Modicon Quantum with Unity
Hot Standby
Quick Start

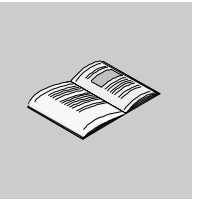
eng September 2004

35008840.00

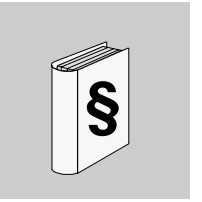
a brand of Schneider Electric

Telemecanique

Table of Contents



Safety Information



Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a Danger or Warning safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

DANGER

DANGER indicates an imminently hazardous situation, which, if not avoided, **will result** in death, serious injury, or equipment damage.

WARNING

WARNING indicates a potentially hazardous situation, which, if not avoided, **can result** in death, serious injury, or equipment damage.

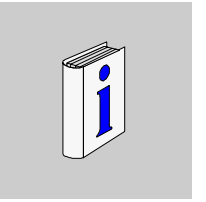
CAUTION

CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result** in injury or equipment damage.

PLEASE NOTE

Electrical equipment should be serviced only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material. This document is not intended as an instruction manual for untrained persons.

About the Book



At a Glance

Document Scope

This Quick Start guide offers the essential facts needed to start using a Modicon Quantum with Unity Hot Standby system consisting of the Unity Pro software and the Modicon Quantum Hot Standby with Unity 140 CPU 671 60. This guide augments the Modicon Quantum Hot Standby with Unity User Manual. Users of legacy Quantum Hot Standby systems should note that significant differences exist between Unity and legacy systems.
Hot Standby systems offer fault-tolerance through constantly available redundant automation control systems.

Note: Software Requirements
Required to use a Quantum Modicon Hot Standby with Unity system:
- Unity Pro 2.0 or higher
- CRA firmware: Release 1.25 or higher

Note: Who should use this document?
- Anyone who is starting to use a Hot Standby system.
- You should have knowledge of programmable logic controllers (PLCs/ controllers). Familiarity with automation controls is expected.
- You should possess a working knowledge of the Unity Pro software. Familiarity with Concept, ProWORX, or Modsoft will help.

Note: Nomenclature
This guide uses both the word, controller, and the acronym, PLC. Both refer to a programmable logic controller like the 140 CPU 671 60.

Because Modicon Quantum Hot Standby with Unity systems deliver fault-tolerant availability through redundancy, use a Modicon Quantum Hot Standby with Unity system when downtime cannot be tolerated. Redundancy means that two backplanes are configured identically.

Note: Identical Configurations
A Modicon Quantum Hot Standby with Unity system must have identical configurations:

- identical 140 CPU 671 60s which contain both a CPU and a Copro
- identical versions of the EXEC
- identical power supplies
- identical RIO Heads
- identical cabling and cabling systems
- identical I/O drops
- identical sequential placement on the backplane

Validity Note

The data and illustrations found in this book are not binding. We reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be construed as a commitment by Schneider Electric.

Related Documents

Title of Documentation	Reference Number
Modicon Quantum Hot Standby with Unity User Manual	UNY USE 10710 V20E

Product Related Warnings

Schneider Electric assumes no responsibility for any errors that may appear in this document. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When controllers are used for applications with technical safety requirements, please follow the relevant instructions.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this product related warning can result in injury or equipment damage.

User Comments

We welcome your comments about this document. You can reach us by e-mail at techpub@schneider-electric.com.

Setting-up a Hot Standby System

1

At a Glance

Purpose

This guide explains the essential steps needed to set up and start a Modicon Quantum with Unity Hot Standby system.

What's in this Chapter?

