RFID JOURNAL VIRTUAL EVENTS

November 3, 2022

RFID in Manufacturing 2022

RFID in Manufacturing

Sponsored by

METAL CRAFT

ID MADE BETTER®

Planning Your Manufacturing Line Deployment

Doug Harvel

Planning Your Manufacturing Line Deployment

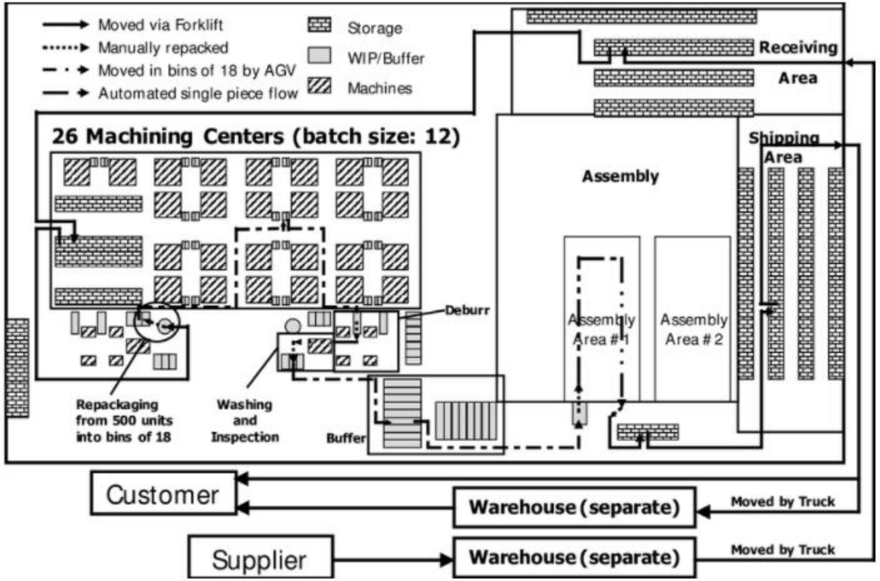
- What are the reasons for implementing?
- How do we implement RFID in Manufacturing?
- Who should we include in the planning?
- Where is the best place to implement?
- Timeline for implementing

What are the reasons for implementing?

- Implementing for customer compliance does NOT have an ROI

- What do you manufacture?

Where to inject RFID in your Flow



Who Should be Involved

- Business Owners
- Plant stake holders
- IT department
- Business partners

How do we implement RFID in Manufacturing?

- Will you be using preprinted/encoded or print and encode at the plant

- What method is being used for encoding?

- GS1 standards dictate that for a 96-bit RFID tag, the serial number in an Electronic Product Code (EPC) can be no more than 38 bits and that, therefore, when read as a decimal numeral, it must be less than or equal to 274,877,906,943. In simple terms, this 12-digit decimal number can be shown as XXX,XXX,XXX,XXX.
- In your serialization scheme, you can use the first two to four positions for the COO, with the first position designating who encoded the EPC. For example, you can make the first digit a 0 if the EPC was encoded internally, or a 1 if it was encoded by your RFID label vendor or service bureau.

Questions??

THANK YOU

6 Reasons to Use RFID in Manufacturing

Colynn Black, RFID Business Development Director

Metalcraft, Inc.

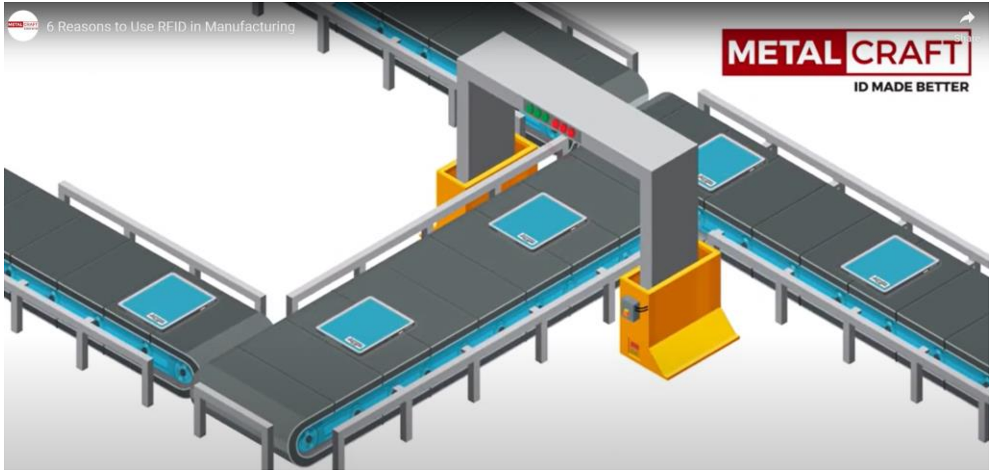
6 Reasons to Use RFID in Manufacturing

1. **Improve accuracy and reliability in your supply chain**
   - Impact of errors
     - Missed shipments
     - Deliveries shipped to wrong location
     - Depleting inventory
     - Lost product

6 Reasons to Use RFID in Manufacturing

2. Improve production line efficiency

- Streamline and segment processes
- Free up resources (equipment/people)



6 Reasons to Use RFID in Manufacturing

3. **Track equipment maintenance**
   - Increase uptime
   - Extend equipment life
   - Increase OTS

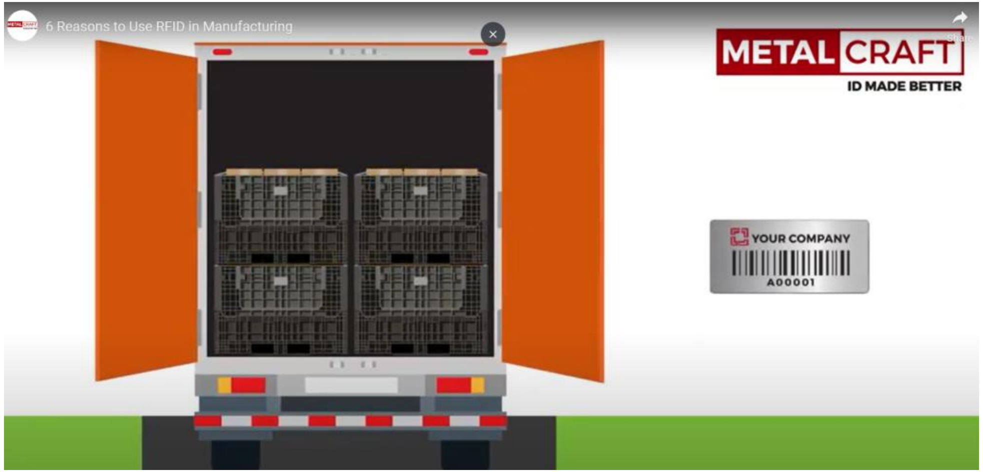
6 Reasons to Use RFID in Manufacturing

4. **Increase accuracy of inventory management**
   - Provides real time inventory visibility
   - Aids in production planning
   - Monitor and prevent shrinkage
   - Minimize labor costs
   - Better inventory forecasting

6 Reasons to Use RFID in Manufacturing

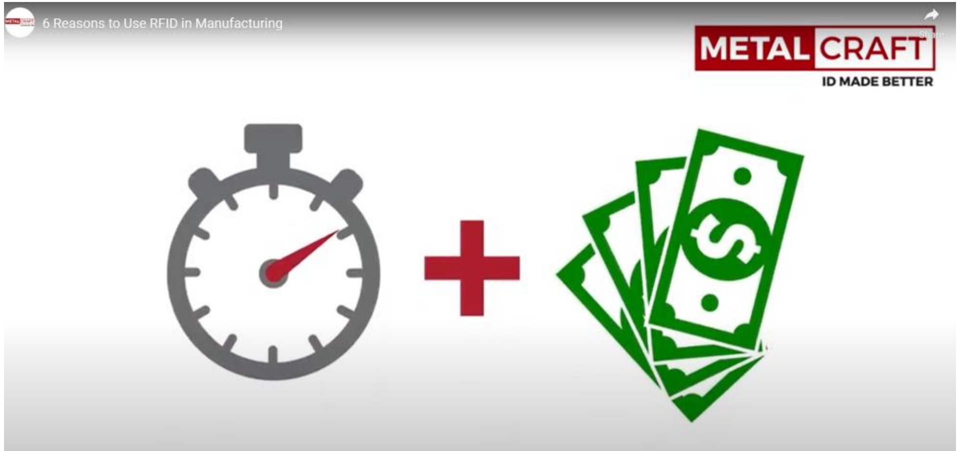
5. Maximize asset utilization

- – Ensure return of mobile assets
- – Record valuable information about assets



6 Reasons to Use RFID in Manufacturing

- **Improve overall ROI**
  - – Low barriers to entry
    - Infrastructure
    - Consumables
  - – Increased benefits


6 Reasons to Use RFID in Manufacturing

RFID in Manufacturing – Case Studies

- **Valley Chrome**
  - Background – leading manufacturer of chrome -plated aftermarket diesel truck parts like bumpers, cab panels, visors, etc.

