Using IUT-…-IO RFID Readers in Allen-Bradley PLC

Reader Setup and Add-On Instructions for Use with Easy and Expert Mode

Contents

Introduction 2
Add-On Instructions 2
PF_IUT_IO_Parameters_All 2
Trigger Commands 3
Input Parameters 3
InOut Parameters 3
Out Parameters 7
PF_IUT_IO_Parameters_ICE2 7
PF_RFID_Easy_Long 7
Input Parameters 7
In/Out Parameters 8
Output parameters 11
PF_RFID_Easy_Short 12
PF_RFID_Expert_Long 13
Input Parameters 14
In/Out Parameters 16
Output parameters 18

PEPPERL+FUCHS

Timing Diagrams Expert mode 19
Timing Diagrams Easy Mode 20
Example 21

Introduction

This document will show you how to use your IUT-…-IO-V1-FRx-xx readers in an Allen-Bradley PLC using Studio 5000 Add-On instructions.

Add-On Instructions

Add-on Instructions are easy-to-use function blocks for use in the Rockwell Automation Studio 5000 software. The example projects are written in Studio 5000 V32 but the add-on instructions will work in any project V32 and higher. Easy mode does not require the use of function blocks but Add-On Instructions will make integration of the reader in to your PLC program easier. Expert mode does require function blocks. In either mode all parameters can be read or written.

Instruction	Description
PF_IUT_IO_Parameters_All	Reads and writes all IO-Link parameters (ICE1, ICE2, and ICE11 IO-Link masters)
PF_IOLink_Param_ReadWrite3	Internal Add-on that takes care of one ISDU parameter (ICE1, ICE2, and ICE11 IO-Link masters) [imported automatically]
PF_IUT_IO_Parameters_ICE2	Reads and writes all IO-Link parameters (ICE2 IO-Link master only)
PF_IOLink_Param_ICE2	Internal Add-on that takes care of one ISDU parameter (ICE2 IO-Link master only) [imported automatically]
PF_RFID_Easy_Long	Easy triggering of read and write commands when the reader is configured for Easy mode (Use only with long data representation)
PF_RFID_Easy_Short	Easy triggering of read and write commands when the reader is configured for Easy mode (Use only with short data representation)
PF_RFID_Expert_Long	Easy triggering of read and write commands, UHF configuration and filtering when the B40 reader is configured for Expert Mode
PF_RF_Exp_Rcv	Internal Add-on that takes care of receiving the Expert Mode fragmentation protocol [imported automatically]
PF_RF_Exp_Snd	Internal Add-on that takes care of sending the Expert Mode fragmentation protocol [imported automatically]

PF_IUT_IO_Parameters_All

This Add-on instruction will read or write all IO-Link parameters to the IUT-…-IO RFID heads. It can be used along with various IO-Link masters including ICE1, ICE2, and ICE11.

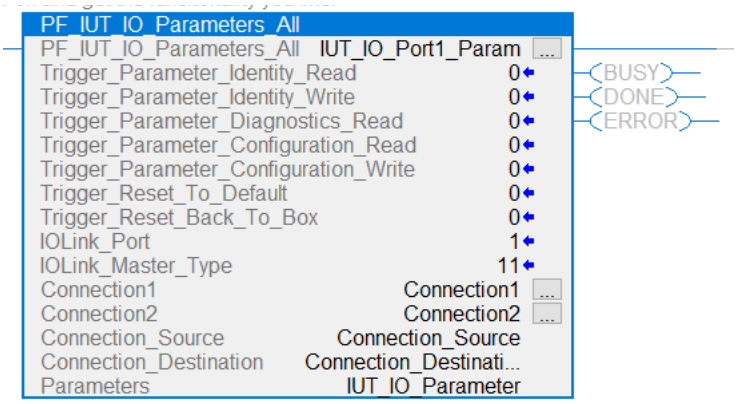


Figure 1: Add-on to read and write all IO-Link parameters

Trigger Commands

The specific trigger command is executed in the off-to-on transition of the below trigger bits. Only one trigger can be sent at a time. All triggers must be off before another one is executed.

Trigger_Parameter_Identity_Read - Reads Identification relation parameters such as Vendor Name, Text, ID, Serial Number....

Trigger_Parameter_Identity_Write - Writes the Application, Function, Location, and Installation Tags

Trigger_Parameter_Diagnostics_Read - Read Diagnostics including Status, Device Characteristics, Temp indicator, Temp Monitor...

Trigger_Parameter_Configuration_Read - Reads RFID Specific parameters such as mode, power...

Trigger_Parameter_Configuration_Write - Writes RFID Specific parameters such as mode, power...

Trigger_Reset_To_Default - Resets all parameters to factory default values. Read all parameters after execution. This command will not reset the Application, Function, Location, or Installation Tags

Trigger_Reset_Back_To_Box - Resets all parameters to default including tags and IO-Link module will go offline (power cycle required after)

Input Parameters

IOLink_Port - Port to read or write IO-Link Parameters, typically 1-8 but this is IO-Link master specific

IOLink_Master_Type – The IO-Link master you are connecting to

0 = ICE1…
1 = Allen Bradley point IO
2 = ICE2…
11 = ICE11…

InOut Parameters

Connection1 – First message instruction. The service code must be set according to the IO-Link master used. Additionally, the Connection_Source and Connection_Destination must be used in the message instruction. See Figure 2. Additionally click on the communication tab and point to the IO-Link master in the Ethernet tree.

Service Code Settings

ICE1 = 4B

AB = 4D

ICE2 = 4B

ICE11 = 4B

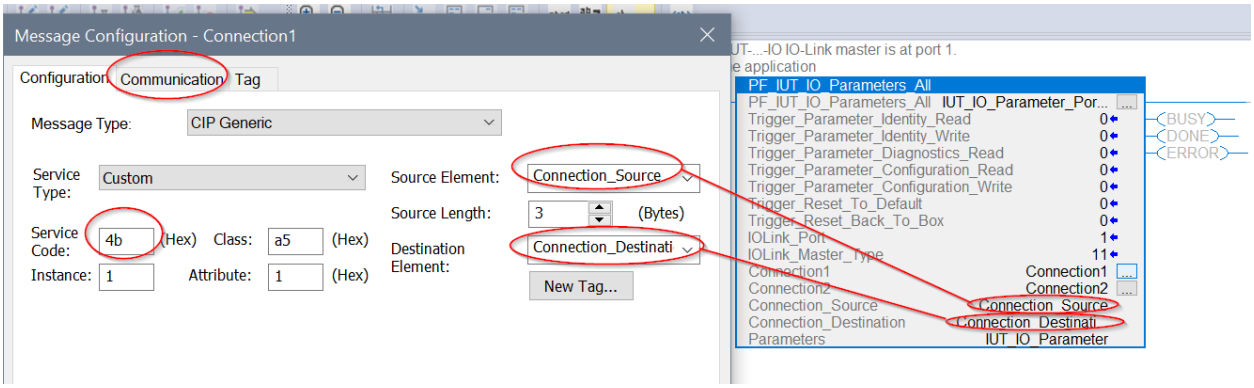


Figure 2: Connection 1 message instruction setup

Connection2 – Second message instruction. The service code must be set according to the IO-Link master used. Additionally, the Connection_Source and Connection_Destination must be used in the message instruction. See Figure 3. Additionally click on the communication tab and point to the IO-Link master in the Ethernet tree.

