CP2000

Norbert

May 17, 2020

1 Bevezetés

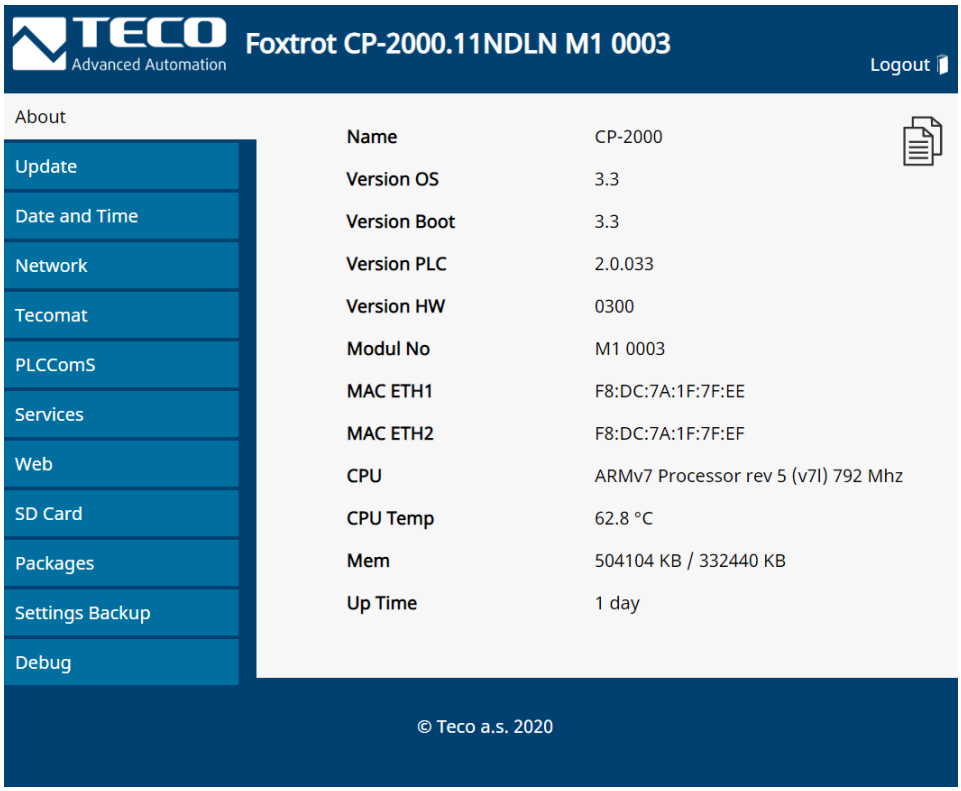
A riportban átbeszéljük a következő pontokat:

- Konfiguráció/Beállítások
- ModBus kommunikáció (Master/Slave)
- Apple Home Kit
- IP kamera

2 Konfiguráció és alap beállítások

A PLC alap konfigurációja eltér a régebbi CP-1XXX-es modellektől és az új modelleket most már egy webes felületen tudunk könnyedén konfigurálni.
A konfigurációs weboldalt a PLC IP címe alatt találunk amit a DHCP-szerver oszt ki és a 8080 vagy 8443-as porton találunk.
Ha teljesen új készülékről van szó, az ETH1 interfész a gyártótól kapott IP címet használja ami `http://192.168.134.176:8080` vagy `https://192.168.134.176:8443` lehet.
Ahhoz hogy ezt a címet elérjük, számítógépünknek a PLC-vel azonos helyi IP tartományú hálózaton kell lennie. Abban az esetben ha a PLC egy DHCP szerver által kiosztott IP címet kapott akkor a PLC-n lévő gombok segítségével a PLC képernyőjén meg tekinthetjük IP címet.
A kijelzett címet beírjuk a böngészőbe és ezzel elérjük a PLC konfigurációs felületet:
Alap belépési adatok a készülék hátoldalán találhatóak:
username: Teco
password: 1234

A konfigurációs felületet a következőképpen is elérhetjük: `https://foxtrot.local:8443`. Ez azért előnyös mert így nem kell tudnunk/kijelezni a PLC IP-címet. **Ez a módszer kizárólag olyan számítógépeken működik ami támogatja az úgy nevezett Zero-configuration networking-et.**



A beépített konfigurációs felületen a következőket találjuk:

- PLC firmware frissítése
- PLC idő és időzóna beállítása
- Automatikus idő szinkronizáció az NTP szerverrel
- log fájlok elérése (system start, system boot, system update)
- Hálózati interfész beállítások (ETH1, ETH2, WLAN1, WLAN2, LTE1)
- PLC paraméterek beállítása (EPSNET kommunikációs protokoll, PLC web szerver, TecoRoute szolgáltatás beállítási, PLC log fálj elérés)
- PLCComS szerver beállítások
- Szolgáltatás beállítások (FTP,Avahi, Samba)
- Web szerver beállítások (felhasználó és tanúsítvány beállítás
- SD Kártya (Egyes készülékekben ez a rész nem használható)
- Csomagok (Node.js, teco-notification és teco-homebridge) kezelése
- Ütemezett biztonsági mentes készítés

A PLC kezelőfelületére való belépés után, hozzáadhatunk és eltávolíthatunk különböző felhasználókat (Web / User fül alatt).
Ha elfelejtettük a belépési adatokat, azokat a PLC-n található gombok segítségével vissza állíthatjuk és belépés után, új felhasználó létrehozása szükséges.

3 Modbus kommunikacio (Master/Salve)

Modbus RTU egy nyílt soros protokoll amit a Modicon cég fejlesztett ki 1979-ben. Az Open Systems Interconnection (OSI) modell-ben a Modbus a 7. szinten van ami egy "Kliens/Szerver" kommunikációt biztosít két BUS-os egység között. A protokoll információt cserél egy időben a "master" és a "slave" állomás között ami soros adatátvitelen keresztül történik. Egyik állomás "utasítást" ad a másik "válaszol" rá.
Az Ethernet változatát, Modbus TCP-nek nevezik ami az 502-és portot használja. Az információ csere az eredeti Modbus RTU-n alapszik annyi különbséggel hogy az Ethernet változat több klienssel és szerverrel tud egyszerre kommunikálni. A *ModbusRTUlib* csomag kommunikációs lehetőséget biztosít mind a két módon, "Master" és "Slave".

