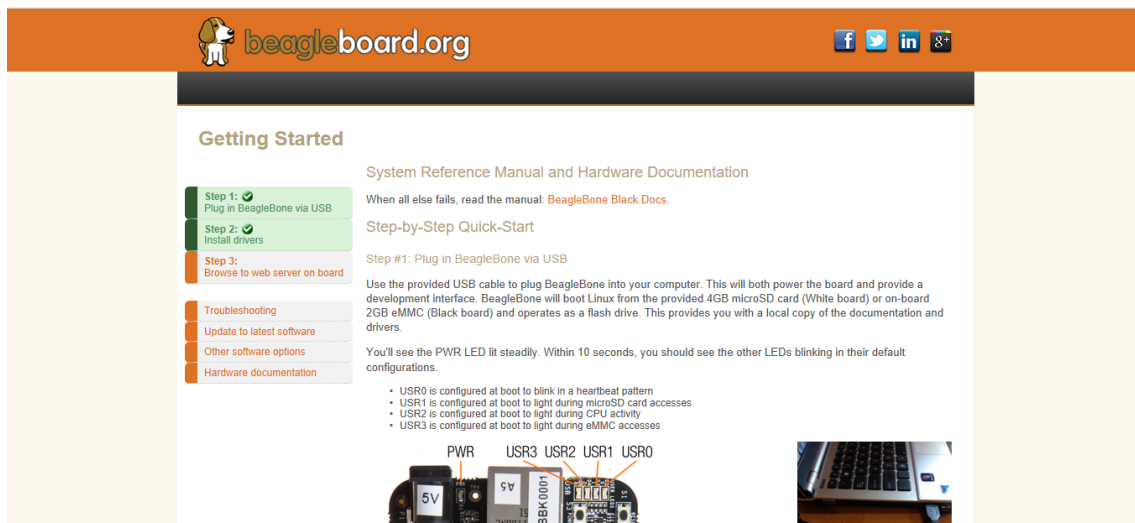


Figura 1: "Step 1" y "Step 2" finalizados.

Después de instalar el driver, el tercer paso sería conectarse al link <http://192.168.7.2> para ver si se ha conectado o no. Esa dirección sería la IP de la BBB vía USB. En caso de conectarse aparecerá el siguiente mensaje:

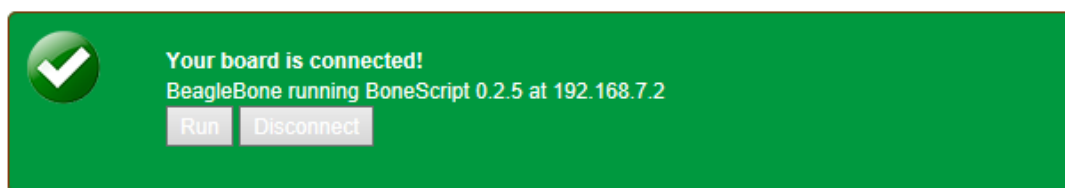


Figura 2: "Step 3".Mensaje de conexión de la BBB.

Denbora errealeko PLC-aren implementazioa Linux-a duen ARM mikrokontrolagailuaren bidez

Una vez que la BBB esté conectada, hay que meterse en el link <https://beagleboard.org/latest-images>, donde se encuentran las imágenes de Linux Debian y elegiremos la más nueva. El Debian 8.7 es el más nuevo pero hemos visto que como root (administrador) no es posible entrar. Entonces, elegiremos la Debian 8.6. Después de descargar la imagen, hay que descomprimirlo vía 7zip:

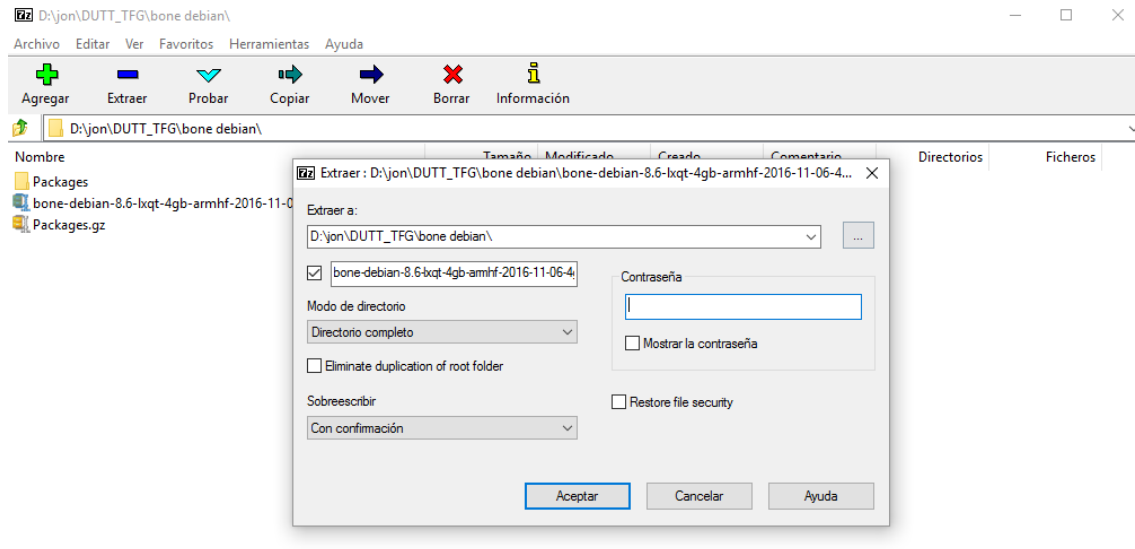


Figura 3: Descomprimiendo la imagen Debian vía 7zip.

Después de descomprimirlo, se consigue el .img y se introducirá en la tarjeta SD con el programa Win32:

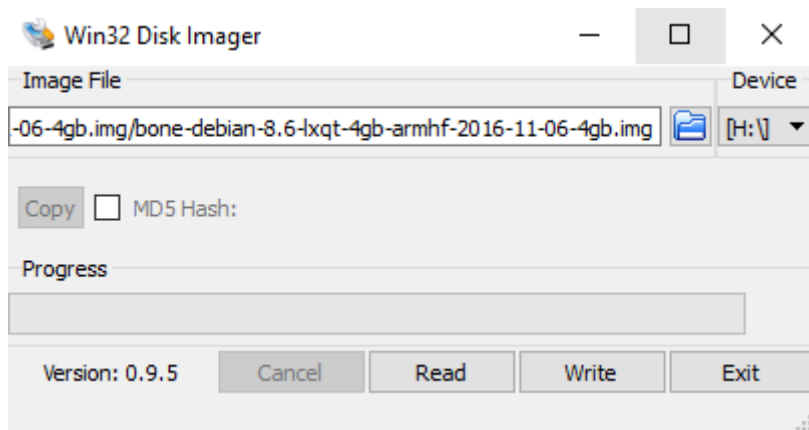


Figura 4: Win32 Disk Imager.

Después de introducir la imagen en la SD, entraremos en la página <http://www.putty.org/> y descargaremos la aplicación “putty”. Esta aplicación la usaremos como el terminal de nuestra BBB. Para ello, introduciremos su IP:

Denbora errealeko PLC-aren implementazioa Linux-a duen ARM mikrokontrolagailuaren bidez

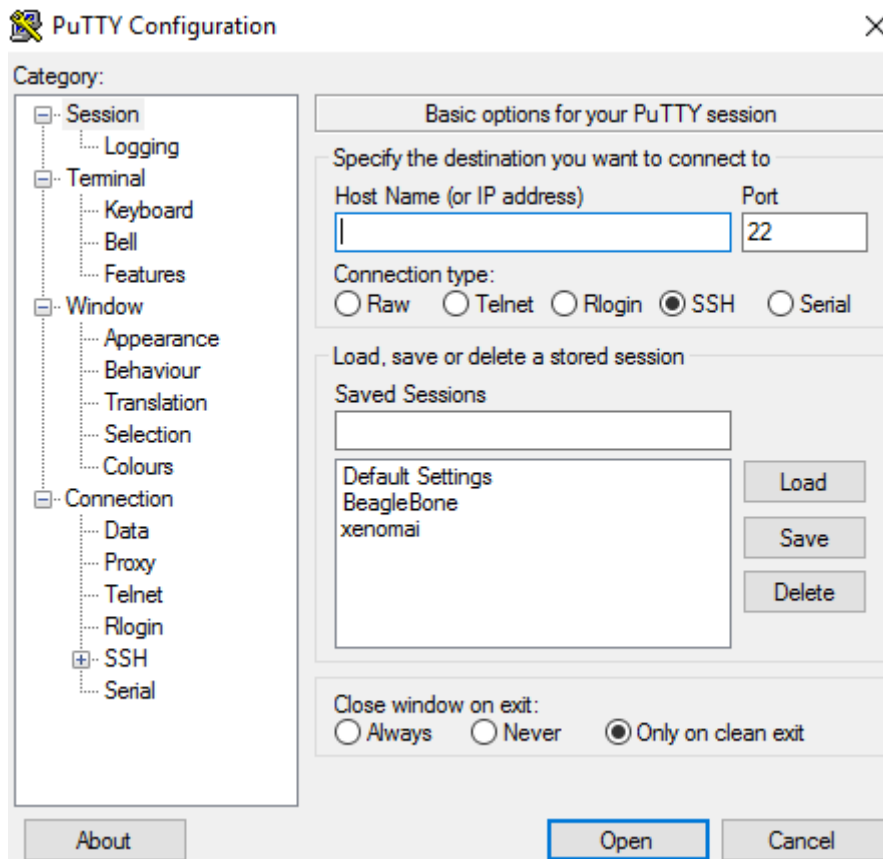


Figura 5: Configuración de Putty poniendo la IP correcta.

Si introducimos la BBB sin la SD veríamos que tiene la imagen de fábrica:

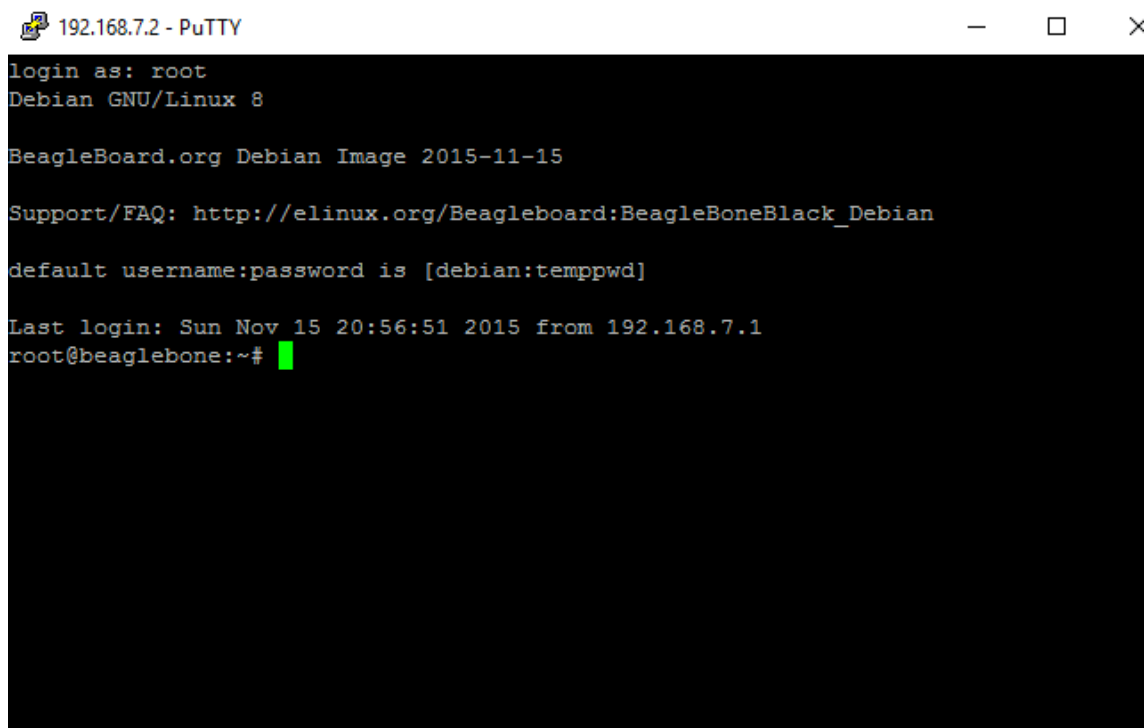
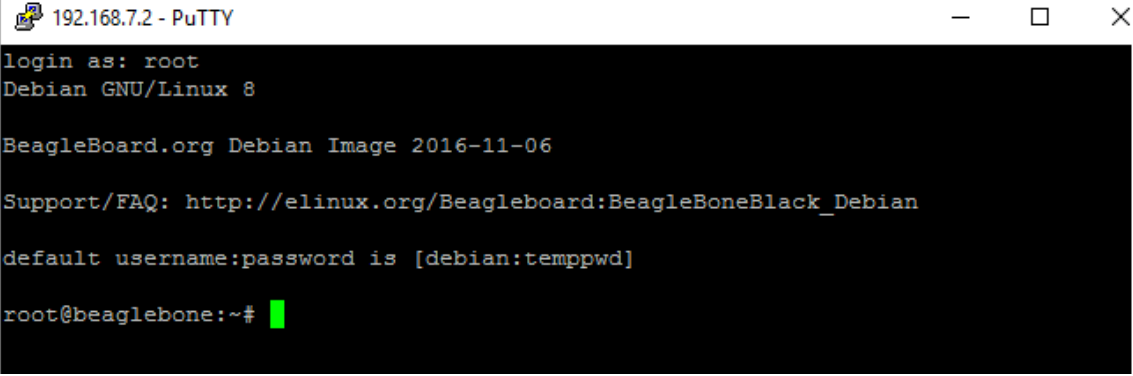


Figura 6: Terminal de Putty con la imagen de fábrica.

Si nos conectamos como “debian”, delante de cada comando debemos de poner “sudo” para mandarlo como administrador. En caso de conectarse como “root”, no tendríamos que usar “sudo” ya que estamos trabajando como administrador.

Introduciendo la SD, veríamos que la imagen está actualizada:



```
192.168.7.2 - PuTTY
login as: root
Debian GNU/Linux 8

BeagleBoard.org Debian Image 2016-11-06

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:temppwd]

root@beaglebone:~#
```

Figura 7: La nueva imagen de la BBB.

Si queremos cambiarle el Kernel, tendríamos que entrar en este link <https://rcn-ee.com/repos/debian/pool/main/l/linux-upstream/> y elegiríamos el que quisiéramos y tendríamos que escribir las siguientes funciones:

```
cd /opt/scripts/tools
git pull
-> specific version ./update_kernel.sh --kernel 4.4.14-bone-rt-r11
better:
./update_kernel.sh --bone-rt-kernel -stable
```

Después de meter la imagen y el kernel nuevo para tener mejor Real-Time, tendremos que cambiar el CPU Governor de “ondemand” a “performance” para que sea a 1GHz.

Edit file /etc/init.d/cpufrequtils and change the entry GOVERNOR="ondemand" to performance.

Después de hacer esto, todas las actualizaciones de la SD lo flashearemos en la eMMC de la BBB.

