

Project Documentation

File: BallBeam_PID_V31_Final.project

Date: 16.05.2023

Profile: ?

Table of contents

1 Device: Device	3
1.1 PLC Logic: Plc Logic	3
1.1.1 Application: Application	4
1.2 Device: Realtime_Data	70
1.2.1 Device: motion_axis_Axis_Y_cmd_input	70
1.2.2 Device: motion_cmd_input	71
1.2.3 Device: motion_state_functions_somo_rt_data	72
1.2.4 Device: ethercat_master_instances_ethercatmaster	77
2 GlobalTextList: GlobalTextList	78

1 Device: Device

Users and Groups

Users:

Groups

Access Rights

View

Modify

Execute

Add/remove children

Symbol Rights

Information

Name: ctrlX CORE ARM64
Vendor: Bosch Rexroth AG
Categories: PLCs
Type: 4096
ID: 1001 0080
Version: 1.18.20.0
Order number: dependent on customer specific configuration
Description: ctrlX CORE on processor platform "ARM 64 bit"
Eingänge: 8192
Ausgänge: 8192
Merker: 8192
Remanenter Speicher: default 64KB auf 128KB konfigurierbar

1.1 PLC Logic: Plc Logic

1.1.1 Application: Application

1.1.1.1 Folder: CheckFunctions

1.1.1.1.1 POU: CheckDivDInt

```
1      // Short description: Test function to increase machine safety by detecting
      // unauthorized division by zero of INT, UINT, DINT, UDINT, WORD and DWORD
      // variables
2      //
3      // Functional description: This function will be called implicitly before the
      // division takes places.
4      // The divisor will be checked for 0.
5      // In case of 0, an error is entered in the plc logger and the application is
      // stopped due to an access violation to avoid machine damage.
6      // In this case, check the callstack (menu Debug/Callstack) and modify the
      // faulty logic.
7      //
8      // Handling particularities: This function increases the cycle time of all tasks
      // using divisions.
9      // The increase depends on the number of the divisions.
10     // After inserting/removing this function, no online-change will be possible!
11     // A complete download will be necessary!
12     // The implementation has to be adjusted to your application/machine.
13     // In case of a division by zero, the PLC-Task will generate an access violation
      // which leads to an application stop.
14     // Please verify whether this default behaviour can damage your machine or lead
      // to a complex recommissioning.
15     // In this case, adapt the error reaction to your application.
16     //
17     // If your application isn't too time-critical, you should leave this function
      // while commissioning the machine and production in your application.
18     // If you remove this function, ensure that your code is faultless.
19     //
20     // Do not change the declaration of this function! Add only local variables.
21     // Do not change the attributes above the function header!
22     {attribute 'check_div_int32'}
23     {attribute 'no_check'}
24     {attribute 'signature_flag' := '1073741824'}
25     FUNCTION CheckDivDInt : DINT
26     VAR_INPUT
27         divisor : DINT ; // Divisor of the division
```

1.1.1.1.1 POU: CheckDivDInt

```
28  END_VAR
29  VAR
30      ptNull : POINTER TO BYTE := 0; // Nullpointer for triggering an access
        violation
31      strError : STRING;           // Diagnostic text for the PLC logger entry
32  END_VAR
33
1
2      (* The implementation has to be adjusted to your application/machine. *)
3      (* This is an exemplary implementation of the function. *)
4      (* Per default, an access violation is generated to avoid machine damage *)
5      (* Check the Call Stack (menu Debug/Callstack) and modify the faulty logic!
        *)
6
7      (* ***** *)
8      IF divisor = 0 THEN
9          strError := 'CheckDivDInt: Invalid division by 0 - check the callstack';
10         // Default implementation for APPLICATION WILL BE STOPPED:
11         // Set an error into the plc logger
12         LogAdd2 ( hLogger := CmpLog . LOG_STD_LOGGER , udiCmpID := 16#1000 , udiClassID :=
            LogClass . LOG_ERROR , udiErrorID := 1 , udiInfoID := 1 , pszInfo := strError );
13         // Triggering an access violation --> Application will be stopped
14         ptNull ^ := 0; (* Triggering an access violation *)
15
16         // // Example for APPLICATION WILL NOT BE STOPPED:
17         // // Set the divisor to 1
18         // // --> no access violation (div by zero) will be triggered
19         // // --> keep PLC-task running
20         // // Set a warning into the plc logger
21         // // NOTE: There is still an implementation error in your application code!
22         //
23         // LogAdd2(hLogger:= CmpLog.LOG_STD_LOGGER,udiCmpID:= 16#1000,udiClassID:=
            LogClass.LOG_WARNING,udiErrorID:= 1,udiInfoID:= 1,pszInfo:= strError);
24         // CheckDivDInt := 1; //Division by 1 instead of zero
25         // return;
26
27     ELSE // division is ok
28         CheckDivDInt := divisor;
29     END_IF;
30
```

1.1.1.1.2 POU: CheckDivLInt

```
1      // Short description: Test function to increase machine safety by detecting
      // unauthorized division by zero of LINT, ULINT and LWORD variables
2      //
3      // Functional description: This function will be called implicitly before the
      // division takes places.
4      // The divisor will be checked for 0.
5      // In case of 0, an error is entered in the plc logger and the application is
      // stopped due to an access violation to avoid machine damage.
6      // In this case, check the callstack (menu Debug/Callstack) and modify the
      // faulty logic.
7      //
8      // Handling particularities: This function increases the cycle time of all tasks
      // using divisions.
9      // The increase depends on the number of the divisions.
10     // After inserting/removing this function, no online-change will be possible!
11     // A complete download will be necessary!
12     // The implementation has to be adjusted to your application/machine.
13     // In case of a division by zero, the PLC-Task will generate an access violation
      // which leads to an application stop.
14     // Please verify whether this default behaviour can damage your machine or lead
      // to a complex recommissioning.
15     // In this case, adapt the error reaction to your application.
16     //
17     // If your application isn't too time-critical, you should leave this function
      // while commissioning the machine and production in your application.
18     // If you remove this function, ensure that your code is faultless.
19     //
20     // Do not change the declaration of this function! Add only local variables.
21     // Do not change the attributes above the function header!
22     {attribute 'check_div_int64'}
23     {attribute 'no_check'}
24     {attribute 'signature_flag' := '1073741824'}
25     FUNCTION CheckDivLInt : LINT
26     VAR_INPUT
27         divisor : LINT; // Divisor of the division
28     END_VAR
29     VAR
30         ptNull : POINTER TO BYTE := 0; // Nullpointer for triggering an access
      // violation
31         strError : STRING; // Diagnostic text for the PLC logger entry
32     END_VAR
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
```

1.1.1.1.2 POU: CheckDivLInt

```
4      (* Per default, an access violation is generated to avoid machine damage *)
5      (* Check the Call Stack (menu Debug/Callstack) and modify the faulty logic!
6      *)
7
8      (* ***** *)
9      IF divisor = 0 THEN
10         strError := 'CheckDivLInt: Invalid division by 0 - check the callstack' ;
11         // Default implementation for APPLICATION WILL BE STOPPED:
12         // Set an error into the plc logger
13         LogAdd2 ( hLogger := CmpLog . LOG_STD_LOGGER , udiCmpID := 16#1000 , udiClassID :=
14                   LogClass . LOG_ERROR , udiErrorID := 1 , udiInfoID := 1 , pszInfo := strError ) ;
15         // Triggering an access violation --> Application will be stopped
16         ptNull ^ := 0 ; (* Triggering an access violation *)
17
18         // // Example for APPLICATION WILL NOT BE STOPPED:
19         // // Set the divisor to 1
20         // // --> no access violation (div by zero) will be triggered
21         // // --> keep PLC-task running
22         // // Set a warning into the plc logger
23         // // NOTE: There is still an implementation error in your application code!
24         //
25         // LogAdd2(hLogger:= CmpLog.LOG_STD_LOGGER,udiCmpID:= 16#1000,udiClassID:=
26                   LogClass.LOG_WARNING,udiErrorID:= 1,udiInfoID:= 1,pszInfo:= strError);
27         // CheckDivLInt := 1; //Division by 1 instead of zero
28         // return;
29
30     ELSE // division is ok
31         CheckDivLInt := divisor ;
32     END_IF ;
```

1.1.1.1.3 POU: CheckDivLReal

```
1      // Short description: Test function to increase machine safety by detecting
2      // unauthorized division by zero of variables of type LREAL
3      //
4      // Functional description: This function will be called implicitly before the
5      // division takes places.
6      // The divisor will be checked for 0.0.
7      // In case of 0.0, an error is entered in the plc logger and the application is
8      // stopped due to an access violation to avoid machine damage.
9      // In this case, check the callstack (menu Debug/Callstack) and modify the
10     // faulty logic.
11     //
12     // Handling particularities: This function increases the cycle time of all tasks
13     // using divisions.
14     // The increase depends on the number of the divisions.
15     // After inserting/removing this function, no online-change will be possible!
16     // A complete download will be necessary!
17     // The implementation has to be adjusted to your application/machine.
```

1.1.1.1.3 POU: CheckDivLReal

```
13      // In case of a division by zero, the PLC-Task will generate an access violation
14      // which leads to an application stop.
15      // Please verify whether this default behaviour can damage your machine or lead
16      // to a complex recommissioning.
17      // In this case, adapt the error reaction to your application.
18      //
19      // If your application isn't too time-critical, you should leave this function
20      // while commissioning the machine and production in your application.
21      // If you remove this function, ensure that your code is faultless.
22      //
23      // Do not change the declaration of this function! Add only local variables.
24      // Do not change the attributes above the function header!
25      {attribute 'check_div_real64'}
26      {attribute 'no_check'}
27      {attribute 'signature_flag' := '1073741824'}
28      FUNCTION CheckDivLReal : LREAL
29      VAR_INPUT
30          divisor : LREAL; // Divisor of the division
31      END_VAR
32      VAR
33          ptNull : POINTER TO BYTE := 0; // Nullpointer for triggering an access
34          violation
35          strError : STRING; // Diagnostic text for the PLC logger entry
36      END_VAR
```

```
1
2      (*****
3          (* The implementation has to be adjusted to your application/machine. *)
4          (* This is an exemplary implementation of the function. *)
5          (* Per default, an access violation is generated to avoid machine damage *)
6          (* Check the Call Stack (menu Debug/Callstack) and modify the faulty logic!
7          *)
8          (*****
9
10      IF divisor = 0 THEN
11          strError := 'CheckDivLReal: Invalid division by 0 - check the callstack';
12          // Default implementation for APPLICATION WILL BE STOPPED:
13          // Set an error into the plc logger
14          LogAdd2 ( hLogger := CmpLog.LOG_STD_LOGGER, udiCmpID := 16#1000, udiClassID :=
15                  LogClass.LOG_ERROR, udiErrorID := 1, udiInfoID := 1, pszInfo := strError );
16          // Triggering an access violation --> Application will be stopped
17          ptNull ^ := 0; (* Triggering an access violation *)
18
19          // // Example for APPLICATION WILL NOT BE STOPPED:
20          // // Set the divisor to 1.0
21          // // --> no access violation (div by zero) will be triggered
22          // // --> keep PLC-task running
23          // // Set a warning into the plc logger
```


1.1.1.1.3 POU: CheckDivLReal

```
21      // // NOTE: There is still an implementation error in your application code!
22      //
23      // LogAdd2(hLogger:= CmpLog.LOG_STD_LOGGER,udiCmpID:= 16#1000,udiClassID:=
24      // LogClass.LOG_WARNING,udiErrorID:= 1,udiInfoID:= 1,pszInfo:= strError);
25      // CheckDivLReal := 1.0; //Division by 1 instead of zero
26      // return;
27      ELSE // division is ok
28          CheckDivLReal := divisor ;
29      END_IF ;
30
```

