7... HYPERNET™ ICON BASED PROGRAMMING

OVERVIEW

The HyperLogger operates in the field based on a program loaded into its memory called a Program Net (Figure 7... -1). The Program Net provides instructions for the HyperLogger including which channels to sample, when to sample, how to process the incoming signals, when to output alarms, and much more.

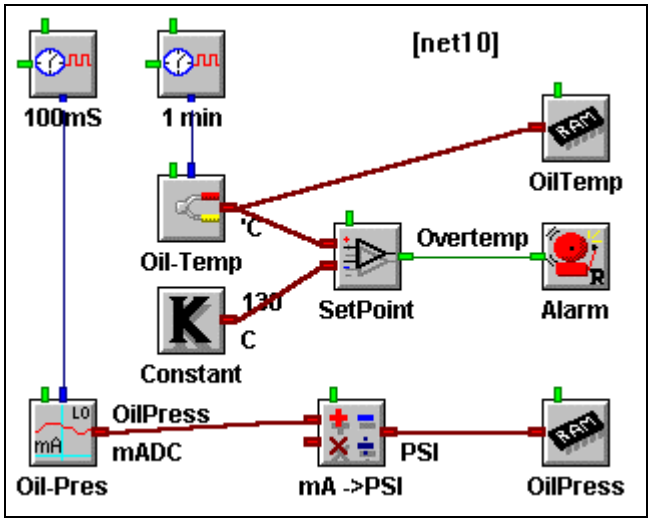


Figure 7... -1: Example Program Net

Development of a Program Net is done on a PC running HyperWare. After development, the Program Net is uploaded directly to the HyperLogger memory via a serial communication link or indirectly via upload to a PCMCIA card. The PCMCIA card can then be plugged into the HyperLogger. Actual development of the Program Net is performed in the *HYPERNET WINDOW* and the serial transfer of the Program Net from the PC to the HyperLogger (or to the PCMCIA card) is performed from within the *HYPERCOMM WINDOW*.

Program Nets are developed through the following sequence of steps:

- Create a new Program Net file for the connected HyperLogger (which automatically determines the installed hardware in the connected HyperLogger)
- Add desired functions to the Program Net by dragging various icons onto the HyperNet workspace
- Add connections between icon terminals indicating signal flow
- Configure the various icon options (eg Celcius or Fahrenheit, type of thermocouple, filtering, equations, etc)
- Configure the Global icon
- Save the Program Net and/or transfer it to a connected HyperLogger for execution.

This chapter describes the HyperNet development environment and provides detailed instruction on constructing Program Nets. Detailed icon configuration information is provided in the Master Icon Reference in Appendix A. For the

technically curious, additional information on the theory of operation of Program Nets is available in Appendix G.

AN EXAMPLE PROGRAM NET

A simple Program Net is shown in Figure 7... -2. In this Program Net, two thermocouple inputs (represented by the two matching icons near the left of the workspace) are sampled periodically (based on the Sample Rate Clock connected to the top of each of the Thermocouple icons) and their values are stored in HyperLogger memory (RAM chip icons). Additionally, the difference between the two thermocouple channels is calculated (by the Math icon) and stored in memory.

