

# QueueLib, 1.0 project documentation

TK Automation

2023-02-19T23:10:57.3402784

# Contents

<b>1</b>	<b>Project information</b>	<b>8</b>
1.1	List of namespaces . . . . .	9
1.2	Metrics . . . . .	9
<b>2</b>	<b>Documentation for namespace Global</b>	<b>10</b>
2.1	Namespace index . . . . .	10
2.2	Metrics . . . . .	15
2.3	Data types . . . . .	16
2.3.1	dtFIFO_P_Element_Int (struct) . . . . .	16
2.3.2	dtFIFO_P_Element_Bool (struct) . . . . .	17
2.3.3	dtFIFO_P_Element_Byte (struct) . . . . .	18
2.3.4	dtFIFO_P_Element_Word (struct) . . . . .	19
2.3.5	dtFIFO_P_Element_Dword (struct) . . . . .	20
2.3.6	dtFIFO_P_Element_Lword (struct) . . . . .	21
2.3.7	dtFIFO_P_Element_Sint (struct) . . . . .	22
2.3.8	dtFIFO_P_Element_Usint (struct) . . . . .	23
2.3.9	dtFIFO_P_Element_Uint (struct) . . . . .	24
2.3.10	dtFIFO_P_Element_Dint (struct) . . . . .	25
2.3.11	dtFIFO_P_Element_Udint (struct) . . . . .	26
2.3.12	dtFIFO_P_Element_Lint (struct) . . . . .	27
2.3.13	dtFIFO_P_Element_Ulint (struct) . . . . .	28
2.3.14	dtFIFO_P_Element_Real (struct) . . . . .	29
2.3.15	dtFIFO_P_Element_Lreal (struct) . . . . .	30
2.3.16	dtFIFO_P_Element_String (struct) . . . . .	31
2.3.17	dtFIFO_P_Element_Wstring (struct) . . . . .	32
2.3.18	dtFIFO_P_Element_Time (struct) . . . . .	33
2.3.19	dtFIFO_P_Element_Ltime (struct) . . . . .	34
2.3.20	dtFIFO_P_Element_Date (struct) . . . . .	35
2.3.21	dtFIFO_P_Element_Ldate (struct) . . . . .	36
2.3.22	dtFIFO_P_Element_Dt (struct) . . . . .	37
2.3.23	dtFIFO_P_Element_Ldt (struct) . . . . .	38
2.3.24	dtFIFO_P_Element_Tod (struct) . . . . .	39
2.3.25	dtFIFO_P_Element_Ltod (struct) . . . . .	40
2.4	Function blocks . . . . .	41
2.4.1	FIFO_S_Int . . . . .	41
2.4.1.1	Methods . . . . .	41
2.4.2	FIFO_C_Int . . . . .	43
2.4.2.1	Methods . . . . .	43
2.4.3	FIFO_FWFT_Int . . . . .	45
2.4.3.1	Methods . . . . .	45
2.4.4	FIFO_UP_Int . . . . .	47
2.4.4.1	Methods . . . . .	47
2.4.5	FIFO_OP_Int . . . . .	49
2.4.5.1	Methods . . . . .	49
2.4.6	LIFO_Int . . . . .	51
2.4.6.1	Methods . . . . .	51
2.4.7	FIFO_S_Sint . . . . .	53

2.4.7.1	Methods	53
2.4.8	FIFO_S_Usint	55
2.4.8.1	Methods	55
2.4.9	FIFO_S_Uint	57
2.4.9.1	Methods	57
2.4.10	FIFO_S_Bool	59
2.4.10.1	Methods	59
2.4.11	FIFO_S_Byte	61
2.4.11.1	Methods	61
2.4.12	FIFO_S_Word	63
2.4.12.1	Methods	63
2.4.13	FIFO_S_Dword	65
2.4.13.1	Methods	65
2.4.14	FIFO_S_Lword	67
2.4.14.1	Methods	67
2.4.15	FIFO_S_Dint	69
2.4.15.1	Methods	69
2.4.16	FIFO_S_Udint	71
2.4.16.1	Methods	71
2.4.17	FIFO_S_Lint	73
2.4.17.1	Methods	73
2.4.18	FIFO_S_Ulint	75
2.4.18.1	Methods	75
2.4.19	FIFO_S_Real	77
2.4.19.1	Methods	77
2.4.20	FIFO_S_Lreal	79
2.4.20.1	Methods	79
2.4.21	FIFO_S_String	81
2.4.21.1	Methods	81
2.4.22	FIFO_S_Wstring	83
2.4.22.1	Methods	83
2.4.23	FIFO_S_Time	85
2.4.23.1	Methods	85
2.4.24	FIFO_S_Ltime	87
2.4.24.1	Methods	87
2.4.25	FIFO_S_Date	89
2.4.25.1	Methods	89
2.4.26	FIFO_S_Ldate	91
2.4.26.1	Methods	91
2.4.27	FIFO_S_Dt	93
2.4.27.1	Methods	93
2.4.28	FIFO_S_Ldt	95
2.4.28.1	Methods	95
2.4.29	FIFO_S_Tod	97
2.4.29.1	Methods	97
2.4.30	FIFO_S_Ltod	99
2.4.30.1	Methods	99
2.4.31	FIFO_C_Bool	101
2.4.31.1	Methods	101
2.4.32	FIFO_C_Byte	103
2.4.32.1	Methods	103
2.4.33	FIFO_C_Word	105
2.4.33.1	Methods	105
2.4.34	FIFO_C_Dword	107
2.4.34.1	Methods	107
2.4.35	FIFO_C_Lword	109
2.4.35.1	Methods	109
2.4.36	FIFO_C_Sint	111

2.4.36.1	Methods	111
2.4.37	FIFO_C_Usint	113
2.4.37.1	Methods	113
2.4.38	FIFO_C_Uint	115
2.4.38.1	Methods	115
2.4.39	FIFO_C_Dint	117
2.4.39.1	Methods	117
2.4.40	FIFO_C_Udint	119
2.4.40.1	Methods	119
2.4.41	FIFO_C_Lint	121
2.4.41.1	Methods	121
2.4.42	FIFO_C_Ulint	123
2.4.42.1	Methods	123
2.4.43	FIFO_C_Real	125
2.4.43.1	Methods	125
2.4.44	FIFO_C_Lreal	127
2.4.44.1	Methods	127
2.4.45	FIFO_C_String	129
2.4.45.1	Methods	129
2.4.46	FIFO_C_Wstring	131
2.4.46.1	Methods	131
2.4.47	FIFO_C_Time	133
2.4.47.1	Methods	133
2.4.48	FIFO_C_Ltime	135
2.4.48.1	Methods	135
2.4.49	FIFO_C_Date	137
2.4.49.1	Methods	137
2.4.50	FIFO_C_Ldate	139
2.4.50.1	Methods	139
2.4.51	FIFO_C_Dt	141
2.4.51.1	Methods	141
2.4.52	FIFO_C_Ldt	143
2.4.52.1	Methods	143
2.4.53	FIFO_C_Tod	145
2.4.53.1	Methods	145
2.4.54	FIFO_C_Ltod	147
2.4.54.1	Methods	147
2.4.55	FIFO_FWFT_Bool	149
2.4.55.1	Methods	149
2.4.56	FIFO_FWFT_Byte	151
2.4.56.1	Methods	151
2.4.57	FIFO_FWFT_Word	153
2.4.57.1	Methods	153
2.4.58	FIFO_FWFT_Dword	155
2.4.58.1	Methods	155
2.4.59	FIFO_FWFT_Lword	157
2.4.59.1	Methods	157
2.4.60	FIFO_FWFT_Sint	159
2.4.60.1	Methods	159
2.4.61	FIFO_FWFT_Usint	161
2.4.61.1	Methods	161
2.4.62	FIFO_FWFT_Uint	163
2.4.62.1	Methods	163
2.4.63	FIFO_FWFT_Dint	165
2.4.63.1	Methods	165
2.4.64	FIFO_FWFT_Udint	167
2.4.64.1	Methods	167
2.4.65	FIFO_FWFT_Lint	169

2.4.65.1	Methods	169
2.4.66	FIFO_FWFT_Ulint	171
2.4.66.1	Methods	171
2.4.67	FIFO_FWFT_Real	173
2.4.67.1	Methods	173
2.4.68	FIFO_FWFT_Lreal	175
2.4.68.1	Methods	175
2.4.69	FIFO_FWFT_String	177
2.4.69.1	Methods	177
2.4.70	FIFO_FWFT_Wstring	179
2.4.70.1	Methods	179
2.4.71	FIFO_FWFT_Time	181
2.4.71.1	Methods	181
2.4.72	FIFO_FWFT_Ltime	183
2.4.72.1	Methods	183
2.4.73	FIFO_FWFT_Date	185
2.4.73.1	Methods	185
2.4.74	FIFO_FWFT_Ldate	187
2.4.74.1	Methods	187
2.4.75	FIFO_FWFT_Dt	189
2.4.75.1	Methods	189
2.4.76	FIFO_FWFT_Ldt	191
2.4.76.1	Methods	191
2.4.77	FIFO_FWFT_Tod	193
2.4.77.1	Methods	193
2.4.78	FIFO_FWFT_Ltod	195
2.4.78.1	Methods	195
2.4.79	FIFO_UP_Bool	197
2.4.79.1	Methods	197
2.4.80	FIFO_UP_Byte	199
2.4.80.1	Methods	199
2.4.81	FIFO_UP_Word	201
2.4.81.1	Methods	201
2.4.82	FIFO_UP_Dword	203
2.4.82.1	Methods	203
2.4.83	FIFO_UP_Lword	205
2.4.83.1	Methods	205
2.4.84	FIFO_UP_Sint	207
2.4.84.1	Methods	207
2.4.85	FIFO_UP_Usint	209
2.4.85.1	Methods	209
2.4.86	FIFO_UP_Uint	211
2.4.86.1	Methods	211
2.4.87	FIFO_UP_Dint	213
2.4.87.1	Methods	213
2.4.88	FIFO_UP_Udint	215
2.4.88.1	Methods	215
2.4.89	FIFO_UP_Lint	217
2.4.89.1	Methods	217
2.4.90	FIFO_UP_Ulint	219
2.4.90.1	Methods	219
2.4.91	FIFO_UP_Real	221
2.4.91.1	Methods	221
2.4.92	FIFO_UP_Lreal	223
2.4.92.1	Methods	223
2.4.93	FIFO_UP_String	225
2.4.93.1	Methods	225
2.4.94	FIFO_UP_Wstring	227

2.4.94.1	Methods	227
2.4.95	FIFO_UP_Time	229
2.4.95.1	Methods	229
2.4.96	FIFO_UP_Ltime	231
2.4.96.1	Methods	231
2.4.97	FIFO_UP_Date	233
2.4.97.1	Methods	233
2.4.98	FIFO_UP_Ldate	235
2.4.98.1	Methods	235
2.4.99	FIFO_UP_Dt	237
2.4.99.1	Methods	237
2.4.100	FIFO_UP_Ldt	239
2.4.100.1	Methods	239
2.4.101	FIFO_UP_Tod	241
2.4.101.1	Methods	241
2.4.102	FIFO_UP_Ltod	243
2.4.102.1	Methods	243
2.4.103	FIFO_OP_Bool	245
2.4.103.1	Methods	245
2.4.104	FIFO_OP_Byte	247
2.4.104.1	Methods	247
2.4.105	FIFO_OP_Word	249
2.4.105.1	Methods	249
2.4.106	FIFO_OP_Dword	251
2.4.106.1	Methods	251
2.4.107	FIFO_OP_Lword	253
2.4.107.1	Methods	253
2.4.108	FIFO_OP_Sint	255
2.4.108.1	Methods	255
2.4.109	FIFO_OP_Usint	257
2.4.109.1	Methods	257
2.4.110	FIFO_OP_Uint	259
2.4.110.1	Methods	259
2.4.111	FIFO_OP_Dint	261
2.4.111.1	Methods	261
2.4.112	FIFO_OP_Udint	263
2.4.112.1	Methods	263
2.4.113	FIFO_OP_Lint	265
2.4.113.1	Methods	265
2.4.114	FIFO_OP_Ulint	267
2.4.114.1	Methods	267
2.4.115	FIFO_OP_Real	269
2.4.115.1	Methods	269
2.4.116	FIFO_OP_Lreal	271
2.4.116.1	Methods	271
2.4.117	FIFO_OP_String	273
2.4.117.1	Methods	273
2.4.118	FIFO_OP_Wstring	275
2.4.118.1	Methods	275
2.4.119	FIFO_OP_Time	277
2.4.119.1	Methods	277
2.4.120	FIFO_OP_Ltime	279
2.4.120.1	Methods	279
2.4.121	FIFO_OP_Date	281
2.4.121.1	Methods	281
2.4.122	FIFO_OP_Ldate	283
2.4.122.1	Methods	283
2.4.123	FIFO_OP_Dt	285