Para ello:

```
cd /opt/scripts/tools/eMMC/
./init-eMMC-flasher-v3.sh
```

Entonces, los LED-s de la BBB empezaran a parpadear y hay q esperar hasta q se apaguen. Al finalizar, podremos quitar la SD y al conectar sin SD tendremos ya la imagen y el kernel nuevo. Para comprobar esto, usaremos los comandos “dmesg” (para ver el kernel) y “cat /etc/issue” (para ver la imagen).

Al Ethernet le daremos una IP fija para luego poder usar en CODESYS. Para ello hay que configurarlo en el “Putty”.

```
nano /etc/network/interfaces
```

Al hacer esto entraremos dentro del fichero y habrá que escribir esto:



Denbora errealeko PLC-aren implementazioa Linux-a duen ARM mikrokontrolagailuaren bidez

```
auto eth0
iface eth0 inet static
address XXX.XXX.XXX.XXX → Por ejemplo: 192.168.1.69
netmask XXX.XXX.XXX.XXX → Por ejemplo: 255.255.255.0
gateway XXX.XXX.XXX.XXX → Por ejemplo: 192.168.1.197
```

Después de escribir, hay que guardar el fichero con “Ctrl+x” y luego para confirmar “y”, entonces se guardara y saldrá al terminal.

Para finalizar con la configuración:

```
connmanctl services
```

Saldrán estas dos opciones:

```
*AR Wired          ethernet_68c90bc37cb4_cable
*AR Wired          ethernet_da8859189f70_cable
```

Figura 8: Direcciones del MAC de Ethernet.

Para saber cuál de los dos es haremos “ifconfig” para elegir la de Ethernet.

```
eth0      Link encap:Ethernet  HWaddr 68:c9:0b:c3:7c:b4
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:173
```

Figura 9: Información de “eth0”.

Al elegir haremos lo siguiente:

```
connmanctl config ethernet_68c90bc37cb4_cable --ipv4 manual 192.168.1.69(gure address-a)
255.255.255.0(gure netmask-a) 192.168.1.197(gure gateway-a) --nameservers 8.8.8.8
```

Después de esto nos pondremos en marcha con CODESYS. Para ello, tendremos que descargar el programa de CODESYS por la página de “CODESYS Store”.



The screenshot displays the product listing for 'CODESYS Development System V3'. On the left, there is a red icon with a white cube and the text 'Free!' above it. The product name 'CODESYS Development System V3' is in bold red text. Below it, the company name '3S-Smart Software Solutions GmbH' and 'Order number: 1101000000' are listed. To the right, there are five red stars and the text '0 Reviews'. The price '€0.00' is shown in a large font. At the bottom, there is a description: 'The CODESYS Development System is an IEC 61131-3 programming tool for the industrial controller and automation technology sector.' and a red link 'Add to My Wishlist'.

Figura 10: La aplicación de CODESYS.

Y para usar la BBB, hay que descargar el siguiente paquete:



Figura 11: Paquete de CODESYS para control de BeagleBone.

Después de instalar el programa entraremos en CODESYS. Entonces, abriremos Herramientas y veremos que “Update BeagleBone Black” no aparece, lo que significa que hay que instalar el paquete de la BBB. Para ello, entraremos en “Herramientas → Administrador de paquetes...”.

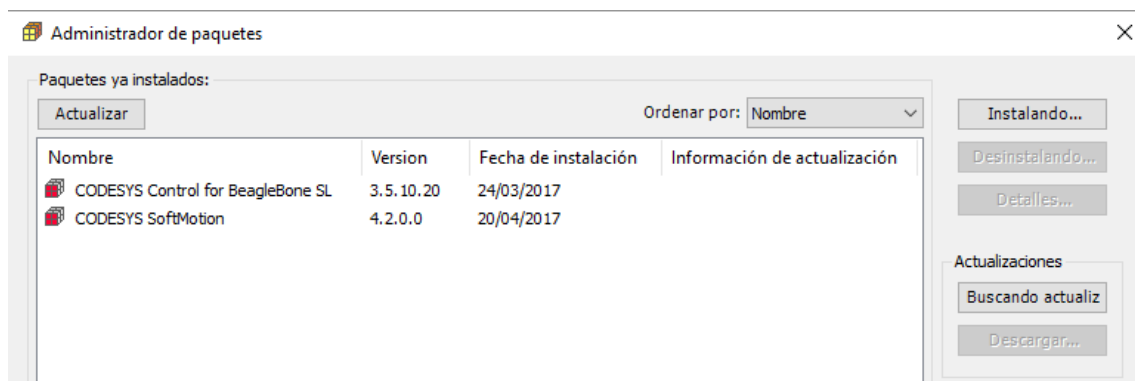


Figura 12: Ventana de “Administrador de paquetes” de CODESYS.

Elegiremos el paquete de la BBB y clicaremos donde pone Instalando... y empezara a instalarse. Después de instalar, cerraremos el programa y al volver a entrar aparecerá en “Herramientas” la opción de “Update BeagleBone Black”. Al entrar ahí le mandará el runtime de CODEYS a la BBB:

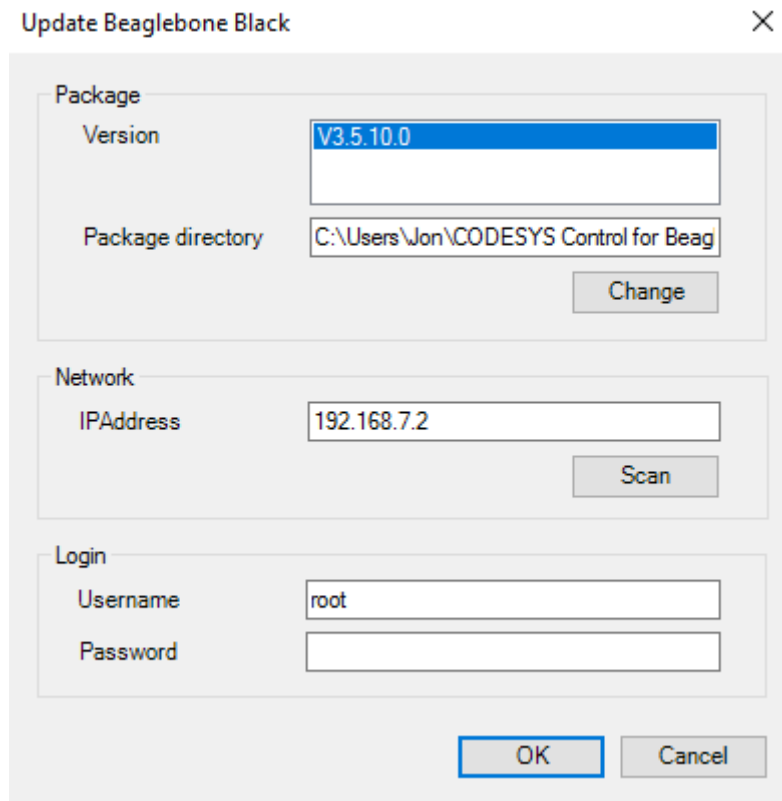


Figura 13: Ventana de "Update Beaglebone Black".

Después de esto, crearemos un nuevo proyecto. "Nuevo proyecto → Proyecto standard → Poner nombre y ubicación → Aceptar". Aparecerá la siguiente imagen:

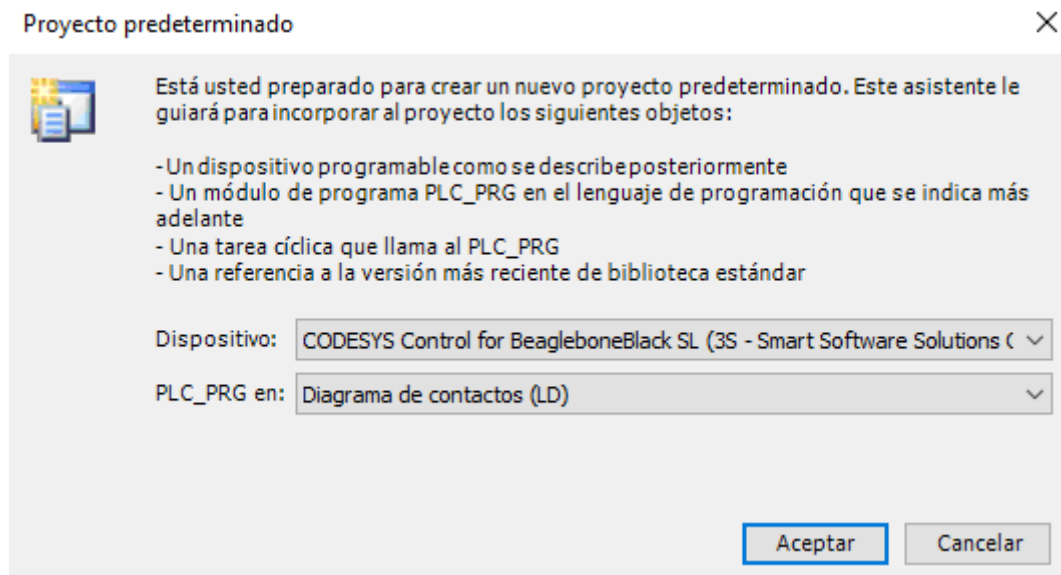


Figura 14: El dispositivo y el lenguaje de programación que elegimos al crear un proyecto.

En "Dispositivo" pondremos el paquete de BBB que hemos instalado. En "PLC\_PRG en" pondremos el lenguaje de programación que queramos usar.

Denbora errealeko PLC-aren implementazioa Linux-a duen ARM mikrokontrolagailuaren bidez

Al aceptar, a la izquierda en la pantalla aparecerá una ventana llamada “Dispositivos”.

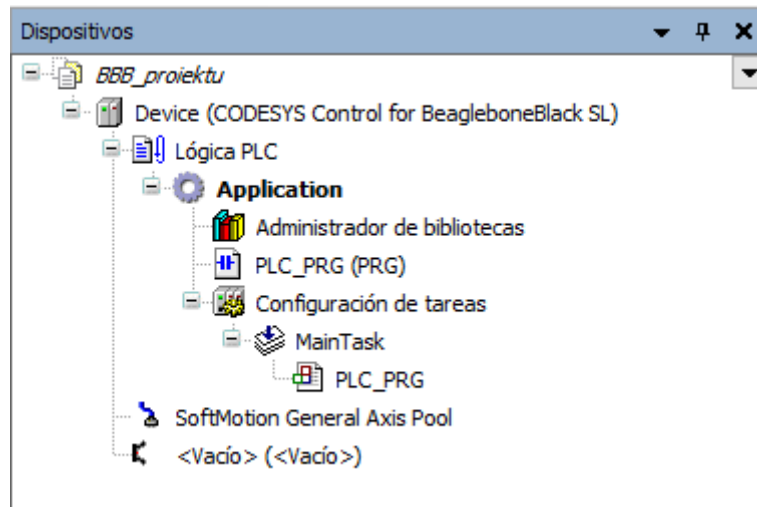


Figura 15: La ventana “Dispositivos” al principio.

Al clicar en “Device” aparecerá la siguiente ventana.

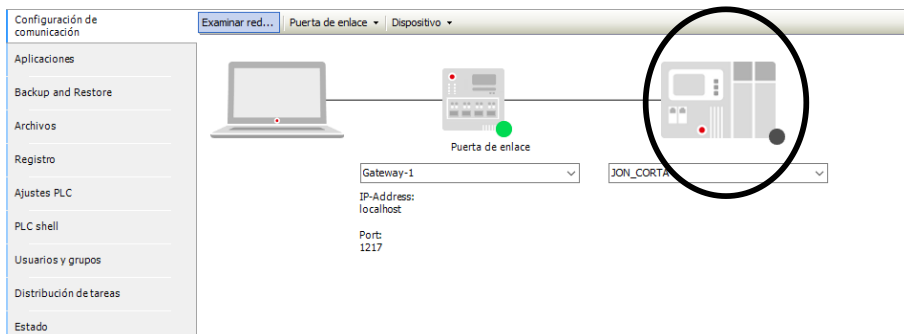


Figura 16: La ventana “Device” enseñando el estado de la comunicación red.

Ahora, clicaremos en “Examinar red...” para hacer la conexión con la BBB:

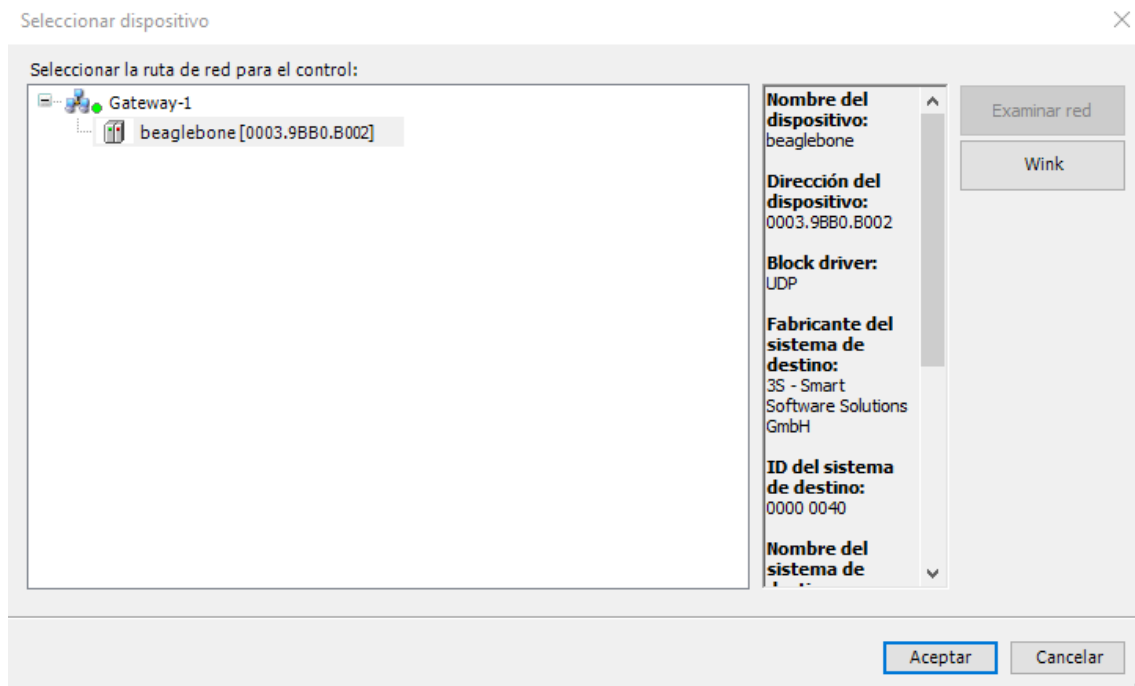


Figura 17: La ventana de después de clicar “Examinar red...”.

