# pbsSoftLogic

# Concepts



- **1 Introduction**
- 2 pbsSoftLogic Installation
- **3 Basic concepts**
- 4 Function Block programming Language
- **5 Quick Startup**
- 6 Runtime Kernel for Linux, transferring License to Controller
- 7 Project Settings facilities

# **1 – Introduction**

pbsSoftLogic is an open RTU/PLC programming environment from pbsControl LTD. pbsSoftlogic is developed by Dot Net Technology. The development version of pbsSoftLogic runs on the Windows operating system, and its runtime runs on any Linux-based controller.

pbsSoftLogic has the following specifications:

- Standard function block programming environment
- Lua programming and C language are used to develop user defined function blocks
- Offline simulation of the program developed in Windows
- More than 400 ready-made and tested function blocks for easy programming.
- Support Modbus, DNP3, IEC101/104, OPC UA, Redis Client, MQTT, SQL Client, GSP, IEC62056-21 drivers
- IEC62351 is supported for DNP3 and IEC104 protocols.
- Supports program monitoring and forcing variables during execution.
- RTU status monitoring tool

- The runtime kernel runs on any Linux-based controller. The Windows CE, win32, and QNX runtime kernels have been removed in version 5.0.



# 2 - pbsSoftLogic installation

pbsSoftLogic IDE runs on Windows operating system which supports Dot Net Frame work 4.7.2.

You can download the latest pbsSoftLogic IDE from http://www.pbscontrol.com

Simply extract the file and run pbsFBEditor.exe. No need for any installation process

Do not copy the pbsSoftLogic folder to the path with special characters such as space, /, (,).

In this document, we assume that you have copied pbsSoftLogic to the c:/pbsSoftLogic folder.

When you run pbsFBEditor.exe you will see the following screen:



# 3 - Basic concepts

Writing programs for SCADA systems is a critical task. Using low-level languages such as C/C++ and C# for such projects is not recommended for the following reasons:

- 1. Non-reusable
- 2. Difficulty in transferring the project to others and training other engineers to continue the project
- 3. High risk during execution for stability and error-free
- 4. Taking a long time to develop the project

Function Block language is a language for control engineers. They can focus on process logic without worrying about the software side. FB is a complete graphical language with many tested and ready-made functions inside.

Using the function block language has the following advantages:

1 –100% reusable .There are many tested and ready-made functions that can be used in different projects with complete documentation.

2 – It is very easy to train control and process engineers for programming in projects.

3 – pbsSoftLogic has been used in many projects and sites for the last few years, so there is no error in the runtime and development environment.

4- You can use pbsSoftLogic and Function block language as a framework for your entire automation projects.

pbssoftLogic is an IDE for developing Function Blocks, simulating and downloading programs on Linux based controllers. You can use Lua Scripting or C language to develop new FB and integrate with IDE.

All pbsSoftLogic FB source code is open source and located in \FBDSourceCode.

The main element of a program is the FB (Function Block). Below you can see some simple examples.

Example1:



In this logic, two signals PMP1\_STS\_RUN and PMP2\_STS\_RUN are input to the OR FB and the output is written to the PUMPING signal.

Example 2:



In Example 2, PM1\_ActivePower is multiplied by 100, PMP2\_Activepower is multiplied by 100, and both results are added together and written to the Power\_Instance signals. (write on two different drivers).

Example 3:



In example 3, a pulse with a period of one second is generated and written to the DTTag1 signal in the iec104 driver.

The main elements of a functional block program are:

- 1 Input/output signals: connected to communication drivers and local I/O.
- 2 FB: Function Blocks.
- 3 Connection between I/O signals, FBs and between FBs.
- 4 Constant signals: different type of Constant Signals: Integer (I), Float (F), Boolean (B), Time (T)

Constant Signal Format: Type # Value.



5- Internal variables: unlimited internal variable is possible in logic, but each instance must have a different name. Variables with the same name in the program have the same value.

From FBList you can select Constant Tags, Global Variables or Driver tags.

∃ 📅 FBList
🕞 InputSignal
🛶 🔁 OutputSignal
🕞 ConstantTime
🕞 ConstantInt
🛛 🔓 ConstantFloat
🕞 ConstantBool
🕞 GlobalVarInput
GlobalVarOutput
· 😐

In the following example you can see constant and global variables usage.



In the Vars tool, you can see a list of all global variables in the open program.

By double-clicking any variable item in the Vars tool, Logic will focus on the variable's location. So you can easily browse and check all the variables.

6 - Comments: You can put a comment anywhere in the logic. Drag a comment element from the FBList and drop it into the logic. Then click on the comment and change its content. A comment is like a dynamically sized yellow text box.

# 🍳 💽 💽 🗶 🔝 🚳 🌰 🕒 🥔 🌽



You can see the list of all comments in the open application from the Comments tool. You can view the location of the comment by double-clicking on the comment list.

Function block programming rules:

1 - FB inputs (left) are always connected to a source. You can connect a source (I/OSignal, Internal Variable and Constant) to different FB inputs. But multiple source to one FB entry is not valid.

2 - FB outputs (right side) can be connected to different signals. (not fixed signals)

3 - There is no limit on the number of levels of FB connections.

In the image below you can see an invalid connection for the CTU Function Block, Reset input. You cannot bind multiple variables to one FB input.



4 - Logical execution: each FB has an execution number. Click FB and press F4, you can see FB properties



window. Scroll through the properties to find ExeSeq.

When you start developing logic, FBEditor automatically increments the ExeSeq value for each FB you use. But you can change its value and in this way you can control how the logic is executed. We recommend that you set all ExeSq numbers manually, because when you copy-paste part of the logic, FBEditor will put the same values for the pasted elements. FBEditor sorts all Fbs based on ExeSeq number and compiles and generates output file based on ExeSeq command.

2025

In the figure below, PulseGen has ExeSeq number equal to 1 and CTU has ExeSeq number equal to 2.



Therefore, at runtime, PulseGen is executed first and then CTU.

In the following example, the ExeSeq value for PulseGen is greater than CTU, but the logic resolves without problems and there is no output for CTU FB in only one RTU cycle.



5 - Logic FB Instance name: Each FB has FBName and instance name. These two properties are equal by default. But you can change the instance name to any unique name in your logic. Suppose you are controlling a pump by Drive1V2 FB. By renaming the FB instance to "Pump1Mng", the compiler will use Pump1Mng as the compile-time identification of the FB. By default it uses the PartID attribute, which is always unique in logic.



In the figure below, we change the name of the PulseGen instance to PulseGen200 and the CTU to CTU100, then you can move the logic through the FB Instance window by double-clicking on the instance names.





In the runtime kernel, the PartID is used as the unique identifier for each FB to manage static data for each FB.

When you make warm update for the logic, because the PartID is assigned by the compiler at compile time, so a different PartID can be assigned to the same FB. But if you use the instance name, the runtime kernel will always use the instance name to handle static FB data, so ensures proper warm update at runtime.

Note: If you want 100% warm logic update, you must use instance name for critical function blocks.

6 - You can write your logic in several POUs (Program Organization Unit). All POUs are merged into one file based on ExeSeq number and passed to the runtime kernel. Therefore, multiple POUs are more suitable for structured programming, and there is no difference in runtime if you write all programs in the main POU or split it into multiple POUs.

7- Logic execution always starts from FB with the smallest ExeSeq number. First, all input signals are read and transferred to the FB, the logic of the FB is solved, and then all outputs are written to the output signals. Then the next FB is executed. Note that the FBs for ExeSeq are always sorted in ascending order in the final program file transferred to the execution core.

# 5 - Quick Startup

In this part, we will write a simple logic with pbsSoftLogic and run on Linux controller.

Step1: Make a new Application with pbsSoftLogic . Run pbsFBEditor.exe . In File Menu, Select New.



At first step you should make a directory for your project. You can make anywhere in your system with proper name related to your project. Suppose we will make Quick1 Directory in pbsSoftLogic VSLE Path.

揭 Save As

	is PC > Windows/_OS(C:) > PIP2u	12 > pbsSoftLogic > PSLE > VS	SLE > Qucil	:1	
Organize 🔻 New folde	er				
📌 Quick access	Name	Date modified	Туре	Size	
🐔 SkyDrive 🛛 🖈				No items match	your search.
🕹 Downloads 🛛 🖈					
🔜 Desktop 🛛 🖈					
📙 projects 🛛 🖈					
🔮 Documents 🛛 🖈					
📰 Pictures 🛛 🖈					
🚅 G:\ 🛛 🖈					
📸 EA 🛷					
📙 utility 👘 🖈					
MMNew					
📙 Modbus Test - PBS					
sql					
test1					
痜 OneDrive					
💻 This PC					
🔜 Desktop					
🔮 Documents					
🖶 Downloads					
🎝 Music					
📰 Pictures					
📑 Videos					
🏪 Windows7_OS (C:)					
🙆 CD Drive (D:)					
🕳 Local Disk (E:)					
😽 Lenovo_Recovery ((					
🥩 Network					
🔩 Homearoup					
5.					
<b>1</b> 1	al				
File name: Quici	ч 				
Save as type:					

Click the Save button, the save form will automatically close and you are ready to configure and program. You could not close the MainPOU and it will remain open as long as the project is open.

🛃 pbsSoftLogic Ver 2.0.0 - [Function Bl	sck Program]	- 0 ×
f≈ FBList ₽	ainPOU	POUs #
CAPP2/11/20cbs/2011/2015/2011		General Comments         Imputs         Properties         Imputs         Properties         Imputs         Imputs
U:\PIP2012\bbsSoftLoaic\PSLE	VSLE VUCKTVUICKTXMI	100% //

Step2 : Select your RTU Type . Click on project setting button



When the project settings page opens, "GeneralBeagleBone" is displayed as the controller by default.

🖳 Options		=
General Time Setting I	AN Setting Stats   License   Kernel	
Logic Scan Time(ms)	100	Drivers List
RTU	GeneralBeagleBone	
RTU IP	0 0 0 0	
Save	Exit	Reset Delete Delete Controller Logic Configuration

Open the "RTU" combo box, you can see the different RTUs supported by pbsSoftLogic. For now, use the same "BIP-6202". Enter the IP of the RTU, for example 192.168.1.164. Keep the logic scan time at 100ms. This is the loop time to read all the driver inputs, solve the logic and write the driver outputs. Every 100ms the above sequence will repeat, but the whole sequence may only take 2ms to complete and the processor will be in sleep mode for 98ms. Click the Save button to save the settings.

Logic Scan Time(ms)	100	
RTU	BPI-6202 BPI-6202 Linux ECU1251D	-
RTU IP	ECU150 IAS2000 RTU-550E PDR-600 FTU-600	

Step3: Define Local IO for RTU

Right-click in the list of drivers and select the New Driver command:

🖳 Options						-
General Time Setting LAN	Setting Stats License Kernel					
			Drivers	_ist		
LogicScanTime(ms)	100	Name	Path	Туре	Enable	
RTU	BPI-6202			New Expl	Driver orer	
RTU IP				Edit Dele	te Driver	
Save	Exit	Reset Controller	Delete Logic	Delete Configura	e ation	

It will show list of Supported Drivers of pbsSoftLogic :

Select Local \_IO and choose a unique name for the driver. (For example, LIO)

🛃 Options							
General Time :	Setting LAN	Setting Stats	License Kernel				
					Drivers	_ist	
				Name	Path	Туре	Enable
Logic Scan Ti	me(ms)	100	🔲	law Driver		~	
			pbsSoftLogic N	lew Driver		^	
RTU		BPI-6202	Driver		•		
				LOCAL_IO	1		
			Name	ModbusMaster			
RTILIP				DNP3Slave			
mon		U U	Instance	DNP3Master IEC8705Master			
				IEC8705Slave			
				Make Driver			
						11	
Sa	ve	Exit		Reset Controller	Delete Logic	Delete Configuratio	on

🛃 pbsSoftLogi	c New Driver	-		×
Driver	LOCAL_10		•	
Name				
	Make Driv	er		

Click the "Make Driver" button. pbsSoftLogic will add the BPI-6202 configuration file to your project.

🛃 Options							_	
General Time Setting LAN	Setting Stats   License   Kernel							
				DriversL	.ist			
			Name	Path	Туре	Enable		
Logic Scan Time(ms)	100	▶	LIO	<b>MLIO</b>	LOCAL_IO			
BTU	BPI-6202 👻							
RTU IP	192 168 1 164							
						_		
- 1	1	F	Reset	Delete	Delete	1		
Save	Exit	Co	Introller	Logic	Configuratio	n		

LOCAL\_IO is a generic driver for modeling the hardware functionality of RTU. pbsSoftLogic creates different Local\_IO configuration files based on the RTU type. Local IO manages all IO ports located on the main CPU, such as LED, WDT, Digital IO, etc.

Right click on LIO Driver and Select "Edit" option.

🖳 Options							-
General Time Setting L	AN Setting Stats License Kernel						
				Drivers	List		
			Name	Path	Туре	Enable	
Logic Scan Time(ms)	100	▶	LIO	ALI0	LOCAL_IO		
			New Drive	er			
			Explorer				
			Edit				
RTU	BPI-6202		Delete Driv	ver			
RTU IP	192 168 1 164						
Save	Exit		Reset	Delete	Delete	_	
				LUYIC			

The default editor for the LocalIO Driver will open:

```
🖶 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="0" 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>
```

For more information about BPI-6202 local I/O, please refer to pbsSoftLogic Runtime Kernel Configuration for BPI-6202/04.

You can explore driver too, and open driver folder.

📙   🛃 📕 🖛   L	10			-	_
File Home	Share	: View			
$\leftarrow \rightarrow \land \uparrow$	« PI	P2012 > pbsSoftLogic > PSLE >	VSLE > Qucik1 > LIO	ٽ ~	S
👆 Downloads	* ^	Name	Date modified	Туре	Siz
📃 Desktop	*	📔 Local_IO.xml	10/16/2016 10:41	XML File	
📙 projects	*				
🔮 Documents	*				
📰 Pictures	*				
🚅 G:\	*				
<b>∭</b> 2 F:\	*				
📙 utility	*				
MMNew					
📙 Modbus Tes	it - P				
📙 sql					
📙 test1					
痜 OneDrive					
💻 This PC					
📃 Desktop					
🗎 Documents					

pbsSoftLogic creates a directory for each driver in the project directory.

If you open project directory you will see following file configuration:



Quick1.xml: source of your logic

Quick1.cfg: Project Configuration file.

Quick1.lx: Compiled Project Configuration file that is transferred to RTU

Quick1.c11: Compiled Logic that is transferred to RTU

LIO Directory: Directory for Local\_IO Driver. Insider LIO directory, pbsSoftLogic is make Local\_IO.xml file.

Step 4 Function Block Programming :

In the left panel, you can see various ready-made function blocks, and in the middle panel, the function block programming area.

Open the Timers group and select PulseGen. Drag and drop it into the application area.



"PulseGen" generates continuous pulses with equal duration (low and high).

When the input "Trg" changes from low to high (0 to 1), the pulse train at the output "Q" starts with a low and high duration equal to the input "Time".