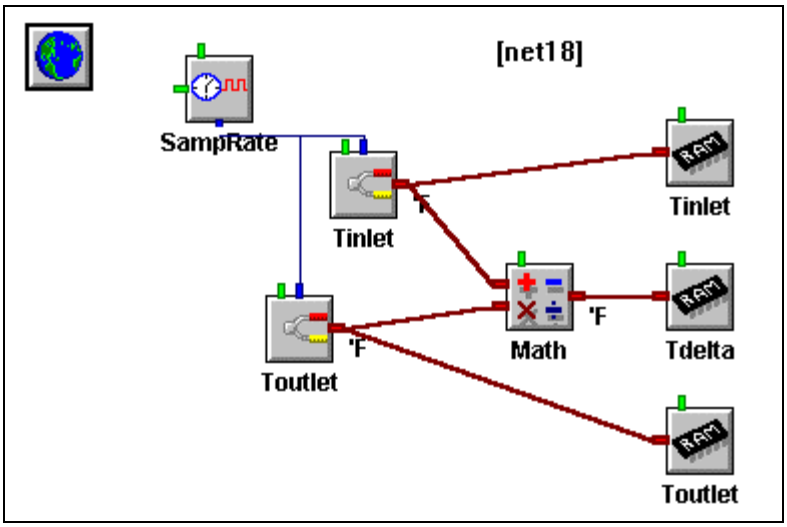


Figure 7... -2: Temperature sampling Program Net

HYPERNET DEVELOPMENT WINDOW FEATURES AND TOOLS

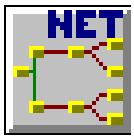


Figure 7... -3: HyperNet window button

Accessing the HyperNet Window

To enter the *HYPERNET WINDOW* from within the *HYPERCOMM WINDOW*, click on the *HYPERNET* button on the toolbar. The HyperNet Window will open

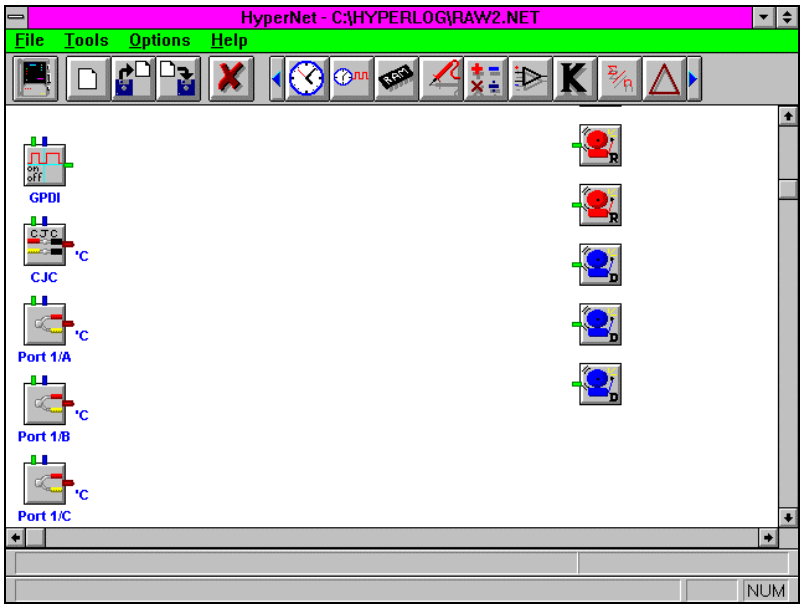


Figure 7... -4: HyperNet, Program Net development window

(Figure 7... -4) displaying the default Net or the last edited Net.

Returning to the HyperComm Window

From within the HyperNet Window, to return to the HyperComm Window, click on the HyperComm button (small HyperLogger graphic) at the left end of toolbar.

HyperNet Window Topology

When the HyperNet Window opens, a default Net or the last edited Net will be displayed on the workspace (Figure 7... -4). Use the slide bars at the right edge and bottom of the workspace to shift the display.

At the top of the window is a Menu Bar and below that, an Button / Icon Tool Bar. Passing the cursor over the various icons results in a short descriptor display on the Status Bar at the lower left corner of the window.

Descriptions of the main button / icon groups follow:

Figure 7... -5: Return to HyperComm button

HyperComm Access

A single click returns the screen to the HyperComm Window. If the Program Net currently displayed in the workspace has been changed, a dialog will open prompting the User with an option to save the edited Program Net.

Create New Net

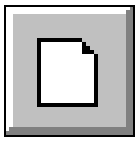


Figure 7... -6: New Net

Clicking on New Net will automatically query the HyperLogger, then update the workspace display showing icons for the connected HyperLogger's hardware configuration including any User installed modules and options. This requires a serial connection between the HyperLogger and the PC. To have a serial connection, the HyperLogger and the PC must be linked via RS-232 or Modem *and* the connection must be established from within the HyperComm Window (Chapter 5).

