

Observatory Sciences Ltd



LabVIEW Interface for the Delta Tau PMAC Motion Controller

Developers Manual

Issue 1.1

June 12, 2014



Document Control

Authors

Name	Company	Signature
Alan Greer	Observatory Sciences Ltd	

Document Approval

Name	Role	Company
Alan Greer	Author	Observatory Sciences Ltd
Philip Taylor	Reviewer	Observatory Sciences Ltd

Revision History

Author	Issue	Date	Description
Alan Greer	V1-0	18-July-2013	Initial release.
Alan Greer	V1-1	12-June-2014	Updated to reflect latest installation instructions.

Table of Contents

1	Introduction	4
1.1	Purpose	4
1.2	Intended Audience	4
1.3	Document Overview	4
1.4	References	4
2	Installing the LabVIEW PMAC Toolkit	5
2.1	Software Requirements	5
2.2	Hardware Requirements	5
2.3	Windows Installation	5
2.4	Licensing	8
3	Using the LabVIEW PMAC Toolkit	10
3.1	Understanding the Basics	10
3.2	Higher Level Control	10
3.3	VI List and Descriptions (API)	10
3.3.1	OsIPmac_Open.vi	11
3.3.2	OsIPmac_Close.vi	11
3.3.3	OsIPmac_Status.vi	12
3.3.4	OsIPmac_Command.vi	12
3.3.5	OsIPmac_AsciiCommand.vi	14
3.3.6	OsIPmac_SetEnabled.vi	14
3.3.7	OsIPmac_GetEnabled.vi	15
3.3.8	OsIPmac_SetVelocity.vi	15
3.3.9	OsIPmac_GetVelocity.vi	16
3.3.10	OsIPmac_SetAcceleration.vi	17
3.3.11	OsIPmac_GetAcceleration.vi	17
3.3.12	OsIPmac_MoveAbsolute.vi	18
3.3.13	OsIPmac_MoveRelative.vi	18
3.3.14	OsIPmac_Home.vi	19
3.3.15	OsIPmac_GetPosition.vi	20
3.3.16	OsIPmac_GetError.vi	20
3.3.17	OsIPmac_AxisPVE.vi	21
3.3.18	OsIPmac_GetStatus.vi	21
4	Example Application - Step By Step Guide	23
4.1	Running the Application	23
4.2	Application Block Diagrams	24

1 Introduction

This document describes how to use the NI LabVIEW software for the control of a Delta Tau Turbo PMAC motion controller over Ethernet. The software has been designed and produced by Observatory Sciences Ltd in collaboration with Delta Tau UK Ltd.

1.1 Purpose

This document provides a detailed guide of how to install and use the PMAC LabVIEW toolkit to communicate with and control a Delta Tau PMAC motion controller. A reasonable level of understanding of LabVIEW is assumed.

A thorough description of every VI is presented along with some simple example applications that can be used "out of the box", by simply loading the VI into the LabVIEW development environment. Detailed descriptions of how to setup a PMAC motion controller is outside the scope of this document, but suitable documentation is available, see reference [1].

1.2 Intended Audience

This document is intended for developers of motion control software for the Delta Tau PMAC motion controller. Knowledge of the PMAC motion controller is assumed, as is a degree of proficiency with LabVIEW.

1.3 Document Overview

Section 2 provides details of how to install the LabVIEW PMAC Toolkit and information regarding the licensing and the free 30 day trial. Section 3 discusses how to use the toolkit, providing detailed explanations of each of the VIs available for use. Section 4 describes the example application provided with the toolkit, providing details on how to run the example VI and how the example VI works.

1.4 References

[1] Turbo PMAC User Manual 3Ax-602264-TUxx, September 12, 2008

[2] Turbo PMAC/PMAC2 Software Reference Manual 3Sx-01.937-xSxx, December 19, 2012

2 Installing the LabVIEW PMAC Toolkit

Installation of the LabVIEW PMAC toolkit is straightforward as it comprises only of native LabVIEW VIs. These VIs can simply be added to an existing project. The VIs are provided with a free 30 day trial. A license key can be purchased which removes the thirty day restriction; this is achieved by simply adding the key to the relevant VI as explained below.

2.1 Software Requirements

- LabVIEW development version 2009 or later.
- Windows XP, Windows VISTA, Windows 7.
- LabVIEW VI Package Manager required for installation.
- No PMAC drivers are required, no additional software is needed to use these VIs. The VI driver set communicates with the PMAC using a native protocol over Ethernet.

2.2 Hardware Requirements

- Standard PC with network connection.
- Delta Tau Turbo PMAC motion controller with Ethernet interface.

2.3 Windows Installation

Download the latest VIPM package available from the Observatory Sciences Ltd website (<http://www.observatorysciences.co.uk/>). The package can be found under the Downloads->Delta Tau Drivers or Downloads->LabVIEW Software pages.

Open the package with VIPM and you will be presented with the installation screen.