In the FB list panel, drag and drop the "ConstantBool" object and connect it to the "Trg" input. Then drag and drop the "OutputSignal" object and connect it to the "Q" output. Leave the "Time" input unconnected.



When an input port is not connected to any signal, it takes the default value that is preset for each input port (you can change the default input values of the input port).

CNT B#True Timers PulseGen Time Q	DRV LIO:LED_Run
PulseGen	🖳 Driver Output Signal 🛛 — 🔲 🗙
	Filter
	<ul> <li>□-LI0</li> <li>□-SYS.Reset (SYS_0)</li> <li>□-SYS.WDTEnable (SYS_1)</li> <li>□-SYS.ModemPW (SYS_2)</li> <li>□-SYS.ModemConnectCmd (SYS_3)</li> <li>□-SYS.ModemConnected (SYS_4)</li> <li>□-SYS.ModemRSSI (SYS_5)</li> <li>□-LED_Pwr (D0_0)</li> <li>□-LED_Func (D0_1)</li> <li>□-LED_Func (D0_2)</li> </ul>

Right-click on the output signal connected to the "Q" port and select the "LIO:LED\_Run" signal.

Double-click on the "LED\_Run" signal and close the "Driver Output Signal" form.

Click on Save and compile button at top.



It will save and compile your logic.

# 6- Runtime Kernel for Linux, transferring License to

In this chapter we will discuss the pbsSoftLogic runtime structure inside the RTU. The pbsSoftLogic runtime core is based on a very simple concept.

pbsSLKLX: This is the main program in Linux that starts automatically when the RTU boots or you can load it manually for diagnostic purposes.

Communication drivers: Various communication drivers supported by pbsSoftLogic.

In the figure below you can see how the above components are connected to each other.



# Runtime Structure

When you run pbsSLKLX, the following steps are performed on the RTU:

- 1. pbsSLKLX loads the file /home/pbsLX/logic.cfg. This file contains all the communication protocols defined for the active project.
- 2. pbsSLKLX dynamically loads the communication driver library into memory.
- 3. pbsSLKLX gets the driver parameters from the logic.cfg file and passes them one by one to the communication driver library.
- 4. pbsSLKLX adds the communication blocks, slots and finally the driver tags.
- 5. pbsSLKLX calls the pbsInit function to initialize the communication driver.
- 6. pbsSLKLX repeats steps 2, 3, 4, 5 for all the drivers defined in logic.cfg.
- 7. pbsSLKLX starts reading input tags for all communication drivers by calling pbsReadTag.
- 8. pbsSLKLX solves the RTU logic once.
- 9. pbsSLKLX writes all output tags of the communication drivers by calling the pbsWrite method.
- 10. Repeats steps 7, 8, 9.

As is clear from the sequence above, the communication drivers have an integrated API interface to communicate with pbsSLKLX.

The execution time of steps 7, 8 and 9 is the actual logical scan time of the RTU. You can see the actual logical scan time when connecting to the RTU by pbsSoftLogic IDE. The actual logical scan time is shown at the bottom left of the screen. In the figure below, the actual logical scan time is 1 ms.



You cannot control the actual logical scan time, but you can set the logical scan time in the project settings page.

For this project logical scan time is 100 msec .

🛃 Options							-
General Time Setting LAN	Setting Stats License Kernel						
				Drivers	List		
			Name	Path	Туре	Enable	
Logic Scan Time(ms)	100	▶	LIO	<b>ALIO</b>	LOCAL_IO		
RTU	BPI-6202 💌						
570.15							
RIUIP	192 168 1 164						
Save	Exit	C	Reset ontroller	Delete Logic	Delete Configuratio	n	

This means that every 100 milliseconds, the RTU reads all inputs defined in the drivers (step 7), executes the logic (step 8), and writes all outputs defined in the drivers (step 9). The execution time for steps 7, 8, and 9 is 1 millisecond. Therefore, the RTU processor is in sleep mode for 99 milliseconds.

pbsSLKLX has only one thread to read the driver input tags, execute the RTU logic, and write the driver output tags. But there is no limit in the driver library to take up a large number of CPU threads.

When you reduce the logical scan time and bring it closer to the actual logical scan time, the RTU processor utilization increases. Therefore, you should choose the logical scan time based on your process conditions and it should always be longer than the actual logical scan time.

You can use less than 1ms for the logical scan time, setting 0.1 or 0.01ms is also possible.

Please note that the pbsSLKLX scan time is different from the communication protocol scan time. You can read Modbus Slaves every second, but for example, pbsSLKLX reads/writes driver tags every 10ms.

Slave and Master Drivers: There is a conceptual difference between Master drivers such as Modbus TCP/RTU Master, IEC101/104 Master and Slave drivers such as Modbus TCP/RTU Slave, DNP3 Slave, IEC101/104 Slave.

<u>Master Drivers</u>: In the figure below, you can see that the Modbus Master Driver reads digital and analog inputs from Slave devices and writes digital and analog outputs to Slave devices.

pbsSLKLX reads input digital and analog tags from drivers and writes output digital and analog tags to drivers.



In RTU logic, you must use the input signals on the left side of the function blocks to read from the driver, and you must use the output signals on the right side of the function blocks to write to the driver.



<u>Slave Drivers</u>: In the figure below, you can see that Modbus Master devices (HMI panels, SCADA software) read digital and analog tags from the Modbus Slave driver and write digital and analog output tags to the Modbus Slave driver.



pbsSLKLX writes digital and analog input tags to the Modbus Slave driver and reads digital and analog outputs from the Modbus Slave driver. These tags are written to the Modbus Slave driver by the external Modbus Master device.

In slave drivers, you should use DO and AO tags that are read from the driver on the left side of the function blocks and DI and AI tags that are written to the driver on the right side of the function block.



#### Linux Run Time Kernel

pbsSoftLogic has two parts:

- 1 Engineering Station. Runs on Windows operating system
- 2 Runtime Engine. Runs on Linux embedded inside RTU

In this part we will talk about Linux runtime engine.

The pbsSoftLogic runtime engine for Linux has the following modules:

- It locates at /home/pbsLX directory
- /home/pbsLX/pbsSLKLX file is main runtime module. It is an executable Linux file.
- /home/pbsLX/Imp/ logic monitoring protocol, for monitoring logic by IDE and Lua Interface module
- /home/pbsLX/fblib/ pbsSoftlogic internal, user and lua function blocks. For each FB group there is one dynamic library.
- /home/pbsLX/drvlib/ driver folder
- The "License.lic" file is for the runtime license. Without the license file, the program will run for 30 minutes.
- "Upload.zip" When you transfer a project to RTU, the last project folder is always transferred as the "upload.zip" file. You can use this file in case you lose the project.
- "getcpu.sh" Used to read CPU usage by LocalIO.
- "getmac.sh" Generated by the runtime kernel to check license.

When you unzip the kernel runtime, you can see the following directories:



For each RTU, you need a license file for normal operation. Without a license, the device will run for 30 minutes and you will need to restart the controller.

When you purchase a pbsSoftLogic license, you will receive a license key. To activate the license, follow these steps:

1- Open the project settings page and select the License tab.

🖳 Project Options	Contraction of the local division of the loc	- AND	_ <b>_</b> X
General Time Setting LAN Setting Stats Lice	nse Kernel		
		Manage License for Controller	
	License Key		
	Site		
	GetLicense from Web	Read License Copy License to from Controller Controller	
1			

2 - Make sure that your computer on which pbsSoftLogic IDE is installed is connected to the Internet and the controller is also connected to the network with the computer.

- 3 Copy the controller license key and paste it into the License key text box.
- 4 Write a description of your project, project name, location, country,
- 5 Click on "Get License from Web". You will connect to the pbsControl website and get the licenses.
- 6 Modbus, DNP3, IEC and BACNet check boxs will be checked normally.
- 7 Click "Copy License to Controller." This will transfer the license file to the controller.
- 8 Restart Controller.

If you have a controller and want to check its license, click "Read license from controller."

2025

#### Working with FileZilla

You can use FileZilla client utility to explore and edit RTU Files and directories.

Download filezilla from https://filezilla-project.org/

Run filezilla client you will see following page :

Image: control 192.168.233.136 - FileZilla         File Edit View Transfer Server Bookmarks Help         Image: control 192.168.233.136         Host: 192.168.233.136         Username: root         Password         Status:       Directory listing of "/root" successful         Status:       Retrieving directory listing of "/"         Status:       Retrieving directory listing of "/"	f: Port:	Quickconnect		- 0	×
Local site:         C\\PIP2012\pbsSoftL gic\PSLE\target\pbs2008RTUV2RC8\           Image: pbs2008RTUV2RC8         Image: pbs2008RTUV2RC8_1           Image: pbs2008RTUV2RC8_2         Image: pbs2008RTUV2RC8_2           Image: pbs2008RTUV2RC8_2         Image: pbs2		Remote site: / / ? bin ? boot ? dev ? etc ? initrdimg			> < <
- ☐ Timezone - uk - Utility ⊕ G VSLE		Filename	Filesize Filetype File folder	Last modified 12/2/2016 2:07:00 AM	
Filename  gpdata pbsLX sqldb qlsynclog Chat-Module-UC15-connect interfaces.default 3 files and 5 directories. Total size: 1,298 bytes	Filesize Filetype File folder File folder File folder File folder File folder 190 File 699 DEFAULT File	boot dev etc home bib lost+found mmt opt c 21 directories	File folder File folder File folder File folder File folder File folder File folder File folder File folder File folder	12/2/2016 2:10:00 AM 1/11/2017 12:52:00 PM 1/11/2017 12:52:00 PM 12/2/2016 153:00 AM 12/2/2016 153:00 AM 12/2/2016 153:00 AM 12/2/2016 1:52:00 AM 12/17/2016 4:56:00 AM	1 4 4
Server/Local file Direction Remote file		- ).	Size Priority		Statu >
Queued files Failed transfers Successful transfers	z 💿 🔯 🖧 🕅 (			Queue: empty ENG 8:06 AM	••

Type RTU IP at host field. Type root and root password in user name and password fields.

All RTU directories are showing at right panels and your PC directories at left s panels.

Double click on home directory. You will go inside home directory. Double click on pbsLX .pbsLX directory is runtime kernel of pbsSoftLogic for Linux Operating system .

Filename	Filesize	Filetype	Last modified
📙			
		File folder	1/11/2017 1:00:00 PM
		File folder	1/11/2017 1:00:00 PM
Imp		File folder	1/11/2017 1:00:00 PM
logic.c11	1,349	C11 File	1/11/2017 1:05:00 PM
logic.cfg	24,893	CFG File	1/11/2017 1:05:00 PM
pbsSLKLX	474,232	File	1/11/2017 1:00:00 PM

If you install debian X86 Linux on your pc over VMWare or Virtual Box, then you can easily use pbsSoftLogic Debian X86 Runtime and test all functionalities of pbsSoftLogic on your PC.

Note : for transferring files between Windows and Linux Systems , always set Transfer File Type to Binary. you can find this option in Edit Menu , Setting menu and Transfers Segment .

Settings	×	(
Settings Select page: Connection FTP Active mode Passive mode FTP Proxy SFTP Generic proxy File exists action Interface Date/time format Filesize format File lists Language File editing File editing Ligging Debua	Default transfer type: Auto Auto ASCII Binary Automatic file type classification Treat the following filetypes as ASCII files: am asp bat c c c c c c c c c file S Treat files without extension as ASCII file Treat files without extension as ASCII file Treat dotfiles as ASCII files Dotfiles are filenames starting with a dot, e.ghtaccess	
OK		

By default it is set to "Auto" that is damaging Linux files at transfer time from Windows to RTU.

For editing RTU configuration files in windows you need to use NotePad++ Editor to not damage Text file format when transfer to windows System .

Install NotePad++ utility from <a href="https://notepad-plus-plus.org/">https://notepad-plus-plus.org/</a>

At first time that you View/Edit any Linux Configuration file , Filezilla will ask you for Custom Editor .

In this Stage set Nodepad++ as default editor in Filezilla . This will change File Editing Option in Setting page as following :



For changing Network Interfaces in RTU, View/Edit /etc/network/interfaces file.



For transferring files from RTU to Windows, Select File, Right click on name of File or Directory and run "Download" command.

You can delete or Rename files inside RTU by selecting file, right click on name of file and select Delete or Rename Commands.

# 7 - Project Settings facilities

There are many facilities in setting page in pbsSoftLogic Editor.

Open Setting Page you can see following tabs:

🛃 Options							_
General Time Setting LA	N Setting Stats License Kernel						
				Drivers	List		
			Name	Path	Туре	Enable	
LogicScan Time(ms)	100	▶	LIO	<b>MLIO</b>	LOCAL_IO		
RTU	BPI-6202 💌						
nioir	192  168  1  164						
Save	Exit	C	Reset Controller	Delete Logic	Delete Configuratio	m	

- General
- Time Setting
- LAN Setting
- Stats
- License
- Kernel

General Tab: In This page you can set following parameters:

- Logic Scan time (Msec)
- Controller Type
- Controller IP address
- Communication Drivers
- Restart Controller
- Delete Logic
- Delete Configuration

Time Setting:

🖳 Options	
General Time Setting LAN Setting Stats License Kernel	
Set Controller TimeZone GMT+1 Change TimeZone	
Read Controller Tue 03 Jun 2025 09:42:13 AM BST	
Time	
SetTime NTP server time.windows.com	
Set Controller Time with PC	
Set Controller Time with PCUTC	
Write To RTC	

Set the controller's time zone: Select your location from the list box and click Change time zone.

Read Controller Time: Reads the current date, time, and time zone of the controller.

Time Setting: Sets the controller's time from an NTP server, which can be a computer on the network or any time website. However, the controller must be connected to the Internet.

Set controller time with computer: This command sets the controller time from the computer running pbsSoftLogic.

Set controller time with PC UTC: This command sets the controller time from the computer running pbsSoftLogic in UTC.

Write to RTC: Write the latest time to the controller's RTC.

LAN Settings:

🖳 Options		_	×
General Time Setting	LAN Setting Stats License Kernel		
ECU1251 LAN RPI	GPRS-3G		
Read LAN1 Settings	Read 4G Settings1 auto eth0 iface eth0 inet static		
Write LAN1 Settings	Write 46 Settings1 address 192.168.1.136 netmask 255.255.255.0		
Read LAN2 Settings	Read 4G Settings2 gateway 192.168.1.1 allow-hotplug eth0		
Write LAN2 Settings	Write 4G Settings2		
Sample static Lan Set auto eth0 iface eth0 inet static address 192.168.1.20 netmask 255.255.255 network 192.168.1.0 gateway 192.168.1.1	ting 00 5.0		
Sample dhcp Lan Setti auto ethO iface ethO inet dhcp	ing		

Depending on the RTU type, you can use different pages to configure the LAN. For ECU1251D, you can see the following functions:

Read LAN1/2 Settings: Reads the current LAN1/2 settings from the controller.

Write LAN1/2 Settings: Writes the new LAN1/2 settings to the controller.

To change the controller IP address:

- 1 First read the LAN settings.
- 2 Change the IP address for each LAN port and other settings.
- 3 Write the new settings to the controller.

#### Setting 4G for ECU1251D

There are two files for 4G settings in ECU1251D. As with LAN settings, first read the configuration, make changes and write the new settings to RTU.

🛃 Options			_	×
General Time Setting	LAN Setting Stats Licen	se Kernel		
ECU1251 LAN RPI	GPRS-3G			
Read LAN1 Settings	Read 4G Settings1	ABORT "BUSY"	_	
Write LAN1 Settings	Write 4G Settings1	ABORT "NO DIALTONE" ABORT "ERROR"		
Read LAN2 Settings	Read 4G Settings2	ABORT "NO ANSWER" TIMEOUT 30		
Write LAN2 Settings	Write 4G Settings2	<pre>## AT ##OK ATEO ##OK ATE;+CSUB;+CSQ;+CPIN?;+COPS?;+CGREG?;&amp;D2 # Insert the APN provided by your network operator, default apn is internet OK AT+CGDCONT=1,"IP","ssg.nie.co.uk" OK ATD*99#</pre>		
Sample static Lan Sett auto eth0 iface eth0 inet static address 192, 168, 1.20 netmask 255, 255, 255 network 192, 168, 1.0 gateway 192, 168, 1.1	ing 10 .0	CONNECT		
Sample dhcp Lan Setti auto eth0 iface eth0 inet dhcp	ng			

#### 🛃 Options

1				
General Time Setting	LAN Setting Stats	Elicense	Kernel	
ECU1251 LAN RPI	GPRS-3G			
Read LAN1 Settings	Read 4G Setti	ngs1	user GA566	
		I I	password adQC4sD3	
Write LAN1 Settings	Write 4G Setti	ngs1	/dev/ttyUSB2	
	(		460800	
Read LAN2 Settings	Read 4G Setti	ngs2	nodetach	
			arteste	
Write LAN2 Settings	Write 4G Setti	nas2	debug	
white Baitz octaings	white the second	Igoz	useneerdns	
			defaultroute	
			connect '/usr/sbin/chat -s -v -f /etc/ppp/4g-internet'	
Sample static Lan Se	tting			
auto eth0				
address 192 168 1 2	200			
netmask 255.255.25	5.0			
network 192.168.1.0	)			
gateway 192.168.1.	•			
1				
Sample dhon Lan Set	tina			
campic andp currot				
auto eth0				
iface eth0 inet dhcp		1		

– 🗆 X

# Controller Stat tab:

Read CPU Information: It will Read Hardware Information from controller

Linux Command: cat	/pro	c/cpuinfo
--------------------	------	-----------

Options	— — X
ieneral   Time Setting   L4	AN Setting Stats   License   Kernel
CPU Info	processor : 0
	model name : ARMv7 Processor rev 5 (v71)
Mem Info	BogoMIPS : 48.00
Linux Version	Features : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 l
	CPU implementer : 0x41
Flash Info	CPU architecture: 7
	CPU variant : UXU
Free mem	CPU part : UxcU7
Dragona	CPU revision : 5
Flucess	
uname	processor : 1
	model name : ARNO/ Processor rev 5 (V/I)
hwclock	Bogonirs : 48.00
	Features : hair thumb fastmuit vrp easp heon vrpv3 tis vrpv4 faiva faivt vrpd32 f
devices	CPU implementer : 0X41
CPIL Freq	CPU architecture: /
crorreq	
Isusb T V	CPU particion - E
uptime	processor · 2
	model name · iDMv7 Processor rev 5 (v71)
itconfig Iscpu	
date in route	Features : half thumb fastmult vfn edan neon vfnv3 tis vfnv4 idiva idivt vfnd32 i
date ih ionte	CPU implementer : 0x41
ss s i t u	CPU architecture: 7
	CPU variant : 0x0
Isof netstat	

Read memory information: shows detail of memory usage of controller

# Linux Command : cat /proc/meminfo

🚽 Options			
General   Time Setting   LAN :	Setting Stats License	Kernel	
CPU Info	MemTotal:	2055820	kB
	MemFree:	1379408	kB
Mem Info	MemAvailable:	1822868	kB
Linux Marsian	Buffers:	45560	kB
Linux version	Cached:	416000	kB
Flash Info	SwapCached:	0	kB
	Active:	339012	kB
Free mem	Inactive:	208168	kB
	Active(anon):	88412	kB
Process	Inactive(anon):	6592	kB
	Active(file):	250600	kB
unallie	Inactive(file):	201576	kB
hwclock	Unevictable:	0	kB
	Mlocked:	0	kB
devices	HighTotal:	1310708	kB
	HighFree:	789200	kB
CPU Freq	LowTotal:	745112	kB
	LowFree:	590208	kB
ISUSD I Y	SwapTotal:	0	kB
uptime	SwapFree:	0	kB
	Dirty:	360	kB
ifconfig Iscpu	Writeback:	0	kB
	AnonPages:	85692	kB
date ip route	Mapped:	99560	kB
	Shmem:	9388	kB
33 3 1 C U	KReclaimable:	44488	kB
Isof netstat	Slab:	98308	kB
	SReclaimable:	44488	kB

Read Version: Read Controller Linux Version, GCC compiler version

# Linux Command : cat /proc/version

🖶 Options					- 🗆
General Time Setting LA	N Setting Stats	License Kernel			
CPU Info	Linux vers	sion 5.4.0-BPI-K	ernel (dailybuild) 13 [linaro-7.4-20]	BuildA33A4OiContainer_ 19.02 revision	ubuntu2004x64-a)
Mem Info	56ec6f6b99	9cc167ff0c2f8e1a	2eed33b1edc85d4]	(Linaro GCC 7.4-2019.02	))
Linux Version	#4 SMP Wed	a Jul 3 16:51:53	UTC 2024		
Flash Info					
Free mem					
Process					
uname					
hwclock					
devices					
CPU Freq					
Isusb T V					
uptime					
ifconfig Iscpu					
date ip route					
ss s l t u					
lsof netstat					

Read Flash Information: Linux Command : df -h

Options						
neral   Time Setting   LA	N Setting Stats Lic	ense Kernel				
CPU Info	Filesystem	1K-blocks	Used	Available	Use%	Mounted on
	/dev/root	7238368	5899832	1322152	82%	/
Mem Info	devtmpfs	895812	0	895812	0%	/dev
	tmpfs	1027908	0	1027908	0%	/dev/shm
Linux Version	tmpfs	205584	3132	202452	2%	/run
Elseh Info	tmpfs	5120	0	5120	0%	/run/lock
	tmpfs	1027908	0	1027908	0%	/sys/fs/cgroup
Free mem	tmpfs	10240	40	10200	1%	/mnt/ramdisk
	tmpfs	205580	8	205572	1%	/run/user/110
Process	tmpfs	205580	0	205580	0%	/run/user/0
uname						
hwclock						
devices						
CPU Freq						
Isusb T V						
uptime						
ifconfig Iscpu						
date ip route						
ss s l t u						
Isof netstat						

# Read Free Memory: Linux Comamnd free

🖳 Options							- 🗆	×
General   Time Setting   LAN 9	Setting Stats	License Kernel						
General Time Setting LAN S CPU Info Mem Info Linux Version Flash Info Free mem Process uname hwclock devices CPU Freq Isusb T V	Setting Stats Mem: Swap:	License Kernel	used 171132 0	free 1378284 0	shared 9400	buff/cache 506404	available 1822088	
ifconfig Iscpu date ip route ss s I t u Isof netstat								

# List of Process : Linux Command : ps aux

🚽 Options											- 0	×
General   Time Setting   LA	N Setting Stats	Licen	se   Ke	rnel								
CPU Info	USER	PID	*CPU	% ME M	VSZ	RSS	TTY	STAT	START	TIME	COMMAND	
	root	1	0.0	0.3	33564	7956	?	Ss	Jun02	0:13	/sbin/init	
Mem Info	root	2	0.0	0.0	0	0	?	S	Jun02	0:00	[kthreadd]	
Linux Version	root	3 4	0.0	0.0	0	0	? 2	I< I<	Jun02 Jun02	0:00	[rcu_gp] [rcu_par_gn]	
	root	8	0.0	0.0	ň	ň	2	 T<	Jun02	0:00	[mm nercnu wa]	
Flash Info	root	ğ	0.0	0.0	ň	0	2	ŝ	Jun02	0:00	[ksoftirgd/0]	
Free mem	root	10	0.0	0.0	0	0	2	I	Jun02	0:30	[rcu sched]	
	root	11	0.0	0.0	0	0	2	s	Jun02	0:00	[migration/0]	
Process	root	12	0.0	0.0	Ō	Ō	?	s	Jun02	0:00	[cpuhp/0]	
	root	13	0.0	0.0	0	0	?	s	Jun02	0:00	[cpuhp/1]	
uname	root	14	0.0	0.0	0	0	?	s	Jun02	0:00	[migration/1]	
hweleck	root	15	0.0	0.0	0	0	?	S	Jun02	0:00	[ksoftirqd/1]	
IIITCIUCK	root	18	0.0	0.0	0	0	?	S	Jun02	0:00	[cpuhp/2]	
devices	root	19	0.0	0.0	0	0	?	s	Jun02	0:00	[migration/2]	
	root	20	0.0	0.0	0	0	?	s	Jun02	0:00	[ksoftirqd/2]	
CPU Freq	root	23	0.0	0.0	0	0	?	S	Jun02	0:00	[cpuhp/3]	
	root	24	0.0	0.0	0	0	?	S	Jun02	0:00	[migration/3]	
	root	25	0.0	0.0	0	0	?	S	Jun02	0:00	[ksoftirqd/3]	
untime	root	28	0.0	0.0	0	0	?	S	Jun02	0:00	[kdevtmpfs]	
aprime	root	29	0.0	0.0	0	0	?	I<	Jun02	0:00	[netns]	
ifconfia Iscou	root	31	0.0	0.0	0	0	?	S	Jun02	0:00	[kauditd]	
	root	33	0.0	0.0	0	0	?	S	Jun02	0:00	[oom_reaper]	
date ip route	root	34	0.0	0.0	0	0	?	I<	Jun02	0:00	[writeback]	
	root	35	0.0	0.0	0	0	?	S	Jun02	0:00	[kcompactd0]	
SS S I T U	root	36	0.0	0.0	0	0	?	SN	Jun02	0:00	[ksmd]	
lsof netstat	root	146	0.0	0.0	0	0	?	I<	Jun02	0:00	「kintearitvd]	

#### uname : linux command : uname -a

- Ontions	0.5			_	$\sim$
Uptions				-	^
General   Time Setting   LAN	Setting Stats	license   Kernel			
CPU Info	Linux cs62	2 5.4.0-BPI-Kernel #4 SMP Wed Jul 3 16:51:53 UTC	2024 armv71		 
Mem Info	armv/1 ar	V/I GNO/Linux			
Linux Version					
Flash Info					
Free mem					
Process					
uname					
hwclock					
devices					
CPU Freq					
Isusb T V					
uptime					
ifconfig Iscpu					
date ip route					
ss s l t u					
lsof netstat					

#### hwclock : Linux command : hwclock

🖶 Options	
General Time Setting	LAN Setting Stats License Kernel
CPU Info	2025-06-03 10:16:45.549196+00:00
Mem Info	
Linux Version	
Flash Info	
Free mem	
Process	
uname	
hwclock	
devices	
CPU Freq	
Isusb T V	
uptime	
ifconfig Iscpu	
date ip route	
ss s l t u	
Isof netstat	

Devices : Linux command : ls /dev/

💀 Options							
General Time Setting LAN	Setting Stats License Kernel						
CPU Info	apm_bios						
	autofs						
Mem Info	block						
Linux Version	bone						
	btrfs-control						
Flash Info	bus						
	cachefiles						
Free mem	char						
	console						
Process	counterO						
	counter1						
uname	counter2						
hwclock	cpu_dma_latency						
	cuse						
devices	disk						
	dma_heap						
CPU Freq	dri						
	fd						
Isusb I V	full						
unting	fuse						
uprilie	gpiochipO						
ifeopfig Isepu	gpiochip1						
incoming incopa	gpiochip2						
date ip route	gpiochip3						
	hwrng						
ss s l t u	i2c-0						
	i2c-1						
isof netstat	i2c-2						

#### CPU Freq : Linux Command : cpufreq-info



# Lsusb : linux command : Lsusb

🖳 Opt	ions												
Genera	Time Setting	LAN S	etting	Stats	Licen	se Ke	rnel						
	CPU Info		Bus	004 1	Device	001:	ID	1d6b:0001	Linux	Foundation	1.1	root	hub
_			Bus	002 1	Device	002:	ID	1a86:8091	QinHer	ng Electroni	.cs U	IS0 H	В
	Mem Info		Bus	002 1	Device	001:	ID	1d6b:0002	Linux	Foundation	2.0	root	hub
L	inux Version		Bus	003 1	Device	001:	ID	1d6b:0001	Linux	Foundation	1.1	root	hub
_			bus	001 1	Device	001:	τD	1000:0002	LINUX	Foundation	2.0	root	nup
	Flash Info												
	Free mem												
	Process												
	uname												
	hwclock												
	devices												
	CPU Freq												
Isu	usb T V												
	uptime												
ifee	onfig Iscpu												
dat	e ip route												
\$5	s I t u												
Isa	f netstat												

# T : linux command : lsusb -t

V : linux command : lsusb -v

Options		- 🗆 X
eral Time Setting L4	N Setting Stats License Kernel	
CPU Info	Rug 004 Device 001, ID 1465,0001 Linux Reundstien 1.1 rest hub	1
Mem Info	Device Descriptor:	
Linux Version	bLength 18 bDescriptorType 1	
Flash Info	bcdUSB 1.10	
	bDeviceClass 9 Hub	
Free mem	bDeviceSubClass 0	
Process	bDeviceProtocol 0 Full speed (or root) hub bMaxPacketSize0 64	
uname	idVendor Ox1d6b Linux Foundation idProduct Ox0001 1.1 root hub	
hwalaak	bcdDevice 5.04	
NWCIUCK	iManufacturer 3 Linux 5.4.0-BPI-Kernel ohci hcd	
devices	iProduct 2 Generic Platform OHCI controller	
	iSerial 1 1c1c400.usb	
CPU Freq	bNumConfigurations 1	
	Configuration Descriptor:	
	bLength 9	
untime	bDescriptorType 2	
aptillo	wTotalLength 0x0019	
nfig Iscpu	bNumInterfaces 1	
	bConfigurationValue 1	
e ip route	iConfiguration 0	
	bmAttributes OxeO	
	Self Powered	
of netstat	Kemote Wakeup	
	I naxPower UMA	

#### Uptime : linux Command : uptime



#### Ifconfig : linux command : ifconfig

🖳 Options	– 🗆
General Time Setting LAN Setting	Stats License Kernel
CPU Info eth	0 Link encap:Ethernet HWaddr 02:53:ec:98:99:1e
	UP BROADCAST MULTICAST MTU:1500 Metric:1
Mem Into	RX packets:0 errors:0 dropped:0 overruns:0 frame:0
Linux Version	TX packets:O errors:O dropped:O overruns:O carrier:O collisions:O txqueuelen:1000
Elash Info	RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
	Interrupt:41 Base address:0xa000
Free mem	
eth	1 Link encap:Ethernet HWaddr 00:af:02:54:c8:57
Process	inet addr:192.168.1.164 Bcast:192.168.1.255 Mask:255.255.255.0
uname	inet6 addr: fe80::e47f:5514:9af6:39b/64 Scope:Link
	UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
hwclock	AX packets:2636/6 errors:0 dropped:1 overruns:0 frame:0
- devices	aplicions:0 transfer:1000
devices	RX hytes·27924292 (27 9 MR) TX hytes·590499 (590 4 KR)
CPU Freq	Interrupt:40
Isusb T V	
	Link encap:Local Loopback
uptime	inet6 addr: 127.0.0.1 Mask:255.0.0.0
	IN LOOPBICK RINNING MTU:65536 Metric:1
irconfig Iscpu	RX packets:236 errors:0 dropped:0 overruns:0 frame:0
date ip route	TX packets:236 errors:0 dropped:0 overruns:0 carrier:0
	collisions:0 txqueuelen:1000
ss s l t u	RX bytes:18777 (18.7 KB) TX bytes:18777 (18.7 KB)
lsof netstat wla	nO Link encap:Ethernet HWaddr c0:f5:35:24:12:00

#### Lscpu : linux command : lscpu

🚽 Options General Time Setting LAN Setting Stats License Kernel Architecture: armv71 CPU Info Byte Order: Little Endian Mem Info CPU(s): 4 On-line CPU(s) list: 0-3 Linux Version Thread(s) per core: 1 Core(s) per socket: 4 Flash Info Socket(s): 1 Free mem Vendor ID: ARM Model: 5 Cortex-A7 rOp5 Process Model name: 

 MOGET Handler

 Stepping:
 rOp5

 BogoMIPS:
 48.00

 Plags:
 half thumb fastmult vfp edsp neon vfpv3 tls vfpv4

 uname hwclock devices CPU Freq Isusb T V uptime ifconfig Iscpu date ip route ss s l t u

lsof

netstat

Date : linux command : date

Ontions		-							
e ultri e ui	1			CI.	ata I				
General   Time Setting	LAN	156	etting	56	ats	Licens	e   Kernel		
CPU Info			Tue	03	Jun	2025	11:28:06	AM BST	
Mem Info	)								
Linux Version	)								
Flash Info	)								
Free mem									
Process									
uname									
hwclock									
devices									
CPU Freq									
Isusb T V									
uptime	)								
ifconfig Iscpu									
date ip route									
ss s l t u									
lsof netstat									

Iproute : linux command : ip route

🛃 Optio	ns						-	
General	Time Setting LAN S	Setting Stats Lice	nse Kernel					
	CPU Info	default via 19	2.168.1.1 de	v eth1 proto dhc	p metric 100	)		
	Mam Info	169.254.0.0/16	dev eth1 so	ope link metric	1000			
	Melli Inio	192.168.1.0/24	dev etni pr	oto kernel scope	link src 19	2.168.1.164 met)	cic 100	
Lir	nux Version							
	Flash Info							
	Free mem							
	Process							
	uname							
	hwclock							
	devices							
	CPU Freq							
Isus	b T V							
	uptime							
ifcor	nfig Iscpu							
date	ip route							
\$\$	sltu							
lsof	netstat							

🖳 Options							-	
General   Time Setting   LAN	Setting St	tats L	icense   K	ernel				
CPU Info	Netid	State	Recv-Q	Send-Q	Local Address	:Port Pee	<i>c</i> Addres:	s:Port
Mare Infe	u_str	ESTAB	0	0	/run/systemd/journal/stdout	27800		* 2726
Mem Inro	u_str	ESTAB	0	0	*	77247		* 7809
Linux Version	u_str	ESTAB	0	0	*	27323		* 2630
	u_str	ESTAB	0	0	/run/user/110/bus	28753		* 2631
Flash Info	u_str	ESTAB	0	0	/run/user/110/bus	26305		* 2732
	u_str	ESTAB	0	0	*	26292		* 2630
Free mem	u_str	ESTAB	0	0	/run/systemd/journal/stdout	21610		* 2271
	u_str	ESTAB	0	0	/run/systemd/journal/stdout	26272		* 2557
Process	u_str	ESTAB	0	0	/run/dbus/system_bus_socket	21000		* 2099
uname	u_str	ESTAB	0	0	/run/user/110/bus	27324		* 2871
	u_str	ESTAB	0	0	/run/dbus/system_bus_socket	27905		* 2730
hwclock	u_str	ESTAB	0	0	*	26291		* 2630
	u_str	ESTAB	0	0	*	28777		* 2805
devices	u_str	ESTAB	0	0	*	27316		* 2731
	u_str	ESTAB	0	0	*	25576		* 2627
CPU Freq	u_str	ESTAB	0	0	*	23574		* 2357
	u_str	ESTAB	0	0	/run/systemd/journal/stdout	21813		* 2181
ISUSD I Y	u_str	ESTAB	0	0	/run/systemd/journal/stdout	20071		* 2291
untime	u_str	ESTAB	0	0	*	20999		* 2100
aptime	u_str	ESTAB	0	0	/run/systemd/journal/stdout	26301		* 2629
ifconfig Iscou	u_str	ESTAB	0	0	*	26285		* 2731
	u_str	ESTAB	0	0	/run/systemd/journal/stdout	26302		* 2629
date ip route	u_str	ESTAB	0	0	*	27302		* 2790
	u_str	ESTAB	0	0	*	28716		* 2732
sssltu	u_str	ESTAB	0	0	*	27503		* 2878
lsof netstat	u str	ESTAB	0	0	/run/svstemd/iournal/stdout	27947		* 2739

ss-s : linux command : ss -s

🛃 Options

```
General Time Setting LAN Setting Stats License Kernel
           CPU Info
                                          Total: 337
                                          TCP: 14 (estab 4, closed 2, orphaned 0, timewait 0)
           Mem Info
                                                                            IP
1
                                                                                                  IPv6
                                          Transport Total

        Transport Total
        IF

        RAW
        1
        0
        1

        UDP
        6
        4
        2

        TCP
        12
        9
        3

        INET
        19
        13

        FRAG
        0
        0

        Linux Version
          Flash Info
                                                                                        3
                                                                                                  6
          Free mem
                                                                                                 0
            Process
            uname
            hwclock
            devices
           CPU Freq
                 TV
      lsusb
             uptime
     ifconfig Iscpu
     date ip route
       ss sltu
     lsof netstat
```

# Ss – I : linux command : "ss – l

Options					– 🗆 🗙
eneral Time Setting LA	N Setting St.	ats Licens	e Kernel		
	37-64-0	<i><i><i>a</i>+-+-</i></i>			Long L. Addus as (Doub
CPU Into	Netia	State	Recv-Q	send-Q	Local Address:Port
Mem Info	ni	UNCONN	U	0	rtni:syst
Mail IIIO	ni	UNCONN	0	0	rtnl:Netw
Linux Version	ni	UNCONN	0	0	rtn1:Kern
	ni	UNCONN	U	0	rtnl:wnoo '
Flash Info	ni	UNCONN	U	0	rtni:wpa
	ni	UNCOMN	0	0	rtnl:avan
Free mem	ni	UNCONN	U	0	rtnl:syst
Droopen	ni	UNCOMN	U	0	rtnl:wnoo
FIUCESS	ni	UNCONN	U O	U	rtn1:wpa_
uname	nl	UNCONN	U	0	rtnl:Netw
	ni	UNCONN	U O	U	rtnl:avan
hwclock	ni	UNCONN	U	U	rtnl:syst
	ni	UNCONN	0	U	rtn1:syst
devices	ni	UNCONN	4288	U	tcpdiag:ss/7
CDU Free	ni	UNCONN	704	U	tcpdiag:kern
LFU Freq	nl	UNCONN	0	0	selinux:kern
leush T V	nl	UNCONN	0	0	audit:kern
	nl	UNCONN	0	0	audit:syst
uptime	nl	UNCONN	0	0	audit:-766
	nl	UNCONN	0	0	audit:syst
ifconfig Iscpu	nl	UNCONN	0	0	fiblookup:kern
	nl	UNCONN	0	0	uevent:udis
date ip route	nl	UNCONN	0	0	uevent:syst
	nl	UNCONN	0	0	uevent:syst
22 2 1 L U	nl	UNCONN	0	0	uevent:upow
leaf netetat	nl	UNCONN	0	0	uevent:-129

# Ss -t : linux command : ss -t -a

# 🖳 Options

General Time Setting LAN Setting Stats License Kernel

CPU Info	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
	LISTEN	0	1	0.0.0:20001	0.0.0:*	
Mem Info	LISTEN	0	4096	127.0.0.53%lo:domain	0.0.0:*	
	LISTEN	0	128	0.0.0:ssh	0.0.0:*	
Linux Version	LISTEN	0	5	127.0.0.1:ipp	0.0.0:*	
Flash Info	LISTEN	0	100	0.0.0:1883	0.0.0:*	
	ESTAB	0	112	192.168.1.164:ssh	192.168.1.213:50276	
Free mem	ESTAB	0	0	192.168.1.164:ssh	192.168.1.213:50275	
	ESTAB	0	0	192.168.1.164:ssh	192.168.1.213:50228	
Process	ESTAB	0	0	192.168.1.164:ssh	192.168.1.213:49895	
	LISTEN	0	128	[::]:ssh	[::]:*	
uname	LISTEN	0	5	[::1]:ipp	[::]:*	
hwclock	LISTEN	0	100	[::]:1883	[::]:*	
devices						
CPU Freq						
Isusb T V						
uptime						
ifconfig Iscpu						
date ip route ss s l t u Isof netstat						

# Ss –u : linux command : ss –u –a

ptions					- 🗆
ral   Time Setting   LA	N Setting Stats	License Kernel			
CPU Info	State Re	cv-Q Send-Q	Local Address:Port	Peer Address:Port	Process
	UNCONN 0	0	127.0.0.53%lo:domain	0.0.0:*	
Mem Into	ESTAB O	0	192.168.1.164%eth1:bootpc	192.168.1.1:bootps	
Linux Version	UNCONN 0	0	0.0.0:37990	0.0.0:*	
LINUX TOISION	UNCONN 0	0	0.0.0:mdns	0.0.0:*	
Flash Info	UNCONN O	0	[::]:mdns	[::]:*	
Erec men	UNCONN 0	0	[::]:46996	[::]:*	
Fiee mem					
Process					
uname					
hwclock					
devices					
CPU Freq					
susb T V					
uptime					
onfig Iscpu					
te ip route					
s s l t u					
sof netstat					

```
Lsof : linux command : lsof
```

Options					-		×
eneral Time Setting	LAN Setting Stats	License Kernel					
		·					
CPU Info	COMMAND	PID TID TASKCMI	USER	FD TYPE	DEVICE	SIZE/OFF	
	systemd	1	root	cwd DIR	179,2	4096	
Mem Info	systemd	1	root	rtd DIR	179,2	4096	
Linux Massian	systemd	1	root	txt REG	179,2	1075228	
LINUX VEISION	systemd	1	root	mem REG	179,2	112216	
Flash Info	systemd	1	root	mem REG	179,2	1422692	
	systemd	1	root	mem REG	179,2	83464	
Free mem	systemd	1	root	mem REG	179,2	38400	
	systemd	1	root	mem REG	179,2	26020	
Process	systemd	1	root	mem REG	179,2	306564	
[	systemd	1	root	mem REG	179,2	21968	
uname	systemd	1	root	mem REG	179,2	1675680	
hwclock	systemd	1	root	mem REG	179,2	13824	
	systemd	1	root	mem REG	179,2	9772	
devices	systemd	1	root	mem REG	179,2	403240	
	systemd	1	root	mem REG	179,2	116964	
CPU Freq	systemd	1	root	mem REG	179,2	362672	
	systemd	1	root	mem REG	179,2	104024	
	systemd	1	root	mem REG	179,2	87428	
untime	systemd	1	root	mem REG	179,2	17988	
aptine	systemd	1	root	mem REG	179,2	107840	
ifconfig Iscpu	systemd	1	root	mem REG	179,2	668232	
	systemd	1	root	mem REG	179,2	267424	
date ip route	systemd	1	root	mem REG	179,2	185768	
	systemd	1	root	mem REG	179,2	17888	
ss s l t u	systemd	1	root	mem REG	179,2	219056	
Isof netstat	svstemd	1	root	mem REG	179.2	21864	
(	1						

#### Netstate: Linux command : netstat

Options							- 🗆	×
eneral   Time Setting   LAN	Setting Stat	S License Ke	ernel					
CPU Info	Active 1	nternet con	nections (w/o se	rvers)				
Mem Info	Proto Re	cv-Q Send-Q 0 0	Local Address cs6202.lan:ssh	1	Foreign Addres kamiooT420.lan	s :50228	State ESTABLISHED	- 1
Linux Version	tcp		cs6202.lan:ssh cs6202.lan:ssh	1	kamjooT420.lan kamjooT420.lan	:50281	ESTABLISHED ESTABLISHED	1
Flash Info	tcp udp	0 112	cs6202.lan:ssh cs6202.lan:boot	pc 1	kamjooT420.lan RUT200.lan:boo	:50282 tps	ESTABLISHED ESTABLISHED	
Free mem	Active U Proto Re	NIX domain : fCnt Flags	sockets (w/o ser Type	vers) State	I-Node	- Path		
Process	unix 3	[]	DGRAM		13634	/run/sys	stemd/notify	
uname	unix 2	[]	DGRAM		16298	/run/sys	stemd/journal/sys	31
hwclock	unix 18 unix 10		DGRAM DGRAM		16310 16317	/run/sys /run/sys	stemd/journal/dev stemd/journal/soc	r- sk
devices	unix 2 unix 2	[]	DGRAM DGRAM		155863 22519	/run/use /run/wps	er/O/systemd/noti a supplicant∕wlar	.f 10
CPU Freq	unix 2	[]	DGRAM	CONNECTE	22524 27800	/run/wpa	a_supplicant/p2p-	-d
Isusb T V	unix 3	[]	STREAM	CONNECTE	D 27000 D 77247 D 27323	/ L ull/ 5 y	scena, journar, sco	.0
uptime	unix 3	[]	STREAM	CONNECTE	D 28753	/run/use	er/110/bus	
ifconfig Iscpu	unix 3 unix 3	[]	STREAM STREAM	CONNECTE: CONNECTE:	D 26305 D 26292	/run/use	er/110/bus	
date ip route	unix 3 unix 3	[] []	STREAM STREAM	CONNECTE: CONNECTE:	D 21610 D 26272	/run/sys /run/sys	stemd/journal/sto stemd/journal/sto	lo lo
ss s l t u	unix 3 unix 3	[]	DGRAM STREAM	CONNECTE	27291 D 21000	/run/db	us/svstem bus soc	sk