Ahí elegiremos “beaglebone” y cuando se conecte el LED apagado se pondrá en verde:

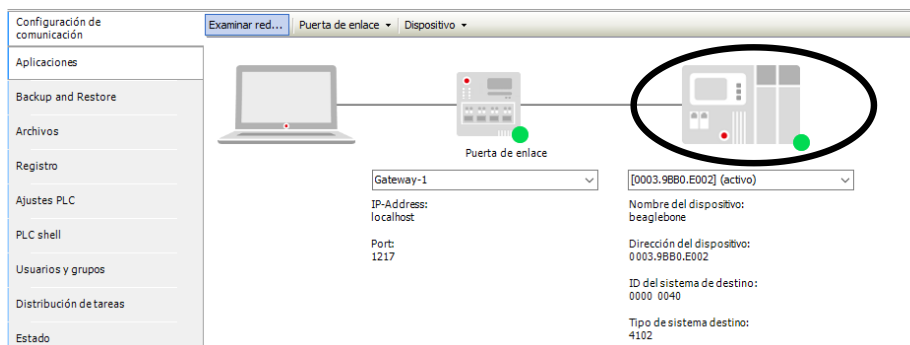


Figura 18: En la ventana “Device” con la BBB conectada.

En “Device”, entraremos en “Ajustes PLC” y haremos algunos cambios.

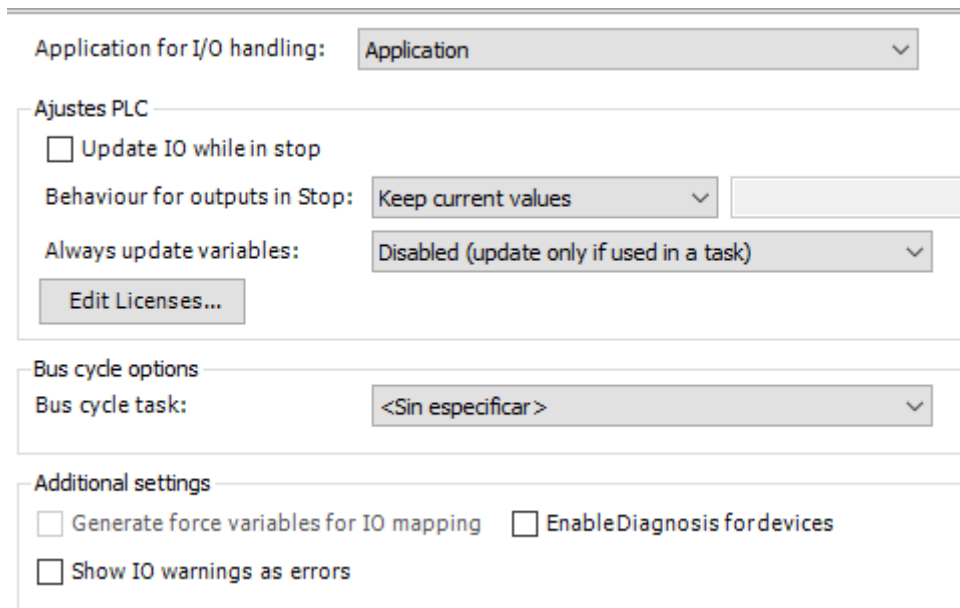


Figura 19: La ventana “Ajustes PLC” de la ventana de “Device”.

En “Always update variables”, en vez de “Disabled” hay que poner “Enabled 2”. Volveremos a la ventana “Dispositivos” y ahora clicaremos con el botón de la derecha en donde pone “<vacío>” y elegiremos “Establecer dispositivo”.

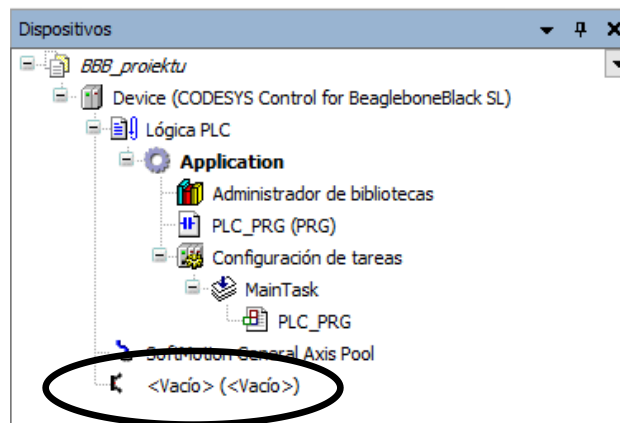


Figura 20: La ventana “Dispositivos” para configurar los GPIOs.

Entonces, estableceremos el dispositivo que se muestra en la siguiente imagen:

Denbora errealeko PLC-aren implementazioa Linux-a duen ARM mikrokontrolagailuaren bidez

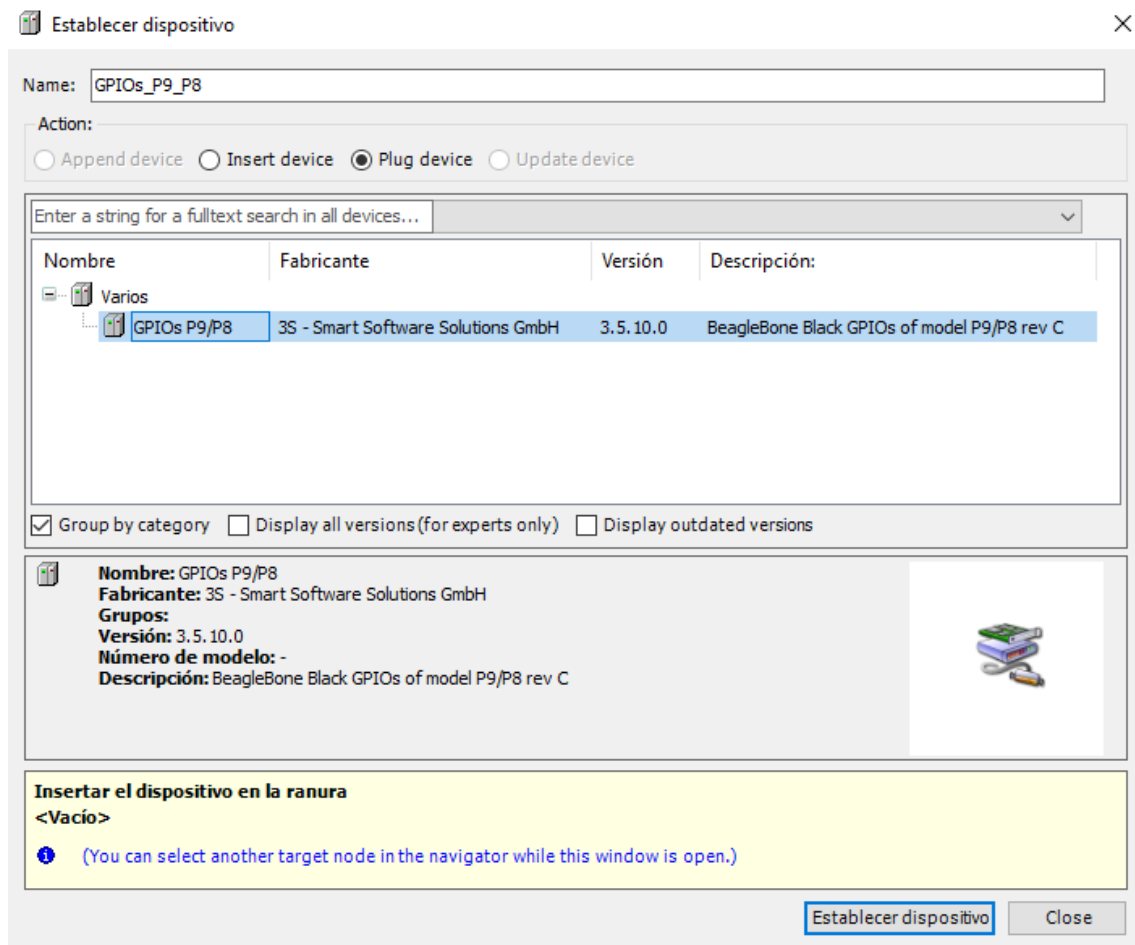


Figura 21: La ventana “Establecer dispositivo” para elegir los GPIOs.

Entonces, clicaremos “Establecer dispositivo” y aparecería en la ventana “Dispositivos”:

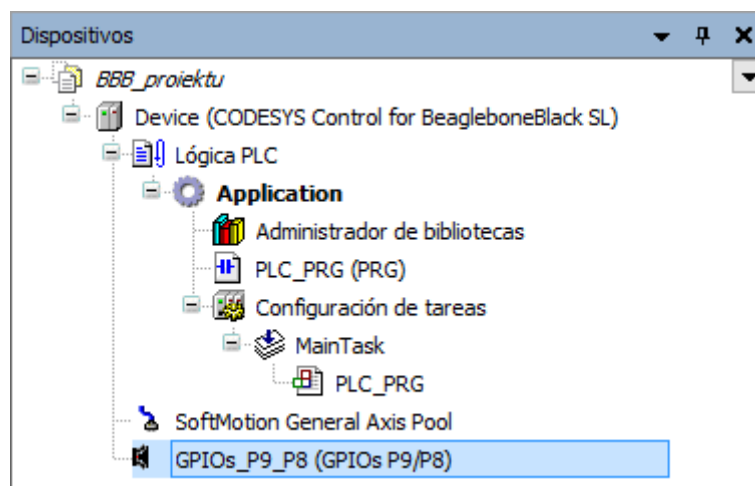


Figura 22: En la ventana “Dispositivos” introducido “GPIOs\_P9\_P8”.

Ahora haremos la configuración de los pines. Para ello, entraremos en “GPIOs\_P9\_P8” y se abrirá la próxima ventana:

| Parámetro | Tipo                | Valor    | V... | Unidad | Descripción |
|-----------|---------------------|----------|------|--------|-------------|
| GPIO_0    | Enumeration of BYTE | not used | n... |        | n/c         |
| GPIO_1    | Enumeration of BYTE | not used | n... |        | n/c         |
| GPIO_2    | Enumeration of BYTE | not used | n... |        | P9 Pin 22   |
| GPIO_3    | Enumeration of BYTE | not used | n... |        | P9 Pin 21   |
| GPIO_4    | Enumeration of BYTE | not used | n... |        | P9 Pin 18   |
| GPIO_5    | Enumeration of BYTE | not used | n... |        | P9 Pin 17   |
| GPIO_6    | Enumeration of BYTE | not used | n... |        | n/c         |
| GPIO_7    | Enumeration of BYTE | Output   | n... |        | P9 Pin 42   |
| GPIO_8    | Enumeration of BYTE | not used | n... |        | P8 Pin 35   |
| GPIO_9    | Enumeration of BYTE | not used | n... |        | P8 Pin 33   |
| GPIO_10   | Enumeration of BYTE | not used | n... |        | P8 Pin 31   |
| GPIO_11   | Enumeration of BYTE | not used | n... |        | P8 Pin 32   |
| GPIO_12   | Enumeration of BYTE | not used | n... |        | n/c         |
| GPIO_13   | Enumeration of BYTE | not used | n... |        | n/c         |

Figura 23: La ventana “GPIOs Configuración” de la “GPIOs\_P9\_P8”.

Para realizar la prueba he elegido el pin “P9 Pin 42” de la BBB como salida. Entonces, le ponemos el valor de “Output”.

Después, entraremos en “GPIOs Asignación E/S” y le daremos nombre al pin elegido. Como lo hemos elegido como salida, hay que elegir “digital outputs 0-31”.

| Variable | Asignación | Canal                  | Dirección | Tipo  | Unidad | Descripción |
|----------|------------|------------------------|-----------|-------|--------|-------------|
|          |            | digital inputs 0-31    | %ID0      | DWORD |        |             |
|          |            | digital inputs 32-63   | %ID1      | DWORD |        |             |
|          |            | digital inputs 64-95   | %ID2      | DWORD |        |             |
|          |            | digital inputs 96-127  | %ID3      | DWORD |        |             |
|          |            | digital outputs 0-31   | %QD0      | DWORD |        |             |
|          |            | digital outputs 32-63  | %QD1      | DWORD |        |             |
|          |            | digital outputs 64-95  | %QD2      | DWORD |        |             |
|          |            | digital outputs 96-127 | %QD3      | DWORD |        |             |
|          |            | analog input 0         | %IW8      | UINT  |        |             |
|          |            | analog input 1         | %IW9      | UINT  |        |             |
|          |            | analog input 2         | %IW10     | UINT  |        |             |
|          |            | analog input 3         | %IW11     | UINT  |        |             |
|          |            | analog input 4         | %IW12     | UINT  |        |             |
|          |            | analog input 5         | %IW13     | UINT  |        |             |

Figura 24: La ventana “GPIOs Asignación E/S” de “GPIOs\_P9\_P8”.

Entonces aparecerán todos los pines del 0 al 31 y tendremos que elegir el nuestro, que será el “GPIO\_7”. Le daremos un nombre cualquiera para poder identificarlo. Con esto acabaríamos con la configuración.



| Variable | Asignación | Canal                 | Dirección | Tipo  | Unidad | Descripción |
|----------|------------|-----------------------|-----------|-------|--------|-------------|
|          |            | digital inputs 96-127 | %ID3      | DWORD |        |             |
|          |            | digital outputs 0-31  | %QD0      | DWORD |        |             |
|          |            | GPIO_0                | %QX0.0    | BOOL  |        | n/c         |
|          |            | GPIO_1                | %QX0.1    | BOOL  |        | n/c         |
|          |            | GPIO_2                | %QX0.2    | BOOL  |        | P9 Pin 22   |
|          |            | GPIO_3                | %QX0.3    | BOOL  |        | P9 Pin 21   |
|          |            | GPIO_4                | %QX0.4    | BOOL  |        | P9 Pin 18   |
|          |            | GPIO_5                | %QX0.5    | BOOL  |        | P9 Pin 17   |
|          |            | GPIO_6                | %QX0.6    | BOOL  |        | n/c         |
| LED      |            | GPIO_7                | %QX0.7    | BOOL  |        | P9 Pin 42   |
|          |            | GPIO_8                | %QX1.0    | BOOL  |        | P8 Pin 35   |
|          |            | GPIO_9                | %QX1.1    | BOOL  |        | P8 Pin 33   |
|          |            | GPIO_10               | %QX1.2    | BOOL  |        | P8 Pin 31   |
|          |            | GPIO_11               | %QX1.3    | BOOL  |        | P8 Pin 32   |

Figura 25: En la ventana “GPIOs Asignación E/S” con el nombre “LED” en el canal “GPIO\_7”.

## 2.2. EtherCAT

### Configuración del módulo de EtherCAT

#### EL1004/EL2004    EK1100 CABECERA

Módulo: pin 1 → CODESYS: bit 0

Módulo: pin 2 → CODESYS: bit 1

Módulo: pin 7 → CODESYS: bit 2

Módulo: pin 8 → CODESYS: bit 3

Para comenzar con EtherCAT, en CODESYS en “Dispositivos” clicaremos con el botón derecho encima de “Device” y le daremos a “Agregar el dispositivo” y ahí elegiremos “EtherCAT”.

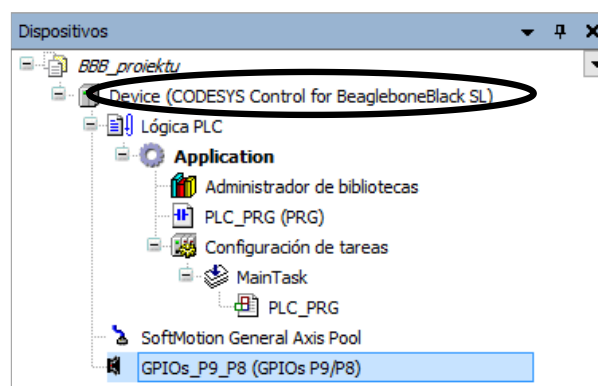


Figura 269: En la ventana “Dispositivos” la opción “Device”.