2.4.123.1 Methods . . . . .	285
2.4.124 FIFO_OP_Ldt . . . . .	287
2.4.124.1 Methods . . . . .	287
2.4.125 FIFO_OP_Tod . . . . .	289
2.4.125.1 Methods . . . . .	289
2.4.126 FIFO_OP_Ltod . . . . .	291
2.4.126.1 Methods . . . . .	291
2.4.127 LIFO_Bool . . . . .	293
2.4.127.1 Methods . . . . .	293
2.4.128 LIFO_Byte . . . . .	295
2.4.128.1 Methods . . . . .	295
2.4.129 LIFO_Word . . . . .	297
2.4.129.1 Methods . . . . .	297
2.4.130 LIFO_Dword . . . . .	299
2.4.130.1 Methods . . . . .	299
2.4.131 LIFO_Lword . . . . .	301
2.4.131.1 Methods . . . . .	301
2.4.132 LIFO_Sint . . . . .	303
2.4.132.1 Methods . . . . .	303
2.4.133 LIFO_Usint . . . . .	305
2.4.133.1 Methods . . . . .	305
2.4.134 LIFO_Uint . . . . .	307
2.4.134.1 Methods . . . . .	307
2.4.135 LIFO_Dint . . . . .	309
2.4.135.1 Methods . . . . .	309
2.4.136 LIFO_Udint . . . . .	311
2.4.136.1 Methods . . . . .	311
2.4.137 LIFO_Lint . . . . .	313
2.4.137.1 Methods . . . . .	313
2.4.138 LIFO_Ulint . . . . .	315
2.4.138.1 Methods . . . . .	315
2.4.139 LIFO_Real . . . . .	317
2.4.139.1 Methods . . . . .	317
2.4.140 LIFO_Lreal . . . . .	319
2.4.140.1 Methods . . . . .	319
2.4.141 LIFO_String . . . . .	321
2.4.141.1 Methods . . . . .	321
2.4.142 LIFO_Wstring . . . . .	323
2.4.142.1 Methods . . . . .	323
2.4.143 LIFO_Time . . . . .	325
2.4.143.1 Methods . . . . .	325
2.4.144 LIFO_Ltime . . . . .	327
2.4.144.1 Methods . . . . .	327
2.4.145 LIFO_Date . . . . .	329
2.4.145.1 Methods . . . . .	329
2.4.146 LIFO_Ldate . . . . .	331
2.4.146.1 Methods . . . . .	331
2.4.147 LIFO_Dt . . . . .	333
2.4.147.1 Methods . . . . .	333
2.4.148 LIFO_Ldt . . . . .	335
2.4.148.1 Methods . . . . .	335
2.4.149 LIFO_Tod . . . . .	337
2.4.149.1 Methods . . . . .	337
2.4.150 LIFO_Ltod . . . . .	339
2.4.150.1 Methods . . . . .	339
2.4.151 FIFO_C_8b . . . . .	341
2.4.151.1 Methods . . . . .	341
2.4.152 FIFO_C_16b . . . . .	343

2.4.152.1 Methods . . . . .	343
2.4.153 FIFO_C_32b . . . . .	345
2.4.153.1 Methods . . . . .	345
2.4.154 FIFO_C_64b . . . . .	347
2.4.154.1 Methods . . . . .	347
<b>3 Documentation for namespace CODESYS_Control_Win_V3_x64</b>	<b>349</b>
3.1 Namespace index . . . . .	349
3.2 Metrics . . . . .	350
3.3 Programs . . . . .	351
3.3.1 FIFO_S_ManTest . . . . .	351
3.3.2 FIFO_C_ManTest . . . . .	352
3.3.3 FIFO_FWFT_ManTest . . . . .	353
3.3.4 FIFO_UP_ManTest . . . . .	354
3.3.5 LIFO_ManTest . . . . .	355
3.3.6 FIFO_C_UnitTest . . . . .	356
3.3.7 FIFO_S_UnitTest . . . . .	357
3.3.8 FIFO_FWFT_UnitTest . . . . .	358
3.3.9 FIFO_UP_UnitTest . . . . .	359
3.3.10 FIFO_OP_UnitTest . . . . .	360
3.3.11 LIFO_UnitTest . . . . .	361
3.3.12 FIFO_OP_ManTest . . . . .	362
3.3.13 FIFO_C_xxb_UnitTest . . . . .	363



# Chapter 1

## Project information

PROJECT	INFORMATION
Company name	TK Automation
Company URL	<a href="https://github.com/tkucic/QueueLib">https://github.com/tkucic/QueueLib</a>
Project name	QueueLib
Project version	1.0
Project URL	
Content description	Library that holds FIFO and LIFO buffer classes for use in IEC programs
Contact person	Toni Kucic
Content generation date	2023-02-19T23:10:57.3402784

## 1.1 List of namespaces

- Global
- CODESYS\_Control\_Win\_V3\_x64

## 1.2 Metrics

Namespace	Data types	Program	Function Block	Function	Class
Global	25	0	154	0	0
CODESYS_Control_Win_V3_x64	0	13	0	0	0
<b>Total</b>	<b>25</b>	<b>13</b>	<b>154</b>	<b>0</b>	<b>0</b>

Namespace	Lines of code	Lines of comments	Lines in total	Maintainable size
Global	4205	529	6008	5337
CODESYS_Control_Win_V3_x64	419	111	656	634
<b>Total</b>	<b>4624</b>	<b>640</b>	<b>6664</b>	<b>5971</b>

## Chapter 2

# Documentation for namespace Global

### 2.1 Namespace index

- **Data types**

- dtFIFO\_P\_Element\_Int
- dtFIFO\_P\_Element\_Bool
- dtFIFO\_P\_Element\_Byte
- dtFIFO\_P\_Element\_Word
- dtFIFO\_P\_Element\_Dword
- dtFIFO\_P\_Element\_Lword
- dtFIFO\_P\_Element\_Sint
- dtFIFO\_P\_Element\_Usint
- dtFIFO\_P\_Element\_Uint
- dtFIFO\_P\_Element\_Dint
- dtFIFO\_P\_Element\_Udint
- dtFIFO\_P\_Element\_Lint
- dtFIFO\_P\_Element\_Ulint
- dtFIFO\_P\_Element\_Real
- dtFIFO\_P\_Element\_Lreal
- dtFIFO\_P\_Element\_String
- dtFIFO\_P\_Element\_Wstring
- dtFIFO\_P\_Element\_Time
- dtFIFO\_P\_Element\_Ltime
- dtFIFO\_P\_Element\_Date
- dtFIFO\_P\_Element\_Ldate
- dtFIFO\_P\_Element\_Dt
- dtFIFO\_P\_Element\_Ldt
- dtFIFO\_P\_Element\_Tod
- dtFIFO\_P\_Element\_Ltod

- **Function blocks**

- FIFO\_S\_Int
- FIFO\_C\_Int

- FIFO\_FWFT\_Int
- FIFO\_UP\_Int
- FIFO\_OP\_Int
- LIFO\_Int
- FIFO\_S\_Sint
- FIFO\_S\_Usint
- FIFO\_S\_Uint
- FIFO\_S\_Bool
- FIFO\_S\_Byte
- FIFO\_S\_Word
- FIFO\_S\_Dword
- FIFO\_S\_Lword
- FIFO\_S\_Dint
- FIFO\_S\_Udint
- FIFO\_S\_Lint
- FIFO\_S\_Ulint
- FIFO\_S\_Real
- FIFO\_S\_Lreal
- FIFO\_S\_String
- FIFO\_S\_Wstring
- FIFO\_S\_Time
- FIFO\_S\_Ltime
- FIFO\_S\_Date
- FIFO\_S\_Ldate
- FIFO\_S\_Dt
- FIFO\_S\_Ldt
- FIFO\_S\_Tod
- FIFO\_S\_Ltod
- FIFO\_C\_Bool
- FIFO\_C\_Byte
- FIFO\_C\_Word
- FIFO\_C\_Dword
- FIFO\_C\_Lword
- FIFO\_C\_Sint
- FIFO\_C\_Usint
- FIFO\_C\_Uint
- FIFO\_C\_Dint
- FIFO\_C\_Udint
- FIFO\_C\_Lint
- FIFO\_C\_Ulint
- FIFO\_C\_Real
- FIFO\_C\_Lreal
- FIFO\_C\_String
- FIFO\_C\_Wstring
- FIFO\_C\_Time

- FIFO\_C\_Ltime
- FIFO\_C\_Date
- FIFO\_C\_Ldate
- FIFO\_C\_Dt
- FIFO\_C\_Ldt
- FIFO\_C\_Tod
- FIFO\_C\_Ltod
- FIFO\_FWFT\_Bool
- FIFO\_FWFT\_Byte
- FIFO\_FWFT\_Word
- FIFO\_FWFT\_Dword
- FIFO\_FWFT\_Lword
- FIFO\_FWFT\_Sint
- FIFO\_FWFT\_Usint
- FIFO\_FWFT\_Uint
- FIFO\_FWFT\_Dint
- FIFO\_FWFT\_Udint
- FIFO\_FWFT\_Lint
- FIFO\_FWFT\_Ulint
- FIFO\_FWFT\_Real
- FIFO\_FWFT\_Lreal
- FIFO\_FWFT\_String
- FIFO\_FWFT\_Wstring
- FIFO\_FWFT\_Time
- FIFO\_FWFT\_Ltime
- FIFO\_FWFT\_Date
- FIFO\_FWFT\_Ldate
- FIFO\_FWFT\_Dt
- FIFO\_FWFT\_Ldt
- FIFO\_FWFT\_Tod
- FIFO\_FWFT\_Ltod
- FIFO\_UP\_Bool
- FIFO\_UP\_Byte
- FIFO\_UP\_Word
- FIFO\_UP\_Dword
- FIFO\_UP\_Lword
- FIFO\_UP\_Sint
- FIFO\_UP\_Usint
- FIFO\_UP\_Uint
- FIFO\_UP\_Dint
- FIFO\_UP\_Udint
- FIFO\_UP\_Lint
- FIFO\_UP\_Ulint
- FIFO\_UP\_Real
- FIFO\_UP\_Lreal

- FIFO\_UP\_String
- FIFO\_UP\_Wstring
- FIFO\_UP\_Time
- FIFO\_UP\_Ltime
- FIFO\_UP\_Date
- FIFO\_UP\_Ldate
- FIFO\_UP\_Dt
- FIFO\_UP\_Ldt
- FIFO\_UP\_Tod
- FIFO\_UP\_Ltod
- FIFO\_OP\_Bool
- FIFO\_OP\_Byte
- FIFO\_OP\_Word
- FIFO\_OP\_Dword
- FIFO\_OP\_Lword
- FIFO\_OP\_Sint
- FIFO\_OP\_Usint
- FIFO\_OP\_Uint
- FIFO\_OP\_Dint
- FIFO\_OP\_Udint
- FIFO\_OP\_Lint
- FIFO\_OP\_Ulint
- FIFO\_OP\_Real
- FIFO\_OP\_Lreal
- FIFO\_OP\_String
- FIFO\_OP\_Wstring
- FIFO\_OP\_Time
- FIFO\_OP\_Ltime
- FIFO\_OP\_Date
- FIFO\_OP\_Ldate
- FIFO\_OP\_Dt
- FIFO\_OP\_Ldt
- FIFO\_OP\_Tod
- FIFO\_OP\_Ltod
- LIFO\_Bool
- LIFO\_Byte
- LIFO\_Word
- LIFO\_Dword
- LIFO\_Lword
- LIFO\_Sint
- LIFO\_Usint
- LIFO\_Uint
- LIFO\_Dint
- LIFO\_Udint
- LIFO\_Lint