  - Opportunity/Challenge – managing production process producing over 300 bumpers/day

RFID in Manufacturing – Case Studies

- **Valley Chrome**
  - Solution – automating process with Universal RFID tags – performance, durability and cost-effectiveness
  - Result – cut wasted time, improved efficiency of their processes and automated redundant task

RFID in Manufacturing – Case Studies

- **Hawk Technology**
  - Opportunity/Challenge – track parts from raw material through manufacturing and installation into the final product

RFID in Manufacturing – Case Studies

- **Hawk Technology**
  - Solution – Onsite Printable Universal Mini RFID Tags; survived rigorous tests (wash process, a paint process with oven curing and simulation of worst-case scenario in-field conditions
  - Result – time savings on the line and in the field

THANK YOU

RFID JOURNAL VIRTUAL EVENTS

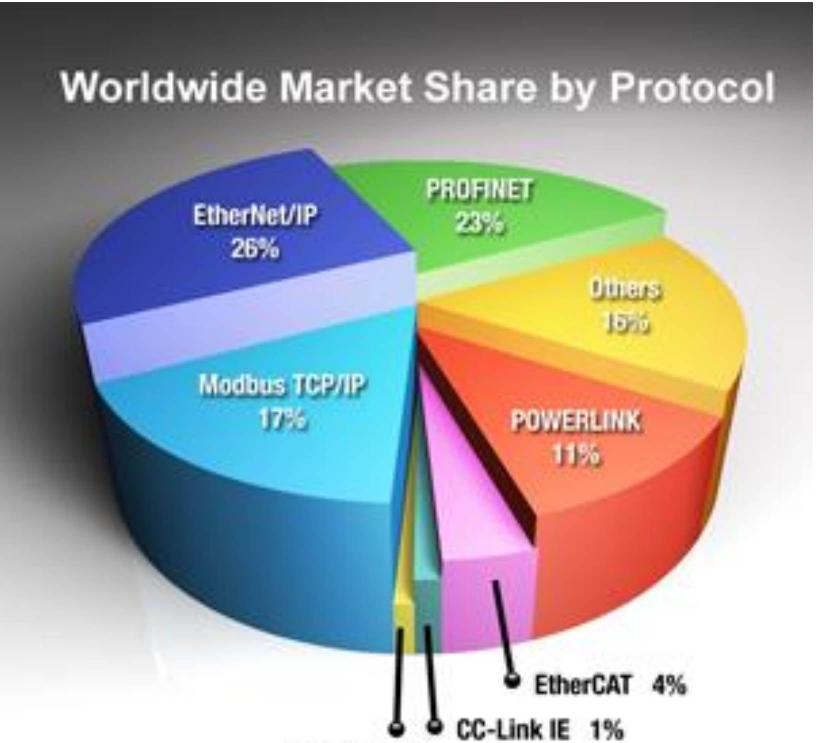
November 3, 2022

RFID in Manufacturing 2022

Integrating RFID With PLCs and Manufacturing Systems

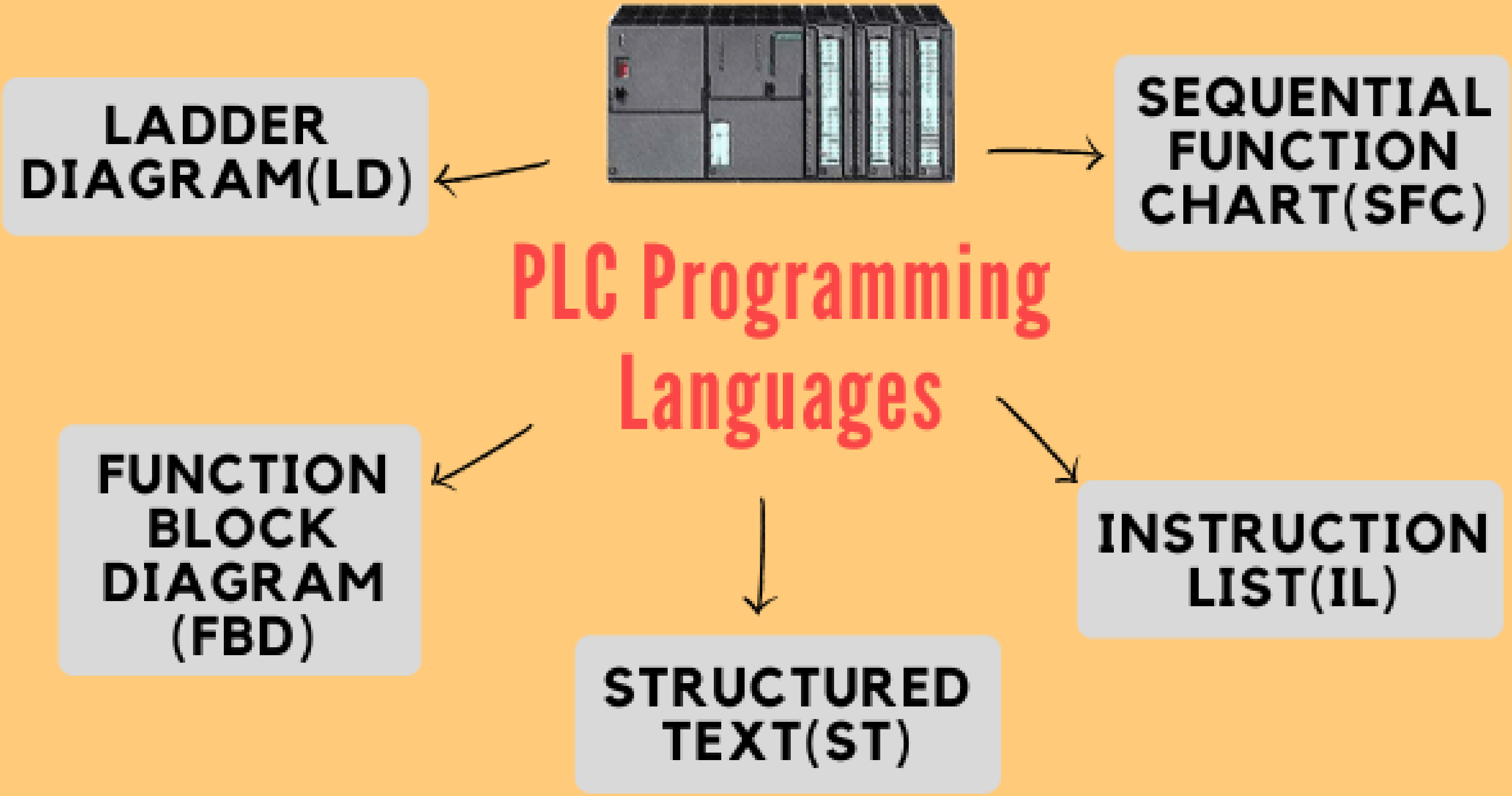
Kevin Berisso, Ph.D.
Director

THE UNIVERSITY OF MEMPHIS®
Automatic Identification Lab



Source: IMC Research

Protocol	Company	Region
Sinec H1	Siemens	Europe
Ethernet/IP	Allen Bradley (Rockwell Automation)	US
CC-Link	Mitsubishi Electric	Asia
Modbus	Schneider Electric (Modicon)	All over



PLC Programming Languages

LADDER DIAGRAM(LD)

SEQUENTIAL FUNCTION CHART(SFC)

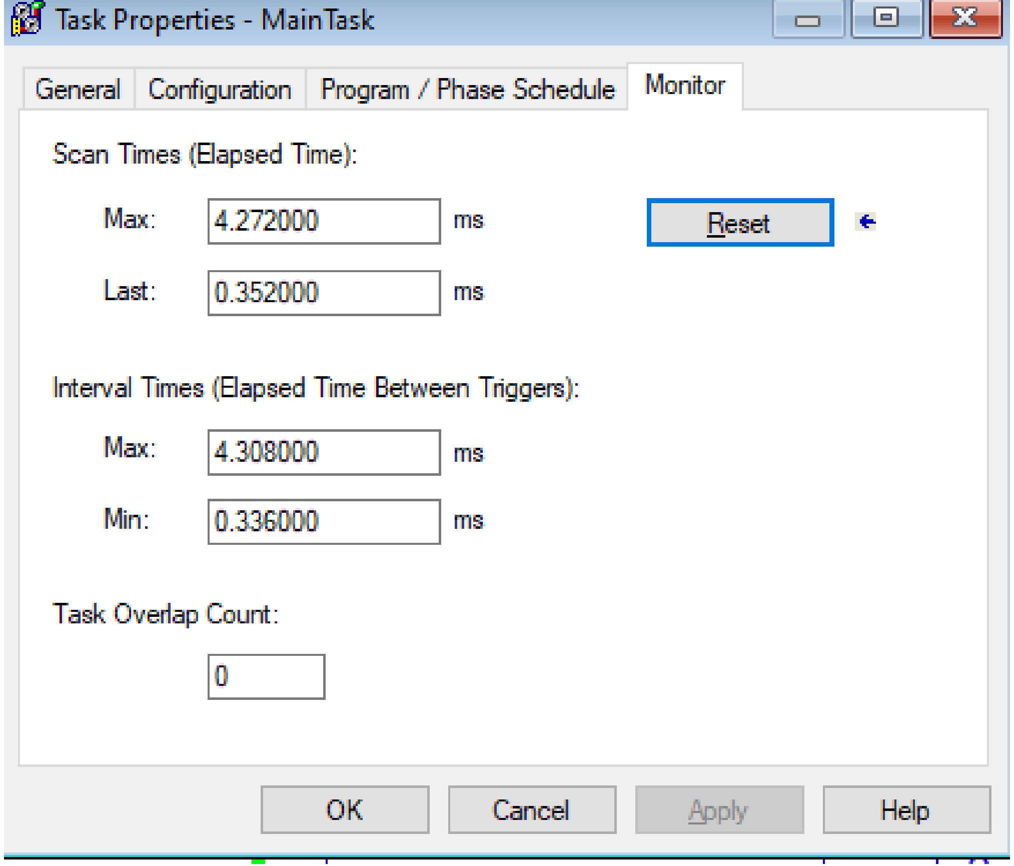
FUNCTION BLOCK DIAGRAM (FBD)

