invensys
Foxboro

LIN

Programmer Wizard

User Guide

HA029822

October 2011 (Issue 2)

Contents

1 Introduction

The purpose of this document is to allow Users to quickly access information concerning a particular point of interest in the Programmer Wizard.

The purpose of the Programmer Wizard is to simplify the generation of all required LIN Function Blocks in the instrument database. On completion of this wizard, one PROGCTRL block will be generated, and an appropriate number of PROGCHAN blocks and SEGMENT blocks are automatically placed in the database and used to operate the Setpoint Program.

The PROGCTRL block controls the execution of the Setpoint Program, while the PROGCHAN block contains the data and options for the profiled channel. Each SEGMENT block is used to identify up to 4 changes to the Setpoint.

What is a Setpoint?

Setpoint is the target value that an automatic control system, e.g. PID controller, will aim to reach. For example, a boiler control system might have a temperature setpoint that is a temperature the control system aims to meet.

2 PROG_WIZ

PROGCTRL Name

This wizard simplifies the generation of a Program Template file (.uyw file) that is used by the Programmer Editor and the instrument.

It is used to define or edit a template file containing information about the limits and properties of a Program. When complete it automatically creates or updates the Program Template file and a PROG_WIZ compound in this database (.dbf). The PROG_WIZ compound contains 1 PROGCTRL block used to control the overall execution of the Setpoint Program, up to 8 PROGCHAN blocks one for each profiled channel, and up to 8 SEGMENT blocks providing a maximum of 32 program segments (each SEGMENT block offering 4 program segments).

Tip!
The Program Template file generated using this wizard can be referenced by another instrument (see Use Existing Program Template File and Node: on the next page), that supports the PROGCTRL block, PROGCHAN block and the SEGMENT block.

Use existing PROGCTRL block

Select this radio button to enable the list of PROGCTRL blocks that already exist in the database.

Picklist

Select a specific PROGCTRL block already associated with the Program Template file of the instrument from the enabled list of PROGCTRL blocks that already exist in the database.

Make new PROGCTRL block

Select this radio button to enable the text field used to name a new PROGCTRL block.

New PROGCTRL Block name

Enter a 4 character name for the new PROGCTRL block in a new PROG_WIZ compound (of the same name) that will be associated with a particular Program Template file.

- This is a prefix that is applied to the remaining blocks that will be automatically created.

Note
This is restricted to four characters to allow automatic naming of related PROGCHAN blocks and SEGMENTS blocks.

Help (☑)

You can use this checkbox to show or hide information in this dynamic window.

What is the PROGCTRL block ?

The PROGCTRL block provides control for the overall execution of a single Program configured using the Programmer Editor, i.e. one PROGCTRL block per Program.

Program Template File

The Program Template file (.uyw file) is used to define the limits of a Program in the instrument and describes how each Profiled Channel see Name on Profiled Channels page, is presented in the Programmer Editor. It contains the information generated from the Programmer Wizard. It must only be edited using the wizard and defines the constraints of a Program file for an instrument. The names of channels, Digital Event Outputs, User Values are shown in the Programmer Editor on the Channel page in the Program pane.

This page shows the Program Template file associated with this database (.dbf) and automatically added to the List of files to be Downloaded.

Use existing Program Template File

Select this radio button to edit the Program Template file shown in the **File** field.

File

Enter a name used to identify the required Program Template file.

Note
This changes dynamically as the Program Template file radio buttons are selected.

Browse...

Use this to launch a standard Browse dialog.

This can be used to define a unique path to an Instrument Folder containing the Program Template file.

Note
For correct operation the Program Template file and Program file must reside in the same instrument folder.

Beware
Be careful when editing an existing Program Template file.
The Program Template file, as shown in the File field, could be associated with other instruments. Any changes to this file will affect the Program of all instruments associated with it.

Make new Program Template file

Select this radio button to enable the **File** field allowing a name to be entered.

Enter a name used to identify the required Program Template file in the **File** field.

File

Enter a name used to identify the required Program Template file.

