pbsBPI-6202

pbssoftLogic runtime for BPI-6202

Ver 1.0

Dec-2024

1. Installation

This document describes how to use pbsSoftLogic for the BananaPI BPI-6202 board.

pbsSoftLogic can use the following BPI-6202 resources :

- PWR LED
- Run LED
- Func LED
- Modem
- Watch Dog
- Serial Ports
- CAN IO Card with pbsCAN Protocols

After installing Ubuntu MATE on the board, you need to enable the ssh server.
Connect the BPI-6202 to the monitor and keyboard and turn it on.
The username and password are as follows:
Username = scada
Password = Szscada@123
Log in to Ubuntu and set up the network to connect to the internet.
You can change the root password. The default root password is bananapi.
Launch the Linux terminal and install the following software:

apt-get update
apt-get install openssh-server

edit /etc/ssh/sshd_config by nano utility and change PermitRootLogin to yes:
PermitRootLogin yes

Save the file and restart the BPI-6202.

You can connect to the BPI-6202 using an SSH client software such as Putty and Filezilla.

If you want to use Email Publishing Driver, SQLServer connection by TDS and MQTT in pbsSoftLogic, connect to BPI-6202 as root and install the following software, otherwise you do not need to install them.

for Email Publishing Driver install following modules on the BPI-6202 :
apt-get install curl
apt-get install python3-pycurl
apt-get install libcurl4-openssl-dev

for SQLite Driver install following modules on the BPI-6202:
apt-get install   freetds-dev
apt-get install   freetds-bin

for MQTT Driver install following modules on the BPI-6202:
apt-get install mosquitto-clients
apt-get install mosquitto-dev
apt-get install mosquitto
apt-get install libmosquitto-dev

Unzip the pbsSoftLogic_BPI6202.zip folder, which is the pbsSoftLogic runtime kernel for BPI-6202, and transfer it to the controller as follows:

Name	Date modified	Type	Size
etc	12/18/2024 10:54 AM	File folder	
home	12/18/2024 10:56 AM	File folder	
mnt	12/18/2024 10:54 AM	File folder	
readme.txt	12/20/2023 12:06 PM	TXT File	1 KB

Transfer content of the home folder to home folder of BPI-6202. Use filezilla Client for transferring to the controller.

When using filezilla, make sure the transfer type is set to binary, otherwise it will damage the transferred files. It is on Auto by default.
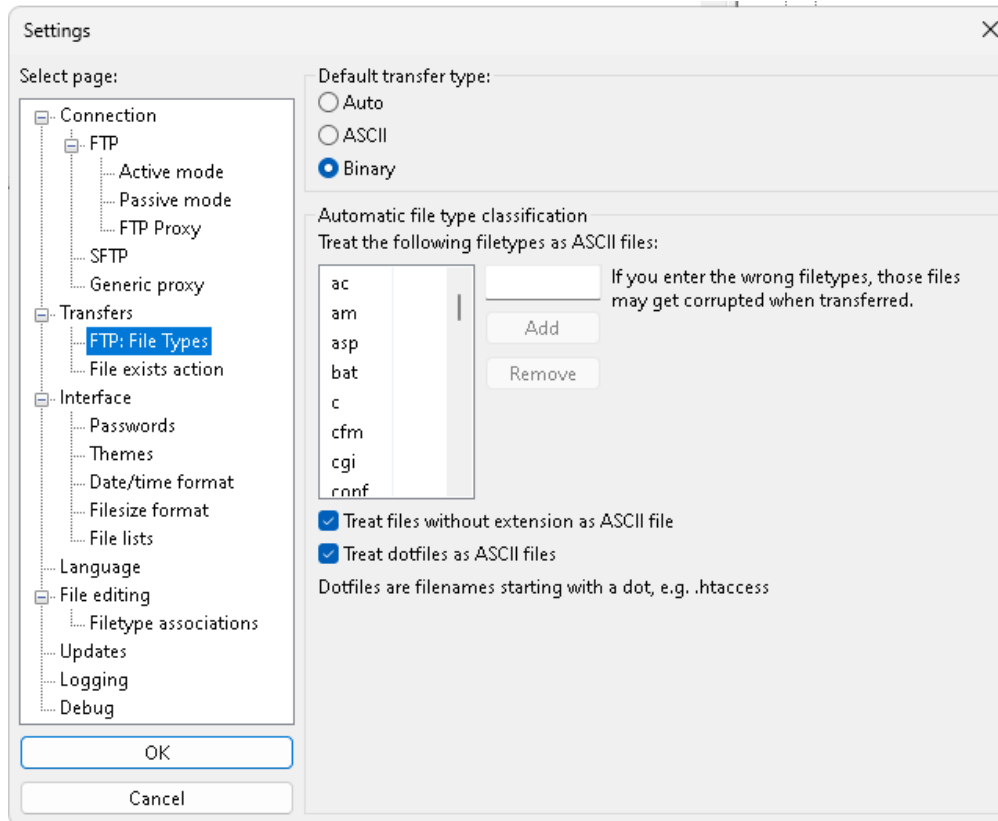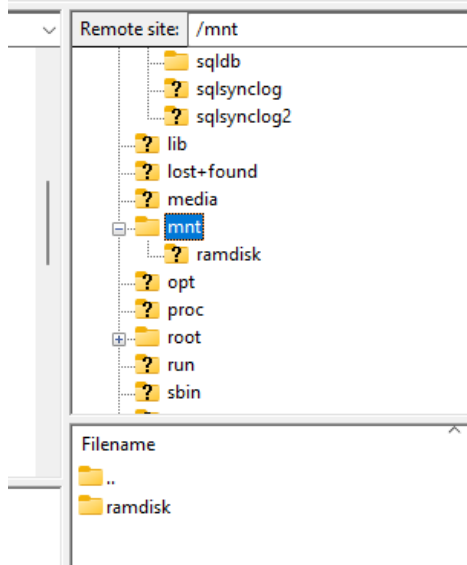

In the home folder, you can see the following folders:

The runtime core is the pbsLX folder. Other folders are for storing data if you are using some drivers.