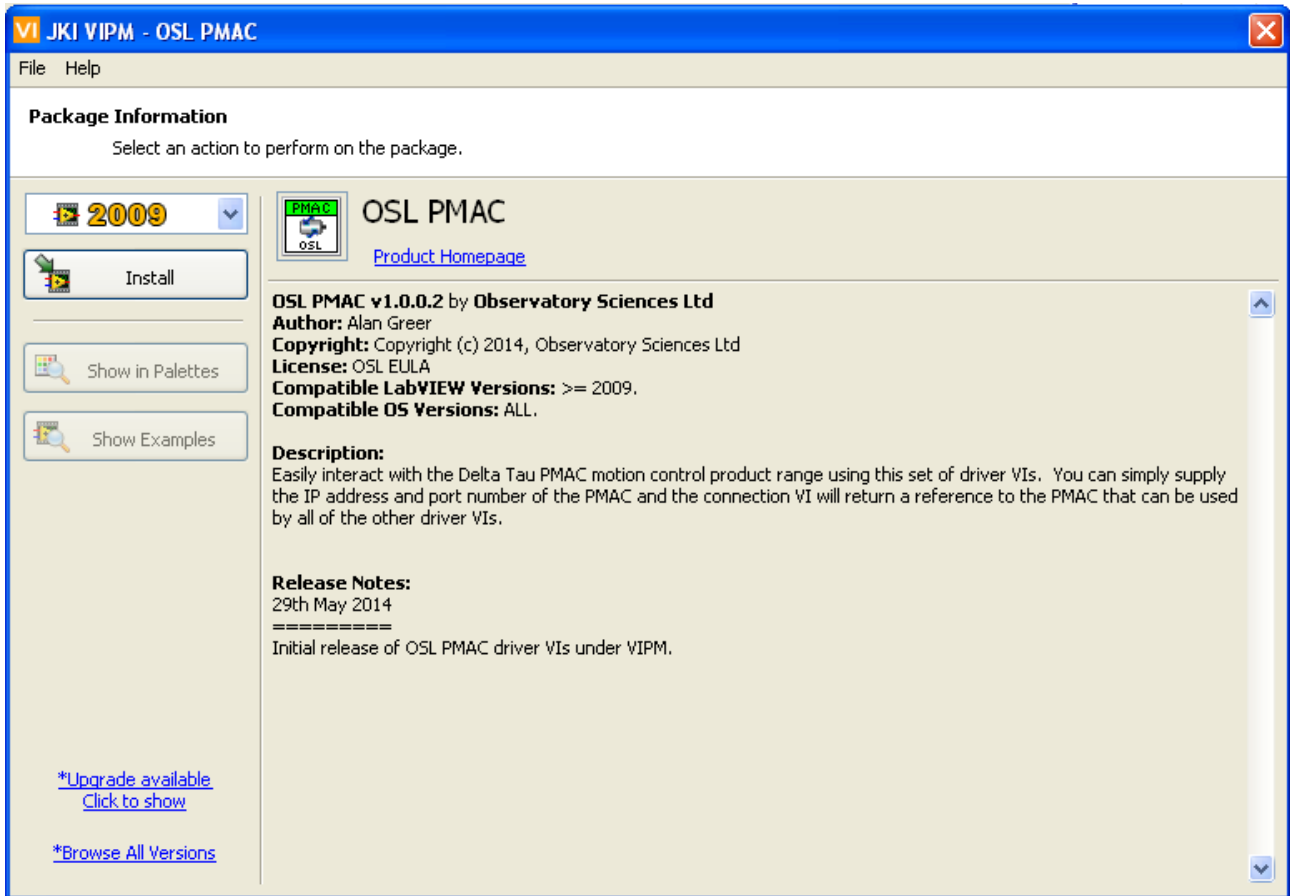


Figure 1 Installer welcome screen.

The version of the software is listed on this screen. Select the version of LabVIEW that you would like to install the package into and click on the "Install" button to begin the installation. You must accept the End User License Agreement to continue with the installation, see figure 2 below.

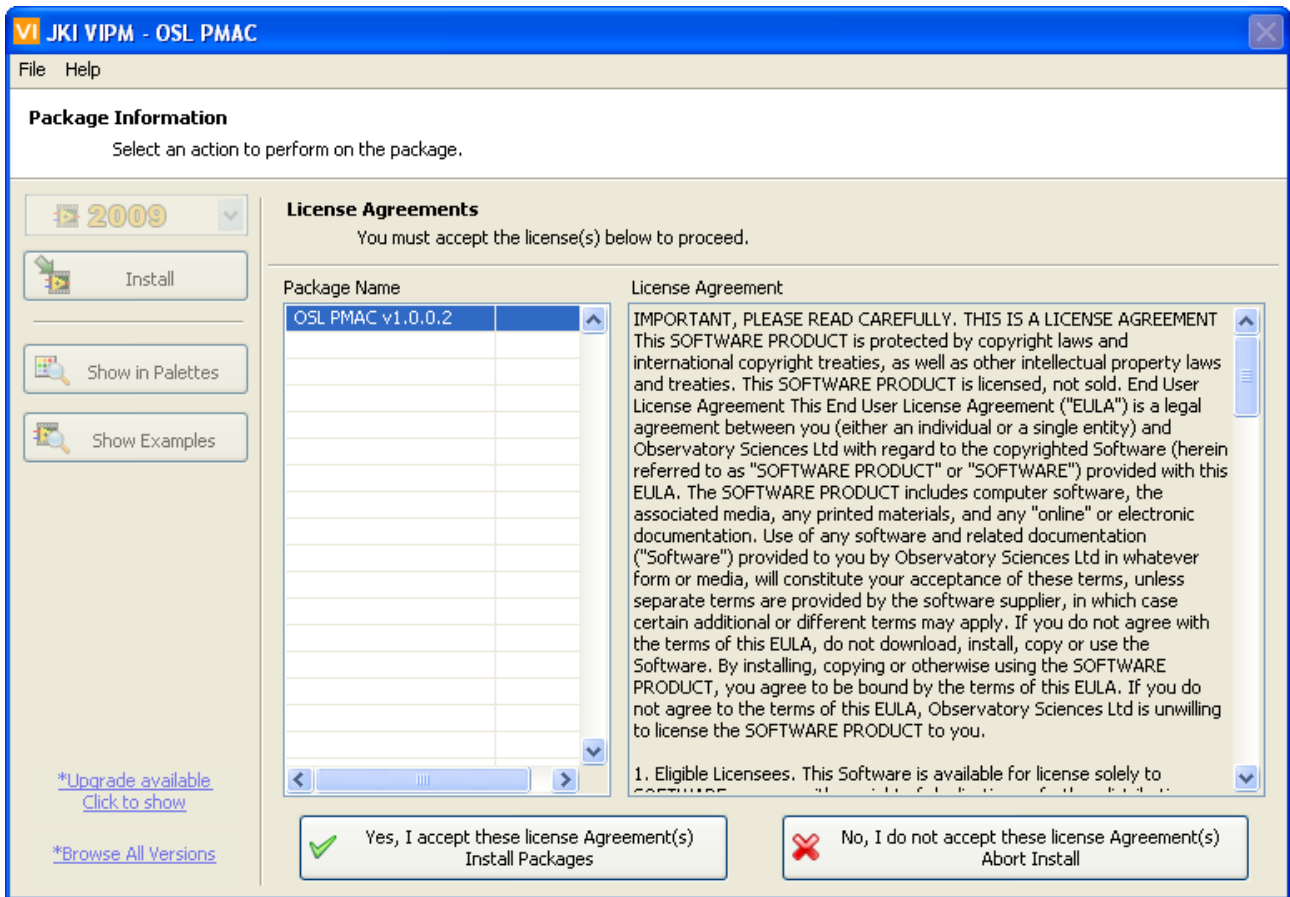


Figure 2 Installer EULA screen.

Once the license agreement has been accepted the installation will begin. During the installation process the LabVIEW application is started if necessary.

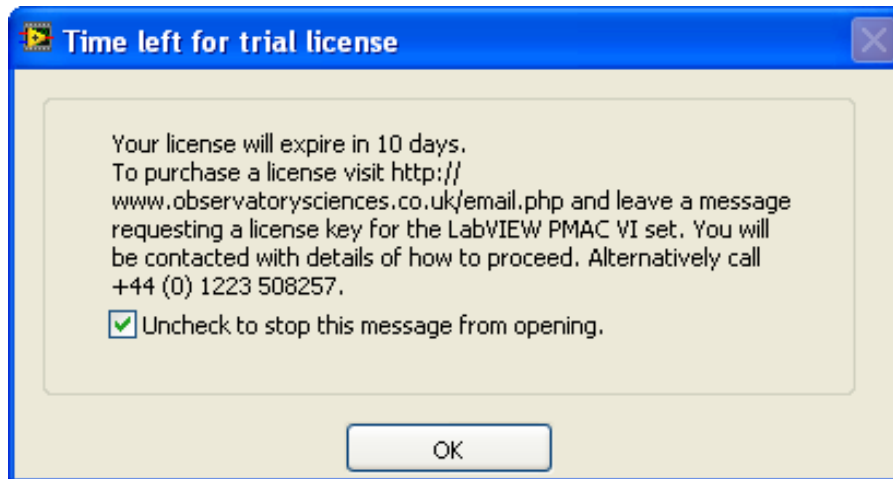


Figure 5 Trial time remaining notification.

The notification can be turned off by un-checking the box in the dialog. It will return if the VI is reloaded.

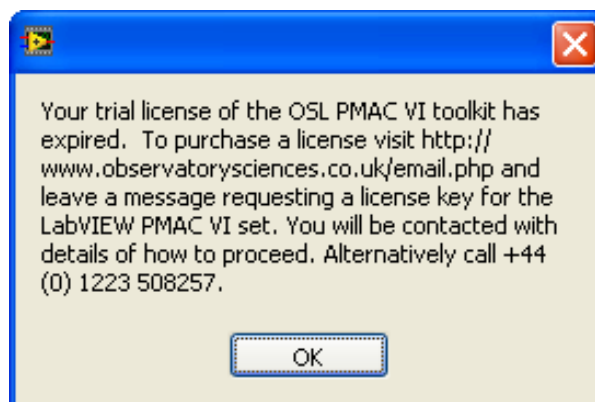


Figure 6 Trial period expired.