This chapter contains the following topics:

Selecting and Installing Hardware

Hardware Requirements

Ensure the required hardware configuration is available

Quantity	Equipment	Requirement	Description
2	Identical Quantum CPU racks	Must contain • one (1) Power Supply module, • one (1) CPU module (140 CPU 671 60) • one (1) RIO communication processor module (140 CRP 93x 00)	
1	RIO drop rack	Must contain • one (1) Power Supply module • one (1) RIO communication adapter module (140 CRA 93x 00)	Note: Optional: Input and/or Output modules could be installed in this rack.
1	MT-RJ/MT-RJ fiber optic cable (490 NOR xxx xx)		Establishes a direct link between the HSBY Link Port of the CPU modules

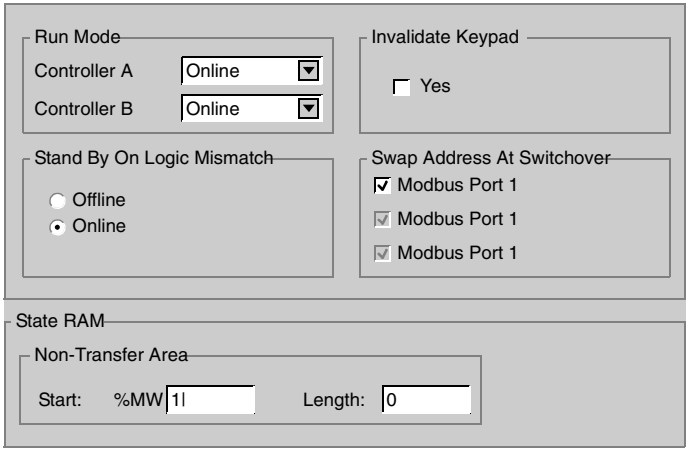
RIO Stations

Ensure that each RIO communication processor module (CRP) links to the other identical module: use two coaxial sections terminated with F connectors. Each of these sections links one CRP coaxial port and one side of a coaxial splitter (MA-0186-100). The bottom of the splitter conducts the data to the coaxial trunk line. Insert one tap (MA-0185-100) for each RIO station. The last of these taps receives one trunk terminator (52-0422-000) on its output port. The bottom of each tap has to be connected to the RIO station (CRA port) with a coaxial link equipped with F connectors.

Selecting Hot Standby Options with Unity Pro

Application Software Setup

Setting-up a Hot Standby application requires configuring four dialogs, and configuring must occur during the initial setup.



Schneider Electric recommends

Option	Set	Description
Run Mode	Online for Controller A and B	After the project download, controllers A and B are coupled together, therefore, building an effective redundant system.
Invalidate Keypad	Check box NOT checked	Control the redundant state through both the Control Register bit and keypad entries.
Stand By on Logic Mismatch	Online	The redundant system acts as a Primary / Standby system, even if a modification was made to one of the controllers' project.
Swap Address At Switchover Modbus Port 1	Check box checked	A 128 offset is added to the Standby unit's slave Modbus port address.

Non-Transfer Area

A "Non Transfer Area" may be specified by means of a starting %MW reference field and a length field.
This "Non-Transfer Area" defines a portion of the output registers' (%MW) State RAM. This defined portion (the referenced positions of data memory) will not be transferred from Primary to Standby at each scan cycle.
A null start value is an illegal entry, as %MW0 does not exist on Quantum.

Configured Options and Command Register Bits

The configured items on Hot Standby dialog, respectively the "Run Mode" selection for controller A and B, the "Invalidate Keypad" check box state, the "Standby state on logic mismatch", and the "Modbus Port Swap Address" functionality, correspond to specific bits of the system Control (Command) Register (%SW60). These bits control the operating conditions of the redundancy (these selections will dictate the bit of the Primary Unit's Control Register; this one being cyclically copied onto the Standby Unit's Control Register).

Selecting a Communication Method

Setting-up Communication

Choose one of four communication methods to communicate between the Unity Pro programming panel and the controller.

- Modbus
- Modbus Plus
- Ethernet (TCP/IP)
- USB

To select the appropriate parameters for the chosen method,

Step	Action
1	Enter "PLC | Set Address "
2	Select the appropriate communication method.
3	Set the relevant parameters.

Using the Test Connection

The "Test Connection" of the "Set Address" dialog allows you to test the effectiveness of the selected communication method and its parameters.