Copy existing Program Template file

Select this radio button to enable you to create a copy of the Program Template file shown in the **ToFile** field at the location specified in the **Node:**.

This reveals the **To File** field that allows you to enter a name for the copy of the file shown in the **File** field.

Note
Only one Program Template file can be used at a time.

Copy File

Enter a name used to identify the required Program Template file.

Browse...

Use this to launch a standard Browse dialog.

This can be used to define a unique path to an Instrument Folder containing the Program Template file.

Note
For correct operation the Program Template file and Program file must reside in the same instrument folder.

To File

Enter a name used to identify the copy of the .uyw file shown in the **Copy File** field.

Note
Only enabled if Copy Existing Program Template File is selected.

Node:

This shows the Node Address and the instrument type in brackets of the instrument retaining the Program Template file.

The information is shown in **Filepath** of the PROGCTRL block in the PROG_WIZ compound of the database or the node address of the source instrument when the Program Template file has been copied.

Note
LIN Instruments configured to operate as a Redundant Pair (Duplex) employ consecutive odd and even Hexadecimal LIN Node Address values. An even LIN Node Address will always be allocated to the primary device.

IMPORTANT NOTE
Both Program Template file and Program file must be stored at the location, defined in PROGCTRL.File.Filepath. When this parameter is not configured the Program Template file and Program file are saved on the instrument that contains the PROGCTRL block.

What is the Program Template file ?

The Program Template file contains information about the limits and the properties of the Program used by the instrument, i.e. the number of channels (profiles), the maximum number of segments allowed in the Program, events and conditions.

The Program Template file is automatically added to the list of files to be downloaded

when the database is saved.Program Segments

Program Segments divide a program into specific operations, e.g. STEP the Setpoint changes instantaneously from its current Setpoint value to a new value at the beginning of the next segment, DWELL the Setpoint remains constant for a specified period at the specified Target Setpoint, RAMPTIME the Setpoint starts a controlled change from the current Setpoint to a Target Setpoint for the duration of the segment, RAMPRATE the Setpoint starts a controlled change from the current Setpoint to a Target Setpoint determined by the rate specified, or END this segment is read-only, and used to denote the end of the Program. Each segment corresponding to a single page in a SEGMENT block.

Use the picklist to select the total number of segments allowed in the profiled channel(s).

Picklist

This picklist shows how many segments are allowed in the Program.

A maximum of 32 segments may be configured.

Profiled Channels

This page provides parameters that configure the constraints of the defined profiled channel(s). Each profile channel is the equivalent of one PROGCHAN block in the database.

Name

Enter a name used to identify the process variable profiled by this channel.

Low Limit

Enter the low limit value for the Target Setpoint.

High Limit

Enter the high limit value for the Target Setpoint.

SP Decimal Places

Select the number of decimal places for the setpoint value.

- SCI indicates that scientific notation will be used, e.g. 1.341e+10.
- 0, 1, 2, 3 and 4 indicate the number of decimal places used.

This column corresponds to *Config.DP_PV* in the PROGCHAN block.

Holdback Decimal Places

Select the number of decimal places for the Holdback Value.

- SCI indicates that scientific notation will be used, e.g. 1.341e+10.
- 0, 1, 2, 3 and 4 indicate the number of decimal places used.

This column corresponds to *Config.DP_Hback* in the PROGCHAN block.

Units

Enter text used to indicate the units for the process variable, i.e. Deg.C, Deg.F, K or R.

These fields are actually free format text fields and will accept any text.

Rate Decimal Places

Select the number of decimal places for a RampRate segment.

- SCI indicates that scientific notation will be used, e.g. 1.341e+10.
- 0, 1, 2, 3 and 4 indicate the number of decimal places used.

This picklist corresponds to *Config.DP_Rate* in the PROGCHAN block.

Note
This is only available when configuring a single profiled channel.

What is a Profiled Channel ?

A Profiled Channel is the pattern of control for a single process variable value derived from the plant/system.

What is the PROGCHAN block ?