For the trial period all of the VIs are fully functional, providing the opportunity to develop a complete application during this time. Once satisfied with the VI set the license can be activated by simply setting a single string value to the key provided, resulting in no disruption to the developed application. License keys have no time limit.

To purchase a license visit <http://www.observatorysciences.co.uk/email.php> and leave a message requesting a license key for the LabVIEW PMAC VI set. You will be contacted with details of how to proceed. Alternatively call +44 (0) 1223 508257.

3 Using the LabVIEW PMAC Toolkit

3.1 Understanding the Basics

The LabVIEW PMAC Toolkit uses a TCP socket communications protocol to communicate with the PMAC over Ethernet. Three levels of VI are provided.

- 1) At the lowest level TCP sockets are opened, LabVIEW queues are created and references to hold data structures are stored. Control of the reference provides protection against multi-threaded applications corrupting the communications with the PMAC controller; you can place as many VIs in as many loops as you like and the communications order is preserved so no data is corrupted. The lower level VIs handle the locking and unlocking of the reference and so these VIs are never used directly by a programmer. They are not listed in section 3.3 for this reason.
- 2) The next level of VIs provide an interface to the basic communication protocol. The VIs allow any type of PMAC data packet to be sent to the PMAC and the raw data (bytes) is returned. This level of VI is provided for advanced users so that all available interactions can be achieved using the LabVIEW PMAC toolkit, but for most users this will never be required. See section 3.3.4 for more details.
- 3) The main public interface VIs provide a simple to use set of control and status VIs, offering the most commonly used features for individual PMAC axes.

Note developers are free to make use of any of the VIs and controls provided, there is no restriction placed on how they should be utilised.

Most of the public VIs are built on top of the ASCII command interface provided by the PMAC controller. All PMAC variables and motor parameters can be set and read using this interface. A basic ASCII VI is provided (see section 3.3.5) to allow a developer the option of creating many more helper VIs if necessary. The complete list of PMAC variables available can be found in Ref [2].

3.2 Higher Level Control

When using the LabVIEW PMAC Toolkit the general sequence of operations is as follows:

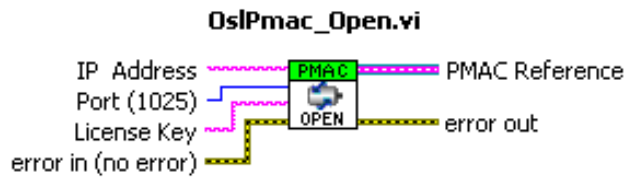
- 1) Open a connection.
- 2) Check the connection status.
- 3) Read status, write commands.
- 4) Repeat 2 and 3.
- 5) Close the connection once complete.

Multiple connections to different PMAC controllers can be opened but only one connection to an individual PMAC controller should ever be opened. The reference returned by the open VI must be used in all further VI calls. Data corruption can occur if more than one connection is opened to the PMAC.

3.3 VI List and Descriptions (API)

The following sections list each VI provided along with a description of the inputs, outputs and purpose of the VI.

3.3.1 OslPmac_Open.vi



Inputs:

Name	Type	Description
IP Address	String	The IP address of the Delta Tau PMAC controller to attempt a connection with.
Port	Unsigned 16	The port of the Delta Tau PMAC controller to attempt a connection with. Defaults to 1025.
License Key	String	License key obtained from Observatory Sciences Ltd to enable unlimited use of the VIs. If this is left blank then a 30 day trial period is provided.
Error in	Error	Standard LabVIEW error input.

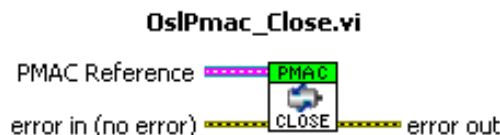
Outputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Error out	Error	Standard LabVIEW error output.

Description:

Attempts to open a connection to a Delta Tau PMAC controller on the specified IP address and port. The reference to the PMAC returned by this VI should be passed to all other VI calls.

3.3.2 OslPmac_Close.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Error in	Error	Standard LabVIEW error input.

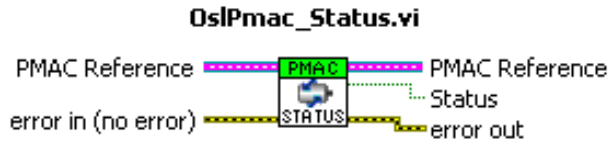
Outputs:

Name	Type	Description
Error out	Error	Standard LabVIEW error output.

Description:

Closes a connection to the referenced Delta Tau PMAC controller. The reference is invalidated by this VI and as such is not passed as an output.

3.3.3 OsIPmac_Status.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Error in	Error	Standard LabVIEW error input.

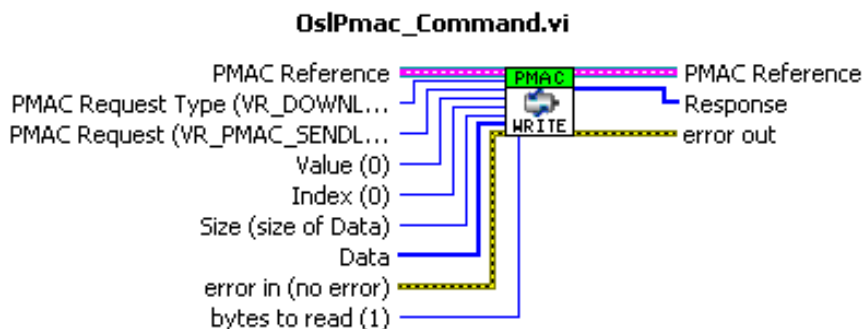
Outputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Status	Boolean	The current status of the connection to the PMAC controller. If the connection is lost or the application failed to make a connection then this value will be false. If the last VI communicating with the PMAC completed successfully then this value will be true.
Error out	Error	Standard LabVIEW error output.

Description:

This VI returns the current status of the connection to the PMAC controller. A status of true means the application is currently connected and calls can be made. A status of false means the connection has been lost or was never made.

3.3.4 OsIPmac_Command.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
PMAC Request Type	Enumeration	VR_UPLOAD VR_DOWNLOAD. PMAC specific request type. For full details see the PMAC user manual (Ref [2]).
PMAC Request	Enumeration	VR_PMAC_SENDBLINE VR_PMAC_GETLINE VR_PMAC_FLUSH VR_PMAC_GETMEM VR_PMAC_SETMEM VR_PMAC_SETBIT VR_PMAC_SETBITS VR_PMAC_PORT VR_PMAC_GETRESPONSE VR_PMAC_READREADY VR_PMAC_RESPONSE VR_PMAC_GETBUFFER VR_PMAC_WRITEBUFFER VR_PMAC_WRITEERROR VR_FWDOWNLOAD VR_IPADDRESS. PMAC specific request value. For full details see the PMAC user manual (Ref [2]).
Value	Unsigned 16	This value is request specific. For full details see the PMAC user manual (Ref [2]).
Index	Unsigned 16	This value is request specific. For full details see the PMAC user manual (Ref [2]).
Size	Unsigned 16	This value should be set to the size of the array of data passed in below. If it is set to zero then the size is automatically calculated from the data array.
Data	Array of Unsigned 8	The actual data to send to the PMAC controller.
bytes to read	Integer 32	The number of bytes to read in response to the supplied data. Defaults to 1 byte (the acknowledgement byte).
Error in	Error	Standard LabVIEW error input.

Outputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Response	Array of Unsigned 8	The response data returned by the PMAC controller.
Error out	Error	Standard LabVIEW error output.

Description:

This is the standard communications VI that is used by many other calls within the VI set. It is not expected to be used explicitly by a developer but if specialist control of the PMAC is required then it can be achieved by implementing calls to this VI. It performs a request (write/read) operation on the PMAC controller, returning the raw data as an array of unsigned 8 bit integer values.

3.3.5 OsIPmac_AsciiCommand.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Command	String	The ASCII command to send to the PMAC controller.
Error in	Error	Standard LabVIEW error input.

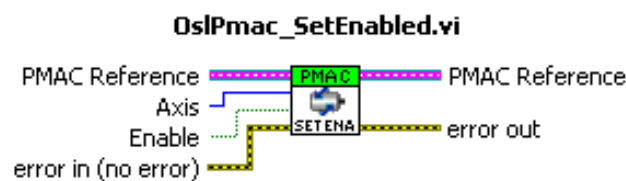
Outputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Response	String	The ASCII response received from the PMAC controller.
Error out	Error	Standard LabVIEW error output.

Description:

This VI provides an ASCII command interface to the PMAC controller. All on-line commands defined in the software reference manual (Ref [2]) can be sent and any responses read using this VI.

3.3.6 OsIPmac_SetEnabled.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Axis	Integer	The axis number to set.
Enable	Boolean	Enable (true) or disable (false) the axis. This sets Ixx00.
Error in	Error	Standard LabVIEW error input.

Outputs:

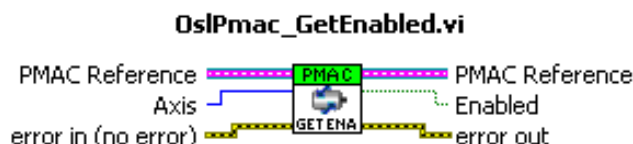
Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains

		information relating to the Ethernet connection.
Error out	Error	Standard LabVIEW error output.

Description:

This VI enables or disables a specific axis by setting the PMAC I variable Ixx00 to either 1 or 0.

3.3.7 OsIPmac_GetEnabled.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Axis	Integer	The axis number to check.
Error in	Error	Standard LabVIEW error input.

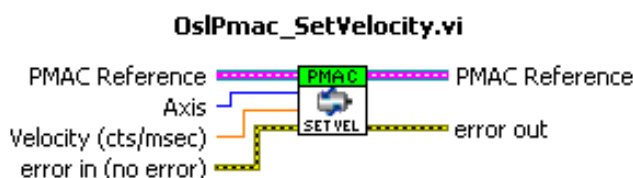
Outputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Enabled	Boolean	Enabled state (Ixx00) of the axis.
Error out	Error	Standard LabVIEW error output.

Description:

This VI returns the current enabled state of the specified axis. It reads the PMAC I variable Ixx00.

3.3.8 OsIPmac_SetVelocity.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Axis	Integer	The axis number to set.

Velocity	Double	Set the velocity of the axis in units of counts/millisecond.
Error in	Error	Standard LabVIEW error input.

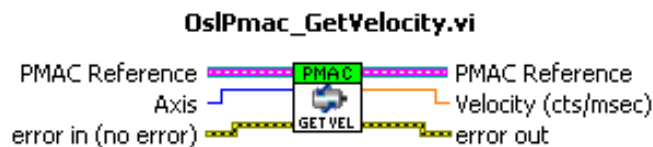
Outputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Error out	Error	Standard LabVIEW error output.

Description:

This VI sets the jog velocity of an axis. The velocity is specified in units of counts/millisecond. This sets the value of the PMAC variable Ixx22.

3.3.9 OslPmac_GetVelocity.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Axis	Integer	The axis number to check.
Error in	Error	Standard LabVIEW error input.

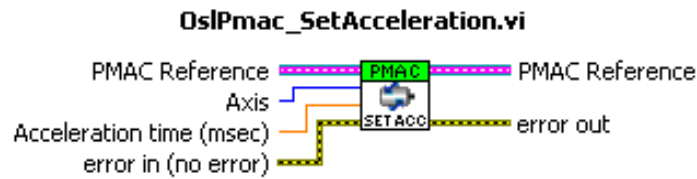
Outputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Velocity	Double	Currently set velocity in counts/millisecond.
Error out	Error	Standard LabVIEW error output.

Description:

This VI returns the current jog velocity of the specified axis in units of counts/millisecond. It reads the PMAC I variable Ixx22.

3.3.10 OslPmac_SetAcceleration.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Axis	Integer	The axis number to set.
Acceleration time	Double	Set the acceleration time of the axis in units of milliseconds. This is the time it takes for the axis to reach its velocity.
Error in	Error	Standard LabVIEW error input.

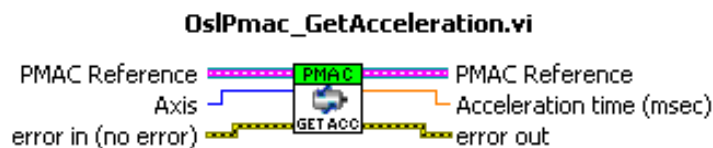
Outputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Error out	Error	Standard LabVIEW error output.

Description:

This VI sets the acceleration time of an axis. The acceleration time is specified in units of milliseconds. This sets the value of the PMAC variable Ixx20.

3.3.11 OslPmac_GetAcceleration.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Axis	Integer	The axis number to check.
Error in	Error	Standard LabVIEW error input.

Outputs:

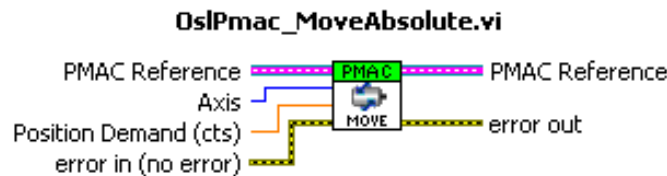
Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.

Acceleration time	Double	Currently set acceleration time in milliseconds.
Error out	Error	Standard LabVIEW error output.

Description:

This VI returns the current acceleration of the specified axis in units of milliseconds. It reads the PMAC I variable Ixx20.

3.3.12 OsIPmac_MoveAbsolute.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Axis	Integer	The axis number to move.
Position Demand	Double	The demand position in counts that the axis should move to.
Error in	Error	Standard LabVIEW error input.

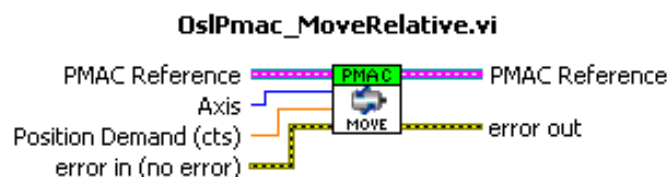
Outputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Error out	Error	Standard LabVIEW error output.

Description:

This VI jogs the axis to the specified demand position. For example if the axis was set to 1 and the position demand set to 10000 then this VI would issue the command "#1J=10000" to the PMAC controller. The demand position is set in counts.

3.3.13 OsIPmac_MoveRelative.vi



Inputs:

Name	Type	Description
------	------	-------------

PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Axis	Integer	The axis number to move.
Position Demand	Double	The demand position relative to the axis current position in counts that the axis should move by.
Error in	Error	Standard LabVIEW error input.