STRUCTURED TEXT(ST)

INSTRUCTION LIST(IL)

python™
Java
php
JavaScript
JS
C++
Ruby
Scala
node JS
C#

Parts per Minute	Parts per Second	ms per Part
60	1.0	1000
100	1.6	600
200	3.3	300
400	6.7	150
800	10.0	75
1000	16.7	60
6000	100	10



Scan Times (Elapsed Time):

Max: 4.272000 ms

Last: 0.352000 ms

Interval Times (Elapsed Time Between Triggers):

Max: 4.308000 ms

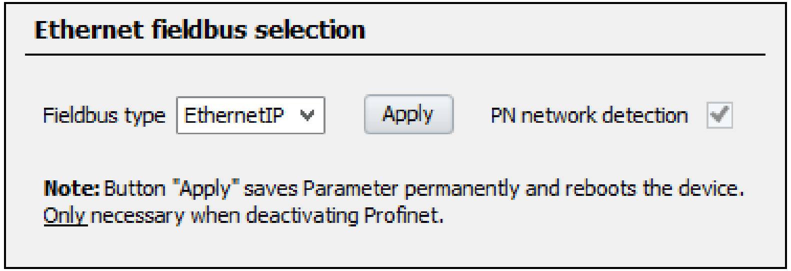
Min: 0.336000 ms

Task Overlap Count:

0

Logic	Main Program (µs)			Main Task (ms)		
	L23	L24	L306	L23	L24	L306
Empty Project	128	33	54	1.894	1.790	0.065
GSV only	290	103	64	2.304	1.750	0.074
FFL (100) + GSV	316	102	65	2.230	1.787	0.074

SICK RFU610

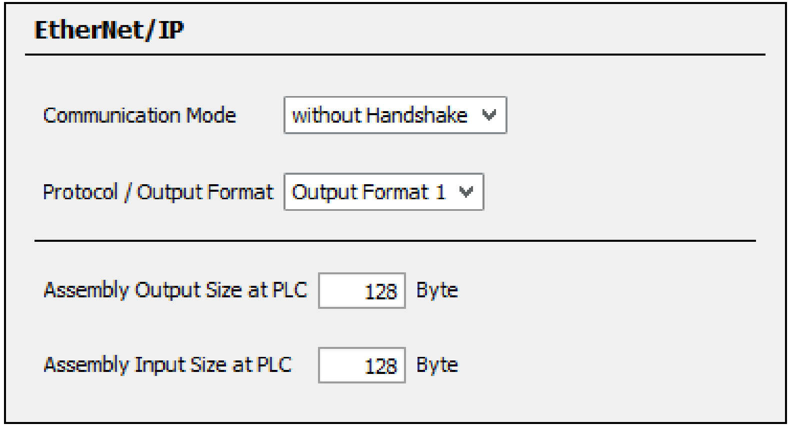


Ethernet fieldbus selection

Fieldbus type: EthernetIP

PN network detection ☑

Note: Button "Apply" saves Parameter permanently and reboots the device. Only necessary when deactivating Profinet.



EtherNet/IP

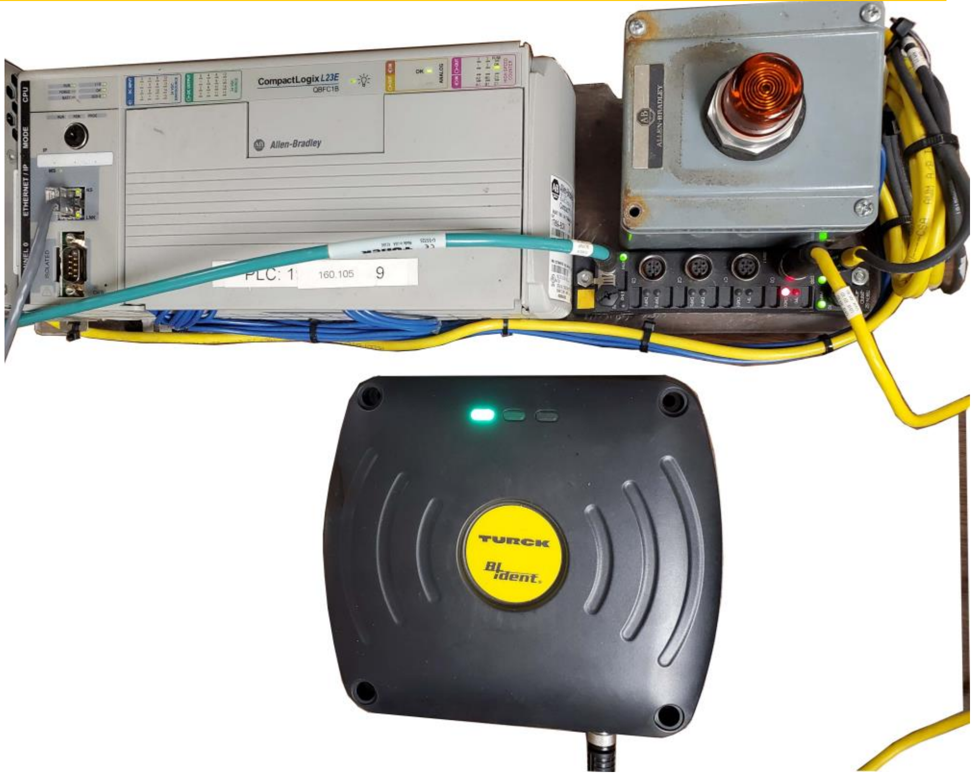
Communication Mode: without Handshake

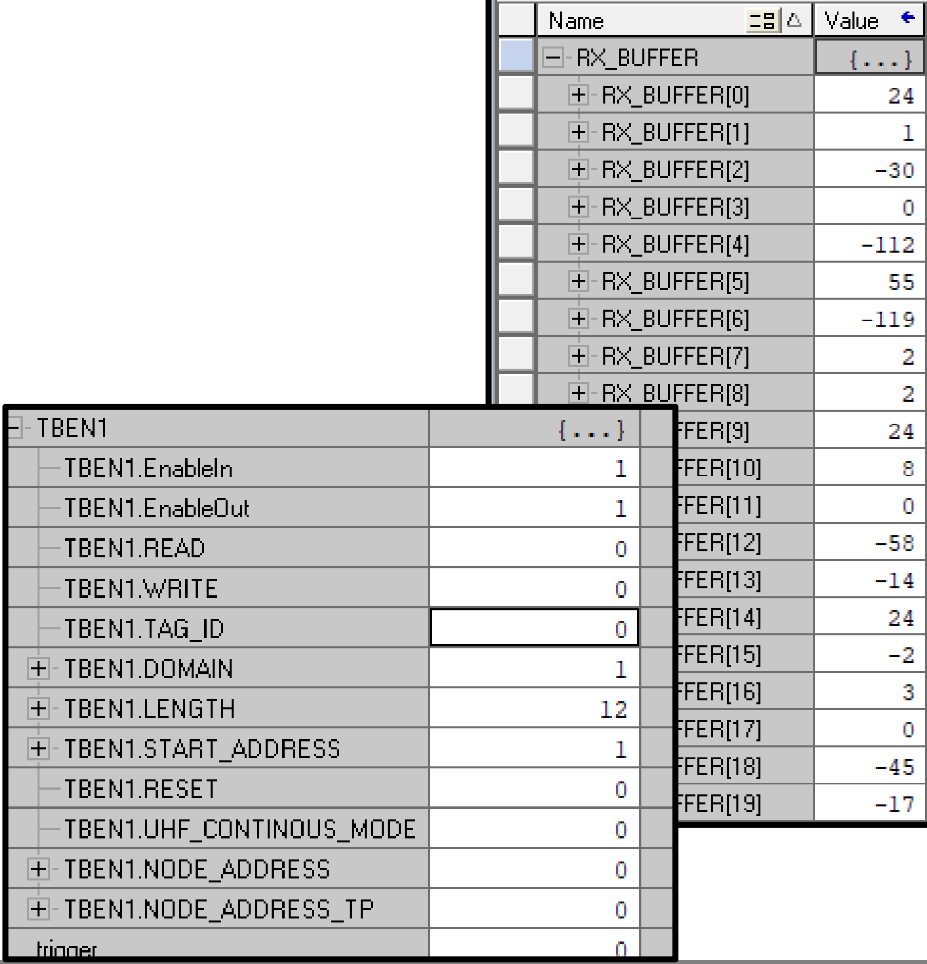
Protocol / Output Format: Output Format 1