The PROGCHAN block is used to configure the data and options of one channel being profiled by the Program, i.e. one PROGCHAN block per Profiled Channel. Each PROGCHAN block is controlled by a single PROGCTRL block and can be associated with up to 8 SEGMENT blocks providing a maximum of 32 segments per channel, or less segments but more digital events and user values.

What is Holdback ?

Holdback provides a method to allow a measured PV (process variable) to 'catch up' with its' SP (setpoint), if it has deviated from it by more than a specified amount (*PROGCHAN.Monitor.HbVal*).

User Values

This page provides the configuration parameters for User Values.

The total number of User Values is not limited by the number of channels entered. Additional PROGCHAN blocks will be created if more User Values are declared than the number of given channels can support. However, these additional blocks will not be profiled as only the User Values are being used.

Name

Use this to enter a name identifying an individual analogue value for the segment.

These parameters correspond to the *Monitor.UVal1*, *Monitor.UVal2*, *Monitor.UVal3*, and *Monitor.UVal4* fields in the PROGCHAN block, and the *Segment n.UVal1_n*, *Segment n.UVal2_n*, *Segment n.UVal3_n*, and *Segment n.UVal4_n* in the SEGMENT block.

Low Limit

Enter the low limit for the User Value.

High Limit

Enter the high limit for the User Value.

Decimal Places

Select the number of decimal places for the User Value.

- SCI indicates that scientific notation will be used, e.g. 1.341e+10.
- 0, 1, 2, 3 and 4 indicate the number of decimal places used.

These parameters correspond to *Config.DP_UV1* to *Config.DP_UV4* in the PROGCHAN block.

Units

Enter text used to indicate the units for the User Value, i.e. Deg.C, Deg.F, K or R.

These fields are actually free format text fields and will accept any text.

What is a User Value ?

User values are general purpose analogue values that can be configured in any Step, Dwell, RampRate, or RampTime segment.

PROGCHAN.Monitor.UVal1 to *PROGCHAN.Monitor.UVal4* should be wired to an output block to allow the value to be written to the connected field as the Segment is started.

Digital Events

This page provides parameters that configure the text string used to indicate the operation of the corresponding Digital Event.

The total number of Digital Events is not limited by the number of channels entered. Additional PROGCHAN blocks will be created if more Digital Events are declared than the number of given channels can support. However, these additional blocks will not be profiled as only the Digital Events are being used.

Name

Use this to enter a name identifying an individual digital event in the Programmer Editor.

The parameters in this column correspond to the *Names.EvName1* to *Names.EvName16* fields in the PROGCHAN block.

The *Segment n.Events_n* bitfields in each SEGMENT block display the digital events currently being executed in this segment of the Program.

Off Text

The parameters in this column are used to configure a text string associated with each Digital Event Off value.

On Text

The parameters in this column are used to configure a text string associated with each Digital Event On value.

What is a Digital Event ?

Digital Events are general purpose on/off values that can be configured in any Step, Dwell, RampRate, or RampTime segment.

PROGCHAN.Monitor.EventOut bitfields should be wired to an output block to enable/disable a connected digital field as the Segment is started.

Conditions

This page provides parameters that configure the Wait and Exit conditions for the Program.

These conditions can be configured to

- prevent the transition to the next segment until a specific configuration is met (Wait)
- terminate the segment when a defined state has occurred (Exit)

The total number of Wait and Exit Conditions are not limited by the number of channels entered. If more Wait and Exit Conditions are declared than the number of given channels can provide, additional PROGCHAN blocks will be created. However, these additional blocks will not be profiled as only the conditions are being used.

Wait Conditions

Condition Name

Use this to enter a name identifying an individual Wait Condition.

The parameters in this column correspond to the *Names2.WaName1* to *Names2.WaName8*fields in the PROGCHAN block.

Off Text

The parameters in this column are used to configure a text string associated with each Wait Condition Off value.

On Text

The parameters in this column are used to configure a text string associated with each Wait Condition On value.