Al elegir esa opción se abrirá la siguiente ventana:

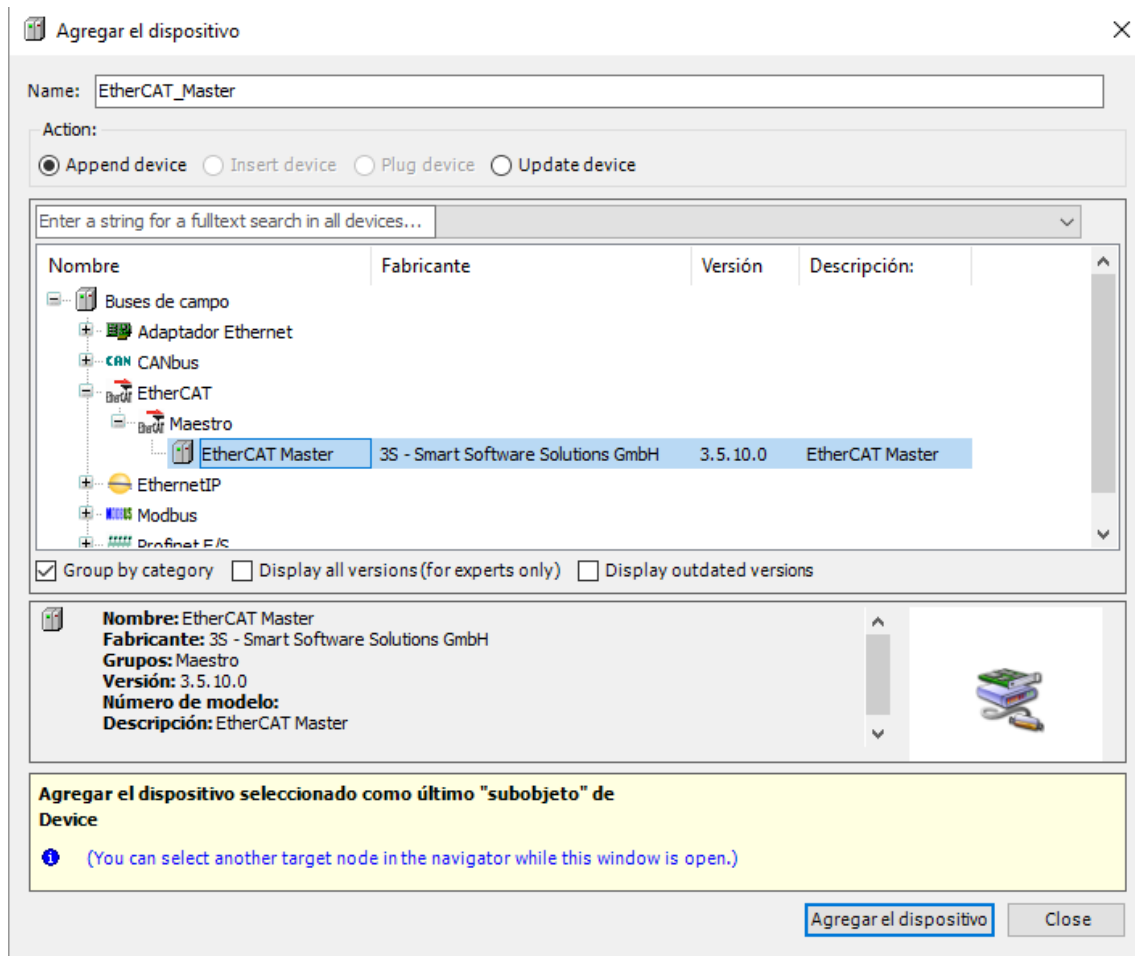


Figura 30: La ventana “Agregar el dispositivo” elegido EtherCAT.

Después de “Agregar dispositivo”, aparecería en “Dispositivos”. Entonces, en CODESYS haríamos “Iniciar la sesión” y encima de “EtherCAT” clicaríamos con el botón derecho y elegiríamos “Buscar el dispositivo”. Con esto, deberían de aparecer los módulos de EtherCAT conectados a la BBB. Pero para ello, tendríamos que entrar en la página de BECKHOFF y descargar la extensión .xml correspondiente a nuestros módulos. Entonces, “Herramientas”→”Repositorio de dispositivos” y aquí debemos de instalar las extensiones correspondientes eligiéndolas y luego dando a “aceptar”. Ahora al hacer “Buscar el dispositivo” aparecerían nuestros módulos conectados. Para ello, primero conectaremos los módulos vía Ethernet a nuestra BBB.

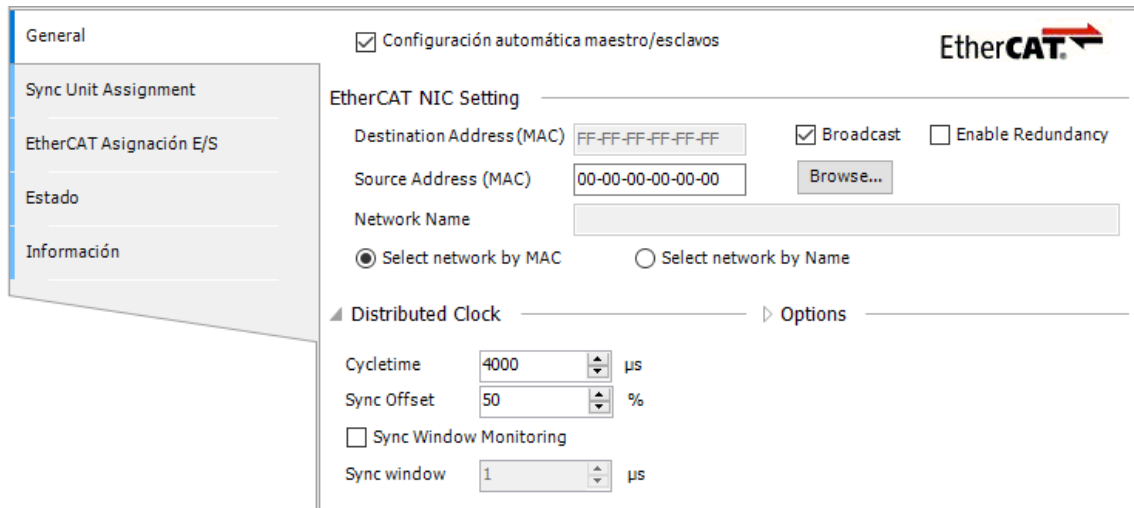


Figura 31: La ventana de EtherCAT.

En “Source Address” debemos de introducir el MAC de nuestro Ethernet, entonces clicando en “Browse” elegiremos “eth0”.

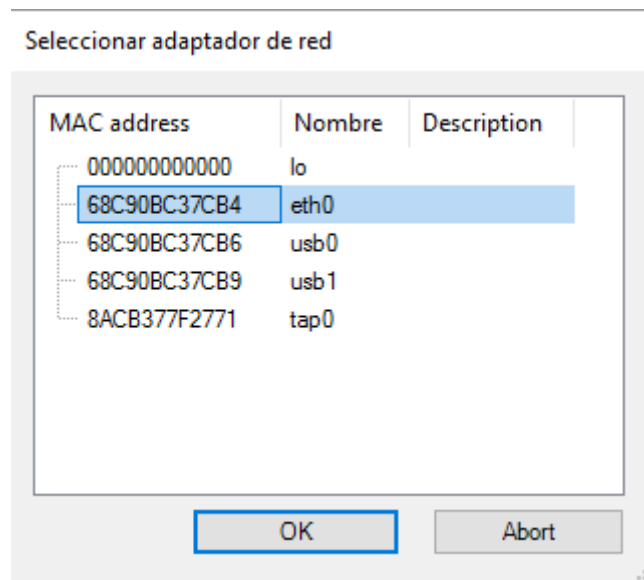


Figura 32: La ventana “Browse” de “Source Address”.

Después de esto, haremos “Iniciar la sesión” y clicando en “Ethercat” con el botón derecho elegiremos “Buscar dispositivo”. Entonces, aparecerán nuestros módulos conectados:

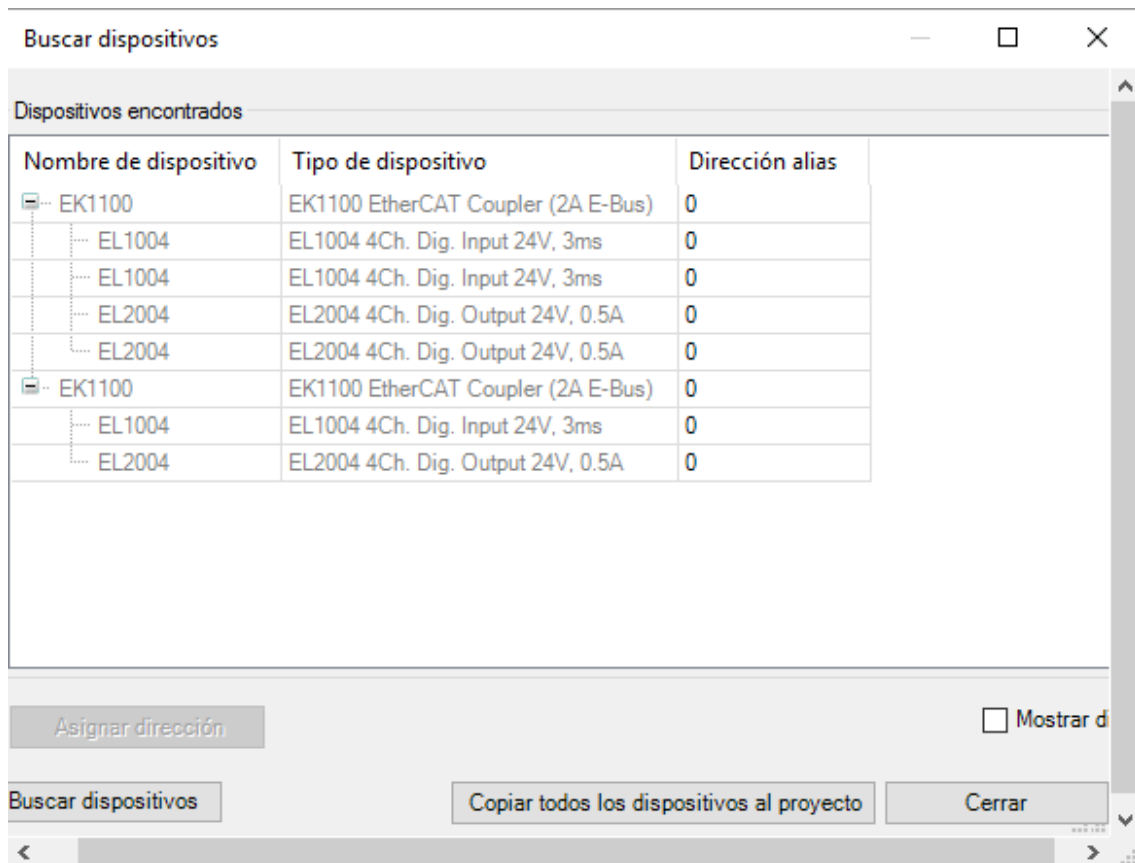


Figura 33: La ventana “Buscar dispositivos”.

Cuando aparezcan los nombres de nuestros módulos haremos “Copiar todos los dispositivos al proyecto” y así entraran todos dentro de “EtherCAT master” como esclavos. Entonces, clicaríamos “Inicio” y todos se pondrían en verde. Esto quiere decir que esta correcto y lo único que faltaría sería la configuración de los pines para probar si funciona.

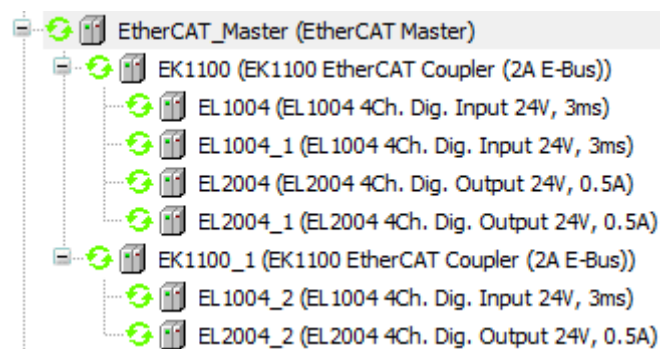


Figura 34: En “Dispositivos” EtherCAT conectado.

Las entradas son los módulos nombrados “EL1004” y las salidas son los módulos “EL2004”. Entonces, haciendo la configuración de cada uno podremos ver si funcionan. La configuración de entrada se podrá hacer en cualquier modulo del “EL1004” pero para que funcione tendremos que elegir el modulo correcto para poder configurar sus pines en CODESYS. Entonces clicaremos dos veces en el módulo que queramos y le pondremos nombre a cualquier pin.

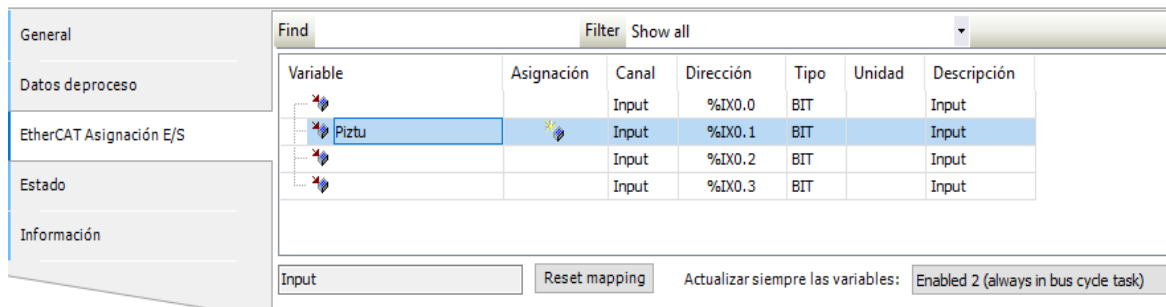


Figura 35: La ventana “EtherCAT Asignación E/S” de los módulos de entrada de EtherCAT.

Ahora, elegiremos el pin de salida que usaremos en el próximo programa para ver si funciona.

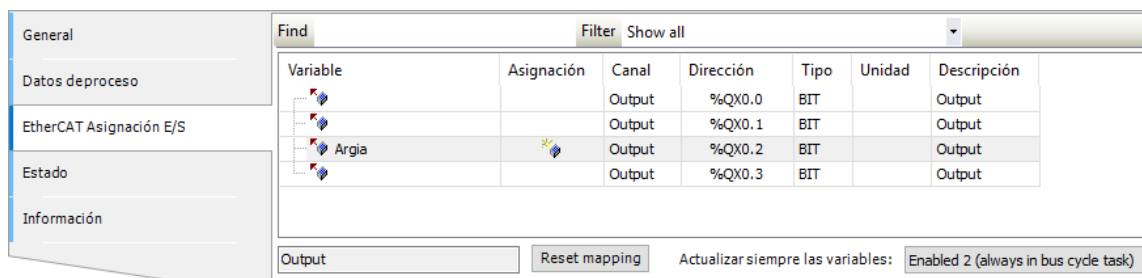


Figura 36: La ventana “EtherCAT Asignación E/S” de los módulos de salida de EtherCAT.