Assembly Output Size at PLC: 128 Byte

Assembly Input Size at PLC: 128 Byte

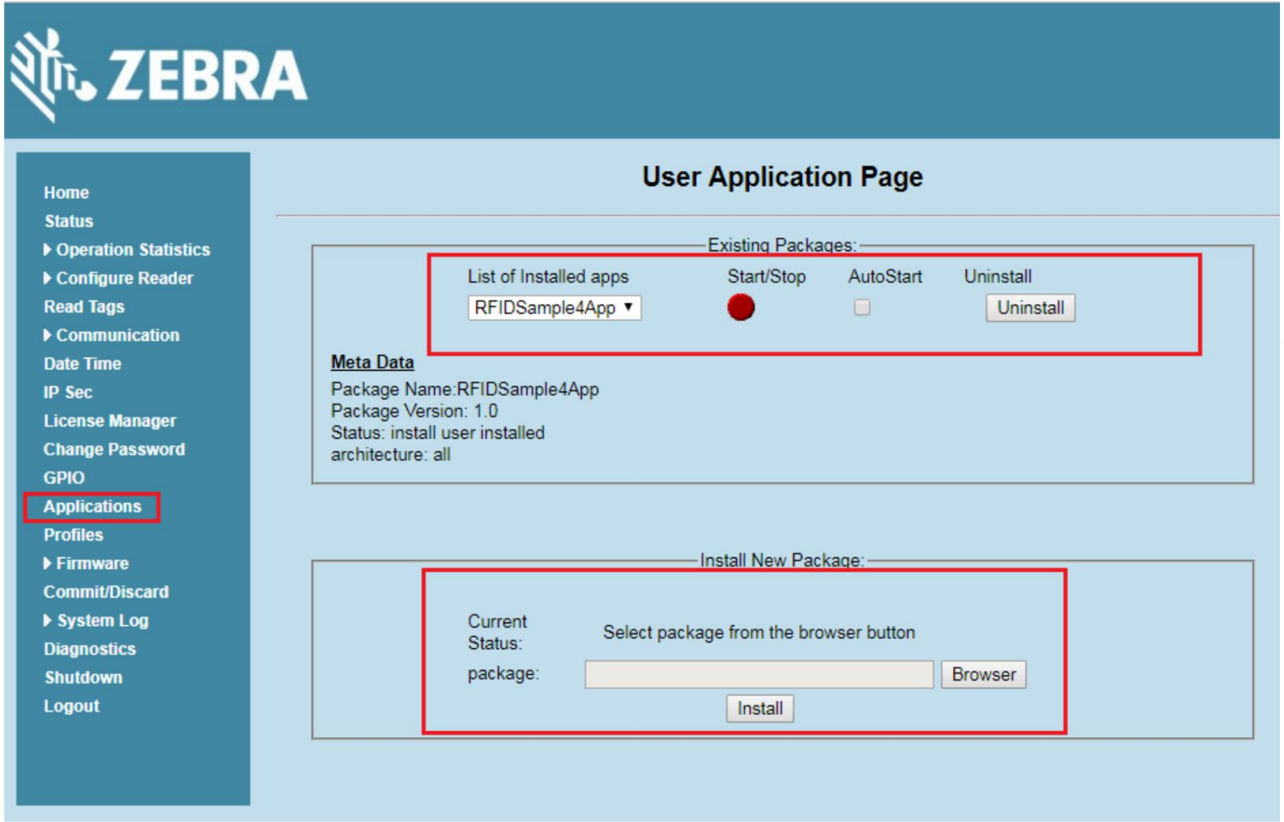
TURCK





Name	Value
RX_BUFFER	{...}
RX_BUFFER[0]	24
RX_BUFFER[1]	1
RX_BUFFER[2]	-30
RX_BUFFER[3]	0
RX_BUFFER[4]	-112
RX_BUFFER[5]	55
RX_BUFFER[6]	-119
RX_BUFFER[7]	2
RX_BUFFER[8]	2
FFER[9]	24
FFER[10]	8
FFER[11]	0
FFER[12]	-58
FFER[13]	-14
FFER[14]	24
FFER[15]	-2
FFER[16]	3
FFER[17]	0
FFER[18]	-45
FFER[19]	-17

TBEN1	{...}
TBEN1.EnableIn	1
TBEN1.EnableOut	1
TBEN1.READ	0
TBEN1.WRITE	0
TBEN1.TAG_ID	0
TBEN1.DOMAIN	1
TBEN1.LENGTH	12
TBEN1.START_ADDRESS	1
TBEN1.RESET	0
TBEN1.UHF_CONTINOUS_MODE	0
TBEN1.NODE_ADDRESS	0
TBEN1.NODE_ADDRESS_TP	0
trigger	0



User Application Page

Existing Packages:

List of Installed apps	Start/Stop	AutoStart	Uninstall
RFIDSample4App ▾	●	□	Uninstall

Meta Data

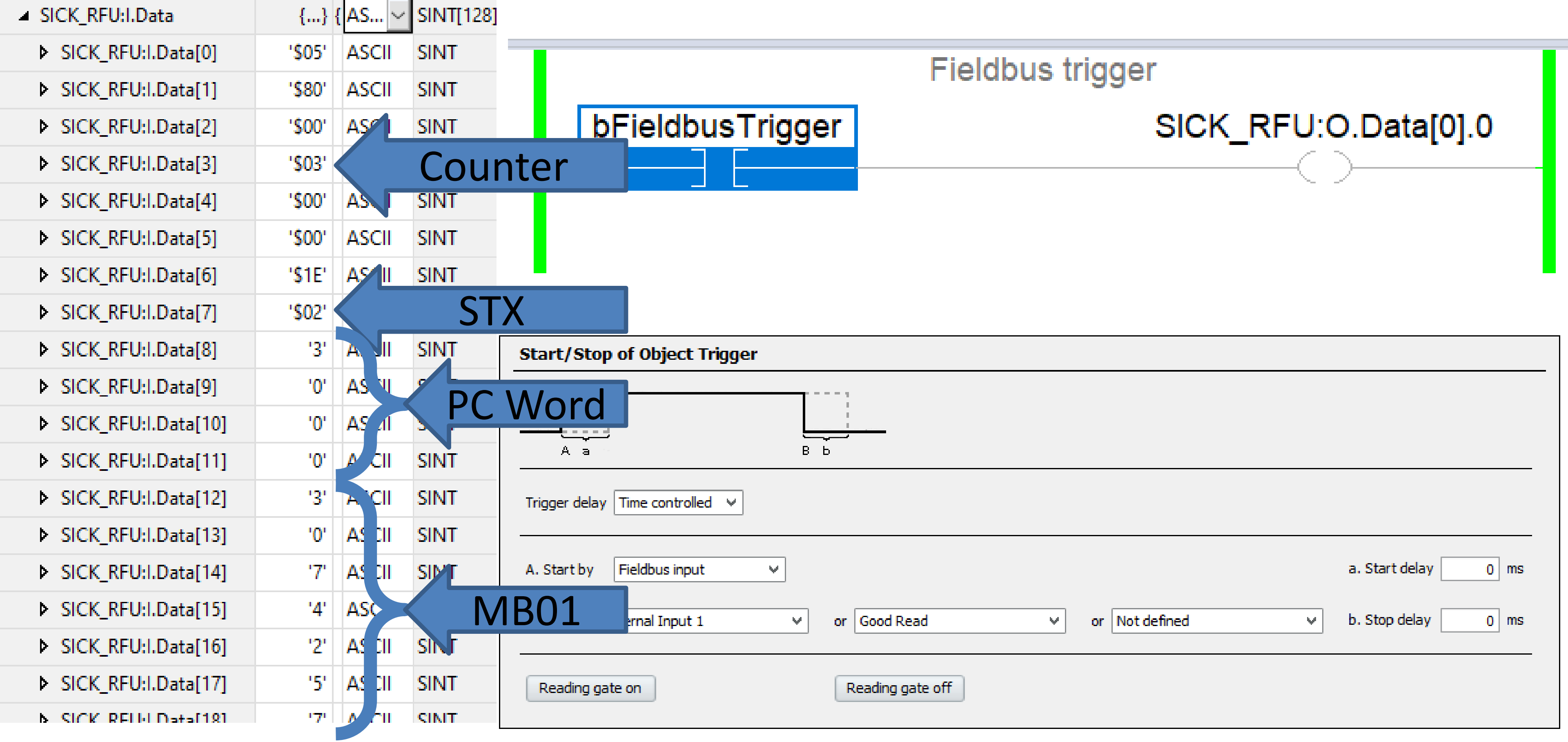
Package Name:RFIDSample4App
Package Version: 1.0
Status: install user installed
architecture: all