- LIFO\_Ulint
- LIFO\_Real
- LIFO\_Lreal
- LIFO\_String
- LIFO\_Wstring
- LIFO\_Time
- LIFO\_Ltime
- LIFO\_Date
- LIFO\_Ldate
- LIFO\_Dt
- LIFO\_Ldt
- LIFO\_Tod
- LIFO\_Ltod
- FIFO\_C\_8b
- FIFO\_C\_16b
- FIFO\_C\_32b
- FIFO\_C\_64b

## 2.2 Metrics

Data types	Programs	Function Blocks	Functions	Classes
25	0	154	0	0

Lines of code	Lines of comments	Lines in total	Maintainable size
4205	529	6008	5337



## 2.3 Data types

### 2.3.1 dtFIFO\_P\_Element\_Int (struct)

Priority FIFO buffer structure

#### 2.3.1.0.1 Components

Name	Type	Documentation
Value	INT	Value member
Priority	INT	Priority member

#### 2.3.1.0.2 Metrics    Number of components: 2

**2.3.2 dtFIFO\_P\_Element\_Bool (struct)**

Priority FIFO buffer structure

**2.3.2.0.1 Components**

Name	Type	Documentation
Value	BOOL	Value member
Priority	INT	Priority member

**2.3.2.0.2 Metrics**    Number of components: 2

**2.3.3 dtFIFO\_P\_Element\_Byte (struct)**

Priority FIFO buffer structure

**2.3.3.0.1 Components**

Name	Type	Documentation
Value	BYTE	Value member
Priority	INT	Priority member

**2.3.3.0.2 Metrics**    Number of components: 2

### 2.3.4 dtFIFO\_P\_Element\_Word (struct)

Priority FIFO buffer structure

#### 2.3.4.0.1 Components

Name	Type	Documentation
Value	WORD	Value member
Priority	INT	Priority member

#### 2.3.4.0.2 Metrics    Number of components: 2

### 2.3.5 dtFIFO\_P\_Element\_Dword (struct)

Priority FIFO buffer structure

#### 2.3.5.0.1 Components

Name	Type	Documentation
Value	DWORD	Value member
Priority	INT	Priority member

#### 2.3.5.0.2 Metrics    Number of components: 2

### 2.3.6 dtFIFO\_P\_Element\_Lword (struct)

Priority FIFO buffer structure

#### 2.3.6.0.1 Components

Name	Type	Documentation
Value	LWORD	Value member
Priority	INT	Priority member

#### 2.3.6.0.2 Metrics    Number of components: 2

**2.3.7 dtFIFO\_P\_Element\_Sint (struct)**

Priority FIFO buffer structure

**2.3.7.0.1 Components**

Name	Type	Documentation
Value	SINT	Value member
Priority	INT	Priority member

**2.3.7.0.2 Metrics**    Number of components: 2

**2.3.8 dtFIFO\_P\_Element\_Usint (struct)**

Priority FIFO buffer structure

**2.3.8.0.1 Components**

Name	Type	Documentation
Value	USINT	Value member
Priority	INT	Priority member

**2.3.8.0.2 Metrics**    Number of components: 2



**2.3.9 dtFIFO\_P\_Element\_Uint (struct)**

Priority FIFO buffer structure

**2.3.9.0.1 Components**

Name	Type	Documentation
Value	UINT	Value member
Priority	INT	Priority member

**2.3.9.0.2 Metrics**    Number of components: 2

**2.3.10 dtFIFO\_P\_Element\_Dint (struct)**

Priority FIFO buffer structure

**2.3.10.0.1 Components**

Name	Type	Documentation
Value	DINT	Value member
Priority	INT	Priority member

**2.3.10.0.2 Metrics** Number of components: 2

**2.3.11 dtFIFO\_P\_Element\_Udint (struct)**

Priority FIFO buffer structure

**2.3.11.0.1 Components**

Name	Type	Documentation
Value	UDINT	Value member
Priority	INT	Priority member

**2.3.11.0.2 Metrics**    Number of components: 2

**2.3.12 dtFIFO\_P\_Element\_Lint (struct)**

Priority FIFO buffer structure

**2.3.12.0.1 Components**

Name	Type	Documentation
Value	LINT	Value member
Priority	INT	Priority member

**2.3.12.0.2 Metrics** Number of components: 2

**2.3.13 dtFIFO\_P\_Element\_Ulint (struct)**

Priority FIFO buffer structure

**2.3.13.0.1 Components**

Name	Type	Documentation
Value	ULINT	Value member
Priority	INT	Priority member

**2.3.13.0.2 Metrics** Number of components: 2

**2.3.14 dtFIFO\_P\_Element\_Real (struct)**

Priority FIFO buffer structure

**2.3.14.0.1 Components**

Name	Type	Documentation
Value	REAL	Value member
Priority	INT	Priority member

**2.3.14.0.2 Metrics**    Number of components: 2

**2.3.15 dtFIFO\_P\_Element\_Lreal (struct)**

Priority FIFO buffer structure

**2.3.15.0.1 Components**

Name	Type	Documentation
Value	LREAL	Value member
Priority	INT	Priority member

**2.3.15.0.2 Metrics** Number of components: 2

**2.3.16 dtFIFO\_P\_Element\_String (struct)**

Priority FIFO buffer structure

**2.3.16.0.1 Components**

Name	Type	Documentation
Value	STRING[None]	Value member
Priority	INT	Priority member

**2.3.16.0.2 Metrics**    Number of components: 2



**2.3.17 dtFIFO\_P\_Element\_Wstring (struct)**

Priority FIFO buffer structure

**2.3.17.0.1 Components**

Name	Type	Documentation
Value	wstring	Value member
Priority	INT	Priority member

**2.3.17.0.2 Metrics**    Number of components: 2

**2.3.18 dtFIFO\_P\_Element\_Time (struct)**

Priority FIFO buffer structure

**2.3.18.0.1 Components**

Name	Type	Documentation
Value	TIME	Value member
Priority	INT	Priority member

**2.3.18.0.2 Metrics**    Number of components: 2

**2.3.19 dtFIFO\_P\_Element\_Ltime (struct)**

Priority FIFO buffer structure

**2.3.19.0.1 Components**

Name	Type	Documentation
Value	LTIME	Value member
Priority	INT	Priority member

**2.3.19.0.2 Metrics**    Number of components: 2

**2.3.20 dtFIFO\_P\_Element\_Date (struct)**

Priority FIFO buffer structure

**2.3.20.0.1 Components**

Name	Type	Documentation
Value	DATE	Value member
Priority	INT	Priority member

**2.3.20.0.2 Metrics** Number of components: 2

**2.3.21 dtFIFO\_P\_Element\_Ldate (struct)**

Priority FIFO buffer structure

**2.3.21.0.1 Components**

Name	Type	Documentation
Value	LDATE	Value member
Priority	INT	Priority member

**2.3.21.0.2 Metrics** Number of components: 2

**2.3.22 dtFIFO\_P\_Element\_Dt (struct)**

Priority FIFO buffer structure

**2.3.22.0.1 Components**

Name	Type	Documentation
Value	DT	Value member
Priority	INT	Priority member

**2.3.22.0.2 Metrics** Number of components: 2

**2.3.23 dtFIFO\_P\_Element\_Ldt (struct)**

Priority FIFO buffer structure

**2.3.23.0.1 Components**

Name	Type	Documentation
Value	LDATE_AND_TIME	Value member
Priority	INT	Priority member

**2.3.23.0.2 Metrics** Number of components: 2

**2.3.24 dtFIFO\_P\_Element\_Tod (struct)**

Priority FIFO buffer structure

**2.3.24.0.1 Components**

Name	Type	Documentation
Value	TOD	Value member
Priority	INT	Priority member

**2.3.24.0.2 Metrics**    Number of components: 2



**2.3.25 dtFIFO\_P\_Element\_Ltod (struct)**

Priority FIFO buffer structure

**2.3.25.0.1 Components**

Name	Type	Documentation
Value	LTIME_OF_DAY	Value member
Priority	INT	Priority member

**2.3.25.0.2 Metrics** Number of components: 2

## 2.4 Function blocks

### 2.4.1 FIFO\_S\_Int

Standard FIFO buffer. Allows Enqueuing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.1.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO INT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.1.0.2 Incode documentation

Developed entirely with methods

#### 2.4.1.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.1.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.1.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.1.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	INT	Value to add to queue

**2.4.1.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	INT	Value to add to queue

**2.4.1.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.1.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	INT	Return value

### 2.4.2 FIFO\_C\_Int

Circular FIFO buffer. Allows Enqueuing when there is a free space in front of the tail

#### 2.4.2.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO INT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.2.0.2 Incode documentation

Developed entirely with methods

#### 2.4.2.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.2.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.2.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	INT	Return value

##### 2.4.2.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	INT	Value to add to queue

**2.4.2.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.2.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.2.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	INT	Return value

### 2.4.3 FIFO\_FWFT\_Int

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

#### 2.4.3.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO INT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.3.0.2 Incode documentation Developed entirely with methods

#### 2.4.3.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

#### 2.4.3.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.3.1.1 Method Dequeue : **BOOL** Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	INT	Return value

##### 2.4.3.1.2 Method Enqueue : **BOOL** Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	INT	Value to add to queue

**2.4.3.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.3.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.3.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	INT	Return value

#### 2.4.4 FIFO\_UP\_Int

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

##### 2.4.4.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_INT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

##### 2.4.4.0.2 Incode documentation

Developed entirely with methods

##### 2.4.4.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

##### 2.4.4.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.4.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	INT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.4.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	INT	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.4.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.4.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.4.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	INT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.5 FIFO\_OP\_Int

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.5.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_INT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.5.0.2 Incode documentation

Developed entirely with methods

#### 2.4.5.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.5.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.5.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	INT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.5.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	INT	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.5.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.5.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.5.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	INT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.6 LIFO\_Int

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.6.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO INT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.6.0.2 Incode documentation Developed entirely with methods

#### 2.4.6.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.6.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.6.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	INT	Return value

##### 2.4.6.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	INT	Value to add to the stack

**2.4.6.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.6.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.6.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	INT	Return value

### 2.4.7 FIFO\_S\_Sint

Standard FIFO buffer. Allows Enqueing if buffer is not full. After being filled, it has to be emptied completely before further equeueing

#### 2.4.7.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO SINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.7.0.2 Incode documentation Developed entirely with methods

#### 2.4.7.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.7.1 Methods

- Dequeue
- Enqueue
- isFull
- Peek
- isEmpty

##### 2.4.7.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	SINT	Value to add to queue

##### 2.4.7.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	SINT	Value to add to queue

**2.4.7.1.3 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.7.1.4 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	SINT	Return value

**2.4.7.1.5 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

### 2.4.8 FIFO\_S\_Usint

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.8.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO USINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.8.0.2 Incode documentation

Developed entirely with methods

#### 2.4.8.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.8.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.8.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.8.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	USINT	Value to add to queue



**2.4.8.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	USINT	Value to add to queue

**2.4.8.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.8.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	USINT	Return value

### 2.4.9 FIFO\_S\_Uint

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.9.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO UINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.9.0.2 Incode documentation

Developed entirely with methods

#### 2.4.9.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.9.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.9.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.9.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UINT	Value to add to queue

**2.4.9.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	UINT	Value to add to queue

**2.4.9.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.9.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UINT	Return value

### 2.4.10 FIFO\_S\_Bool

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.10.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO BOOL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.10.0.2 Incode documentation