Open Net and Save Net

Figure 7... -7: Open and Save Nets

Clicking on the Open Net button (or selecting *File/Open Net* from the menu bar) results in the opening of the standard File Open dialog box with the default file sort set to *.NET. Previously saved Program Nets can be retrieved.

The Save Net button will save the currently displayed Program Net to disk under the current Program Net filename with the extension *.NET. If a filename has not been assigned, the standard File Save As dialog box will open, allowing for User input of a Program Net filename.

> ***TIP:*** *Use the drop down menu selection `File / Save Net As' to save Nets with a different filename.*

Delete Icon or Connection (Link)

Figure 7... -8: Delete Button

To delete a connection (link) between icons, click on the Delete icon (the Status Bar will display *Select Object to Delete* and the cursor will change shape) then on one of the ends of the connection to be deleted. (To select the end, the cursor must be over the icon terminal). To delete another connection, repeat the procedure. If multiple connections originate at an icon terminal, they will all be deleted.

To delete an icon, click on the Delete button and then on the icon to be deleted. If an icon is deleted, all connections into and out of the icon will also be deleted.

If the Delete function has been selected and no items are to be deleted, clicking on any open space in the workspace will disable the Delete function.

Program Net Icon Selection Bar

Included in the Program Net Icon Bar is a collection of various icons to be used in the development of Program Nets. Clicking on the arrows at the left and right edge of the bar will scroll the bar left or right displaying additional icons.

Figure 7... -9: Icon assortment (with scroll arrows at each end)

PROGRAM NET CONSTRUCTION

Program Nets are developed in HyperNet and saved as files with the filename extension *.NET.

Program Nets must be developed to match the existing hardware installed in the target HyperLogger. For example a Program Net that uses a modem function will only work in a HyperLogger that has a modem installed.

Configuration details that must match include:

- Interface Modules - Program Nets must match the HyperLogger installed Interface Module types.
- Port - The Port used in the Program Net and the Port used for an Interface Module in the HyperLogger must match.
- Channel Configurations - Interface Modules equipped with hardware configuration switches must have their switches set to match the Program Net on a channel by channel basis.

> *For Example: If an HLIM-1 is installed in Port 3 and has the Channel B configured (via the Channel B hardware configuration switch) for VDC-HI, then any Program Net loaded into the HyperLogger must have Channel 3B used as a VDC-HI channel (or optionally, not used).*

If a Program Net that does not match the target HyperLogger hardware configuration is uploaded to the HyperLogger memory, a configuration mismatch error will display and the upload will not occur. This checking of compatibility is handled automatically within HyperWare during the upload process and insures that hardware and software compatibility exists.

> ***TIP:*** *Users that have a number of HyperLoggers in use may find it convenient to create a subdirectory in their HyperWare directory for each of the HyperLoggers with which they work. The subdirectory names may correlate to the HyperLogger ID or Unit Name which are User programmed (See HyperNet Programming in Chapter 7). As Program Nets are developed for each HyperLogger or configuration, they can then be saved into the corresponding subdirectory.*

To Develop a New Program Net...

To develop a new Program Net, the target HyperLogger hardware configuration must be known. After configuring the HyperLogger hardware as required for the data logging application (ie installing modules, setting channel configuration switches, etc), establish a serial connection to the HyperLogger from the HyperComm Window.

Switch to the HyperNet Window and click on the CreateNew Net button *and the connected HyperLogger will be automatically polled for its hardware configuration.* The workspace will update and show icons representing the standard System Base hardware and any User installed hardware.

Before editing of the workspace begins, this unconnected Program Net should be saved to disk by clicking on the File Save button or using the `File / Save Net As' selection from the menu. This unconnected Program Net can then be used as a starting Net for development of varying Program Nets based on the same HyperLogger configuration.