Install New Package:

Current Status: Select package from the browser button

package: [ ] Browser

Install



SICK_RFU:I.Data	{...}	{AS...	SINT[128]
SICK_RFU:I.Data[0]	'$05'	ASCII	SINT
SICK_RFU:I.Data[1]	'$80'	ASCII	SINT
SICK_RFU:I.Data[2]	'$00'	ASCII	SINT
SICK_RFU:I.Data[3]	'$03'		
SICK_RFU:I.Data[4]	'$00'	ASCII	SINT
SICK_RFU:I.Data[5]	'$00'	ASCII	SINT
SICK_RFU:I.Data[6]	'$1E'	ASCII	SINT
SICK_RFU:I.Data[7]	'$02'		
SICK_RFU:I.Data[8]	'3'	ASCII	SINT
SICK_RFU:I.Data[9]	'0'	ASCII	SINT
SICK_RFU:I.Data[10]	'0'	ASCII	SINT
SICK_RFU:I.Data[11]	'0'	ASCII	SINT
SICK_RFU:I.Data[12]	'3'	ASCII	SINT
SICK_RFU:I.Data[13]	'0'	ASCII	SINT
SICK_RFU:I.Data[14]	'7'	ASCII	SINT
SICK_RFU:I.Data[15]	'4'	ASCII	
SICK_RFU:I.Data[16]	'2'	ASCII	SINT
SICK_RFU:I.Data[17]	'5'	ASCII	SINT
SICK_RFU:I.Data[18]	'7'	ASCII	SINT

Fieldbus trigger

bFieldbusTrigger

SICK_RFU:O.Data[0].0

Start/Stop of Object Trigger

Trigger delay: Time controlled

A. Start by: Fieldbus input

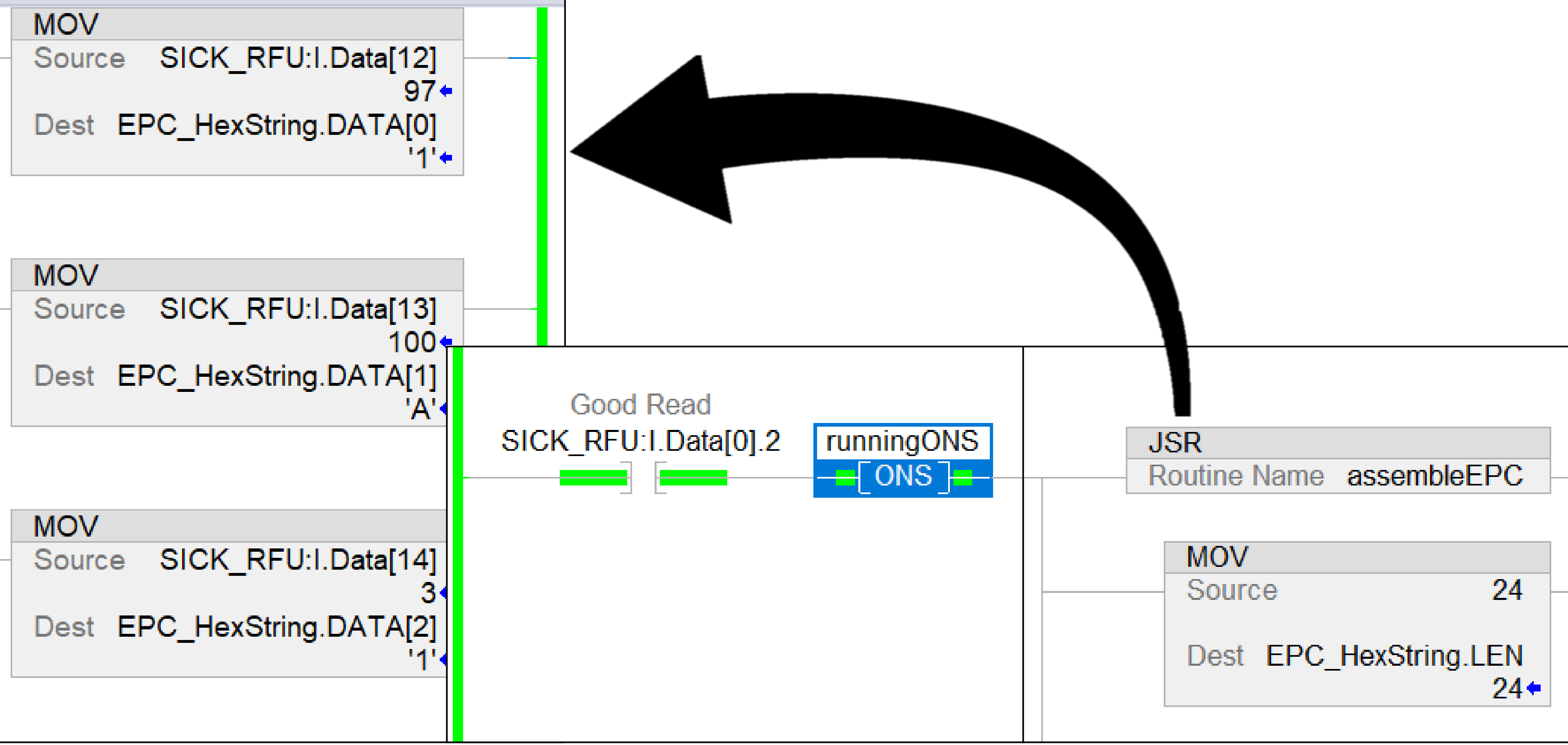
a. Start delay: 0 ms

rnal Input 1 or Good Read or Not defined

b. Stop delay: 0 ms

Reading gate on

Reading gate off



MOV
Source SICK_RFU:I.Data[12]
97
Dest EPC_HexString.DATA[0]
'1'

MOV
Source SICK_RFU:I.Data[13]
100
Dest EPC_HexString.DATA[1]
'A'

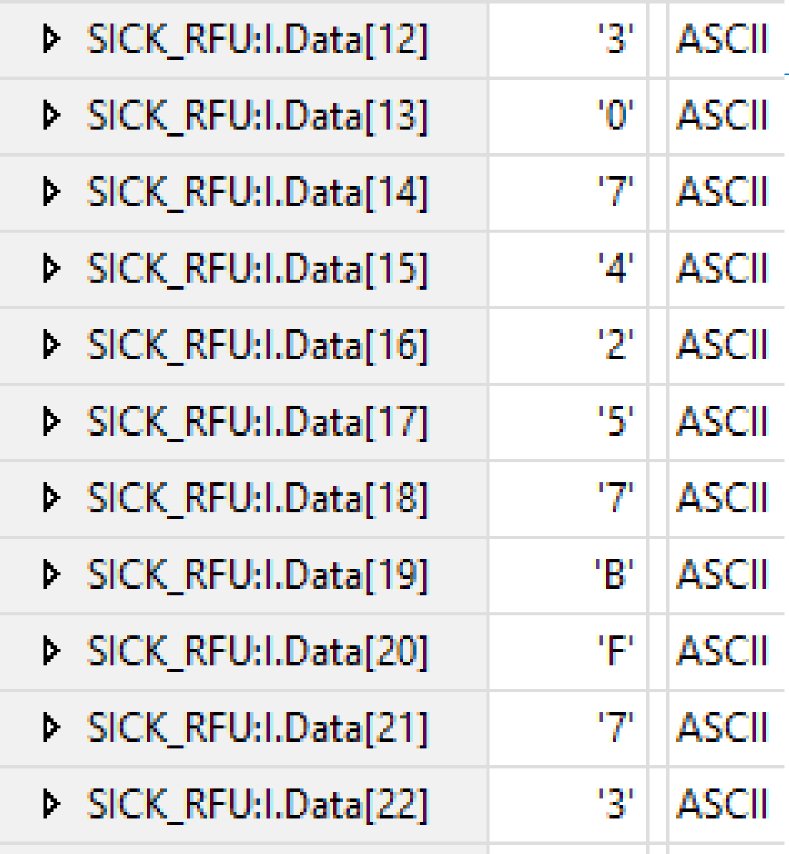
MOV
Source SICK_RFU:I.Data[14]
3
Dest EPC_HexString.DATA[2]
'1'