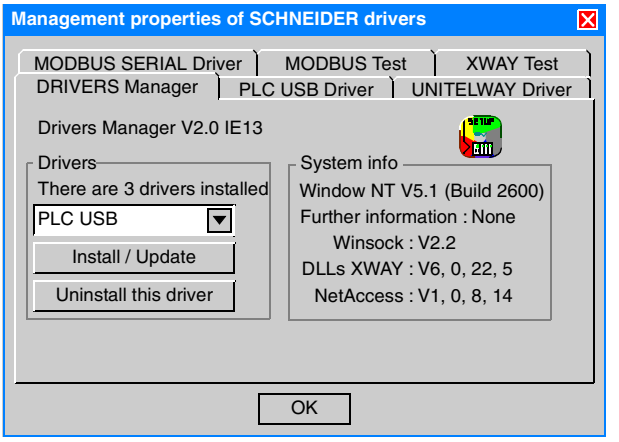
Modbus Communication

The most common communication method is Modbus.

Default Parameters
RS 232 mode
9600 bauds
8 data bits (RTU (binary) mode)
even parity
1 stop bit
1 unit slave address

These parameters may be manually changed with the keypad. For example, to speed up the transfer time, you may change the baud rate from 9600 to 19,200.

Use Drivers Manager Version 2.0 IE13 or later



Modbus Plus Communication

If a PCI or PCMCIA Modbus Plus communication card has been installed in your PC, and if you have a Modbus Plus network cable (at least a straight cable with one DB9 male connector at each side), you will also be able to communicate with the controller thorough its Modbus Plus port, provided the Modbus Plus drivers have been previously and properly installed. This port also reacts on a non configured controller unit.
Reference AS MBKT 185 gives you two of these terminal DB9 connectors
The controller's Modbus Plus port address may be manually changed with the keypad.

Ethernet Modbus/TCP Communication

If a 140 NOE 771 xx Ethernet module is installed in the rack, an Ethernet connection is possible allowing communication from your PC to the Ethernet communication module (NOE 771 xx). If you use a nonconfigured CPU unit, the non configured NOE Ethernet port will respond to an IP Address that is derived from its MAC address.
Example: suppose the MAC Address of the targeted NOE adapter is 00.00.54.10.BF.A6. The default IP Address would be 84.16.191.166
Provided your PC has an IP address (and a subnet mask) compatible with this default IP address, a ping command would easily check if the NOE adapter responds to that address. Through this default IP address, you can download an application program into the controller.
The project configuration will usually assign a particular address to this port, and after starting, the controller responds to this new IP address.

USB Communication

Provided you own the required communication cable (referenced UNY XCA USB 033), you will be able to use the USB port now available with the new Modicon Quantum High End CPUs.

USB link switchover is not possible because the USB link is connected to only one CPU, allowing Unity Pro to communicate only to this local controller. Therefore, USB cannot be used for transparent access to the Primary controller.

Setting-up Primary and Standby Controllers

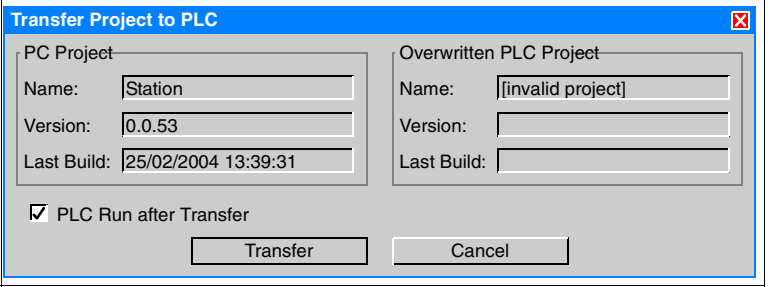
Connecting to the Primary-to-be Controller

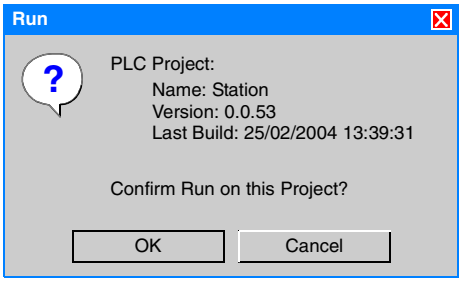
You can set up a Modicon Quantum Hot Standby with Unity redundant system without any effective application program: under these conditions a project could be only made using the controller's configuration.