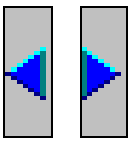
As Program Nets are developed, they should be saved with a filename different from this basic Program Net. Use the `File / Save Net As' drop-down menu to save Nets with different filenames.

To Open an Existing Net for Editing...

If a previously developed Program Net is to be edited, open the Program Net file by clicking on the Open File button and selecting the desired file.

Icon Placement

To add an icon from the Tool Bar to the workspace, click on the desired icon and while holding the mouse button down, drag the icon onto the workspace. Release the button when the icon is positioned in the approximate desired location. Icons can be relocated within the workspace by the same technique... even after connections have been made.



The Icon Toolbar contains more icons than are visually displayed at the top of the window. To see additional icons, click on the left and right arrows at the ends of the boolbar to spin to additional icons.

The HyperNet workspace utilizes dynamic panning. As an icon (or connection) is dragged near the edge of the screen, the workspace will pan.

Using Grids

If desired, a grid structure can be enabled on the workspace that provides a visual grid and/or `snap to grid' function. Select `Options / Grid' and select the desired operation.

The Snap function is merely an aid to align icons neatly within the workspace.

Changing Fonts

Labeling text surrounds icons as they are placed. The font, size, color and effects used for this text can be changed through the drop-down menu `Options / Font' and its corresponding dialog.

Icons

Icons are the main building blocks used in the development of a Program Net. Icons within a Program Net graphically represent different items ranging from hardware input channels to intermediate processing functions to hardware outputs and more. A partial listing of available HyperNet icons is in Table 7... -1.

NOTE: A complete icon listing with detailed setup and application information is supplied for reference in Appendix A.

Input Icons	Output Icons	Processing Icons	Special Icons
VDC-LO	Relay Output	Math	Sample Rate Clock
CJC	Pager Alarm	Delta Function	Global Settings
Thermocouple	LCD Message	Average Function	Warm-Up Timer
Event	Digital Output	Count Accumulator	Probe Point
Frequency		Comparator	
Count	Memory	Logical AND	
Start/Stop Clock		Integral Function	

Table 7... -1: Partial listing of icon functions available for Program Net construction

Icon Topology

Icons share many similar features including their graphic appearance (input and output terminals, etc) and configuration techniques. Figure 7... -10 shows the topology of an icon with its various terminals.

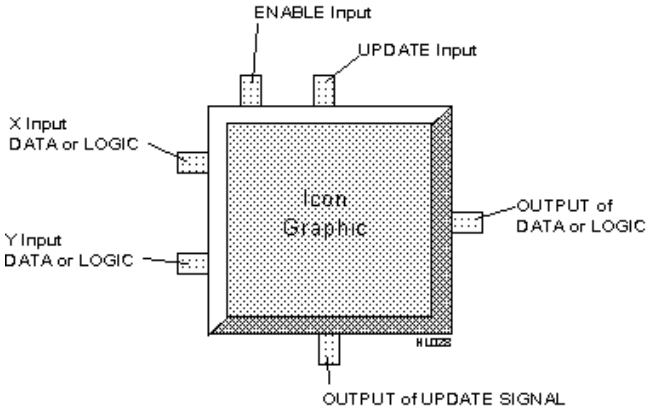


Figure 7... -10: Icon topology and terminal types

TERMINALS

All icons (except the Global Icon) have terminals for the addition of signal connections. An explanation for the various types of terminals follows:

LOGIC / DATA INPUT TERMINALS

On the left side of the icon are typically one or two inputs for Logic or Data signal types. Data enters the icon for processing through these Input terminals. Some two input icons (eg Math) do not require signals to be connected to both Input terminals for operation.

LOGIC / DATA OUTPUT TERMINAL

On the right side of the icon is a single Output terminal. After processing of an input signal(s) is completed, the output is updated. Depending on the type of icon and the User configuration, this Output terminal may or may not be updated every time the Update and/or an Input terminal is updated. The Enable input (description follows), the type of processing that the icon performs, and User specified parameters within the Configuration dialog all effect when the Output terminal is updated.