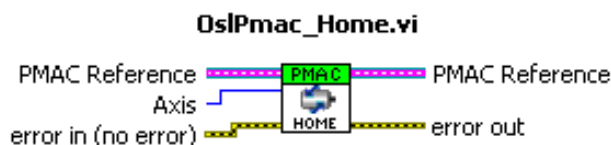
Outputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Error out	Error	Standard LabVIEW error output.

Description:

This VI jogs the axis to the specified demand position relative to its current position. For example if the axis was set to 1 and the position demand set to 10000 then this VI would issue the command "#1J^10000" to the PMAC controller. If the axis was already at position 10000 then after this move the axis would be at position 20000. The relative position demand is set in counts.

3.3.14 OslPmac_Home.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Axis	Integer	The axis number to home.
Error in	Error	Standard LabVIEW error input.

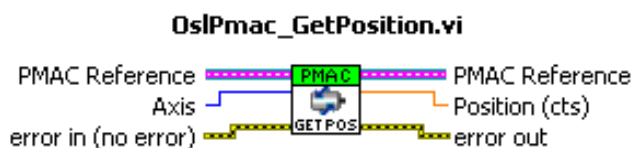
Outputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Error out	Error	Standard LabVIEW error output.

Description:

This VI starts a home move on the specified axis. For example if the axis is set to 1 then this VI would issue the command "#1HM" to the PMAC controller.

3.3.15 OslPmac_GetPosition.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Axis	Integer	The axis number to check.
Error in	Error	Standard LabVIEW error input.

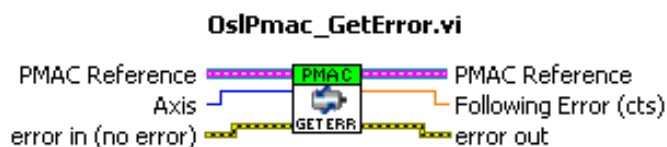
Outputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Position	Double	Current position of the axis in counts.
Error out	Error	Standard LabVIEW error output.

Description:

This VI returns the current position of the specified axis in counts.

3.3.16 OslPmac_GetError.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Axis	Integer	The axis number to check.
Error in	Error	Standard LabVIEW error input.

Outputs:

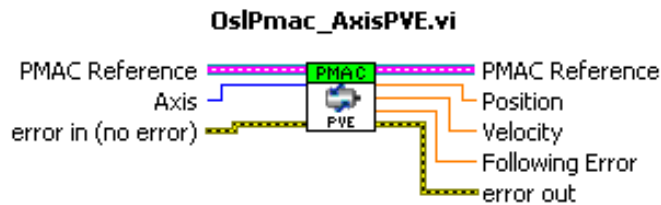
Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Following Error	Double	Current following error of the axis in counts.

Error out	Error	Standard LabVIEW error output.
------------------	-------	--------------------------------

Description:

This VI returns the current following error of the specified axis in counts.

3.3.17 OslPmac_AxisPVE.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Axis	Integer	The axis number to check.
Error in	Error	Standard LabVIEW error input.

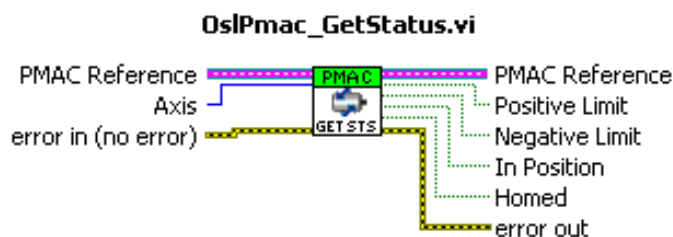
Outputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Position	Double	Current position of the axis in counts.
Velocity	Double	Current velocity of the axis in counts/millisecond.
Following Error	Double	Current following error of the axis in counts.
Error out	Error	Standard LabVIEW error output.

Description:

This VI returns the current position, velocity and following error of the specified axis. It reads the values from the PMAC in one command which is more efficient than using a separate call for each of the values.

3.3.18 OslPmac_GetStatus.vi



Inputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Axis	Integer	The axis number to check.
Error in	Error	Standard LabVIEW error input.

Outputs:

Name	Type	Description
PMAC Reference	Reference	Reference to the Delta Tau PMAC controller. This contains information relating to the Ethernet connection.
Positive Limit	Boolean	State of the PMAC positive limit flag for the specified axis.
Negative Limit	Boolean	State of the PMAC negative limit flag for the specified axis.
In Position	Boolean	In position state of the specified axis.
Homed	Boolean	Has the specified axis got a valid reference position.
Error out	Error	Standard LabVIEW error output.

Description:

This VI returns status items for the specified axis. The values returned are the positive and negative limits, the in position and homed state. It reads the values from the PMAC in one command which is more efficient than using a separate call for each of the values. The PMAC limit flags will be true if the axis is currently on a hard or (PMAC) soft limit.

4 Example Application - Step By Step Guide

4.1 Running the Application

This section describes the test application provided with the VI set, demonstrating methods for connecting to a PMAC and providing status about a selected axis. It also provides methods for setting the velocity and a demand position for the axis.

The front panel of the application is shown below.



Figure 7. Demonstration application front panel.

When the application is first opened in LabVIEW the IP Address input box is blank. Before running the VI the IP address of the PMAC controller should be entered. The default port number of 1025 should work with all PMAC controllers. Once the IP address is correct start the VI and it connects to the PMAC. If the VI cannot connect then it will stop execution.

Once connected the status items on the right hand side will update for the specific axis selected. Starting from the top right hand side:

Velocity - This shows the current velocity in counts/millisecond. It can be changed for the current axis by setting a new value in the input box.

Position Demand - Entering a new value into this input box will send a position demand to the selected axis. The position demand is specified in counts.

Home - Click on this button to initiate a home (PMAC command "#nHM" where n is the axis number) search.

Position - This indicator displays the current position of the selected axis in counts.

Following Error - This indicator displays the current following error of the selected axis in counts.

In Position - This indicator is lit if the selected axis is currently in position.

Homed - This indicator is lit if the selected axis has successfully completed a home position search.

Positive Limit - This indicator is lit if the selected axis has reached its positive limit switch (software or hardware limit).

Negative Limit - This indicator is lit if the selected axis has reached its negative limit switch (software or hardware limit).

The selected axis can be changed by simply selecting a new axis from the drop down menu on the left hand side of the panel. When a new axis is selected the status information updates to reflect the current state of the new axis.

To exit the application simply click on the Exit button.

4.2 Application Block Diagrams

The diagram below shows the overall block diagram of the application.