Step	Action
1	From the main menu, run Build I Analyze Project.
2	Correct any errors.
3	Run Rebuild All Project.
4	Assuming there is no error, power up one of the controllers.
5	When the keypad displays "No Conf" (having displayed "Initializing" for a few seconds after the power up), you then may try to establish a connection with this controller using the selected communication path.
6	From the main menu, run PLC I Connect.

Transferring Data on to the Primary-to-be Controller

Assuming the connection has been successful, you may download a project into the controller's memory.

Step	Action
1	From the main menu, run PLC I Transfer Project to PLC. 
2	You may check the "PLC Run after transfer" check box.
3	Click Transfer.

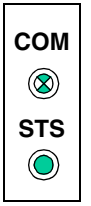
Step	Action
4	This action opens a dialog box showing the project name, its version number and the date and time of the last build.  Note: As far as the controller's memory is supposed to be cleared on a new CPU unit, these kinds of information are available for the "PC project" only, and not for the due to be "Overwritten PLC Project".
5	At the end of this transfer, confirm the switch to the Run State for this Project by clicking OK in the Run dialog.

Clearing a Controller's Memory

A full clearance of the controller's memory requires about 2 days, power off, and battery connection removed. A quick way to force a controller to enter a "No Conf" mode is to disconnect the communication cable during the download process of an application project whose configuration is different from the one staying in the controller's memory. Pay attention, anyway: this type of operation doesn't realize a complete reset of the memory (e.g. the communications parameters are not reset)

Checking the CPU's Status

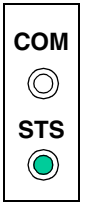
At this stage, look at the controller's front panel. The COM LED is blinking, and the STS LED is in a steady-on state. This condition usually indicates that the system is not redundant and that the controller's Redundant Copro completed both its boot after a power up and its self-tests.

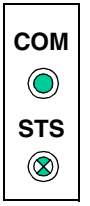


The keypad should display a "Run Primary" indication (in the relevant case, just act on the ESC key in order to make the keypad display its first "page").

Setting-up the Standby-to-be Controller

You may now power up the second controller, which will be the Standby-to-be controller.
It's supposed here that the Standby unit controller's memory is clear or at least in a "No Conf" state. In these circumstances, the memory update of the Standby unit, starting from the Primary unit's memory, will be done automatically. An automatic transfer should also occur when the configuration staying in the Standby-to-be controller is different from the one staying in the Primary Controller.
In a first step, the keypad of this second controller displays "Initializing" for a few seconds after the power up. During this phase, looking at this controller's front panel, you should notice that the STS LED is lit in a steady on state, indicating that the system is not (yet) redundant. By the way, this also indicates that the controller's Redundancy Copro has done its boot after a power up, and has ended its self-tests. The COM LED is off for a while, indicating there is no communication activity between Primary and Standby controllers.





The COM LED will then blink for some seconds, and then switch to a steady on state, so indicating there is some communication activity between Primary and Standby controllers. At that point, the STS LED should have turned to a blinking state, indicating so that the system is now redundant, and that data are exchanged from the primary to the Standby controller.
The keypad of the first controller should display "Run Primary" and the second controller should display "Run Standby". To return the display to the default, press the ESC key and the keypad displays its first page.

Checking the CRP LED Status

When the Hot Standby system is operational, the green "Ready" LED is steady on for both the Primary's and Standby's RIO communication processor module (CRP). Also, the Primary CRP's green "Com Act" LED should be steady on, while the LED of the Standby CRP should be blinking.

Checking the CRA LED Status

When the Hot Standby system is operational, the green "Ready" LED is steady on for the RIO communication adapter module(s) (CRA).
Also, the CRA's green "Com Act" LED will be steady on if any I/O module has been configured.
Additionally, the CRA's red "Fault" LED label will be steady on if the I/O modules that are physically recognized on the rack do not match the I/O modules that have been configured for that rack.