## 2.3. PROFINET

### Configuración del módulo de PROFINET

**4 DI ST DC24V(6ES7 131-4BD01-0AA0) 4 DO ST DC24V/0.5A(6ES7 132-4BD02-0AA0)**

Módulo: pin 1 → CODESYS: bit 0

Módulo: pin 5 → CODESYS: bit 1

Módulo: pin 2 → CODESYS: bit 2

Módulo: pin 6 → CODESYS: bit 3

Al principio, sería lo mismo que EtherCAT. Dentro de “Dispositivos” clicaríamos con el botón de la derecha en “Device” y haríamos “Agregar el dispositivo” pero en este caso elegiríamos “Profinet”.

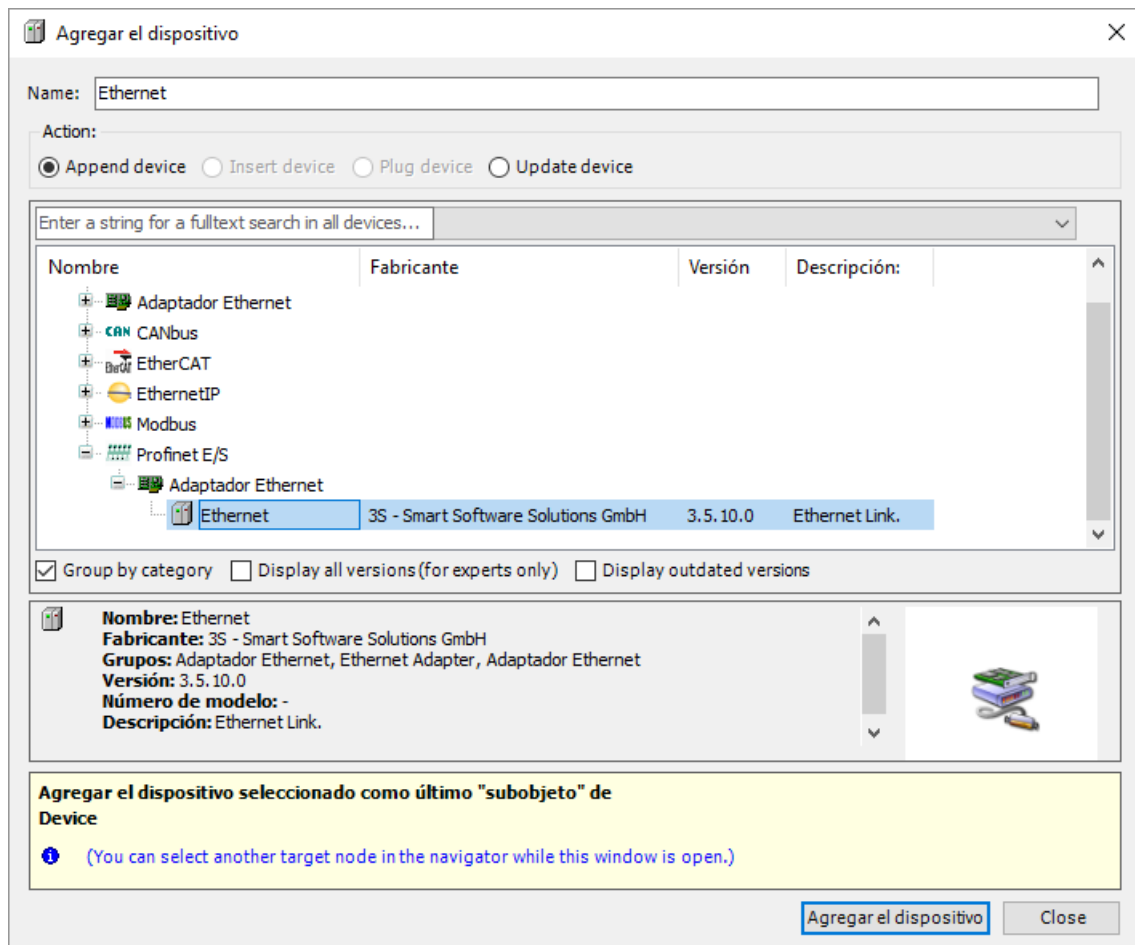


Figura 37: “Agregar el dispositivo” para PROFINET.

Después de clicar “Agregar el dispositivo” aparecería en “Dispositivos”. Entonces, entraríamos ahí y nos conectaríamos con el módulo PROFINET vía IP.

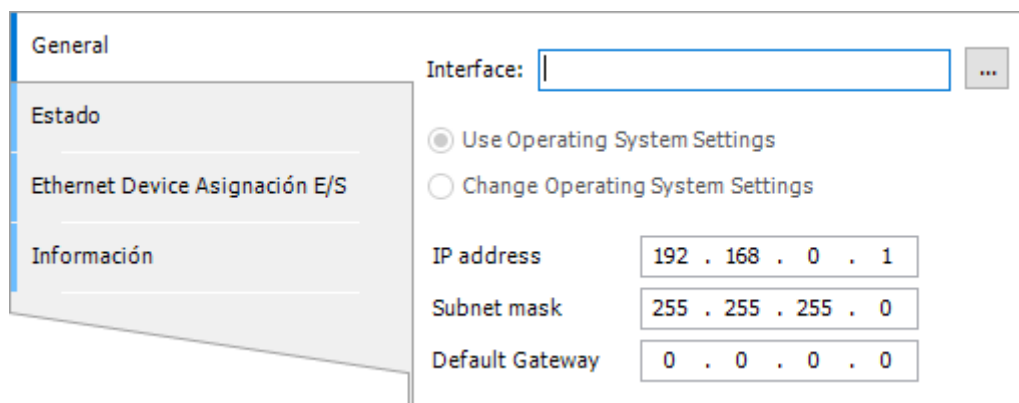


Figura 38: La ventana de PROFINET.

En esta ventana, clicaríamos en los tres puntos del “Interface” y de las opciones que aparecen tendríamos que elegir “eth0”.

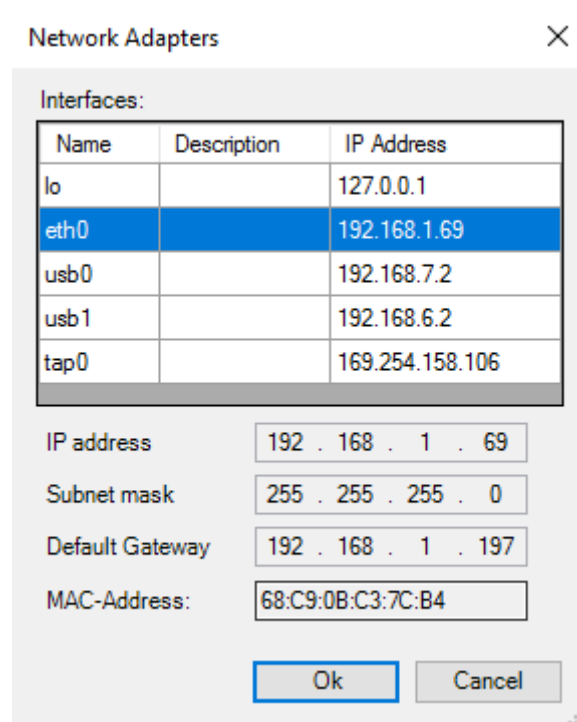


Figura39: Las opciones del "Interface".

Después de conectarse, clicaremos con el botón derecho encima de "Ethernet" y elegiremos otra vez "Agregar el dispositivo". En esa ventana elegiremos "Profinet E/S" → "Maestro Profinet E/S" → "PN-Controller" y le daremos a "Agregar dispositivo" copiándolo en CODESYS.

Denbora errealeko PLC-aren implementazioa Linux-a duen ARM mikrokontrolagailuaren bidez

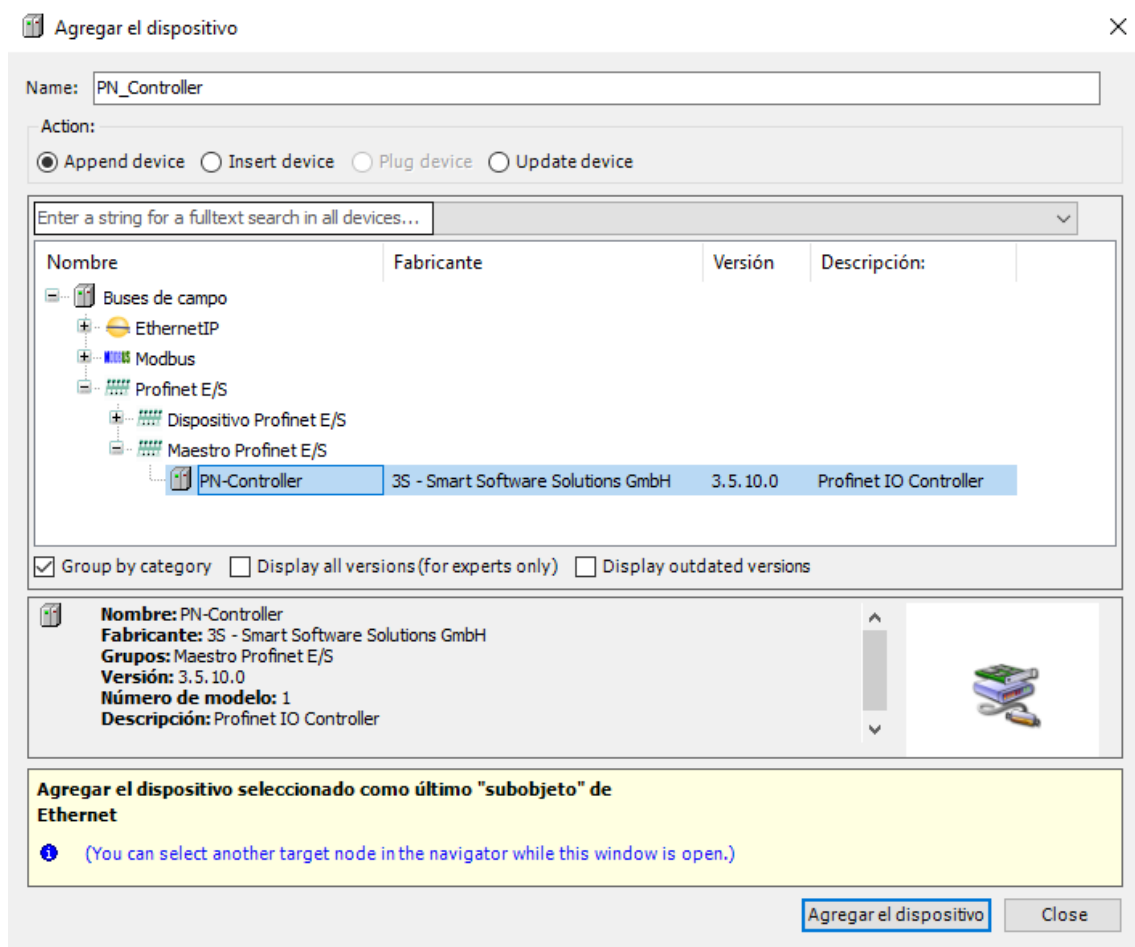


Figura 40: "Agregar el dispositivo" para instalar "PN-Controller".

Después de "Agregar el dispositivo" entraremos en la ventana de "PN-Controller".

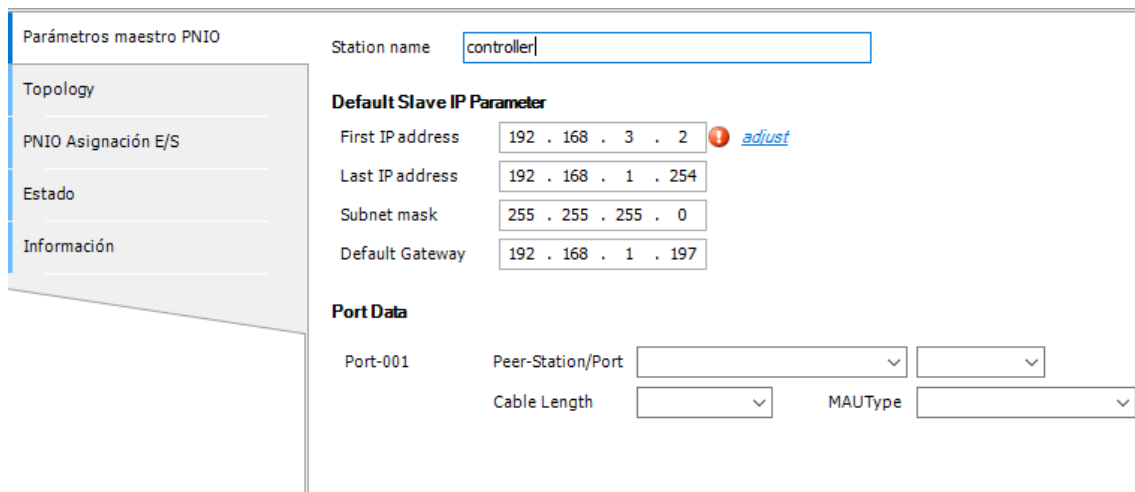


Figura 41: La ventana "Parámetros maestro PNIO" de "PN-Controller".

En este caso, pondremos la IP del esclavo clicando en "adjust". Después pasaremos a la parte de "Topology".



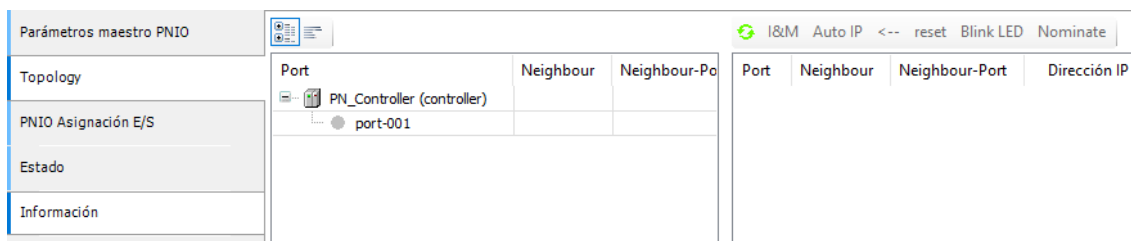


Figura 42: La ventana “Topology” de “PN-Controller”.

Para empezar, clicaremos en el “Refresh” verde, así encontrara los módulos que están conectados.



Figura 43: La ventana “Topology” de “PN-Controller” actualizada.

Después, al igual que en EtherCAT hay que buscar los .xml del PROFINET pero en este caso en la página de SIEMENS ya que los módulos son de esta casa. Al encontrar nuestros módulos hay que descargarlos e instalarlos en “Repositorio de dispositivos”. Luego haremos “Iniciar sesión” y elegiremos “Buscar el dispositivo” en “PN-Controller” para encontrar nuestros módulos y poder copiarlos en CODESYS.