For example, when using an Average icon, the Output terminal will only be updated with a new value after a User specified number of inputs have been averaged.

ENABLE INPUT TERMINAL

Many icons have an Enable input terminal located near their top left corner that will accept a Logic input (True/False). Depending on the state of the input signal, the icon is enabled or disabled for processing.

> *NOTE: if the Enable input is not connected, it defaults to the Enabled state.*

A simple application of the Enable input might be in an engine temperature recording application. A Thermocouple icon may be enabled / disabled by a Logic signal that is True when the engine ignition is ON. With this configuration, temperature recording will only occur when the engine is running.

The Enable input can also be used for optimizing the speed of Program Nets and/or minimizing the amount of data collected as the processing normally done by an icon in a Program Net is not performed if the Enable input is False. For example, a Program Net may be built that has several input temperatures that are scanned on a fairly high speed basis. By use of the Enable terminal, the data flow to HyperLogger memory may be disabled during normal operating conditions and enabled when abnormal temperature readings are detected.

UPDATE INPUT TERMINAL

The Input signal icons (Thermocouple, VDC, GPDI, etc) are all equipped with an Update input terminal located in the center top of the graphic. Whenever this Update terminal receives an Update command, it proceeds to update its output terminal value (assuming that its Enable terminal is True or not connected).

UPDATE OUTPUT TERMINAL

The Sample Rate Clock and the Warm-up icon have a special output terminal located at the bottom center of the graphic called an Update output terminal. This terminal sends a command to the connected icon to Update its output (eg take a sample, process an equation, etc).

Details on the Sample Rate Clock and the Warmup icon are covered in the Master Icon Reference in Appendix A .

Configuring Icons

Most of the icons within HyperWare must be configured before they can be used. User configuration entails selection of various icon operational parameters such as names, input ranges, sampling rates, etc and is simply done through dialog boxes associated with each icon.

To configure an icon that has been placed on the workspace, double-click on the graphic and an Icon Configuration dialog box will appear. Each icon has a unique dialog box with programmable parameters to meets its needs. However, many parameters are common to the different types of icons. A Thermocouple Input icon Configuration Dialog is shown in Figure 7... -11 and a description of configuration parameters typically seen in a configuration dialog box follow.

Various standard Windows techniques are used to select the different parameters within the dialog box from text entry and editing to selection via radio buttons.

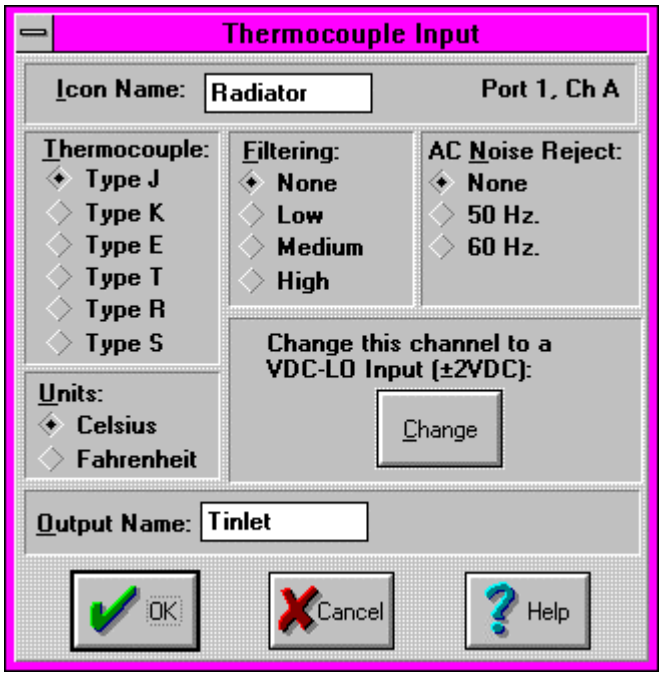


Figure 7... -11: Thermocouple input icon configuration dialog box

ICON NAME

Each icon can have an 8 character name assigned that displays on the workspace under the icon.