Exit Conditions

Condition Name

Use this to enter a name identifying an individual Exit Condition.

The parameters in this column correspond to the *Names2.ExName1* to *Names2.ExName8* fields in the PROGCHAN block.

Off Text

The parameters in this column are used to configure a text string associated with each Exit Condition Off value.

On Text

The parameters in this column are used to configure a text string associated with each Exit Condition On value.

What is a Wait Condition ?

Wait Conditions are used to prevent the Program proceeding to the next segment until a specific configuration is met.

What is a Exit Condition ?

Exit Conditions are used to terminate non-ramping segments when a defined state has occurred.

Power Fail Recovery

This page provides parameters that configure the action that will be taken if a power failure occurs and the instrument hot-starts.

Action on Power Fail Recovery

Select the action that will be used if a power failure occurs and the instrument hot-starts.

- RampBack

Select RampBack so that when power is restored the Program will ramp to the Target Setpoint. If the Segment is configured as RampRate or RampTime, the PV will Ramp to the Target Setpoint, if configured as Step the PV will Step to the Target Setpoint, and if configured as Dwell, the Dwell time will not resume until the Target Setpoint is obtained at the last rate used in the Program.

- Abort

Select Abort so that when power is restored the Program will abort and enter the Idle state, if *Monitor.CmndHshk.Idle* is TRUE.

- Continue

Select Continue so that when power is restored the Program continues from where it was interrupted when power was lost. All parameters, such as the setpoint and time remaining in the active segment will be restored to their power-down values.

Note
This Power Fail Recovery configuration is recommended for applications that need to bring the measured PV to the setpoint as soon as possible.

- TestTime

Select TestTime so that when power is restored before the period configured in *Monitor.TestT1* the Program will continue. However, if power is restored after the period configured in *Monitor.TestT1* but before *Monitor.TestT2* the Program will RampBack using the rules specified, see *RampBack*. If power is not restored until after the period configured in *Monitor.TestT2* the Program will abort and enter the Idle state, if *Monitor.CmndHshk.Idle* is TRUE.

- Hold

Select Hold so that when power is restored the Program will hold and enter the Held state, if *Monitor.CmndHshk.Held* shows TRUE.

Note
The TestT1 and TestT2 parameters are only available when TestTime is selected in the Action on Power Fail Recovery picklist.

TestT1

Use this to enter the time limit, 00:00:01 to 24:59:59, allowing the Program to apply a Continue strategy.

If power is restored before this time limit the Program will continue from where it was interrupted when power was lost, see Continue in **Action on Power Fail Recovery**. However, if power is restored after this period but before the TestT2 time limit, the Program will RampBack using the rules specified, see RampBack in **Action on Power Fail Recovery**.

Using 00:00:00 will not allow the Program to apply the Continue strategy, see TestT2.

This parameter corresponds to the *Monitor.TestT1* field in the PROGCTRL block.

TestT2

Use this to enter the time limit, 00:00:01 to 24:59:59, allowing the Program to apply a RampBack strategy.

If power is restored before this time limit the Program to apply the RampBack strategy after a power failure.

Using 00:00:00 will not allow the Program to apply the RampBack strategy causing the Program to abort and enter the Idle state, if *Monitor.CmndHshk.Idle* shows TRUE.

This parameter corresponds to the *Monitor.TestT2* field in the PROGCTRL block.

What is Hot-Start ?

A hot-start is an attempt to restart the instrument using values that were operating in the strategy before the power failure occurred.

Some instruments can be configured to hot-start/cold-start allowing the instrument to attempt a hot start. If the hot start fails, the instrument attempts to carry out a cold start. If the cold start fails the database will be cleared and the instrument will enter an ‘Idle’ state and remain there until physically restarted.

Summary

This page provides a summary of the blocks required for the configuration and includes a description of the Program Template file.

Note
Existing Program Template files can be edited at any time by simply launching the Programmer Wizard.

Summary of requested Values

This shows a summary of the features configured in this wizard.

The values that are applied to each corresponding parameter are derived from the database and Program file.