3.1 Modbus master function block

A *ModbusRTUlib* könyvtár a következő function blokkokat és function-ket tartalmazza:

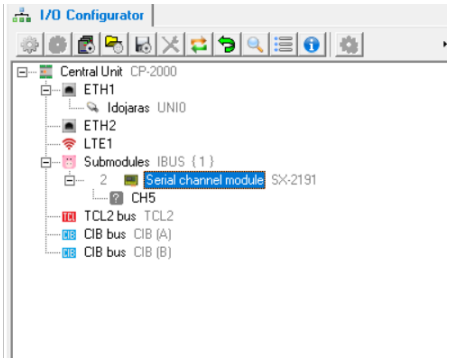
- ModbusCmd Egy "utasítás" ad egy kommunikáció létrehozására.
- ModbusRTUmas Kommunikációt biztosít ModbusCmd használatával soros porton keresztül
- fbModbusRTUmas2 Késleltetett kommunikációt biztosít ModbusCmd használatával soros porton keresztül
- ModbusCmdTCP Egy "utasítás" ad egy Ethernetes kommunikáció létrehozására.
- ModbusTCPmas Kommunikációt biztosít ModbusCmd használatával Ethernet, TCP-n keresztül
- fbModbusTCPmas2 Késleltetett kommunikációt biztosít ModbusCmd használatával Ethernet, TCP-n keresztül
- fbModbusUDPmas Késleltetett kommunikációt biztosít ModbusCmd használatával Ethernet, UDP-n keresztül

Modbus mód és a könyvtár kizárólag olyan csatornákon használható amik UNI módba vannak rakva. *ModbusRTUmas* és *ModbusRTUmas2* soros csatornán használhatóak aminek a paraméteréri a soros csatornánál adhatóak meg (CH1_ uni,...,CH10_ uni).
ModbusTCPmas es *ModbusTCPmas2* különböző IP című "slave" készülékekre van. Minden egyes "slave" készülék címet egyesével kell beírnunk különböző *ModbusCmd* blokk-ba (Ahogy a példákban majd látni fogjuk).

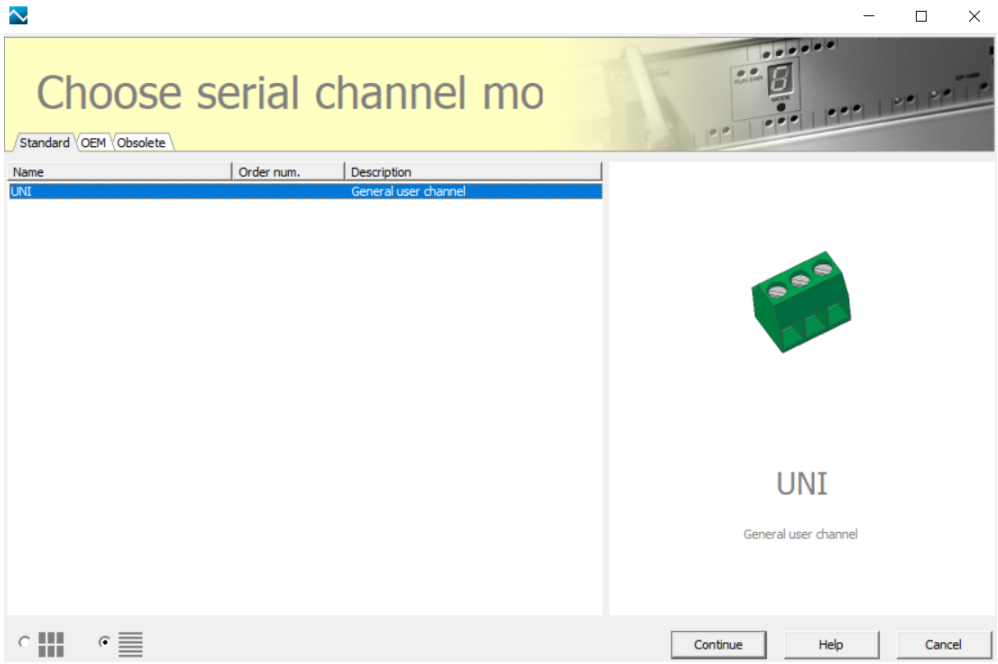
ModbusUDPmas különböző IP című kommunikációra használandó és hasonló az előzőleg megemlített *ModbusTCPmas* csak UDP protokoll-t használ.

UDP protokollon gyorsabb kommunikációt kapcsolatot érhetünk el.

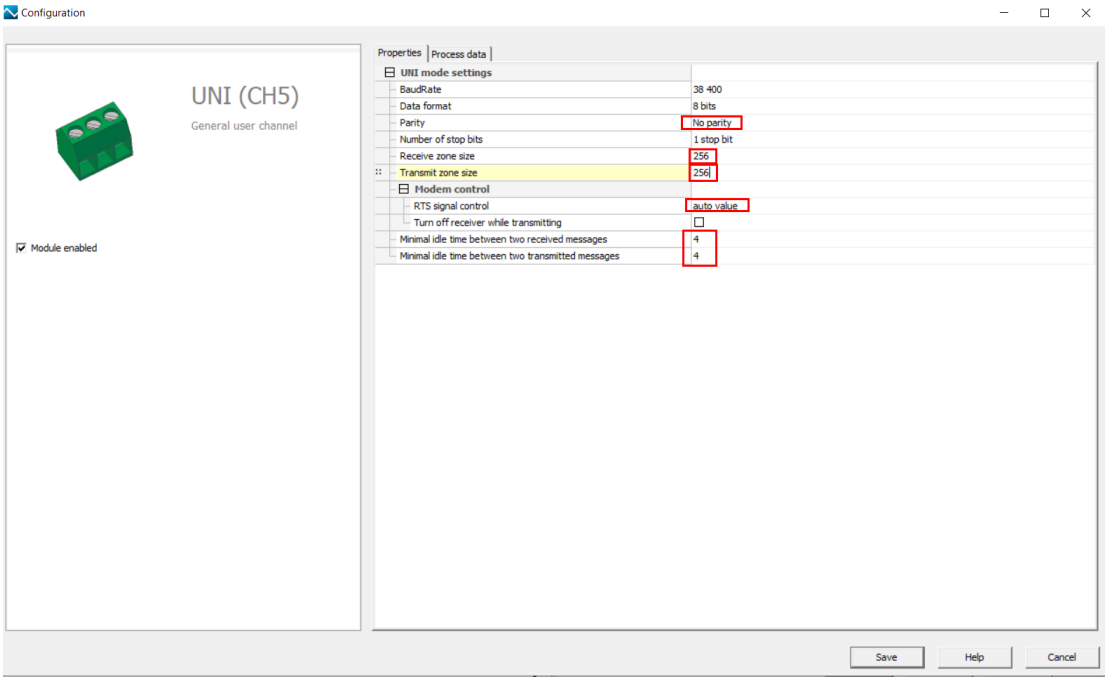
Ahhoz hogy a PLC készülékünk képes legyen az előzőleg megemlített soros kapcsolatra a következő konfiguráció szükséges:



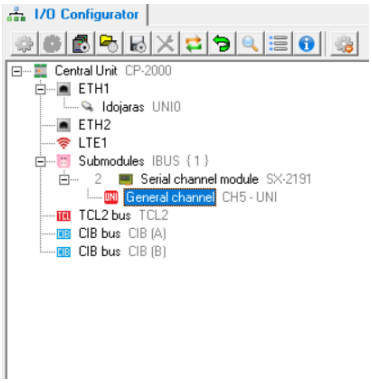
A Mosaic alkalmazásba az I/O Configurator-ba kétszer rákattintunk a "CH5" ami lehet "CH1 ... CH10" és a következő ablak megjelenik ahol kiválasztjuk a UNI csatornát majd a jobb also sarokba "Continue"



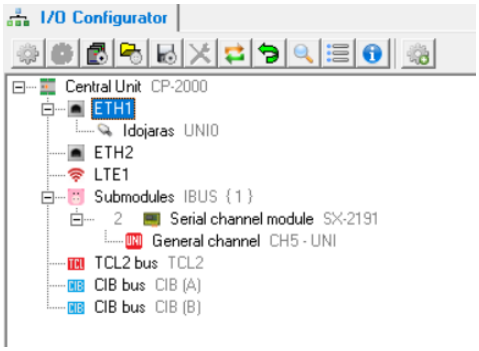
A pirosan bejelölt értékeket beírjuk majd a "Save"-el elmentjük.



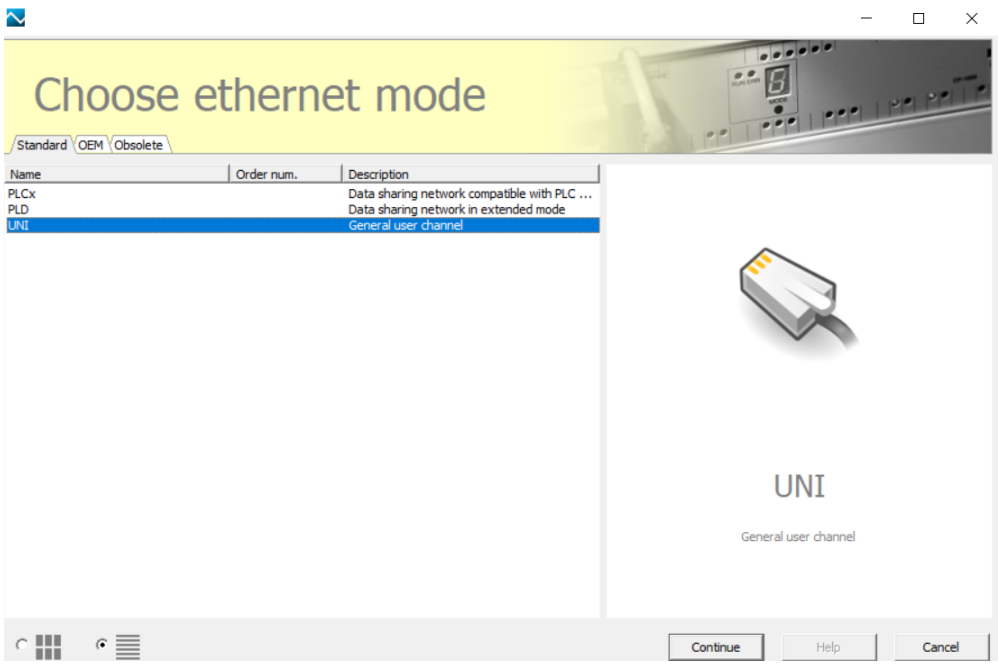
Sikeres beállítás után megjelenik a UNI kapcsolatunk az I/O Configurator-ba



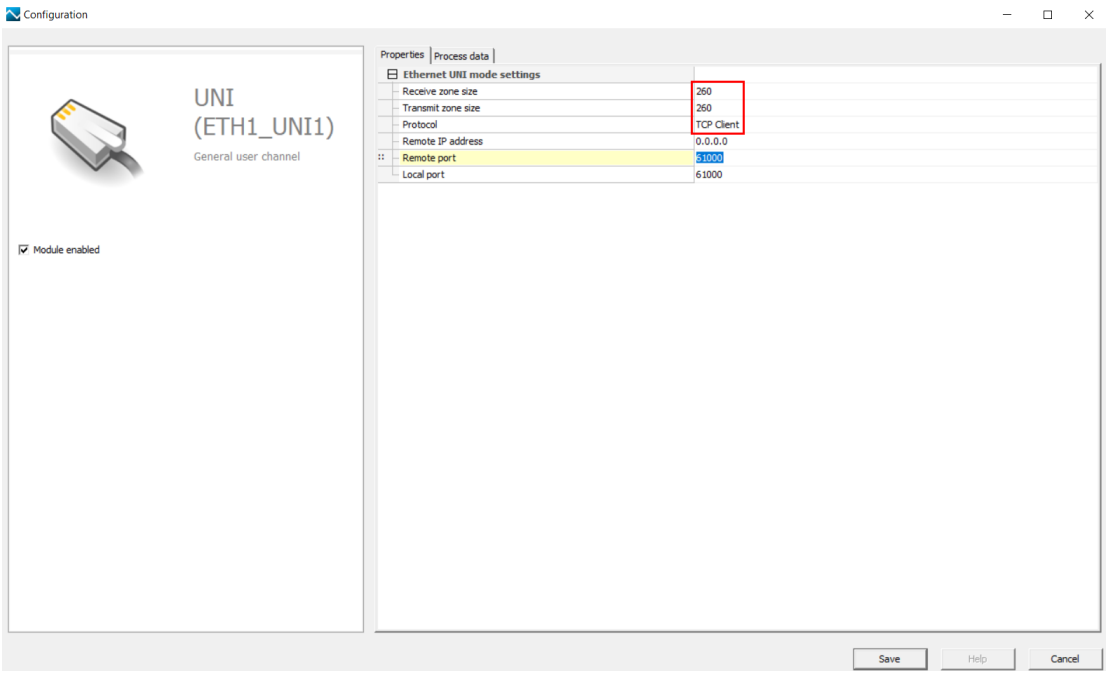
Ha TCP kapcsolatot szeretnénk használni akkor a következő konfiguráció szükséges:



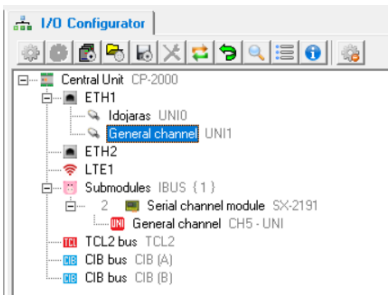
A Mosaic alkalmazásba az I/O Configurator-ba kétszer rákattintunk az ETH1 -re és hozzáadunk egy UNI kapcsolatot:



A pirosan bejelölt értékeket beírjuk majd a ”Save”-el elmentjük.