Developed entirely with methods

#### 2.4.10.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.10.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.10.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.10.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BOOL	Value to add to queue

**2.4.10.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	BOOL	Value to add to queue

**2.4.10.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.10.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BOOL	Return value

### 2.4.11 FIFO\_S\_Byte

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.11.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO BYTE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.11.0.2 Incode documentation

Developed entirely with methods

#### 2.4.11.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.11.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.11.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.11.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BYTE	Value to add to queue

**2.4.11.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	BYTE	Value to add to queue

**2.4.11.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.11.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BYTE	Return value

### 2.4.12 FIFO\_S\_Word

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.12.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO WORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.12.0.2 Incode documentation Developed entirely with methods

#### 2.4.12.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.12.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

#### 2.4.12.1.1 Method isEmpty : BOOL Returns TRUE if FIFO is empty. Reading not allowed

##### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

#### 2.4.12.1.2 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	WORD	Value to add to queue



**2.4.12.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	WORD	Value to add to queue

**2.4.12.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.12.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	WORD	Return value

### 2.4.13 FIFO\_S\_Dword

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.13.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DWORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.13.0.2 Incode documentation

Developed entirely with methods

#### 2.4.13.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.13.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.13.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.13.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DWORD	Value to add to queue

**2.4.13.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	DWORD	Value to add to queue

**2.4.13.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.13.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DWORD	Return value

### 2.4.14 FIFO\_S\_Lword

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.14.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LWORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.14.0.2 Incode documentation

Developed entirely with methods

#### 2.4.14.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.14.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.14.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.14.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LWORD	Value to add to queue

**2.4.14.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	LWORD	Value to add to queue

**2.4.14.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.14.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LWORD	Return value

### 2.4.15 FIFO\_S\_Dint

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.15.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.15.0.2 Incode documentation

Developed entirely with methods

#### 2.4.15.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.15.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.15.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.15.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DINT	Value to add to queue

**2.4.15.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	DINT	Value to add to queue

**2.4.15.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.15.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DINT	Return value

### 2.4.16 FIFO\_S\_Udint

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.16.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO UDINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.16.0.2 Incode documentation

Developed entirely with methods

#### 2.4.16.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.16.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.16.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.16.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UDINT	Value to add to queue



**2.4.16.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	UDINT	Value to add to queue

**2.4.16.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.16.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UDINT	Return value

### 2.4.17 FIFO\_S\_Lint

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.17.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.17.0.2 Incode documentation

Developed entirely with methods

#### 2.4.17.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.17.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.17.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.17.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LINT	Value to add to queue

**2.4.17.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	LINT	Value to add to queue

**2.4.17.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.17.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LINT	Return value

### 2.4.18 FIFO\_S\_Ulint

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.18.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO ULINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.18.0.2 Incode documentation

Developed entirely with methods

#### 2.4.18.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.18.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.18.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.18.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	ULINT	Value to add to queue

**2.4.18.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	ULINT	Value to add to queue

**2.4.18.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.18.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	ULINT	Return value

### 2.4.19 FIFO\_S\_Real

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.19.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO REAL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.19.0.2 Incode documentation

Developed entirely with methods

#### 2.4.19.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.19.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.19.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.19.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	REAL	Value to add to queue

**2.4.19.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	REAL	Value to add to queue

**2.4.19.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.19.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	REAL	Return value

### 2.4.20 FIFO\_S\_Lreal

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.20.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LREAL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.20.0.2 Incode documentation Developed entirely with methods

#### 2.4.20.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.20.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

#### 2.4.20.1.1 Method isEmpty : BOOL Returns TRUE if FIFO is empty. Reading not allowed

##### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

#### 2.4.20.1.2 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LREAL	Value to add to queue



**2.4.20.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	LREAL	Value to add to queue

**2.4.20.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.20.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LREAL	Return value

### 2.4.21 FIFO\_S\_String

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.21.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO STRING[None]	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.21.0.2 Incode documentation

Developed entirely with methods

#### 2.4.21.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.21.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.21.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.21.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	STRING[None]	Value to add to queue

**2.4.21.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	STRING[None]	Value to add to queue

**2.4.21.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.21.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	STRING[None]	Return value

### 2.4.22 FIFO\_S\_Wstring

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.22.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO wstring	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.22.0.2 Incode documentation Developed entirely with methods

#### 2.4.22.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.22.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

#### 2.4.22.1.1 Method isEmpty : BOOL Returns TRUE if FIFO is empty. Reading not allowed

##### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

#### 2.4.22.1.2 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	wstring	Value to add to queue

**2.4.22.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	wstring	Value to add to queue

**2.4.22.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.22.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	wstring	Return value

### 2.4.23 FIFO\_S\_Time

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.23.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO TIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.23.0.2 Incode documentation

Developed entirely with methods

#### 2.4.23.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.23.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.23.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.23.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TIME	Value to add to queue

**2.4.23.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	TIME	Value to add to queue

**2.4.23.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.23.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TIME	Return value

### 2.4.24 FIFO\_S\_Ltime

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.24.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LTIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.24.0.2 Incode documentation Developed entirely with methods

#### 2.4.24.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.24.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

#### 2.4.24.1.1 Method isEmpty : BOOL Returns TRUE if FIFO is empty. Reading not allowed

##### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

#### 2.4.24.1.2 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME	Value to add to queue



**2.4.24.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	LTIME	Value to add to queue

**2.4.24.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.24.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME	Return value

### 2.4.25 FIFO\_S\_Date

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.25.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DATE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.25.0.2 Incode documentation

Developed entirely with methods

#### 2.4.25.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.25.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.25.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.25.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DATE	Value to add to queue

**2.4.25.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	DATE	Value to add to queue

**2.4.25.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.25.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DATE	Return value

### 2.4.26 FIFO\_S\_Ldate

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.26.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LDATE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.26.0.2 Incode documentation

Developed entirely with methods

#### 2.4.26.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.26.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.26.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.26.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE	Value to add to queue

**2.4.26.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	LDATE	Value to add to queue

**2.4.26.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.26.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE	Return value

### 2.4.27 FIFO\_S\_Dt

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.27.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.27.0.2 Incode documentation

Developed entirely with methods

#### 2.4.27.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.27.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.27.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.27.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DT	Value to add to queue

**2.4.27.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	DT	Value to add to queue

**2.4.27.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.27.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DT	Return value

### 2.4.28 FIFO\_S\_Ldt

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.28.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LDATE_AND_TIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.28.0.2 Incode documentation

Developed entirely with methods

#### 2.4.28.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.28.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.28.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

###### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.28.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATA_AND_TIME	Value to add to queue



**2.4.28.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	LDATE_AND_TIME	Value to add to queue

**2.4.28.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.28.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE_AND_TIME	Return value

## 2.4.29 FIFO\_S\_Tod

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

### 2.4.29.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO TOD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.29.0.2 Incode documentation

 Developed entirely with methods

### 2.4.29.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

### 2.4.29.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

#### 2.4.29.1.1 Method isEmpty : BOOL

 Returns TRUE if FIFO is empty. Reading not allowed

##### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

#### 2.4.29.1.2 Method Dequeue : BOOL

 Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TOD	Value to add to queue

**2.4.29.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	TOD	Value to add to queue

**2.4.29.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.29.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TOD	Return value

### 2.4.30 FIFO\_S\_Ltod

Standard FIFO buffer. Allows Enqueueing if buffer is not full. After being filled, it has to be emptied completely before further enqueueing

#### 2.4.30.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LTIME_OF_DAY	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.30.0.2 Incode documentation

Developed entirely with methods

#### 2.4.30.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	28	2	40	36

#### 2.4.30.1 Methods

- isEmpty
- Dequeue
- Enqueue
- isFull
- Peek

##### 2.4.30.1.1 Method isEmpty : BOOL

Returns TRUE if FIFO is empty. Reading not allowed

###### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

##### 2.4.30.1.2 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME_OF_DAY	Value to add to queue

**2.4.30.1.3 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_INPUT	Value	LTIME_OF_DAY	Value to add to queue

**2.4.30.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation

**2.4.30.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME_OF_DAY	Return value

### 2.4.31 FIFO\_C\_Bool

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.31.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO BOOL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.31.0.2 Incode documentation

Developed entirely with methods

#### 2.4.31.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.31.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.31.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BOOL	Return value

##### 2.4.31.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	BOOL	Value to add to queue

**2.4.31.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.31.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.31.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BOOL	Return value

### 2.4.32 FIFO\_C\_Byte

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.32.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO BYTE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.32.0.2 Incode documentation

Developed entirely with methods

#### 2.4.32.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.32.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.32.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BYTE	Return value

##### 2.4.32.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	BYTE	Value to add to queue



**2.4.32.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.32.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.32.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BYTE	Return value

### 2.4.33 FIFO\_C\_Word

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.33.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO WORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.33.0.2 Incode documentation

Developed entirely with methods

#### 2.4.33.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.33.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.33.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	WORD	Return value

##### 2.4.33.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	WORD	Value to add to queue

**2.4.33.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.33.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.33.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	WORD	Return value

### 2.4.34 FIFO\_C\_Dword

Circular FIFO buffer. Allows Enqueuing when there is a free space in front of the tail

#### 2.4.34.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DWORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.34.0.2 Incode documentation

Developed entirely with methods

#### 2.4.34.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.34.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.34.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DWORD	Return value

##### 2.4.34.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DWORD	Value to add to queue

**2.4.34.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.34.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.34.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DWORD	Return value

### 2.4.35 FIFO\_C\_Lword

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.35.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LWORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.35.0.2 Incode documentation

Developed entirely with methods

#### 2.4.35.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.35.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.35.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LWORD	Return value

##### 2.4.35.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LWORD	Value to add to queue

**2.4.35.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.35.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.35.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LWORD	Return value

### 2.4.36 FIFO\_C\_Sint

Circular FIFO buffer. Allows Enqueuing when there is a free space in front of the tail

#### 2.4.36.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO SINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.36.0.2 Incode documentation

Developed entirely with methods

#### 2.4.36.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.36.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.36.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	SINT	Return value

##### 2.4.36.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	SINT	Value to add to queue



**2.4.36.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.36.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.36.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	SINT	Return value

### 2.4.37 FIFO\_C\_Usint

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.37.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO USINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.37.0.2 Incode documentation

Developed entirely with methods

#### 2.4.37.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.37.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.37.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	USINT	Return value

##### 2.4.37.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	USINT	Value to add to queue

**2.4.37.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.37.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.37.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	USINT	Return value

### 2.4.38 FIFO\_C\_Uint

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.38.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO UINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.38.0.2 Incode documentation

Developed entirely with methods

#### 2.4.38.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.38.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.38.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UINT	Return value

##### 2.4.38.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	UINT	Value to add to queue

**2.4.38.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.38.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.38.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UINT	Return value

### 2.4.39 FIFO\_C\_Dint

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.39.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.39.0.2 Incode documentation

Developed entirely with methods

#### 2.4.39.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.39.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.39.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DINT	Return value

##### 2.4.39.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DINT	Value to add to queue

**2.4.39.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.39.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.39.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DINT	Return value

## 2.4.40 FIFO\_C\_Udint

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

### 2.4.40.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO UDINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.40.0.2 Incode documentation

 Developed entirely with methods

### 2.4.40.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

### 2.4.40.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.40.1.1 Method Dequeue : BOOL

 Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UDINT	Return value

#### 2.4.40.1.2 Method Enqueue : BOOL

 Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	UDINT	Value to add to queue



**2.4.40.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.40.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.40.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UDINT	Return value

### 2.4.41 FIFO\_C\_Lint

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.41.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

**2.4.41.0.2 Incode documentation** Developed entirely with methods

#### 2.4.41.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.41.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

**2.4.41.1.1 Method Dequeue : BOOL** Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LINT	Return value

**2.4.41.1.2 Method Enqueue : BOOL** Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LINT	Value to add to queue

**2.4.41.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.41.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.41.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LINT	Return value

## 2.4.42 FIFO\_C\_Ulint

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

### 2.4.42.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO ULINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.42.0.2 Incode documentation