Service Code Settings

ICE1 = 4C

AB = 4E

ICE2 = 4C

ICE11 = 4C

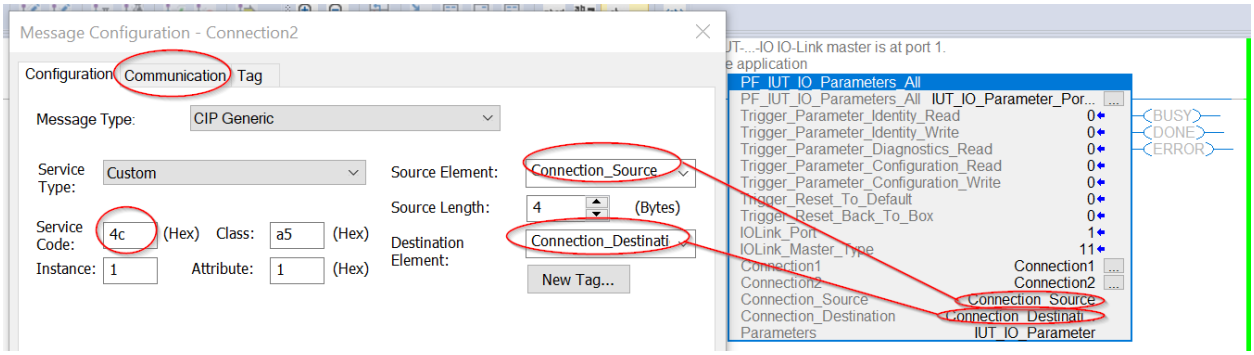


Figure 3: Connection 2 message instruction setup

Connection_Source - Used as source file for connections 1 and 2's MSG Instruction

Connection_Destination - Used as Destination file for connections 1 and 2's MSG Instruction

Parameters – A user defined type PF_IUT_F191_All_Parameters. The parameter set is broken in to three parts Identity, Diagnostics and Configuration. For a detailed description of each parameter see the IO-Link parameter datasheet or the IUT-…IO- user manual.

▲ Parameters	{...}		PF_IUT_F191_All_Parameters
▷ Parameters.Identity	{...}		PF_Identification3
▷ Parameters.Diagnostics	{...}		PF_Diagnostics
▷ Parameters.Configuration	{...}		PF_IUT_F191_Parameters

Figure 4: Parameter set for identification, diagnostics and configuration

▲ Parameters.Identity
▷ Parameters.Identity.Vendor_Name
▷ Parameters.Identity.Vendor_Text
▷ Parameters.Identity.Product_Name
▷ Parameters.Identity.Product_ID
▷ Parameters.Identity.Product_Text
▷ Parameters.Identity.Serial_Number
▷ Parameters.Identity.Hardware_Revision
▷ Parameters.Identity.Firmware_Revision
▷ Parameters.Identity.Application_Specific_Tag
▷ Parameters.Identity.Function_Tag
▷ Parameters.Identity.Location_Tag
▷ Parameters.Identity.Product_URI
▷ Parameters.Identity.Installation_Tag

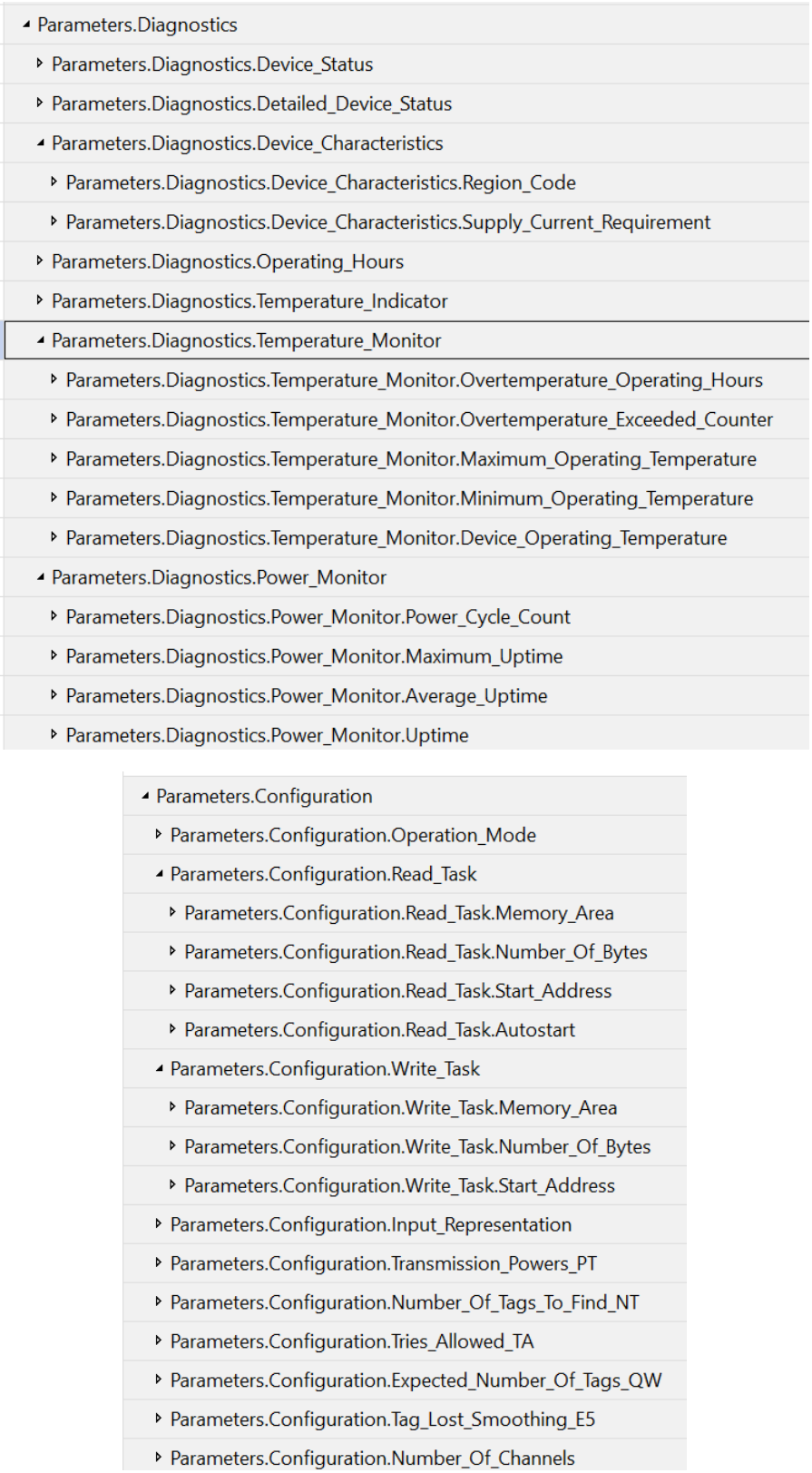


Figure 5: Identity, diagnostic and configuration parameters expanded

Out Parameters

BUSY – On when the Add-on is busy executing

DONE - The instruction has completed without error

ERROR - The execution of the instruction has failed

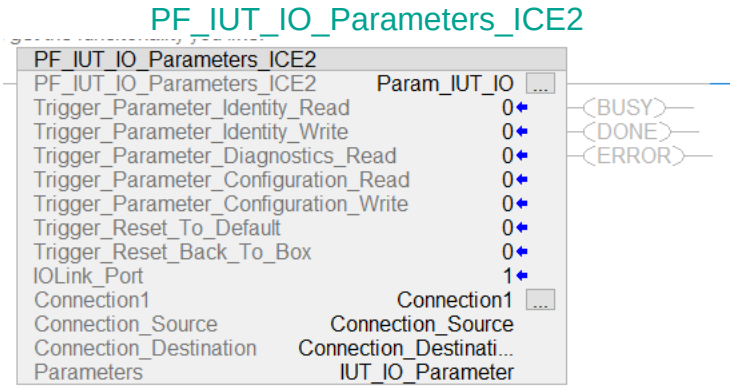


Figure 6: Read and Write parameter instruction for use specifically with the ICE2 IO-Link master.