For an explanation of the pbsLX folder, refer to the pbsSoftLogic user guide.

pbssoftLogic runtime requires ramdisk to run. So please create a folder in /mnt named ramdisk.

Using filezilla, edit the /etc/fstab file on BPI-6202 and add the following line to it:

```
LABEL=writable  /      ext4    defaults    0  0
tmpfs        /mnt/ramdisk      tmpfs      rw,size=10M      0  0
```

This command will convert the /mnt/ramdisk folder as a real ramdisk in Linux.

In the unzipped pbsSoftLogic_BPI6202.zip folder, you can see the etc/init.d folder. Copy the /etc/init.d/xpsle file to the same path on the BPI-6202.

Connect to the controller by putty utility by root user and execute following command :

chmod +x /etc/init.d/xpsle

ln -s /etc/init.d/xpsle /etc/rc5.d/S97xpsle

This command executes pbsSLKLX, which is the pbsSoftLogic runtime kernel, at boot time.

You can restart the BPI-6202. The pbsSoftLogic runtime is now ready to use.

2 – Programming

To use pbsSoftLogic IDE, please refer to pbsSoftLogic User Guide. In this section, we will explain the local IO for BPI-6202.
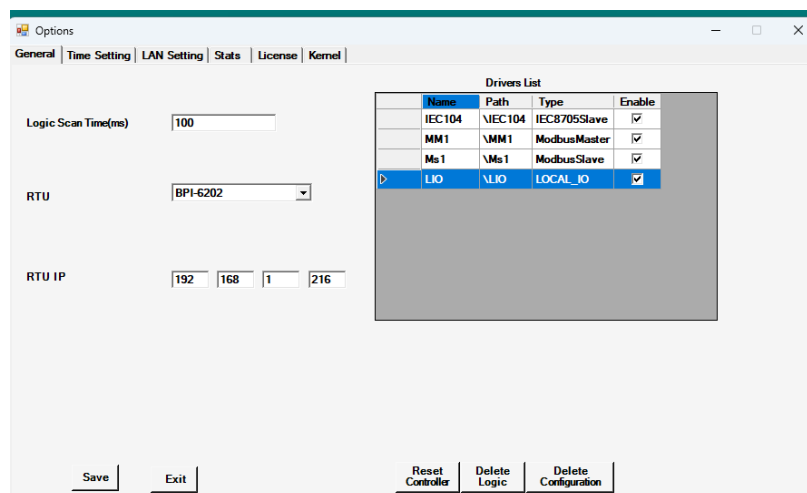
In pbsSoftLogic, Local IO is a driver that manages all the resources placed on the main CPU, such as LED, GPIO, Modem, Watch dog, etc.

Run pbsSoftLogic and create a new project and open the project settings.
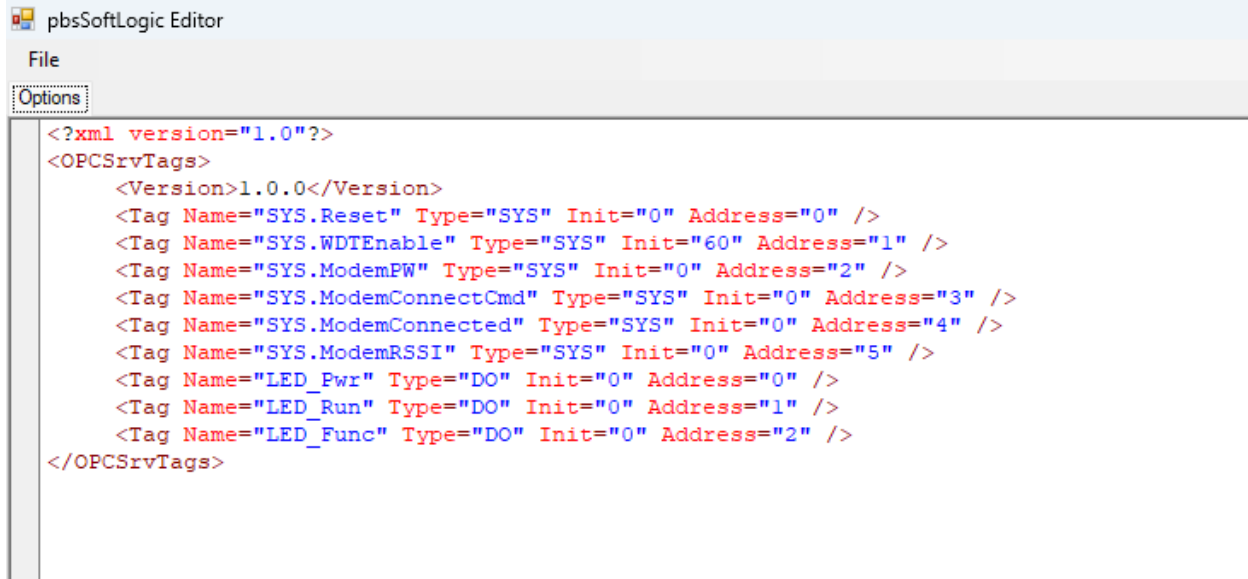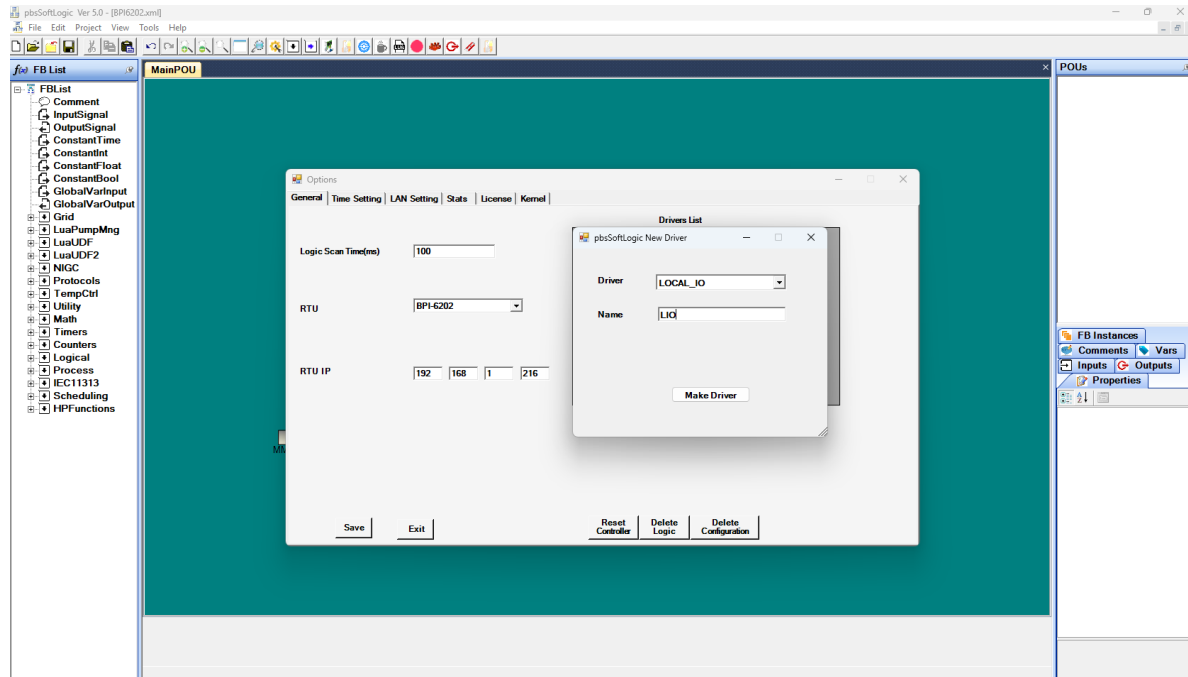