Developed entirely with methods

### 2.4.42.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

### 2.4.42.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.42.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	ULINT	Return value

#### 2.4.42.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	ULINT	Value to add to queue

**2.4.42.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.42.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.42.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	ULINT	Return value

### 2.4.43 FIFO\_C\_Real

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.43.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO REAL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.43.0.2 Incode documentation

Developed entirely with methods

#### 2.4.43.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.43.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.43.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	REAL	Return value

##### 2.4.43.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	REAL	Value to add to queue

**2.4.43.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.43.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.43.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	REAL	Return value

## 2.4.44 FIFO\_C\_Lreal

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

### 2.4.44.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LREAL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.44.0.2 Incode documentation

 Developed entirely with methods

### 2.4.44.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

### 2.4.44.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.44.1.1 Method Dequeue : BOOL

 Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LREAL	Return value

#### 2.4.44.1.2 Method Enqueue : BOOL

 Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LREAL	Value to add to queue



**2.4.44.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.44.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.44.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LREAL	Return value

### 2.4.45 FIFO\_C\_String

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.45.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO STRING[None]	Externally allocated buffer. Must be in format AR- RAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.45.0.2 Incode documentation

Developed entirely with methods

#### 2.4.45.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of com- ments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.45.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.45.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	STRING[None]	Return value

##### 2.4.45.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	STRING[None]	Value to add to queue

**2.4.45.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.45.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.45.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	STRING[None]	Return value

## 2.4.46 FIFO\_C\_Wstring

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

### 2.4.46.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO wstring	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.46.0.2 Incode documentation

 Developed entirely with methods

### 2.4.46.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

### 2.4.46.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.46.1.1 Method Dequeue : BOOL

 Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	wstring	Return value

#### 2.4.46.1.2 Method Enqueue : BOOL

 Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	wstring	Value to add to queue

**2.4.46.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.46.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.46.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	wstring	Return value

## 2.4.47 FIFO\_C\_Time

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

### 2.4.47.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO TIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.47.0.2 Incode documentation

Developed entirely with methods

### 2.4.47.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

### 2.4.47.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.47.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TIME	Return value

#### 2.4.47.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	TIME	Value to add to queue

**2.4.47.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.47.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.47.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TIME	Return value

## 2.4.48 FIFO\_C\_Ltime

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

### 2.4.48.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LTIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.48.0.2 Incode documentation

Developed entirely with methods

### 2.4.48.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

### 2.4.48.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.48.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME	Return value

#### 2.4.48.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LTIME	Value to add to queue



**2.4.48.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.48.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.48.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME	Return value

## 2.4.49 FIFO\_C\_Date

Circular FIFO buffer. Allows Enqueuing when there is a free space in front of the tail

### 2.4.49.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DATE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.49.0.2 Incode documentation

Developed entirely with methods

### 2.4.49.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

### 2.4.49.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.49.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DATE	Return value

#### 2.4.49.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DATE	Value to add to queue

**2.4.49.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.49.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.49.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DATE	Return value

### 2.4.50 FIFO\_C\_Ldate

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.50.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LDATE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.50.0.2 Incode documentation

Developed entirely with methods

#### 2.4.50.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.50.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.50.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE	Return value

##### 2.4.50.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LDATE	Value to add to queue

**2.4.50.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.50.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.50.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE	Return value

### 2.4.51 FIFO\_C\_Dt

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.51.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.51.0.2 Incode documentation

Developed entirely with methods

#### 2.4.51.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.51.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.51.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DT	Return value

##### 2.4.51.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DT	Value to add to queue

**2.4.51.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.51.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.51.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DT	Return value

### 2.4.52 FIFO\_C\_Ldt

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.52.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LDATE_AND_TIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.52.0.2 Incode documentation

Developed entirely with methods

#### 2.4.52.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.52.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.52.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE_AND_TIME	Return value

##### 2.4.52.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LDATE_AND_TIME	Value to add to queue



**2.4.52.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.52.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.52.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE_AND_TIME	Return value

### 2.4.53 FIFO\_C\_Tod

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.53.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO TOD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.53.0.2 Incode documentation

Developed entirely with methods

#### 2.4.53.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.53.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.53.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TOD	Return value

##### 2.4.53.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	TOD	Value to add to queue

**2.4.53.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.53.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.53.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TOD	Return value

### 2.4.54 FIFO\_C\_Ltod

Circular FIFO buffer. Allows Enqueueing when there is a free space in front of the tail

#### 2.4.54.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LTIME_OF_DAY	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.54.0.2 Incode documentation

Developed entirely with methods

#### 2.4.54.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.54.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.54.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME_OF_DAY	Return value

##### 2.4.54.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LTIME_OF_DAY	Value to add to queue

**2.4.54.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.54.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.54.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME_OF_DAY	Return value

### 2.4.55 FIFO\_FWFT\_Bool

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

#### 2.4.55.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO BOOL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.55.0.2 Incode documentation Developed entirely with methods

#### 2.4.55.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

#### 2.4.55.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.55.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BOOL	Return value

##### 2.4.55.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	BOOL	Value to add to queue

**2.4.55.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.55.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.55.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BOOL	Return value

### 2.4.56 FIFO\_FWFT\_Byte

First word fall through FIFO buffer. After Dequeueing the whole array is shifted one space to the left so the next element is always at the beginning

#### 2.4.56.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO BYTE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.56.0.2 Incode documentation Developed entirely with methods

#### 2.4.56.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

#### 2.4.56.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.56.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BYTE	Return value

##### 2.4.56.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	BYTE	Value to add to queue



**2.4.56.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.56.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.56.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BYTE	Return value

## 2.4.57 FIFO\_FWFT\_Word

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

### 2.4.57.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO WORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.57.0.2 Incode documentation Developed entirely with methods

#### 2.4.57.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

### 2.4.57.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.57.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	WORD	Return value

#### 2.4.57.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	WORD	Value to add to queue

**2.4.57.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.57.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.57.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	WORD	Return value

### 2.4.58 FIFO\_FWFT\_Dword

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

#### 2.4.58.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DWORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.58.0.2 Incode documentation Developed entirely with methods

#### 2.4.58.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

#### 2.4.58.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.58.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DWORD	Return value

##### 2.4.58.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DWORD	Value to add to queue

**2.4.58.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.58.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.58.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DWORD	Return value

### 2.4.59 FIFO\_FWFT\_Lword

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

#### 2.4.59.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LWORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.59.0.2 Incode documentation Developed entirely with methods

#### 2.4.59.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

#### 2.4.59.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.59.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LWORD	Return value

##### 2.4.59.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LWORD	Value to add to queue

**2.4.59.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.59.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.59.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LWORD	Return value

## 2.4.60 FIFO\_FWFT\_Sint

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

### 2.4.60.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO SINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.60.0.2 Incode documentation Developed entirely with methods

### 2.4.60.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

### 2.4.60.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.60.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	SINT	Return value

#### 2.4.60.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	SINT	Value to add to queue



**2.4.60.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.60.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.60.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	SINT	Return value

### 2.4.61 FIFO\_FWFT\_Usint

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

#### 2.4.61.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO USINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.61.0.2 Incode documentation Developed entirely with methods

#### 2.4.61.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

#### 2.4.61.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.61.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	USINT	Return value

##### 2.4.61.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	USINT	Value to add to queue

**2.4.61.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.61.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.61.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	USINT	Return value

### 2.4.62 FIFO\_FWFT\_Uint

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

#### 2.4.62.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO UINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.62.0.2 Incode documentation Developed entirely with methods

#### 2.4.62.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

#### 2.4.62.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.62.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UINT	Return value

##### 2.4.62.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	UINT	Value to add to queue

**2.4.62.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.62.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.62.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UINT	Return value

### 2.4.63 FIFO\_FWFT\_Dint

First word fall through FIFO buffer. After Dequeueing the whole array is shifted one space to the left so the next element is always at the beginning

#### 2.4.63.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.63.0.2 Incode documentation Developed entirely with methods

#### 2.4.63.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

#### 2.4.63.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.63.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DINT	Return value

##### 2.4.63.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DINT	Value to add to queue

**2.4.63.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.63.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.63.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DINT	Return value

## 2.4.64 FIFO\_FWFT\_Udint

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

### 2.4.64.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO UDINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.64.0.2 Incode documentation Developed entirely with methods

#### 2.4.64.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

### 2.4.64.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.64.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UDINT	Return value

#### 2.4.64.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	UDINT	Value to add to queue



**2.4.64.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.64.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.64.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UDINT	Return value

### 2.4.65 FIFO\_FWFT\_Lint

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

#### 2.4.65.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.65.0.2 Incode documentation Developed entirely with methods

#### 2.4.65.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

#### 2.4.65.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.65.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LINT	Return value

##### 2.4.65.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LINT	Value to add to queue

**2.4.65.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.65.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.65.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LINT	Return value

### 2.4.66 FIFO\_FWFT\_Ulint

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

#### 2.4.66.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO ULINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.66.0.2 Incode documentation Developed entirely with methods

#### 2.4.66.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

#### 2.4.66.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.66.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	ULINT	Return value

##### 2.4.66.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	ULINT	Value to add to queue

**2.4.66.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.66.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.66.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	ULINT	Return value

## 2.4.67 FIFO\_FWFT\_Real

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

### 2.4.67.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO REAL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.67.0.2 Incode documentation Developed entirely with methods

#### 2.4.67.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

### 2.4.67.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.67.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	REAL	Return value

#### 2.4.67.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	REAL	Value to add to queue

**2.4.67.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.67.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.67.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	REAL	Return value

## 2.4.68 FIFO\_FWFT\_Lreal

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

### 2.4.68.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LREAL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.68.0.2 Incode documentation Developed entirely with methods

#### 2.4.68.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

### 2.4.68.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.68.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LREAL	Return value

#### 2.4.68.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LREAL	Value to add to queue



**2.4.68.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.68.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.68.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LREAL	Return value

## 2.4.69 FIFO\_FWFT\_String

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

### 2.4.69.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO STRING[None]	Externally allocated buffer. Must be in format AR- RAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.69.0.2 Incode documentation Developed entirely with methods

#### 2.4.69.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of com- ments	Lines in total	Maintainable size
0	5	22	1	31	28

### 2.4.69.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.69.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	STRING[None]	Return value

#### 2.4.69.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	STRING[None]	Value to add to queue

**2.4.69.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.69.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.69.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	STRING[None]	Return value

## 2.4.70 FIFO\_FWFT\_Wstring

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

### 2.4.70.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO wstring	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.70.0.2 Incode documentation Developed entirely with methods

#### 2.4.70.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

### 2.4.70.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.70.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	wstring	Return value

#### 2.4.70.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	wstring	Value to add to queue

**2.4.70.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.70.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.70.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	wstring	Return value

## 2.4.71 FIFO\_FWFT\_Time

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

### 2.4.71.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO TIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.71.0.2 Incode documentation Developed entirely with methods

#### 2.4.71.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

### 2.4.71.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.71.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TIME	Return value

#### 2.4.71.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	TIME	Value to add to queue

**2.4.71.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.71.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.71.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TIME	Return value

## 2.4.72 FIFO\_FWFT\_Ltime

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

### 2.4.72.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LTIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.72.0.2 Incode documentation Developed entirely with methods

#### 2.4.72.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

### 2.4.72.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.72.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME	Return value

#### 2.4.72.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LTIME	Value to add to queue



**2.4.72.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.72.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.72.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME	Return value

### 2.4.73 FIFO\_FWFT\_Date

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

#### 2.4.73.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DATE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.73.0.2 Incode documentation Developed entirely with methods

#### 2.4.73.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

#### 2.4.73.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.73.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DATE	Return value

##### 2.4.73.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DATE	Value to add to queue

**2.4.73.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.73.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.73.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DATE	Return value

## 2.4.74 FIFO\_FWFT\_Ldate

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

### 2.4.74.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LDATE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.74.0.2 Incode documentation Developed entirely with methods

#### 2.4.74.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

### 2.4.74.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.74.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE	Return value