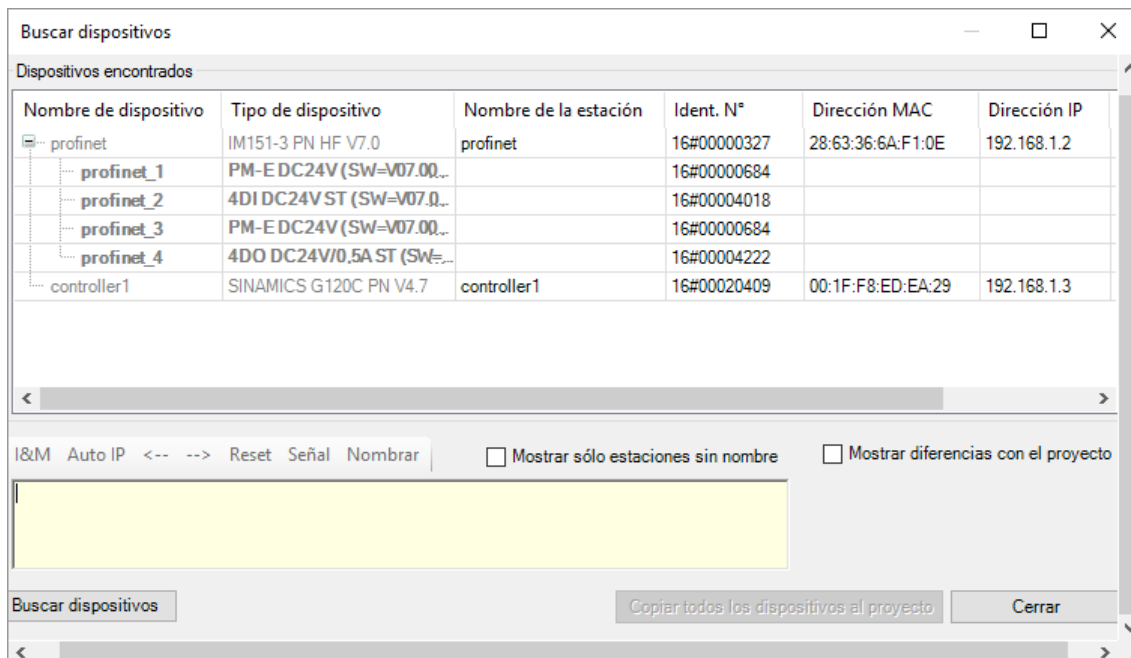


Figura 44: La ventana de “Buscar dispositivos” de “PN-Controller”.

El nombre del dispositivo “profinet” es nuestro módulo de PROFINET. El nombre “controller1” es el inversor de PROFINET.

Como antes, entraremos en “Topology” y ahora con los módulos introducidos clicaremos en el “Refresh” verde y luego con la flecha de la derecha de “Auto IP” lo meteremos en nuestro programa.



Figura 45: La ventana “Topology” de “PN-Controller” actualizada.

Como el problema del Jitter no se ha solucionado, mientras envía los paquetes suele dar errores entonces tendríamos que incrementar los valores del “send clock” y “watchdog” para que no ocurra. Entonces, pondríamos a 4ms el “send clock” y a 20ms el “watchdog” de los módulos “profinet” y “controller1”.

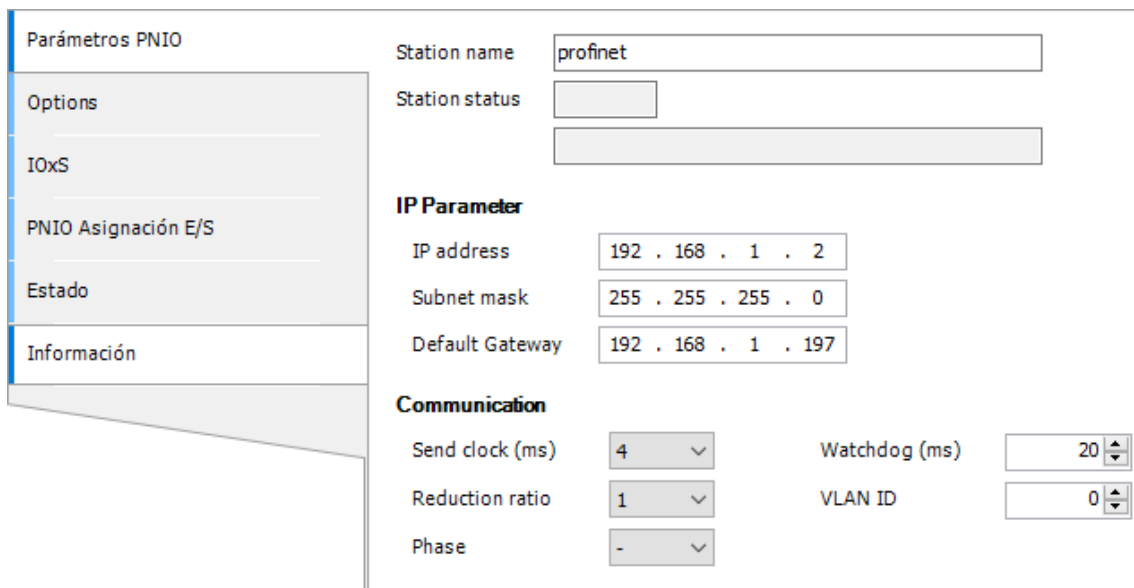


Figura 46: La ventana “Parámetros PNIO” del PROFINET.

Después de configurar todo, clicaremos en “Iniciar la sesión” y si todo se ponen en verde significa que no hay ningún error:

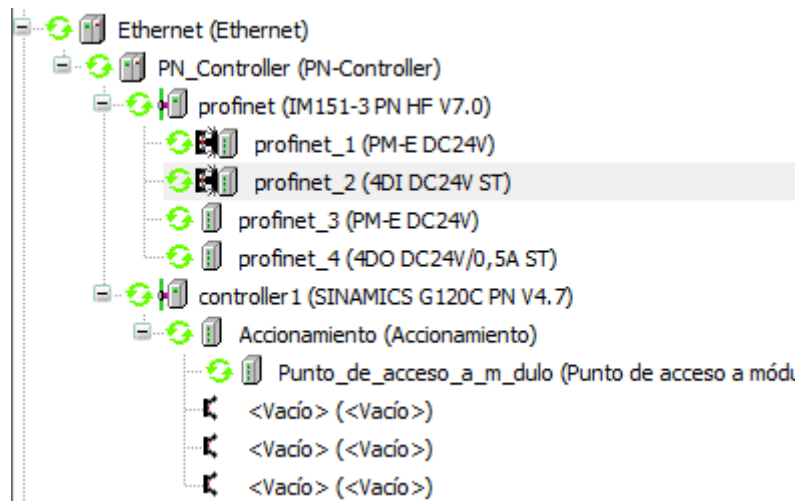


Figura 47: La parte de PROFINET en “Dispositivos”.

Ahora para ver si las entradas y salidas del módulo funcionan, haremos una prueba encendiendo un LED. Para ello, elegiré “profinet\_2” como entrada y “profinet\_4” como salida.

Dentro de “profinet\_2” entraremos en “PNIO Module Asignación E/S” y elegiremos el bit que queramos usar como entrada. En CODESYS aparecen 8 bits, pero en el módulo solo tiene 4 bits.

The screenshot shows the 'PNIO Module Asignación E/S' configuration window. The left sidebar has 'Parámetros PNIO', 'PNIO Module Asignación E/S', 'Estado', and 'Información'. The main area has a 'Find' bar and a table with columns: 'Variable', 'Asignación', 'Canal', 'Dirección', 'Tipo', 'Unidad', and 'Descripción'. The table lists bits 0 through 7. Bit 1 is selected and has the variable 'pizteko' assigned to it. Below the table are 'Reset mapping' and 'Actualizar siempre las variables: Enabled 2 (always in bus cycle task)' buttons. A legend at the bottom indicates that a yellow star icon means 'Create new variable' and a blue star icon means 'Map to existing variable'.

| Variable | Asignación | Canal    | Dirección | Tipo  | Unidad | Descripción |
|----------|------------|----------|-----------|-------|--------|-------------|
|          |            | Entradas | %IB2      | USINT |        |             |
|          |            | Bit0     | %IX2.0    | BOOL  |        |             |
| pizteko  |            | Bit1     | %IX2.1    | BOOL  |        |             |
|          |            | Bit2     | %IX2.2    | BOOL  |        |             |
|          |            | Bit3     | %IX2.3    | BOOL  |        |             |
|          |            | Bit4     | %IX2.4    | BOOL  |        |             |
|          |            | Bit5     | %IX2.5    | BOOL  |        |             |
|          |            | Bit6     | %IX2.6    | BOOL  |        |             |
|          |            | Bit7     | %IX2.7    | BOOL  |        |             |

Figura 48: La ventana “PNIO Module Asignación E/S” de “PROFINET\_2”.

Dentro de “profinet\_4” entraremos en “PNIO Module Asignación E/S” y elegiremos el bit que queramos usar como salida.

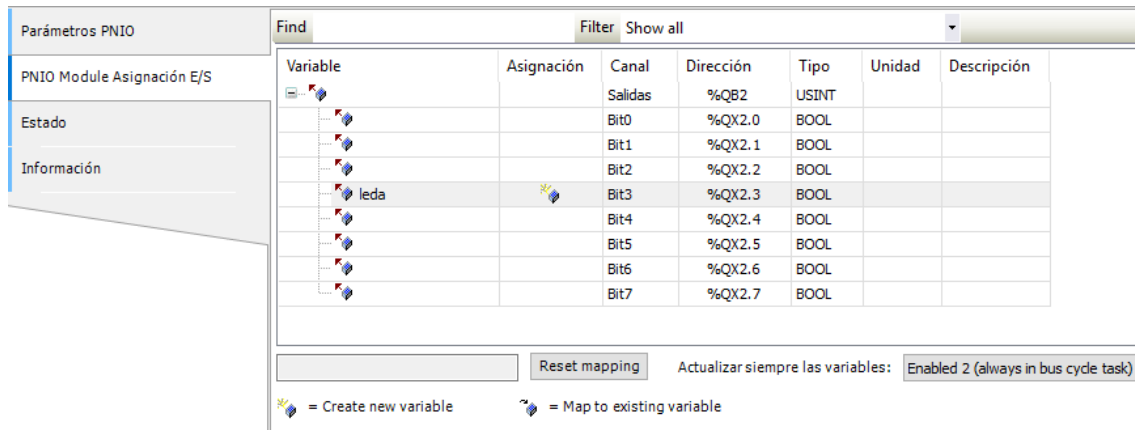


Figura 49: La ventana “PNIO Module Asignación E/S” de “PROFINET\_4”.

## 2.4. CANopen

### Configuración del módulo de CANopen

#### (KL1804-1/KL1804-2)INPUT

##### KL1804-1

Módulo: pin 1 → CODESYS: bit 0  
 Módulo: pin 3 → CODESYS: bit 1  
 Módulo: pin 5 → CODESYS: bit 2  
 Módulo: pin 7 → CODESYS: bit 3

##### KL1804-2

Módulo: pin 1 → CODESYS: bit 4  
 Módulo: pin 3 → CODESYS: bit 5  
 Módulo: pin 5 → CODESYS: bit 6  
 Módulo: pin 7 → CODESYS: bit 7

#### (KL2404-1/KL2404-2)OUTPUT

##### KL2404-1

Módulo: pin 1 → CODESYS: bit 0  
 Módulo: pin 5 → CODESYS: bit 1  
 Módulo: pin 4 → CODESYS: bit 2  
 Módulo: pin 8 → CODESYS: bit 3

##### KL2404-2

Módulo: pin 1 → CODESYS: bit 4  
 Módulo: pin 5 → CODESYS: bit 5  
 Módulo: pin 4 → CODESYS: bit 6  
 Módulo: pin 8 → CODESYS: bit 7

## **KL9010 bus terminal**

Para la comunicación CANbus hay que usar la capa “RS485 CAN CAPE”. Para que esta capa funcione, hay que configurar el “can0” en la terminal de la BBB. Para ello hay que seguir los siguientes pasos:

```
modprobe can
```

```
modprobe can-dev
```

```
modprobe can-raw
```

```
ip link set can0 up type can bitrate 125000
```

Al usar el último comando nos dirá que “can0” no se encuentra:

```
Cannot find device "can0"
```

Entonces, hay que introducir el “can0” dentro de la BBB. Para ello tenemos que seguir este proceso:

```
sh -c "echo 'BB-CAN1'> /sys/devices/platform/bone_capemgr/slots"
```

Después de esto ya estaría incorporado. Para comprobarlo tendremos que meter el siguiente comando:

```
cat /sys/devices/platform/bone_capemgr/slots
```

Nos aparecería el siguiente mensaje:

```
0: PF---- -1
1: PF---- -1
2: PF---- -1
3: PF---- -1
4: P-O-L- 0 Override Board Name,00A0,Override Manuf,BB-CAN1
```

Ahora meteremos el comando que antes nos daba error y tendría q funcionar sin ningún problema:

```
ip link set can0 up type can bitrate 125000
```

```
ifconfig can0 up
```

Para ver que se ha configurado bien:

```
Ifconfig
```

Entonces, aparecería “can0” en el próximo mensaje:

```
192.168.7.2 - PuTTY
root@beaglebone:/# ifconfig
can0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
-00
          UP RUNNING NOARP  MTU:16  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:187

eth0      Link encap:Ethernet  HWaddr 68:c9:0b:c3:7c:b4
          UP BROADCAST MULTICAST DYNAMIC  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:173

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:15848 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15848 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:1220960 (1.1 MiB)  TX bytes:1220960 (1.1 MiB)

usb0      Link encap:Ethernet  HWaddr 68:c9:0b:c3:7c:b6
          inet addr:192.168.7.2  Bcast:192.168.7.3  Mask:255.255.255.252
          inet6 addr: fe80::6ac9:bff:fec3:7cb6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1620 errors:0 dropped:0 overruns:0 frame:0
          TX packets:664 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:286545 (279.8 KiB)  TX bytes:153905 (150.2 KiB)
```

Figura 50: El mensaje del comando “ifconfig” con la configuración del CAN en activo.

En esta imagen podemos ver las configuraciones de la BBB. Al principio “can0” no estaba y ahora después de configurarlo, aparece.

