

LED Gateway Siemens Communications Manual

John Machado
jrmachado@electro-matic.com (248)924-1261

July 2025

Contents

1	Introduction	3
2	Commissioning	4
2.1	Hardware Configuration	5
2.2	Import Interface	6
2.3	Interface Data Block Settings	8
2.4	Interface Defaults	9
3	PLC Interface	10
3.1	Global	11
3.2	Rows	12
3.3	Segments	14
4	Example	16

1 Introduction

This manual describes how to commission and use a PLC with our LED Gateway. There are three main sections:

1. **Commissioning**: how to setup a PLC project
2. **PLC Interface**: a description of the PLC interface
3. **Example**: creation of a table using the PLC interface

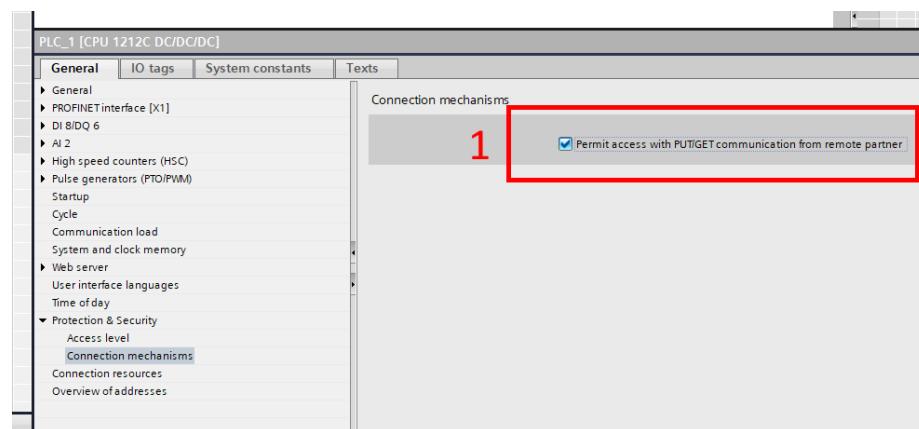
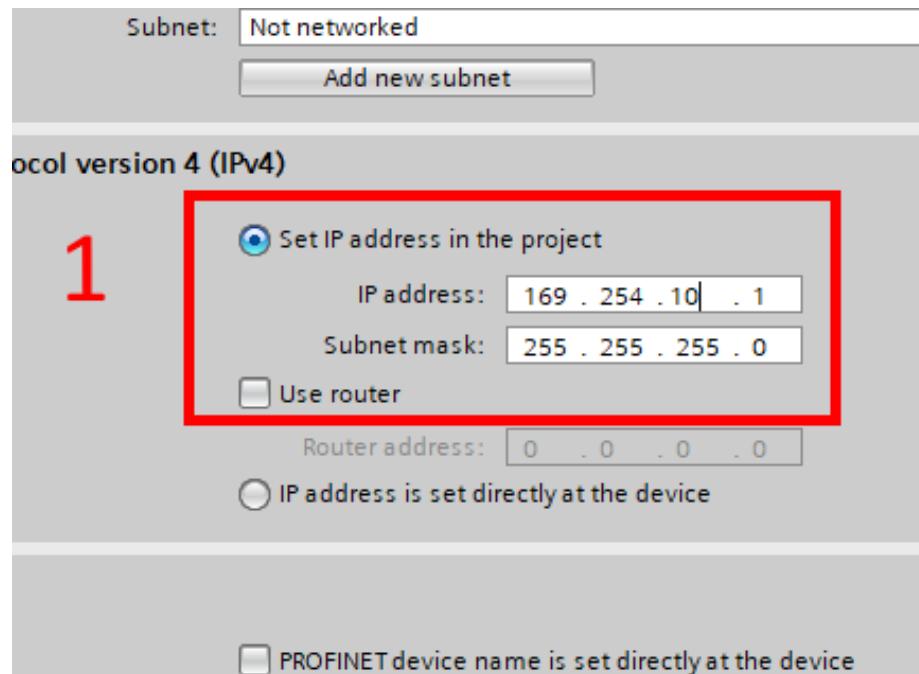
2 Commissioning

The commissioning process has 4 steps.

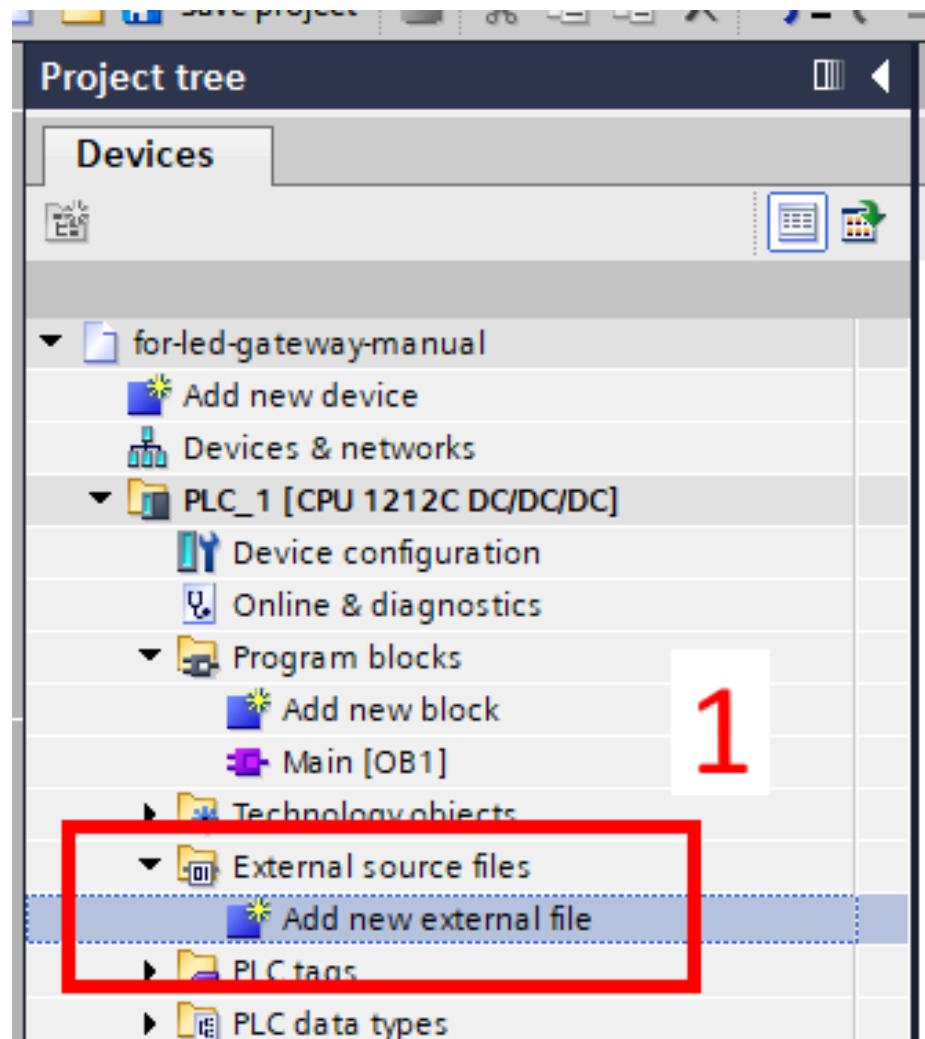
1. Hardware Configuration
2. Import Interface
3. Interface Data Block Settings
4. Interface Defaults

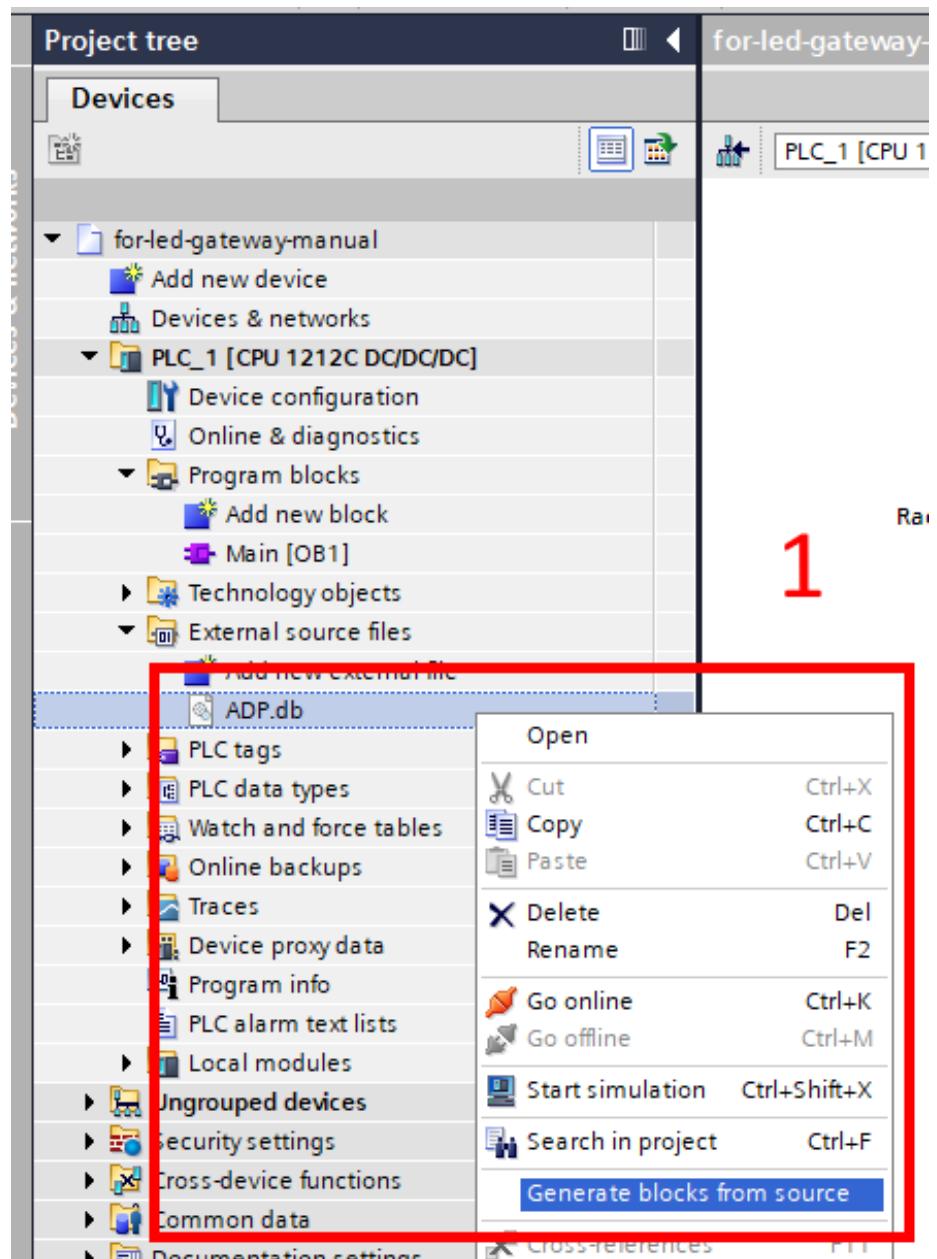
There is no connection to manage on the PLC. The gateway will connect to and scan the PLC at 2 Hz. Connection diagnostics are available in the gateway web user interface.

2.1 Hardware Configuration

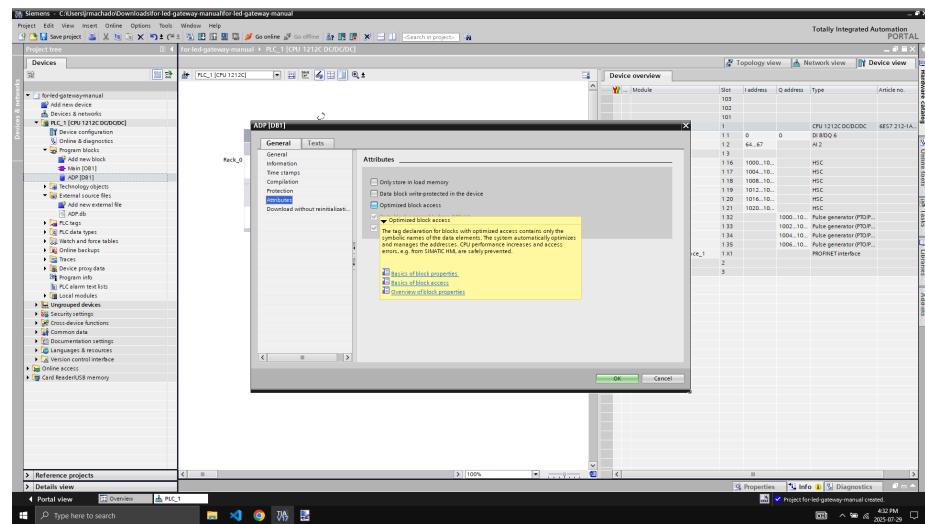


2.2 Import Interface





2.3 Interface Data Block Settings



2.4 Interface Defaults

Default values for each ADP_FORMAT element is -1. TEXT elements are empty strings by default. Many formatting options are optional and will use the default values specified in the **Defaults** web user interface page. Some are necessary for a message to display. Below is an example of what can be loaded as default values in an interface to display a simple welcome message.

- ADP_FORMAT[55]: 0
 - sets format segment 0 to row 0 (first row)
- TEXT[0]: WELCOME TO LED GATEWAY

3 PLC Interface

1. ADP_FORMAT

- options to change how text is displayed
 - ADP_FORMAT[0:304]
 - length: 304 bytes
 - sections: 3
- (a) Global Formatting
- applicable across the entire face of the sign
 - ADP_FORMAT[0:4]
 - length: 4 bytes
 - sections: 1
- (b) Row Formatting
- applicable to specified row
 - ADP_FORMAT[5:54]
 - length: 50 bytes
 - sections: 10 rows
- (c) Segment Formatting
- applicable to a segment of text, which is a group of characters that share common formatting
 - ADP_FORMAT[55:304]
 - length: 250 bytes
 - sections: 50 segments

2. ADP_TEXT

- an array of length 30 strings whose index specifies in which format segment the text will appear
- ADP_TEXT[0:49]
- length: 1600 bytes ($50 * (30 + 2)$)
- sections: 50

3.1 Global

- hold time
 - when messages do not scroll, the display holds on sections for this amount of time in seconds
 - ADP_FORMAT[0]
 - valid values:
 - * 0 to 99
- scroll speed
 - message scroll speed
 - ADP_FORMAT[1]
 - valid values:
 - * 1: slowest
 - * 2: slow
 - * 3: normal
 - * 4: fast
 - * 5: fastest
- vertical alignment
 - vertical alignment of entire sign face
 - ADP_FORMAT[2]
 - valid values:
 - * 1: top
 - * 2: middle
 - * 3: fill
 - * 4: bottom
- in mode
 - hold or scroll
 - ADP_FORMAT[3]
 - valid values:
 - * 1: top
 - * 2: middle
 - * 3: fill
 - * 4: bottom

3.2 Rows

- font weight
 - bold or normal
 - ADP_FORMAT[5 + row # * 5]
 - valid values:
 - * 1: bold
 - * 2: normal
 - Note, not all font sizes are available with all font weights. When an invalid combination is used, text may not display on sign.

- font size
 - size in pixels
 - ADP_FORMAT[6 + row # * 5]
 - valid values:
 - * bold:
 - 5: Bold
 - 11: Bold
 - 14: Bold
 - 15: Bold
 - 16: Bold
 - 22: Bold
 - 30: Bold
 - 32: Bold
 - 40: Bold
 - * normal:
 - 5: Normal
 - 7: Normal
 - 9: Normal
 - 11: Normal
 - 14: Normal
 - 15: Normal
 - 16: Normal
 - 22: Normal
 - 24: Normal
 - 30: Normal
 - 32: Normal
 - 40: Normal
 - 64: Normal
 - 71: Normal
 - 80: Normal
 - 88: Normal
- horizontal alignment
 - horizontal alignment in the row
 - ADP_FORMAT[7 + row # * 5]
 - valid values:
 - * 1: left
 - * 2: center
 - * 3: right

3.3 Segments

- row number
 - Segments are assigned to rows. Segments appear in order of index from left to right in a row. If an invalid row number is entered the segment will not be displayed.
 - ADP_FORMAT[55 + segment # * 5]
 - valid values:
 - * 1: 0 to 9
- background color
 - color of the background of a given character cell
 - ADP_FORMAT[56 + segment # * 5]
 - valid values:
 - * 1: black
 - * 2: red
 - * 3: green
 - * 4: yellow
 - * 5: mix 1
 - * 6: mix 2
 - * 7: mix 3
 - * 8: mix 4
 - * 9: blue
 - * 10: white
- foreground color
 - color of the text for a given character cell
 - ADP_FORMAT[57 + segment # * 5]
 - valid values:
 - * 1: black
 - * 2: red
 - * 3: green
 - * 4: yellow
 - * 5: mix 1
 - * 6: mix 2
 - * 7: mix 3
 - * 8: mix 4
 - * 9: blue
 - * 10: white

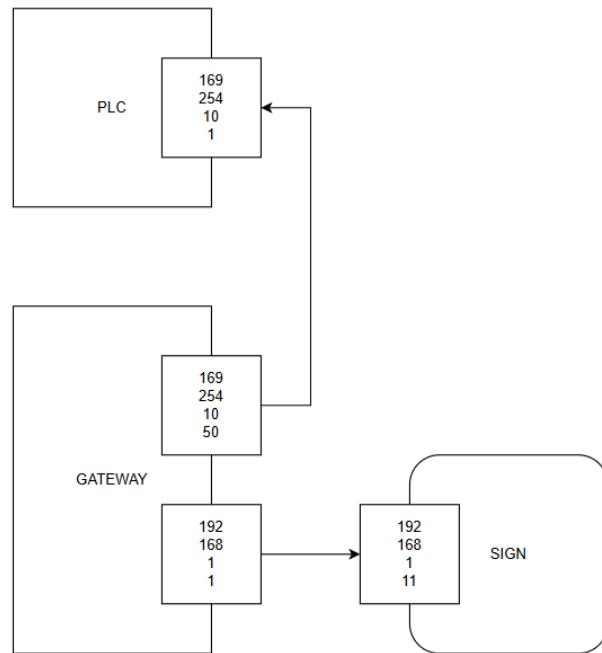
- flash
 - makes text flash
 - ADP_FORMAT[58 + segment # * 5]
 - valid values:
 - * 1: on
 - * 2: off

4 Example

This section will walk you through the creation of a table for an LED sign. Many steps in the gateway web user interface (UI) are not typically necessary because the gateway will ship with a certain level of configuration. These steps have been labeled as redundant, but have been included to give a complete picture of the communication pathway between the PLC, gateway and sign.

1. setup

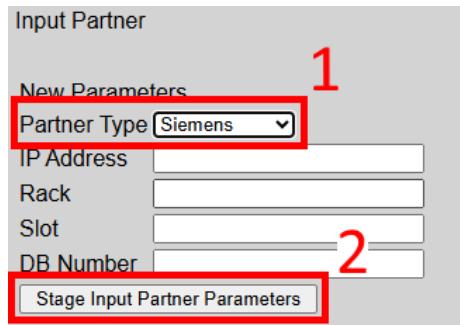
- PLC: 169.254.10.1
- Gateway:
 - X1P1: 169.254.10.50
 - X1P2: 192.168.1.1
- Sign: 192.168.1.11



2. gateway UI

(a) REDUNDANT: change input partner in gateway to SIEMENS

- no other settings need to be entered, just change the type and stage



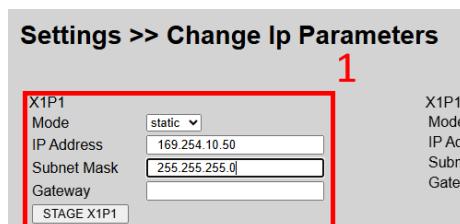
(b) REDUNDANT: reboot gateway

- this needs to be done to load some Siemens specific communication settings in the gateway



(c) set ip parameters

- i. set x1p1 parameters in gateway UI
 - this interface will communicate with the PLC



- ii. **REDUNDANT**: set x1p2 parameters in gateway UI
- this interface will communicate with the sign

X1P2

Mode static

IP Address 192.168.1.1

Subnet Mask 255.255.255.0

Gateway

STAGE X1P2

- (d) **REDUNDANT**: set data flow
- we will be reading from a PLC and sending an ADP message to a sign

Data Flow Settings

Input

- ADP
- API
- PLC

Output

- ADP
- Display Port
- WEB

Set Flow

- (e) set defaults
- defaults will be applied to any formatting codes that are missing or invalid

Settings >> Defaults

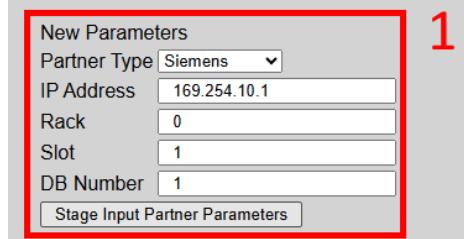
background_color	black
drive	E
flash	off
font_size	5
font_weight	normal
foreground_color	red
hold_time	3
horizontal_alignment	center
in_mode	scroll
line_spacing	1
out_mode	hold
scroll_speed	fastest
vertical_alignment	middle
wrap	off

SUBMIT

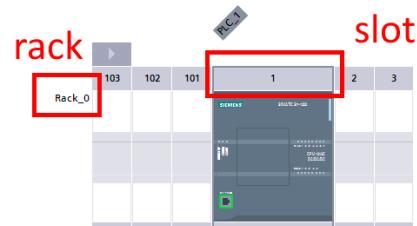
(f) set ip partner parameters

i. set input partner to SIEMENS

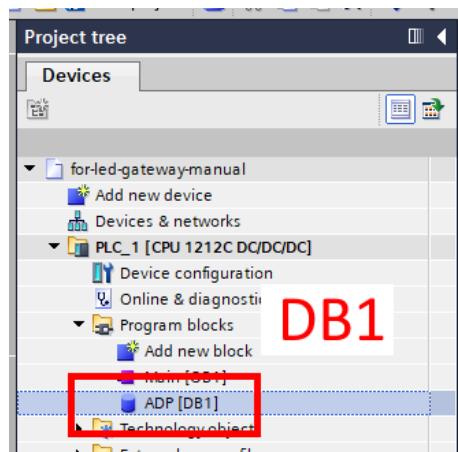
- this time, we need to fill in all settings and stage



- The ip address is a common setting but rack and slot are more unique to Siemens addressing. These are settings that can be obtained in your project's hardware configuration.



- The DB Number is the absolute addressing number of the data block in the PLC. You can see this listed in square brackets next to the data block's symbolic name in the project tree in TIA Portal.



ii. set output partner

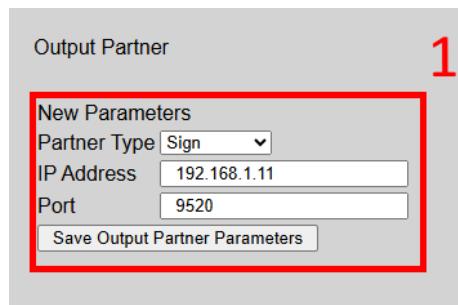


Figure 1: Caption

- the sign's ip address is display when the sign boots

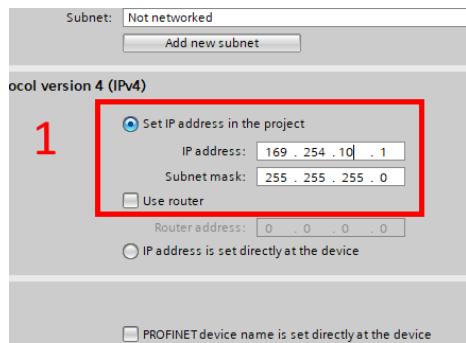


- the sign's port will either be 3001 or 9520

3. tia portal

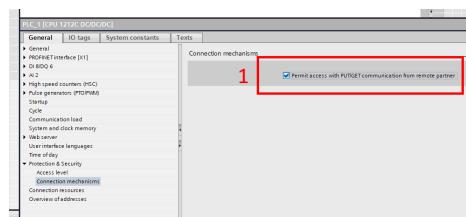
(a) set plc ip address

- the PLC's ip address should be addressable by X1P1



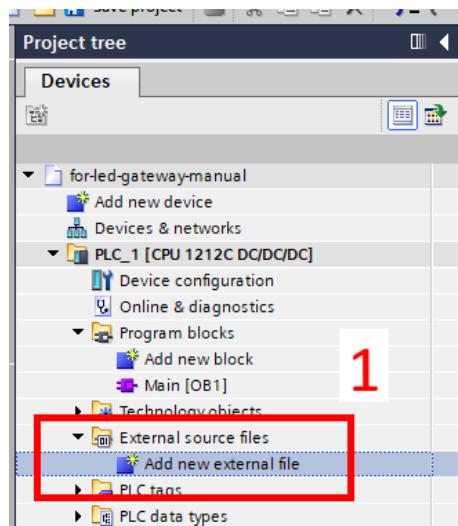
(b) enable get/put

- this setting allows third party s7 drivers to read plc data

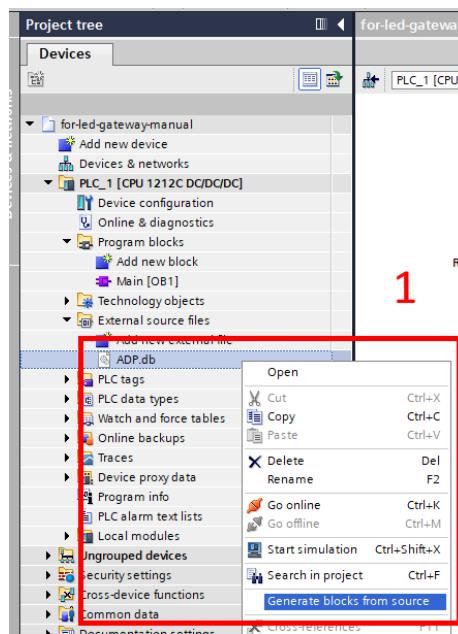


(c) create interface

- the interface is a data block in the PLC that is structured so that the gateway can read it
- import the interface's source file

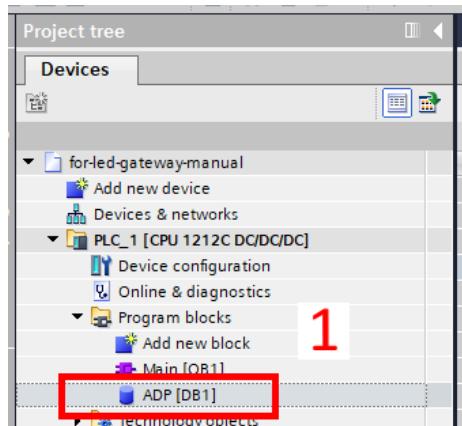


- generate a data block from the source file



(d) create welcome message

- open the data block resulting from step (e)



- set segment 0's row number to 0 (first row)

Name	Data type	Offset	Start value	1
FORMAT[0]	Byte	54.0	16#0000	
FORMAT[55]	Byte	55.0	0	
FORMAT1591	Byte	56.0	16#00FF	
FORMAT281	Byte	28.0	16#00FF	

- set segment 0's text to your welcome message

Name	Data type	Offset	Start value	1
Static				
FORMAT	Array[0..304] of Byte	0.0		
TEXT	Array[0..49] of String	306.0		
TEXT[0]	String[30]	306.0	"WELCOME TO LED GATEWAY"	
TEXT[1]	String[30]	358.0		

- load to PLC
- the sign will show your welcome message



(e) create table

- These steps show how to format a message to create a table. This manual shows changes to the interface's values directly through a watch table, but in practice a user's code would load process data.
- i. create header row
 - We will determine our column spacing as we create a header row.

"/\$PFORMAT[0]	\$00100000	Hex	16400	16400	<input checked="" type="checkbox"/>	format segment 0 row number
"/\$PFORMAT[1]	\$00100001	Hex	16401	16401	<input checked="" type="checkbox"/>	format segment 0 foreground color
"/\$PFORMAT[2]	\$00100002	Hex	16402	16402	<input checked="" type="checkbox"/>	format segment 0 back
"/\$PFORMAT[3]	\$00100003	Hex	16403	16403	<input checked="" type="checkbox"/>	text for format segment 0
"/\$PFORMAT[4]	F00100000000	String	"PART"	"PART"	<input checked="" type="checkbox"/>	format segment 1 row number
"/\$PFORMAT[5]	\$00100005	Hex	16405	16405	<input checked="" type="checkbox"/>	format segment 1 foreground color
"/\$PFORMAT[6]	\$00100006	Hex	16406	16406	<input checked="" type="checkbox"/>	format segment 1 back
"/\$PFORMAT[7]	\$00100007	Hex	16407	16407	<input checked="" type="checkbox"/>	text for format segment 1
"/\$PFORMAT[8]	\$00100008	Hex	16408	16408	<input checked="" type="checkbox"/>	format segment 2 row number
"/\$PFORMAT[9]	\$00100009	Hex	16409	16409	<input checked="" type="checkbox"/>	format segment 2 foreground color
"/\$PFORMAT[10]	\$0010000A	Hex	1640A	1640A	<input checked="" type="checkbox"/>	format segment 2 back
"/\$PFORMAT[11]	\$0010000B	Hex	1640B	1640B	<input checked="" type="checkbox"/>	text for format segment 2
"/\$PFORMAT[12]	F00100000000	String	"COUNT"	"COUNT"	<input checked="" type="checkbox"/>	format segment 3 row number
"/\$PFORMAT[13]	\$0010000C	Hex	1640C	1640C	<input checked="" type="checkbox"/>	format segment 3 foreground color
"/\$PFORMAT[14]	\$0010000D	Hex	1640D	1640D	<input checked="" type="checkbox"/>	format segment 3 back
"/\$PFORMAT[15]	\$0010000E	Hex	1640E	1640E	<input checked="" type="checkbox"/>	text for format segment 3
"/\$PFORMAT[16]	\$0010000F	Hex	1640F	1640F	<input checked="" type="checkbox"/>	format segment 4 row number
"/\$PFORMAT[17]	\$00100010	Hex	16410	16410	<input checked="" type="checkbox"/>	format segment 4 foreground color
"/\$PFORMAT[18]	\$00100011	Hex	16411	16411	<input checked="" type="checkbox"/>	format segment 4 back
"/\$PFORMAT[19]	F00100000000	String	"TIME"	"TIME"	<input checked="" type="checkbox"/>	text for format segment 4
"/\$PFORMAT[20]	\$00100010	Hex	16410	16410	<input checked="" type="checkbox"/>	format segment 5 row number
"/\$PFORMAT[21]	\$00100011	Hex	16411	16411	<input checked="" type="checkbox"/>	format segment 5 foreground color
"/\$PFORMAT[22]	\$00100012	Hex	16412	16412	<input checked="" type="checkbox"/>	format segment 5 back
"/\$PFORMAT[23]	F00100000000	String	" "	" "	<input checked="" type="checkbox"/>	text for format segment 5

- Spaces are manually added to get the horizontal alignment for each column in the row. In this example:
 - column 1 has 9 spaces
 - * PART_____
 - * this column will be left aligned
 - column 2 has 5 spaces
 - * COUNT
 - * this column will be right aligned
 - column 3 has 5 spaces
 - * _TIME
 - * this column will be right aligned



ii. populate rows 1 and 2

- Content is added to these rows to get gauge vertical alignment.

Name	Address	Display format	Monitor value	Modify value	Tag comment
"ADP_FORMATEC1	N011.0B82	Hex	164FF	164FF	vertical alignment
"ADP_FORMATEC51	N011.0B855	Hex	16400	16400	format segment 1 row number
"ADP_FORMATEC57	N011.0B857	Hex	164FF	164FF	format segment 1 foreground color
"ADP_FORMATEC58	N011.0B858	Hex	164FF	164FF	format segment 1 back
"ADP_B[1]D1	P0E01.0B5306.0	String	Y	Y	text for format segment 2
"ADP_FORMATEC60	N011.0B860	Hex	16400	16400	format segment 3 row number
"ADP_FORMATEC62	N011.0B862	Hex	164FF	164FF	format segment 3 foreground color
"ADP_FORMATEC63	N011.0B863	Hex	164FF	164FF	format segment 3 back
"ADP_B[2]D1	P0E01.0B5308.0	String	COUNT	COUNT	text for format segment 1
"ADP_FORMATEC65	N011.0B865	Hex	16400	16400	format segment 2 row number
"ADP_FORMATEC67	N011.0B867	Hex	164FF	164FF	format segment 2 foreground color
"ADP_FORMATEC68	N011.0B868	Hex	164FF	164FF	format segment 2 back
"ADP_B[3]D1	P0E01.0B5309.0	String	TIME	TIME	text for format segment 2
"ADP_FORMATEC70	N011.0B870	Hex	16401	16401	format segment 3 row number
"ADP_FORMATEC72	N011.0B872	Hex	164FF	164FF	format segment 3 foreground color
"ADP_FORMATEC73	N011.0B873	Hex	164FF	164FF	format segment 3 back
"ADP_B[4]D1	P0E01.0B5402.0	String	V	V	text for format segment 3
"ADP_FORMATEC75	N011.0B875	Hex	164FF	164FF	format segment 4 row number
"ADP_FORMATEC76	N011.0B876	Hex	164FF	164FF	format segment 4 foreground color
"ADP_FORMATEC78	N011.0B878	Hex	164FF	164FF	format segment 4 back
"ADP_B[5]D1	P0E01.0B5413.0	String	TIME	TIME	text for format segment 4
"ADP_FORMATEC80	N011.0B880	Hex	164FF	164FF	format segment 5 row number
"ADP_FORMATEC82	N011.0B882	Hex	164FF	164FF	format segment 5 foreground color
"ADP_FORMATEC83	N011.0B883	Hex	164FF	164FF	format segment 5 back
"ADP_B[6]D1	P0E01.0B5444.0	String	END	END	text for format segment 5
"ADP_FORMATEC85	N011.0B885	Hex	16402	16402	format segment 6 row number
"ADP_FORMATEC87	N011.0B887	Hex	164FF	164FF	format segment 6 foreground color
"ADP_B[7]D1	P0E01.0B5448.0	String	Y	Y	text for format segment 6
"ADP_FORMATEC89	N011.0B889	Hex	164FF	164FF	format segment 7 row number
"ADP_FORMATEC92	N011.0B892	Hex	164FF	164FF	format segment 7 foreground color
"ADP_B[8]D1	P0E01.0B5530.0	String	-	-	text for format segment 7
"ADP_FORMATEC95	N011.0B895	Hex	164FF	164FF	format segment 8 row number
"ADP_FORMATEC97	N011.0B897	Hex	164FF	164FF	format segment 8 foreground color
"ADP_B[9]D1	P0E01.0B5532.0	String	0	0	text for format segment 8



- The table can vertically aligned to the center

Name	Address	Display format	Monitor value	Modify value	Tag comment
"ADP_FORWARD2	N081.0B82	Hex	16403	16403	vertical alignment



iii. fill in remaining data



- row 1:
 - part number
 - * "ABC" is 3 characters and this column needs to be 9 characters so we append 6 spaces for left alignment.
 - * ABC_____
 - count
 - * "123" is 3 characters and this column needs to be 5 characters so we prepend 2 spaces for right alignment.
 - * __123
 - time
 - * "9" is 1 character and this column needs to be 5 characters so we prepend 4 spaces for right alignment.
 - * ___9

- row 2:
 - part number
 - * "DEF" is 3 characters and this column needs to be 9 characters so we append 6 spaces for left alignment.
 - * DEF_____
 - count
 - * "45" is 2 characters and this column needs to be 5 characters so we prepend 3 spaces for right alignment.
 - * ___45
 - time
 - * "876" is 3 character and this column needs to be 5 characters so we prepend 2 spaces for right alignment.
 - * __876

iv. apply formatting

Name	Address	Display format	Monitor value	Modify value	...	Tag comment
*'OP_FORMAT[2]	0x01 0B8E2	Hex	16403	16403		vertical alignment
*'OP_FORMAT[5]	0x01 0B8E55	Hex	16400	16400		format segment 0 row number
*'OP_FORMAT[7]	0x01 0B8E57	Hex	16409	16409		format segment 0 foreground color
*'OP_FORMAT[8]	0x01 0B8E58	Hex	16408	16408		format segment 0 background color
*'OP_TEXT[0]	PK01 0B1206.D	String	'PART'	'PART'		text for format segment 0
*'OP_FORMAT[9]	0x01 0B8E60	Hex	16400	16400		format segment 1 row number
*'OP_FORMAT[12]	0x01 0B8E62	Hex	16409	16409		format segment 1 foreground color
*'OP_FORMAT[14]	0x01 0B8E64	Hex	16408	16408		format segment 1 background color
*'OP_FORMAT[15]	0x01 0B8E65	Hex	16407	16407		format segment 2 row number
*'OP_FORMAT[16]	0x01 0B8E66	Hex	1640F	1640F		format segment 2 foreground color
*'OP_FORMAT[17]	0x01 0B8E67	Hex	1640E	1640E		format segment 2 background color
*'OP_FORMAT[18]	0x01 0B8E68	Hex	1640D	1640D		format segment 3 row number
*'OP_FORMAT[19]	0x01 0B8E69	Hex	1640C	1640C		format segment 3 foreground color
*'OP_FORMAT[20]	0x01 0B8E6A	Hex	1640B	1640B		format segment 3 background color
*'OP_FORMAT[21]	0x01 0B8E6B	String	'REC'	'REC'		text for format segment 3
*'OP_FORMAT[25]	0x01 0B8E75	Hex	16401	16401		format segment 4 row number
*'OP_FORMAT[26]	0x01 0B8E76	Hex	16402	16402		format segment 4 foreground color
*'OP_FORMAT[27]	0x01 0B8E77	Hex	1640F	1640F		format segment 4 background color
*'OP_FORMAT[28]	0x01 0B8E78	Hex	1640F	1640F		format segment 5 row number
*'OP_FORMAT[29]	0x01 0B8E79	String	'123'	'123'		text for format segment 4
*'OP_FORMAT[30]	0x01 0B8E7A	Hex	16401	16401		format segment 5 foreground color
*'OP_FORMAT[31]	0x01 0B8E7B	Hex	16402	16402		format segment 5 background color
*'OP_FORMAT[32]	0x01 0B8E83	Hex	1640F	1640F		format segment 6 row number
*'OP_FORMAT[33]	0x01 0B8E84	String	'TIME'	'TIME'		text for format segment 2
*'OP_FORMAT[34]	0x01 0B8E85	String	'COUNT'	'COUNT'		text for format segment 1
*'OP_FORMAT[35]	0x01 0B8E86	String	'PART'	'PART'		text for format segment 0
*'OP_FORMAT[36]	0x01 0B8E87	String	'REC'	'REC'		text for format segment 3
*'OP_FORMAT[37]	0x01 0B8E88	String	'COUNT'	'COUNT'		text for format segment 1
*'OP_FORMAT[38]	0x01 0B8E89	String	'TIME'	'TIME'		text for format segment 2
*'OP_FORMAT[39]	0x01 0B8E8A	String	'123'	'123'		text for format segment 4
*'OP_FORMAT[40]	0x01 0B8E8B	String	'45'	'45'		text for format segment 5
*'OP_FORMAT[41]	0x01 0B8E8C	String	'876'	'876'		text for format segment 6



- The header row was blue so each of the three segments foreground color byte had 9 loaded.
- The first data row was green so each of the three segments foreground color byte had a 3 loaded.
- The second data row was red so each of the tree segments foreground color byte was left unaltered.
- the last column of the second row was flashing so a 2 was loaded into the flash byte.