#### 2.4.74.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LDATE	Value to add to queue

**2.4.74.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.74.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.74.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE	Return value

## 2.4.75 FIFO\_FWFT\_Dt

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

### 2.4.75.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.75.0.2 Incode documentation Developed entirely with methods

#### 2.4.75.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

### 2.4.75.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.75.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DT	Return value

#### 2.4.75.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DT	Value to add to queue

**2.4.75.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.75.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.75.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DT	Return value

## 2.4.76 FIFO\_FWFT\_Ldt

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

### 2.4.76.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LDATE_AND_TIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.76.0.2 Incode documentation

Developed entirely with methods

### 2.4.76.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

### 2.4.76.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.76.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE_AND_TIME	Return value

#### 2.4.76.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LDATE_AND_TIME	Value to add to queue



**2.4.76.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.76.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.76.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE_AND_TIME	Return value

## 2.4.77 FIFO\_FWFT\_Tod

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

### 2.4.77.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO TOD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.77.0.2 Incode documentation Developed entirely with methods

#### 2.4.77.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

### 2.4.77.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.77.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TOD	Return value

#### 2.4.77.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	TOD	Value to add to queue

**2.4.77.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.77.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.77.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TOD	Return value

## 2.4.78 FIFO\_FWFT\_Ltod

First word fall through FIFO buffer. After Dequeuing the whole array is shifted one space to the left so the next element is always at the beginning

### 2.4.78.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LTIME_OF_DAY	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.78.0.2 Incode documentation Developed entirely with methods

#### 2.4.78.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	22	1	31	28

### 2.4.78.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.78.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME_OF_DAY	Return value

#### 2.4.78.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LTIME_OF_DAY	Value to add to queue

**2.4.78.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.78.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

**Method Interface**

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.78.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

**Method Interface**

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME_OF_DAY	Return value

## 2.4.79 FIFO\_UP\_Bool

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

### 2.4.79.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_BOOL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.79.0.2 Incode documentation

Developed entirely with methods

### 2.4.79.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

### 2.4.79.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.79.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BOOL	Return value
VAR_OUTPUT	Priority	INT	Return priority

#### 2.4.79.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	BOOL	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.79.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.79.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.79.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BOOL	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.80 FIFO\_UP\_Byte

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.80.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_BYTE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.80.0.2 Incode documentation

Developed entirely with methods

#### 2.4.80.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.80.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.80.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BYTE	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.80.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	BYTE	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.80.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.80.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.80.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BYTE	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.81 FIFO\_UP\_Word

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.81.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_WORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.81.0.2 Incode documentation

Developed entirely with methods

#### 2.4.81.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.81.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.81.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	WORD	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.81.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	WORD	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.81.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.81.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.81.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	WORD	Return value
VAR_OUTPUT	Priority	INT	Return priority

## 2.4.82 FIFO\_UP\_Dword

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

### 2.4.82.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_DWORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.82.0.2 Incode documentation

Developed entirely with methods

### 2.4.82.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

### 2.4.82.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.82.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DWORD	Return value
VAR_OUTPUT	Priority	INT	Return priority

#### 2.4.82.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DWORD	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.82.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.82.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.82.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DWORD	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.83 FIFO\_UP\_Lword

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.83.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_LWORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.83.0.2 Incode documentation

Developed entirely with methods

#### 2.4.83.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.83.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.83.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LWORD	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.83.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LWORD	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.83.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.83.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.83.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LWORD	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.84 FIFO\_UP\_Sint

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.84.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_SINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.84.0.2 Incode documentation

Developed entirely with methods

#### 2.4.84.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.84.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.84.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	SINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.84.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	SINT	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.84.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.84.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.84.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	SINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.85 FIFO\_UP\_Usint

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.85.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_USINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.85.0.2 Incode documentation

Developed entirely with methods

#### 2.4.85.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.85.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.85.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	USINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.85.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	USINT	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.85.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.85.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.85.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	USINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.86 FIFO\_UP\_Uint

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.86.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_UINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.86.0.2 Incode documentation

Developed entirely with methods

#### 2.4.86.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.86.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.86.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.86.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	UINT	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.86.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.86.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.86.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

## 2.4.87 FIFO\_UP\_Dint

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

### 2.4.87.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_DINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.87.0.2 Incode documentation

Developed entirely with methods

### 2.4.87.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

### 2.4.87.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.87.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

#### 2.4.87.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DINT	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.87.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.87.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.87.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

## 2.4.88 FIFO\_UP\_Udint

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

### 2.4.88.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_UDINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.88.0.2 Incode documentation

Developed entirely with methods

### 2.4.88.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

### 2.4.88.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.88.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UDINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

#### 2.4.88.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

##### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	UDINT	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.88.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.88.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.88.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UDINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.89 FIFO\_UP\_Lint

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.89.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_LINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.89.0.2 Incode documentation

Developed entirely with methods

#### 2.4.89.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.89.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.89.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.89.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LINT	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.89.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.89.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.89.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.90 FIFO\_UP\_Ulint

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.90.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_ULINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.90.0.2 Incode documentation Developed entirely with methods

#### 2.4.90.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.90.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.90.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	ULINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

#### 2.4.90.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	ULINT	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.90.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.90.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.90.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	ULINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.91 FIFO\_UP\_Real

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.91.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_REAL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.91.0.2 Incode documentation

Developed entirely with methods

#### 2.4.91.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.91.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.91.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	REAL	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.91.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	REAL	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.91.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.91.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.91.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	REAL	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.92 FIFO\_UP\_Lreal

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.92.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_LREAL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.92.0.2 Incode documentation

Developed entirely with methods

#### 2.4.92.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.92.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.92.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LREAL	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.92.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	LREAL	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.92.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.92.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.92.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LREAL	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.93 FIFO\_UP\_String

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.93.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_STRING	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.93.0.2 Incode documentation

Developed entirely with methods

#### 2.4.93.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.93.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.93.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	STRING[None]	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.93.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	STRING[None]	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.93.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.93.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.93.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	STRING[None]	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.94 FIFO\_UP\_Wstring

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.94.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_WSTRING	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.94.0.2 Incode documentation

Developed entirely with methods

#### 2.4.94.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.94.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.94.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	wstring	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.94.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	wstring	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.94.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.94.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.94.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	wstring	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.95 FIFO\_UP\_Time

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.95.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_TIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.95.0.2 Incode documentation

Developed entirely with methods

#### 2.4.95.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.95.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.95.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TIME	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.95.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	TIME	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.95.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.95.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.95.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TIME	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.96 FIFO\_UP\_Ltime

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.96.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_LTIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.96.0.2 Incode documentation

Developed entirely with methods

#### 2.4.96.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.96.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.96.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.96.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	LTIME	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.96.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.96.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.96.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.97 FIFO\_UP\_Date

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.97.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_DATE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.97.0.2 Incode documentation

Developed entirely with methods

#### 2.4.97.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.97.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.97.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DATE	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.97.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DATE	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.97.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.97.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.97.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DATE	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.98 FIFO\_UP\_Ldate

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.98.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_LDATE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.98.0.2 Incode documentation

Developed entirely with methods

#### 2.4.98.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.98.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.98.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.98.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LDATE	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.98.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.98.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.98.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.99 FIFO\_UP\_Dt

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.99.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_DT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.99.0.2 Incode documentation

Developed entirely with methods

#### 2.4.99.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.99.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.99.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.99.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DT	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.99.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.99.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.99.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.100 FIFO\_UP\_Ldt

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.100.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_LDT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.100.0.2 Incode documentation

Developed entirely with methods

#### 2.4.100.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.100.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.100.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE_AND_TIME	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.100.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	LDATE_AND_TIME	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.100.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.100.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.100.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE_AND_TIME	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.101 FIFO\_UP\_Tod

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.101.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_TOD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.101.0.2 Incode documentation Developed entirely with methods

#### 2.4.101.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.101.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.101.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TOD	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.101.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	TOD	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.101.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.101.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.101.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TOD	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.102 FIFO\_UP\_Ltod

Unordered Priority FIFO buffer. Allows Enqueuing if buffer is not full. Dequeues the highest priority elements first.

#### 2.4.102.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_LTOD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.102.0.2 Incode documentation Developed entirely with methods

#### 2.4.102.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	38	10	58	45

#### 2.4.102.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.102.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME_OF_DAY	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.102.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LTIME_OF_DAY	Value to add to queue
VAR_INPUT	Priority	INT	Return priority

**2.4.102.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.102.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.102.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME_OF_DAY	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.103 FIFO\_OP\_Bool

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.103.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_BOOL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.103.0.2 Incode documentation

Developed entirely with methods

#### 2.4.103.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.103.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.103.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BOOL	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.103.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	BOOL	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.103.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.103.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.103.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BOOL	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.104 FIFO\_OP\_Byte

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.104.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_BYTE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.104.0.2 Incode documentation

Developed entirely with methods

#### 2.4.104.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.104.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.104.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BYTE	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.104.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	BYTE	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.104.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.104.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.104.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BYTE	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.105 FIFO\_OP\_Word

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.105.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_WORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.105.0.2 Incode documentation Developed entirely with methods

#### 2.4.105.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.105.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.105.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	WORD	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.105.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	WORD	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.105.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.105.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.105.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	WORD	Return value
VAR_OUTPUT	Priority	INT	Return priority

## 2.4.106 FIFO\_OP\_Dword

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

### 2.4.106.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_DWORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

### 2.4.106.0.2 Incode documentation Developed entirely with methods

### 2.4.106.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

### 2.4.106.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

#### 2.4.106.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DWORD	Return value
VAR_OUTPUT	Priority	INT	Return priority

#### 2.4.106.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DWORD	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.106.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.106.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.106.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DWORD	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.107 FIFO\_OP\_Lword

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.107.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_LWORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.107.0.2 Incode documentation

Developed entirely with methods

#### 2.4.107.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.107.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.107.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LWORD	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.107.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LWORD	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.107.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.107.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.107.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LWORD	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.108 FIFO\_OP\_Sint

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.108.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_SINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.108.0.2 Incode documentation Developed entirely with methods

#### 2.4.108.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.108.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.108.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	SINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.108.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

###### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	SINT	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.108.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.108.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.108.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	SINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.109 FIFO\_OP\_Usint

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.109.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_USINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.109.0.2 Incode documentation

Developed entirely with methods

#### 2.4.109.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.109.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.109.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	USINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.109.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	USINT	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.109.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.109.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.109.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	USINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.110 FIFO\_OP\_Uint

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.110.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_UINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.110.0.2 Incode documentation

Developed entirely with methods

#### 2.4.110.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.110.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.110.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.110.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	UINT	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.110.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.110.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.110.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.111 FIFO\_OP\_Dint

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.111.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_DINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.111.0.2 Incode documentation

Developed entirely with methods

#### 2.4.111.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.111.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.111.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.111.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DINT	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.111.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.111.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.111.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.112 FIFO\_OP\_Udint

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.112.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_UDINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.112.0.2 Incode documentation

Developed entirely with methods

#### 2.4.112.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.112.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.112.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UDINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.112.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	UDINT	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.112.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.112.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.112.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UDINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.113 FIFO\_OP\_Lint

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.113.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_LINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.113.0.2 Incode documentation

Developed entirely with methods

#### 2.4.113.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.113.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.113.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.113.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LINT	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.113.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.113.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.113.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.114 FIFO\_OP\_Ulint

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.114.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_ULINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.114.0.2 Incode documentation

Developed entirely with methods

#### 2.4.114.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.114.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.114.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	ULINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.114.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	ULINT	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.114.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.114.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.114.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	ULINT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.115 FIFO\_OP\_Real

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.115.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_REAL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.115.0.2 Incode documentation

Developed entirely with methods

#### 2.4.115.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.115.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.115.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	REAL	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.115.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	REAL	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.115.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.115.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.115.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	REAL	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.116 FIFO\_OP\_Lreal

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.116.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_LREAL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.116.0.2 Incode documentation