This add-on instruction is exactly the same as the PF_IUT_IO_Parameters_All add-on except it is optimized for the ICE2 IO-Link masters. The IOLink_Master_Type and Connection2 have been removed.

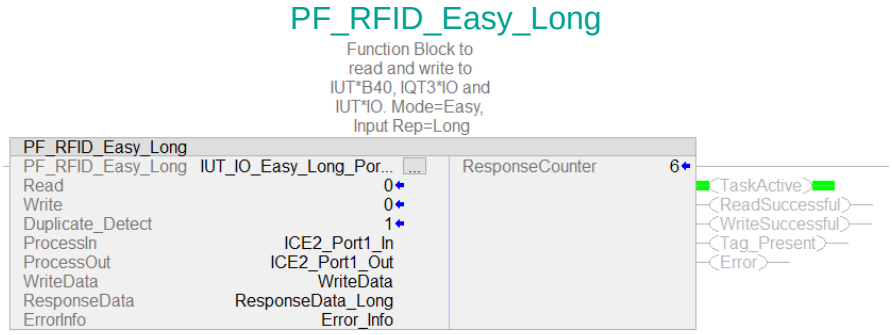


Figure 7: Easy Mode Function Block

This Add-On instruction will read or write data based on the Easy mode IO-Link parameters. Only one bit is required to read or write [**Read, Write**]. Make sure to set the mode to “Easy” and the input representation to “Long”.

Input Parameters

Read – Triggers the reading of tags. (If not in Autoread mode)

0 = No reading

1 = Read all the time

Write – Triggers the writing of tags. (If not in Autoread mode)

0 = No write

1 = Write all the time

Duplicate_Detect – Searches for duplicate tags in *ResponseData* array. Not used when writing

0 = Adds tag to *ResponseData* array at location [0] and shifts all other tags down in array

1 = Searches the *ResponseData* array first. If the tag has not been read before then adds tag to *ResponseData* array at location [0] and shifts all other tags down in array (default).

In/Out Parameters

The **ProcessIn** and the **ProcessOut** Variables are the 32byte process data arrays used by the RFID IO-Link UHF reader. Because a typical IO-Link master contains up to 8 IO-Link masters these arrays are usually created by copying a ports process data to and from these arrays using Copy instructions or Add-on instructions. See Figures 8, 9, and 10 for ICE1, ICE2, and ICE11 examples respectively.

ProcessIn – The input data of the IUT-…-IO reader

ProcessOut – The output data of the IUT-…-IO reader

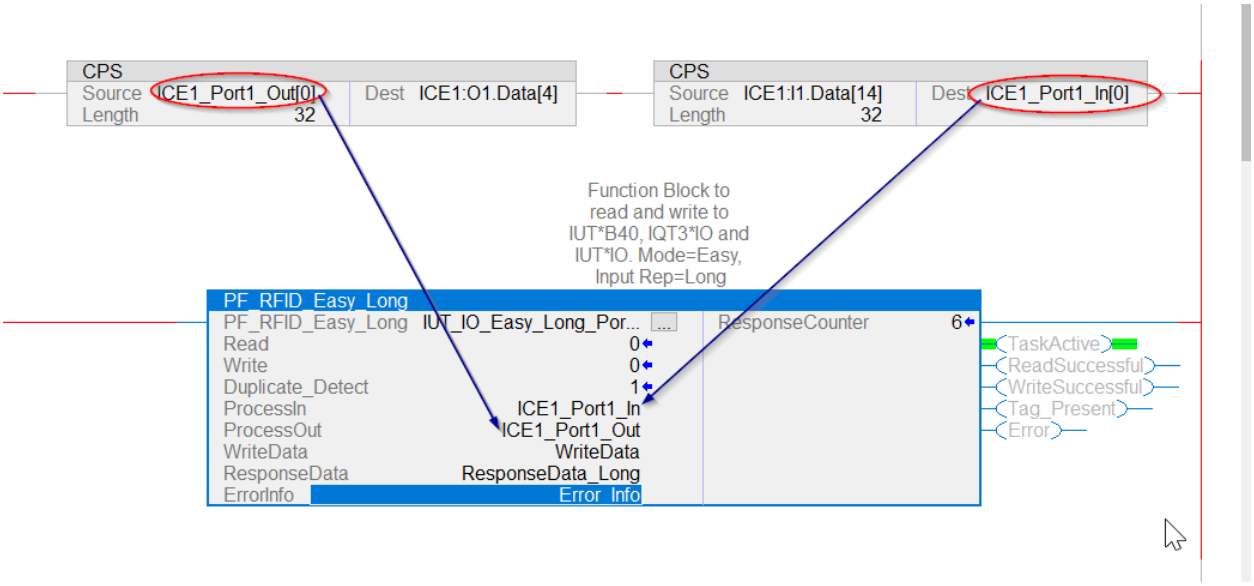


Figure 8: Input and output variables used in ProcessIn and ProcessOut variables for created from ICE1 IO-Link master Port 1

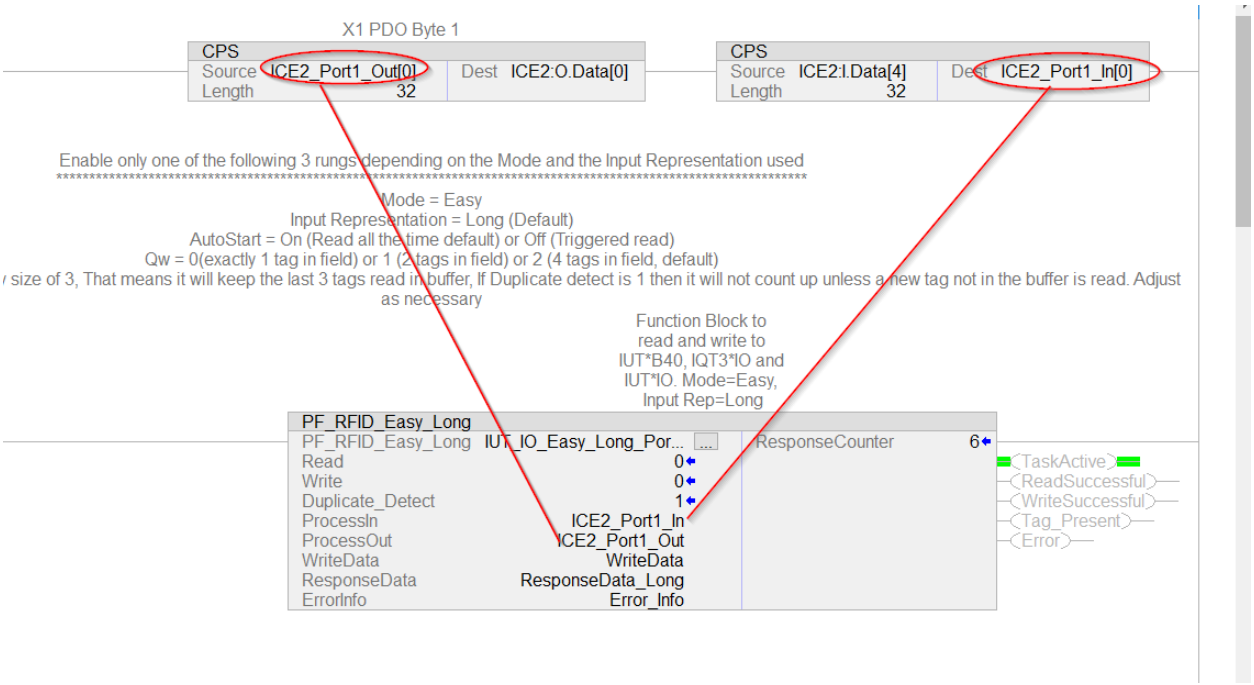


Figure 9: Input and output variables used in ProcessIn and ProcessOut variables for created from ICE2 IO-Link master Port 1