1.1.1.1.4 POU: CheckDivReal

```
1      // Short description: Test function to increase machine safety by detecting
2      // unauthorized division by zero of variables of type REAL.
3      // Functional description: This function will be called implicitly before the
4      // division takes places.
5      // The divisor will be checked for 0.0.
6      // In case of 0.0, an error is entered in the plc logger and the application is
7      // stopped due to an access violation to avoid machine damage.
8      // In this case, check the callstack (menu Debug/Callstack) and modify the
9      // faulty logic.
10     //
11     // Handling particularities: This function increases the cycle time of all tasks
12     // using divisions.
13     // The increase depends on the number of the divisions.
14     // After inserting/removing this function, no online-change will be possible!
15     // A complete download will be necessary!
16     // The implementation has to be adjusted to your application/machine.
17     // In case of a division by zero, the PLC-Task will generate an access violation
18     // which leads to an application stop.
19     // Please verify whether this default behaviour can damage your machine or lead
20     // to a complex recommissioning.
21     // In this case, adapt the error reaction to your application.
22     //
23     // If your application isn't too time-critical, you should leave this function
24     // while commissioning the machine and production in your application.
25     // If you remove this function, ensure that your code is faultless.
26     //
27     // Do not change the declaration of this function! Add only local variables.
28     // Do not change the attributes above the function header!
29     {attribute 'check_div_real32'}
30     {attribute 'no_check'}
31     {attribute 'signature_flag' := '1073741824'}
32     FUNCTION CheckDivReal : REAL
33     VAR_INPUT
34         divisor : REAL ; // Divisor of the division
35     END_VAR
36     VAR
```

1.1.1.1.4 POU: CheckDivReal

```
30     ptNull : POINTER TO BYTE := 0; // Nullpointer for triggering an access
      violation
31     strError : STRING;           // Diagnostic text for the PLC logger entry
32 END_VAR
33

1
  (*****
2     (* The implementation has to be adjusted to your application/machine. *)
3     (* This is an exemplary implementation of the function. *)
4     (* Per default, an access violation is generated to avoid machine damage *)
5     (* Check the Call Stack (menu Debug/Callstack) and modify the faulty logic!
      *)
6
7     (*****
8     IF divisor = 0 THEN
9         strError := 'CheckDivReal: Invalid division by 0 - check the callstack';
10        // Default implementation for APPLICATION WILL BE STOPPED:
11        // Set an error into the plc logger
12        LogAdd2 ( hLogger := CmpLog.LOG_STD_LOGGER, udiCmpID := 16#1000, udiClassID :=
          LogClass.LOG_ERROR, udiErrorID := 1, udiInfoID := 1, pszInfo := strError );
13        // Triggering an access violation --> Application will be stopped
14        ptNull ^ := 0; (* Triggering an access violation *)
15
16        // // Example for APPLICATION WILL NOT BE STOPPED:
17        // // Set the divisor to 1
18        // // --> no access violation (div by zero) will be triggered
19        // // --> keep PLC-task running
20        // // Set a warning into the plc logger
21        // // NOTE: There is still an implementation error in your application code!
22        //
23        // LogAdd2(hLogger:= CmpLog.LOG_STD_LOGGER,udiCmpID:= 16#1000,udiClassID:=
          LogClass.LOG_WARNING,udiErrorID:= 1,udiInfoID:= 1,pszInfo:= strError);
24        // CheckDivReal := 1.0; //Division by 1 instead of zero
25        // return;
26
27    ELSE // division is ok
28        CheckDivReal := divisor;
29    END_IF;
30
```

1.1.1.1.5 POU: CheckLRangeSigned

```
1      // Short description: Test function to increase machine safety by checking if
      // value is within signed range (LINT)
2      //
3      // Functional description: This function will be called implicitly before any
      // access to range signed values.
4      // In case of invalid access, an error is entered in the diagnostic logbook and
      // the application is stopped due to an access violation to avoid machine damage.
5      // In this case, check the callstack (menu Debug/Callstack) and modify the
      // faulty logic.
6      //
7      // Handling particularities: This function increases the cycle time of all tasks
      // accessing range signed values.
8      // The increase depends on the number of the read/write accesses.
9      //
10     // After inserting/removing this function, no online-change will be possible!
11     // A complete download will be necessary!
12     // The implementation has to be adjusted to your application/machine.
13     // In case of a division by zero, the PLC-Task will generate an access violation
      // which leads to an application stop.
14     // Please verify whether this default behaviour can damage your machine or lead
      // to a complex recommissioning.
15     // In this case, adapt the error reaction to your application.
16     //
17     // If your application isn't too time-critical, you should leave this function
      // while commissioning the machine and production in your application.
18     // If you remove this function, ensure that your code is faultless.
19     //
20     // Do not change the declaration of this function! Add only local variables.
21     // Do not change the attributes above the function header!
22     {attribute 'no_check'}
23     {attribute 'check_lrange_signed'}
24     {attribute 'signature_flag' := '1073741824'}
25     FUNCTION CheckLRangeSigned : LINT
26     VAR_INPUT
27         value : LINT ;
28         lower : LINT ;
29         upper : LINT ;
30     END_VAR
31     VAR
32         ptNull : POINTER TO BYTE := 0 ; // Nullpointer for triggering an access
      // violation
33         strError : STRING ;              // Diagnostic text for the PLC logger entry
34     END_VAR
35
```

```
1      (*****
```

1.1.1.1.5 POU: CheckLRangeSigned

```
2      (* The implementation has to be adjusted to your application/machine. *)
3      (* This is an exemplary implementation of the function. *)
4      (* Per default, an access violation is generated to avoid machine damage *)
5      (* Check the Call Stack (menu Debug/Callstack) and modify the faulty logic!
6      *)
7
8      (*****
9
10     IF ( value < lower ) THEN
11         strError := 'CheckLRangeSigned: Lower value ' ;
12         strError := CONCAT ( STR1 := strError , STR2 := LINT_TO_STRING ( lower ) ) ;
13         strError := CONCAT ( STR1 := strError , STR2 := ' underrun by ' ) ;
14         strError := CONCAT ( STR1 := strError , STR2 := LINT_TO_STRING ( value ) ) ;
15         // Default implementation for APPLICATION WILL BE STOPPED:
16         // Set an error into the plc logger
17         LogAdd2 ( hLogger := CmpLog.LOG_STD_LOGGER , udiCmpID := 16#1000 , udiClassID :=
18             LogClass.LOG_ERROR , udiErrorID := 1 , udiInfoID := 1 , pszInfo := strError ) ;
19         // Triggering an access violation --> Application will be stopped
20         ptNull ^ := 0 ; (* Triggering an access violation *)
21
22         // // Example for APPLICATION WILL NOT BE STOPPED:
23         // // Limit the faulty value to the lower value
24         // // Set a warning into the plc logger
25         // // NOTE: There is still an implementation error in your application code!
26         //
27         // LogAdd2(hLogger:= CmpLog.LOG_STD_LOGGER,udiCmpID:=
28             16#1000,udiClassID:= LogClass.LOG_WARNING,udiErrorID:= 1,udiInfoID:= 1,pszInfo:=
29             strError);
30         // CheckLRangeSigned := lower; //set value to lower value
31         // return;
32
33     ELSIF ( value > upper ) THEN
34         strError := 'CheckLRangeSigned: Upper value ' ;
35         strError := CONCAT ( STR1 := strError , STR2 := LINT_TO_STRING ( upper ) ) ;
36         strError := CONCAT ( STR1 := strError , STR2 := ' overrun by ' ) ;
37         strError := CONCAT ( STR1 := strError , STR2 := LINT_TO_STRING ( value ) ) ;
38
39         // Default implementation for APPLICATION WILL BE STOPPED:
40         // Set an error into the plc logger
41         LogAdd2 ( hLogger := CmpLog.LOG_STD_LOGGER , udiCmpID := 16#1000 , udiClassID :=
42             LogClass.LOG_ERROR , udiErrorID := 1 , udiInfoID := 1 , pszInfo := strError ) ;
43         // Triggering an access violation --> Application will be stopped
44         ptNull ^ := 0 ; (* Triggering an access violation *)
45
46         // // Example for APPLICATION WILL NOT BE STOPPED:
47         // // Limit the faulty value to the upper value
48         // // Set a warning into the plc logger
49         // // NOTE: There is still an implementation error in your application code!
50         //
51         // LogAdd2(hLogger:= CmpLog.LOG_STD_LOGGER,udiCmpID:= 16#1000,udiClassID:=
52             LogClass.LOG_WARNING,udiErrorID:= 1,udiInfoID:= 1,pszInfo:= strError);
```

```
46      // CheckLRangeSigned := upper; //set value to upper value
47      // return;
48
49      ELSE // value ist within valid range
50          CheckLRangeSigned := value ;
51      END_IF
52
```

1.1.1.1.6 POU: CheckLRangeUnsigned

```
1      // Short description: Test function to increase machine safety by checking if
      // value is within unsigned range (ULINT, LWORD)
2      //
3      // Functional description: This function will be called implicitly before any
      // access to range signed values.
4      // In case of invalid access, an error is entered in the diagnostic logbook and
      // the application is stopped due to an access violation to avoid machine damage.
5      // In this case, check the callstack (menu Debug/Callstack) and modify the
      // faulty logic.
6      //
7      // Handling particularities: This function increases the cycle time of all tasks
      // accessing range signed values.
8      // The increase depends on the number of the read/write accesses.
9      //
10     // After inserting/removing this function, no online-change will be possible!
11     // A complete download will be necessary!
12     // The implementation has to be adjusted to your application/machine.
13     // In case of a division by zero, the PLC-Task will generate an access violation
      // which leads to an application stop.
14     // Please verify whether this default behaviour can damage your machine or lead
      // to a complex recommissioning.
15     // In this case, adapt the error reaction to your application.
16     //
17     // If your application isn't too time-critical, you should leave this function
      // whil comissioning the machine and production in your application.
18     // If you remove this function, ensure that your code is faultless.
19     //
20     // Do not change the declaration of this function! Add only local variables.
21     // Do not change the attributes above the function header!
22     {attribute 'no_check'}
23     {attribute 'check_lrange_unsigned'}
24     {attribute 'signature_flag' := '1073741824'}
25     FUNCTION CheckLRangeUnsigned : ULINT
26     VAR_INPUT
27         value : ULINT ;
28         lower : ULINT ;
29         upper : ULINT ;
30     END_VAR
31     VAR
32         ptNull : POINTER TO BYTE := 0 ; // Nullpointer for triggering an access
      // violation
```

1.1.1.1.6 POU: CheckLRangeUnsigned

```
33      strError : STRING ;           // Diagnostic text for the PLC logger entry
34  END_VAR
35
1
2      (*****
3          (* The implementation has to be adjusted to your application/machine. *)
4          (* This is an exemplary implementation of the function. *)
5          (* Per default, an access violation is generated to avoid machine damage *)
6          (* Check the Call Stack (menu Debug/Callstack) and modify the faulty logic!
7          *)
8      (*****
9
10     IF ( value < lower ) THEN
11         strError := 'CheckLRangeUnsigned: Lower value ' ;
12         strError := CONCAT ( STR1 := strError , STR2 := ULINT_TO_STRING ( lower ) ) ;
13         strError := CONCAT ( STR1 := strError , STR2 := ' underrun by ' ) ;
14         strError := CONCAT ( STR1 := strError , STR2 := ULINT_TO_STRING ( value ) ) ;
15         // Default implementation for APPLICATION WILL BE STOPPED:
16         // Set an error into the plc logger
17         LogAdd2 ( hLogger := CmpLog . LOG_STD_LOGGER , udiCmpID := 16#1000 , udiClassID :=
18             LogClass . LOG_ERROR , udiErrorID := 1 , udiInfoID := 1 , pszInfo := strError ) ;
19         // Triggering an access violation --> Application will be stopped
20         ptNull ^ := 0 ; (* Triggering an access violation *)
21
22         // // Example for APPLICATION WILL NOT BE STOPPED:
23         // // Limit the faulty value to the lower value
24         // // Set a warning into the plc logger
25         // // NOTE: There is still an implementation error in your application code!
26         //
27         // CheckLRangeUnsigned := lower; //set value to lower value
28         // return;
29
30     ELSIF ( value > upper ) THEN
31         strError := 'CheckLRangeUnsigned: Upper value ' ;
32         strError := CONCAT ( STR1 := strError , STR2 := ULINT_TO_STRING ( upper ) ) ;
33         strError := CONCAT ( STR1 := strError , STR2 := ' overrun by ' ) ;
34         strError := CONCAT ( STR1 := strError , STR2 := ULINT_TO_STRING ( value ) ) ;
35
36         // Default implementation for APPLICATION WILL BE STOPPED:
37         // Set an error into the plc logger
38         LogAdd2 ( hLogger := CmpLog . LOG_STD_LOGGER , udiCmpID := 16#1000 , udiClassID :=
39             LogClass . LOG_ERROR , udiErrorID := 1 , udiInfoID := 1 , pszInfo := strError ) ;
40         // Triggering an access violation --> Application will be stopped
41         ptNull ^ := 0 ; (* Triggering an access violation *)
42
43         // // Example for APPLICATION WILL NOT BE STOPPED:
44         // // Limit the faulty value to the upper value
45         // // Set a warning into the plc logger
```