Select BPI_6202 as the RTU type and enter the controller IP address.

Right-click on the list of drivers and add a new Local_IO to the project and name it LIO.

Double-click on LIO Driver and you will see the following screen:

```xml
pbsSoftLogic Editor
File
Options
<?xml version="1.0"?>
<OPCSrvTags>
    <Version>1.0.0</Version>
    <Tag Name="SYS.Reset" Type="SYS" Init="0" Address="0" />
    <Tag Name="SYS.WDTEnable" Type="SYS" Init="60" Address="1" />
    <Tag Name="SYS.ModemPW" Type="SYS" Init="0" Address="2" />
    <Tag Name="SYS.ModemConnectCmd" Type="SYS" Init="0" Address="3" />
    <Tag Name="SYS.ModemConnected" Type="SYS" Init="0" Address="4" />
    <Tag Name="SYS.ModemRSSI" Type="SYS" Init="0" Address="5" />
    <Tag Name="LED_Pwr" Type="DO" Init="0" Address="0" />
    <Tag Name="LED_Run" Type="DO" Init="0" Address="1" />
    <Tag Name="LED_Func" Type="DO" Init="0" Address="2" />
</OPCSrvTags>
```

You can use the above tags in your logic:

SYS.Reset : when SYS.Reset set to 1 , for value more than watch dog time , will reset BPI-6202.

SYS.WDTEnable : if set to 0 , WDT is disabled , when set to 60 for defining 30 sec WDT . If you need more WDT time, increase this value.

SYS.ModemPW: if Set to 0 , Modem is disabled . if set to 1 , modem is enabled .

SYS.ModemConnectCMD : Changing from 0 to 1 turns the modem on and tries to connect to the mobile network. Changing from 1 to 0 turns the modem off.

SYS.ModemConnected: When it changes to 1, it indicates that the modem is properly connected to the network.

SYS.ModemRSSI : Shows Modem RSSI signal .