Understanding the Status Register

Primary and Standby Unit System Status Register (%SW61)

Schneider Electric recommends that you become familiar with the Hot Standby status register bit positions.[1.]
Bits 15 and 16 identify the state of the controller to which your PC is connected. Reversely, bits 13 and 14 identify the state of the other controller.
For example, with the programming panel connected to the Primary unit, a bit sequence such as (1,1,1,0) for bits 13 to 16 would normally indicate that this controller is the Primary unit, and the other controller is the Standby unit. Reversely, with the programming panel connected to the Standby unit, the same bit sequence should be (1,0,1,1).[2.]
These bit indications must match the information displayed on both keypads.
A very important piece of information, given by bit 11, permits you to know if the current controller is to be considered as the "A-Controller" or the "B-Controller". This A versus B attribute is selected after identifying the MAC address of the embedded Ethernet port: A-Controller is the controller whose CPU module shows the smaller MAC address value. Reversely, B-Controller is the controller whose CPU module shows the greater MAC address value.[3].
Unity use the IEC convention for referencing registers so

- Bit 13 is %SW61.3
- Bit 14 is %SW61.2
- Bit 15 is %SW61.1
- Bit 16 is %SW61.0

[1.] A dedicated EFB exists (HSBY_ST), which allows the user to split onto application boolean variables the state of these bit positions. In turn these bits may easily be used to animate an Operator Screen on Unity Pro. Anyway, no doubt a user trained to directly identify these bits will be an efficient one.

[2.] The fact that the status register holds potentially different contents on Primary and Standby proves that the contents of the status register on the Standby is not updated with the contents of the Primary's status register.

[3.] That means that some provisions should be made, at the application level, in order to be able to cope with a maintenance operation resulting in a MAC address relative height inversion.

Understanding the Control Register

Primary and Standby System Unit Control Register (%SW60)

Schneider Electric recommends you become familiar with the Hot Standby control register.
Locate the "Invalidate keypad" bit, which is the LSB. The Invalidate Keypad bit (bit 15) dictates the state of the "A-Controller", and bit 14 dictates the state of the "B-Controller".
An additional screen menu, named Diag, appears on the controller's LCD display screen. These menus describe the state of t5he controller.

LCD Display Screen Menu	Description
Diag: off keypad	Offline mode selected using the keypad
Diag: off %sw60	Offline mode selected using the control register and possibly the keypad
Diag: plug&run	Online mode selected using either the keypad or the control register

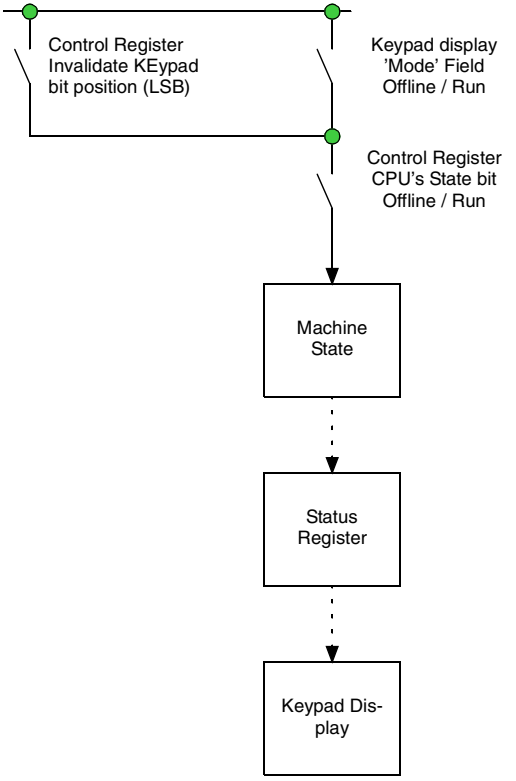
The Diag field indicates whether the current CPU is in Run mode (plug&run) or not. If the current controller is not in RUN mode, the keypad displays the reason for the offline state: keypad and / or control register.