Sikeres beállítás után megjelenik az ETH UNI kapcsolatunk az I/O Configurator-ba:



UDP használata esetén ismételjük meg az Ethernetes részt és a "Protocol"-nál válasszuk ki az UDP-t.

3.2 Slave

Ha a készüléket "slave"-ként szeretnénk használni akkor az előző konfigurációk használhatóak, csak egy pár érteket kell átírni:

Soros kapcsolatnál:

- Parity: **even parity**
- Transmit zone size: **255**

Ethernet kapcsolatnál:

- Transmit zone size: **259**
- Receive zone size: **259**
- Protocol: **TCP Server**

3.3 TCP kapcsolat példa

A következő példa egy egyszerű ModBus TCP amit periódusosan 6 WORDs olvas be a ”slave” egységről, Ethernet-en keresztül:

```
PROGRAM prgMain
 VAR
 LastError : STRING;
 MdbError : STRING;
 MdbData : ARRAY [0..5] OF UINT;
 Command : ARRAY [1..1] OF TCmdStructTCP;
 ModbusMas : ModbusTCPmas;
 END_VAR
 ModbusCmdTCP(Gr := 1, IP := '192.168.134.41', UnitID := 0, FNC :=
    03,
 StAdr := 0, NoPoint := 6, PtrData := ADR(MdbData),
 Cmd := Command[1]);
 ModbusMas(EN := 1, GrSel := 1,
 MaxCmd := sizeof(Command)/sizeof(TCmdStructTCP),
 chanCode := ETH1_uni0, Commands := Command[1]);
 IF GetModbusErrTxt(ErrorCode := ModbusMas.ErrCode, ErrTxt :=
    MdbError) THEN
 LastError := MdbError;
 END_IF;
END_PROGRAM

  END_VAR
```

3.4 Komplex TCP kapcsolat példa

Egy komplex példa ST nyelvben írva, ahol 80 darab ”slave” egységtől fogunk információt kérni. A ”slave” egységek IP címei 192.168.2.101 és 192.168.2.181 között találhatóak. Sima CFC nyelvben is adaptálható a példa ami nagyon hasonlít az ExampleModbusTCP projektre ahol csak 10 darab készülékről kértünk információt.

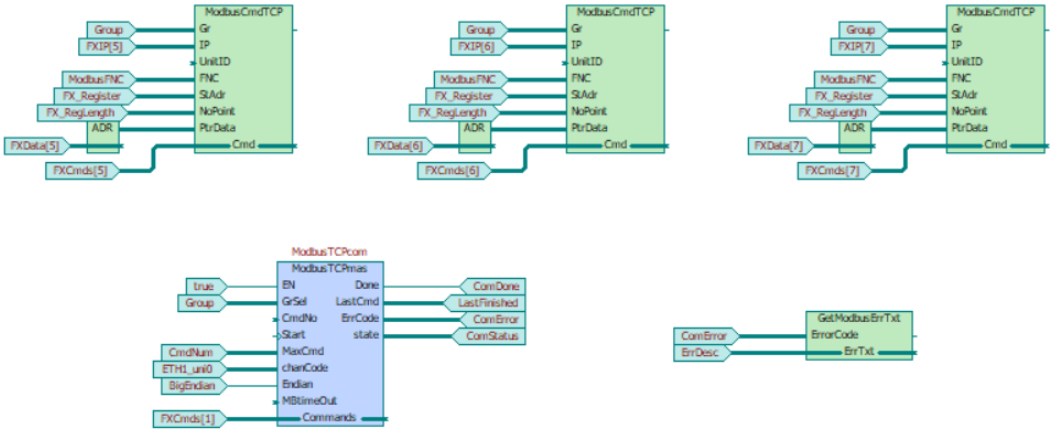
```
TYPE
 TDataBlock1 : STRUCT
 Temperature1 : REAL;
 Temperature2 : REAL;
 END_STRUCT;
END_TYPE
VAR_GLOBAL
 Temps : ARRAY [1..80] OF TDataBlock1;
 Control : ARRAY [1..80] OF BOOL;
 CmdTCP1 : ARRAY [1..160] OF TCmdStructTCP;
END_VAR
PROGRAM prgMain
 VAR
 i : UINT;
```

```
MDB_TCP : ModbusTCPmas;
GrSel : USINT := 1;
WriteControl : BOOL;
WriteControlR : R_TRIG;
DoneF : F_TRIG;
StartCommand : USINT;
StartErrCnt : UDINT;
ErrCnt : UDINT;
ErrorTxt : STRING;
LastErrorTxt : STRING;
END_VAR

IF ProgramIsChanged() THEN
FOR i := 1 TO 80 DO
ModbusCmdTCP(Gr:=1, FNC:=03, UnitID:=0,
IP:='192.168.2.'+UINT_TO_STRING(i+100),
StAdr:=0, NoPoint:=SIZEOF(TDataBlock1)/2,
PtrData:=adr(Temps[i]), Cmd:=CmdTCP1[i*2-1]);
END_FOR;
FOR i := 1 TO 80 DO
ModbusCmdTCP(Gr:=2, FNC:=05, UnitID:=0,
IP:='192.168.2.'+UINT_TO_STRING(i+100),
StAdr:=0, NoPoint:=1,
PtrData:=adr(Control[i]), Cmd:=CmdTCP1[i*2]);
END_FOR;
END_IF;

WriteControlR(CLK := WriteControl);
IF WriteControlR.Q THEN
GrSel := 2;
StartCommand := MDB_TCP.LastCmd;
StartErrCnt := ErrCnt;
END_IF;
MDB_TCP(EN := true, GrSel := GrSel,
MaxCmd := sizeof(CmdTCP1)/sizeof(TCmdStructTCP),
chanCode := ETH1_uni0, Commands := CmdTCP1[1]);
IF GetModbusErrTxt(ErrorCode := MDB_TCP.ErrCode, ErrTxt :=
   ErrorTxt) THEN
LastErrorTxt := ErrorTxt;
ErrCnt := ErrCnt + 1;
END_IF;
DoneF(CLK := MDB_TCP.Done);
IF DoneF.Q AND StartCommand = MDB_TCP.LastCmd THEN
IF StartErrCnt = ErrCnt THEN
GrSel := 1;
WriteControl := false;
ELSE
StartErrCnt := ErrCnt;
END_IF;
END_IF;
END_PROGRAM
```