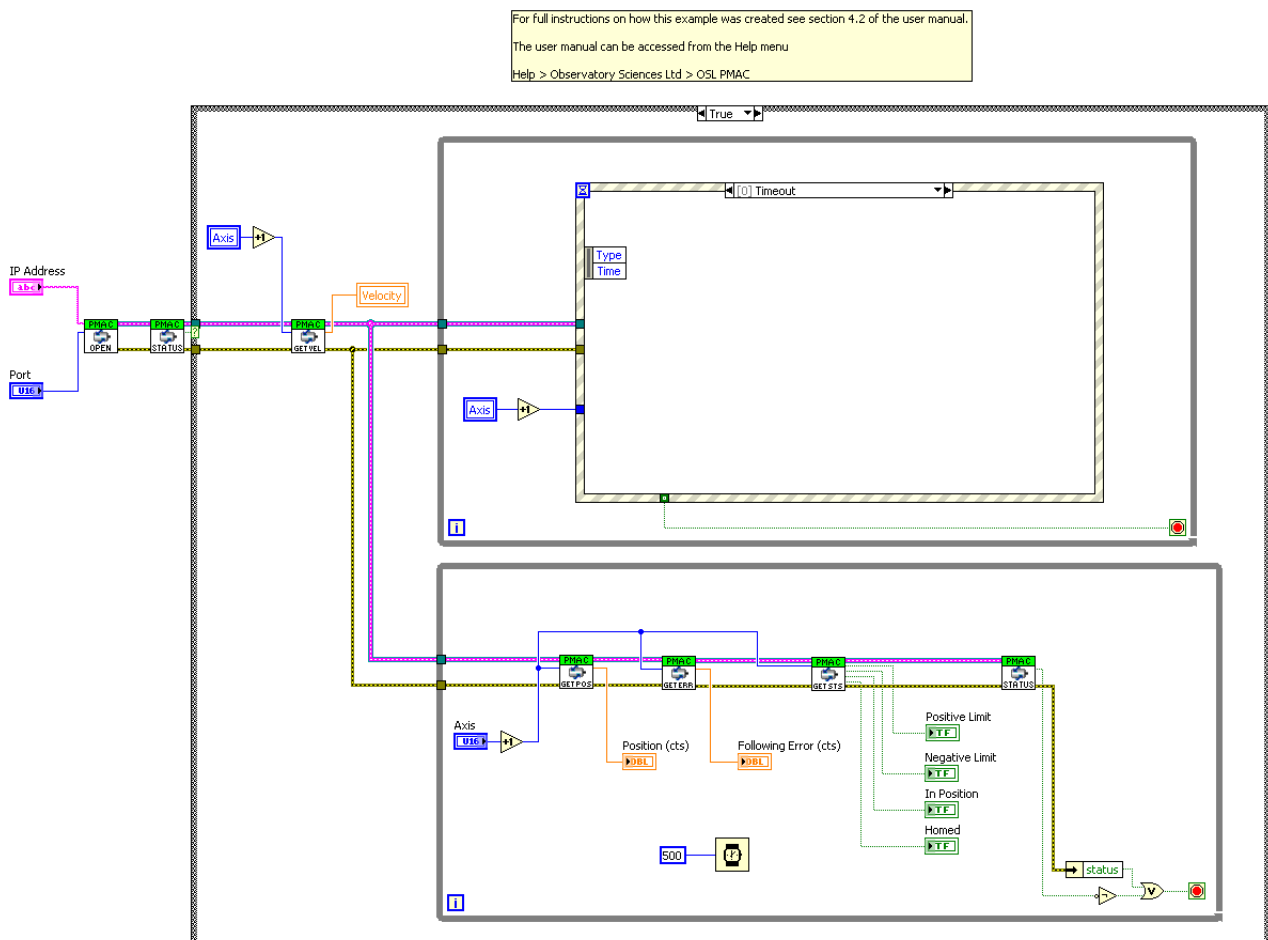


Figure 8. Block diagram for the test application.

When the application is started the values of "IP Address" and "Port" are passed into the open VI (3.3.1) which attempts to establish communications with the PMAC. The status of the connection is then checked using the status VI (3.3.3). If the status is bad then nothing more happens and the application simply stops, but if the status is good then the application enters the "True" case statement and continues. Once the connection has been made the velocity of the currently selected axis is checked using the get velocity VI (3.3.9) and then two while loops are started. The lower while loop executes with a 500 ms delay and simply checks the status of the currently selected axis. The diagram below shows the loop.

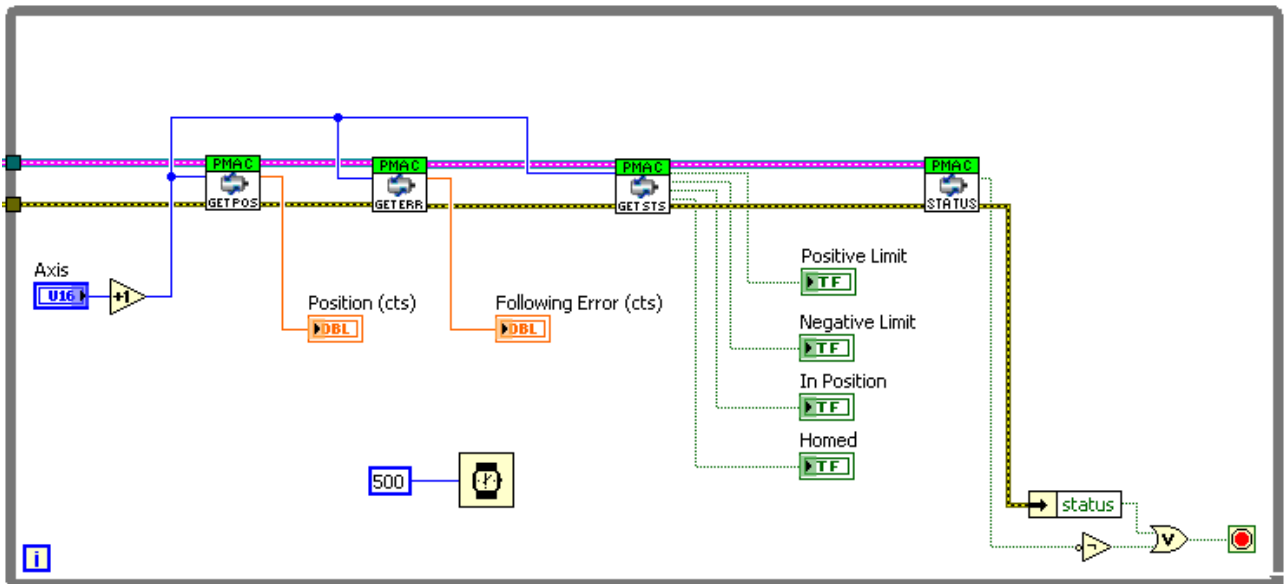


Figure 9. Test application status while loop.

The axis number is read and used to make calls to the get position (3.3.15), get error (3.3.16) and get status (3.3.18) VIs. The return values from these calls are used to populate the status indicators. The status (3.3.3) VI is also checked, and if communication is lost with the PMAC then the loop is stopped.

The upper loop in the block diagram is used to handle user events which include clicking on any of the buttons or selecting a new axis. The upper loop is shown in the diagram below.