1.1.1.1.6 POU: CheckLRangeUnsigned

```
42      // // NOTE: There is still an implementation error in your application code!
43      //
44      // LogAdd2(hLogger:= CmpLog.LOG_STD_LOGGER,udiCmpID:= 16#1000,udiClassID:=
      LogClass.LOG_WARNING,udiErrorID:= 1,udiInfoID:= 1,pszInfo:= strError);
45      // CheckLRangeUnsigned := upper; //set value to upper value
46      // return;
47
48      ELSE // value ist within valid range
49          CheckLRangeUnsigned := value ;
50      END_IF
51
```

1.1.1.1.7 POU: CheckRangeSigned

```
1      // Short description: Test function to increase machine safety by checking if
      value is within signed range (INT, DINT)
2      //
3      // Functional description: This function will be called implicitly before any
      access to range signed values.
4      // In case of invalid access, an error is entered in the diagnostic logbook and
      the application is stopped due to an access violation to avoid machine damage.
5      // In this case, check the callstack (menu Debug/Callstack) and modify the
      faulty logic.
6      //
7      // Handling particularities: This function increases the cycle time of all tasks
      accessing range signed values.
8      // The increase depends on the number of the read/write accesses.
9      //
10     // After inserting/removing this function, no online-change will be possible!
11     // A complete download will be necessary!
12     // The implementation has to be adjusted to your application/machine.
13     // In case of a division by zero, the PLC-Task will generate an access violation
      which leads to an application stop.
14     // Please verify whether this default behaviour can damage your machine or lead
      to a complex recommissioning.
15     // In this case, adapt the error reaction to your application.
16     //
17     // If your application isn't too time-critical, you should leave this function
      while commissioning the machine and production in your application.
18     // If you remove this function, ensure that your code is faultless.
19     //
20     // Do not change the declaration of this function! Add only local variables.
21     // Do not change the attributes above the function header!
22     {attribute 'no_check'}
23     {attribute 'check_range_signed'}
24     {attribute 'signature_flag' := '1073741824'}
25     FUNCTION CheckRangeSigned : DINT
26     VAR_INPUT
27         value : DINT ;
28         lower : DINT ;
29         upper : DINT ;
```

1.1.1.1.7 POU: CheckRangeSigned

```
30  END_VAR
31  VAR
32      ptNull : POINTER TO BYTE := 0; // Nullpointer for triggering an access
        violation
33      strError : STRING;           // Diagnostic text for the PLC logger entry
34  END_VAR
35
1
2      (*****
3          (* The implementation has to be adjusted to your application/machine. *)
4          (* This is an exemplary implementation of the function. *)
5          (* Per default, an access violation is generated to avoid machine damage *)
6          (* Check the Call Stack (menu Debug/Callstack) and modify the faulty logic!
7          *)
8          (*****
9
10         IF ( value < lower ) THEN
11             strError := 'CheckRangeSigned: Lower value ';
12             strError := CONCAT ( STR1 := strError , STR2 := DINT_TO_STRING ( lower ) );
13             strError := CONCAT ( STR1 := strError , STR2 := ' underrun by ' );
14             strError := CONCAT ( STR1 := strError , STR2 := DINT_TO_STRING ( value ) );
15             // Default implementation for APPLICATION WILL BE STOPPED:
16             // Set an error into the plc logger
17             LogAdd2 ( hLogger := CmpLog . LOG_STD_LOGGER , udiCmpID := 16#1000 , udiClassID :=
18                 LogClass . LOG_ERROR , udiErrorID := 1 , udiInfoID := 1 , pszInfo := strError );
19             // Triggering an access violation --> Application will be stopped
20             ptNull ^ := 0; (* Triggering an access violation *)
21
22             // // Example for APPLICATION WILL NOT BE STOPPED:
23             // // Limit the faulty value to the lower value
24             // // Set a warning into the plc logger
25             // // NOTE: There is still an implementation error in your application code!
26             //
27             // LogAdd2(hLogger:= CmpLog.LOG_STD_LOGGER,udiCmpID:= 16#1000,udiClassID:=
28                 LogClass.LOG_WARNING,udiErrorID:= 1,udiInfoID:= 1,pszInfo:= strError);
29             // CheckRangeSigned := lower; //set value to lower value
30             // return;
31
32         ELSIF ( value > upper ) THEN
33             strError := 'CheckRangeSigned: Upper value ';
34             strError := CONCAT ( STR1 := strError , STR2 := DINT_TO_STRING ( upper ) );
35             strError := CONCAT ( STR1 := strError , STR2 := ' overrun by ' );
36             strError := CONCAT ( STR1 := strError , STR2 := DINT_TO_STRING ( value ) );
37
38             // Default implementation for APPLICATION WILL BE STOPPED:
39             // Set an error into the plc logger
40             LogAdd2 ( hLogger := CmpLog . LOG_STD_LOGGER , udiCmpID := 16#1000 , udiClassID :=
41                 LogClass . LOG_ERROR , udiErrorID := 1 , udiInfoID := 1 , pszInfo := strError );
```


1.1.1.1.7 POU: CheckRangeSigned

```
37 // Triggering an access violation --> Application will be stopped
38 ptNull ^ := 0; (* Triggering an access violation *)
39
40 // // Example for APPLICATION WILL NOT BE STOPPED:
41 // // Limit the faulty value to the upper value
42 // // Set a warning into the plc logger
43 // // NOTE: There is still an implementation error in your application code!
44 //
45 // LogAdd2(hLogger:= CmpLog.LOG_STD_LOGGER,udiCmpID:= 16#1000,udiClassID:=
46 // LogClass.LOG_WARNING,udiErrorID:= 1,udiInfoID:= 1,pszInfo:= strError);
47 // CheckRangeSigned := upper; //set value to upper value
48 // return;
49
50 ELSE // value ist within valid range
51     CheckRangeSigned := value ;
52 END_IF
```

1.1.1.1.8 POU: CheckRangeUnsigned

```
1 // Short description: Test function to increase machine safety by checking if
2 // value is within unsigned range (UINT, UDINT, WORD, DWORD)
3 //
4 // Functional description: This function will be called implicitly before any
5 // access to range unsigned values.
6 //
7 // In case of invalid access, an error is entered in the diagnostic logbook and
8 // the application is stopped due to an access violation to avoid machine damage.
9 // In this case, check the callstack (menu Debug/Callstack) and modify the
10 // faulty logic.
11 //
12 // Handling particularities: This function increases the cycle time of all tasks
13 // accessing range signed values.
14 // The increase depends on the number of the read/write accesses.
15 //
16 // After inserting/removing this function, no online-change will be possible!
17 // A complete download will be necessary!
18 // The implementation has to be adjusted to your application/machine.
19 // In case of a division by zero, the PLC-Task will generate an access violation
20 // which leads to an application stop.
21 // Please verify whether this default behaviour can damage your machine or lead
22 // to a complex recommissioning.
23 // In this case, adapt the error reaction to your application.
24 //
25 // If your application isn't too time-critical, you should leave this function
26 // while commissioning the machine and production in your application.
27 // If you remove this function, ensure that your code is faultless.
28 //
29 // Do not change the declaration of this function! Add only local variables.
30 // Do not change the attributes above the function header!
31 {attribute 'check_range_unsigned'}
32 {attribute 'signature_flag' := '1073741824'}
```

1.1.1.1.8 POU: CheckRangeUnsigned

```
24 {attribute 'no_check'}
25 FUNCTION CheckRangeUnsigned : UDINT
26 VAR_INPUT
27     value : UDINT ;
28     lower : UDINT ;
29     upper : UDINT ;
30 END_VAR
31 VAR
32     ptNull : POINTER TO BYTE := 0 ; // Nullpointer for triggering an access
violation
33     strError : STRING ;           // Diagnostic text for the PLC logger entry
34 END_VAR
35

1
2  (*****
3    (* The implementation has to be adjusted to your application/machine. *)
4    (* This is an exemplary implementation of the function. *)
5    (* Per default, an access violation is generated to avoid machine damage *)
6    (* Check the Call Stack (menu Debug/Callstack) and modify the faulty logic!
7    *)
8
9  (*****
10
11  IF ( value < lower ) THEN
12      strError := 'CheckRangeUnsigned: Lower value ' ;
13      strError := CONCAT ( STR1 := strError , STR2 := UDINT_TO_STRING ( lower ) ) ;
14      strError := CONCAT ( STR1 := strError , STR2 := ' underrun by ' ) ;
15      strError := CONCAT ( STR1 := strError , STR2 := UDINT_TO_STRING ( value ) ) ;
16      // Default implementation for APPLICATION WILL BE STOPPED:
17      // Set an error into the plc logger
18      LogAdd2 ( hLogger := CmpLog . LOG_STD_LOGGER , udiCmpID := 16#1000 , udiClassID :=
LogClass . LOG_ERROR , udiErrorID := 1 , udiInfoID := 1 , pszInfo := strError ) ;
19      // Triggering an access violation --> Application will be stopped
20      ptNull ^ := 0 ; (* Triggering an access violation *)
21
22      // // Example for APPLICATION WILL NOT BE STOPPED:
23      // // Limit the faulty value to the lower value
24      // // Set a warning into the plc logger
25      // // NOTE: There is still an implementation error in your application code!
26      //
27      // LogAdd2(hLogger:= CmpLog.LOG_STD_LOGGER,udiCmpID:= 16#1000,udiClassID:=
LogClass.LOG_WARNING,udiErrorID:= 1,udiInfoID:= 1,pszInfo:= strError);
28      // CheckRangeUnsigned := lower; //set value to lower value
29      // return;
30
31  ELSEIF ( value > upper ) THEN
32      strError := 'CheckRangeUnsigned: Upper value ' ;
33      strError := CONCAT ( STR1 := strError , STR2 := UDINT_TO_STRING ( upper ) ) ;
34      strError := CONCAT ( STR1 := strError , STR2 := ' overrun by ' ) ;
```

```

32     strError := CONCAT ( STR1 := strError , STR2 := UDINT_TO_STRING ( value ) );
33
34     // Default implementation for APPLICATION WILL BE STOPPED:
35     // Set an error into the plc logger
36     LogAdd2 ( hLogger := CmpLog . LOG_STD_LOGGER , udiCmpID := 16#1000 , udiClassID :=
37     LogClass . LOG_ERROR , udiErrorID := 1 , udiInfoID := 1 , pszInfo := strError );
38     // Triggering an access violation --> Application will be stopped
39     ptNull ^ := 0 ; (* Triggering an access violation *)
40
41     // // Example for APPLICATION WILL NOT BE STOPPED:
42     // // Limit the faulty value to the upper value
43     // // Set a warning into the plc logger
44     // // NOTE: There is still an implementation error in your application code!
45     // LogAdd2(hLogger:= CmpLog.LOG_STD_LOGGER,udiCmpID:= 16#1000,udiClassID:=
46     LogClass.LOG_WARNING,udiErrorID:= 1,udiInfoID:= 1,pszInfo:= strError);
47     // CheckRangeUnsigned := upper; //set value to upper value
48     // return;
49     ELSE // value ist within valid range
50         CheckRangeUnsigned := value ;
51     END_IF
52