CFC példa 10 ”slave” készülék esetén:




```
TYPE
  T_FXData : STRUCT
    Par1 : INT;
    Par2 : INT;
    Par3 : INT;
    Par4 : INT;
    Par5 : INT;
    Par6 : INT;
    Par7 : INT;
    Par8 : INT;
    Par9 : INT;
    Par10 : INT;
  END_STRUCT;
END_TYPE
VAR_GLOBAL RETAIN
  FXIP : ARRAY [1..9] OF STRING [22] := ['192.168.33.171',
                                         '192.168.33.172',
                                         '192.168.33.173',
                                         '192.168.33.174',
                                         '192.168.33.175',
                                         '192.168.33.176',
                                         '192.168.33.177',
                                         '192.168.33.178',
                                         '192.168.33.179'];
END_VAR

VAR_GLOBAL
  FXData : ARRAY [1..9] OF T_FXData;
  FXCmds : ARRAY [1..9] OF TCmdStructTCP;
END_VAR
```

4 Apple Home Kit

A következő rövid részben létrehozunk egy kapcsolat az Apple Home-kit (kizárólag iOS / macOS készülékeken fut) és a CP2000-es készülék között amivel egy egyszerű lámpát fogunk kezelni.
A CP2000-es készülékbe beépített PLCComS szerver-t fogjuk használni amit utolsóként fogunk konfigurálni. A kommunikáció a következőképpen zajlik:



4.1 Mosaic

A Mosaic alkalmazásban, hozzunk létre egy új project-et és a beépített könyvtárak segítségével rakjunk össze egy lámpa kapcsolót.

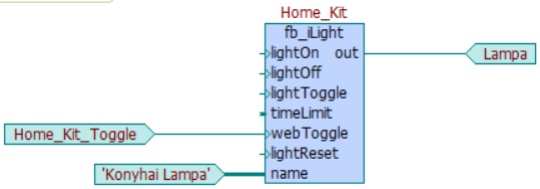


Figure 1: Blokk

Itt a Home_ Kit_ Toggle lesz a bemeneti változónk amit egy opcionális webes felületen tudunk kezelni. A kimeneti változó, "Lampa" egyértelműen a lámpára utal aminek értéke 0 (Ki) vagy 1 (Be) lehet. Amikor az fb_ iLight blokkot behúzzunk a programba, fontos hogy jelöljük meg publikusra.
Bármilyen támogatott blokkot (iControlLib könyvtár) amit szeretnénk az Apple Home Kit alkalmazásba látni, publikusra kell tenni és egy nevet adni neki amit majd az Apple Home Kit-ben fogunk látni.

```
FUNCTION_BLOCK Simple_Lamp
  VAR_INPUT
  Home_Kit_Toggle {PUBLIC} : BOOL R_EDGE;
  END_VAR
  VAR_OUTPUT
  END_VAR
  VAR_IN_OUT
  END_VAR
  VAR
  Home_Kit {PUBLIC} : fb_iLight;
  Lampa   : BOOL;
  END_VAR
```

```
13    VAR_TEMP
14    END_VAR
```

Miután megcsináltuk a "Function Blokkot", behúzzuk a Fő Programba majd az F9 lenyomásával, összerakjuk. A kapcsolónk megvan amit az Apple Home Kit alkalmazás tud kezelni. A következő lépés a CP2000-es konfigurálása.
A project manager-be beállítjuk hogy a publikus értékeket, a mi helyzetünkbe fb_ iLight exportáljuk a PLC-re. Ez azért szükséges mert a PLCComS szerver beolvassa ezt a publikus értékeket és így tudja az Apple Home Kit alkalmazás kezelni.

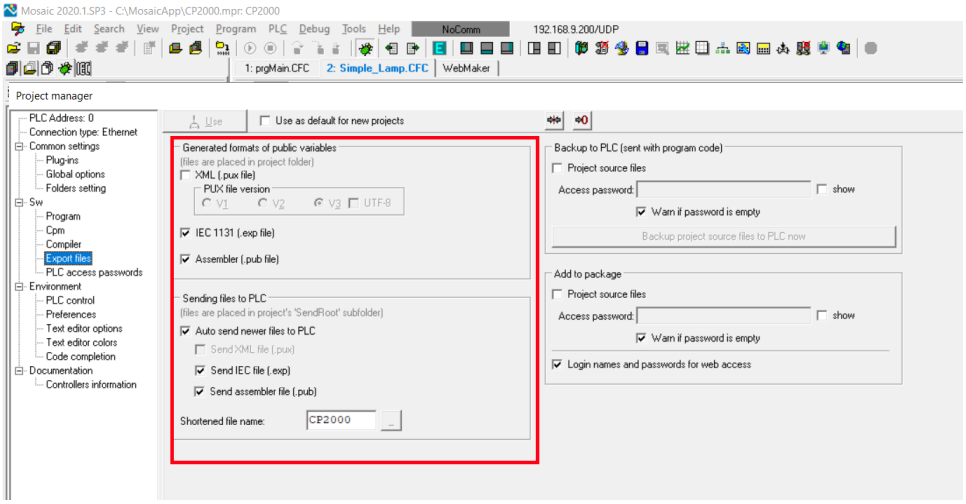


Figure 2: Project Manager konfiguráció

Project manager → SW → Export files és itt bejelöljük a beállításokat ahogy az ábrán latjuk. A "Shortened file name"-hez beírjuk a projekt nevet és megjegyezzük.