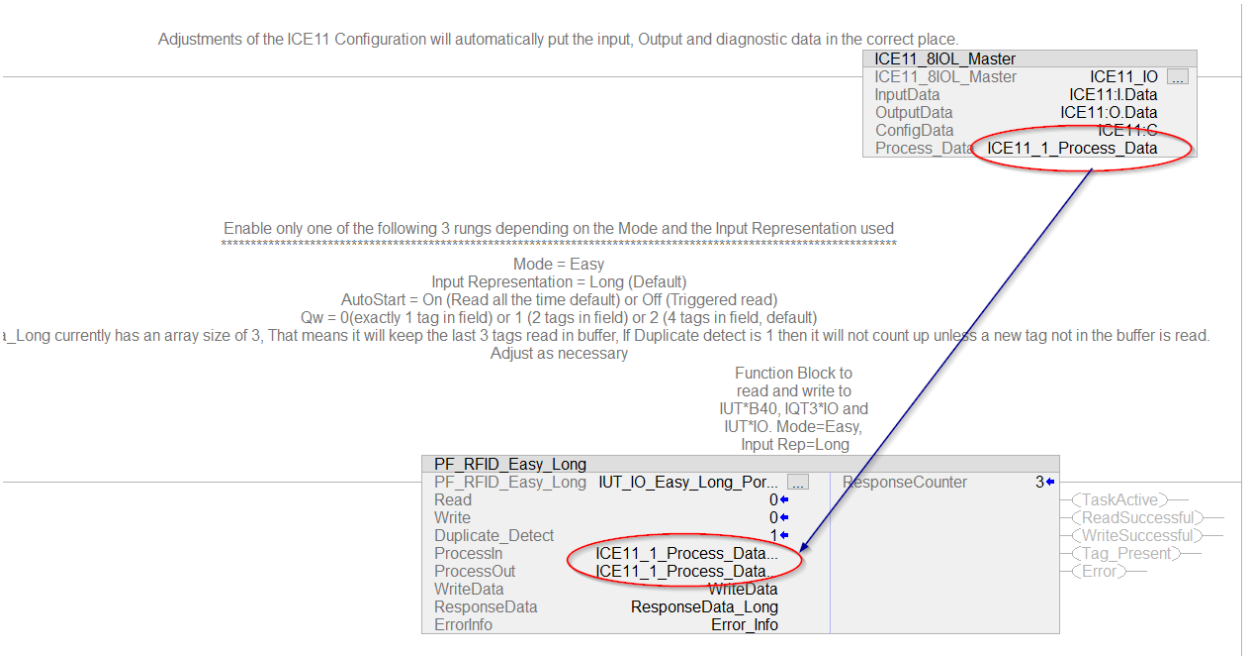


Figure 10: Input and output variables used in ProcessIn and ProcessOut for created from ICE11 Add-on on port 1

WriteData – The string of data that is used when writing to tags. This data only takes affect with the Off-to-On transition of the **Write** bit. The datatype is a STRING variable of type PF_String192.

ResponseData – This is the location where all read data and all responses to write commands are placed. The array size will determine how many tags are stored. The array can be as small as 1 or as large as the PLC memory allows (See Figure 11 for an example for storing 10 tags).

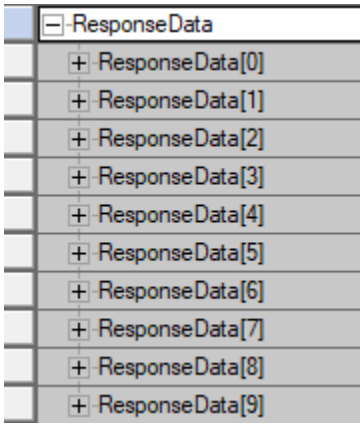


Figure 11: Array size set to 10 to store up to 10 tags

If the array size is more than 1 the newest data is placed in the [0] position and all older data is moved down in the array (See Figure 12).

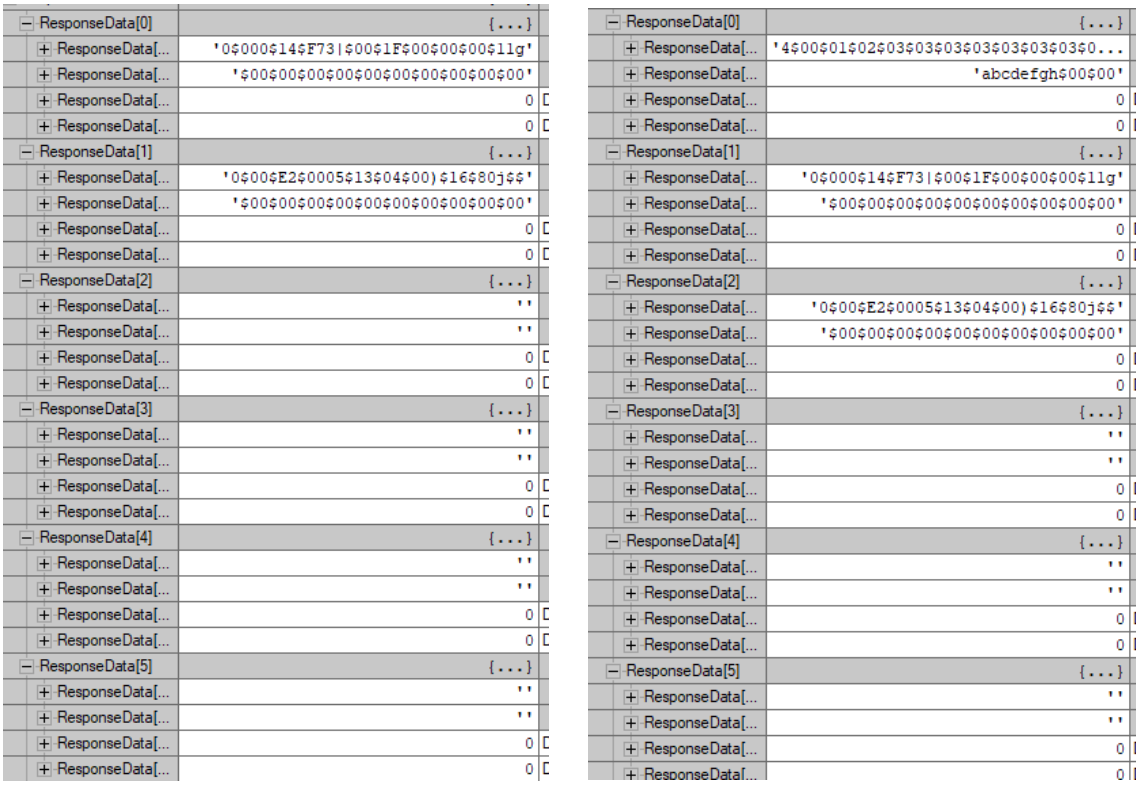


Figure 12: New tag added to position [0] and all older tags moved one index higher

The data for the tag contains up to 4 pieces of information including EPC, USER, RSSI and Power (See Figure 13).

EPC – The UII/EPC code, the EPC code will also include the PC (Protocol Control Word). This is the value searched when checking for duplicate tags. No checking is done when writing tags.

USER – The second set of data read from the tag. It could be the user data or the TID code. Not used when writing tags.

RSSI – The RSSI value of tag (Strength of the tag response when tag first enters the field. Typically only useful if the tag is already there when the read or write command is activated.