Encima, “can0” hay que ponerlo manual para que cada vez que te conectes solo tengas que meter los dos siguientes comandos:

```
ip link set can0 up type can bitrate 125000
```

```
ifconfig can0 up
```

Entonces, para que sea estático:

```
cd /etc/network
```

```
nano interfaces
```

Con el comando “nano” entraríamos en ese fichero y tendríamos opciones de hacer cambios ahí dentro. En este caso habría que escribir lo siguiente:

```
auto can0
iface can0 inet manual
    #pre-up ip link set $IFACE type can bitrate 125000 listen-only off
    pre-up /sbin/ip link set $IFACE type can bitrate 125000 triple-sampling on
    up /sbin/ifconfig $IFACE up
    down /sbin/ifconfig $IFACE down
```



Denbora errealeko PLC-aren implementazioa Linux-a duen ARM mikrokontrolagailuaren bidez

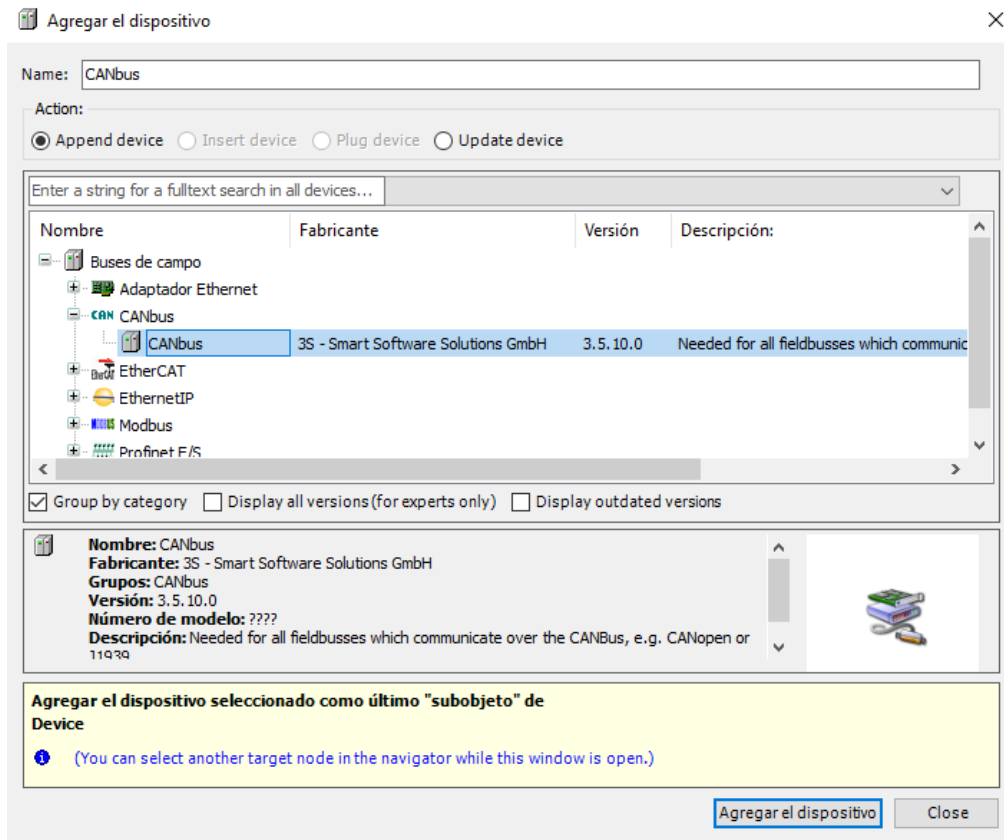


Figura 52: La ventana “Agregar el dispositivo” para CANopen.

Después de “Agregar el dispositivo”, entraremos en “CANbus” donde pone “generalidades” y en “Velocidad en baudios” pondremos 125000bit/s porque antes lo hemos configurado así en la terminal de la BBB.

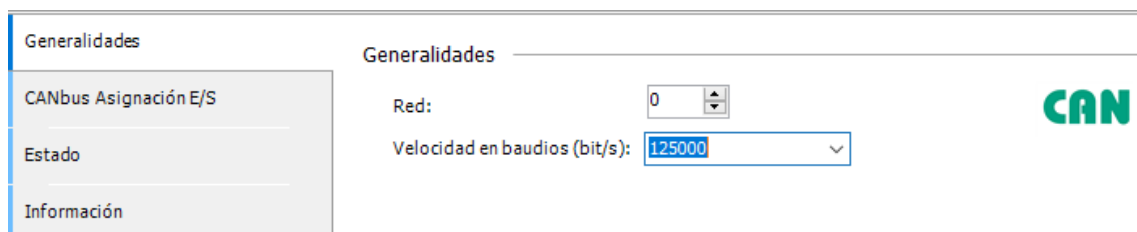


Figura 53: La ventana “Generalidades”del CANbus.

Luego, encima de “CANbus” clicaremos con el botón derecho y en “Agregar el dispositivo” elegiremos “CANopen\_manager”.



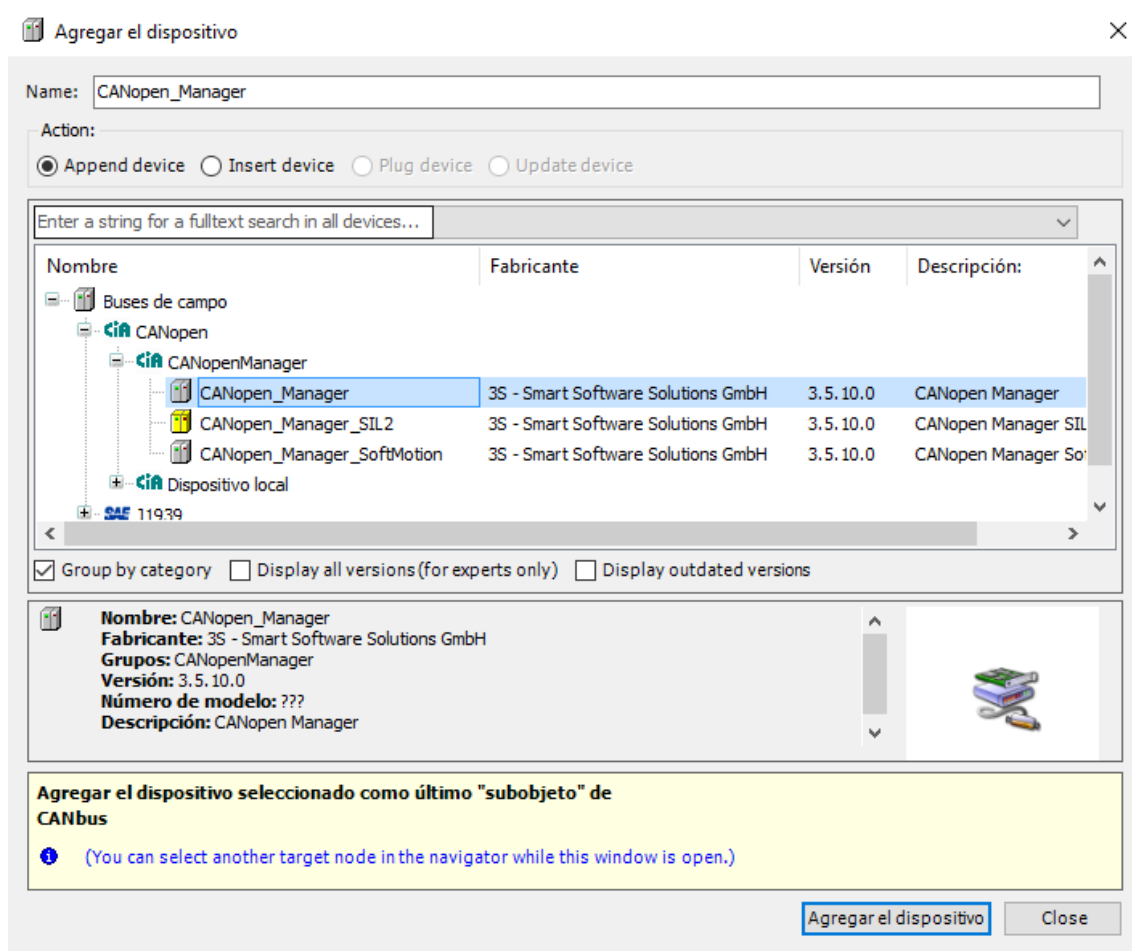


Figura 54: "Agregar el dispositivo" para instalar "CANopen\_Manager".

Dentro de "CANopen\_manager" en "Generalidades" hay que hacer un cambio. En la casilla de NMT hay que poner el "tick" como aparece en la siguiente foto:

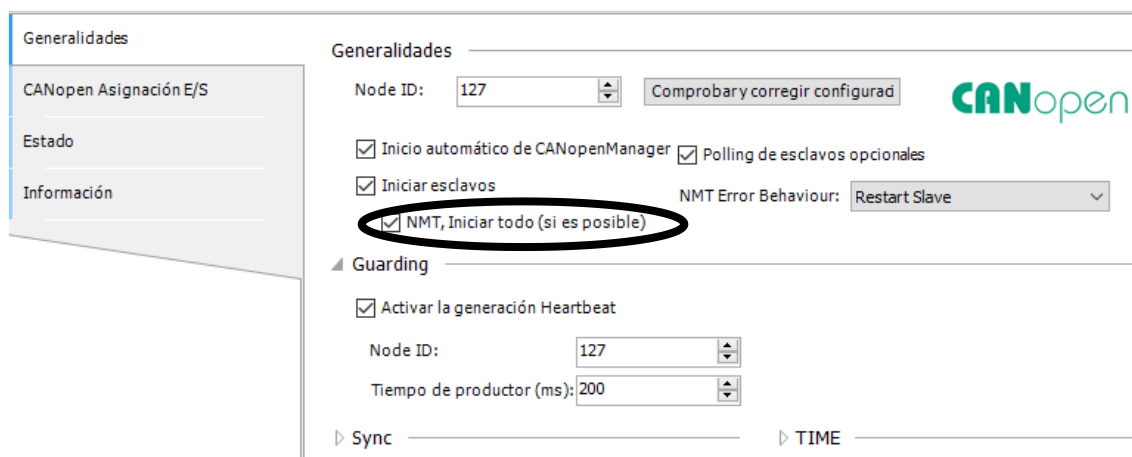


Figura 55: La ventana "Generalidades" de "CANopen\_Manager".

Después haremos "Iniciar la sesión" y encima de "CANopen\_Manager" con el botón derecho "Buscar dispositivo". Para ello, tendremos que instalar los .eds de nuestros

módulos desde la página de BECKHOFF y luego “Herramientas”→“Repositorio de dispositivos”→”Instalar” y así se quedarían instalados en CODESYS. Ahora podríamos buscar nuestros dispositivos.

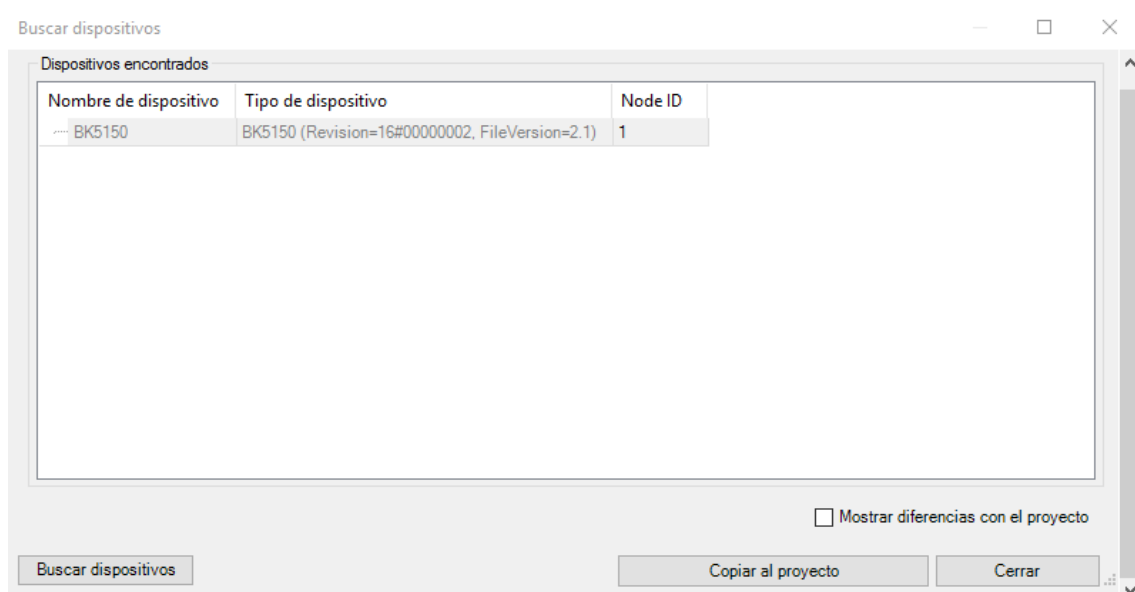
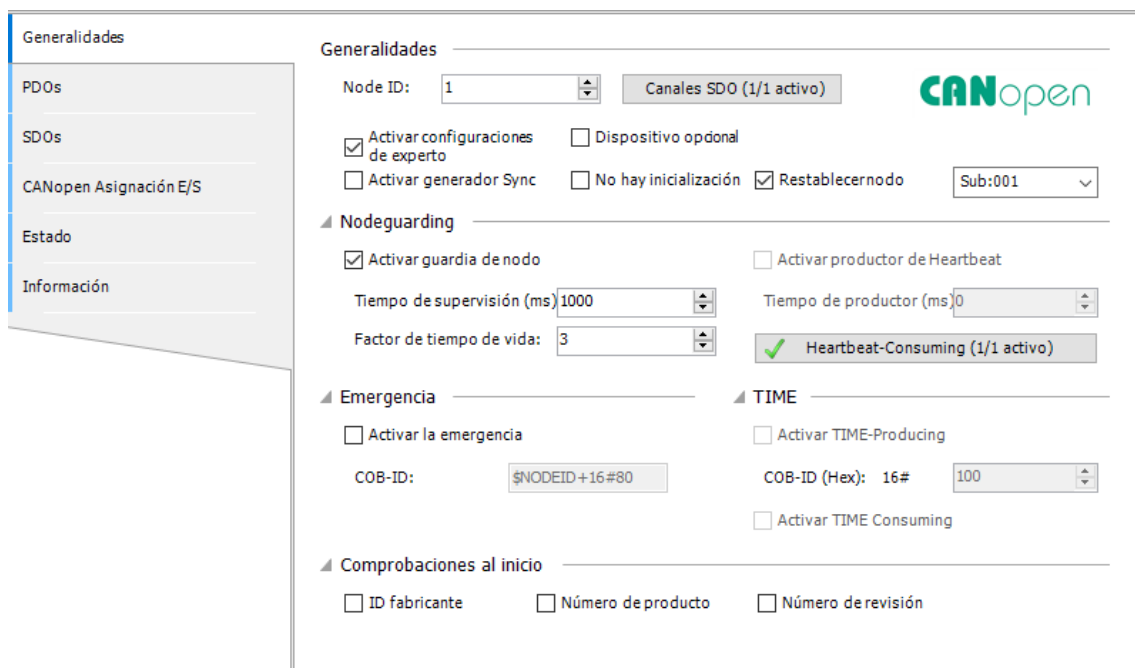


Figura 56: La ventana “Buscar dispositivos” de “CANopen\_Manager”.