Good Read
SICK_RFU:I.Data[0].2

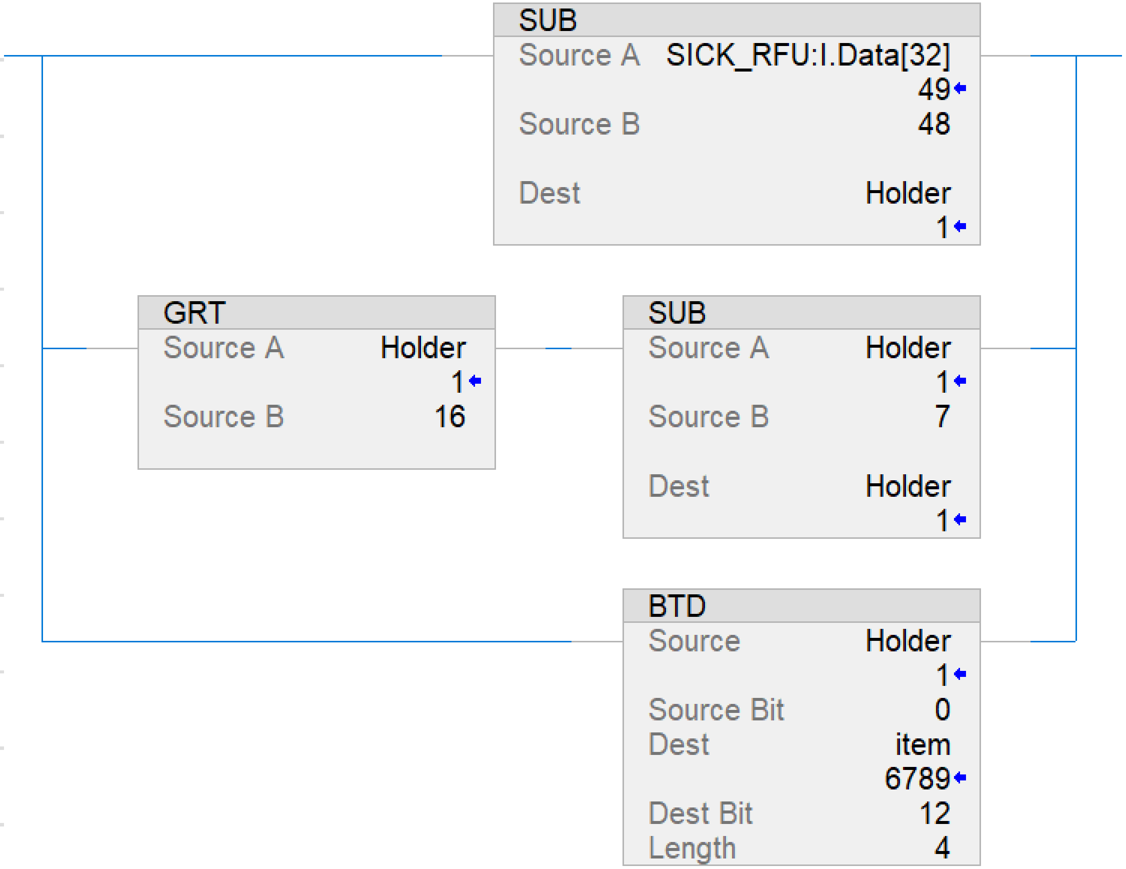
runningONS
[ONS]

JSR
Routine Name assembleEPC

MOV
Source 24
Dest EPC_HexString.LEN
24



SICK_RFU:I.Data[12]	'3'	ASCII
SICK_RFU:I.Data[13]	'0'	ASCII
SICK_RFU:I.Data[14]	'7'	ASCII
SICK_RFU:I.Data[15]	'4'	ASCII
SICK_RFU:I.Data[16]	'2'	ASCII
SICK_RFU:I.Data[17]	'5'	ASCII
SICK_RFU:I.Data[18]	'7'	ASCII
SICK_RFU:I.Data[19]	'B'	ASCII
SICK_RFU:I.Data[20]	'F'	ASCII
SICK_RFU:I.Data[21]	'7'	ASCII
SICK_RFU:I.Data[22]	'3'	ASCII

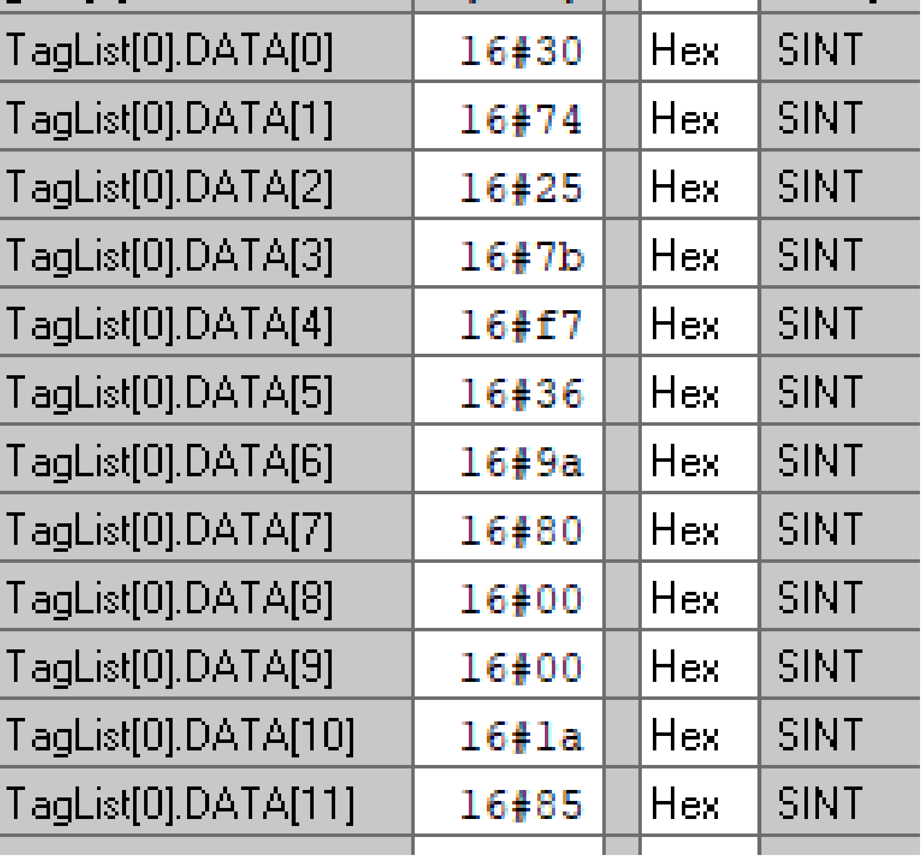


SUB
Source A SICK_RFU:I.Data[32]
49
Source B 48
Dest Holder
1

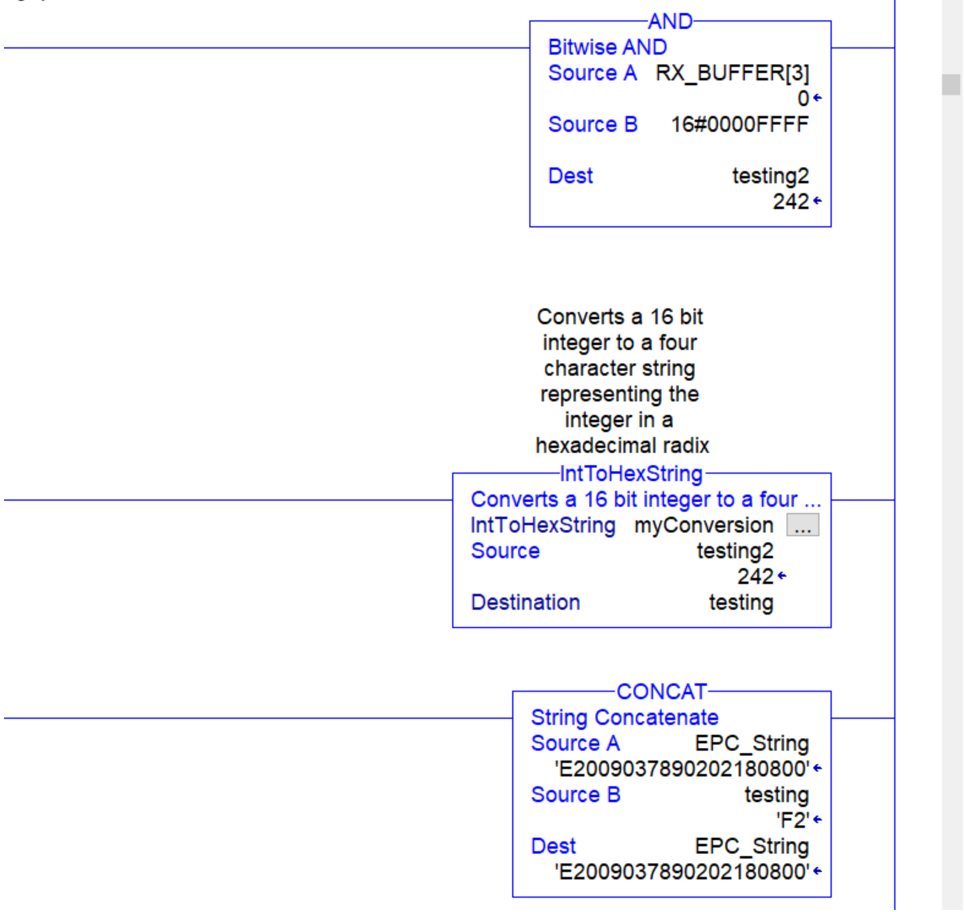
GRT
Source A Holder
1
Source B 16

SUB
Source A Holder
1
Source B 7
Dest Holder
1

BTD
Source Holder
1
Source Bit 0
Dest item
6789
Dest Bit 12
Length 4



TagList[0].DATA[0]	16#30	Hex	SINT
TagList[0].DATA[1]	16#74	Hex	SINT
TagList[0].DATA[2]	16#25	Hex	SINT
TagList[0].DATA[3]	16#7b	Hex	SINT
TagList[0].DATA[4]	16#f7	Hex	SINT
TagList[0].DATA[5]	16#36	Hex	SINT
TagList[0].DATA[6]	16#9a	Hex	SINT
TagList[0].DATA[7]	16#80	Hex	SINT
TagList[0].DATA[8]	16#00	Hex	SINT
TagList[0].DATA[9]	16#00	Hex	SINT
TagList[0].DATA[10]	16#1a	Hex	SINT
TagList[0].DATA[11]	16#85	Hex	SINT



```
AND
Bitwise AND
Source A  RX_BUFFER[3]
                     0←
Source B    16#0000FFFF

Dest          testing2
                  242←
```