Power – The power setting when the tag was read or written

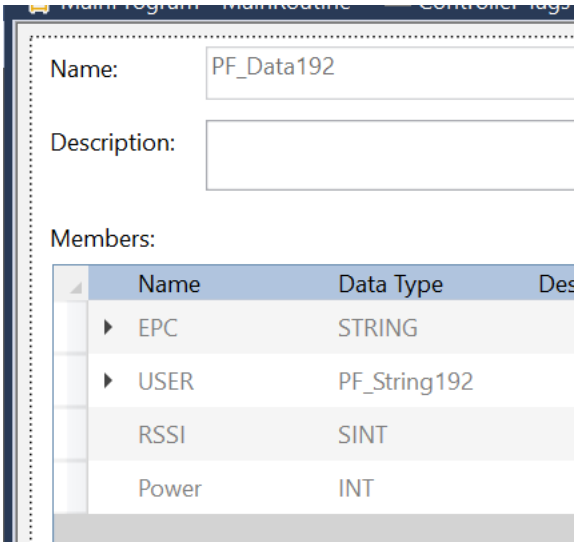


Figure 13: UDT for responses to reading and writing of tags

ErrorInfo – The error code and error string that is returned when the **Error** bit goes on

Output parameters

TaskActive – A reading or writing command is currently running in the head

ReadSuccess – A tag was read successfully

WriteSuccess – A tag was written successfully

Tag_Present – A tag is currently in front of the head

Error – An error occurred, See ErrorInfo for error details

ResponseCounter – Counts up with every successful read or write attempt. No count up on reading if Duplicate_Detect is enabled and a duplicate tag is read.

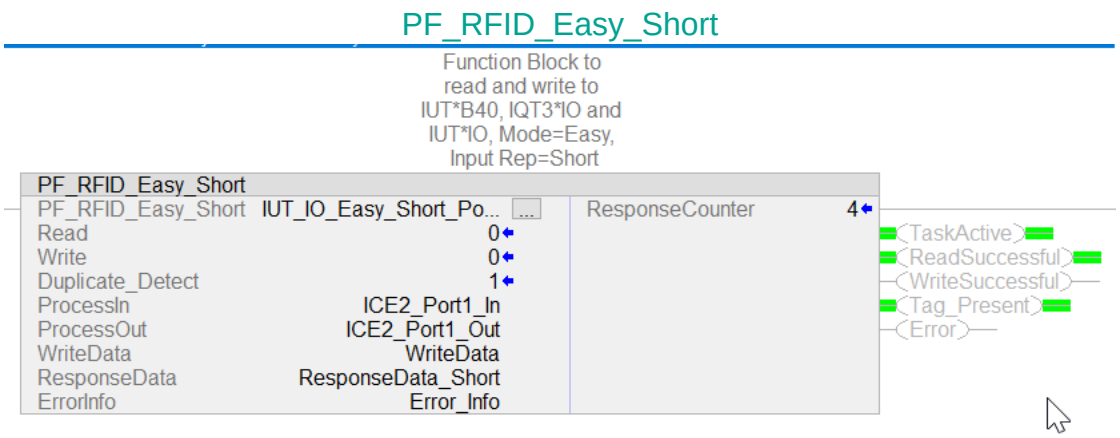


Figure 14: Short form Easy mode add-on instruction

This Add-on is exactly the same as the PF_RFID_Easy_Long Add-on except for these differences. Additionally with the short data representation multiple tags in field are not allowed. The mode must be set to “Easy” and input representation parameter must be set to “Short”

1. The Input_Representation parameter must be set to 16#80 (Short). This can be done using the PF_IUT_IO_Parameters_xxx Add-ons.

Parameters	{...}		PF_IUT_F19...	
Parameters.Identity	{...}		PF_Identific...	
Parameters.Diagnostics	{...}		PF_Diagnos...	
Parameters.Configuration	{...}		PF_IUT_F19...	
Parameters.Configuration.Operation_Mode	16#80	Hex	SINT	0=Expert Mode, 16#80=Easy Mode, Defines the operation mo
Parameters.Configuration.Read_Task	{...}		PF_65_Read...	Parameters associated with reading in Easy mode
Parameters.Configuration.Write_Task	{...}		PF_66_Write...	Parameters associated with writing in Easy mode
Parameters.Configuration.Input_Representation	16#80	Hex	SINT	0=Long, 16#80=Short, Defines the content of the read data. I
Parameters.Configuration.Transmission_Powers_PT	{...}	Decimal	INT[5]	Defines the set of different power levels, which the RFID read/
Parameters.Configuration.Number_Of_Tags_To_Find_NT	-1	Decimal	SINT	-1=Off, 1 .. 20 Number of Tags to Find, Expert Mode only: Def
Parameters.Configuration.Tries_Allowed_TA	2	Decimal	SINT	Defines the number of access attempts during the execution c
Parameters.Configuration.Expected_Number_Of_Tags_QW	2	Decimal	SINT	0=exactly 1, 1=~2, 2=~4, 3=~8, 4=~16, Defines the number c
Parameters.Configuration.Tag_Lost_Smoothing_E5	5	Decimal	SINT	0 ..10, Defines the number of unsuccessful read/write attempt
Parameters.Configuration.Number_Of_Channels	4	Decimal	SINT	1 .. 50, Defines the number of transmission channels with whic

Figure 15: Input Representation change in order to use the PF_RFID_Easy_Short Add-on

2. The short data form does not include the EPC or length fields. The tag information has been shortened to Data, RSSI, and Power.

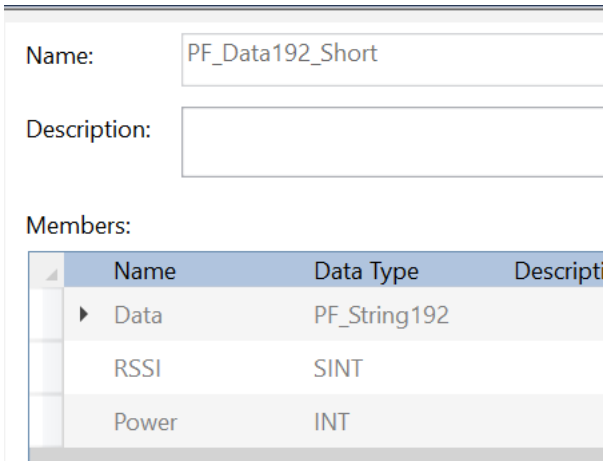


Figure 16: Short Data tag format

PF_RFID_Expert_Long

Triggers Reading, Writing, and Parameter commands when the reader is in “expert” mode.

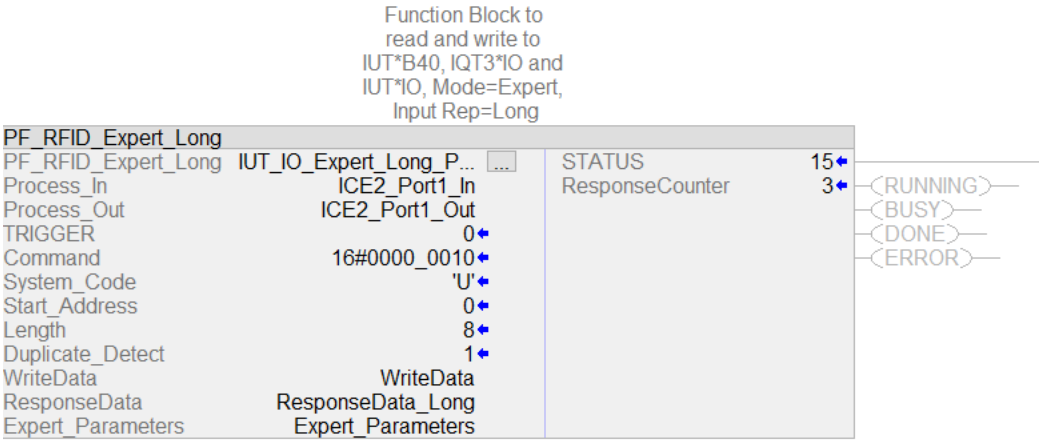


Figure 17: Expert mode function bloc

Both the Mode and Input representation must be set to “Expert” and “Long” respectively. The PF_IUT_IO_Parameters_xx Add-ons can be used to set all parameters.

