AN-303

Configuring Protege Input EOL Resistors Using Commands

Application Note

Last Published: 01-Jul-24 3:45 PM

Contents

Introduction

Protege GX and Protege WX inputs can be connected to an array of EOL monitored or dry contact devices such as magnetic switches and PIR motion detectors. Each input may be individually configured for normally open and normally closed configurations, with or without EOL resistors for tamper and short condition monitoring.

When using an input with EOL resistor configuration, the controller generates an alarm condition when the state of the input changes between open and closed, and generates a tamper alarm condition when a wire fault (short circuit) or a cut wire (tamper) occurs in the line .

While the most common End of Line (EOL) resistor configurations for inputs can be selected from the available options within Protege GX and Protege WX, custom resistor configurations can also be easily achieved through the use of specific commands, providing sites with unlimited configuration possibilities.

3 resistor configurations also enable you to use input doubling where each input can be treated as two separate inputs with shared tamper and short conditions. This can be used to implement antimasking functionality.

Prerequisites

The controller and all expander modules using inputs with custom EOL must meet the following minimum requirements.

Module	Firmware Version	Notes
Protege GX Controller	2.08.1460 or higher	
Protege WX Controller	4.00.1998 or higher	
PRT-ZX16-DIN PRT-ZX8-DIN PRT-HZX16-DIN	1.12.044 or higher	Protege WX does not allow more than 16 inputs to be programmed for each module. Therefore it is currently not possible to double the inputs of 16 input expanders in Protege WX. This is a known issue.
PRT-ZX1	1.0.003 or higher	
PRT-IO84-DIN	1.02.015 or higher	
PRT-RDM2-DIN-485 PRT-HRDM-DIN	1.12.602 or higher	
PRT-KLCS	1.09.045 or higher	

Configuring End of Line Resistors

Configuration is achieved by adding commands to the programming of each input (**Programming | Inputs**).

1. Enable configurable EOL resistor settings on the input.
2. Define the number of resistors that will be configured.
3. Define the resistance value of each resistor.
4. Define the input state that will be reported in the software for each resistor state.
5. Define the percentage of hysteresis.

The **resistor states** (see page 7) are the states that the configured EOL will transition through, in sequence, as the resistance on the input is changed from having no resistance (shorted to ground) to most resistance (open circuit).

The **input state** (see page 7) is what will be reported in the software for each of those resistor states. Resistor states can be mapped to whichever available input state you would like to have reported in the software.

In summary, the resistor states are the states the configured EOL will move through as the resistance on the input is changed, and the input state is what will be reported in the software for each of those resistor states.

End of Line Resistor Commands

The following table describes the EOL resistor programming commands and their functions:

Command	Reference
`EOL = Res`	Enables configurable EOL resistor settings on the input. If this command is absent or incorrect the input will not recognize the EOL resistor configuration commands.
`NumberOfEOLResistors = <value>`	Defines the number of resistors that will be configured for the input. Valid values are 1, 2 and 3.
`ConfigEOLRes1 = <value>`	Defines the resistance value of the first EOL resistor in ohms.
`ConfigEOLRes2 = <value>`	Defines the resistance value of the second EOL resistor in ohms.
`ConfigEOLRes3 = <value>`	Defines the resistance value of the third EOL resistor in ohms.
`ConfigEOLResState1 = <value>`	Defines the input state that will be reported when the first resistor state is reached.
`ConfigEOLResState2 = <value>`	Defines the input state that will be reported when the second resistor state is reached.
`ConfigEOLResState3 = <value>`	Defines the input state that will be reported when the third resistor state is reached.
`ConfigEOLResState4 = <value>`	Defines the input state that will be reported when the fourth resistor state is reached.
`ConfigEOLResState5 = <value>`	Defines the input state that will be reported when the fifth resistor state is reached.
`ConfigEOLResState6 = <value>`	Defines the input state that will be reported when the sixth resistor state is reached.
`ConfigEOLHysteresis = <value>`	Defines the percentage of hysteresis to be applied. See Hysteresis Calculation (see next page).

Programming Rules

- Resistor values are specified in ohms (Ω). For example, a 1K resistor will have a value of 1000.
- With a 2 resistor configuration, a total of 4 distinct resistor states will be created from the 3 unique thresholds that result. For more information, see Programming 2 Resistor EOL (next page).
- With a 3 resistor configuration, a total of 6 distinct resistor states will be created from the 5 unique thresholds that result. These states are used to support input doubling, with a high input and low input on the same physical input terminal. For more information, see Input Doubling with 3 Resistor EOL (page 9).
- The number of `ConfigEOLRes` and `ConfigEOLResState` commands required in the programming is dependent on the number of resistors and must be entered according to the configuration of the input.
- When configuring inputs on the controller, you must apply the commands to the inputs with a **Module type** of Controller (CP), not the inputs assigned to the onboard reader expander.
- After configuration commands are added or edited, a module update is necessary to send programming to the required devices. Save the input programming, then navigate to the expander programming. Right click on the relevant expander record and click **Update module**.
- The EOL resistor commands override the **Input end of line (EOL)** and **Contact type** settings in the **Options** tab of the input programming.

Hysteresis Calculation

You can use the `ConfigEOLHysteresis` command to introduce hysteresis to the input state calculations. This should be used for fine-tuning the operation if the input is 'chattering', i.e. rapidly changing states. Chattering can occur when there is noise in the circuit, which may depend on the type of cable and other components used, the length of the cable run and environmental factors such as vibration.

Hysteresis provides some tolerance for resistance changes caused by noise, preventing the controller from reporting state changes unnecessarily. It is recommended that you use the minimum required hysteresis value to mitigate any issues that you observe (e.g. 0 - 2.5%).

Do not use a hysteresis value that is larger than any of the individual resistors in the circuit. For example, consider a circuit with 10K and 1K resistors and 10% hysteresis. Because the hysteresis value is so high, the controller cannot distinguish between the 10K state (10K resistor only) and the 11K state (10K + 1K), so the input state will not change when the resistance increases to 11K.

The hysteresis calculation for inputs is: **Hysteresis (%) = ConfigEOLHysteresis / 100**

The following table demonstrates examples of controller input hysteresis values.

ConfigEOLHysteresis Value	Hysteresis
0	0%
100	1%
250	2.5%
500	5%
750	7.5%
1000	10%

Programming 2 Resistor EOL

This section covers the input states and resistors for programming custom 2 resistor EOL.

Input States

The following table shows the available input states that will be reported in Protege GX and Protege WX. The input state value is entered as the **ConfigEOLResState** command **<value>** for the corresponding resistor state.

Input State Value	Reported Input State
0	Closed
1	Opened
2	Short Circuit
3	Tamper

Resistor States

The following table describes the resistor states.

Resistor State Command	Description
ConfigEOLResState1	State 1 is the initial input resistance state - 0 Ω
ConfigEOLResState2	State 2 is the input resistance state that meets the value of the first resistor - R1 Ω
ConfigEOLResState3	State 3 is the input resistance state that meets the value of the sum of the first and second resistors - R1 + R2 Ω
ConfigEOLResState4	State 4 is the final input resistance state - infinity Ω (open circuit). It is always the last state programmed.

If programming a single resistor, omit state 4 and use state 3 to represent open circuit.

2 Resistor EOL Examples

Examples of the commands required to program a 2 resistor EOL configuration are illustrated below.

Normally Closed Configuration

Below is a typical example of programming for a Normally Closed 2 resistor EOL configuration, where each resistor is 1K (a value of 1000).

For a Normally Closed contact type the **Closed** state (0) comes before the **Opened** state (1).

1. Navigate to **Programming | Inputs** and select or add the inputs to be programmed.
2. Scroll down to the **Commands** section and enter the following commands:
   - `EOL = Res`
   - `NumberOfEOLResistors = 2`
   - `ConfigEOLRes1 = 1000`
   - `ConfigEOLRes2 = 1000`
   - `ConfigEOLResState1 = 2`

- `ConfigEOLResState2 = 0`
- `ConfigEOLResState3 = 1`
- `ConfigEOLResState4 = 3`

3. Enter a hysteresis command, such as:
   - `ConfigEOLHysteresis = 250`
4. Click **Save**.
5. If you are configuring expander module inputs, navigate to the relevant expander programming (e.g. **Expanders | Input expanders**). Right click on each expander record that supports the inputs modified above, and click **Update module**.

Normally Open Configuration

Below is a typical example of programming for a Normally Open 2 resistor EOL configuration, where each resistor is 1K (a value of 1000).

For a Normally Open contact type the **Opened** state (1) comes before the **Closed** state (0).

1. Navigate to **Programming | Inputs** and select or add the inputs to be programmed.
2. Scroll down to the **Commands** section and enter the following commands:
   - `EOL = Res`
   - `NumberOfEOLResistors = 2`
   - `ConfigEOLRes1 = 1000`
   - `ConfigEOLRes2 = 1000`
   - `ConfigEOLResState1 = 2`
   - `ConfigEOLResState2 = 1`
   - `ConfigEOLResState3 = 0`
   - `ConfigEOLResState4 = 3`
3. Enter a hysteresis command, such as:
   - `ConfigEOLHysteresis = 250`
4. Click **Save**.
5. If you are configuring expander module inputs, navigate to the relevant expander programming (e.g. **Expanders | Input expanders**). Right click on each expander record that supports the inputs modified above, and click **Update module**.

Input Doubling with 3 Resistor EOL

Input doubling is a method of using different resistor values to allow an input to be treated as two separate inputs. This is commonly used to implement antimasking functionality.

Input Doubling Programming Rules

The following additional rules apply specifically to input doubling programming.

- Input doubling is supported for 3 resistor EOL configurations only.
- For each configured input you will need to program an additional input record as the **high input** to represent the second input monitored by the same physical connection.
- The **Module input** of the high input must be at an offset of the maximum number of physical inputs of the module. For example, for PRT-RDM2-DIN-485 modules, the physical input 1 will have a high input at address 9.
- EOL programming commands are not required for the high input. These apply to the physical input only.

Input States

The following table shows the available input states that will be reported in Protege GX. The input state value is entered as the `ConfigEOLResState` command `<value>` for the corresponding resistor state.

Input State Value	Reported Input State
`0`	Both Inputs Closed
`1`	Low Input Opened. High Input Closed
`2`	Short Circuit
`3`	Both Inputs Opened
`5`	High Input Opened. Low Input Closed
`6`	Tamper

Resistor States

The following table describes the resistor states for input doubling with a 3 resistor EOL.

Resistor State Command	Description
`ConfigEOLResState1`	State 1 is the initial input resistance state - 0 Ω
`ConfigEOLResState2`	State 2 is the input resistance state that meets the value of the first resistor - R1 Ω
`ConfigEOLResState3`	State 3 is the input resistance state that meets the value of the sum of the first and second resistors - R1 + R2 Ω
`ConfigEOLResState4`	State 4 is the input resistance state that meets the value of the sum of the first and third resistors - R1 + R3 Ω
`ConfigEOLResState5`	State 5 is the input resistance state that meets the value of the sum of the first, second and third resistors - R1 + R2 + R3 Ω
`ConfigEOLResState6`	State 6 is the final input resistance state - infinity Ω (open circuit)

Programming Input Doubling with a 3 Resistor EOL

Examples of the commands required to program input doubling are illustrated below.

Normally Closed Configuration

Below is an example of programming for a Normally Closed 3 resistor EOL configuration, where the first resistor is 1K (a value of 1000), the second resistor 1K (a value of 1000), and the third resistor 2K4 (a value of 2400).

For a Normally Closed contact type the **Low Input Opened** state (1) comes before the **High Input Opened** state (5). The **Both Inputs Closed** state (0) comes before the **Both Inputs Opened** state (3).

1. Navigate to **Programming | Inputs** and select or add the physical input to be programmed.
2. Scroll down to the **Commands** section and enter the following commands:
   - `EOL = Res`
   - `NumberOfEOLResistors = 3`
   - `ConfigEOLRes1 = 1000`
   - `ConfigEOLRes2 = 1000`
   - `ConfigEOLRes3 = 2400`
   - `ConfigEOLResState1 = 2`
   - `ConfigEOLResState2 = 0`
   - `ConfigEOLResState3 = 1`
   - `ConfigEOLResState4 = 5`
   - `ConfigEOLResState5 = 3`
   - `ConfigEOLResState6 = 6`
3. Enter a hysteresis command, such as:
   - `ConfigEOLHysteresis = 250`
4. Click **Save**.
5. Click **Add** and create the corresponding high input record.
   - Set the **Module address** to the same as the physical input.
   - Set the **Module input** to the value of the physical input + the maximum physical inputs of the module the physical input is connected to.

     For example, if the physical input is input 5 on a PRT-ZX16-DIN, the high input needs to be input 21.
6. Click **Save**.
7. If you are configuring expander module inputs, navigate to the relevant expander programming (e.g. **Expanders | Input expanders**). Right click on each expander record that supports the inputs modified above, and click **Update module**.

Normally Open Configuration

Below is an example of programming for a Normally Open 3 resistor EOL configuration, where the first resistor is 1K (a value of 1000), the second resistor 1K (a value of 1000), and the third resistor 2K4 (a value of 2400).

For a Normally Open contact type the **High Input Opened** state (5) comes before the **Low Input Opened** state (1). The **Both Inputs Opened** state (3) comes before the **Both Inputs Closed** state (0).

1. Navigate to **Programming | Inputs** and select or add the physical inputs to be programmed.
2. Scroll down to the **Commands** section and enter the following commands:
   - `EOL = Res`
   - `NumberOfEOLResistors = 3`
   - `ConfigEOLRes1 = 1000`
   - `ConfigEOLRes2 = 1000`
   - `ConfigEOLRes3 = 2400`
   - `ConfigEOLResState1 = 2`
   - `ConfigEOLResState2 = 3`
   - `ConfigEOLResState3 = 5`
   - `ConfigEOLResState4 = 1`
   - `ConfigEOLResState5 = 0`
   - `ConfigEOLResState6 = 6`
3. Enter a hysteresis command, such as:
   `ConfigEOLHysteresis = 250`
4. Click **Save**.
5. Click **Add** and create the corresponding high input record.
   - Set the **Module address** to the same as the physical input.
   - Set the **Module input** to the value of the physical input + the maximum physical inputs of the module the physical input is connected to.

     For example, if the physical input is input 6 on a PRT-ZX16-DIN, the high input needs to be input 22.
6. Click **Save**.
7. If you are configuring expander module inputs, navigate to the relevant expander programming (e.g. **Expanders | Input expanders**). Right click on each expander record that supports the inputs modified above, and click **Update module**.

Designers & manufacturers of integrated electronic access control, security and automation products.
Designed & manufactured by Integrated Control Technology Ltd.