Summary of Function Blocks to be written to the database

This shows the total number of blocks that have been added to the database to support the maximum limits of the Program Template file.

There will always be one PROGCTRL block, but the total number of remaining blocks created using this wizard is dependant on the configured features.

Program Template File Information

This shows a summary of the Program Template file information configured on the Program Template File page.

3 Appendix A

What is LIN?

Local Instrument Network or LIN is a communications protocol providing a token-passing masterless Network which allows peer-to-peer communications and file transfer between instruments. It is supported via Ethernet, Arcnet and Serial communications connections.

A LIN Database groups data into blocks of related data. For example, a function block can represent an input, an output, a controller, and so on. The LIN configuration tool (LINtools) and display packages (i.e. User Screen Editor) recognise different types of function block, and handle them appropriately.

Communications between LIN and Modbus instruments require a Gateway. This is an interface between the LIN communications protocol and the Serial communications protocol, generally provided by blocks in the database. It involves the mapping of data from the LIN Database to the Modbus registers and digitals.

What is a LIN Database ?

A LIN Database (.dbf) is a software program that runs in an instrument on the LIN. The running LIN Database takes in signals from sensors in an outside entity (e.g. an industrial plant), processes them in specified ways, and then outputs signals to actuators in the entity to control its behaviour in the required manner.

The cycle of signal input to the LIN Database, signal processing, and signal output to the entity is repeated continuously while the Strategy runs.

More than one LIN instrument can be involved in controlling a single entity, but only one LIN Database can run in a single LIN instrument at a time.

A LIN Database can work in conjunction with one or more LIN Sequences running in the LIN instrument. It can also make use of LIN Actions stored in action files in the LIN instrument.

In LINtools Engineering Studio, a LIN Database is represented and configured graphically as an arrangement of LIN function blocks interconnected by wires.

Note
The Strategy of an I/O Instrument is configured using the iTools Engineering Studio.

What is a LIN Function Block ?

A LIN Database (.dbf) uses a block-structured approach to configuring a Strategy, where a variety of ready-made function blocks perform the processing required.

A function block is an instance of a reusable module of program code. All function blocks exist in specific categories in the template library. Each is dedicated to a particular type of processing operation, e.g. the ADD2 template adds two numbers. In general, function blocks take in analogue and/or digital signals via their inputs, process them in a variety of ways and then pass the results on via their outputs. By ‘wiring’ the function blocks together, the signals can flow between them to execute the control strategy.

4 Index

International sales and service

Invensys Operations Management
5604 Granite Parkway, Suit 1000
Plano
Texas 75024
469-365-6400

Invensys Canada – Ontario
Invensys Operations Management
5050 South Service Road
Suite 200
Burlington ON L7L 5Y7
Canada
1-905-333-1311

Invensys Canada – Montreal
Invensys Operations Management
4 Lake Road
Dollard-des-Ormeaux
Quebec H9B 3H9
Canada
1-514-421-4210
Tel: +9714 8074700
Fax: +9714 807477

Latin America
Invensys Systems Argentina
Nuñez 4334
(1430) Buenos Aires
Argentina
Tel: +54-11-6345-2100

Middle East
Jebel Ali Free Zone
P.O. Box 61495
Dubai
United Arab Emirates
Tel: +9714 8074700
Fax: +9714 807477

Asia Pacific
IPS (S) Pte Ltd
15 Changi Business Park Central
Singapore 486057
Tel: +65 6829 8888
Fax: +65 6829 8401

Europe and Africa
Invensys Systems France
S.A.10
Avenue du Centaure B.P. 8255 Cergy
95801 Cergy Pontoise Cedex
France
Tel: +33 1 34 43 25 25
Fax: +33 1 34 43 25 00

United Kingdom – Worthing
Invensys Eurotherm Ltd
Tel: +44 1903 268500
Fax: +44 1903 265982
Email: info.uk@eurotherm.com
www.eurotherm.co.uk

invensys
Foxboro

HA029822 (CN27764)