Érdekességképpen .pub file a következően néz ki, amibe látjuk a publikusra jelölt változót:

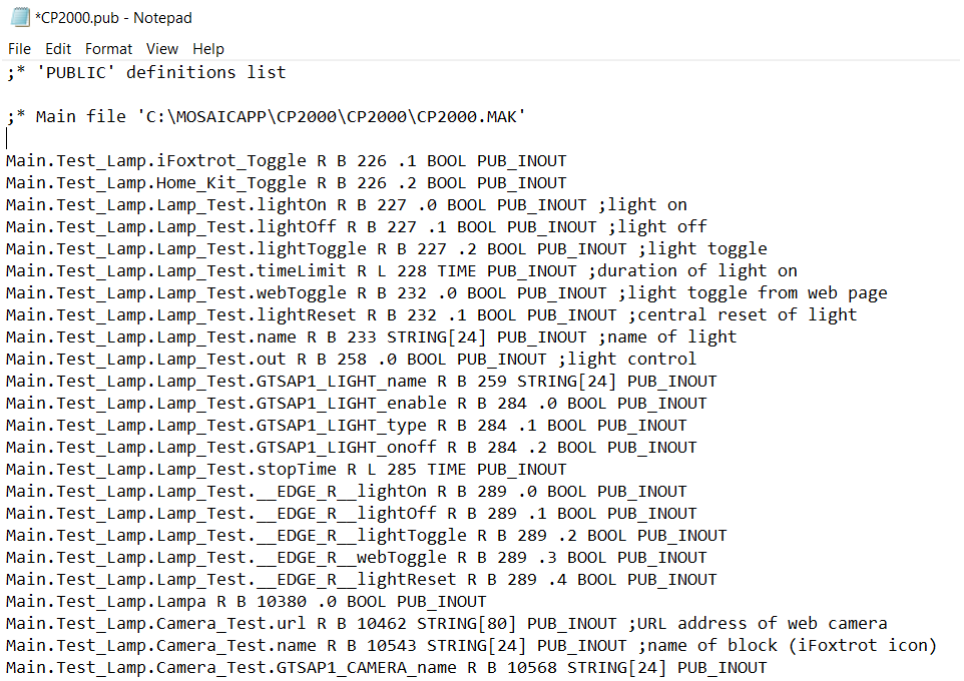


Figure 3: CP2000 pub fajl

Elmentjük a beállításokat, majd nyomjunk egy F9-et és töltsük fel a programot a PLC-re.
Következő teendőnk, a PLCComS konfigurálásra.
Nyissuk meg a kedvenc böngészőnket és lépjünk be a PLC kezelőfelületére amit a PLC_ IP:8443 alatt találnunk. Pl.: 192.168.9.200:8443. A belépési adatok a PLC hátoldalán találhatóak. Alaphelyzetbe a belépési adatok:
Username: Teco
Password: 1234
(Érdemes ezeket a PLC-n megváltoztatni a nagyon biztonság érdekében).
Miután beléptünk a kezelőfelületre, lépjünk a baloldalon található PLCComS menüpontra majd ”Servers”:

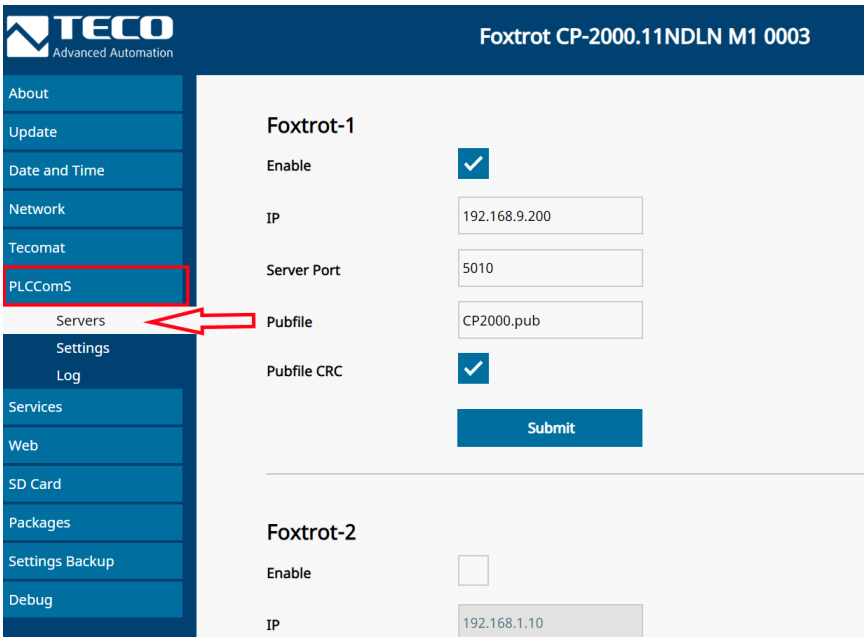


Figure 4: Tecomat kezelőfelület

Itt engedélyezzük a Foxtrot-1 -et és írjuk be a következő beállításokat:
IP: a PLC IP címe
Server Port: 5010
Pubfile: Az előzőleg exportál-t file neve (CP2000.pub)
Public CRC: Pipa
Miután beállítottuk a paramétereket, kattintsunk a “Submit” gombra.
Ezzel sikeresen elindítottunk egy PLCComS szerver-t a publikus értékünkkel.

4.2 Putty ellenőrzés

Ha szeretnénk letesztelni hogy a szerverünk valóban üzemel, Putty segítségével ellenőrizhetjük. Putty megnyitás után, beírjuk a következő paramétereket amivel elérjük a PLCComS szerverünket:
IP address: PLC IP cime
Port: 5010
Connection type: Telnet

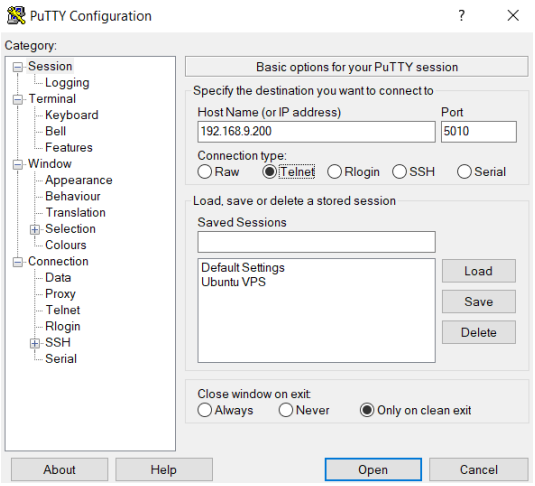


Figure 5: Putty kommunikáció indítása

Próbaképpen írjuk be a "GETINFO:" parancsot ami a következő választ adja:

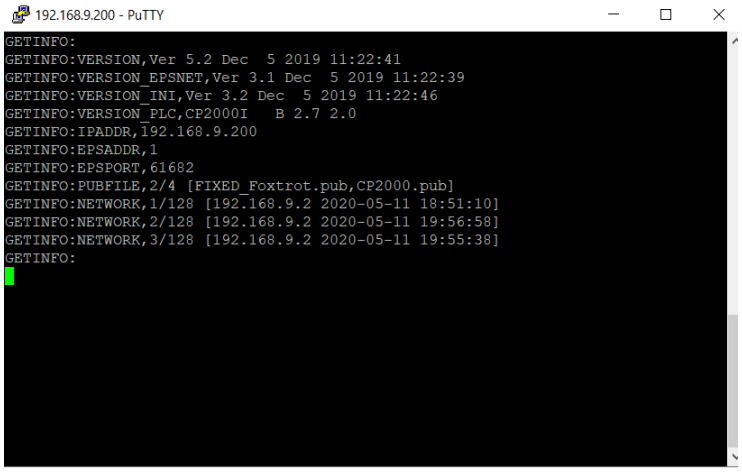


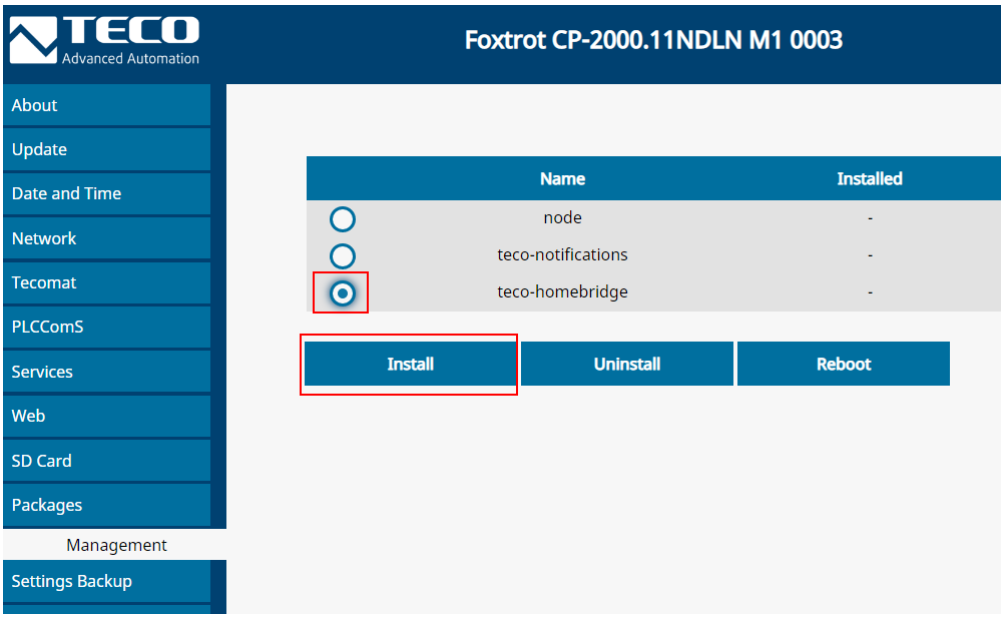
Figure 6: PLCComS teszt parancs

Itt látjuk hogy a "GETINFO:PUBFILE" válasz alatt ott van az előzőleg exportált és megnevezett CP2000.pub fájl.

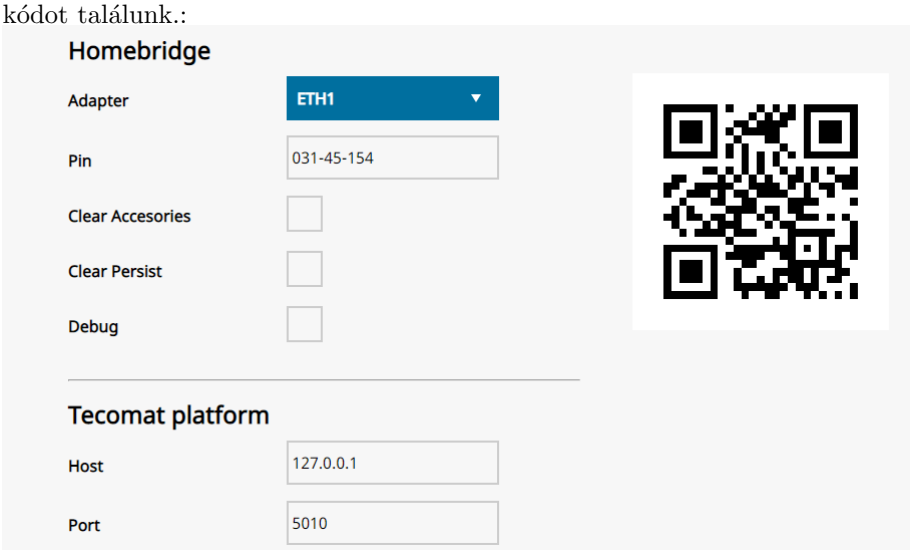
4.3 Apple Home Bridge telepites

Lépjünk vissza a PLC webes kezelőfelületére majd a bal oldali részen válasszuk ki a "Packages" menüpontot majd "Management".

Miután megnyílt az új felület, a középen található "teco-homebridget" válasszuk ki majd kattintsunk az "Install"-ra:



Telepítés után fontos hogy indítsuk újra a CP2000-t, amit simán a "Reboot" gomb meg nyomásával megteszünk.
Újraindítás után megjelenik egy új menüpont a baloldalon a "Packages" → "Management" fül alatt: "teco-homebridge".
Nyissuk meg a és a következő weboldalra érkezünk ahol egy QR kódot / pin kódot találunk.:



Az iOS készüléken nyissuk meg a Home Kit alkalmazást majd a QR kód beolvasóval és a Pin kód segítségével párosítsuk eszközeinket.

Sikeres párosítás után megjelenik a "Konyhai Lampa" a telefonon amit most már könnyedén tudunk kezelni.