OUTPUT NAME

The output signal can be assigned an 8 character name which displays above the Output Terminal of the icon. This name is

commonly referenced by icons connected to this Output terminal. For example, if this Thermocouple icon were connected to one of a Math icon's input terminals, the name Tinlet would be referenced within the Math icon as an Input terminal name.

UNITS

Many of the icons can output their signals with various units such as Degrees C or F, V or mV, Degrees or Ohms, etc. Radio buttons are typically used to select one of the Unit types.

SIGNAL TYPE OR RANGE

This Thermocouple icon supports 6 different types of thermocouple. Similarly, other icons have User configurable ranges or types.

OTHER PARAMETERS

Most of the icons have additional parameters such as filtering, equations, data types, etc that are all User programmable.

Icon Assortment

A complete reference listing of all of the icons available within HyperWare is included in the Master Icon Listing in Appendix A. Details on Configuration, applications, and proper usage are described.

Global Icon

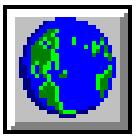


Figure 7... - 12: Global icon

One special icon that is included in every Program Net is called the Global icon. Within the Global icon's Configuration dialog are options for various global Program Net options. These options include Program Net Name, memory utilization mode, clock resolution, and modem paramteres. The operation of this icon should be reviewed in the Master Icon Reference in Appendix A before attempting to construct a Program Net.

Making Connections Between Icons

Lines are used within a Program Net to indicate the flow of signals between icons. (See Figure 7... -1)

Signal Types

Three types of signals can be communicated between icons in a Program Net. The three signal types are differentiated by color and their different functions follow:

DATA (RED)

Numerical values are transmitted from icon to icon via Data type signal connections. HyperNet allows Data connections to be made ONLY between Data terminals on icons to preclude mismatching of signal types.

SIGNALS COMMUNICATED VIA DATA TYPE CONNECTIONS	EXAMPLE ICONS WITH DATA TYPE TERMINALS
TEMPERATURE	OUTPUTS FROM THERMOCOUPLE ICONS
VOLTAGE	OUTPUTS FROM VOLTAGE ICONS
PRESSURE	MATH (INPUTS AND OUTPUTS)
FLOW	INPUTS TO COMPARATORS
	OUTPUTS FROM COUNTERS

Table 7... -2: Example Data type signals and Icons

LOGIC (GREEN)

True / False values are transmitted from icon to icon via Logic type signal connections. Logic signals have only two states. As with the other signal types, HyperNet only allows Logic signals to be connected to Logic type terminals on icons.

SIGNALS COMMUNICATED VIA LOGIC TYPE CONNECTIONS	EXAMPLE ICONS WITH LOGIC TYPE TERMINALS
TRUE OR FALSE CONDITIONS	INPUTS TO RELAY ALARM ICONS
	OUTPUTS FROM COMPARATORS
	ENABLE INPUTS TO SAMPLE RATE CLOCKS
	INPUT TO MESSAGE ICON
	OUTPUT FROM START/STOP CLOCK
	INPUTS TO COUNTERS

Table 7... -3: Example Logic type signals and icons

UPDATE (BLUE)

Update commands are a special type of signal that is generated ONLY by Sample Rate Clock icons. This signal is connected to the Update input on icons and commands them to update their outputs. As with the other signal types, HyperNet only allows Update signals to be connected to Update type terminals on icons.

SIGNALS COMMUNICATED VIA UPDATE TYPE CONNECTIONS	EXAMPLE ICONS WITH UPDATE TYPE TERMINALS
THE UPDATE COMMAND GENERTED BY THE SAMPLE RATE CLOCK ICON	INPUT ICONS HAVE AN UPDATE TERMINAL

Table 7... -4: Example Update signals and icons

Adding Signal Connections Between Icons

Connections between two icons are drawn by locating the mouse cursor over the first icon's terminal (note how the cursor changes when properly located over a terminal), clicking and holding the button down, then dragging a connection line to the second icon's terminal and releasing the button.