Developed entirely with methods

#### 2.4.116.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.116.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.116.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LREAL	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.116.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	LREAL	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.116.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.116.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.116.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LREAL	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.117 FIFO\_OP\_String

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.117.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_STRING	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.117.0.2 Incode documentation

Developed entirely with methods

#### 2.4.117.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.117.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.117.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	STRING[None]	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.117.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	STRING[None]	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.117.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.117.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.117.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	STRING[None]	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.118 FIFO\_OP\_Wstring

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.118.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_WSTRING	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.118.0.2 Incode documentation

Developed entirely with methods

#### 2.4.118.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.118.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.118.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	wstring	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.118.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	wstring	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.118.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.118.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.118.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	wstring	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.119 FIFO\_OP\_Time

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.119.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_TIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.119.0.2 Incode documentation

Developed entirely with methods

#### 2.4.119.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.119.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.119.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TIME	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.119.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	TIME	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.119.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.119.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.119.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TIME	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.120 FIFO\_OP\_Ltime

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.120.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_LTIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.120.0.2 Incode documentation

Developed entirely with methods

#### 2.4.120.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.120.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.120.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.120.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	LTIME	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.120.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.120.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.120.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.121 FIFO\_OP\_Date

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.121.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_DATE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.121.0.2 Incode documentation Developed entirely with methods

#### 2.4.121.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.121.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.121.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DATE	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.121.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DATE	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.121.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.121.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.121.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DATE	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.122 FIFO\_OP\_Ldate

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.122.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_LDATE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.122.0.2 Incode documentation

Developed entirely with methods

#### 2.4.122.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.122.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.122.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.122.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LDATE	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.122.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.122.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.122.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.123 FIFO\_OP\_Dt

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.123.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_DT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.123.0.2 Incode documentation Developed entirely with methods

#### 2.4.123.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.123.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.123.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DT	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.123.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DT	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.123.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.123.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.123.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DT	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.124 FIFO\_OP\_Ldt

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.124.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_LDT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.124.0.2 Incode documentation Developed entirely with methods

#### 2.4.124.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.124.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.124.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE_AND_TIME	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.124.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	LDATE_AND_TIME	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.124.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.124.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.124.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE_AND_TIME	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.125 FIFO\_OP\_Tod

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.125.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_TOD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.125.0.2 Incode documentation Developed entirely with methods

#### 2.4.125.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.125.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.125.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TOD	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.125.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	TOD	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.125.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.125.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.125.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TOD	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.126 FIFO\_OP\_Ltod

Ordered Priority FIFO buffer. Allows Enqueing if buffer is not full. Dequeues the highest priority elements first. Buffer is ordered from lowest to highest priority. e.g. 0 = Lowest priority

#### 2.4.126.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO dt-FIFO_P_Element_LTOD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.126.0.2 Incode documentation Developed entirely with methods

#### 2.4.126.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	37	6	53	43

#### 2.4.126.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.126.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME_OF_DAY	Return value
VAR_OUTPUT	Priority	INT	Return priority

##### 2.4.126.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LTIME_OF_DAY	Value to add to queue
VAR_INPUT	Priority	INT	Priority to add to queue

**2.4.126.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.126.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.126.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME_OF_DAY	Return value
VAR_OUTPUT	Priority	INT	Return priority

### 2.4.127 LIFO\_Bool

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.127.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO BOOL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.127.0.2 Incode documentation Developed entirely with methods

#### 2.4.127.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.127.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.127.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BOOL	Return value

##### 2.4.127.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	BOOL	Value to add to the stack

**2.4.127.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.127.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.127.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BOOL	Return value

### 2.4.128 LIFO\_Byte

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.128.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO BYTE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.128.0.2 Incode documentation Developed entirely with methods

#### 2.4.128.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.128.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.128.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BYTE	Return value

##### 2.4.128.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	BYTE	Value to add to the stack

**2.4.128.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.128.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.128.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BYTE	Return value

### 2.4.129 LIFO\_Word

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.129.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO WORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.129.0.2 Incode documentation Developed entirely with methods

#### 2.4.129.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.129.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.129.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	WORD	Return value

##### 2.4.129.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	WORD	Value to add to the stack

**2.4.129.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.129.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.129.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	WORD	Return value

### 2.4.130 LIFO\_Dword

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.130.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DWORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.130.0.2 Incode documentation Developed entirely with methods

#### 2.4.130.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.130.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.130.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DWORD	Return value

##### 2.4.130.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DWORD	Value to add to the stack

**2.4.130.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.130.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.130.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DWORD	Return value

### 2.4.131 LIFO\_Lword

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.131.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LWORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.131.0.2 Incode documentation Developed entirely with methods

#### 2.4.131.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.131.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.131.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LWORD	Return value

##### 2.4.131.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LWORD	Value to add to the stack

**2.4.131.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.131.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.131.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LWORD	Return value

### 2.4.132 LIFO\_Sint

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.132.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO SINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.132.0.2 Incode documentation Developed entirely with methods

#### 2.4.132.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.132.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.132.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	SINT	Return value

##### 2.4.132.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	SINT	Value to add to the stack

**2.4.132.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.132.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.132.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	SINT	Return value

### 2.4.133 LIFO\_Usint

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.133.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO USINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.133.0.2 Incode documentation Developed entirely with methods

#### 2.4.133.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.133.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.133.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	USINT	Return value

##### 2.4.133.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	USINT	Value to add to the stack

**2.4.133.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.133.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.133.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	USINT	Return value

### 2.4.134 LIFO\_Uint

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.134.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO UINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.134.0.2 Incode documentation Developed entirely with methods

#### 2.4.134.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.134.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.134.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UINT	Return value

##### 2.4.134.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	UINT	Value to add to the stack

**2.4.134.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.134.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.134.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UINT	Return value

### 2.4.135 LIFO\_Dint

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.135.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.135.0.2 Incode documentation Developed entirely with methods

#### 2.4.135.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.135.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.135.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DINT	Return value

##### 2.4.135.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DINT	Value to add to the stack

**2.4.135.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.135.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.135.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DINT	Return value

### 2.4.136 LIFO\_Udint

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.136.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO UDINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.136.0.2 Incode documentation Developed entirely with methods

#### 2.4.136.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.136.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.136.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UDINT	Return value

##### 2.4.136.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	UDINT	Value to add to the stack

**2.4.136.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.136.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.136.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	UDINT	Return value

### 2.4.137 LIFO\_Lint

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.137.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.137.0.2 Incode documentation Developed entirely with methods

#### 2.4.137.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.137.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.137.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LINT	Return value

##### 2.4.137.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LINT	Value to add to the stack

**2.4.137.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.137.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.137.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LINT	Return value

### 2.4.138 LIFO\_Ulint

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.138.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO ULINT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.138.0.2 Incode documentation Developed entirely with methods

#### 2.4.138.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.138.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.138.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	ULINT	Return value

##### 2.4.138.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	ULINT	Value to add to the stack

**2.4.138.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.138.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.138.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	ULINT	Return value

### 2.4.139 LIFO\_Real

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.139.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO REAL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.139.0.2 Incode documentation Developed entirely with methods

#### 2.4.139.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.139.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.139.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	REAL	Return value

##### 2.4.139.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	REAL	Value to add to the stack

**2.4.139.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.139.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.139.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	REAL	Return value

### 2.4.140 LIFO\_Lreal

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.140.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LREAL	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.140.0.2 Incode documentation Developed entirely with methods

#### 2.4.140.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.140.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.140.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LREAL	Return value

##### 2.4.140.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	LREAL	Value to add to the stack

**2.4.140.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.140.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.140.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LREAL	Return value

### 2.4.141 LIFO\_String

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.141.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO STRING[None]	Externally allocated buffer. Must be in format AR- RAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.141.0.2 Incode documentation Developed entirely with methods

#### 2.4.141.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of com- ments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.141.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.141.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	STRING[None]	Return value

##### 2.4.141.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	STRING[None]	Value to add to the stack

**2.4.141.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.141.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.141.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	STRING[None]	Return value

### 2.4.142 LIFO\_Wstring

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.142.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO wstring	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.142.0.2 Incode documentation Developed entirely with methods

#### 2.4.142.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.142.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.142.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	wstring	Return value

##### 2.4.142.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	wstring	Value to add to the stack

**2.4.142.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.142.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.142.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	wstring	Return value

### 2.4.143 LIFO\_Time

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.143.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO TIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.143.0.2 Incode documentation Developed entirely with methods

#### 2.4.143.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.143.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.143.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TIME	Return value

##### 2.4.143.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	TIME	Value to add to the stack

**2.4.143.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.143.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.143.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TIME	Return value

### 2.4.144 LIFO\_Ltime

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.144.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LTIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.144.0.2 Incode documentation Developed entirely with methods

#### 2.4.144.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.144.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.144.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME	Return value

##### 2.4.144.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	LTIME	Value to add to the stack

**2.4.144.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.144.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.144.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME	Return value

### 2.4.145 LIFO\_Date

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.145.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DATE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.145.0.2 Incode documentation Developed entirely with methods

#### 2.4.145.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.145.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.145.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DATE	Return value

##### 2.4.145.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DATE	Value to add to the stack

**2.4.145.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.145.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.145.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DATE	Return value

### 2.4.146 LIFO\_Ldate

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.146.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LDATE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.146.0.2 Incode documentation Developed entirely with methods

#### 2.4.146.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.146.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.146.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE	Return value

##### 2.4.146.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LDATE	Value to add to the stack

**2.4.146.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.146.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.146.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE	Return value

### 2.4.147 LIFO\_Dt

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.147.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DT	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.147.0.2 Incode documentation Developed entirely with methods

#### 2.4.147.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.147.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.147.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DT	Return value

##### 2.4.147.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	DT	Value to add to the stack

**2.4.147.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.147.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.147.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DT	Return value

### 2.4.148 LIFO\_Ldt

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.148.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LDATE_AND_TIME	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.148.0.2 Incode documentation Developed entirely with methods

#### 2.4.148.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.148.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.148.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE_AND_TIME	Return value

##### 2.4.148.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface



Direction	Name	Type	Documentation
VAR_INPUT	Value	LDATE_AND_TIME	Value to add to the stack

**2.4.148.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.148.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.148.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LDATE_AND_TIME	Return value

### 2.4.149 LIFO\_Tod

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.149.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO TOD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.149.0.2 Incode documentation Developed entirely with methods

#### 2.4.149.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.149.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.149.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TOD	Return value

##### 2.4.149.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	TOD	Value to add to the stack

**2.4.149.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.149.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.149.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	TOD	Return value

### 2.4.150 LIFO\_Ltod

LIFO / Stack buffer. Pushing places new value on top of the stack and popping removes the element from the top.

#### 2.4.150.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LTIME_OF_DAY	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.150.0.2 Incode documentation Developed entirely with methods

#### 2.4.150.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 1

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	27

#### 2.4.150.1 Methods

- Pop
- Push
- isEmpty
- isFull
- Peek

##### 2.4.150.1.1 Method Pop : BOOL Returns the top LIFO element stored in the buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME_OF_DAY	Return value

##### 2.4.150.1.2 Method Push : BOOL Inserts new value on the top of the LIFO buffer

###### Method Interface

Direction	Name	Type	Documentation
VAR_INPUT	Value	LTIME_OF_DAY	Value to add to the stack

**2.4.150.1.3 Method isEmpty : BOOL** Returns TRUE if LIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.150.1.4 Method isFull : BOOL** Returns TRUE if LIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.150.1.5 Method Peek : UINT** Returns the index of the next element to Pop. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LTIME_OF_DAY	Return value

### 2.4.151 FIFO\_C\_8b

Circular FIFO buffer of generic type that works with 8 bit values (based on BYTE). Allows Enqueuing when there is a free space in front of the tail. Strings and data structures are not supported.