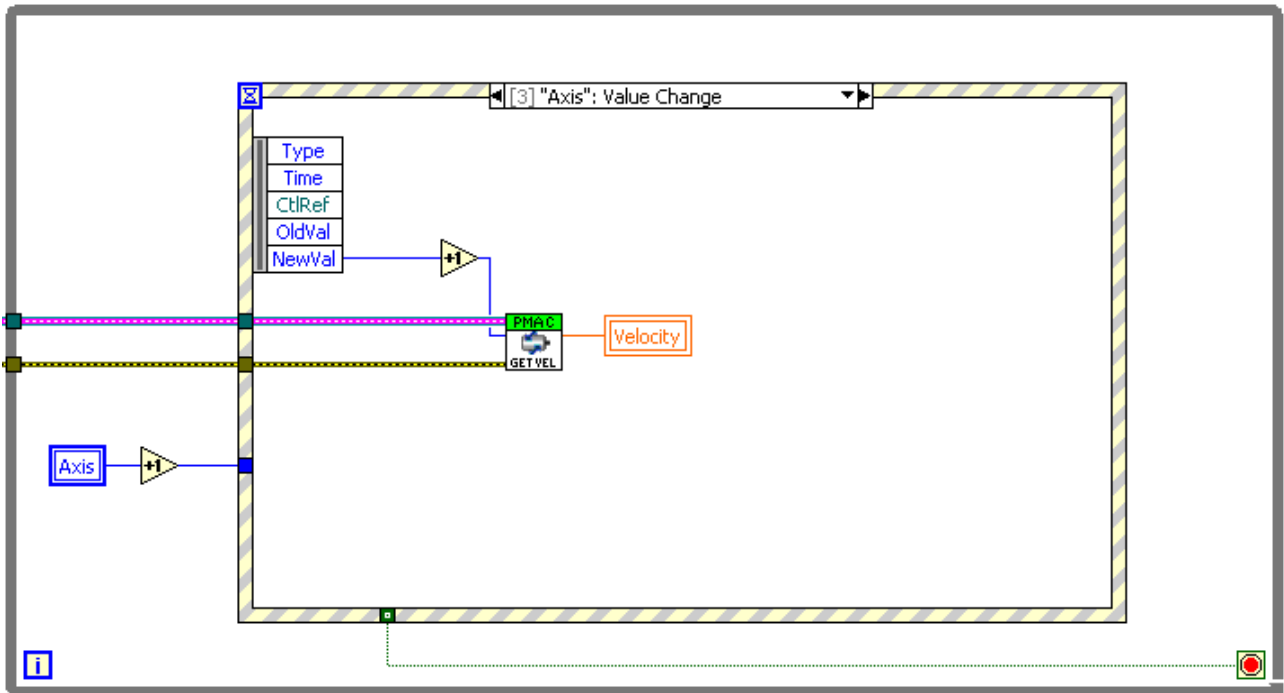


Figure 10. Test application user event while loop.

This loop contains an event structure to handle the events that might occur during operation. The selected axis number is passed into each event case. Each event case is explained in detail below.

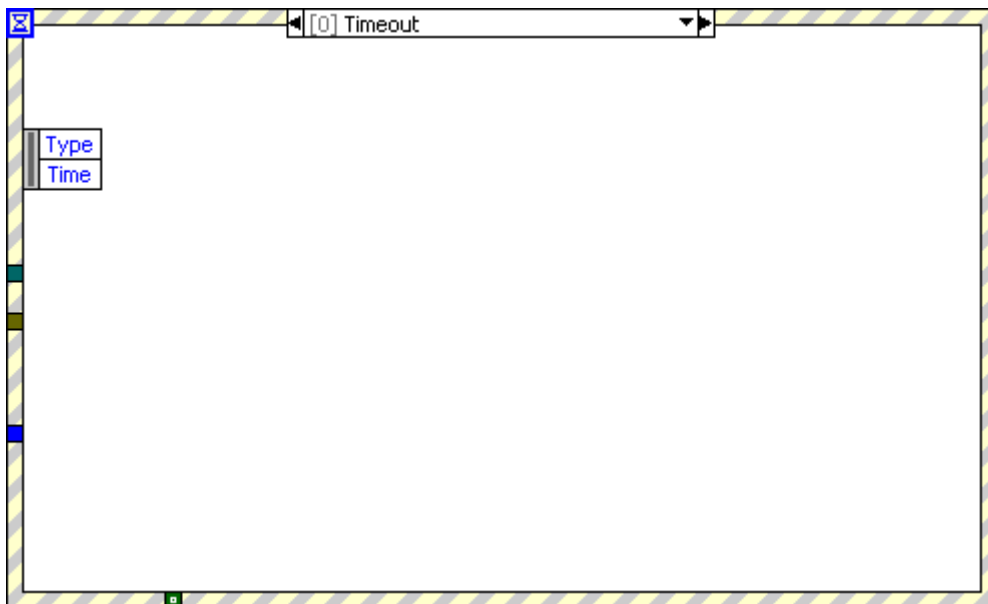


Figure 11. Test application timeout event.

This event case is unused.

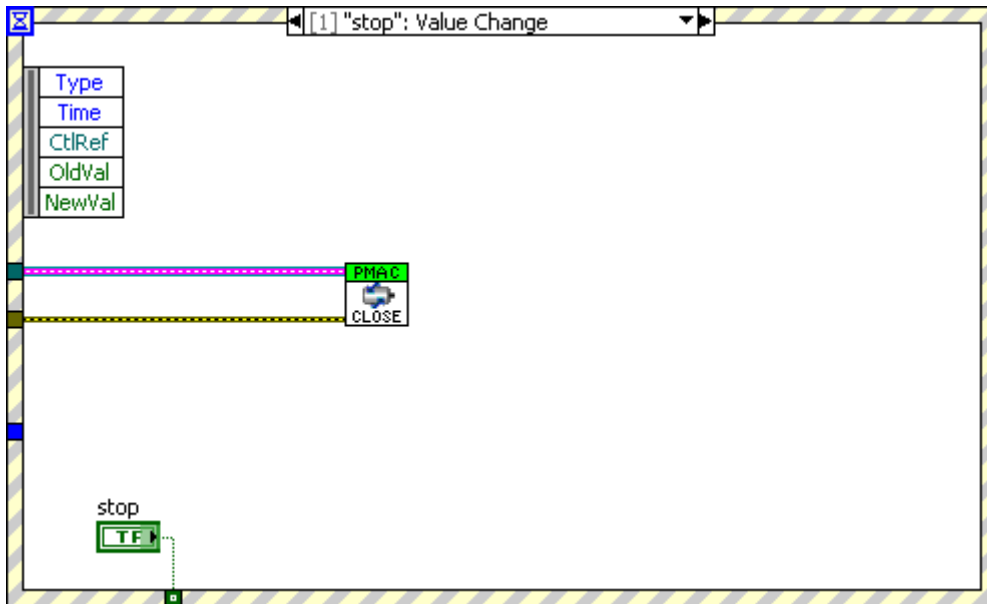


Figure 12. Test application stop value change event.

If the operator clicks on the exit button then this event case is trigger. Communication with the PMAC is stopped using the close VI (3.3.2) and the value of the stop button is passed to the loop to stop the loop from executing. This will result in the application exiting.

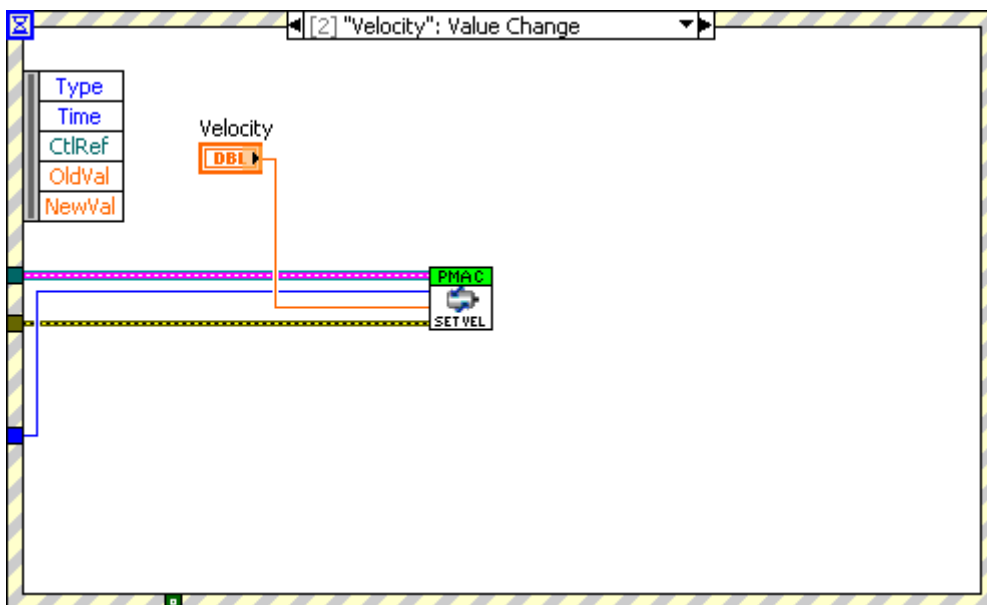


Figure 13. Test application velocity value change event.