```

1.1.1.1.9 POU for Implicit Checks: CheckBounds

```

1     // Short description: Test function to increase machine safety by checking for
2     // unauthorized array indices accesses.
3     // Functional description: This function is implicitly called prior to each
4     // array access and verifies if the access is within the valid array bounds.
5     // If the access is authorized, the function returns the index to the calling
6     // POU and the access is completed.
7     // If the access is unauthorized, an exception is triggered as default error
8     // reaction. This exception results in an application stop and the system is set to
9     // safe state.
10    // Handling particularities: Sporadically, the implemented default error
11    // reaction has TO be adjusted TO the application.
12    // For example, stopping of the application can be invalid and can result in
13    // complicated recommissioning or even injury to people or damage to machines.
14    // Thus, adjust the default error reaction to your application.
15    // After inserting/removing this function in/from the project, no online change
16    // is possible.
17    // A complete download is required.
18    //
19    // Further information: The implicit calls of this check function are realized
20    // in all POU's but not in POU's of protected (*.compiled library)
21    // and open (*.library) libraries or in POU's with the attribute {attribute
22    // 'no_check'}. Consequently, they do not increase the runtime.
23    // If the check is also to be performed in open libraries, enter the compiler

```

1.1.1.1.9 POU for Implicit Checks: CheckBounds

```
definition
16 // checks_in_libs in the field Compiler-Defines of the Build tab in the property
   dialog of the application node.
17 // An increased runtime depends on the number of programmed array accesses.
18 //
19 // To reduce task runtimes, add the attribute 'no_check' to the POUs.
20 // Checks performed by the check functions (CheckPointer, CheckBounds, CheckDiv,
   CheckRange) are not performed in these POUs.
21 // Thus, these POUs have to be checked by the user regarding pointer
   programming, divisions and array accesses.
22 // For more information, refer to the F1-help.
23 //
24 // In the standard implementation, it is checked if the array access takes place
   within the valid array boundaries.
25 //
26 // In case of invalid access (outside the valid array boundaries), an error is
   entered in the PLC logger and the application is stopped to avoid injury to
   persons and damage to machines.
27 // In this case, check the callstack (menu Debug/Callstack) and change the
   incorrect logic.
28 //
29 // If your application is not time-critical, use this function during
   commissioning as well as during the production phase of the machine.
30 // Only remove this function if you are sure that your user code is free from
   errors!
31 //
32 // Do not change the declaration of this function! Add only local variables.
33 // Do not change the attributes above the function header!
34 {attribute 'check_bounds'}
35 {attribute 'no_check'}
36 {attribute 'signature_flag' := '1073741824'}
37 FUNCTION CheckBounds : DINT
38 VAR_INPUT
39     index , lower , upper : DINT ;    // Index of the array, lower bound and upper
   bound
40 END_VAR
41 VAR
42     ptNull : POINTER TO BYTE := 0 ; // Nullpointer for triggering an access
   violation
43     strError : STRING ;                // Diagnostic text for the PLC logger entry
44 END_VAR
45

1
   (*****
2       (* The implementation has to be adjusted to your application/machine. *)
3       (* This is an exemplary implementation of the function. *)
4       (* Per default, an access violation is generated in case of an faulty acces
   to avoid machine damage. *)
5       (* In this case, check the Call Stack (menu Debug/Callstack) and modify the
   faulty logic. *)
```

1.1.1.1.9 POU for Implicit Checks: CheckBounds

```
6
7
8  IF index < lower THEN
9      strError := 'CheckBounds: Lower arrayindex ';
10     strError := CONCAT ( STR1 := strError , STR2 := DINT_TO_STRING ( lower ) );
11     strError := CONCAT ( STR1 := strError , STR2 := ' underrun by ' );
12     strError := CONCAT ( STR1 := strError , STR2 := DINT_TO_STRING ( index ) );
13     // Default implementation for APPLICATION WILL BE STOPPED:
14     // Set an error into the plc logger
15     LogAdd2 ( hLogger := CmpLog . LOG_STD_LOGGER , udiCmpID := 16#1000 , udiClassID :=
16     LogClass . LOG_ERROR , udiErrorID := 1 , udiInfoID := 1 , pszInfo := strError );
17     // Triggering an access violation --> Application will be stopped
18     ptNull ^ := 0 ; (* Triggering an access violation *)
19
20     // // Example for APPLICATION WILL NOT BE STOPPED:
21     // // Limit the faulty index to the lower index
22     // // --> no memory will be overwritten
23     // // --> Keep PLC-task running
24     // // Set a warning into the plc logger
25     // // NOTE: There is still an implementation error in your application code!
26     // // LogAdd2(hLogger:= CmpLog.LOG_STD_LOGGER,udiCmpID:= 16#1000,udiClassID:=
27     LogClass.LOG_WARNING,udiErrorID:= 1,udiInfoID:= 1,pszInfo:= strError);
28     // //
29     // CheckBounds := lower; //access to the lower index only
30     // return;
31
32  ELSIF index > upper THEN
33      strError := 'CheckBounds: Higher arrayindex ';
34      strError := CONCAT ( STR1 := strError , STR2 := DINT_TO_STRING ( upper ) );
35      strError := CONCAT ( STR1 := strError , STR2 := ' overrun by ' );
36      strError := CONCAT ( STR1 := strError , STR2 := DINT_TO_STRING ( index ) );
37
38     // Default implementation for APPLICATION WILL BE STOPPED:
39     // Set an error into the plc logger
40     LogAdd2 ( hLogger := CmpLog . LOG_STD_LOGGER , udiCmpID := 16#1000 , udiClassID :=
41     LogClass . LOG_ERROR , udiErrorID := 1 , udiInfoID := 1 , pszInfo := strError );
42     // Triggering an access violation --> Application will be stopped
43     ptNull ^ := 0 ; (* Triggering an access violation *)
44
45     // // Example for APPLICATION WILL NOT BE STOPPED:
46     // // Limit the faulty index to the upper index
47     // // --> no memory will be overwritten
48     // // --> Keep PLC-task running
49     // // Set a warning into the plc logger
50     // // NOTE: There is still an implementation error in your application code!
51     // // LogAdd2(hLogger:= CmpLog.LOG_STD_LOGGER,udiCmpID:= 16#1000,udiClassID:=
52     LogClass.LOG_WARNING,udiErrorID:= 1,udiInfoID:= 1,pszInfo:= strError);
53     // //
54     // CheckBounds := upper; //access to the upper index only
55     // return;
```

```
52
53     ELSE // access to array is within lower and upper index --> ok
54         CheckBounds := index ;
55     END_IF
56
```

1.1.1.1.10 POU for Implicit Checks: CheckPointer

```
1      // Short description: Test function to increase machine safety by detecting
      // unauthorized accesses via POINTER and REFERENCE variables
2      //
3      // Functional description: This function is implicitly called prior to each
      // pointer access and verifies if the access is invalid to 0.
4      // If the access is authorized, the function returns the pointer to the calling
      // POU and the access is completed.
5      // If the access is unauthorized, an exception is triggered as default error
      // reaction. This exception results in an application stop and the system is set to
      // safe state.
6      // It is recommended to adapt an appropriate error response to the machine
      // application.
7      //
8      // Handling particularities: Sporadically, the implemented default error
      // reaction has to be adjusted to the application.
9      // For example, stopping of the application can be invalid and can result in
      // complicated recommissioning or even injury to people or damage to machines.
10     // Thus, adjust the default error reaction to your application.
11     // After inserting/removing this function in/from the project, no online change
      // is possible.
12     // A complete download is required.
13     //
14     // Further information: The implicit calls of this check function are realized
      // in all POU's but not in POU's of protected (*.compiled library)
15     // and open (*.library) libraries or in POU's with the attribute {attribute
      // 'no_check'}. Consequently, they do not increase the runtime.
16     // If the check is also to be performed in open libraries, enter the compiler
      // definition
17     // checks_in_libs in the field Compiler-Defines of the Build tab in the property
      // dialog of the application node.
18     // An increased runtime depends on the number of programmed pointer accesses or
      // the use of REFERENCE variables.
19     //
20     // To reduce task runtimes, add the attribute 'no_check' to the POU's.
21     // Checks performed by the check functions (CheckPointer, CheckBounds, CheckDiv,
      // CheckRange) are not performed in these POU's.
22     // Thus, these POU's have to be checked by the user regarding pointer
      // programming, divisions and array accesses.
23     // For more information, refer to the F1-help.
24     //
25     // In the standard implementation, it is checked if the pointer access takes
      // place to Adress 16#0 (NULL-Pointer).
26     // In this case, check the callstack (menu Debug/Callstack) and change the
```

```
incorrect logic.
27 //
28 // If your application is not time-critical, use this function during
  commissioning as well as during the production phase of the machine.
29 // Only remove this function if you are sure that your user code is free from
  errors!
30 //
31 // Do not change the declaration of this function! Add only local variables.
32 // Do not change the attributes above the function header!
33 {attribute 'check_pointer'}
34 {attribute 'signature_flag' := '1073741824'}
35 {attribute 'no_check'}
36 FUNCTION CheckPointer : POINTER TO BYTE
37 VAR_INPUT
38   ptToTest : POINTER TO BYTE; // Pointer target address
39   iSize : DINT; // Size of the type to which the pointer
  points. (z.B.: 20 für POINTER TO ARRAY [1..10] OF WORD)
40   iGran : DINT; // Granularity of the pointer access. This
  is the size of the largest unstructured data type in the type to which the
  pointer points.
41 // (e.g.: 2 for POINTER TO ARRAY [0..9] OF INT).
42   bWrite : BOOL; // Indicates read or write access. TRUE =
  write access.
43 END_VAR
44 VAR
45   ptNull : POINTER TO BYTE := 0; // Nullpointer for triggering an access
  violation
46   strError : STRING (128); // Diagnostic text for the PLC logger
  entry
47   i : INT; // Index variable
48 END_VAR
49

1 IF ptToTest = 0 THEN //Just check NULL-Pointer
2
3   IF bWrite THEN // write access
4     strError := 'CheckPointer: Invalid write-access to 16#0';
5   ELSE
6     strError := 'CheckPointer: Invalid read-access to 16#0';
7   END_IF
8   strError := CONCAT ( STR1 := strError , STR2 := ' - Size: ' );
9   strError := CONCAT ( STR1 := strError , STR2 := TO_STRING ( iSize ) );
10  strError := CONCAT ( STR1 := strError , STR2 := ' - See callstack for reason!' )
11  ;
12  LogAdd2 ( hLogger := CmpLog . LOG_STD_LOGGER , udiCmpID := 16#1000 , udiClassID :=
  LogClass . LOG_ERROR , udiErrorID := 1 , udiInfoID := 1 , pszInfo := strError );
13  // Triggering an access violation --> Application will be stopped
14  // Sporadically, this default error reaction has to be adjusted to the
  application.
15  // For example, stopping of the application can be invalid and can result in
```

```
16      complicated recommissioning or even injury to people or damage to machines.
17      // Thus, adjust the default error reaction to your application.
18      ptNull ^ := 0; //Triggering an access violation
19  ELSE // pointer is not a NULL-Pointer
20      CheckPointer := ptToTest;
21  END_IF
22
```

1.1.1.2 Folder: MotionInterface

1.1.1.2.1 Global Variable List: GlobalAxisDefines

```
1  VAR_GLOBAL CONSTANT
2      // Note: Set MOTIF_CONFIG.CONFIG_MODE_AXS to TE_AXIS_IDX_CONFIG_MODE.GLOB_VAR
      to use this list
3      // You can find the parameterlist MOTIF_CONFIG in the library
      CXA_MotionInterfaceUser
4
5      // Step 1: Define your Axis-Refs. These defines can be used for PLCopen FBs
      and also for accessing the AxisInterface
6      // Example PLCopen: "fbReadParameter( Execute:= , Axis:= vAxis1);"
7      // Example Axis-Interface: "arAxisCtrl_gb[vAxis1.AxisNo].Admin._OpMode :=
      ModeAb;"
8      // Notes: AxisName has to match the name defined in motion configuration.
      AxisNo has to be unique in range defined in MOTIF_CONFIG.
9
10
11
12      Axis_Y : MB_AXISIF_REF := ( AxisName := 'Axis_Y' , AxisNo := 1 ) ;
13
14
15
16      // Step 2: Add the above declared Axis-Refs into this list, that is forwarded
      to TE_AxisInterfaceMainProg(). The order does not matter.
17      AXIF_CONFIG_INDEXES : ARRAY [ MOTIF_CONFIG . MIN_AXIS_INDEX .. MOTIF_CONFIG .
      MAX_AXIS_INDEX ] OF MB_AXISIF_REF := [ Axis_Y ] ; //,Axis_X
18
19
20
21  END_VAR
22
```