Converts a 16 bit integer to a four character string representing the integer in a hexadecimal radix

```
IntToHexString
Converts a 16 bit integer to a four ...
IntToHexString  myConversion  ...
Source              testing2
                        242←
Destination          testing
```

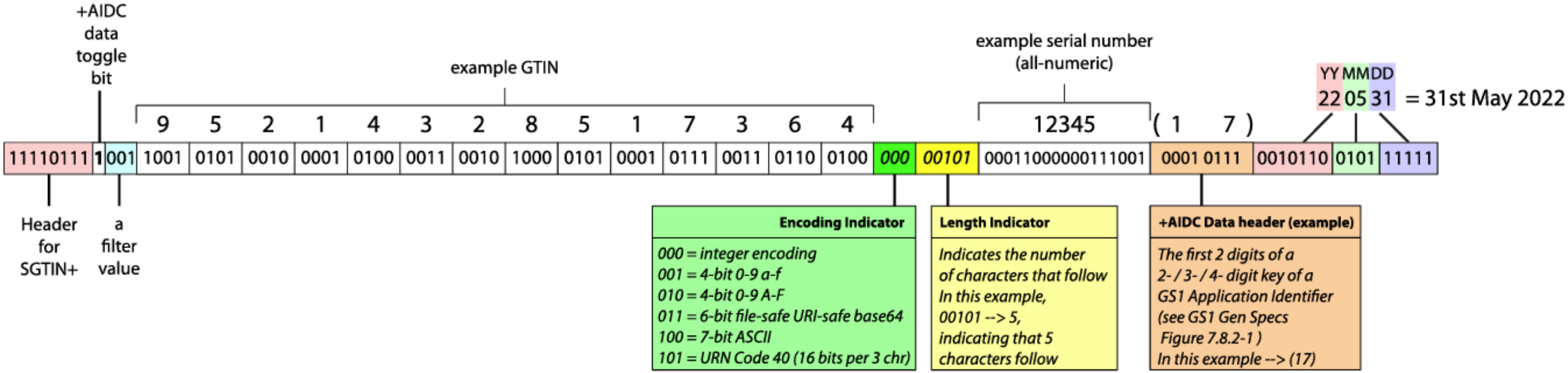
```
CONCAT
String Concatenate
Source A          EPC_String
  'E2009037890202180800'←
Source B             testing
                       'F2'←
Dest              EPC_String
  'E2009037890202180800'←
```

Logic	Main Program (µs)			Main Task (ms)		
	L23	L24	L306	L23	L24	L306
Empty Project	128	33	54	1.894	1.790	0.065
GSV only	290	103	64	2.304	1.750	0.074
FFL (100) + GSV	316	102	65	2.230	1.787	0.074
Extract SN	256	149	72	2.630	1.757	0.121

SGTIN+
DSGTIN+
SSCC+

The Global Language of Business

EPC Tag Data Standard (TDS) 2.0



																				YY	MM	DD
			9	5	2	1	4	3	2	8	5	1	7	3	6	4			12345	(1 7)	22	05	31
11110111	1	001	1001	0101	0010	0001	0100	0011	0010	1000	0101	0001	0111	0011	0110	0100	*000*	*00101*	00011000000111001	0001 0111	0010110	0101	11111

Encoding Indicator

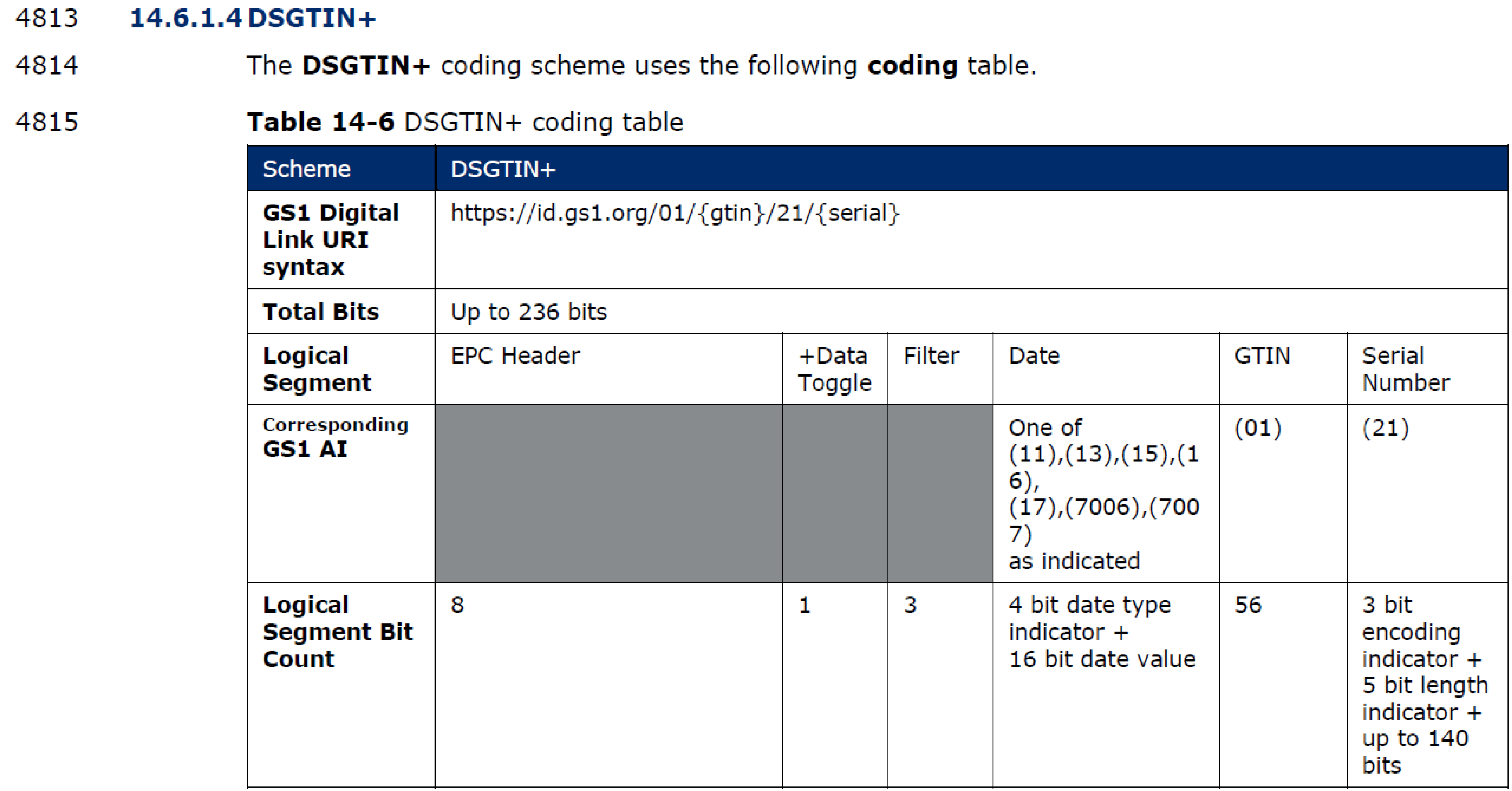
000 = integer encoding
001 = 4-bit 0-9 a-f
010 = 4-bit 0-9 A-F
011 = 6-bit file-safe URI-safe base64
100 = 7-bit ASCII
101 = URN Code 40 (16 bits per 3 chr)

Length Indicator

Indicates the number of characters that follow In this example, 00101 --> 5, indicating that 5 characters follow

+AIDC Data header (example)

The first 2 digits of a 2-/3-/4- digit key of a GS1 Application Identifier (see GS1 Gen Specs Figure 7.8.2-1) In this example --> (17)