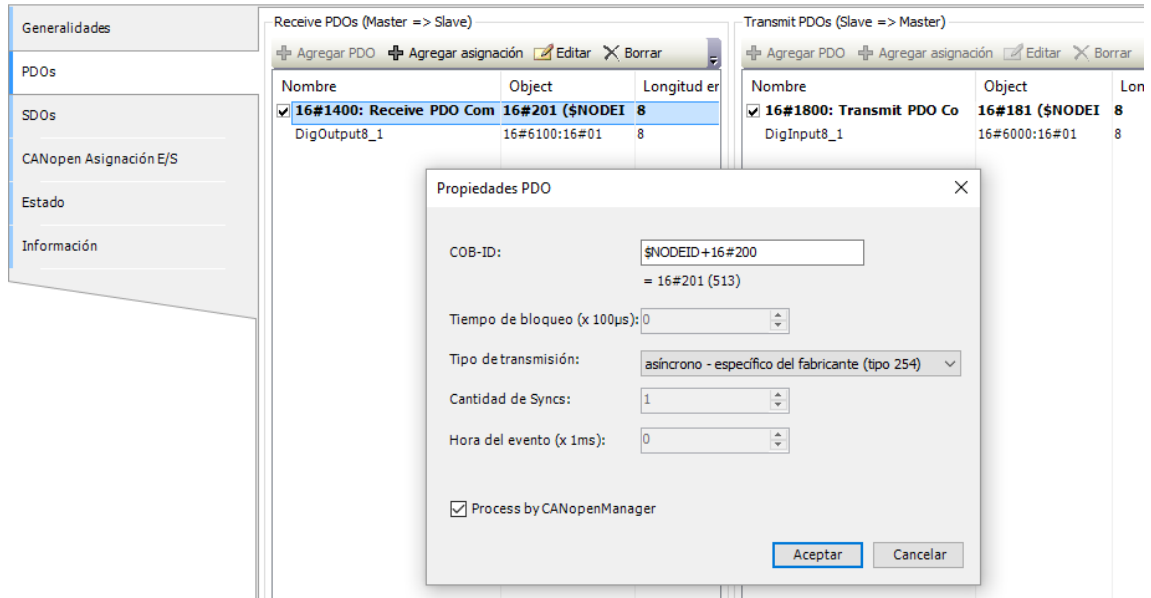
En “CANopen” solo aparece el nombre del módulo principal. Es decir, nosotros le tenemos que decir que módulos E/S tiene conectados. En este caso, solo trabajaremos con las E/S digitales.

Dentro de BK5150 en “Generalidades” hay que cambiar algunos datos y habría que ponerlas así:



Irudia 57: La ventana “Generalidades” de la “BK5150”.

En la siguiente sección, es decir en “PDOs”, nombraremos los módulos esclavos que tiene nuestro modulo principal.



Irudia 58: La ventana “Propiedades PDO” de “PDOs” de la “BK5150”.

Los de la izquierda son las salidas, los de la derecha en cambio, entradas. Encima de 16#1400 y 16#1800 clicaremos con el botón de la derecha para editar y pondremos “Tipo de transmisión”→”tipo 254”.En este caso los datos no se pueden mandar de modo síncrono entonces hay q poner de modo asíncrono.

Después de configurar los módulos esclavos, haremos “Iniciar la sesión” y veremos si todos están en verde:

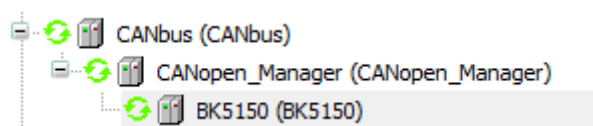


Figura 59: En “Dispositivos” CANopen conectado.

Para finalizar, asignaremos los pines que nosotros queramos.

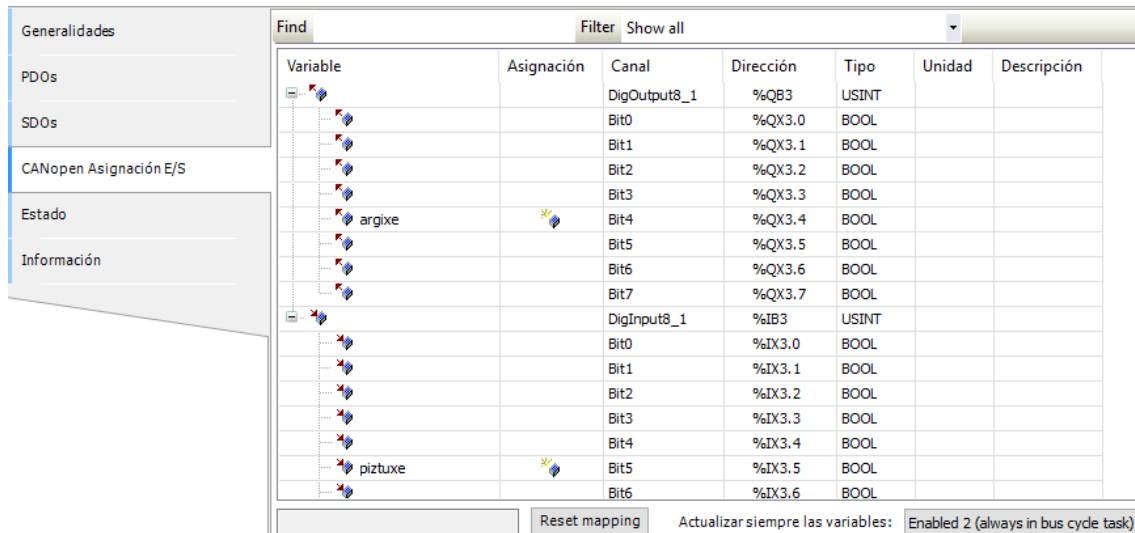


Figura 60: La ventana “CANopen Asignación E/S” de la “BK5150”.

Ahora probaremos con las E/S analógicas:

KL3162 Entradas analógicas

KL4004 Salidas analógicas

Para ello, activaremos el “can0” que antes hemos configurado con los dos siguientes comandos:

```
ip link set can0 up type can bitrate 125000
```

```
ifconfig can0 up
```

Entonces, le pondremos la “CAN CAPE” y cambiaremos los módulos digitales por los analógicos. Ahora, hay que seguir los mismos pasos que con el digital, pero esta vez tendremos que cambiar el “Node ID” a 2 porque la 1 la hemos usado con los módulos digitales. Con esto, podríamos “Buscar dispositivos” y nos aparecería de la siguiente manera:

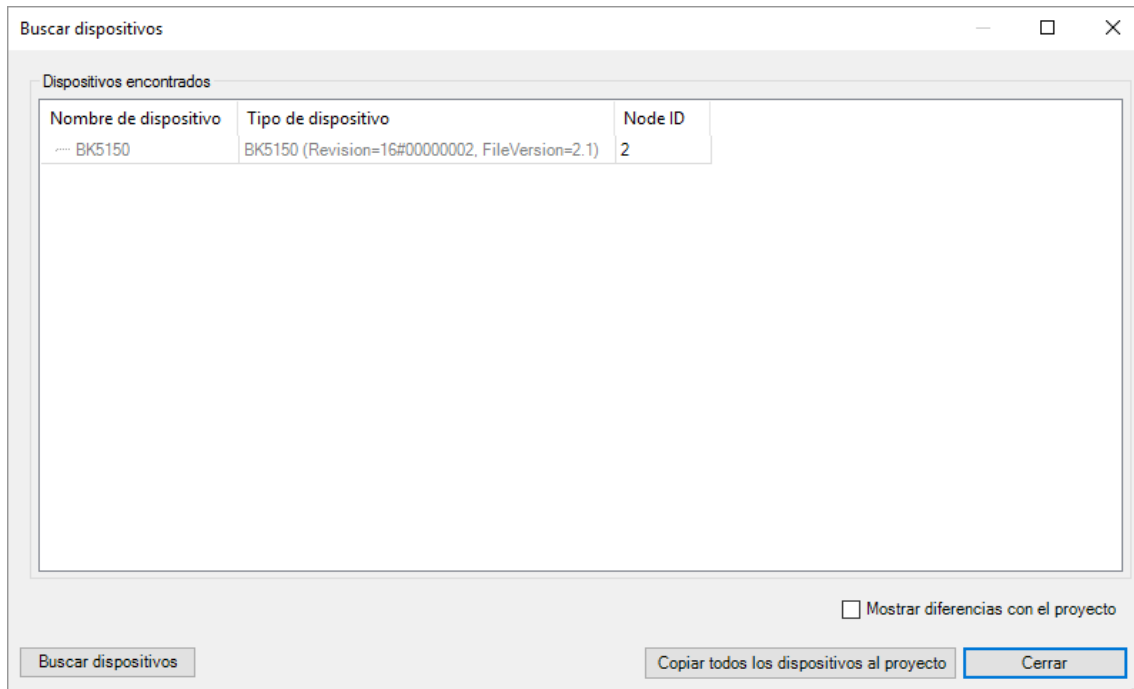
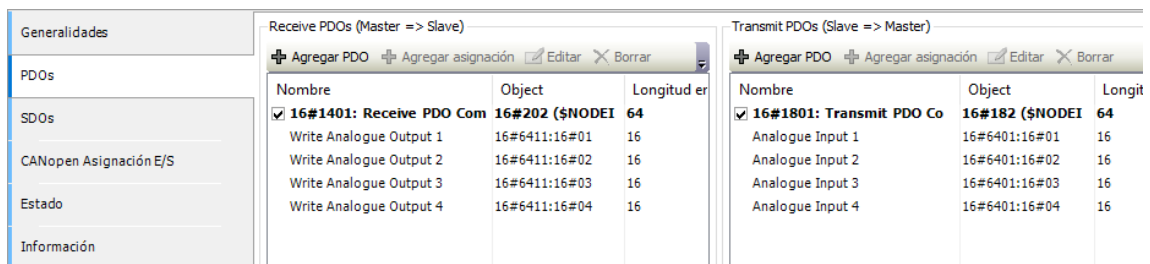


Figura 61: la ventana “Buscar el dispositivo” del CANopen.

En “Generalidades” y “CANopen asignación E/S” sería lo mismo que con el digital. Pero en “PDOs” en vez de digitales tendríamos que dejar solo los analógicos.

En este caso, solo tendríamos una entrada y una salida:



Irudia 62: La ventana “PDOs” de la “BK5150”.

Ahora, clicaremos encima de 16#1401 y 16#1801 con el botón derecho y le editaremos lo siguiente. En la salida (16#1401), pondremos “cíclico (tipo 1-240)” y en la entrada, (16#1801) “asíncrono (tipo 254)”.

Salidas:

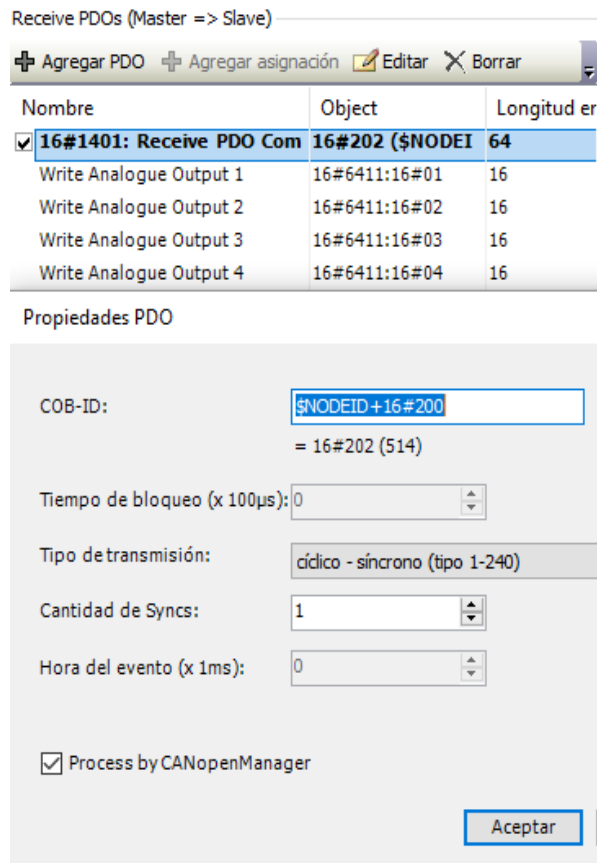


Figura 63: La ventana “Propiedades PDO” de “Receive PDOs”.

Entradas:

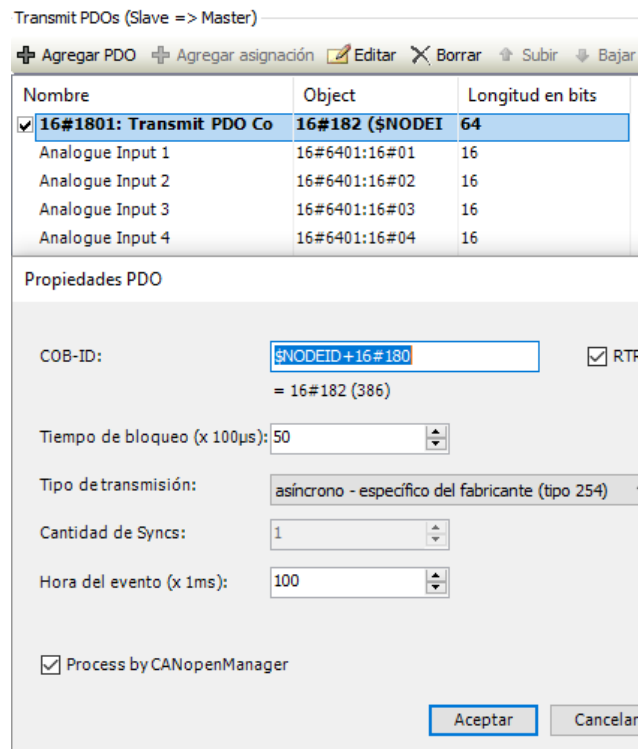


Figura 6427: La ventana “Propiedades PDO” de “Transmit PDOs”.

Para que haga de modo síncrono, en “CANopen\_manager” en “Generalidades” hay que poner “Cycle Period→100000” para que no mande rápido los datos y podamos ver sin errores. Es la siguiente la configuración:

Generalidades

Node ID: 127 Comprobar y corregir configurad **CANopen**

Inicio automático de CANopenManager  Polling de esclavos opcionales

Iniciar esclavos NMT Error Behaviour: Restart Slave

NMT, Iniciar todo (si es posible)

▲ Guarding

Activar la generación Heartbeat

Node ID: 127

Tiempo de productor (ms): 200

▲ Sync

Enable Sync Producing

COB-ID (Hex): 16# 80

Cycle Period (µs): 100000

Window Length (µs): 1200

Enable Sync Consuming

▲ TIME

Enable TIME Producing

COB-ID (Hex): 16# 100

Producer Time (ms): 1000

Figura 65: La ventana “Generalidades” de “CANopen\_Manager”.

Después de configurar todo, hay que nombrar las salidas y entradas que vayamos a usar:

| Variable | Asigna... | Canal                   | Direcci... | Tipo |
|----------|-----------|-------------------------|------------|------|
| irteera  |           | Write Analogue Output 1 | %QW2       | INT  |
|          |           | Write Analogue Output 2 | %QW3       | INT  |
|          |           | Write Analogue Output 3 | %QW4       | INT  |
|          |           | Write Analogue Output 4 | %QW5       | INT  |
| sarrera  |           | Analogue Input 1        | %IW2       | INT  |
|          |           | Analogue Input 2        | %IW3       | INT  |
|          |           | Analogue Input 3        | %IW4       | INT  |
|          |           | Analogue Input 4        | %IW5       | INT  |

Figura 66: La ventana “CANopen Asignación E/S” de la “BK5150”.