1.1.1.2.2 POU: AxisIfMotionProg

```
1  PROGRAM AxisIfMotionProg
2  VAR
3      i : INT ;
4  END_VAR
5
```

```
1  {info 'Set the watchdog of the MotionTask to e.g. T#2ms when not using a virtual
2  control. You can remove this info after your decision'}
3  i := i + 1 ;
4  GVL.iHeartbeat := i ;
5  //#####
6  TE_AxisInterfaceMainProg.mMotionTask ( ) ; // call the AXIF FB with higher Prio
7  //#####
8
```

1.1.1.2.3 POU: AxisIfPlcProg

```
1  PROGRAM AxisIfPlcProg
2  VAR
3
4  END_VAR
5
```

```
1  TE_AxisInterfaceMainProg ( AxisCfgIdx := ADR ( AXIF_CONFIG_INDEXES ) ) ; // call
2  Axis-Interface. Do this only once in your system.
3  IF NOT TE_AxisInterfaceMainProg.InitDone OR TE_AxisInterfaceMainProg.Error
4  THEN
5      RETURN ; // do not continue, when Motion is not in RUN or initialization
6  failed.
7  END_IF
8
```

1.1.1.2.4 POU: AxisInterface

```
1  PROGRAM AxisInterface
2  VAR
3      // LREAL Feedback signal calculated from iBallFeedback integer type
4      rBallFeedback : REAL ;
5      // Raw BallFeedback value from analog input on Drive. In INT type
6      iBallFeedback : INT ;
7      // Raw motor temperature value from Drive. In INT type
8      iTempMotor : INT ;
9      // LREAL motor temperature calculated to deg C from INT type
10     rTempMotor : REAL ;
11     // Raw motor force value from Drive. In INT type
12     iForce : INT ;
```

1.1.1.2.4 POU: AxisInterface

```
13      // LREAL motor force calculated to absolute %
14      rForce : REAL ;
15      // Input from HMI to clear drive errors
16      bClearError : BOOL ;
17      // Setting of beam position command dependent on bBeamZero
18      rBeamPosCmd : LREAL ;
19      rAxis_X_position : LREAL ;
20  END_VAR
21
```

```
1  //-----
2  //----- Axis-access with AxisInterface
3  //-----
4  IF NOT TE_AxisInterfaceMainProg . InitDone OR TE_AxisInterfaceMainProg . Error
5  THEN
6      RETURN ; // do not continue, when Motion is not in RUN or initialization
7      failed.
8  END_IF
9
10 // Interlock if Emergency stop in HMI has been activated. Alarm must be
11 // acknowledged before enable can be set
12 // Additionally no drive enable can be set until axis has no errors
13 IF gvl . iAcknowledgeEMStop = 2 OR gvl . bError_Axis THEN
14     gvl . bEnable := 0 ;
15 END_IF
16
17 // Calculation of variables
18 rBallFeedback := ( -1 * iBallFeedback + 5000 ) * 77.57 ; // Invert, move from
19 // minus and scale to real size
20 rBallFeedback := ( rBallFeedback / 1000 ) ; // Move decimals
21 gvl . rBallFeedback := rBallFeedback - 65 ; // moving zero point with an offset
22 gvl . rBallFeedback3D := gvl . rBallFeedback - 330 ; // moving the ball from center
23 // to start of beam
24
25 // Motor temperature conversion from integer to real value
26 rTempMotor := iTempMotor ;
27 gvl . rTempMotorCalc := rTempMotor / 10 ;
28 // Motor force conversion from integer to real value
29 rForce := iForce ;
30 gvl . rForceMotor := ABS ( rForce / 10 ) ;
31
32 // Manual adjustment of the beam to balance the ball on a horizontal beam
33 // Set the position command signal for the beam
34 // Set beam to 0 deg
```

```
33  IF gvl.bBeamZero THEN
34      // Set the position command signal to 0
35      rBeamPosCmd := 0.0;
36      // Disable the PID controller
37      gvl.bEnablePID := FALSE;
38  ELSE
39      // Set the position command signal to the PID control output
40      rBeamPosCmd := gvl.rPIDCtrlOut;
41  END_IF
42
43
44
45  // Axis error handling
46  arAxisCtrl_gb[1].Admin.ClearError := gvl.bClearError; //Clears axis
47  errors
48  ImcCtrl.Admin.ClearError := gvl.bClearError; // Clears the red light, goes
49  to green
50  gvl.bError_Axis := arAxisStatus_gb[1].Diag.Error;
51
52  //-----
53  // Axis control
54  //-----
55
56  IF bRemoteOn_gb = FALSE THEN
57      //arAxisCtrl_gb[1].PosMode.Position := gvl.rPIDCtrlOut;
58      arAxisCtrl_gb[1].PosMode.Position := rBeamPosCmd + gvl.rJogStep; //
59      position cmd of axis
60      arAxisCtrl_gb[1].PosMode.Velocity := gvl.rAxisMaxRpm; // velocity cmd
61      of axis
62      arAxisCtrl_gb[1].PosMode.DynValues.Acceleration := gvl.rAxisAccel; //
63      axis acceleration cmd
64      arAxisCtrl_gb[1].PosMode.DynValues.Deceleration := gvl.rAxisDecel; //
65      axis deceleration cmd
66      arAxisCtrl_gb[1].Admin._OpModeBits.MODE_POS_ABS := gvl.bEnable; //
67      enabling the axis
68
69      //-----
70      // Axis-Command: Switch all axes to ModeAH with bStopAll = TRUE
71      //-----
72      IF gvl.bStopAll = TRUE THEN
73          IF arAxisStatus_gb[1].Admin.Active THEN
74              //arAxisCtrl_gb[1].Admin._OpMode := ModeAH;
75              gvl.bEnable := FALSE;
76          END_IF
77      ELSE
78          // set to Mode Ab when bStop All = FALSE
79          IF arAxisStatus_gb[1].Admin.Active AND arAxisStatus_gb[1].Admin.
80          _OpModeAckBits.MODE_AH THEN
81              arAxisCtrl_gb[1].Admin._OpMode := ModeAb;
82          END_IF
```

```

76     END_IF
77
78     END_IF
79

```

1.1.1.2.5 Visualization: OverviewAxes

Axis Details / Diagnostics		Status	Actual Values	Setup Mode
Axis: %d	AxisType: %s	Status: %s	Position: %0.2f	Vel: %0.0f Enable
Error	AxisType: %s	Axis: %d	Velocity: %0.2f	Acc: %0.0f Jog+
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	Dec: %0.0f	Jog-
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	Accel: %0.2f	StepWidth: %0.0f Incr
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	PLCopen state: %s	StepWidth: %0.0f Incr
Axis: %d	AxisType: %s	Status: %s	Position: %0.2f	Vel: %0.0f Enable
Error	AxisType: %s	Axis: %d	Velocity: %0.2f	Acc: %0.0f Jog+
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	Dec: %0.0f	Jog-
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	Accel: %0.2f	StepWidth: %0.0f Incr
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	PLCopen state: %s	StepWidth: %0.0f Incr
Axis: %d	AxisType: %s	Status: %s	Position: %0.2f	Vel: %0.0f Enable
Error	AxisType: %s	Axis: %d	Velocity: %0.2f	Acc: %0.0f Jog+
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	Dec: %0.0f	Jog-
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	Accel: %0.2f	StepWidth: %0.0f Incr
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	PLCopen state: %s	StepWidth: %0.0f Incr
Axis: %d	AxisType: %s	Status: %s	Position: %0.2f	Vel: %0.0f Enable
Error	AxisType: %s	Axis: %d	Velocity: %0.2f	Acc: %0.0f Jog+
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	Dec: %0.0f	Jog-
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	Accel: %0.2f	StepWidth: %0.0f Incr
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	PLCopen state: %s	StepWidth: %0.0f Incr
Axis: %d	AxisType: %s	Status: %s	Position: %0.2f	Vel: %0.0f Enable
Error	AxisType: %s	Axis: %d	Velocity: %0.2f	Acc: %0.0f Jog+
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	Dec: %0.0f	Jog-
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	Accel: %0.2f	StepWidth: %0.0f Incr
Diagnose Main/Detail: %s	AxisType: %s	Axis: %d	PLCopen state: %s	StepWidth: %0.0f Incr

General

Visualization size algorithm version: Respecting scrollbar location

Background

Use background color: False

Background color: 16777215

Interface

VAR_IN_OUT

END_VAR

Visual Element List

Frame Id: 0

Element name: GenElemInst_0

Type of element: Frame -> CXA_MotionInterfaceUser.OverviewAxesHeader

Tab Order: default

Static optimized: True

Clipping: False

Show frame: No frame with offset

Scaling type: Anisotropic

Deactivate the background drawing: False

References

CXA_MotionInterfaceUser.OverviewAxesHeader: 0

Position

X: 1
Y: -2
Width: 717
Height: 136
Center
X: 360
Y: 65
Colors
Color
Transparency: 255
Alarm color
Transparency: 255
Appearance
Line width: 1
Line style: Invisible
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
Font variables
Size: , <pt>
Input configuration
OnDialogClosed: Configure...
OnMouseClicked: Configure...
OnMouseDown: Configure...
OnMouseEnter: Configure...
OnMouseLeave: Configure...
OnMouseMove: Configure...
OnMouseUp: Configure...
Toggle
Toggle on up if captured: False
Tap
Tap FALSE: False
Tap on enter if captured: False
Frame Id: 1
Element name: GenElemInst_21
Type of element: Frame -> CXA_MotionInterfaceUser.OverviewOneAxis
Tab Order: default
Static optimized: True
Clipping: False
Show frame: No frame
Scaling type: Anisotropic
Deactivate the background drawing: False
References
CXA_MotionInterfaceUser.OverviewOneAxis
m_Input_AxisIndex: (MOTIF_CONFIG.MIN_AXIS_INDEX+0)
Position

X: 1
Y: 133
Width: 717
Height: 92
Center
X: 359
Y: 178
Colors
Color
Transparency: 255
Alarm color
Transparency: 255
Appearance
Line width: 1
Line style: Solid
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
Font variables
Size: , <pt>
Input configuration
OnDialogClosed: Configure...
OnMouseClicked: Configure...
OnMouseDown: Configure...
OnMouseEnter: Configure...
OnMouseLeave: Configure...
OnMouseMove: Configure...
OnMouseUp: Configure...
Toggle
Toggle on up if captured: False
Tap
Tap FALSE: False
Tap on enter if captured: False
Frame Id: 2
Element name: GenElemInst_23
Type of element: Frame -> CXA_MotionInterfaceUser.OverviewOneAxis
Tab Order: default
Static optimized: True
Clipping: False
Show frame: No frame
Scaling type: Anisotropic
Deactivate the background drawing: False
References
CXA_MotionInterfaceUser.OverviewOneAxis
m_Input_AxisIndex: (MOTIF_CONFIG.MIN_AXIS_INDEX+1)
Position

X: 1
Y: 223
Width: 717
Height: 92
Center
X: 359
Y: 269
Colors
Color
Transparency: 255
Alarm color
Transparency: 255
Appearance
Line width: 1
Line style: Solid
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
Font variables
Size: , <pt>
Input configuration
OnDialogClosed: Configure...
OnMouseClicked: Configure...
OnMouseDown: Configure...
OnMouseEnter: Configure...
OnMouseLeave: Configure...
OnMouseMove: Configure...
OnMouseUp: Configure...
Toggle
Toggle on up if captured: False
Tap
Tap FALSE: False
Tap on enter if captured: False
Frame Id: 5
Element name: GenElemInst_61
Type of element: Frame -> CXA_MotionInterfaceUser.OverviewOneAxis
Tab Order: default
Static optimized: True
Clipping: False
Show frame: No frame
Scaling type: Anisotropic
Deactivate the background drawing: False
References
CXA_MotionInterfaceUser.OverviewOneAxis
m_Input_AxisIndex: (MOTIF_CONFIG.MIN_AXIS_INDEX+2)
Position