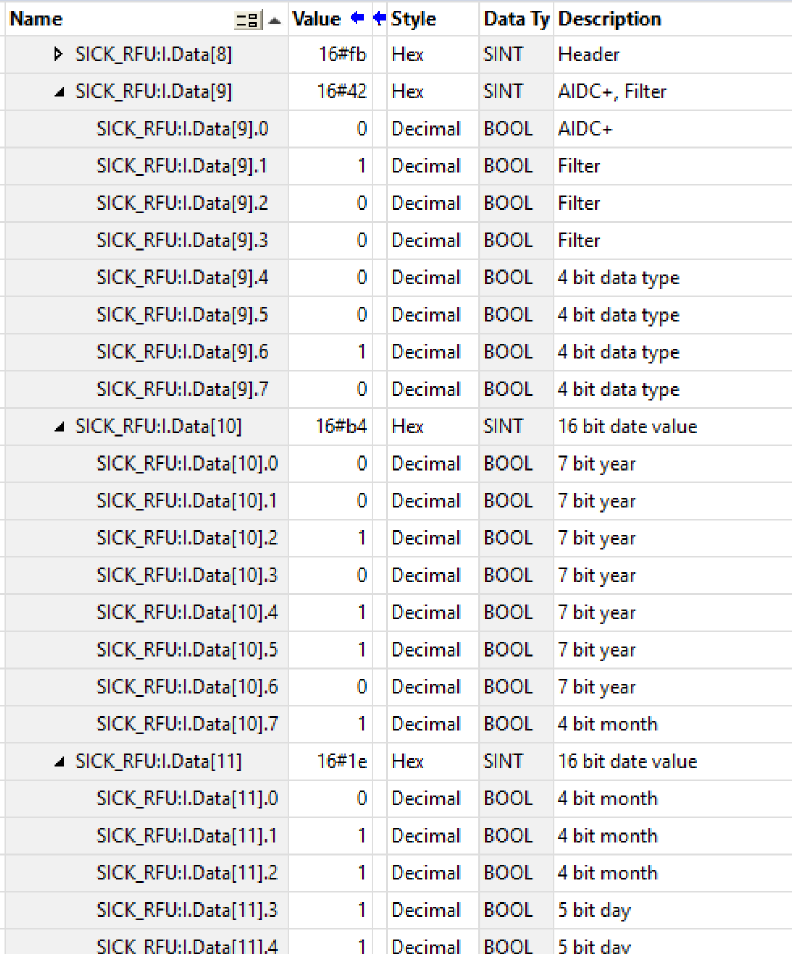


14.6.1.4 DSGTIN+

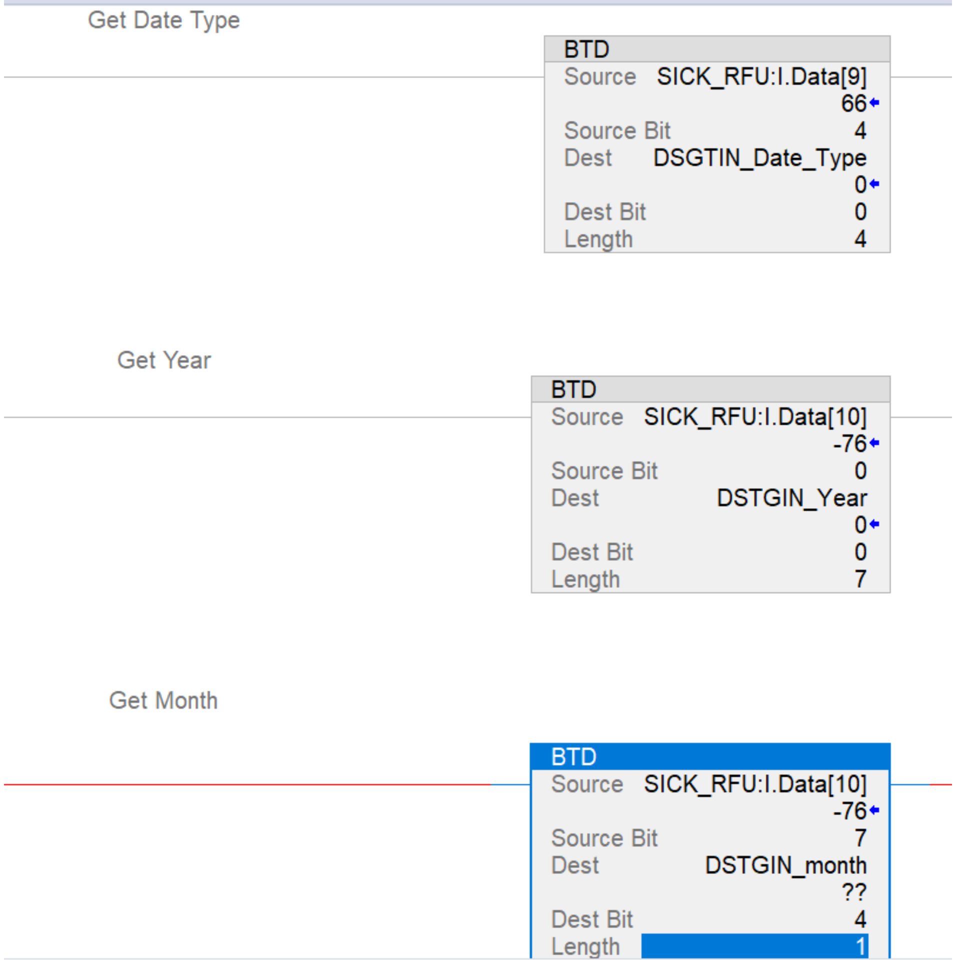
The **DSGTIN+** coding scheme uses the following **coding** table.

Table 14-6 DSGTIN+ coding table

Scheme	DSGTIN+					
GS1 Digital Link URI syntax	https://id.gs1.org/01/{gtin}/21/{serial}					
Total Bits	Up to 236 bits					
Logical Segment	EPC Header	+Data Toggle	Filter	Date	GTIN	Serial Number
Corresponding GS1 AI				One of (11),(13),(15),(16), (17),(7006),(7007) as indicated	(01)	(21)
Logical Segment Bit Count	8	1	3	4 bit date type indicator + 16 bit date value	56	3 bit encoding indicator + 5 bit length indicator + up to 140 bits



Name	Value	Style	Data Ty	Description
▷ SICK_RFU:I.Data[8]	16#fb	Hex	SINT	Header
◢ SICK_RFU:I.Data[9]	16#42	Hex	SINT	AIDC+, Filter
SICK_RFU:I.Data[9].0	0	Decimal	BOOL	AIDC+
SICK_RFU:I.Data[9].1	1	Decimal	BOOL	Filter
SICK_RFU:I.Data[9].2	0	Decimal	BOOL	Filter
SICK_RFU:I.Data[9].3	0	Decimal	BOOL	Filter
SICK_RFU:I.Data[9].4	0	Decimal	BOOL	4 bit data type
SICK_RFU:I.Data[9].5	0	Decimal	BOOL	4 bit data type
SICK_RFU:I.Data[9].6	1	Decimal	BOOL	4 bit data type
SICK_RFU:I.Data[9].7	0	Decimal	BOOL	4 bit data type
◢ SICK_RFU:I.Data[10]	16#b4	Hex	SINT	16 bit date value
SICK_RFU:I.Data[10].0	0	Decimal	BOOL	7 bit year
SICK_RFU:I.Data[10].1	0	Decimal	BOOL	7 bit year
SICK_RFU:I.Data[10].2	1	Decimal	BOOL	7 bit year
SICK_RFU:I.Data[10].3	0	Decimal	BOOL	7 bit year
SICK_RFU:I.Data[10].4	1	Decimal	BOOL	7 bit year
SICK_RFU:I.Data[10].5	1	Decimal	BOOL	7 bit year
SICK_RFU:I.Data[10].6	0	Decimal	BOOL	7 bit year
SICK_RFU:I.Data[10].7	1	Decimal	BOOL	4 bit month
◢ SICK_RFU:I.Data[11]	16#1e	Hex	SINT	16 bit date value
SICK_RFU:I.Data[11].0	0	Decimal	BOOL	4 bit month
SICK_RFU:I.Data[11].1	1	Decimal	BOOL	4 bit month
SICK_RFU:I.Data[11].2	1	Decimal	BOOL	4 bit month
SICK_RFU:I.Data[11].3	1	Decimal	BOOL	5 bit day
SICK_RFU:I.Data[11].4	1	Decimal	BOOL	5 bit day



Get Date Type

BTD	
Source	SICK_RFU:I.Data[9]
	66
Source Bit	4
Dest	DSGTIN_Date_Type
	0
Dest Bit	0
Length	4

Get Year

BTD	
Source	SICK_RFU:I.Data[10]
	-76
Source Bit	0
Dest	DSTGIN_Year
	0
Dest Bit	0
Length	7

Get Month

BTD	
Source	SICK_RFU:I.Data[10]
	-76
Source Bit	7
Dest	DSTGIN_month
	??
Dest Bit	4
Length	1

Questions?

Discussion?

Thank You

Kevin Berisso
University of Memphis
Automatic Identification
Lab
kberisso@memphis.edu

RFID JOURNAL VIRTUAL EVENTS

RFID in Manufacturing

Sponsored by

METAL CRAFT

ID MADE BETTER®

SAVE THE DATE

May 9-11, 2023 | Orange County Convention Center | Orlando, FL

RFID JOURNAL LIVE!