#### 2.4.151.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO BYTE	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.151.0.2 Incode documentation Developed entirely with methods

#### 2.4.151.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.151.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.151.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	BYTE	Return value that has to be casted to correct destination type

##### 2.4.151.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR.INPUT	Value	BYTE	Value to add to queue

**2.4.151.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.151.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.151.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR.OUTPUT	Value	BYTE	Return value that has to be casted to correct destination type

### 2.4.152 FIFO\_C\_16b

Circular FIFO buffer of generic type that works with 16 bit values (based on WORD). Allows Enqueuing when there is a free space in front of the tail. Strings and data structures are not supported.

#### 2.4.152.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO WORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.152.0.2 Incode documentation Developed entirely with methods

#### 2.4.152.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.152.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.152.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	WORD	Return value that has to be casted to correct destination type

##### 2.4.152.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer



#### Method Interface

Direction	Name	Type	Documentation
VAR.INPUT	Value	WORD	Value to add to queue

**2.4.152.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.152.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.152.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR.OUTPUT	Value	WORD	Return value that has to be casted to correct destination type

### 2.4.153 FIFO\_C\_32b

Circular FIFO buffer of generic type that works with 32 bit values (based on DWORD). Allows Enqueuing when there is a free space in front of the tail. Strings and data structures are not supported.

#### 2.4.153.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO DWORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.153.0.2 Incode documentation Developed entirely with methods

#### 2.4.153.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.153.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.153.1.1 Method Dequeue : BOOL Returns the next FIFO element stored in the buffer

##### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	DWORD	Return value that has to be casted to correct destination type

##### 2.4.153.1.2 Method Enqueue : BOOL Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR.INPUT	Value	WORD	Value to add to queue

**2.4.153.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.153.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.153.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR.OUTPUT	Value	DWORD	Return value that has to be casted to correct destination type

### 2.4.154 FIFO\_C\_64b

Circular FIFO buffer of generic type that works with 64 bit values (based on LWORD). Allows Enqueuing when there is a free space in front of the tail. Strings and data structures are not supported.

#### 2.4.154.0.1 Interface

Direction	Name	Type	Documentation
VAR_INPUT	Buffer	POINTER TO LWORD	Externally allocated buffer. Must be in format ARRAY[0..N]. ! Block doesn't check for Null pointer
VAR_INPUT	Size	UINT	Size/Max elements of the buffer
VAR_OUTPUT	NrElements	UINT	Number of elements in the FIFO

#### 2.4.154.0.2 Incode documentation

Developed entirely with methods

#### 2.4.154.0.3 Metrics

- VAR\_INPUT : 2
- VAR\_OUTPUT : 1
- VAR : 2

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	5	20	1	27	28

#### 2.4.154.1 Methods

- Dequeue
- Enqueue
- isEmpty
- isFull
- Peek

##### 2.4.154.1.1 Method Dequeue : BOOL

Returns the next FIFO element stored in the buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR_OUTPUT	Value	LWORD	Return value that has to be casted to correct destination type

##### 2.4.154.1.2 Method Enqueue : BOOL

Inserts new value into the FIFO buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR.INPUT	Value	LWORD	Value to add to queue

**2.4.154.1.3 Method isEmpty : BOOL** Returns TRUE if FIFO is empty. Reading not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.154.1.4 Method isFull : BOOL** Returns TRUE if FIFO is full. Inserting not allowed

#### Method Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**2.4.154.1.5 Method Peek : UINT** Returns the index of the next element to Dequeue. Output paramter holds the element without removing it from the Buffer

#### Method Interface

Direction	Name	Type	Documentation
VAR.OUTPUT	Value	LWORD	Return value that has to be casted to correct destination type

## Chapter 3

# Documentation for namespace CODESYS\_Control\_Win\_V3\_x64

### 3.1 Namespace index

- **Programs**

- FIFO\_S\_ManTest
- FIFO\_C\_ManTest
- FIFO\_FWFT\_ManTest
- FIFO\_UP\_ManTest
- LIFO\_ManTest
- FIFO\_C\_UnitTest
- FIFO\_S\_UnitTest
- FIFO\_FWFT\_UnitTest
- FIFO\_UP\_UnitTest
- FIFO\_OP\_UnitTest
- LIFO\_UnitTest
- FIFO\_OP\_ManTest
- FIFO\_C\_xxb\_UnitTest

## 3.2 Metrics

Data types	Programs	Function Blocks	Functions	Classes
0	13	0	0	0

Lines of code	Lines of comments	Lines in total	Maintainable size
419	111	656	634

## 3.3 Programs

### 3.3.1 FIFO\_S\_ManTest

Manual tests for function block FIFO\_S\_Int

#### 3.3.1.0.1 Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

#### 3.3.1.0.2 Metrics

- VAR : 14

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	0	23	3	31	37



### 3.3.2 FIFO\_C\_ManTest

Manual tests for function block FIFO\_C.Int

#### 3.3.2.0.1 Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

#### 3.3.2.0.2 Metrics

- VAR : 14

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	0	26	3	34	40

### 3.3.3 FIFO\_FWFT\_ManTest

Manual tests for function block FIFO\_FWFT\_Int

#### 3.3.3.0.1 Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

#### 3.3.3.0.2 Metrics

- VAR : 14

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	0	24	3	32	38

### 3.3.4 FIFO\_UP\_ManTest

Manual tests for function block FIFO\_UP\_Int

#### 3.3.4.0.1 Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

#### 3.3.4.0.2 Metrics

- VAR : 17

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	0	26	3	34	43

### 3.3.5 LIFO\_ManTest

Manual tests for function block LIFO\_Int

#### 3.3.5.0.1 Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

#### 3.3.5.0.2 Metrics

- VAR : 14

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	0	23	3	31	37

### 3.3.6 FIFO\_C\_UnitTest

Automated unit tests for function block FIFO\_C\_Int

#### 3.3.6.0.1 Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**3.3.6.0.2 Incode documentation** List of tests Test1: Fifo returns TRUE on successfull enqueue Test2: Fifo fills three values correctly Test3: Fifo reports FULL when full Test4: Fifo reports FALSE on unsuccesfull enqueue Test5: Fifo returns TRUE on successfull dequeue Test6: Fifo dequeues values correctly Test7: Fifo can count correctly Test8: Fifo reports EMPTY when EMPTY Test9: Fifo reports FALSE on unsuccesfull dequeue Test10: Fifo circulates to index 0 after being full and one item dequeued Test11: Peek functionality: Should return the next element but not remove it, it should report correct index of the buffer

#### 3.3.6.0.3 Metrics

- VAR : 17

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	0	36	13	62	53

### 3.3.7 FIFO\_S\_UnitTest

Automated unit tests for function block FIFO\_S\_Int

#### 3.3.7.0.1 Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**3.3.7.0.2 Incode documentation** List of tests Test1: Fifo returns TRUE on successfull enqueue Test2: Fifo fills three values correctly Test3: Fifo reports FULL when full Test4: Fifo reports FALSE on unsuccesfull enqueue Test5: Fifo returns TRUE on successfull dequeue Test6: Fifo dequeues values correctly Test7: Fifo can count correctly Test8: Fifo reports EMPTY when EMPTY Test9: Fifo reports FALSE on unsuccesfull dequeue Test10: Fifo prevents enqueue if its full Test11: Able to enqueue again only after empty flag Test12: Peek functionality: Should return the next element but not remove it, it should report correct index of the buffer

#### 3.3.7.0.3 Metrics

- VAR : 19

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	0	44	14	73	63

### 3.3.8 FIFO\_FWFT\_UnitTest

Automated unit tests for function block FIFO\_FWFT\_Int

#### 3.3.8.0.1 Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**3.3.8.0.2 Incode documentation** List of tests Test1: Fifo returns TRUE on successfull enqueue Test2: Fifo fills three values correctly Test3: Fifo reports FULL when full Test4: Fifo reports FALSE on unsuccesfull enqueue Test5: Fifo returns TRUE on successfull dequeue Test6: Fifo dequeues values correctly Test7: Fifo can count correctly Test8: Fifo reports EMPTY when EMPTY Test9: Fifo reports FALSE on unsuccesfull dequeue Test10: Fifo first word fall trough is working Test11: Peek functionality: Should return the next element but not remove it, it should report correct index of the buffer

#### 3.3.8.0.3 Metrics

- VAR : 17

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	0	35	13	61	52

### 3.3.9 FIFO\_UP\_UnitTest

Automated unit tests for function block FIFO\_UP\_Int

#### 3.3.9.0.1 Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**3.3.9.0.2 Incode documentation** List of tests Test1: Fifo returns TRUE on successfull enqueue Test2: Fifo fills three values correctly Test3: Fifo reports FULL when full Test4: Fifo reports FALSE on unsuccesfull enqueue Test5: Fifo returns TRUE on successfull dequeue Test6: Fifo dequeues values correctly Test7: Fifo can count correctly Test8: Fifo reports EMPTY when EMPTY Test9: Fifo reports FALSE on unsuccesfull dequeue Test10: Fifo returns the highest priority value first Test11: Peek functionality: Should return the next element but not remove it, it should report correct index of the buffer

#### 3.3.9.0.3 Metrics

- VAR : 18

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	0	37	13	63	55



### 3.3.10 FIFO\_OP\_UnitTest

Automated unit tests for function block FIFO\_OP\_Int

#### 3.3.10.0.1 Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**3.3.10.0.2 Incode documentation** List of tests Test1: Fifo returns TRUE on successfull enqueue Test2: Fifo fills three values correctly Test3: Fifo reports FULL when full Test4: Fifo reports FALSE on unsuccesfull enqueue Test5: Fifo returns TRUE on successfull dequeue Test6: Fifo dequeues values correctly Test7: Fifo can count correctly Test8: Fifo reports EMPTY when EMPTY Test9: Fifo reports FALSE on unsuccesfull dequeue Test10: Fifo returns the highest priority value first Test11: Fifo has ordered the buffer correctly according to priority. The buffer is lowest : highest = 0 : N Test12: Peek functionality: Should return the next element but not remove it, it should report correct index of the buffer

#### 3.3.10.0.3 Metrics

- VAR : 19

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	0	38	14	66	57

### 3.3.11 LIFO\_UnitTest

Automated unit tests for function block LIFO\_Int

#### 3.3.11.0.1 Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**3.3.11.0.2 Incode documentation** List of tests Test1: Lifo returns TRUE on successfull Push Test2: Lifo fills three values correctly Test3: Lifo reports FULL when full Test4: Lifo reports FALSE on unsuccesfull Push Test5: Lifo returns TRUE on successfull Pop Test6: Lifo dequeues values correctly Test7: Lifo can count correctly Test8: Lifo reports EMPTY when EMPTY Test9: Lifo reports FALSE on unsuccesfull Pop Test10: Peek functionality: Should return the next element but not remove it, it should report correct index of the buffer

#### 3.3.11.0.3 Metrics

- VAR : 16

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	0	30	12	54	46

### 3.3.12 FIFO\_OP\_ManTest

Manual tests for function block FIFO\_OP\_Int

#### 3.3.12.0.1 Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

#### 3.3.12.0.2 Metrics

- VAR : 17

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	0	26	3	34	43

### 3.3.13 FIFO\_C\_xxb\_UnitTest

Automated unit tests for function block FIFO\_C\_xxb

#### 3.3.13.0.1 Interface

Direction	Name	Type	Documentation
-----------	------	------	---------------

**3.3.13.0.2 Incode documentation** List of tests Test1: Fifo returns TRUE on successfull enqueue Test2: Fifo fills three values correctly Test3: Fifo reports FULL when full Test4: Fifo reports FALSE on unsuccesfull enqueue Test5: Fifo returns TRUE on successfull dequeue Test6: Fifo dequeues values correctly Test7: Fifo can count correctly Test8: Fifo reports EMPTY when EMPTY Test9: Fifo reports FALSE on unsuccesfull dequeue Test10: Fifo circulates to index 0 after being full and one item dequeued Test11: Peek functionality: Should return the next element but not remove it, it should report correct index of the buffer Test12: 16bit handling: Multiple types should be returned

#### 3.3.13.0.3 Metrics

- VAR : 19

Actions	Methods	Lines of code	Lines of comments	Lines in total	Maintainable size
0	0	51	14	81	70