The direction that signal connections can be dragged on the workspace is limited to minimize potential problems with feedback and/or race conditions. Connections can only be drawn from the left to the right on the workspace.

During the connection process, HyperNet checks and disallows recognized illegal connections such as:

- Differing terminal types cannot be interconnected. For example, a Data terminal cannot be connected to a Logic terminal.
- Output terminals can only connect to Input or Enable terminals.
- Connections cannot be made between terminals on the same icon
- Multiple connections to a single Input terminal.

During the construction of Program Nets, it is common for multiple connection lines to originate at an Output terminal, however most icons can only have one or two inputs. Some icons such as the Scroll Tracking and Destination File icons (used in the Post-Processing and HyperTrack Windows) allow for more than two inputs and will automatically add input terminals as connection lines are added.

Saving the Net

After development of the Program Net, the program can be saved to a file by clicking on the Save to Disk icon in the Toolbar.

NET PERFORMANCE

After construction of a Program Net a quick review should be done to insure that the Net's performance in the HyperLogger will meet the User's goal. This check should include proper implementation and for higher speed applications, a processing (execution) speed review.

Program Net Checklist

- Has the Global Icon been configured? Has a Program Name and Description been assigned?
- Have connections been added between icons?
- Have desired measurement units (eg C, F, ohms, etc) been selected?
- Does each icon have a name and output name? Names are not necessary for the icons to function, however, if they are used consistently during the construction of a Program Net, the presence of a User defined name (vs the default name) becomes an indicator that the icon has been configured. Additionally, the added annotation makes comprehension of the net by other Users and/or at a later date even easier.
- Do the Memory icons have names assigned? If omitted, it may be more difficult to identify data during plotting and further post-processing.

Program Net Execution Speed

> *NOTE: The information presented in this section is provided for Users that are attempting to collect data at faster rates (such as Samples per Second).*
>
> *Users utilizing the HyperLogger for data collection at slower rates (eg Samples per Minute) may opt to skip this section and refer to it later when faster rates are required.*

Due to the nearly unlimited flexibility and potential variations of Program Net designs, it is difficult to specify the actual speed performance of Program Nets. In an attempt to relate to the HyperLogger User a feel for the Program Net processing speeds to be expected, the following guidelines are presented.

Additionally, a number of example Program Nets are provided in Appendix B with their approximate execution speeds. From this information, a feel for the speed of execution of most Program Nets can be developed.

Program Net Performance Guidelines

The time required for a Program Net to execute within the HyperLogger is a function of a number of variables including:

- **Total number of icons in the Program Net** - with a greater number of icons to process each time the Program Net loops, more time is required.
- **Type of icons** - different types of icons require various amounts of time to process. For example, a Thermocouple input icon requires a considerable amount of processing time to perform the analog to digital conversions, CJC measurements, and the associated math. On the other end of the speed spectrum, a Data Memory icon merely stores data into a

memory location... an operation which can be performed very quickly.

- **Program Net design -** Program Nets can be developed that have varying execution times that are a function of inputs or values within the Net. A simple example of this is a Program Net that scans a single digital input (eg switch state) and only enables the logging of 10 thermocouple input channels when the switch is closed. In this example, the digital input could be scanned very fast when open, but when it closes, extra processing time is required for the 10 thermocouple channels.

An Empirical Answer...

After optimization of a Program Net with consideration of the above guidelines, the best way to determine the speed performance of a Net is to upload it to a HyperLogger and run it. If maximum speed is the goal, the driving Sample Rate Clock rate can be set to a minimum (eg 1mS) and the program can be tested. Review of the data and actual sample times will result in a very accurate execution rate value.