LED_Pwr: When you change to 1, the power LED turns on. When you change to 0, the power LED turns off. The power LED is a small LED on the board, not on the front LED.
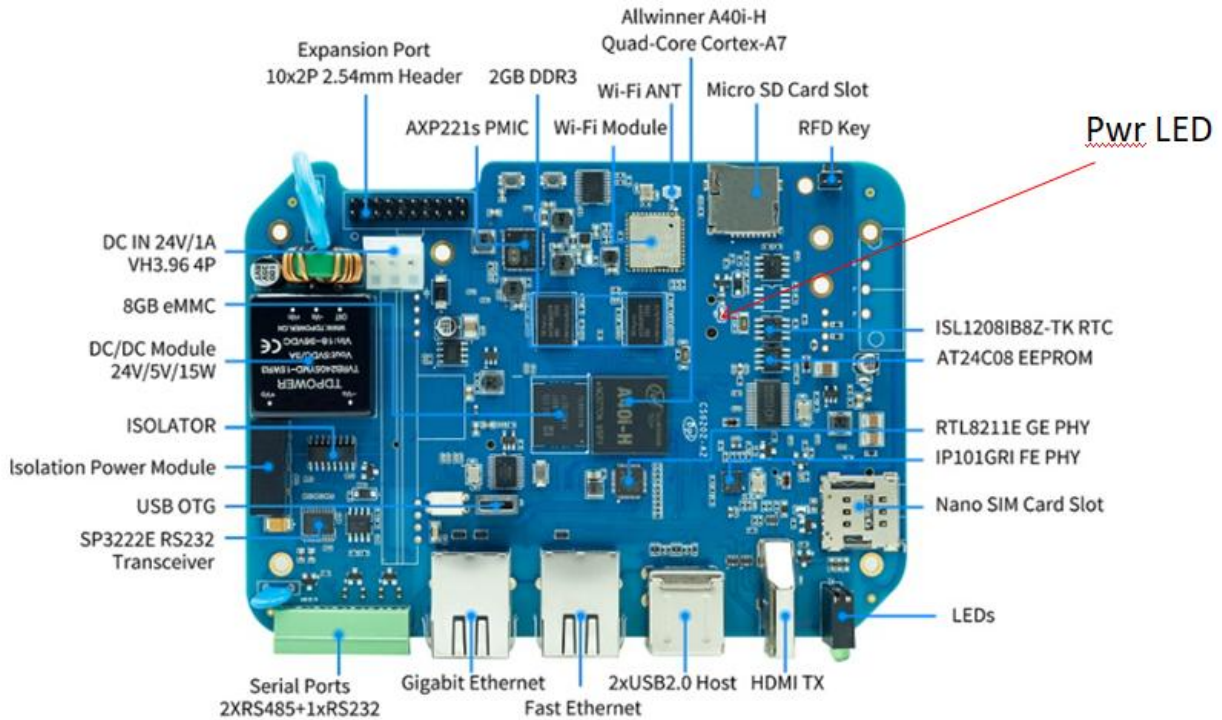
LED_Run: When you change to 1, the Run LED turns on. When you change to 0, the Run LED turns off. The Run LED is on the front side.

LED_Func: When you change to 1, the Func LED turns on. When you change to 0, the Func LED turns off. The Func LED is on the front side.

Now suppose you want to create a pulse generator and connect it to the Run LED. From the Timer Function Blocks, drag and drop a PulseGen function and place it on the MainPOU.
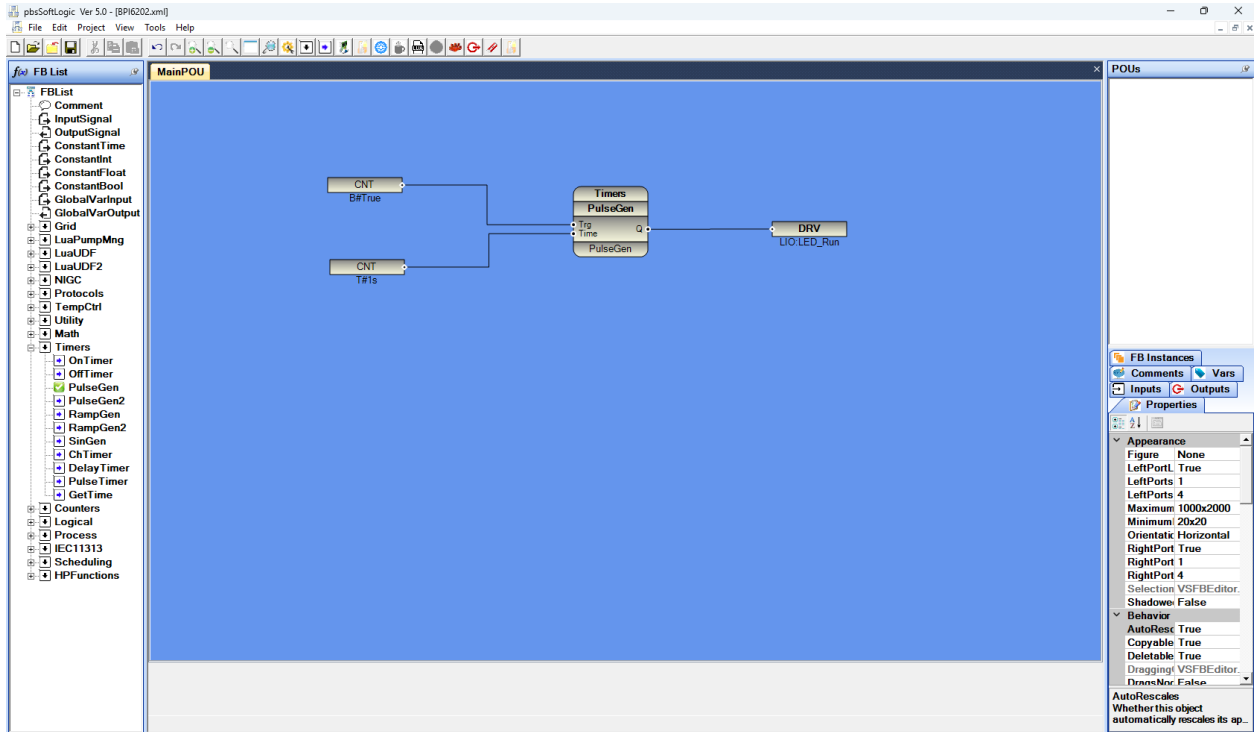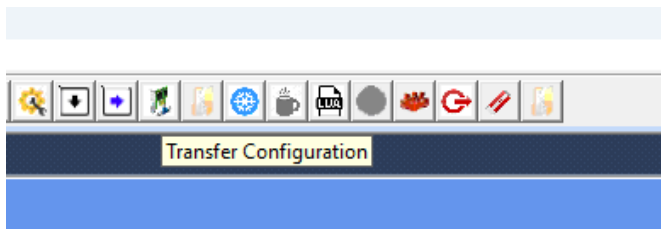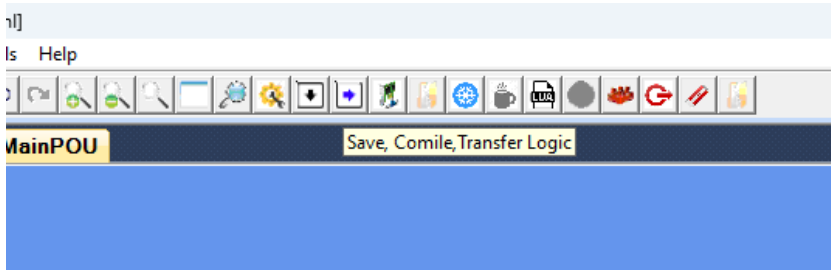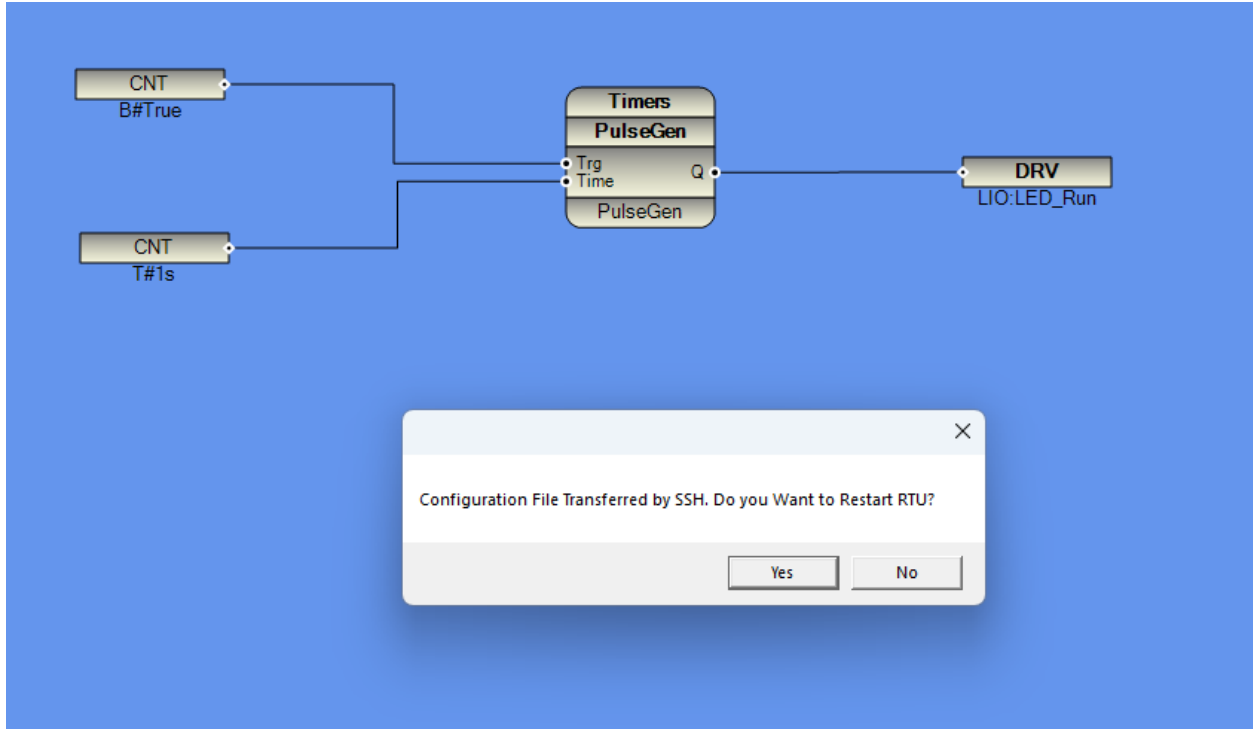

Drag a Constant Bool and connect to Trg input of PulseGen FB .

Drag a constant time and connect to Time input of Pulsegen FB.

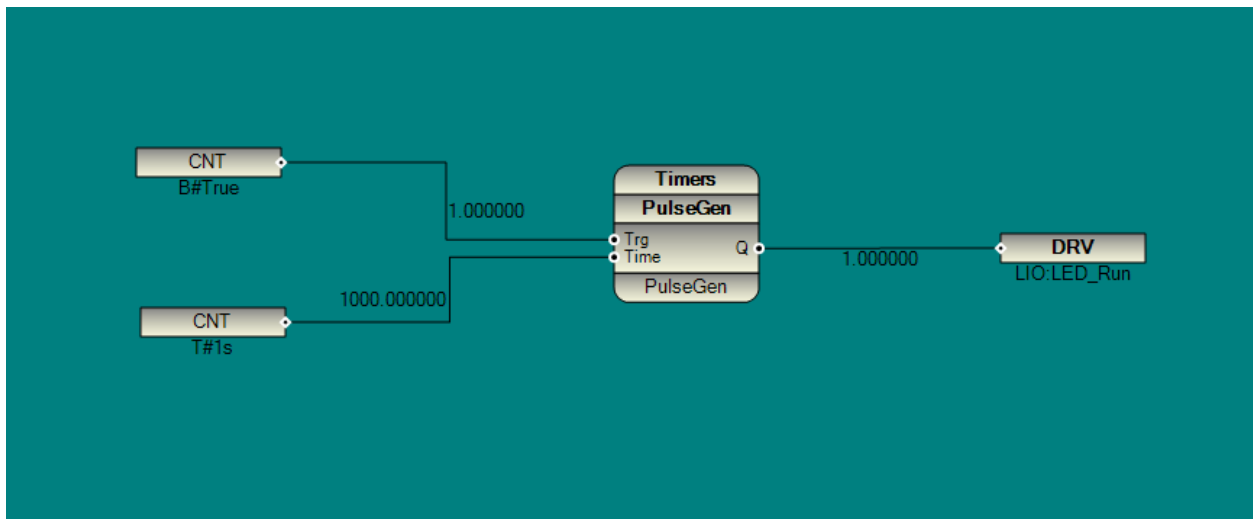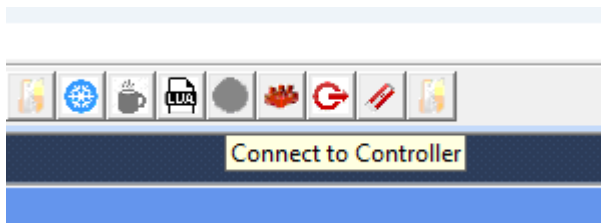Drag an OutputSignal and connect to Q Output of FB .

Save the project and transfer the Logic and Configuration to the BPI-6202 and restart it.

It takes 40 seconds for the BPI-6202 to boot and run the pbsSoftlogic runtime kernel. After that, you can connect to it and monitor your logic.

Click on connect to controller button and it will monitor your logic.

And the BPI-6202 RUN LED flashes on and off every second.

In pbsSoftLogic, the default password for root is root. But if you change it, you need to change it in the tools option menu.
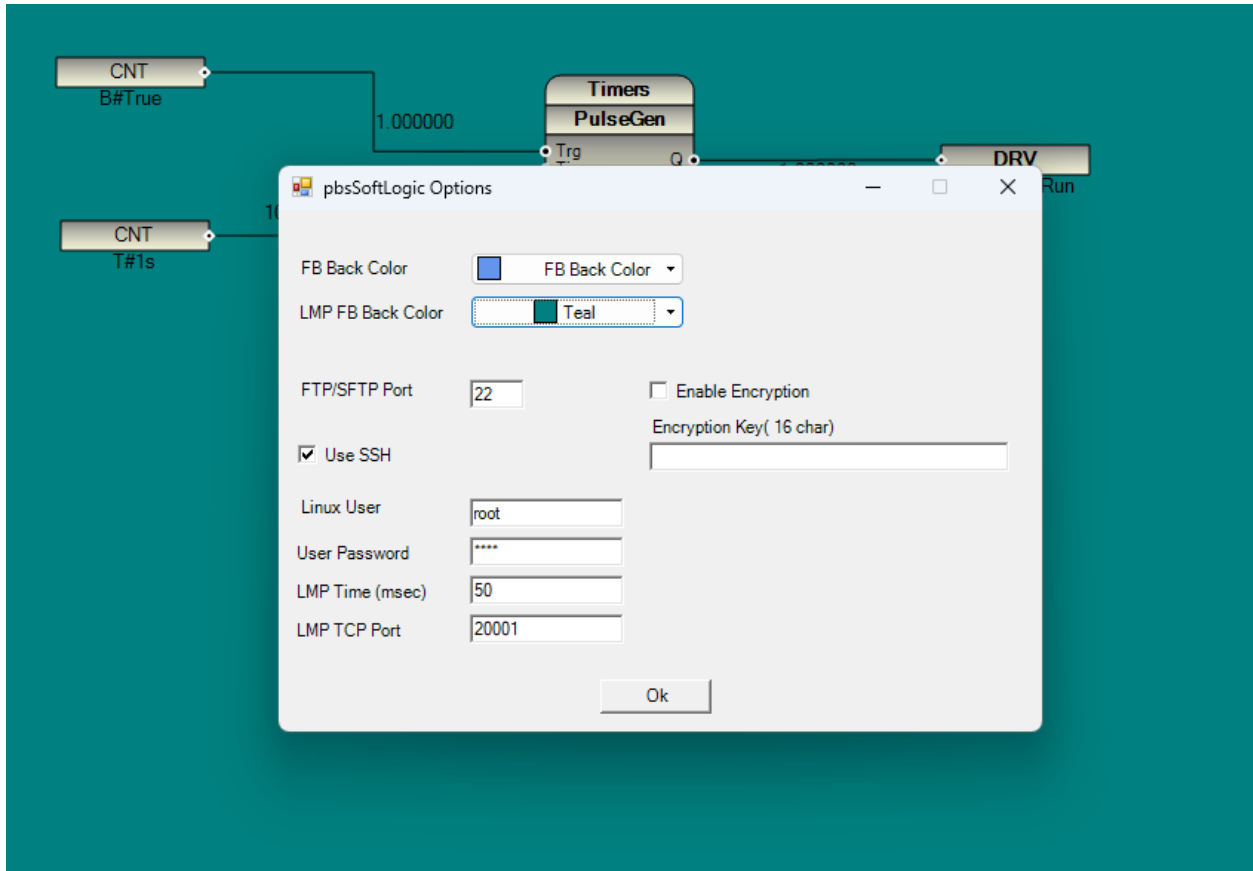

After changing the password, close and re-run pbsSoftLogic IDE.

3- Protocol Configuration

To use protocols such as Modbus, DNP3, IEC104, MQTT, ... you need to add the appropriate driver to your project. For detailed explanations, please refer to the pbsSoftLogic user guide. In This section shows the serial port settings.

In the image below you can see the location, pins and names of the BPI-6202 serial ports.
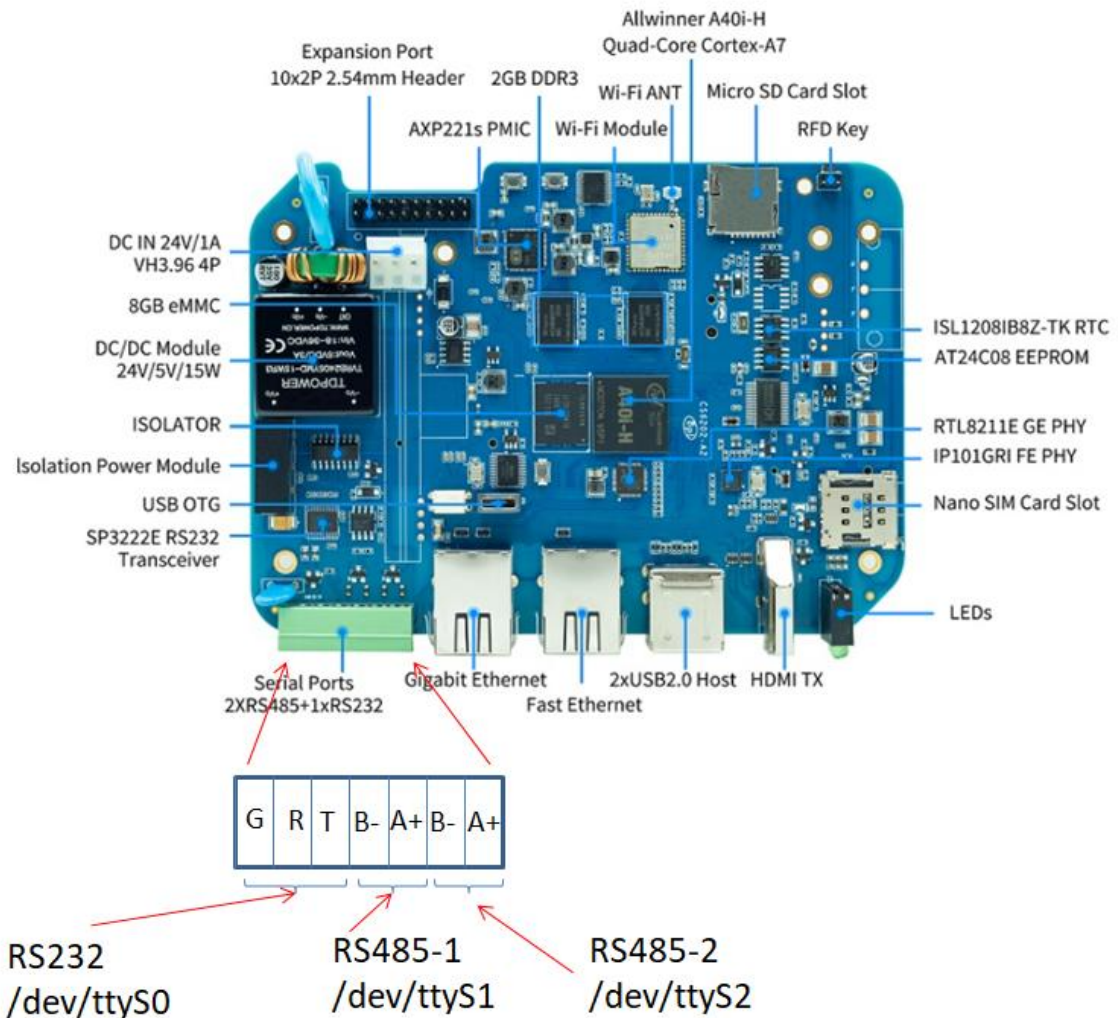

When you want to use serial ports in pbsSoftLogic, you need to use the serial port names.
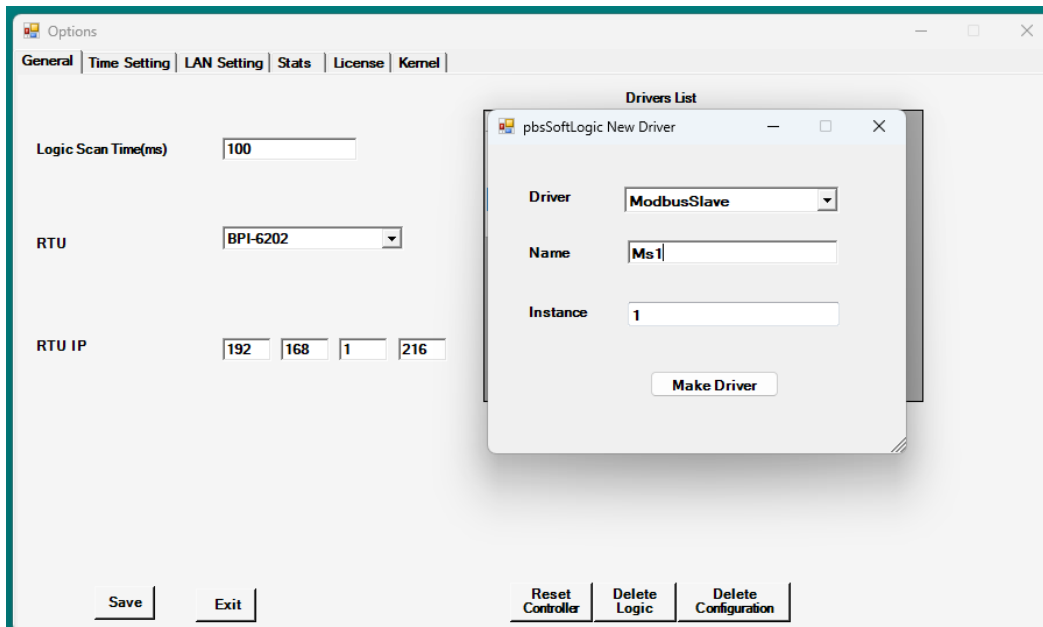
By default, the console is enabled for the RS232 port, if you want to use it as a regular RS232 port, you must first disable the console function in Ubuntu.