Operation_Mode = 16#00
Input_Representation = 16#00

IUT_IO_Parameter.Configuration	{...}		PF_IUT_F191_Para...	
IUT_IO_Parameter.Configuration.Operation_Mode	16#00	Hex	SINT	0=Expert Mode, 16#80=Easy Mode, Defines the operatior
IUT_IO_Parameter.Configuration.Read_Task	{...}		PF_65_Read_Task	Parameters associated with reading in Easy mode
IUT_IO_Parameter.Configuration.Write_Task	{...}		PF_66_Write_Task	Parameters associated with writing in Easy mode
IUT_IO_Parameter.Configuration.Input_Representation	16#00	Hex	SINT	0=Long, 16#80=Short, Defines the content of the read da
IUT_IO_Parameter.Configuration.Transmission_Powers_PT	{...}	Decimal	INT[5]	Defines the set of different power levels, which the RFID re
IUT_IO_Parameter.Configuration.Number_Of_Tags_To_Find_NT	-1	Decimal	SINT	-1=Off, 1 .. 20 Number of Tags to Find, Expert Mode only:
IUT_IO_Parameter.Configuration.Tries_Allowed_TA	2	Decimal	SINT	Defines the number of access attempts during the executi
IUT_IO_Parameter.Configuration.Expected_Number_Of_Tags_QW	2	Decimal	SINT	0=exactly 1, 1=~2, 2=~4, 3=~8, 4=~16, Defines the numb
IUT_IO_Parameter.Configuration.Tag_Lost_Smoothing_E5	5	Decimal	SINT	0 ..10, Defines the number of unsuccessful read/write atter
IUT_IO_Parameter.Configuration.Number_Of_Channels	4	Decimal	SINT	1 .. 50, Defines the number of transmission channels with

Figure 18: Operating mode parameter set to Expert mode

Input Parameters

TRIGGER – In the Off-To-On transition a new command that is located in *Command* is executed

Command – The Command to be executed. See below table for options.

Single Commands – Executes one time on Off-To-On transition

Enhanced Commands – Continuously executes

Command Code(hex)	Function
Read/Write Commands	
01	Single Read EPC + TID
1D	Continuous Read EPC + TID
0A	Single Read EPC
71	Continuous Read EPC
CE	Single Write EPC
10	Single Read EPC + User Data (Byte length must be multiple of 4)
49	Single Read EPC + User Data (Byte length must be multiple of 2)
19	Enhanced Read EPC + User Data (Byte length must be multiple of 4)
4B	Enhanced Read EPC + User Data (Byte length must be multiple of 2)
40	Single Write User Data (Byte length must be multiple of 4)
4A	Single Write User Data (Byte length must be multiple of 2)
1A	Enhanced Write User Data (Byte length must be multiple of 4)
4C	Enhanced Write User Data (Byte length must be multiple of 2)
System Commands	
02	Quit Enhanced command
03	Read Version
5244BF	RD, Reset all UHF Parameters to Default
Read UHF Parameters	
1BE	RC, Read Region Code
2BE	AP, Read Antenna Polarization
2BE	AP, Write Antenna Polarization (Only L supported)
3BE	PT, Read Power
3BF	PT, Write Power (3,4,5,6,8,10,13,15,20,25,30,40,50,60,80,100,125,**150**) mW [power values also depend on the reader used]

4BE	CD, Read Channels
4BF	CD, Write Channels for (Depends on the reader model)
5BE	NC, Read Number of Channels
5BF	NC, Write Number of Channels (1, 2, 3, **4**…50) depends on Model
6BE	TA, Read Number of Attempts
6BF	TA, Write Number of Attempts (1, **2**, 3…10)
7BE	E5, Read Enhanced Status 5
7BF	E5, Write Enhanced Status 5 (1,2,3,4,**5**…10)
8BE	MB, Read Memory Bank to use with Read/Write commands
8BF	MB, Write Memory Bank to use with Read/Write commands (0, 1, 2, **3**)
ABE	QW, Read Q Value
ABF	QW, Write Q Value (0,1,**2**,3,4)
BBE	NT, Read Cancelation Criteria
BBF	NT, Write Cancellation Criteria (1,2…**255**)
DBE	MF, Read Measured Reflections
Filter Commands	
CB	Write Filter Mode (**0**,1,2)
10BE	Read Filter String 1
0CA	Write Filter String 1
11BE	Read Filter String 2
1CA	Write Filter String 2
12BE	Read Filter String 2
2CA	Write Filter String 2

System_Code – The RFID system being used

'U' = UHF RFID system, IUT-B40 or IUT-…-IO

'Q' = HF RFID system IQT3-...-IO

Start_Address – The first byte to start reading or writing. Only used for Read/Write commands

0 = First byte (default)

x = Must be a multiple of two or four bytes depending on the command

Length – The Number of bytes to read or write. Only used for Read/Write commands.

x = Must be a multiple of two or four bytes depending on the command (default 8)

Duplicate_Detect – Searches for duplicate tags in the ***ResponseData*** array. Not used for Write commands.

0 = Adds RFID tag to the ***ResponseData*** array at location [0] and shifts all other tags down in array

1 = Searches the ***ResponseData*** array first. If the RFID tag has not been read before it will be added in the ***ResponseData*** array at location [0] and shifts all other tags down in array (default).

In/Out Parameters

ProcessIn – The input data of the B40 reader. See Figures 8, 9, and 10 for more information.

ProcessOut – The output data of the B40 reader. See Figures 8, 9, and 10 for more information.

WriteData – The string of data that is used when writing to tags. The datatype is a STRING variable of the type PF_String192.

ResponseData – This is the location where all read data and all responses to Write commands are placed. The array size will determine how many tags are stored. The array can be as small as 1 or as large as the PLC memory allows (See Figure 19 for an example for storing 10 tags).

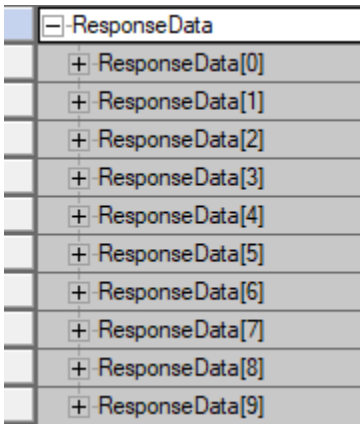


Figure19: Array size set to 10 to store up to 10 tags

If the array size is more than 1 the newest data is placed in the [0] position and all older data is moved down in the array (See Figure 20).

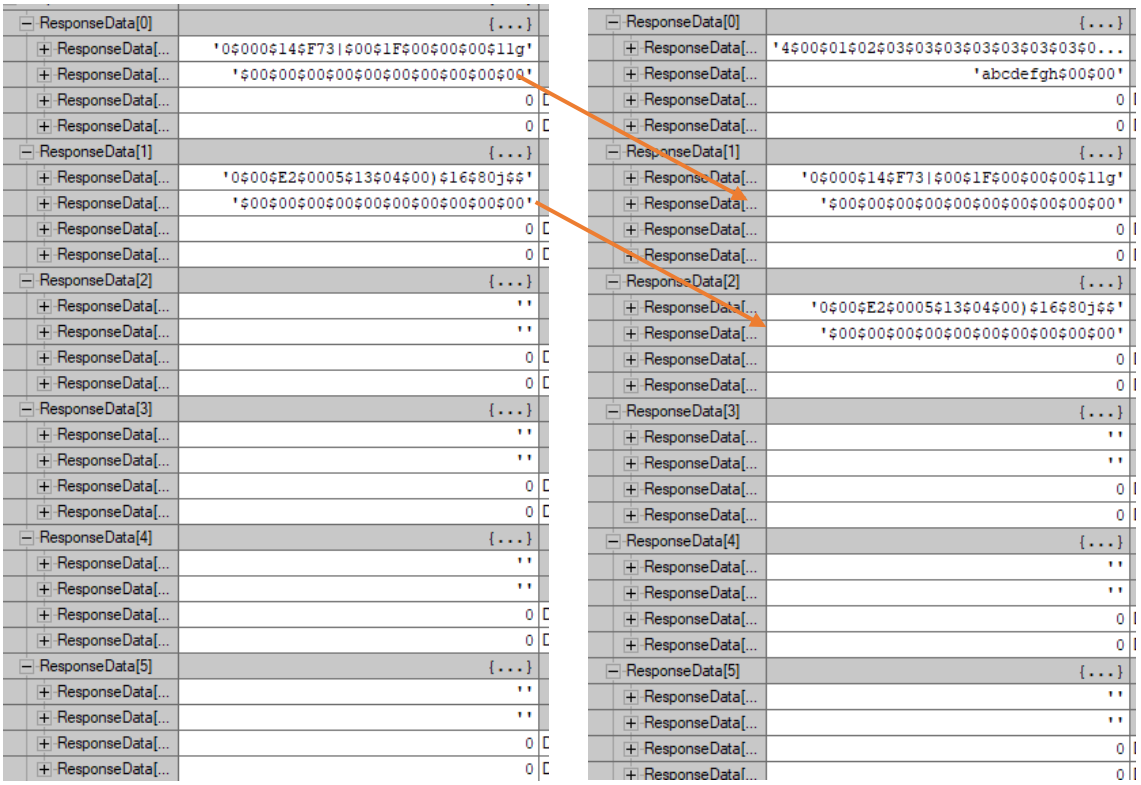


Figure 20: New tag added to position [0] and all older tags moved one index higher

The data for the tag contains up to four pieces of information including EPC, USER, RSSI and Power (See Figure 21).

EPC – The UII/EPC code, the EPC code will also include the PC (Protocol Control Word). This is the value searched when checking for duplicate tags. No checking is done when writing tags.

USER – The second set of data read from the tag. It could be the user data or the TID code. Not used when writing tags.

RSSI – The RSSI value of tag

100 – Tag is very close
0 – Tag is far away

Power – The power setting of the reader when the tag was read or written

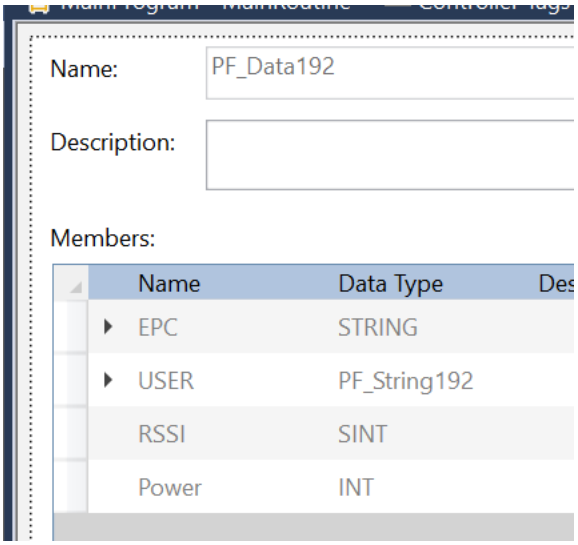


Figure 21: UDT for responses to reading and writing of tags

Expert_Parameters – All data used when reading and writing parameters. It is suggested that you read the parameter to be changed, adjust it as required for the application and write the new parameter to the reader. See Figure 22 below for all parameter options. Some of these parameters are only supported both other readers. See the Add-On Instruction Help text to see which parameters are supported by this reader.

Expert_Parameters	{...}		Expert_Parameters	
Expert_Parameters.RC_Region_Code_01	16#0002	Hex	INT	1BE/xxx, Region where
Expert_Parameters.AP_Polarization_02	'L'	ASCII	SINT	2BE/2BF, C=Combine
Expert_Parameters.PT_Transmit_Power_03	{...}	Decimal	INT[10]	3BE/3BF, Sets the trans
Expert_Parameters.CD_Transmission_Channels_04	{...}	Decimal	SINT[4]	4BE/4BF, Transmission
Expert_Parameters.NC_Number_of_Channels_05	4	Decimal	SINT	5BE/5BF, The number
Expert_Parameters.TA_Number_of_Attempts_06	2	Decimal	SINT	6BE/6BF, The number
Expert_Parameters.E5_Enhanced_Status5_07	5	Decimal	SINT	7BE/7BF, This is how m
Expert_Parameters.MB_Memory_Bank_08	3	Decimal	SINT	8BE/8BF, The memory
Expert_Parameters.QW_Q_Value_0A	2	Decimal	SINT	ABE/ABF, Used in ALO
Expert_Parameters.NT_Cancellation_Criteria_for_Search_0B	255	Decimal	INT	BBE/BBF, 255=not use
[illegible]	[illegible]	[illegible]	DINT	CBE/CBF, 0=no pause
Expert_Parameters.MF_Measure_Reflection_0D	{...}	Decimal	SINT[52]	DBE/xxx, The measure
Expert_Parameters.Filter_Mode	0	Decimal	SINT	xxx/CB, 0=Off, 1=Mod
Expert_Parameters.Filter1	{...}		PF_UHF_Filter	10BE/0CA
Expert_Parameters.Filter2	{...}		PF_UHF_Filter	11BE/1CA
Expert_Parameters.Filter3	{...}		PF_UHF_Filter	12BE/2CA
Expert_Parameters.Version	' (C) P+F IDENT rl 1833785 rl 20.06.23 '		STRING	3/xxx, Version

***Figure 22**: Parameters Read and Written to Read head*

Output parameters

Running – On while a Read or Write command is executing

Busy – A command is in the process of executing, and first answer is not back yet.

Done – A command has completed executing. Not used for Enhanced commands.

Error – An error occurred; see **Status** for details

Status – Detail on the execution of a command; see Figure 23 below

Status	Description	Cause
-1	Busy executing a command	-
-2	Timeout	Not in Expert Mode or reader not connected
0	Command executed correctly	-
5	No tag in sensing range	Put tag in front of reader, or tag does not meet filter criteria
11	Tag present, additional info	-
15	Single command complete	-
4	Bad command	Check parameters are within acceptable range
10	Multiple tags in field with same EPC	Reprogram EPC codes to be unique, ensure only one tag is in the field at a time
14	Buffer overflow	Slow down reading or writing, or too many tags in field

***Figure 23**: Possible Status Codes*

ResponseCounter – Counts up with every successful read or write attempt. Will not count duplicate tags if **Duplicate_Detect** is enabled and the same tag is read twice. Only resets if a read/write command or **Write Parameter** command is sent. **Read Parameter** commands can be executed while a read command is running so that the counter remains unchanged.