Following are three benchmark Program Net execution times that can be used to gain a relative feel for the sampling rates achievable with the HyperLogger. The times were empirically determined with a single channel (as specified) storing directly to memory. In each case, the Program Net consists of one Sample Rate Clock icon, one input channel icon, and one memory icon. No filtering was enabled.

Input Icon Type	Samples per Second (one channel)
VDC-LO	150
Thermocouple Type J	30
Counter (GPDI)	320

Figure 7... -13: Approximate throughput rates for various types of signals / Program Nets

Nets that utilize conditional logging strategies will have varying execution rates. A feel for these Net execution rates can be developed by constructing and running Program Nets that represent each conditonal branch of the Net, then summing the execution times for branches that can could execute simultaneously.

Miscellaneous Program Net Performance Considerations

MISSED SAMPLES

Although it is possible to set Sample Rate Clock rates faster than a Program Net can actually execute, the Program Nets will attempt to run. In these conditions, if a Sample Rate Clock sends the Update command before the Net has been fully

executed, the HyperLogger will finish the Net execution then immediately start processing the Net again.

The fact that an Update command has been processed late is accessible to the User via the Warning icon. One of the options within the Warning icon is to provide a logic output if a *Sample is Missed*. Details on the Warning icon are provided in the Master Icon Reference in Appendix A.

It must be noted that in this situation, although the actual data has not been sampled at the programmed rate, *all collected data will include the actual date/time that the Net executed.*

> ***TIP:*** *To achieve maximum speed of a Program Net, the Sample Rate Clock(s) used in the Net can be set to a rate faster than the Net can possibly process.*

OTHER DETRIMENTS TO SPEED

Any additional processing that is performed during execution of a Program Net such as serial communication (eg HyperTrack operation, Status Queries, etc) and/or display of readings through the HyperLogger front panel LCD will have detrimental effects on the rate at which a Program Net can execute. For optimum speed,wthe LCD should not be displaying updating information (eg Probe Points, battery voltage, etc).

PROGRAM NET DOCUMENTATION

Two features are provided within the HyperNet Window to assist in documenting and the ensuing field wiring of the HyperLogger.

HyperNet Printout

The actual Program Net display can be printed by selecting *Print Net* from the *File* menu.

> ***NOTE:*** *The PC must be set to 256 color mode in order to properly print the Net.*

TSA Wiring Printout

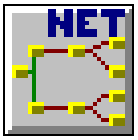
A Terminal Strip Adapter (TSA) I/O Listing can be generated and printed for the Program Net. This listing can then be used during the field wiring of the various I/O signals to the TSA.

Upon completion of the Program Net, select *Save TSA I/O List* from the *File* menu, enter a filename, and the file will be generated and saved as an ASCII text file. A text editor will automatically open the file. The listing can then be printed from within the editor and carried with the HyperLogger to the installation site.

The TSA Listing includes the connections for each of the I/O channels that is used in the Program Net as well as standard connections for External Power, the Digital Port and the CJC connector. Names assigned to Input and Output icons in the Program Net are used for channel identification.

At the bottom of the TSA I/O Listing are various notes relative to other connections such as modem. These notes are imported from a file called NOTES.TXT which is supplied in HyperWare. Reminders and special

installation comments can be added by the User to the NOTES.TXT file by editing the file with any text editor (such as Notepad). After editing, the revisions will appear on the TSA Listing each time a listing is generated from within HyperNet.



PROGRAM NET UPLOAD TO THE HYPERLOGGER

After the Program Net has been checked and saved, clicking on the HyperComm Window button will return HyperWare to the HyperComm Window. The Program Net icon displayed on the PC graphic represents the last edited Program Net. Drag and drop the Program Net icon from the PC to the serially connected HyperLogger or to the PCMCIA card drive to upload the new Program Net.

Refer to Chapter 5, HyperComm Serial Communications for details of the Program Net serial link uploading procedure or to Chapter 6 for details on using the PCMCIA card.

NOTES: