PYRAMID
www.pyramid.tech

IGX

IGX - User Manual

Document ID: 2753757191

Version: v4

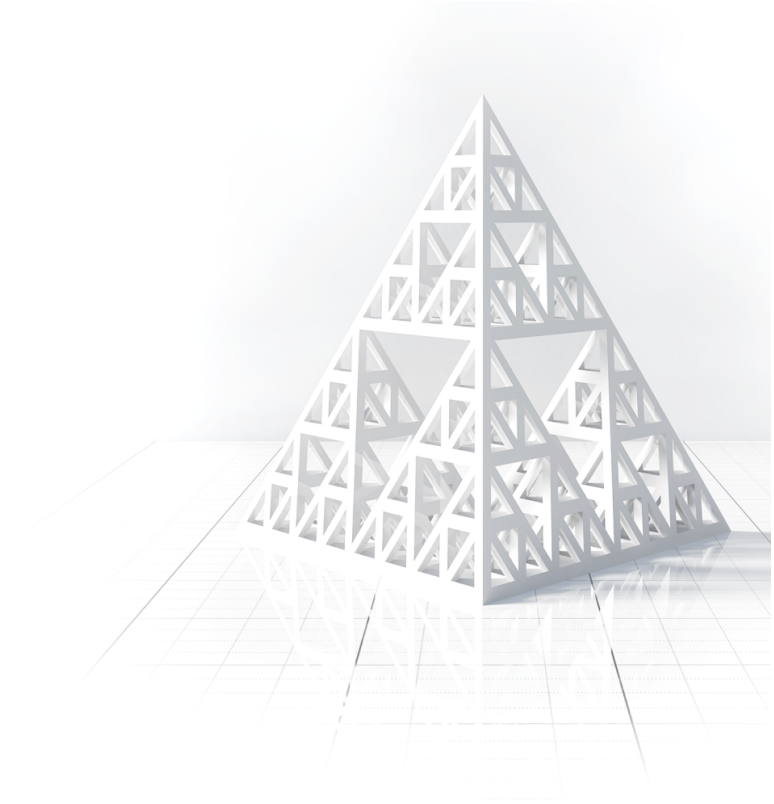


Table of Contents

1 Introduction 4
1.1 IGX Control System Framework 4
2 Document Control 5
2.1 Version History 5
2.2 Approvals 5
2.2.1 Signatures 5
3 IGX Functionality and Features 6
3.1 IGX Network Connection Guide 6
3.1.1 Network Introduction 6
Quick Start 6
3.1.2 IP Networking Basics 7
3.1.3 Setting your Windows 10 IP Address 8
Using DHCP (Automatic IP Configuration) 8
Using Manual Static IP Configuration 8
3.2 IGX Interlock and Permit User Guide 9
3.2.1 Introduction to the Interlock and Permit System 9
3.2.2 Interlocks 9
Interlock Value 10
Interlock Types 10
3.2.3 Permits 11
3.2.4 Reporting and Auditing 11
3.3 IGX Permission Policies and Configuration Control 11
3.3.1 Permission Policy Write Permit 13
3.3.2 Permission Policy Hash 13
3.3.3 Default Permission Policies 13
Develop Policy 13
Protected Policy 14
Prescription Policy 14
3.4 IGX Expression Language Guide 14
3.4.1 Introduction to ExprTk Language 14
Arithmetic Operators 14
Built-in Functions 16
Conditional Expressions 18
3.5 IGX SSH and SFTP Guide 19
3.5.1 Introduction 19

Secure Shell (SSH) Protocol 19
Secure File Transfer Protocol (SFTP) 19
Further Reading 19
3.5.2 Accessing IGX Devices via SSH 19
3.5.3 Accessing IGX Devices via SFTP 20
3.5.4 Basic QNX Console Commands 20
Starting and Stopping the IGX Application 21
4 IGX User-Managed System Care 22
4.1 IGX Firmware Update Guide 22
4.1.1 Introduction 22
Identify the Firmware File 22
Uploading the Firmware 23
Post-Firmware Update 24
Update Validation and Troubleshooting 25
4.2 IGX SD Card Flashing Guide 25
4.2.1 Introduction 25
4.2.2 Get the Micro SD Card 25
4.2.3 Get the Desired Image File 25
4.2.4 Get the Required Tool 25
4.2.5 Flash the Card 26
4.2.6 Card Installation or Removal 26
4.2.7 Reboot the Device 27
4.3 IGX SD Card Imaging Guide 27
4.3.1 Summary 27
4.3.2 Linux Procedure 27
4.3.3 Windows 10 Procedure 28
4.4 Modifying the IGX System XML File 28
4.4.1 Device Nodes 30
4.5 BeagleBone Black Replacement Guide 30
4.5.1 Purpose 30
Procuring a BeagleBone Black 30
Replacement Procedure 31
4.6 Running a Local NTP Time Server 33
4.6.1 Overview 33
4.6.2 Steps to Configure Windows as an NTP Server 33
Verify Configuration 34
Additional Configuration for NTP Clients 34

1 Introduction

Document ID: 2649784673

Author	@Matthew Nichols
Owner	Project Lead
Purpose	Provide clear, concise, and comprehensive information for the safe and effective use, maintenance, and troubleshooting of IGX based products.
Scope	The use of an IGX product by an end user.
Intended Audience	IGX end-users
Process	Standard Manual Creation Process
Training	NOT APPLICABLE

1.1 IGX Control System Framework

IGX is Pyramid's latest software control system framework. It's designed from the ground up for high-reliability applications including medical and light industrial settings. The framework is monolithic and modular, meaning that it comes with many features by default and that the features are all encapsulated such that they can be reused and extended easily. Some key features are:

- Battle proven real-time QNX operating system with highly reliable microkernel.
- Custom web server with low latency and high bandwidth.
- Harmonized IO data available over multiple Ethernet protocols.
- Powerful data acquisition and processing tools.
- Built-in and user configurable safety interlocking system, allows for automatic safety actions.
- User definable expression language for simple scripting.

2 Document Control

2.1 Version History

Version	Description	Saved by	Saved on	Status
v4	Added chapters for NTP servers and permission policies.	Matthew Nichols	Jun 3, 2024 9:32 PM	APPROVED
v3	Fixed conditional expression documentation.	Matthew Nichols	Apr 19, 2024 9:31 PM	APPROVED
v2	SD card imaging instructions and expression language documentation.	Harvey Jules Nett	Apr 1, 2024 3:51 PM	APPROVED
v1	Initial release	Harvey Jules Nett	Dec 1, 2023 5:51 PM	OUTDATED

2.2 Approvals

This document has been reviewed and approved as follows.

Document Control

Current document version: v.1

No reviewers assigned.

2.2.1 Signatures

Monday, Jun 3, 2024, 09:34 PM UTC, (v. 1)
Matthew Nichols signed with meaning **Review**

3 IGX Functionality and Features

3.1 IGX Network Connection Guide

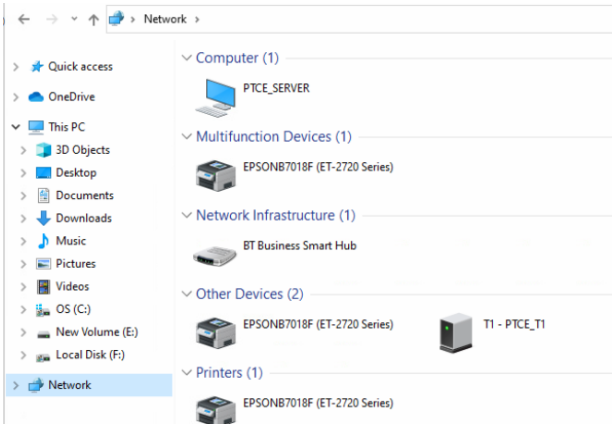
3.1.1 Network Introduction

The IGX framework utilizes Ethernet based communication as the primary medium for data exchange to the outside world. It's used for GUIs, control systems, databasing, scripting, and more. The most common form of communication used by IGX is the HTTP protocol, which is what is used to service the embedded GUI interfaces.

In order to connect with an IGX device, you must have an understanding of basic Ethernet network concepts. This guide will take you through the step-by-step process of connecting to IGX devices and maintaining larger networks with multiple Pyramid and third-party products.

Quick Start

If you are already familiar with network concepts, then you can jump right into using IGX devices. Our products come with DHCP enabled and will attempt to automatically get assigned an IP address if possible. If there is no response to the DHCP request, the device will fall back to the static IP address, which is `192.168.100.20` with a netmask of `255.255.255.0` by default. Once the device has an IP address, it will be discoverable on your network using UPnP, Windows 10 comes with a discovery tool built into the file explorer.



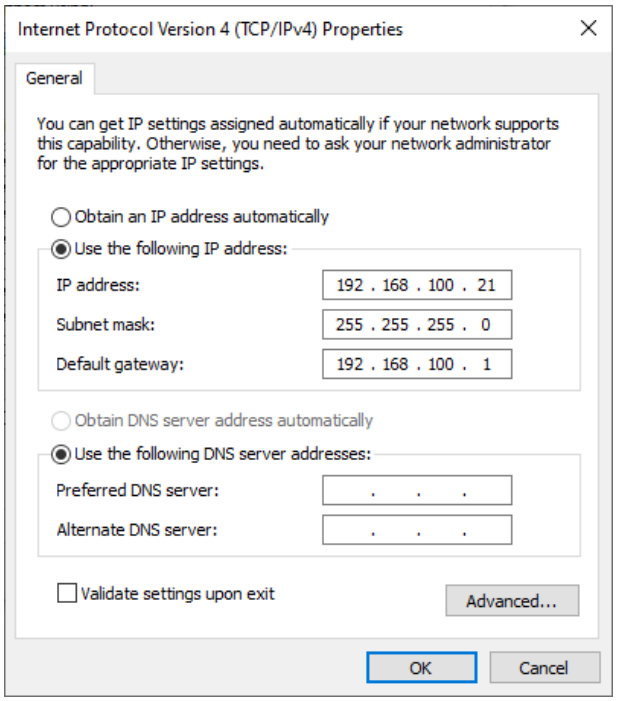
1 Pyramid devices are found under "Other Devices". The device name is customizable.

Make sure network discovery is turned on. You may need to press the refresh button if the device has been added very recently.

Once you see the device icon in the Network page, double-click it to open a new page on your web browser with the device's URL.

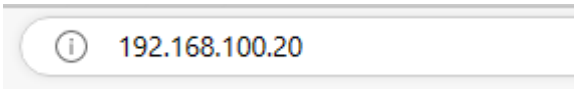
You can then log in and start controlling the device's settings and collecting data.

If the previous steps don't work, or you want to explicitly connect directly to the IGX device, you'll need to configure your computer's IP address settings to cooperate with the device's settings.



2 Windows Ethernet Adaptor settings, note that the computer's assigned IP address is different from the device's IP address.

Refer to your operating system's documentation for specific instructions on how to do this. Once you've set your computer's Ethernet adapter to the right subnet, type the device's IP address directly into the address bar of a new tab on your web browser to open the web interface. You can then log in and start controlling the device's settings and collecting data.



3 Browsers let you directly connect to the device's web GUI.

Note that depending on your device's specific settings and your network configuration, you may need to perform additional configuration steps to ensure that your device is properly connected and configured to work with your computer. Refer to your device's documentation or user guide for more information on how to set up and connect to your device.

3.1.2 IP Networking Basics

Internet Protocol (IP) networks are the foundation of modern digital communication, enabling devices to exchange data over a shared network. IP networks facilitate communication among various devices like computers, smartphones, printers, and of course IGX devices, allowing them to share resources and information.

Here are the basics of IP networks and what a typical user needs to know:

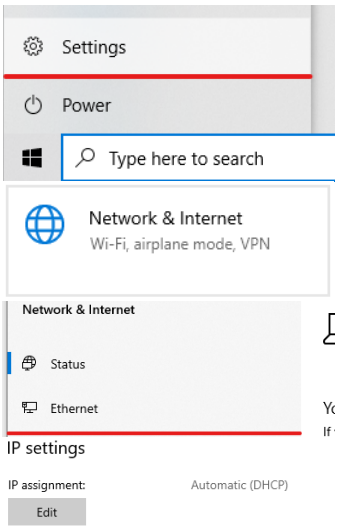
1. **IP Address**: Each device on an IP network is assigned a unique identifier called an IP address. It is used to identify and locate devices on the network.
   There are two versions of IP addresses: IPv4 and IPv6. IPv4 addresses are written as four sets of numbers separated by periods (e.g., 192.168.100.20), while IPv6 addresses use eight sets of four hexadecimal characters separated by colons (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
2. **Subnet Mask**: A subnet mask is a number that defines the range of IP addresses within a network. It helps devices determine whether they are on the same network or a different one. Subnet masks are written in the same format as IP addresses. In order for one computer to talk to another computer without an advanced router scheme, they both must have the same subnet mask.
3. **Routers**: Routers are devices that connect multiple networks and direct data packets between them. They use IP addresses and subnet masks to determine the best path for forwarding the packets. Home routers also often act as Wi-Fi access points, enabling wireless devices to connect to the network.
4. **DHCP**: Dynamic Host Configuration Protocol (DHCP) is a network service that automatically assigns IP addresses and other network settings to devices as they join the network. This simplifies the process of connecting devices and ensures that they have the correct settings.
5. **DNS**: Domain Name System (DNS) is a service that translates human-readable domain names (e.g., www.example.com[1]) into IP addresses. This makes it easier for users to access websites and services without having to remember their numerical IP addresses.

1 http://www.example.com/

3.1.3 Setting your Windows 10 IP Address

This guide will walk you through the steps to set up Ethernet settings on Windows 10 for both Dynamic Host Configuration Protocol (DHCP) and manual static IP configurations.

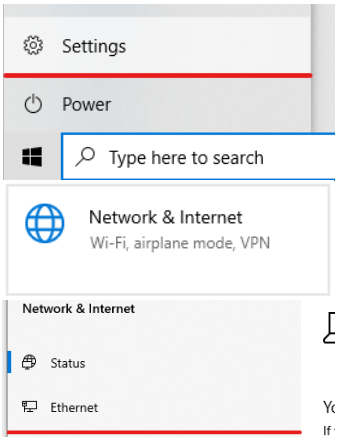
Using DHCP (Automatic IP Configuration)



1. Click on the "Start" button (Windows icon) in the lower-left corner of your screen and select "Settings" (the gear icon).
2. In the "Settings" window, click on "Network & Internet."
3. On the left-hand menu, click on "Ethernet," then select your Ethernet connection from the list on the right.
4. Scroll down to the "IP settings" section and click on "Edit."
5. In the "Edit IP settings" window, select "Automatic (DHCP)" from the drop-down menu, then click on "Save."

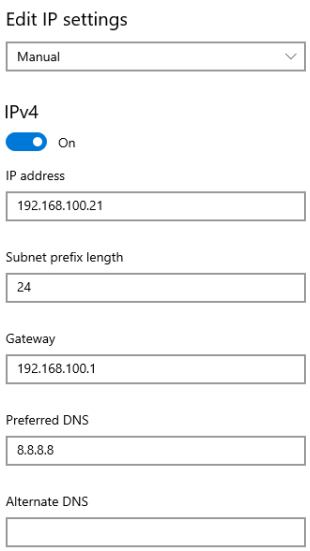
Windows 10 will now automatically obtain an IP address, subnet mask, default gateway, and DNS server addresses from your network's DHCP server. This is frequently a centralized router managed by an IT department or office administrator.

Using Manual Static IP Configuration



1. Click on the "Start" button (Windows icon) in the lower-left corner of your screen and select "Settings" (the gear icon).
2. In the "Settings" window, click on "Network & Internet."
3. On the left-hand menu, click on "Ethernet," then select your Ethernet connection from the list on the right.
4. Scroll down to the "IP settings" section and click on "Edit."
5. In the "Edit IP settings" window, select "Manual" from the drop-down menu.
6. You will now see options for "IPv4" and "IPv6." Toggle the switch to "On" for the IP version you want to configure (usually IPv4).
7. Enter the following information, which can be obtained from your network administrator or ISP:
   - IP address: The static IP address you want to assign to your device (192.168.100.21 if trying to do a direct connection to 192.168.100.20).
   - Subnet mask or prefix length: The subnet mask for your network (usually 255.255.255.0 for basic networks). Prefix length serves the same purpose as the mask but with a different format. Use “24” for 255.255.255.0.
   - Gateway: The IP address of your router or default gateway. This is optional, and only needed if there is a need for internet access and multiple network hops.

   Note: Make sure the IP address you assign is unique and not used by another device on your network.



4 New Windows 10 interface

8. Scroll down to the "Preferred DNS" and "Alternate DNS" fields. Enter the DNS server addresses provided by your ISP or network administrator. If you don't have specific DNS server addresses, you can use public DNS servers like Google's (8.8.8.8 for Preferred DNS and 8.8.4.4 for Alternate DNS). This is only required if internet connections are needed.
9. Click on "Save" to apply the changes.

Windows 10 will now use the static IP address, subnet mask, default gateway, and DNS server addresses you entered for your Ethernet connection.

Remember to verify your settings with your network administrator or ISP to ensure proper network connectivity. If you encounter any issues or need to revert to DHCP, follow the steps for "Using DHCP (Automatic IP Configuration)" to switch back.

3.2 IGX Interlock and Permit User Guide

3.2.1 Introduction to the Interlock and Permit System

This guide provides an overview of the IGX interlock and permit system, which ensures the correct and safe operation of the IGX system.

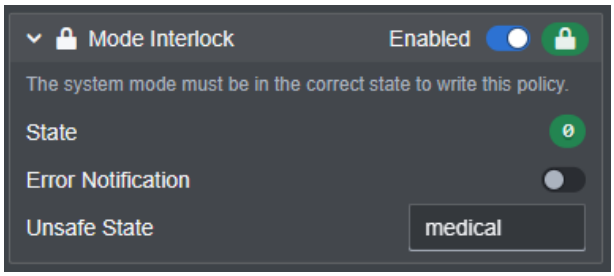
Software-based interlocks are a type of safety mechanism used in industrial environments to prevent accidents, ensure proper operation, and maintain system integrity. They are designed to coordinate the operation of various equipment, processes, or systems by enforcing predefined conditions or rules. In contrast to hardware-based interlocks, which rely on physical components such as switches or relays, software-based interlocks are implemented using a microprocessor and programmable logic. Frequently software-based interlocks are cheaper and easier to maintain than hardware-based interlocks, but they rarely eliminate the need for hardware-based interlocks entirely.

In real-world scenarios, these interlock logic conditions must be regularly monitored and adjusted according to specific situations. This software interlock safety system offers several features:

- Configure interlock parameters through a GUI with descriptive text.
- Enable or disable interlocks, allowing temporary or permanent override of interlocks.
- Create latching interlocks that remain in error until disabled and re-enabled.
- Adjust interlock condition parameters or limits.
- Combine multiple interlocks into a single permit, allowing the operation of protected outputs (e.g., enabling high voltage or radiation sources).

3.2.2 Interlocks

Interlocks are responsible for checking their conditions and reporting their status. To execute any action in response to an interlock, a permit or additional software is required. Interlocks are IGX IO with an integer type.



This interlock detects when the IGX system is in medical mode and prevents otherwise unsafe actions from occurring.

From this GUI interface, you can enable or disabled the interlock and configure the interlock's parameters.

Interlock Value

The interlock value is defined by an integer value corresponding to a particular meaning.

Value	State	Meaning
`0`	OK	Interlock is OK, all error and warning conditions are unmet.
`1`	ERROR	Interlock is in error; the error condition has been met.
`2`	WARNING	Interlock is in warning; the warning condition has been met. Not all interlocks will have a possible warning condition.

As of this writing, there are 3 values defined, however the system may be extended to support more values. Future versions will add these new values to the end of the list, and use new numbers, leaving the old meanings unchanged.

Interlock Types

There are several types of interlocks built into the IGX framework. These pre-made types allow for highly uniform and predictable condition checking that is robust and well tested.

Range Interlocks

These interlocks will check if a given numeric IO is within optional range limits. All of the following parameters are optional. If the parameter is omitted, it will not be used in the conditional checking logic.

Parameter	Meaning
Upper Limit	The value that if exceeded will cause an error.
Lower Limit	The value that if not exceeded will cause an error.
Upper Warning	The value that if exceeded will cause a warning.
Lower Warning	The value that if not exceeded will cause a warning.

These limits will be defined using the same units of the IO that it is checking against.

Tolerance Interlock

These interlocks will check if a given numeric IO is within acceptable tolerance limit of a given command IO. The tolerance is defined as a percentage. For example, you may want a readback input to match a given command output within 5%.

Parameter	Meaning
Minimum	The minimum absolute value that the command must have to enable the conditional checking. For example, a command of 0 should not count, as it will always be out of tolerance.
Tolerance Limit	The value of percentage error that if exceeded will cause an error.
Tolerance Warning	The value of percentage error that if exceeded will cause a warning.

The minimum is defined in the units of the command IO. The tolerance values are always defined as a percentage.

3.2.3 Permits

Permits are responsible for checking the status of their child interlocks and reporting the combined state. They can be in one of two possible states: granted or revoked. A granted permit will have a Boolean value of true, while a revoked permit will have a Boolean value of false.

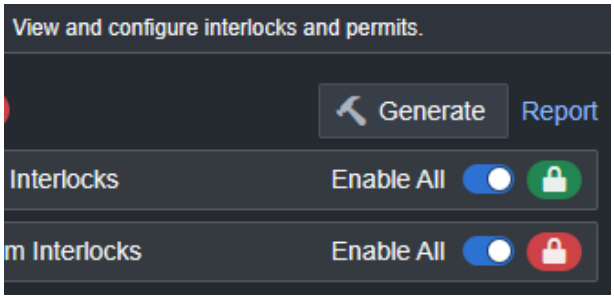
Value	State	Meaning
`true`	GRANTED	Permit is granted. All interlocks are OK or in warning.
`false`	REVOKED	Permit is revoked. At least on interlock is in error.

3.2.4 Reporting and Auditing

It is critical to system safety to regularly conduct interlock parameter audits to ensure that the configuration is still meeting the requirements. IGX includes a system for generating interlock reports in the form of a CSV file containing all the relevant fields.

IGX will have a nested system of interlock managers which each have their own scope that they can report on. In order to generate a report of the global interlock scope, navigate to the following URL `https://<ip>/io/interlocks`.

To generate an interlock report, use the following steps.



5 Interlock report interface

First click the “Generate” button to create a new report.

Then click the report link to download the newly generated report.

Once the report is on your system you can open it using Excel to see the values.

3.3 IGX Permission Policies and Configuration Control

Within the IGX system, each IO may be optionally linked to a distinct permission policy. This policy is designated by a unique identifier in the form of a simple string. It includes a range of critical features that bolster the management and safeguarding of the system's IO activities.

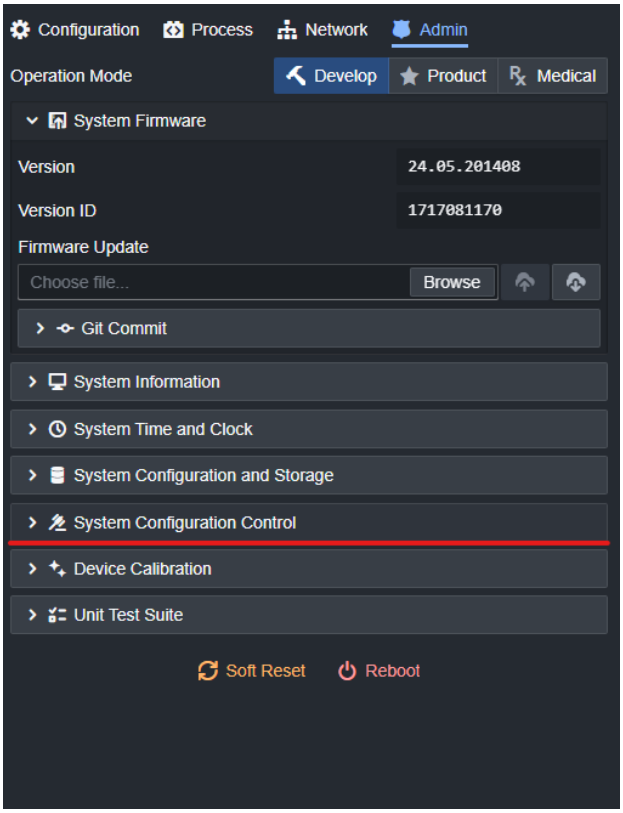
Key Features of IO Permission Policies

1. **Write Permit Control**: Each permission policy includes a write permit. This write permit dictates whether the IO operations governed by the policy are writable. By controlling the write permissions systematically, the system ensures consistent and secure write operations across all IO.

2. **Systematic Write Control**: The inclusion of write permits in the permission policies provides a structured approach to manage write access. This systematic control mechanism ensures that write operations are only performed when authorized, thereby maintaining the integrity and security of the control system.
3. **IO State Monitoring**: Beyond controlling write operations, the permission policy maintains a hash of all IO assigned to it. This hash acts as a comprehensive record, enabling both internal and external systems to monitor the state of all associated IO.
4. **Configuration Consistency**: The hash maintained by the permission policy ensures that the system configuration remains consistent and in the expected state. This eliminates the need for monitoring each IO individually, thereby simplifying the monitoring process and enhancing system reliability.
5. **External and Internal System Integration**: The hashed record of IO states facilitates seamless integration with both internal and external monitoring systems. This integration ensures that any discrepancies or unauthorized changes in the system configuration can be promptly detected and addressed.

The GUI for monitoring and controlling permission policies and configuration control can be found under the admin settings under the “System Configuration Control” section.

Here you will find a list of all available permission policies on the system and their current status.



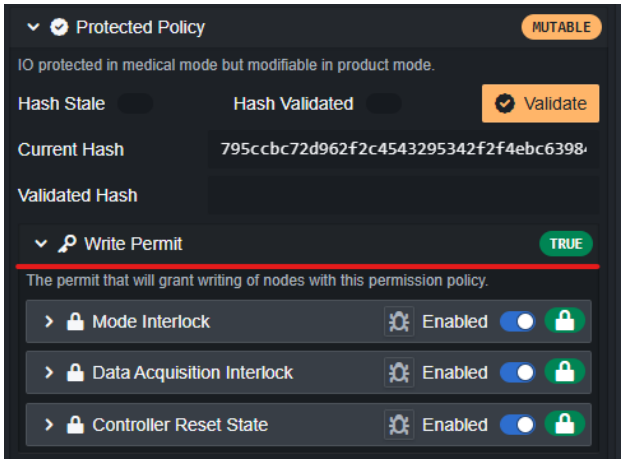
6 Location of Permission Policy Control GUI

3.3.1 Permission Policy Write Permit

Each permission policy has a write permit associated with it. When this permit is granted (true) then the IO are writable, assuming nothing else is preventing it. If the permit is revoked (false) then the IO can only be read.

Different policies will have different sets of configurable interlocks that drive this behavior. These interlocks will observe the system state and react to changes.

See the IGX Interlock and Permit User Guide (see page 9) for more information on how permits and interlocks work.

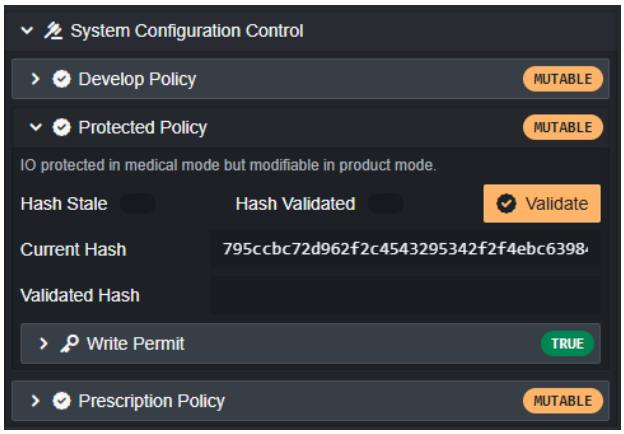


7 Permission Policy Write Permit

3.3.2 Permission Policy Hash

Each permission policy has a SHA-256 hash of all the current IO values under that policy. To ensure smooth performance, the hash is not re-calculated every time any IO changes. Instead, the hash is immediately marked as “stale” and later re-calculated. This ensures that even with heavy IO modifications the CPU will not be burdened with unnecessary hash calculations.

Optionally a user can “validate” a hash by pressing the Validate button. This will save the current hash in the Validated Hash IO and set the Hash Validated IO to true. If the current hash every diverges from the validated hash, Hash Validated will be set back to false.



8 Permission Policy Hash GUI

3.3.3 Default Permission Policies

IGX systems come with a default set of permission policies. Some more advanced control systems will deploy with additional policies. See your specific product user manual in these cases.

Develop Policy

The develop policy is intended to be used on IO that are only mutable during development of the product. These IO are obscure IGX parameters that are difficult to understand and potentially dangerous to change but must still be changeable under strict control. Typically, only Pyramid employees change these IO, however sometimes users may change these IO to help repair broken devices in the field under Pyramid’s supervision. Some examples are:

- Physical device calibration parameters
- Device serial numbers and identifiers
- Safety limits for integral safety components
- Settings related to manufacturing testing

This policy is only writable if the system is in the “develop” operation mode.

Protected Policy

The protected policy is the most commonly used policy. It prevents unwanted modifications to the majority of user configurable parameters on a device or control system. This policy write permit has an interlock which will make the policy unwritable if the system is in “medical” mode. All interlock parameter IO will use this policy by default.

In medical or high-risk industrial systems, these are the parameters that are desirable to “lock down” after they have been tuned to a desired configuration. Some examples are:

- The input range of an electrometer or programmable gain amplifier
- Configurable offsets or user scale factors
- Conversion and sampling frequencies for data acquisition systems
- Expression models for converting dose to charge
- High voltage power supply configuration settings
- The majority of interlock configuration systems
- Process controller configurations

Prescription Policy

The prescription policy is a less used, but still vital policy. It is used for prescriptive IO that will change from session to session of a data acquisition or process controller and so should not be restricted from being written when the operational mode is set to “medical”.

These IO will not be writable after the session starts and only become writable again once the session has been reset. If a process controller is not in the ready state or there is an active data acquisition session, the policy's write permit will be revoked. Some examples are:

- The dose or charge prescription on a dose controller
- The control point table in a dose or motion controller
- Some interlock parameters that have been configured to be prescriptive

3.4 IGX Expression Language Guide

3.4.1 Introduction to ExprTk Language

ExprTk is a powerful and flexible mathematical expression parsing and evaluation library for C++. It allows users to create, manipulate, and evaluate complex mathematical expressions through a simple and intuitive language. In this guide, we'll provide an overview of how you can use the ExprTk language to create expressions, incorporate variables, and use built-in functions and operators.

Arithmetic Operators

Expressions can be built using a variety of built in operators. The following is a list of the mathematical and logical operators. These operators evaluate into a single numerical value.

Arithmetic Operators

Operator	Description
`+`	Addition: Adds two values together.
`-`	Subtraction: Subtracts the second value from the first value.
`*`	Multiplication: Multiplies two values together.
`/`	Division: Divides the first value by the second value.

Operator	Description
`%`	Modulus: Returns the remainder of the division of the first value by the second value.
`^`	Exponentiation: Raises the first value to the power of the second value.

Equalities and Inequalities Operators

The following are operators for checking equalities or inequalities. The results of these operators will always be 0 for false and 1 for true.

Operator	Description
`==` or `=`	Equality: Checks if two values are equal.
`!=` or `<>`	Inequality: Checks if two values are not equal.
`<`	Less Than: Checks if the first value is less than the second value.
`<=`	Less Than or Equal To: Checks if the first value is less than or equal to the second value.
`>`	Greater Than: Checks if the first value is greater than the second value.
`>=`	Greater Than or Equal To: Checks if the first value is greater than or equal to the second value.

Logical Operators

The following are logical operators. The results of these operators will always be 0 for false and 1 for true.

Operator	Description
`true`	True state or any value other than zero, typically 1.
`false`	False state, value of exactly zero.
`and`	Logical AND, True only if x and y are both true. Example: `x and y`
`&`	Similar to AND but with left to right expression short circuiting optimization.
`nand`	Logical NAND, True only if either x or y is false. Example: `x nand y`
`mand(x, y, ...)`	Multi-input logical AND, True only if all inputs are true. Left to right short-circuiting of expressions. Example: `mand(x > y, z < w, u or v, w and x)`
`or`	Logical OR, True if either x or y is true. Example: `x or y`
`|`	Similar to OR but with left to right expression short circuiting optimization.
`nor`	Logical NOR, True only if the result of x or y is false. Example: `x nor y`

Operator	Description
`mor(x, y, ...)`	Multi-input logical OR, True if at least of the inputs are true. Left to right short-circuiting of expressions. Example: `mor(x > y, z < w, u or v, w and x)`
`xor`	Logical XOR, True only if the logical states of x and y differ. Example: `x xor y`
`xnor`	Logical XNOR, True iff the biconditional of x and y is satisfied. Example: `x xnor y`
`not(x)`	Logical NOT, Negate the logical sense of the input.

Variable Assignment Operators

In addition to these operators, there are also a selection of assignment operators to pick from. Unlike the operators above, these will assign the evaluated value to a variable on the lefthand side of the operator.

Operator	Description
`:=`	Assignment: Assigns the value on the right to the variable on the left.
`+=`	Add and Assign: Adds the value on the right to the variable on the left, and assigns the result to the variable on the left.
`-=`	Subtract and Assign: Subtracts the value on the right from the variable on the left, and assigns the result to the variable on the left.
`*=`	Multiply and Assign: Multiplies the value on the right with the variable on the left and assigns the result to the variable on the left.
`/=`	Divide and Assign: Divides the variable on the left by the value on the right and assigns the result to the variable on the left.
`%=`	Modulus and Assign: Takes the modulus of the variable on the left by the value on the right and assigns the result to the variable on the left.

Built-in Functions

ExprTk comes with a wide range of built-in functions and operators to help you create more complex expressions. These include trigonometric functions (e.g., `sin`, `cos`, `tan`), logarithmic functions (e.g., `log`, `log10`, `exp`), and many others. To use a function, simply write its name followed by its arguments in parentheses. For example: `sin(x) * cos(y)`. Some functions accept a variable number of arguments. These functions will be defined with an argument list ending in `...` to show that more arguments can be added.

ExprTk library supports a wide range of built-in functions for various mathematical operations and calculations. Here's a table listing some of the commonly used built-in functions in ExprTk along with a brief description of their behavior:

General Purpose Functions

Function	Description
`abs(x)`	Returns the absolute value of x.
`sgn(x)`	Sign of x, -1 where x < 0, +1 where x > 0, else zero.

Function	Description
`sqrt(x)`	Returns the square root of x.
`root(x, n)`	Nth-Root of x. where n is a positive integer.
`pow(x, y)`	Raises x to the power of y (x^y).
`exp(x)`	Returns the exponential function of x (e^x).
`log(x)`	Returns the natural logarithm (base e) of x.
`log10(x)`	Returns the logarithm (base 10) of x.
`log2(x)`	Returns the logarithm (base 2) of x.
`logn(x, n)`	Base N logarithm of x. where n is a positive integer.
`ceil(x)`	Smallest integer that is greater than or equal to x.
`floor(x)`	Largest integer that is less than or equal to x.
`round(x)`	Round x to the nearest integer.
`roundn(x, n)`	Round x to n decimal places where n > 0 and is an integer. Example: `roundn(1.2345678, 4) == 1.2346`
`frac(x)`	Fractional portion of x.
`trunc(x)`	Integer portion of x.
`max(x, y, ...)`	Largest value of all the inputs.
`min(x, y, ...)`	Smallest value of all the inputs.
`sum(x, y, ...)`	Sum of all the inputs.
`mul(x, y, ...)`	Product of all the inputs.
`avg(x, y, ...)`	Average of all the inputs.
`hypot(x, y)`	Hypotenuse of x and y. `hypot(x, y) = sqrt(x*x + y*y)`
`erf(x)`	Error function of x.
`erfc(x)`	Complimentary error function of x.
`ncdf(x)`	Normal cumulative distribution function.

Trigonometry Functions

Function	Description
`acos(x)`	Returns the arc cosine of x in radians.
`asin(x)`	Returns the arc sine of x in radians.
`atan(x)`	Returns the arc tangent of x in radians.
`atan2(x, y)`	Returns the arc tangent of y/x in radians.
`cos(x)`	Returns the cosine of x, where x is in radians.
`cosh(x)`	Returns the hyperbolic cosine of x.
`sin(x)`	Returns the sine of x, where x is in radians.
`sinh(x)`	Returns the hyperbolic sine of x.
`tan(x)`	Returns the tangent of x, where x is in radians.
`tanh(x)`	Returns the hyperbolic tangent of x.

Conditional Expressions

If statements have multiple possible syntax that can be used.

The functional form `if (x, y, z)`, uses three arguments, `x`, `y`, and `z`, where if `x` is true, `y` is returned and otherwise `z` is returned. For example:

```
if (x > 10, 5, 6)
```

If x is greater than 10, the expression resolves to 5 and otherwise it resolves to 6.

A more conventional form, where the branching expressions are defined after the if statement, with optional curly brackets, can also be used. For example:

```
if (x) z;

if (x) { z; }
```

If, else, and else if statements can also be used. For example:

```
if (condition ...)
{
  ...
}
else if (alternate condition ...)
{
  ...
}
else
{
  ...
}
```

3.5 IGX SSH and SFTP Guide

3.5.1 Introduction

IGX devices run the QNX operating system and can be accessed via standard SSH and SFTP protocols. The default login credentials for IGX devices are username: `root` and password: `root`. This guide provides step-by-step instructions on how to access IGX devices via SSH and SFTP using the command prompt in Windows 10 and WinSCP. Additionally, it covers basic QNX console commands and how to start and stop the IGX application.

Secure Shell (SSH) Protocol

The SSH protocol is a secure network protocol used for remote access to computers. It is designed to provide secure communication between two untrusted networks by using encryption to protect the contents of the communication. SSH is widely used by network administrators to manage remote systems and by developers to access remote development environments.

The SSH protocol uses the TCP/IP protocol suite for communication, and by default, uses port 22. The protocol supports several authentication methods, including password-based authentication, public key authentication, and host-based authentication. SSH also supports tunneling of TCP/IP connections, allowing applications to securely communicate over the SSH connection.

The SSH protocol is defined in several RFCs, including RFC 4250 (The Secure Shell (SSH) Protocol Assigned Numbers), RFC 4251 (The Secure Shell (SSH) Protocol Architecture), and RFC 4253 (The Secure Shell (SSH) Transport Layer Protocol).

Secure File Transfer Protocol (SFTP)

The Secure File Transfer Protocol (SFTP) is a secure alternative to the File Transfer Protocol (FTP) for transferring files between computers. SFTP is based on the SSH protocol and uses the same encryption and authentication mechanisms to protect the contents of the communication.

SFTP uses the TCP/IP protocol suite for communication and typically uses port 22, the same port as SSH. SFTP supports several operations, including file upload, file download, and directory listing. SFTP also supports resume of interrupted transfers and can preserve file attributes such as modification time, permissions, and ownership.

The SFTP protocol is defined in several RFCs, including RFC 913 (FTP server extension), RFC 913 (FTP client extension), and RFC 4253 (The Secure Shell (SSH) Transport Layer Protocol).

Further Reading

If you're interested in learning more about the SSH and SFTP protocols, here are some resources to get you started:

- SSH Protocol Specification: IETF RFC2450[2]
- SFTP Protocol Specification: IETF RFC913[3]
- OpenSSH: https://www.openssh.com/
- PuTTY: https://www.chiark.greenend.org.uk/~sgtatham/putty/
- WinSCP: https://winscp.net/eng/docs/start

3.5.2 Accessing IGX Devices via SSH

To access an IGX device via SSH, follow these steps:

1. Open the Command Prompt on Windows 10 by typing `cmd` into the Start menu.
2. In the Command Prompt, type the following command to connect to the IGX device via SSH:

2 https://www.ietf.org/rfc/rfc4250.txt
3 https://www.ietf.org/rfc/rfc913.txt

```
ssh root@<ip_address>
```

Replace `<ip_address>` with the IP address of the IGX device you wish to connect to.

3. When prompted, enter the password for the root user (default password is `root`).

You should now be connected to the IGX device via SSH and can execute QNX commands.

3.5.3 Accessing IGX Devices via SFTP

To access an IGX device via SFTP, follow these steps:

1. Download and install WinSCP from the official website: https://winscp.net/eng/download.php
2. Open WinSCP and click on the "New Site" button.
3. In the "New Site" window, enter the following information:

```
File protocol: SFTP
Host name: <ip_address>
Port number: 22
User name: root
Password: root
```

Replace `<ip_address>` with the IP address of the IGX device you wish to connect to.

4. Click the "Save" button and then "Login" to connect to the IGX device via SFTP.

You should now be able to browse and transfer files to/from the IGX device via WinSCP.

3.5.4 Basic QNX Console Commands

Command	Description	Example
`ls`	Lists the contents of the current directory.	`ls`
`cd <directory>`	Changes the current directory.	`cd /root/igx`
`pwd`	Prints the current working directory.	`pwd`
`ifconfig`	Displays network interface configuration information.	`ifconfig`
`ping <ip>`	Tests network connectivity.	`ping 192.168.0.1`
`netstat`	Displays network statistics and active connections.	`netstat`
`kill <pid>`	Terminates the process with the specified process ID.	`kill 1234`
`slay <process>`	Like `kill` except you can use process names.	`slay igx`
`ps`	Displays information about running processes.	`ps`
`tar`	Compresses and decompresses files and directories.	`tar -cvf archive.tar file`
`top`	Displays the system resource usage (CPU, memory, etc.) in real-time.	`top`

Command	Description	Example
`shutdown`	Immediately restarts the operating system. IGX should automatically start if using the default configuration.	`shutdown`

To learn more about these commands and their options, check out the official QNX documentation at Official QNX Documentation[4].

Starting and Stopping the IGX Application

The IGX application is located at `/root/igx` and can be started and stopped using the following commands:

- To start the IGX application, use the following command:

  ```
  /root/igx
  ```

  This will automatically kill any other instance of IGX that may be running.
- To stop the IGX application, use the following command:

  ```
  slay igx
  ```

  This will gracefully stop the IGX application and release all resources.

Note that only one instance of the IGX application can run at a time, so it is important to stop the currently running instance before starting a new one.

4 https://www.qnx.com/developers/docs/7.1/#com.qnx.doc.neutrino.user_guide/topic/cmdline.html

4 IGX User-Managed System Care

This section of the manual is designed to assist users who wish to service their IGX devices independently. While user-managed system care can be a rewarding and cost-effective way to maintain your device, it is essential to follow the guidelines provided in this manual to ensure the safety and longevity of your IGX device. Should you require assistance with any of these advanced service mode activities, Pyramid's expert technicians are always available to help.

Before attempting any user-managed system care tasks, make sure to review the safety precautions outlined in the manual, and ensure that you have the necessary tools, equipment, and knowledge to perform the tasks safely and effectively. Additionally, it is essential to work in a clean, well-lit, and organized environment to avoid potential damage to your device.

Here are some general guidelines to follow when performing user-managed system care on your IGX device:

1. **Regular Inspection**: Regularly inspect your IGX device for any signs of wear, damage, or loose connections. This can help you identify potential issues before they become more significant problems. Be sure to consult the device's user manual for specific inspection intervals and procedures.
2. **Firmware Updates**: Keep your IGX device up-to-date with the latest firmware releases from Pyramid. Regular updates can enhance the performance, stability, and security of your device. Refer to the manual for instructions on how to check and install firmware updates. Some medically controlled systems will be restricted to certain versions of IGX that have more formal testing. Please consult your Pyramid representativity in this case.
3. **Cleaning**: Keep your IGX device clean and free of dust, debris, and other contaminants. Regular cleaning can improve the device's performance, prevent overheating, and reduce the risk of damage. Be sure to follow the recommended cleaning procedures outlined in the user manual.
4. **Component Replacement**: Over time, some components in your IGX device may need to be replaced due to wear or damage. Always use genuine Pyramid replacement parts to ensure optimal performance and compatibility. Consult the user manual for guidance on identifying and replacing worn or damaged components.
5. **Troubleshooting**: If you encounter any issues with your IGX device, consult the troubleshooting section of the user manual for possible solutions. If the issue persists, don't hesitate to contact Pyramid's support team for assistance.

Remember, while user-managed system care can be an effective way to maintain your IGX device, Pyramid's team of skilled technicians is always available to provide support and assistance when needed. If you are unsure about any aspect of servicing your device, it is best to consult with Pyramid's experts to ensure the safety and longevity of your investment.

4.1 IGX Firmware Update Guide

4.1.1 Introduction

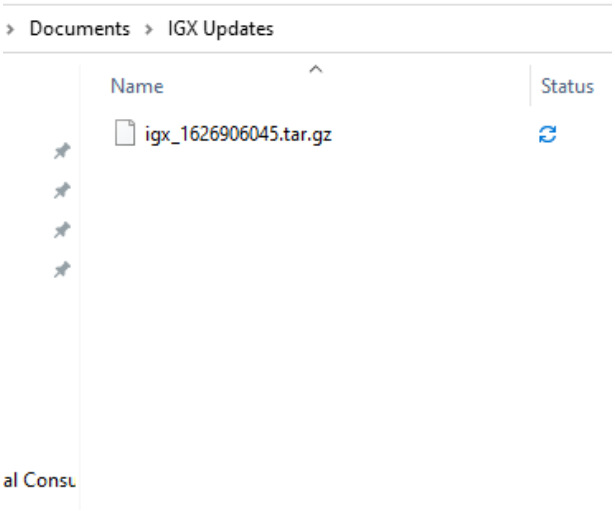
Firmware updates are typically carried out by trained professionals in a Pyramid-controlled facility. However, there may be circumstances where our clients need to perform firmware updates themselves. It is essential to note that improper updating may result in permanent data loss or damage to the device's functionality. Pyramid does not provide any warranty for data or time lost due to improper firmware installations. With this in mind, this guide will outline the steps necessary to successfully update your device's firmware.

Identify the Firmware File

An IGX firmware file is usually stored in a `.tar.gz` format, with the name "igx" followed by the version number. This file contains all the binaries and configurations required for

updating the device. The operating system (QNX) and supporting libraries are stored separately in a different file system.

> Do not decompress the file before uploading it to the device, as the device expects a compressed file and will fail to update without one.



9 A typical IGX firmware file

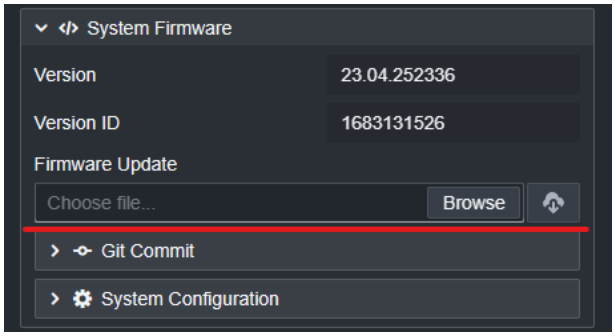
Uploading the Firmware

The process for accessing the update screen may vary between devices and software versions. Typically, there will be an "Admin" link that directs you to the relevant page.

If unsure, you can enter the direct URL into your browser to access the admin page, such as `http://<Device IP Address>/io/admin` (e.g., http://192.168.100.20/io/admin). Most modern browsers do not require the "http://" prefix, so you can omit it.

Upon accessing the admin page, you will see an interface similar to the images shown in the original documentation. The procedure for updating the firmware remains the same, regardless of the interface's appearance.

On the latest versions of IGX, the GUI should look like the following:



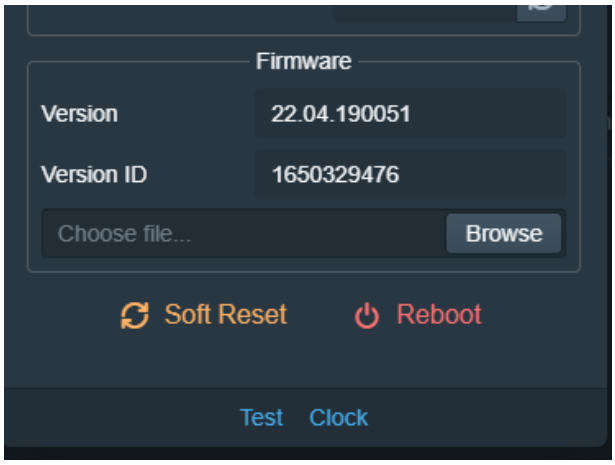
10 IGX firmware upload interface.

Optionally you can click the download button to download a copy of the last uploaded firmware. This is useful for recovering back to an older version.

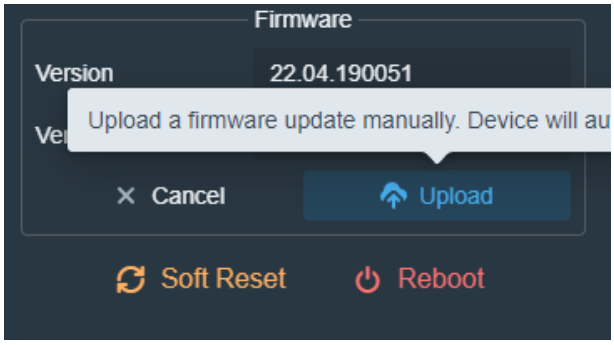
Click on the “Browse” button to select your update file from your local computer.

Then the download button should change into an upload button. Click the upload button to install the selected update.

In older versions of IGX, you may find the admin page looking like this. While the interface itself is different, the procedure of updating will be the same.



11 Before a file is selected

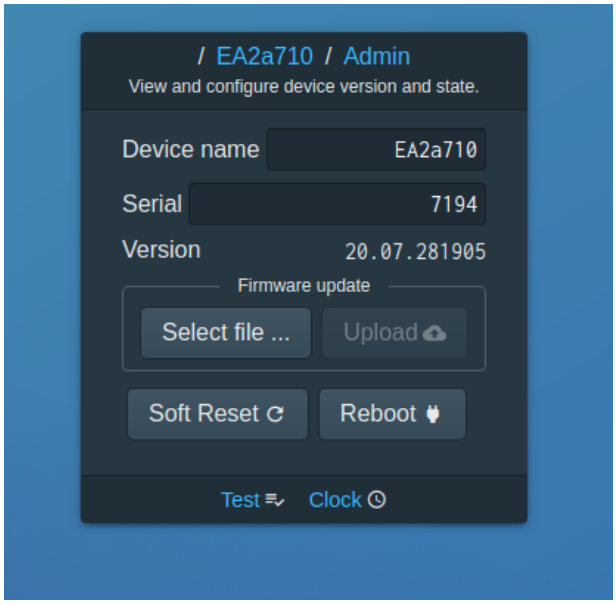


12 After the file is selected

Click on “Browse” and select the update file from your local machine.

After selecting your file, the “Upload” button should appear.

Click the “Upload” button and the update process will begin.



13 Historic file upload interface

Press the “Select file ...” button and select the compressed update file on your system.

Then press the “Upload” button. The update will automatically start and then reset the device.

Post-Firmware Update

The device will automatically restart after the firmware update. You will temporarily lose connection to the device, which is normal.

Do not manually refresh the page during the update or power down the device until it has restarted. If the device has not restarted after 3 minutes or you cannot reconnect, try power cycling the device to resolve any unexpected issues. If the device still malfunctions, contact Pyramid's support team for assistance.

Update Validation and Troubleshooting

Verify that the device is functioning as expected. If necessary, power cycle the device to resolve any issues that may have arisen after the update.

> If you are upgrading from a significantly older firmware version, manually power cycling the device might be required to complete the update.

Contact Pyramid's support team if you require further assistance.

4.2 IGX SD Card Flashing Guide

4.2.1 Introduction

This guide will take you step by step through the standardized procedure of flashing an SD card for the purpose of either updating or creating for the first time a usable card for Pyramid devices.

Our SD cards contain all the firmware and data our devices use.

You might be creating an SD card for the first time for the purpose of manufacturing or trying to fix a card that has been broken beyond simple repair.

4.2.2 Get the Micro SD Card

Any micro SD card that is 32GB should work. We use a microSDHC UHS-I class 10 U1 card made by Samsung, this may change in the future as SD card technology moves quite quickly and manufactures stop making and supporting obsolete cards.

4.2.3 Get the Desired Image File

First you are going to need a file called an ‘image’, the name comes from the fact that it’s a snapshot of the device’s state at one time.

An image file is simply a copy of another SD card contained in a single file.

A common file extension used for images is ‘.iso’ although Pyramid images are usually compressed using gzip so our file extensions are ‘.iso.gz’. **If the image is compress do not decompress the file before flashing.**

Do not try and flash an image located on a network drive. In the past we have found issues with this, just make sure you use a local copy of the file before using it.

4.2.4 Get the Required Tool

There are many programs and tools to flash SD cards, but we can’t make any gaurnties they will work correctly. For example we have found Win32 Disk Imager to sometimes fail and not report the failure correctly. For the sake of constancy and standardization everyone at Pyramid uses the same tool called Etcher.

Download Etcher here[5]

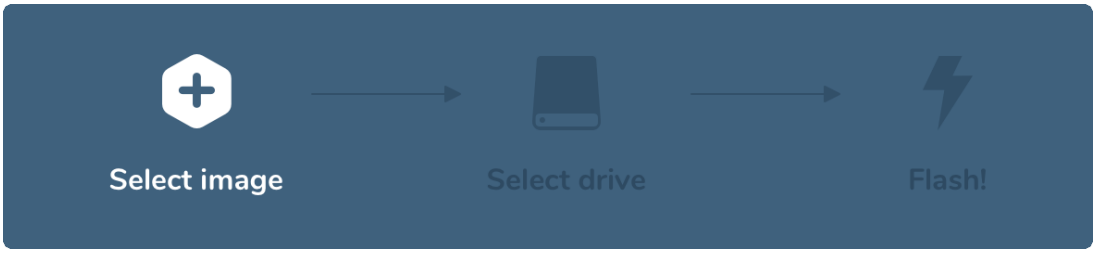
Etcher works on Windows, Linux and Mac, it flashes quickly, handles compressed images without configuration, and validates the card after writing to confirm the flash worked correctly. It also has a nice simple interface to boot!

5 https://www.balena.io/etcher/

4.2.5 Flash the Card

Start up Etcher and then select the image file on your computer's file system.

Insert the card into your computer, you may need a USB or full size SD card adapter depending on the type of computer you have.



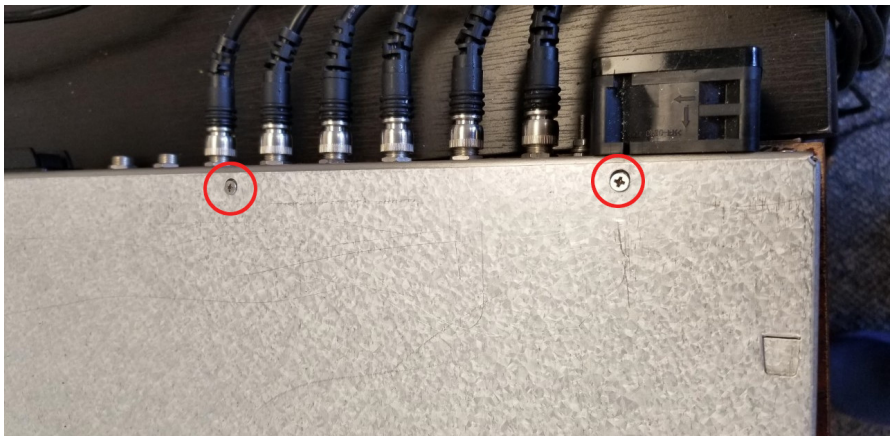
Etcher should immediately recognize the card but you may have to select the correct drive if there are multiple connected to your computer.

Then just press flash, and wait. This should take at least several minutes so feel free to get a coffee.

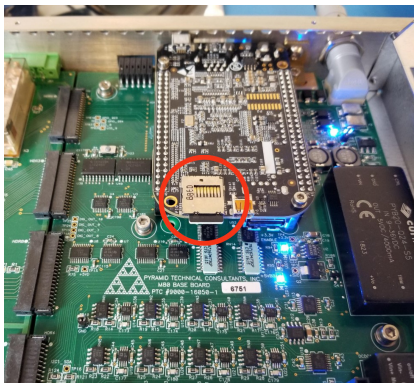
If all went well you should be ready to install the freshly made card. If Etcher gave you an error, just try again, or try a different card.

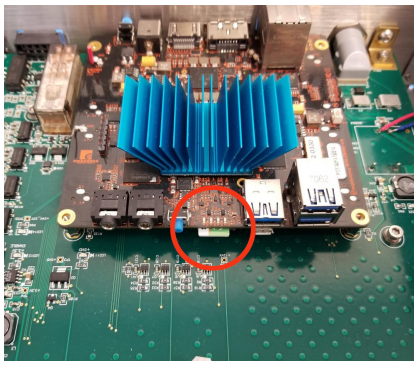
4.2.6 Card Installation or Removal

To install or remove a card from a Pyramid device you must first remove the top panel to gain access to the SD card reader and control board. For our 1U or 2U rack mountable devices, the top panel can be removed by unscrewing the four small Phillips-head screws located on the top of the rear end of the device. Then slide the cover off by pushing back towards the rear end of the device.



Depending on the device the card reader might be located in different locations, but it should look something like in the two example pictures bellow.





To remove the card gently press the card in like a button and it should then pop back out and be easily removed. To install a card do the same but in reverse were you gently press in to lock the card back into place.

4.2.7 Reboot the Device

Once the new card has been installed, you may power on the device normally. The LEDs on the Ethernet port (if there is Ethernet on this device) should be on and flickering with an Ethernet cable attached.

4.3 IGX SD Card Imaging Guide

4.3.1 Summary

This document provides instructions for creating and compressing SD card disk images on Linux and Windows 10 systems.

For Linux, the procedure uses the `dd` and `gzip` command-line tools to create and compress the disk image. Users must identify the SD card device using the `lsblk` command and then execute the `dd` command to create the disk image with the `.iso.gz` extension.

For Windows 10, the process requires the use of Win32 Disk Imager to create the disk image and 7-Zip to compress it. Users must install and run Win32 Disk Imager, select the appropriate SD card device, and create the disk image with the `.iso` extension. Afterward, 7-Zip is used to compress the disk image into a `.iso.gz` file.

Both procedures require sufficient free space on the computer to store the disk image and its compressed version.

4.3.2 Linux Procedure

Creating a disk image of an SD card on Linux involves the use of two command-line tools: `dd` and `gzip`. These tools are typically pre-installed on most Linux distributions. The `dd` command is used for copying and converting data, while `gzip` is a data compression tool. In this procedure, you will create an image of the SD card and compress it as you go.

1. **Identify the SD card device**: To find the correct device file for your SD card, run the following command:

   ```
   lsblk
   ```

   This command lists all the block devices on your system. Look for the SD card in the list, typically named `mmcblk0` or `mmcblk0p1`. Note that the actual name may vary depending on your system.

2. **Create and compress the disk image**: Run the following command to create a disk image of the SD card and compress it using `gzip`:

   ```
   # Standard Command
   sudo dd if=/dev/mmcblk0 | gzip > filename.iso.gz
   # Command With Progress
   sudo dd status=progress if=/dev/mmcblk0 | gzip > filename.iso.gz
   ```

   Replace `/dev/mmcblk0` with the appropriate device file for your SD card and replace `filename` with your desired name for the disk image. This command may take around 20 minutes to complete, depending on the size of the SD card. Ensure that the destination for the output file has enough free space to accommodate the compressed image.

For more information on the `dd` command, refer to the official GNU Coreutils documentation[6]. For additional details on `gzip`, consult the gzip manual page[7].

4.3.3 Windows 10 Procedure

To create and compress an SD card disk image that results in a `.iso.gz` file on Windows 10, follow these steps:

1. **Download and install Win32 Disk Imager**: Download the latest version of Win32 Disk Imager from the official website[8] and install it on your computer.
2. **Insert the SD card**: Insert the SD card into your computer's SD card slot or an external card reader.
3. **Launch Win32 Disk Imager**: Run Win32 Disk Imager and select the appropriate drive letter for your SD card in the "Device" dropdown menu.
4. **Create the disk image**: Click on the folder icon next to the "Image File" field and choose a destination for the output file. Enter a file name with the `.iso` extension, for example, `filename.iso`. Click the "Read" button to start creating the disk image. This process may take some time, depending on the size of the SD card.
5. **Download and install 7-Zip**: Download and install the latest version of 7-Zip[9] if you don't already have it on your computer.
6. **Compress the disk image**: After the disk image has been created, you can compress it using 7-Zip. Right-click the `.iso` file, choose "7-Zip" from the context menu, and select "Add to archive...". In the "Archive format" dropdown, choose "gzip", and click "OK" to start compressing the image. Once the compression is complete, the file will have the `.iso.gz` extension, for example, `filename.iso.gz`.

Ensure that you have enough free space on your computer to store the disk image and its compressed version.

For more information about Win32 Disk Imager, visit the official documentation[10]. For further details on using 7-Zip, consult the 7-Zip help documentation[11].

4.4 Modifying the IGX System XML File

Use WinSCP to login to your device.

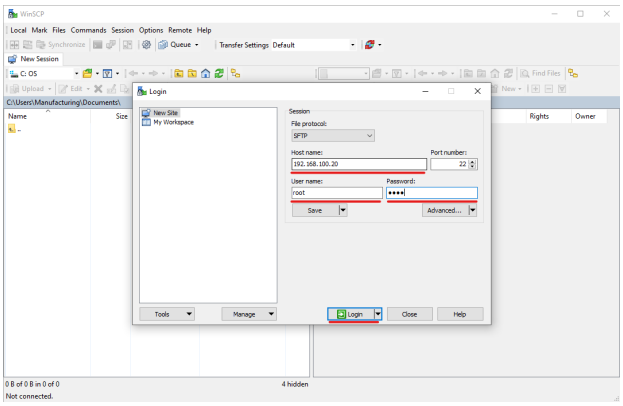
Click on “New Session”.

6 https://www.gnu.org/software/coreutils/manual/html_node/dd-invocation.html
7 https://www.gnu.org/software/gzip/manual/gzip.html
8 https://sourceforge.net/projects/win32diskimager/
9 https://www.7-zip.org/
10 https://sourceforge.net/p/win32diskimager/wiki/Home/
11 https://www.7-zip.org/support.html

Type the IP address of the device and use the username of “root” and passowrd “root”.

Then click the Login button.

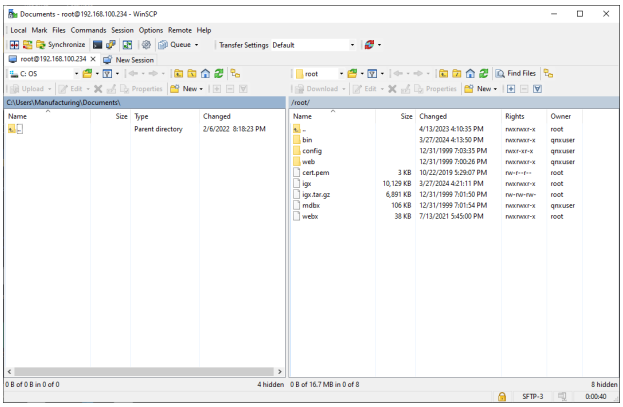
If probpted that the SSH key of the device has changed, you can click “Yes” trust this device.



14 Loging into an IGX device.

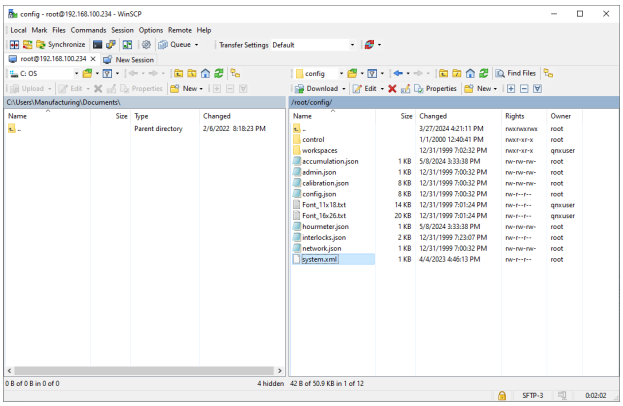
When you login for the first time, you will start in the “/root” folder.

Navigate to “/root/config” by double clicking the config folder in the list to the right.



15 Starting in the /root folder.

In the config folder there will be a variety of files. Look for the “system.xml” file and double click it to start editing.



16 The config folder files.

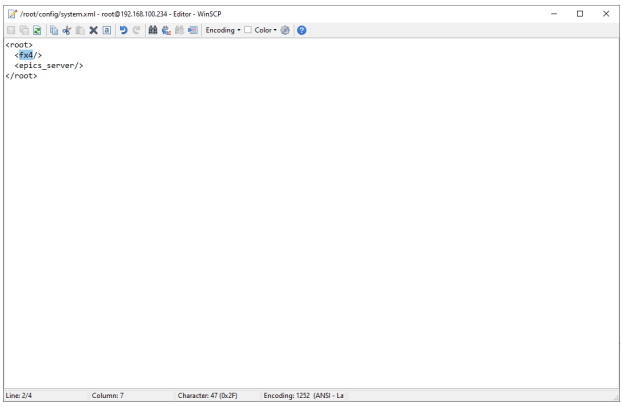
Make the changes you desire to the file.

In this case you can modify the <fx4/> node to the name of a different device. See the table bellow for all the available device nodes.

After modifying the file and saving, power cycle the device in order for the changes to take effect.

If the devices fails to start due to a misconfigured file, you can still use WinSCP

again and revert back the changes you made to recover the device.



17 The system XML file.

4.4.1 Device Nodes

Device	XML Tag Node	Description
FX4-XX	<fx4/>	4 Channel Electrometer
MX1	<mx1/>	8 Channel Data Acquistion System
TX2	<tx2/>	2 Channel Gaussmeter
KX8	<kx8/>	8 Channel Relay Input Chain
IX256-F2	<ix256/>	256 Channel Electrometer

4.5 BeagleBone Black Replacement Guide

4.5.1 Purpose

This document provides step-by-step instructions for replacing the beagle bone found in some Pyramid devices. The beagle bone is a processing unit that Pyramid uses in its designs. If it becomes damaged, it is easy to replace as it is a standard component. This guide ensures that the replacement process is completed safely and correctly.

Replacing a beagle bone in a Pyramid device is a straightforward process, but it is essential to handle the components with caution and care. By following these instructions and taking the necessary precautions, you can successfully replace the beagle bone and ensure the continued operation of your device.

If you require further assistance or are unsure about any of the steps, it is recommended to contact Pyramid support or refer to their documentation. Contact us at support@ptcusa.com[12].

Procuring a BeagleBone Black

To procure a BeagleBone Black, users have several options. They can either purchase it directly from Pyramid or from BeagleBoard.org[13]. BeagleBone Blacks that are purchased from BeagleBoard.org[14] will not come with any SD card. If you are simply going to swap the SD card from your original unit, then this will not be a problem, but if you want a complete replacement, then we recommend buying from Pyramid.

12 mailto:support@ptcusa.com
13 http://BeagleBoard.org
14 http://BeagleBoard.org

Purchase from Pyramid

1. Email sales@ptcusa.com[15], and request a quote for a BeagleBone Black, pre-loaded with an SD card containing the software of your choice.
2. We will reach out to you to verify the software choice and complete the sales order.

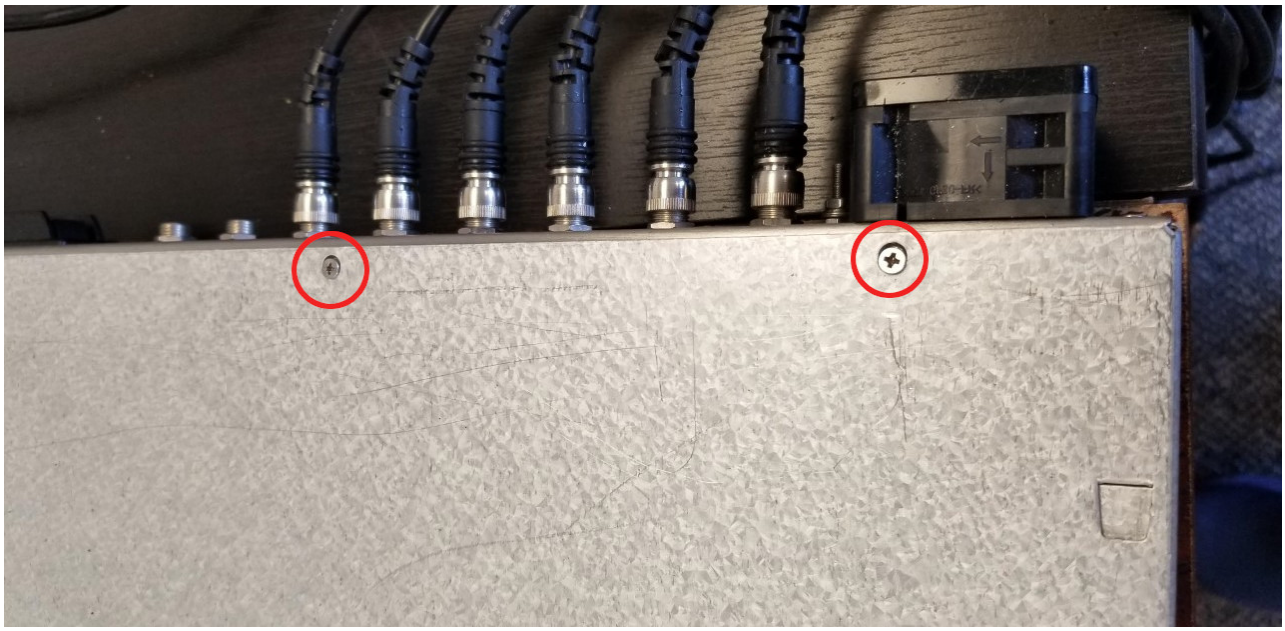
Purchase from BeagleBoard.org

1. Visit the official website: https://beagleboard.org/black
   On the website, you'll find detailed information about the BeagleBone Black, its features, and specifications.
2. Click the "Buy Now" button to see a list of authorized distributors.
3. Choose your preferred distributor from the list and follow their purchase process.

Replacement Procedure

⚡ It is important to handle electronic equipment with caution. Be sure to use appropriate Electrostatic Discharge (ESD) protection, such as an ESD wrist strap, to avoid damaging the components.

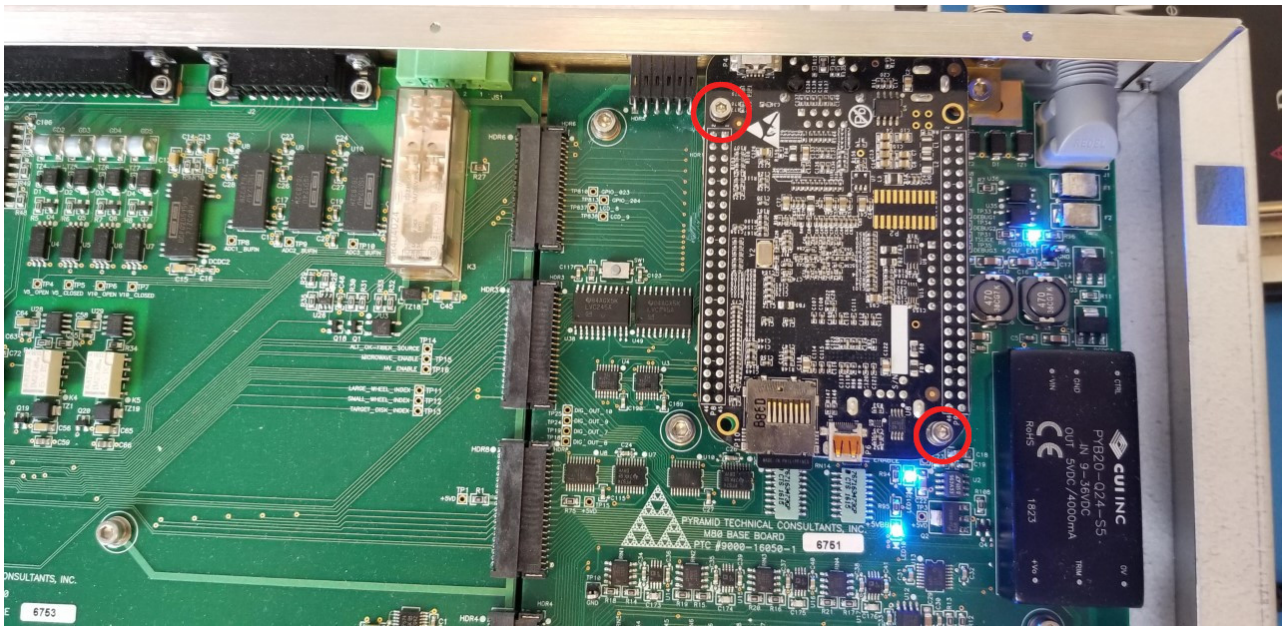
For our 1U or 2U rack mountable devices, the top panel can be removed by unscrewing the four small Phillips-head screws located on the top of the rear end of the device. Then slide the cover off by pushing back towards the rear end of the device.



18 Top Panel Screw Locations

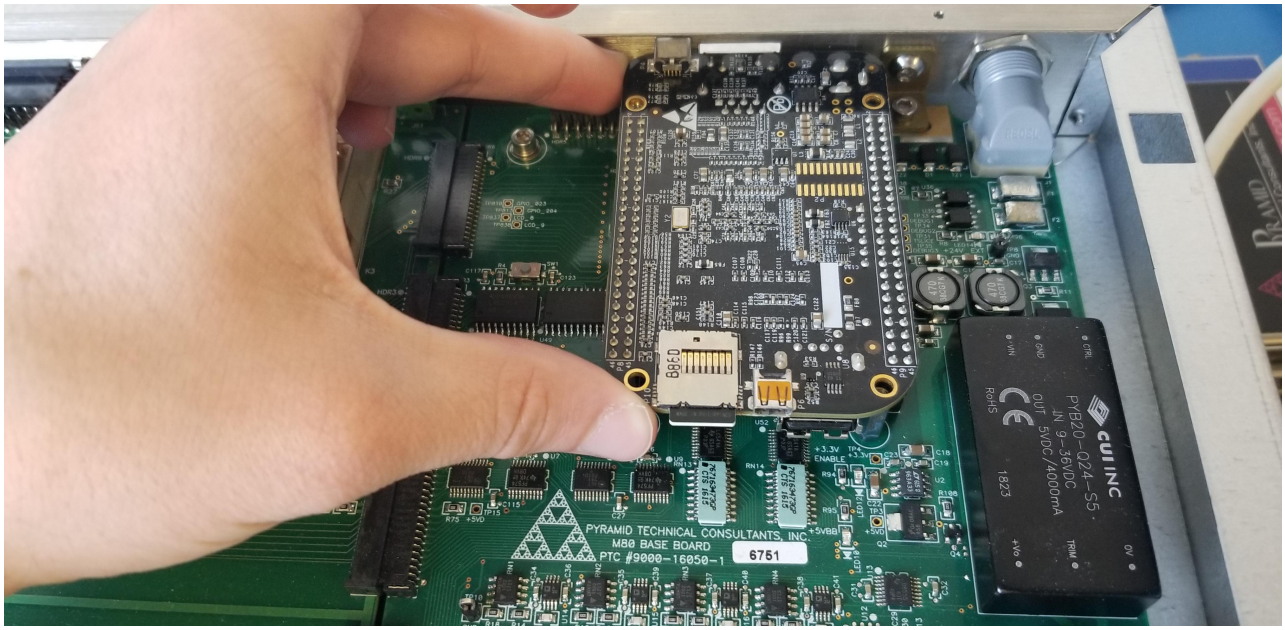
Before the beagle bone can be removed, there will be two or four M3 hex screws that have to be unscrewed. These screws hold the beagle bone in place.

15 mailto:sales@ptcusa.com



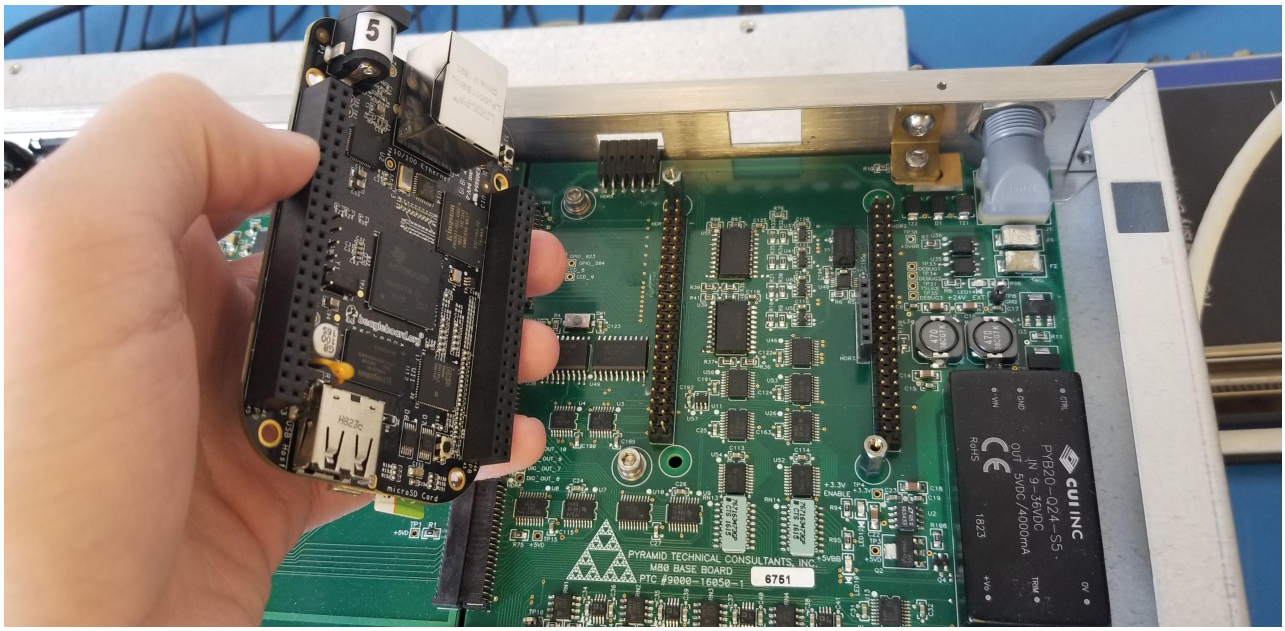
19 Mounting Screw Locations

After unscrewing the mounting screws, grab the board by the corners, and pull it directly up. The fit may be tight, so you might have to wiggle the board slightly to remove it.



20 Where to Grab the Device

Once the old board has been removed, make sure to inspect the pins on the main board. They should all be straight, and nothing should be missing. If there is any damage to the pins or sockets, they should be fixed before installing the new beagle bone.



21 Inspecting the Connector Pins

Take the new beagle bone and put it into position. Press down directly on the corners, making sure that each pin is seated completely into the corresponding socket.

Screw back in the two or four M3 screws to hold the beagle bone in place. Do not over-tighten these screws as it could cause damage to the board.

Slide the top panel back on and re-screw the four small screws into place. Be careful not to over-tighten these screws, as they can strip very easily.

4.6 Running a Local NTP Time Server

4.6.1 Overview

Sometimes when you are running a control system on a closed network, for example in industrial or medical systems, it can be challenging to propagate accurate times across the network without any access to the internet. One solution to this problem is using a Windows computer as a gateway between the internet and the closed control system. The Windows computer gets an accurate time from the internet and passes it along to all the devices downstream in the closed network.

Setting up an NTP (Network Time Protocol) server on Windows 10 or 11 involves configuring the Windows Time service. This is a built-in, but not enabled by default, service that can be configured to run in the background.

Important Notes Before Starting

- Make sure your Windows machine's time is synchronized with a reliable time source to maintain accurate time on your NTP server. This means the Windows machine must have an internet connection or some other source of accurate time information, for example a GPS transceiver.
- Ensure that your network and security policies allow NTP traffic. NTP traffic going to a Windows computer is not typical, your IT department may have a policy against this.

4.6.2 Steps to Configure Windows as an NTP Server

By following these steps, you should be able to configure your Windows 10 or 11 machine to act as an NTP server that other devices can use for time synchronization.

1. **Open Command Prompt as Administrator**
    - Press `Win + X` and select "Command Prompt (Admin)" or "Windows PowerShell (Admin)".
2. **Stop the Windows Time Service**

- Execute the following command to stop the Windows Time service:

```
net stop w32time
```

3. **Configure the Windows Time Service to Use an Internal Time Source**
   - Execute the following command to configure the Windows Time service:

```
w32tm /config /manualpeerlist:"0.pool.ntp.org 1.pool.ntp.org 2.pool.ntp.org
3.pool.ntp.org" /syncfromflags:manual /reliable:YES /update
```

4. **Set the Announce Flags to 5**
   - Execute the following command to configure the Announce Flags registry entry to 5:

```
w32tm /config /reliable:YES /update
```

5. **Enable NTP Server**
   - Execute the following command to set the server as an NTP server:

```
reg add HKLM\\SYSTEM\\CurrentControlSet\\Services\\W32Time\\TimeProviders\
\NtpServer /v Enabled /t REG_DWORD /d 1 /f
```

6. **Start the Windows Time Service**
   - Execute the following command to start the Windows Time service:

```
net start w32time
```

7. **Configure the Windows Firewall to Allow NTP Traffic**
   - Execute the following command to open the NTP port (UDP 123) in the firewall:

```
netsh advfirewall firewall add rule name="NTP" dir=in action=allow
protocol=UDP localport=123
```

Verify Configuration

To verify that your configuration is correct, you can use the following command:

```
w32tm /query /status
```

This command provides status information about the Windows Time service.

Additional Configuration for NTP Clients

Ensure that the client devices are configured to use your Windows NTP server as their time source. This usually involves specifying the IP address of your Windows NTP server in the NTP settings of each client device. If your NTP client is an IGX device, these settings can be found under the “Admin” settings in the system clock configuration section.