4.4 Gyakori kérdések

A QR kod nem jelenik meg a Homebridge oldalon:

Ebben a helyzetben érdemes megnyitni a lap alján található hombridge.log fájl és megnézni mi történik. Általában a rossz konfiguráció esetén a következő hiba üzeneteket kapjuk:

```
2020-05-16 12:53:46.256595 Interface lo was found
2020-05-16 12:53:46.259218 The lo interface has no usable address
2020-05-16 12:53:51.302830 Interface lo was found
2020-05-16 12:53:51.308145 The lo interface has no usable address
2020-05-16 12:53:56.323023 Interface lo was found
2020-05-16 12:53:56.328089 The lo interface has no usable address
2020-05-16 12:54:01.342114 Interface lo was found
```

A homebridge résznél, a "Tecomat platform" alatt a "Host"-nal írjuk át a 127.0.0.1 címet a PLC IP címére: pl.: 192.168.9.200. Miután beírtuk a címet, kattintsunk a "Submit" gombra. és 1-2 perc elteltével frissítsük a weboldalt és a QR kódnak meg kell jelennie.
Érdekesség képen a homebridge.log fájlra kattintva az utolsó soroknak így kell kinézni:

```
{"q":"/opt/TecoSoft/DATA_PLC/PLC_0/ROOT/WWW/HOMEBRIDGE","a":"eth0","c":"031-45-154","i":"192.168.9.200","p":"5010","t":"2000","o'
Interface eth0 was found
Using eth0 interface with address 192.168.9.200
[2020-5-16 13:01:20] Loaded config.json with 0 accessories and 1 platforms.
[2020-5-16 13:01:20] ---
[2020-5-16 13:01:23] Loaded plugin: plugin
[2020-5-16 13:01:23] Registering platform 'homebridge-platform-tecomat.Tecomat'
[2020-5-16 13:01:23] ---
[2020-5-16 13:01:23] Loading 1 platforms...
[2020-5-16 13:01:23] [Foxtrot] Initializing Tecomat platform...
[2020-5-16 13:01:23] [Foxtrot] TecomatPlatform Init
Setup Payload:
X-HM://0023ISYWY5CKP
Enter this code with your HomeKit app on your iOS device to pair with Homebridge:

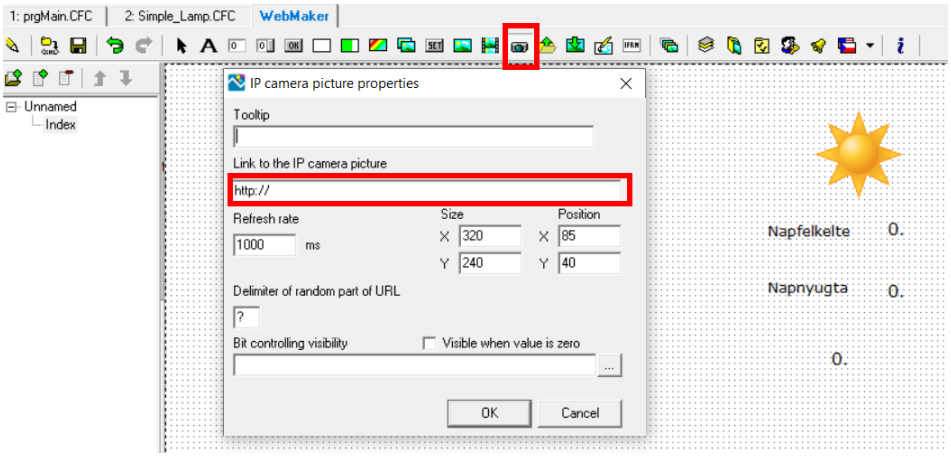
    031-45-154

[2020-5-16 13:01:28] [Foxtrot] DidFinishLaunching
[2020-5-16 13:01:29] Homebridge is running on port 51826.
[2020-5-16 13:01:31] [Foxtrot] PLC Communication server connection detected
[2020-5-16 13:01:31] [Foxtrot] Update accessories
[2020-5-16 13:01:31] [Foxtrot] Make accessories unreachable
[2020-5-16 13:01:31] [Foxtrot] Getting devices...
[2020-5-16 13:01:33] [Foxtrot] Generating accessories from devices...
[2020-5-16 13:01:34] [Foxtrot] New accessory will be registered (Konyhai Lampa [1ad60832-5f90-4a9b-87ce-1445e0776a8b]) :: LIGHT
[2020-5-16 13:01:34] [Foxtrot] New accessory will be registered (MAIN.TEST_LAMP.LAMP_TEST [288c7791-581c-4503-8541-137d66103dbc]
[2020-5-16 13:01:34] [Foxtrot] Registering accessories...
[2020-5-16 13:01:35] [Foxtrot] Setting up accessories...
[2020-5-16 13:01:35] [Foxtrot] Accessories update done
[2020-5-16 13:01:36] [Foxtrot] Konyhai Lampa ENABLE changed
[2020-5-16 13:01:36] [Foxtrot] MAIN.TEST_LAMP.LAMP_TEST ENABLE changed
[2020-5-16 13:01:40] [Foxtrot] onStart: stdout:
Running /mnt/root/opt/teco-homebridge/scripts/generate-hb-codes.js with arguments:
{"d":"/opt/teco-homebridge/.homebridge","o":"/opt/TecoSoft/DATA_PLC/PLC_0/ROOT/WWW/HOMEBRIDGE"}
Homebridge codes were generated to /opt/TecoSoft/DATA_PLC/PLC_0/ROOT/WWW/HOMEBRIDGE
```

A készülék nem jelenik meg a Home Kit alkalmazásban
Győződjünk meg hogy a Function Block "Publikus"-ra van téve a programunkba és elneveztük. Majd indítsuk újra a PLC-t.

5 IP Kamera

A CP2000-es készülék képes megjeleníteni az IP kamera képet, webes felületen. Fontos tudni hogy a PLC kizárólag ".JPEG" (HTTP) kamera képet tud fogadni és nem támogatja az RTSP protokollt. Ahhoz hogy hozzá adjuk a webes felülethez az IP kamera képet a következő a teendőnk.
A Mosaic alkalmazásban a meglévő projekthez a WebMaker-ben kiválasztjuk az "IP camera image" ikont és írjuk be az IP camera címet aminek .JPG ként kell végződnie:



Minden kamerához más HTTP-s elérés tartozik, amit érdemes a gyártó által kiadott leírásban egyeztetni.

Pl.:
Hikvision:
http://IP address of IPC:HTTP port/Streaming/channels/102/httpPreview
Netatmo Presence:
http://IP address of IPC:HTTP port/Unique Token/live/snapshot

Az Apple Home Kit alkalmazásban lévő megjelenéshez használjuk az iControlLib könyvtárban található *fb_ iWebCamera* blokkot amit publikusra kell tenni ahogy az előző fejezetben tettük:

```
Camera_Test {Public} : fb_iWebCamera;
```