X: 1
Y: 313
Width: 717
Height: 92
Center
X: 359
Y: 359
Colors
Color
Transparency: 255
Alarm color
Transparency: 255
Appearance
Line width: 1
Line style: Solid
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
Font variables
Size: , <pt>
Input configuration
OnDialogClosed: Configure...
OnMouseClicked: Configure...
OnMouseDown: Configure...
OnMouseEnter: Configure...
OnMouseLeave: Configure...
OnMouseMove: Configure...
OnMouseUp: Configure...
Toggle
Toggle on up if captured: False
Tap
Tap FALSE: False
Tap on enter if captured: False
Frame Id: 7
Element name: GenElemInst_63
Type of element: Frame -> CXA_MotionInterfaceUser.OverviewOneAxis
Tab Order: default
Static optimized: True
Clipping: False
Show frame: No frame
Scaling type: Anisotropic
Deactivate the background drawing: False
References
CXA_MotionInterfaceUser.OverviewOneAxis
m_Input_AxisIndex: (MOTIF_CONFIG.MIN_AXIS_INDEX+3)
Position

X: 1
Y: 402
Width: 717
Height: 92
Center
X: 359
Y: 448
Colors
Color
Transparency: 255
Alarm color
Transparency: 255
Appearance
Line width: 1
Line style: Solid
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
Font variables
Size: , <pt>
Input configuration
OnDialogClosed: Configure...
OnMouseClicked: Configure...
OnMouseDown: Configure...
OnMouseEnter: Configure...
OnMouseLeave: Configure...
OnMouseMove: Configure...
OnMouseUp: Configure...
Toggle
Toggle on up if captured: False
Tap
Tap FALSE: False
Tap on enter if captured: False
Frame Id: 9
Element name: GenElemInst_67
Type of element: Frame -> CXA_MotionInterfaceUser.OverviewOneAxis
Tab Order: default
Static optimized: True
Clipping: False
Show frame: No frame
Scaling type: Anisotropic
Deactivate the background drawing: False
References
CXA_MotionInterfaceUser.OverviewOneAxis
m_Input_AxisIndex: (MOTIF_CONFIG.MIN_AXIS_INDEX+4)
Position

X: 1
Y: 493
Width: 717
Height: 92
Center
X: 359
Y: 548
Colors
Color
Transparency: 255
Alarm color
Transparency: 255
Appearance
Line width: 1
Line style: Solid
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
Font variables
Size: , <pt>
Color variables
Toggle color: <toggle/tap variable>
Input configuration
OnDialogClosed: Configure...
OnMouseClicked: Configure...
OnMouseDown: Configure...
OnMouseEnter: Configure...
OnMouseLeave: Configure...
OnMouseMove: Configure...
OnMouseUp: Configure...
Toggle
Toggle on up if captured: False
Tap
Tap FALSE: False
Tap on enter if captured: False
Frame Id: 12
Element name: GenElemInst_69
Type of element: Frame -> CXA_MotionInterfaceUser.OverviewOneAxis
Tab Order: default
Static optimized: True
Clipping: False
Show frame: No frame
Scaling type: Anisotropic
Deactivate the background drawing: False
References
CXA_MotionInterfaceUser.OverviewOneAxis

m_Input_AxisIndex: (MOTIF_CONFIG.MIN_AXIS_INDEX+5)

Position
X: 1
Y: 584
Width: 717
Height: 92

Center
X: 359
Y: 639

Colors
Color
Transparency: 255
Alarm color
Transparency: 255

Appearance
Line width: 1
Line style: Solid

Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default

Font
Transparency: 255

Font variables
Size: , <pt>

Color variables
Toggle color: <toggle/tap variable>

Input configuration
OnDialogClosed: Configure...
OnMouseClicked: Configure...
OnMouseDown: Configure...
OnMouseEnter: Configure...
OnMouseLeave: Configure...
OnMouseMove: Configure...
OnMouseUp: Configure...

Toggle
Toggle on up if captured: False

Tap
Tap FALSE: False
Tap on enter if captured: False

Frame Id: 15
Element name: GenElemInst_71
Type of element: Frame -> CXA_MotionInterfaceUser.OverviewOneAxis
Tab Order: default
Static optimized: True
Clipping: False
Show frame: No frame
Scaling type: Anisotropic
Deactivate the background drawing: False

References

CXA_MotionInterfaceUser.OverviewOneAxis
m_Input_AxisIndex: (MOTIF_CONFIG.MIN_AXIS_INDEX+6)

Position

X: 1
Y: 674
Width: 717
Height: 92

Center

X: 359
Y: 729

Colors

Color
Transparency: 255
Alarm color
Transparency: 255

Appearance

Line width: 1
Line style: Solid

Text properties

Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255

Font variables

Size: , <pt>

Color variables

Toggle color: <toggle/tap variable>

Input configuration

OnDialogClosed: Configure...
OnMouseClicked: Configure...
OnMouseDown: Configure...
OnMouseEnter: Configure...
OnMouseLeave: Configure...
OnMouseMove: Configure...
OnMouseUp: Configure...

Toggle

Toggle on up if captured: False

Tap

Tap FALSE: False
Tap on enter if captured: False

1.1.1.3 Global Variable List: GVL

```
1      {attribute 'qualified_only'}
2      {attribute 'linkalways'}
3      VAR_GLOBAL
4          // Enable the drive
5          bEnable : BOOL ;
6          // Enable the PID controllers
7          bEnablePID : BOOL ;
8          // Clears drive error
9          bClearError : BOOL ;
10         // Preset value for the PID
11         rPresetVal : LREAL := 0 ;
12         // Position command setpoint for ball position on beam
13         rBallPosCmd : LREAL ;
14         // kp gain for the Position loop PID
15         rPGainPID : LREAL := 1.36 ;
16         // Maximum allowable beam angle
17         rCtrlMax : LREAL := 15 ;
18         // Minimum allowable beam angle
19         rCtrlMin : LREAL := -15 ;
20         // Integral gain PID Position
21         lICtrlPID : LTIME ;
22         // Derivative gain PID Position
23         lDCtrlPID : LTIME ;
24         // Derivative gain PID Position from HMI
25         lDGain : LTIME := ltime#400ns ;
26         // Integral gain PID Position from HMI
27         lIGain : LTIME ;
28         // Status of the PID
29         bPIDActive : BOOL ;
30         // Error status of PID
31         bPIDError : BOOL ;
32         // Pause the PID
33         bPIDPause : BOOL ;
34         // Output from the velocity PID as a position input to axis[1]
35         rPIDCtrlOut : LREAL ;
36         // Actual ball feedback position with set offset
37         rBallFeedback : REAL ;
38         // Actual motor temperature
39         rTempMotorCalc : REAL ;
40         // Force from motor in %
41         rForceMotor : REAL ;
42         // PID adjustment constant
43         rbCtrlPID : LREAL := 1 ;
44         // PID adjustment constant
45         rcCtrlPID : LREAL := 1 ;
46         // Calculated following error between BallPosCmd and BallFeedback
47         rFollowingError : LREAL ;
48         // For use in local HMI
```

1.1.1.3 Global Variable List: GVL

```
49      rLowWindowSet : LREAL ;
50      // For use in local HMI
51      rHighWindowSet : LREAL ;
52      // Output from the position controller PID to the velocity PID controller
Setpoint
53      rPIDOUT : LREAL ;
54      // Actual beam position feedback
55      rActualBeamPos : LREAL ;
56      // Sets the frequency of the sinus signal
57      rSinusFreq : Lreal ;
58      // Amplitude setpoint from HMI for sinus function
59      rAmplitude : Lreal ;
60      // Position input for direction of step, switches between 2 and 3
61      iSignalType : int ;
62      // Position command from slider in HMI as setpoint in Position loop PID
63      rSliderVal : lreal ;
64      // Left ball position when in step mode. Set from HMI
65      rSetLeftPosition : lreal ;
66      // Right ball position when in step mode. Set from HMI
67      rSetRightPosition : lreal ;
68      // Constant for setting home position at axis center
69      rBallHomePos : lreal := 330 ;
70      // Output from the sinus signal generator to the position PID setpoint
71      rSinusOutputPID : lreal ;
72      // Sets the shape of the sinus with offset of the slope in %
73      rSinusDuty : lreal := 25 ;
74      // Activate the Slider function
75      bSliderOn : bool ;
76      // Activate the sinus input function
77      bSinusOn : bool ;
78      // Command the ball to go to left set position
79      bLeftPos : bool ;
80      // Command the ball to go to right set position
81      bRightPos : bool ;
82      // Commands beam to zero position
83      bCenterPos : bool ;
84      // Activate the Step function
85      bStepOn : bool ;
86      // Error notification if drive has an error
87      bError_Axis : bool ;
88      // Emergency stop command
89      bStopAll : bool ;
90      // Additive beam position command during calibration of beam position
91      rJogStep : lreal ;
92      // Commands beam to zero position
93      bBeamZero : bool ;
94      // Set absolute position servo encoder
95      bSetRefPos : bool ;
96      // Velocity feedback signal based on derivation of the filtered
rBallFeedback signal
97      rVelocity : lreal ;
```

1.1.1.3 Global Variable List: GVL

```
98      // State of acknowledged alarms
99      iAcknowledgeEMStop : uint;
100     // kp gain for the Velocity loop PID
101     rPGainPIDVel : lreal := 15;
102     // Integral gain PID Velocity
103     lICtrlPIDVel : ltime;
104     // Derivative gain PID Velocity
105     lDCtrlPIDVel : ltime;
106     // Derivative gain PID Velocity from HMI
107     lDGainVel : ltime;
108     // Integral gain PID Velocity from HMI
109     lIGainVel : ltime;
110     // Low pass filter coefficient found by calculation in Matlab
111     rFwdCoeff1 : lreal := 0;
112     // Low pass filter coefficient found by calculation in Matlab
113     rFwdCoeff2 : lreal := 0.0174;
114     // Low pass filter coefficient found by calculation in Matlab
115     rBwdCoeff1 : lreal := 0;
116     // Low pass filter coefficient found by calculation in Matlab
117     rBwdCoeff2 : lreal := -0.9826;
118     // Axis[1] acceleration setting in rad/s^2
119     rAxisAccel : lreal := 2;
120     // Axis[1] deceleration setting in rad/s^2
121     rAxisDecel : lreal := 2;
122     // Axis[1] max rpm
123     rAxisMaxRpm : lreal := 500;
124     // Counter for the heartbeat monitor in the HMI to verify connection
125     iHeartbeat : LREAL;
126     // 3d viewer ball positon variable
127     rBallFeedback3D : LREAL;
128
129     END_VAR
130
```

1.1.1.4 POU: PID_ST

```
1  PROGRAM PID_ST
2  VAR
3      // Integral gain for position loop PID
4      lILocal : LTIME;
5
6      // Definition of PID contoller type 03 Velocity loop
7      PID_Velocity : IL_PIDType03;
8      // Definition of PID contoller type 03 Position loop
9      PID_Position : IL_PIDType03;
10
11     // Definition of IIR filtertype 01
12     LP_Filter_Position_Sensor : IL_IIRType01;
13     fbIIR1 : CXA_LOOPCONTROL . IL_IIRType01; // Declaration of filter type
14     tiIR_Coff : CXA_LOOPCONTROL . IL_IIR_COEFF; // Configuration of filter
        coefficient
```