Additional Information about Using the Control Register

If the keypad LED displays a "Mode" field in the "Hot Standby" screen, that gives the intention emitted by the user through the keypad, concerning the Run/Offline (i.e. coupled/uncoupled) state of the current CPU in regard with the Hot Standby system.
In turn the Hot Standby Control Register (%SW60), through bits 15 and 14, permits an external host system to set the mode to either a Run or Offline state for the current CPU and/or the alternate CPU. These bit positions are accessible by the application program and / or an external system such as a SCADA system or a programming panel.
The intentions emitted by the Control Register bits always play their part in the final decision, concerning the Run/Offline state of the CPU units.
The intentions emitted by the keypad Mode selections (Run versus Offline) play their respective part in the final decision, concerning the Run/Offline state of the CPU units, provided the Keypad Disable bit position (LSB) in the HSBY Control Register.[1.] is in the False state ('0'). In these conditions, for a given CPU, either the Control Register bit position governing its Run/Offline CPUs State, and the Mode field value of the CPU's keypad, should ask for a 'Run', in order to get an effective 'Run' state. [2.]

[2.]When keypad becomes validated again, the keypad selections take effect immediately.[1.]The Standby's Control Register is updated on each scan cycle with the contents of the Primary's Control Register 1.

On the keypad, the effective state of the current CPU is displayed by a "State" field in the "Hot Standby" screen menu.



Modifying a Programmed Controller

Further Hot Standby setup

I you intend to use a controller that has had program downloaded, a recommended practice will be to adopt the same kind of progression than the one previously described, in order to succeed in the setup of this Hot Standby system, i.e. one controller at a time.
So, power down one of the units, in order to work at first on a unique one. We suggest you work with the controller having the lower MAC address: it acts as Controller A, initially standing as the Primary unit. A download operation should normally place this unit into a Primary state. Power up the second unit. Depending on the current contents of this 'to-be-Standby' unit, a transfer may automatically occur. If not, it may be necessary to manually ask for this transfer to occur. A safe practice would be to systematically execute a transfer from Primary to Standby, downloading a newer project.
In that latter case, the transfer has to be asked for from the Primary's keypad. ("PLC Operations" => "Hot Standby" => "Transfer" => Enter)

Note: Verifying

- If you fail to establish a connection, first check the cable, but also, use the keypad to check the parameters governing this port. If required, feel free to modify these parameters according to your needs.
- Once having done an important operation on a CPU, such as a project download, it's a good practice to verify with the keypad if the operated machine has now got its expected state. For example, a quick check will verify either on supposed-to-be-primary, and respectively on supposed-to-be-standby, the respective Modbus Plus Port address.

Making Online Modifications

Introduction

Provided the corresponding bit position 13 has been set to 1 (Bit 13 is now %SW60.3 in IEC convention), the Quantum Hot Standby with Unity System will remain "coupled" in case a logic mismatch occurs.
In these conditions, let's examine the operations sequence that would have to be conducted, starting from a coupled system, to get a modified coupled system.

Data / Code Changes_ Primary

Up to a certain amount, the target modifications done on the application project can presumably involve any change, deletion, or addition, either to data or code. These modifications should not involve the controller's configuration, however.
Considering that the update transfer may only occur from Primary to Standby, and considering that only the Primary will execute the whole program, the modifications will usually occur on the Primary.
Per default: Modifications done on the Primary Unit:

- Make the required changes on code and/or data
  Note that a code section will switch to a blank background color as the first change is done on that section
- Run the "Build | Build Project" menu command
  Provided the modifications induced no error, the background color of the modified sections turns back to grey.
- Execute a transfer from modified Primary to Standby, either from the Keypad, or from the control Register - %SW60 - (%SW60.5 asks for this transfer when set to '1')

Data / Code Changes- Standby

Alternative: Modifications done on the Standby Unit:

- Make the required changes on code and/or data
  Note that a code section will switch to a blank background color as the first change is done on that section
- Run the "Build | Build Project" menu command
  Provided the modifications induced no error, the background color of the modified sections turns back to grey.
- In order to be able to execute the update transfer, execute a switchover, switching the Primary unit to Offline first, then to Run state. This switchover may be asked for from the keypad, or by mean of the control register on the Primary Unit.
- Execute a transfer from (new) modified Primary to (new) Standby, either from the Keypad, or from the control Register - %SW60 - ((%SW60.5 asks for this transfer when set to '1')

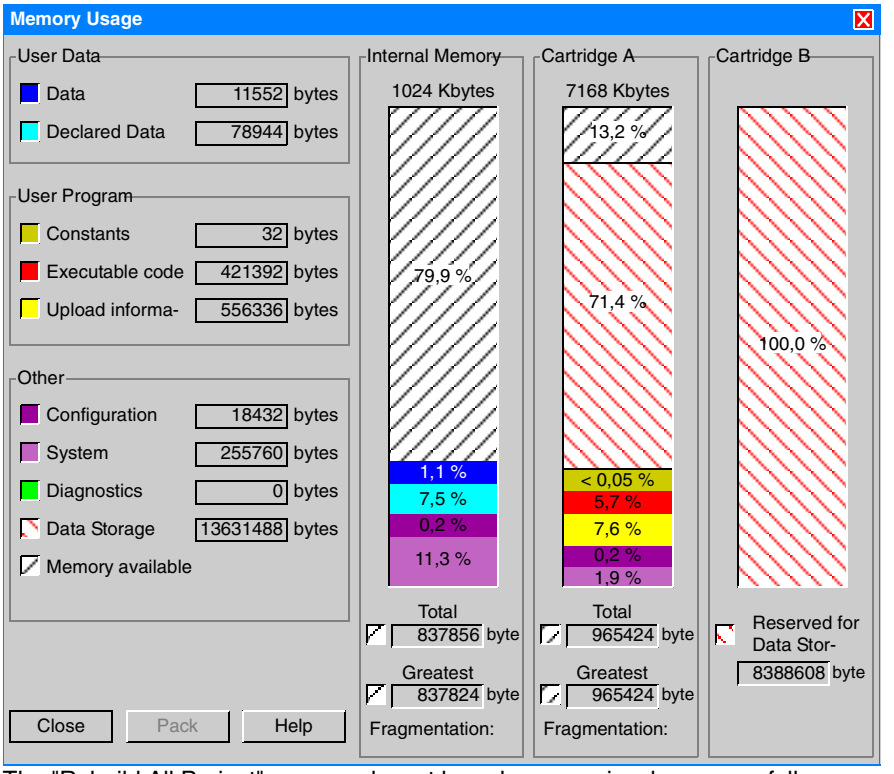
Configuration changes

In case of a configuration modification, a "Rebuild All Project" would then be necessary to assert the project's modifications; what would in turn necessarily imply a "Transfer Project to PLC", and so forth a "Stop" command, directed to the controller.
This case corresponds in fact to the case treated in previous chapter 5: "Further Hot Standby setup"
In these conditions, the Hot Standby architecture is no more active during all this modification entry process! More, during at least one moment, there is one unique controller; and this controller ha to be stopped in order to accept the "Transfer Project to PLC" command.

Understanding Memory Consumption

Memory Consumption

A powerful tool provided by Unity Pro is the Memory Usage functionality, that shows the memory resource occupation of the target project. This evaluation is called running the menu command "PLC I Memory Consumption ".



The "Rebuild All Project" command must have been previously, successfully conducted for Memory Usage command to be accessible. The Memory Usage function will not be accessible if not enough memory space has been allotted to this target project.

Particularly, for a Quantum Hot Standby with Unity system, the values reported by the two first numerical fields must not exceed the scope of a HSBY system. Respectively:

- "Data" [1.] : 128 k-byte max
- "Declared Data" [2.]: 512 k-byte max

[1.] read State RAM, i.e. referenced/located variables

[2.] read State unreferenced/unlocated variables, implicit data, and DFBs internal data