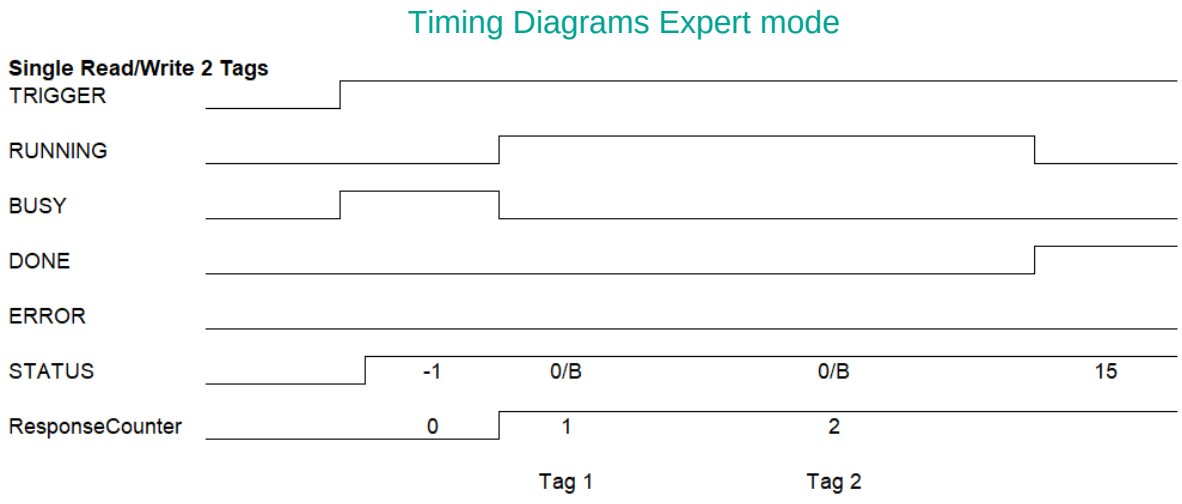


*Figure 24: Timing Diagram for **Single Read/Write 2 Tags***

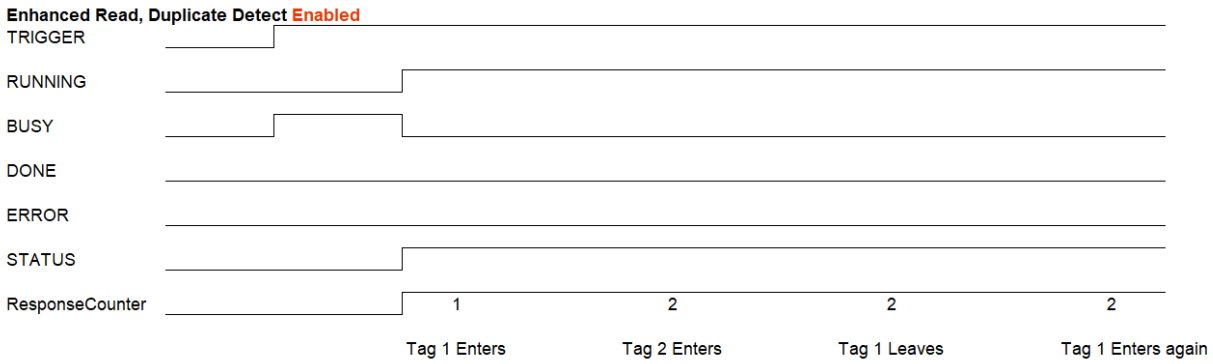


*Figure25: Timing Diagram for **Enhanced Read** with **Duplicate Detect Enabled***

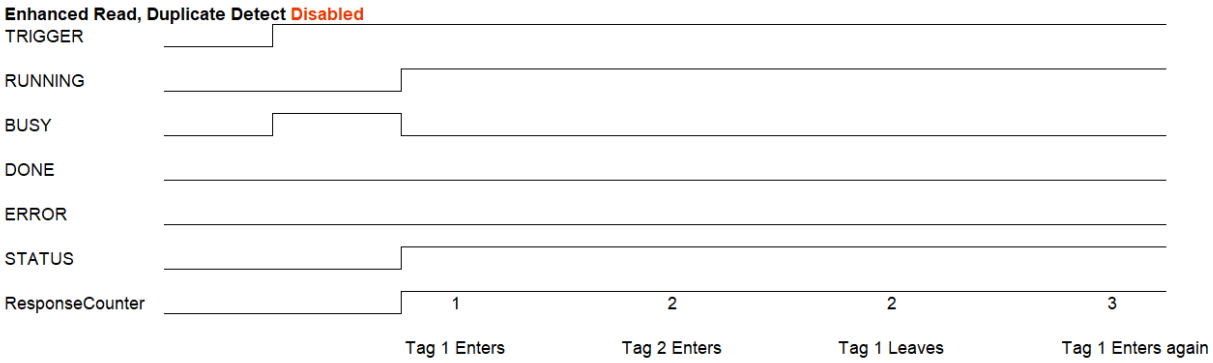


*Figure26: Timing Diagram for **Enhanced Read** with **Duplicate Detect Disabled***

Timing Diagrams Easy Mode

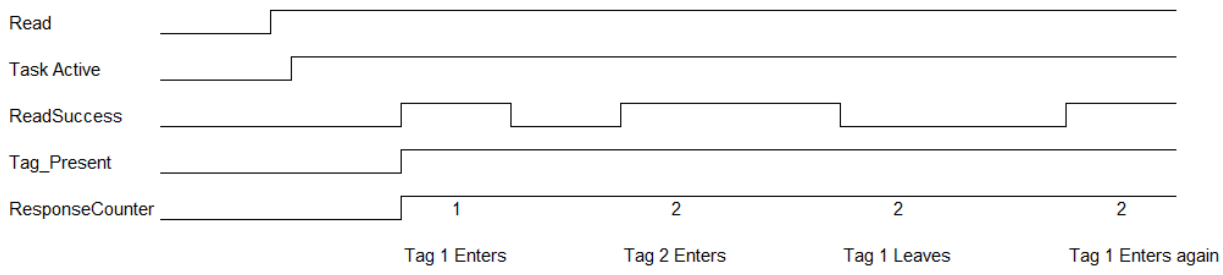


Figure 27: Timing Diagram for reading when Duplicate_Detect is enabled

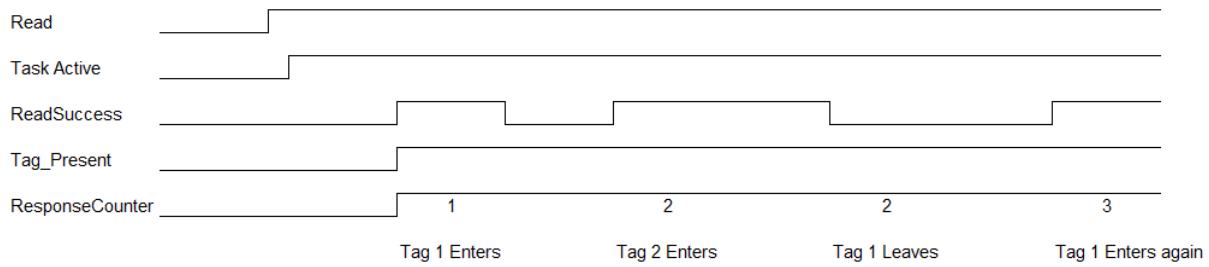


Figure 28: Timing Diagram for reading when Duplicate_Detect is disabled

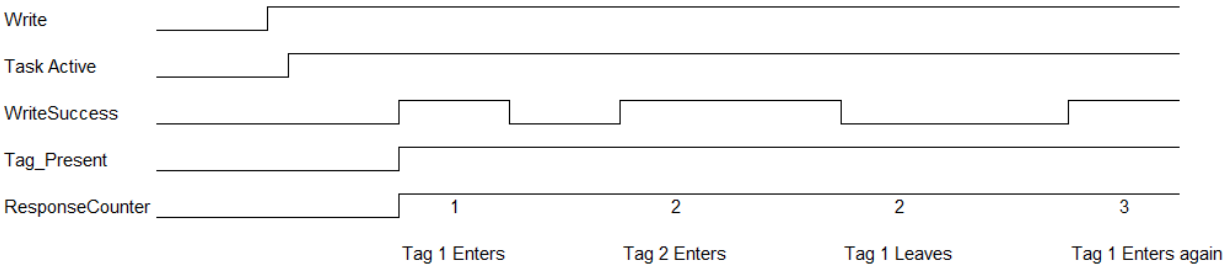


Figure 29: Timing Diagram for writing

Example

See videos and projects that are included