If the value of the velocity is changed in the input box then this event case is triggered. The set velocity VI (3.3.8) is called with the currently selected axis number and the velocity value.

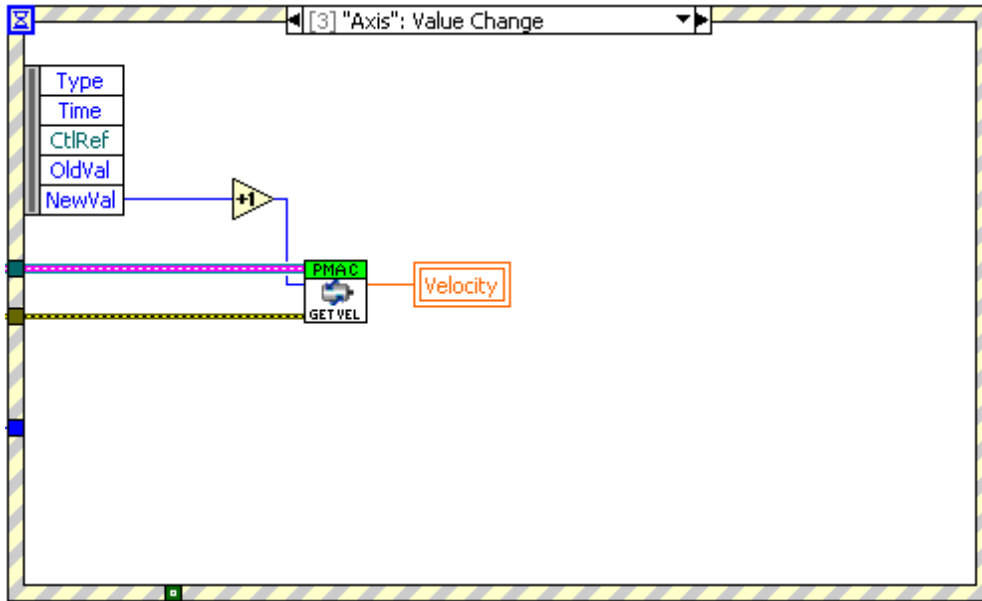


Figure 14. Test application axis value change event.

If the selected axis is changed using the drop down selection menu then this event case is triggered. The get velocity VI (3.3.9) is called with the newly selected axis value and the velocity value is set accordingly.

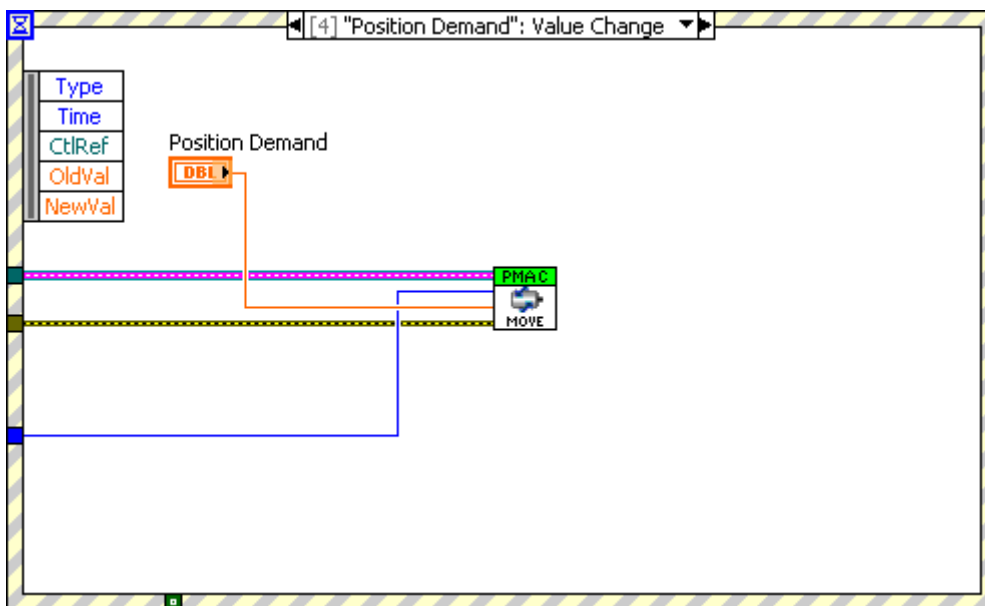


Figure 15. Test application position demand value change event.

If the value of the position demand is changed in the input box then this event case is triggered. The move absolute VI (3.3.12) is called with the currently selected axis number and the position demand value.

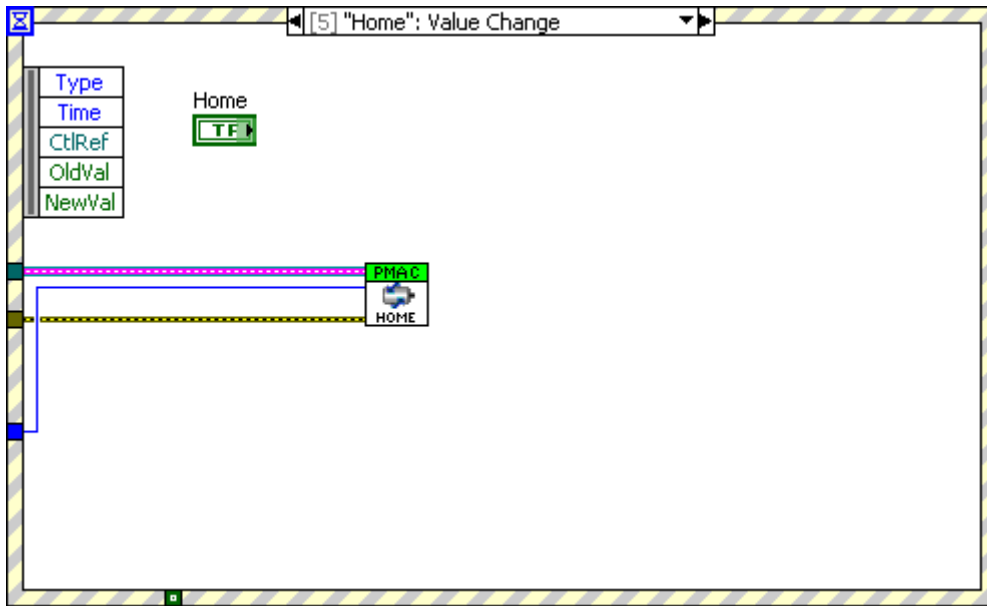


Figure 16. Test application home value change event.

If the home button is clicked then this event case is triggered. The home VI (3.3.14) is called with the currently selected axis number.