Suppose you want to define a Modbus Master for the RS485-1 port for the project.

Right-click in the driver list and add Modbus Master Driver. For example, name it MM1.
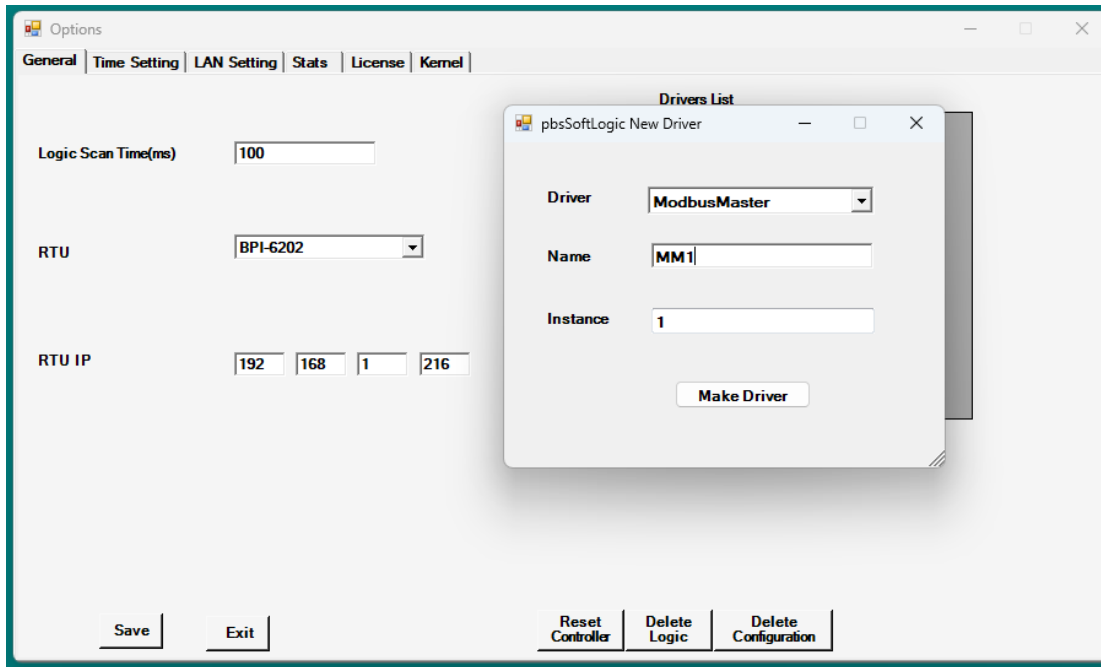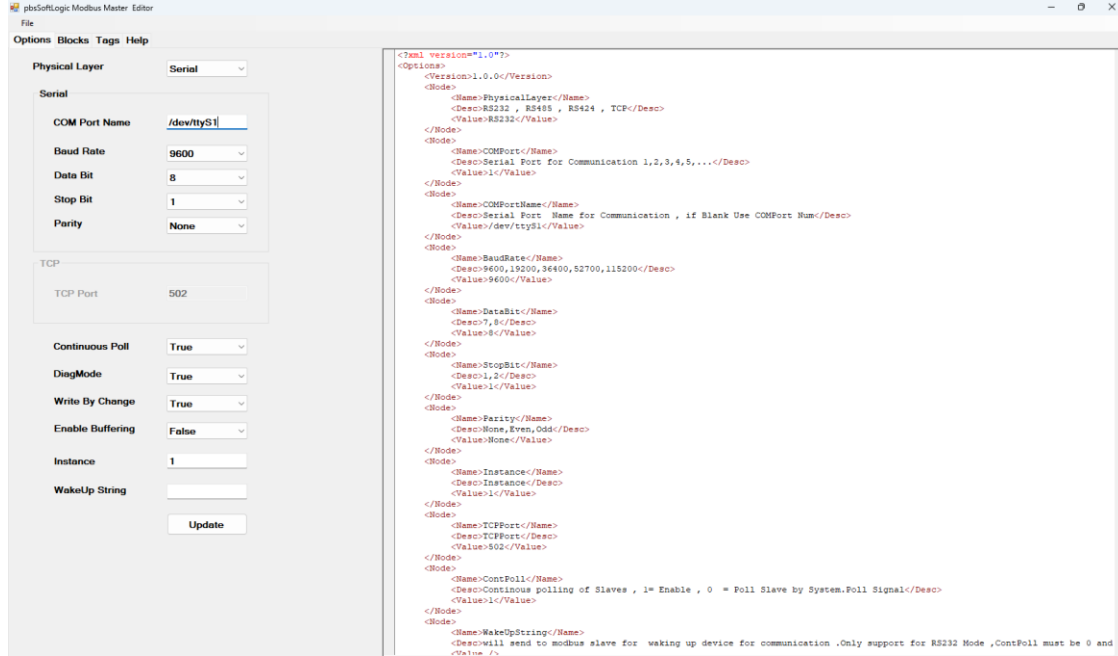

Add One Modbus Slave Driver and name it MS1.

Keep the instance number 1 for both drivers. The instance number must be unique for a driver type. If you want to add another Modbus Master to the project, you must change the instance to 2 for the second driver.

Double-click on MM1 Driver and you will see the following screen:

You need to use the serial port name as the COM port name and set the other parameters.

Likewise, you should use the serial port name as the COM port name for modbus Slave Driver.