1.1.1.4 POU: PID_ST

```
15
16
17     // Output from position loop PID to velocity setpoint
18     rPIDOut : LREAL ;
19
20     rdt : LREAL := 0.002 ; // time step for derivation
21     rVelocity : LREAL ; // Velocity output from derivation of ball position
22     feedback signal
23     position : LREAL ; // stored position, used for derivation
24     prev_pos : LREAL := 0.0 ; // previous position, used for derivation
25     // Output from filtering of the ball position feedback
26     rLP_PosFilter_out : LREAL ;
27     // Angle generator for the sinus function
28     AngleGenerator : IL_AngleGeneratorType01 ;
29     // Sinus signal generator
30     SinusSignal : IL_SignalGeneratorType01 ;
31     // Output from angle generator to sinus generator angle input
32     rSinusAngle : LREAL ;
33     // Output from sinus generator to amplitude calculation
34     rSinusOut : LREAL ;
35     // kp gain for Position loop PID
36     rPLocal : LREAL ;
37     // Derivative gain for position loop PID
38     lDLocal : LTIME ;
39     // kp gain for Velocity loop PID
40     rPLocalVel : LREAL ;
41     // Integral gain for velocity loop PID
42     lILocalVel : LTIME ;
43     // Derivative gain for velocity loop PID
44     lDLocalVel : LTIME ;
45     END_VAR
```

```
1     // Ball position control program
2
3
4     // Assigning global variables
5     GVL.rActualBeamPos := arAxisStatus_gb [ 1 ] . Data . ActualPosition ;
6     gvl.rFollowingError := ABS ( gvl.rBallPosCmd - gvl.rBallFeedback ) ;
7
8     // Setting local position loop pid variables to global variables from HMI
9     // rPLocal := gvl.rPGainPID; // reset DControl to original value
10    rPLocal := gvl.rPGainPID / 1000 ; // reset pControl to original value
11    lILocal := gvl.lICtrlPID ; // reset iControl to original value
12    lDLocal := gvl.lDCtrlPID ; // reset DControl to original value
13    // Setting local velocity loop pid variables to global variables from HMI
14    rPLocalVel := gvl.rPGainPIDVel ; // reset pControl to original value
15    lILocalVel := gvl.lICtrlPIDVel ; // reset iControl to original value
16    lDLocalVel := gvl.lDCtrlPIDVel ; // reset DControl to original value
17
```



```

18 // IIR filter coefficients from c2d conversion in matlab. Forward coefficients
19 // are the numerator values, b0,b1,b2..., and
20 // the Backward coefficients are the denominator values, 1,a1,a2....
21 // Single LP filter at frequency= 8.8 rad/s
22 tIIR_Coff.BackwardCoeffs [ 1 ] := gvl.rBwdCoeff1 ;
23 tIIR_Coff.BackwardCoeffs [ 2 ] := gvl.rBwdCoeff2 ;
24 tIIR_Coff.ForwardCoeffs [ 0 ] := gvl.rFwdCoeff1 ;
25 tIIR_Coff.ForwardCoeffs [ 1 ] := gvl.rFwdCoeff2 ;
26 tIIR_Coff.NumberOfCoeffs := 4 ;
27
28 //-----
29 // Generator for sinus signal for the sinus function in the HMI
30 AngleGenerator (
31     Enable := gvl.bEnablePID ,
32     InOperation => ,
33     Error => ,
34     ErrorID => ,
35     ErrorIdent => ,
36     Pause := ,
37     Frequency := gvl.rSinusFreq ,
38     InPause => ,
39     OutputAngle => rSinusAngle ,
40     ActScanTime => ) ;
41
42 SinusSignal (
43     Enable := gvl.bEnablePID ,
44     InOperation => ,
45     Error => ,
46     ErrorID => ,
47     ErrorIdent => ,
48     Angle := rSinusAngle ,
49     CurveType := 1 ,
50     Duty := gvl.rSinusDuty ,
51     DutyRamp := 25 ,
52     Rounding := ,
53     RoundingRamp := ,
54     OutputValue => rSinusOut ,
55     ActScanTime => ) ;
56
57 // Sinus output to ball position control PID;
58 gvl.rSinusOutputPID := ( rSinusOut ) * gvl.rAmplitude + 330 ;
59
60 //-----
61 // Low pass filter for filtering ball position feedback signal
62 LP_Filter_Position_Sensor (
63     Enable := gvl.bEnablePID ,
64     InOperation => ,
65     Error => ,
66     ErrorID => ,
67     ErrorIdent => ,

```

```

67     Pause := ,
68     Value := gvl.rBallFeedback ,
69     StartValue := ,
70     Coefficients := tIIR_Coff ,
71     InPause => ,
72     ActScanTime => ,
73     OutputValue => rLP_PosFilter_out ) ;
74 //-----
75 // PID function block for position closed loop
76 PID_Position (
77     Enable := gvl.bEnablePID ,
78     InOperation => gvl.bPIDActive ,
79     Error => gvl.bPIDError ,
80     ErrorID => ,
81     ErrorIdent => ,
82     Pause := ,
83     Preset := ,
84     Setpoint := gvl.rBallPosCmd ,
85     Feedback := rLP_PosFilter_out ,
86     PresetValue := GVL.rPresetVal ,
87     HighLimit := GVL.rCtrlMax ,
88     LowLimit := GVL.rCtrlMin ,
89     PControl := rPLocal ,
90     IControl := lILocal ,
91     DControl := lDLocal ,
92     bControl := gvl.rbCtrlPID ,
93     cControl := gvl.rcCtrlPID ,
94     InPause => ,
95     PresetAck => ,
96     ActScanTime => ,
97     HighLimitActive => ,
98     LowLimitActive => ,
99     ControlValue => gvl.rPIDOut ) ;
100 //-----
101 // Finding the derivative from the position signal to get velocity
102 // Calculate velocity using derivation, rdt is the cycle time 0.002s
103
104 gvl.rVelocity := ( ( rLP_PosFilter_out / 1000 ) - prev_pos ) / rdt ;
105
106 // update previous position
107 prev_pos := ( rLP_PosFilter_out / 1000 ) ;
108 //-----
109 // PID function block for velocity closed loop
110 PID_Velocity (
111     Enable := gvl.bEnablePID ,
112     InOperation => gvl.bPIDActive ,
113     Error => ,
114     ErrorID => ,
115     ErrorIdent => ,
116     Pause := ,
117     Preset := ,

```

1.1.1.4 POU: PID_ST

```
118      Setpoint := gvl.rPIDOUT , // rPIDOut_Filter
119      Feedback := gvl.rVelocity ,
120      PresetValue := GVL.rPresetVal ,
121      HighLimit := GVL.rCtrlMax ,
122      LowLimit := GVL.rCtrlMin ,
123      PControl := rPLocalVel ,
124      IControl := lILocalVel ,
125      DControl := lDLocalVel ,
126      bControl := gvl.rbCtrlPID ,
127      cControl := gvl.rcCtrlPID ,
128      InPause => ,
129      PresetAck => ,
130      ActScanTime => ,
131      HighLimitActive => ,
132      LowLimitActive => ,
133      ControlValue => gvl.rPIDCtrlOut ) ;
134
135
136
137
138
139
140
141
```

1.1.1.5 POU: PLC_PRG

```
1      PROGRAM PLC_PRG
2      VAR
3          // Resets position command to 0
4          rPosCmd : LREAL := 0 ;
5          // Selection of different ball position modes from HMI
6          iModeSelection : INT ;
7          // Define MB_Axis_Administration
8          Axis_Admin : MB_AXIS_ADMINISTRATION ;
9      END_VAR
10
11
12      //
13
14      //Setting of variables and calculations that can have a slow cycle time
15
16      //Calculations for the right input from the HMI
17      // Calc ms from nanosec for the D_part gain in the HMI
18      gvl.lDCtrlPID := gvl.lDGain * 1000000 ;
19      // Calc ms from nanosec for the I_part gain in the HMI
20      gvl.lICtrlPID := GVL.lIGain * 1000000 ;
21
22      // Calc ms from nanosec for the D_part gain in the HMI
23      gvl.lDCtrlPIDVel := gvl.lDGainVel * 1000000 ;
```

```
13      // Calc ms from nanosec for the I_part gain in the HMI
14      gvl.lICtrlPIDVel := GVL.lIGainVel * 1000000 ;
15
16      // Setting Remote to off on startup.
17      bRemoteOn_gb := FALSE ;
18
19      //-----
20      // Logic for switching between slider, sinus and square position command
21      // Making variables for buttons in HMI
22
23      // Reset all boolean values TO FALSE when gvl.bEnable is FALSE
24      IF NOT gvl.bEnable THEN
25          gvl.bSliderOn := FALSE ;
26          gvl.bSinusOn := FALSE ;
27          gvl.bLeftPos := FALSE ;
28          gvl.bRightPos := FALSE ;
29          gvl.bCenterPos := FALSE ;
30          gvl.bStepOn := FALSE ;
31          iModeSelection := 4 ; // Set to center position mode
32      ELSE
33          // Mode selection based on the state of the boolean inputs
34          IF gvl.bSliderOn THEN
35              iModeSelection := 0 ; // Slider mode
36              gvl.bSinusOn := FALSE ; // Reset the other boolean inputs to false
37              gvl.bLeftPos := FALSE ;
38              gvl.bRightPos := FALSE ;
39              gvl.bCenterPos := FALSE ;
40              gvl.bStepOn := FALSE ;
41          ELSIF gvl.bSinusOn THEN
42              iModeSelection := 1 ; // Sinusoidal mode
43              gvl.bSliderOn := FALSE ; // Reset the other boolean inputs to false
44              gvl.bLeftPos := FALSE ;
45              gvl.bRightPos := FALSE ;
46              gvl.bCenterPos := FALSE ;
47              gvl.bStepOn := FALSE ;
48          ELSIF gvl.bStepOn THEN
49              iModeSelection := gvl.iSignalType ;
50              gvl.bSliderOn := FALSE ; // Reset the other boolean inputs to false
51              gvl.bSinusOn := FALSE ;
52              gvl.bCenterPos := FALSE ;
53          ELSIF gvl.bCenterPos THEN
54              iModeSelection := 4 ; // Center position mode
55              gvl.bSliderOn := FALSE ; // Reset the other boolean inputs to false
56              gvl.bSinusOn := FALSE ;
57              gvl.bLeftPos := FALSE ;
58              gvl.bRightPos := FALSE ;
59              gvl.bStepOn := False ;
60          ELSE
61              iModeSelection := 4 ; // If all boolean inputs are false, set to center
62              position mode
```

```
62         END_IF
63     END_IF
64
65     //-----
66     // Mode selection in HMI, choose position command
67     CASE iModeSelection OF
68         0 : // Direct value from slider
69             rPosCmd := gvl.rSliderVal ;
70
71         1 : // Sinusoidal input start
72
73             // Sinus output to ball position control PID;
74             rPosCmd := gvl.rSinusOutputPID ;
75
76         2 : // First set value
77             rPosCmd := gvl.rSetLeftPosition ;
78
79         3 : // Second set value
80             rPosCmd := gvl.rSetRightPosition ;
81
82         4 : // Home position of beam
83             rPosCmd := gvl.rBallHomePos ;
84     END_CASE
85
86     // Assign the final position command value to the gvl.rBallPosCmd variable
87     gvl.rBallPosCmd := rPosCmd ;
88
89     //-----
90     // Beam calibration. Set absolute position of servo motor encoder
91     IF gvl.bSetRefPos THEN
92         gvl.bEnable := FALSE ;
93         gvl.bBeamZero := FALSE ;
94         gvl.rJogStep := 0 ;
95         arAxisCtrl_gb [ 1 ] . Admin . SetAbsRef := gvl.bSetRefPos ;
96     ELSE
97         arAxisCtrl_gb [ 1 ] . Admin . SetAbsRef := FALSE ;
98     END_IF
99
100
101
102
103
104
105
106
107
108
109
110
```

1.1.1.6 Symbol Configuration: Symbols

1.1.1.7 Task Configuration: Task Configuration

Max. number of tasks: 20
Max. number of cyclic tasks: 20
Max. number of freewheeling tasks: 20
Max. number of event tasks: 20
Max. number of external event tasks: 20
Max. number of status tasks: 20

System Events:

1.1.1.7.1 Task: MainTask

Priority: 30
Type: Cyclic
Interval: 200 Unit: ms
Watchdog: Active
Watchdog Time: 20 Unit: ms
Watchdog Sensitivity: 1
POUs: PLC_PRG

1.1.1.7.2 Task: MainTaskMotlf

Priority: 30
Type: Cyclic
Interval: 2 Unit: ms
Watchdog: Inactive
POUs: AxisInterface
AxisIfMotionProg
AxisIfPlcProg

1.1.1.7.3 Task: MotionTask

Priority: 29
Type: Cyclic
Interval: 2 Unit: ms
Watchdog: Inactive
POUs: PID_ST

1.1.1.7.4 Task: TrendRecordingTask

Priority: 39

Type: Cyclic

Interval: 100 Unit: ms

Watchdog: Inactive

POUs: VisuTrendStorageAccess.GlobalInstances.g_TrendRecordingManager.CyclicCall

1.1.1.8 Trace: Trace01

Settings:

Record 'Trace01':

Trigger variable:

Trigger edge: None

Post trigger (samples): 0

Trigger value:

Task: MotionTask

Measure in every: 1-th cycle

Record condition:

Buffer size: 2000

Comment:

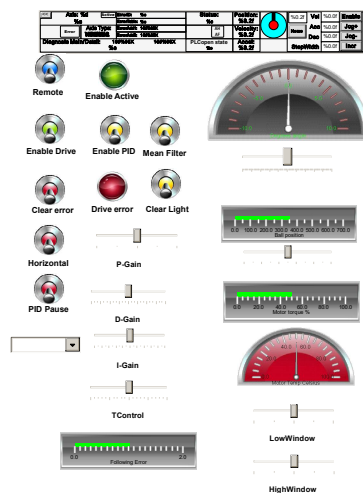
POU for visualisation: False

Variables: PID_ST.PID_Position.Setpoint

PID_ST.PID_Position.Feedback

GVL.rPGainPID

1.1.1.9 Visualization: HMI



General

Visualization size algorithm version: Respecting scrollbar location

Background

Use background color: False

Background color: 16777215

Interface

VAR_IN_OUT

END_VAR

Visual Element List

Meter 180° Id: 6

Element name: GenElemInst_6

Type of element: Meter 180°

Tab Order: default

Static optimized: False

Value: arAxisStatus_gb[1].Data.ActualPosition

Position

X: 768

Y: 119

Width: 361

Height: 323

Background

Image color: Element-Meter-Meter180-Gray

Own image

Image: <default>

Transparency color: Color [A=255, R=0, G=0, B=0]

Optimized drawing: True

Arrow

Arrow type: Thin 3D arrow

Color: Color [A=255, R=255, G=255, B=255]

Angle range: Top

Additional arrow: False

Scale

Subscale position: Outside

Scale type: Lines

Scale start: -10

Scale end: 10

Main scale: 5

Subscale: 1

Scale line width: 1

Scale color: Color [A=255, R=188, G=143, B=143]

Scale in 3D: True

Show scale: True

Frame inside: False

Frame outside: False

Label

Label: Inside

Unit: Degrees angle

Font: Not implemented yet

Scale format (C syntax): %.1f
Maximum text width of labels: 35
Text height of labels: 15
Positioning
Usage of: Default style values
Positioning
Needle movement: 100
Scale movement: 51
Scale length: 50
Label offset: 50
Unit offset: 50
Origin offset: 0
Colors
Color areas
Durable color areas: True
Use color areas for scale: False
TLabel Id: 7
Element name: GenElemInst_7
Type of element: Label
Tab Order: default
Static optimized: True
Text ID: 206
Texts
Text: Remote
Position
X: 341
Y: 179
Width: 150
Height: 30
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
TLabel Id: 9
Element name: GenElemInst_9
Type of element: Label
Tab Order: default
Static optimized: True
Text ID: 223
Texts
Text: Enable Drive
Position
X: 349
Y: 309
Width: 150
Height: 30
Text properties

Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
Slider Id: 14
Element name: Position command
Type of element: Slider
Tab Order: default
Static optimized: False
Position
X: 837
Y: 316
Width: 219
Height: 71
Variable: AxisInterface.rAngleDirect
Move to click: True
Scale
Show scale: True
Scale start: 0
Scale end: 500
Main scale: 50
Subscale: 5
Scale format (C syntax): %i
Scale proportion: 0
Bar
Diagram type: Bottom
Diagram type: Right
Orientation: Horizontal
Running direction: Left to right
Running direction: Top to bottom
Dip Switch Id: 2
Element name: GenElemInst_15
Type of element: Dip Switch
Tab Order: default
Static optimized: False
Position
X: 386
Y: 373
Width: 70
Height: 70
Variable: GVL.bClearError
Image settings
Transparent: False
Transparent color: Color [A=0, R=255, G=255, B=255]
Isotropic type: Isotropic
Horizontal alignment: Left
Vertical alignment: Top
Element behavior: Image toggler

Tap FALSE: False
Background
Image: Element-Switch-DipSwitch-Red
τ Label Id: 17
Element name: GenElemInst_17
Type of element: Label
Tab Order: default
Static optimized: True
Text ID: 423
Texts
Text: Clear error
Position
X: 352
Y: 441
Width: 150
Height: 30
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
● Lamp Id: 3
Element name: GenElemInst_18
Type of element: Lamp
Tab Order: default
Static optimized: False
Position
X: 525
Y: 366
Width: 70
Height: 70
Variable: AxisInterface.bError_Axis
Image settings
Transparent: False
Transparent color: Color [A=0, R=255, G=255, B=255]
Isotropic type: Isotropic
Horizontal alignment: Left
Vertical alignment: Top
Background
Image: Element-Lamp-Lamp1-Red
⊛ Dip Switch Id: 19
Element name: GenElemInst_19
Type of element: Dip Switch
Tab Order: default
Static optimized: False
Position
X: 386
Y: 240

Width: 70
Height: 70
Variable: GVL.bEnable
Image settings
 Transparent: False
 Transparent color: Color [A=0, R=255, G=255, B=255]
 Isotropic type: Isotropic
 Horizontal alignment: Left
 Vertical alignment: Top
Element behavior: Image toggler
Tap FALSE: False
Background
 Image: Element-Switch-DipSwitch-Green
Frame Id: 25
Element name: GenElemInst_25
Type of element: Frame -> CXA_MotionInterfaceUser.OverviewOneAxis
Tab Order: default
Static optimized: True
Clipping: False
Show frame: Frame
Scaling type: Anisotropic
Deactivate the background drawing: False
References
 CXA_MotionInterfaceUser.OverviewOneAxis
 m_Input_AxisIndex: (MOTIF_CONFIG.MIN_AXIS_INDEX+0)
Position
 X: 397
 Y: 12
 Width: 719
 Height: 94
Center
 X: 756
 Y: 59
Colors
 Color
 Transparency: 255
 Alarm color
 Transparency: 255
Appearance
 Line width: 1
 Line style: Solid
Text properties
 Horizontal alignment: Centered
 Vertical alignment: Centered
 Text format: Default
 Font
 Transparency: 255
Font variables
 Size: , <pt>

Color variables

Toggle color: <toggle/tap variable>

Input configuration

OnDialogClosed: Configure...

OnMouseClicked: Configure...

OnMouseDown: Configure...

OnMouseEnter: Configure...

OnMouseLeave: Configure...

OnMouseMove: Configure...

OnMouseUp: Configure...

Toggle

Toggle on up if captured: False

Tap

Tap FALSE: False

Tap on enter if captured: False

* Dip Switch Id: 26

Element name: GenElemInst_26

Type of element: Dip Switch

Tab Order: default

Static optimized: False

Position

X: 386

Y: 107

Width: 70

Height: 70

Variable: bRemoteOn_gb

Image settings

Transparent: False

Transparent color: Color [A=0, R=255, G=255, B=255]

Isotropic type: Isotropic

Horizontal alignment: Left

Vertical alignment: Top

Element behavior: Image toggler

Tap FALSE: False

Background

Image: Element-Switch-DipSwitch-Blue

Bar Display Id: 27

Element name: GenElemInst_27

Type of element: Bar Display

Tab Order: default

Static optimized: False

Value: GVL.rBallFeedback

Center

X: 951

Y: 470

Position

X: 801

Y: 445

Width: 300

Height: 80
Background
Image color: Element-BarDisplayImage-Gray
Own image
Image: <default>
Transparency color: Color [A=255, R=0, G=0, B=0]
Optimized drawing: True
Bar
Diagram type: Scale besides bar
Orientation: Horizontal
Running direction: Left to right
Running direction: Bottom to top
Optimum size for bar: False
Scale
Scale start: 0
Scale end: 700
Main scale: 100
Subscale: 50
Scale line width: 1
Scale color: Color [A=255, R=255, G=255, B=255]
Scale in 3D: True
Element frame: False
Label
Unit: Ball position
Font: Not implemented yet
Scale format (C syntax): %.1f
Maximum text width of labels: 38
Text height of labels: 15
Positioning
Horizontal offset: 0
Vertical offset: 0
Horizontal scaling: 0
Vertical scaling: 0
Colors
Graph color: Color [A=255, R=0, G=255, B=0]
Bar background: False
Frame color: Color [A=255, R=212, G=208, B=200]
Switch whole color: False
Use gradient color for bar: False
Color range markers: No markers
🔊 Meter 180° Id: 29
Element name: GenElemInst_29
Type of element: Meter 180°
Tab Order: default
Static optimized: False
Value: GVL.rTempMotorCalc
Position
X: 835
Y: 713

Width: 261
Height: 250
Background
Image color: Element-Meter-Meter180-Red
Own image
Image: <default>
Transparency color: Color [A=255, R=0, G=0, B=0]
Optimized drawing: True
Arrow
Arrow type: Thin 3D arrow
Color: Color [A=255, R=255, G=255, B=255]
Angle range: Top
Additional arrow: False
Scale
Subscale position: Outside
Scale type: Lines
Scale start: 0
Scale end: 100
Main scale: 20
Subscale: 5
Scale line width: 1
Scale color: Color [A=255, R=255, G=255, B=255]
Scale in 3D: True
Show scale: True
Frame inside: False
Frame outside: False
Label
Label: Inside
Unit: Motor Temp Celsius
Font: Not implemented yet
Scale format (C syntax): %.1f
Maximum text width of labels: 34
Text height of labels: 14
Positioning
Usage of: Default style values
Positioning
Needle movement: 100
Scale movement: 51
Scale length: 50
Label offset: 50
Unit offset: 50
Origin offset: 0
Colors
Color areas
Durable color areas: True
Use color areas for scale: False
Bar Display Id: 0
Element name: GenElemInst_31
Type of element: Bar Display

Tab Order: default
Static optimized: False
Value: GVL.rForceMotor
Center
X: 956
Y: 638
Position
X: 806
Y: 613
Width: 300
Height: 88
Background
Image color: Element-BarDisplayImage-Gray
Own image
Image: <default>
Transparency color: Color [A=255, R=0, G=0, B=0]
Optimized drawing: True
Bar
Diagram type: Scale besides bar
Orientation: Horizontal
Running direction: Left to right
Running direction: Bottom to top
Optimum size for bar: False
Scale
Scale start: 0
Scale end: 100
Main scale: 20
Subscale: 5
Scale line width: 1
Scale color: Color [A=255, R=255, G=255, B=255]
Scale in 3D: True
Element frame: False
Label
Unit: Motor torque %
Font: Not implemented yet
Scale format (C syntax): %.1f
Maximum text width of labels: 34
Text height of labels: 14
Positioning
Horizontal offset: 0
Vertical offset: 0
Horizontal scaling: 0
Vertical scaling: 0
Colors
Graph color: Color [A=255, R=0, G=255, B=0]
Bar background: False
Frame color: Color [A=255, R=212, G=208, B=200]
Switch whole color: False
Use gradient color for bar: False

Color range markers: No markers

*Dip Switch Id: 32
Element name: GenElemInst_33
Type of element: Dip Switch
Tab Order: default
Static optimized: False
Position
X: 531
Y: 242
Width: 70
Height: 70
Variable: GVL.bEnablePID
Image settings
Transparent: False
Transparent color: Color [A=0, R=255, G=255, B=255]
Isotropic type: Isotropic
Horizontal alignment: Left
Vertical alignment: Top
Element behavior: Image toggler
Tap FALSE: False
Background
Image: Element-Switch-DipSwitch-Yellow

†Label Id: 34
Element name: GenElemInst_35
Type of element: Label
Tab Order: default
Static optimized: True
Text ID: 145
Texts
Text: Enable PID
Position
X: 488
Y: 310
Width: 150
Height: 30
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255

●Lamp Id: 35
Element name: GenElemInst_36
Type of element: Lamp
Tab Order: default
Static optimized: False
Position
X: 527
Y: 119

Width: 70
Height: 70
Variable: GVL.bPIDActive
Image settings
Transparent: False
Transparent color: Color [A=0, R=255, G=255, B=255]
Isotropic type: Isotropic
Horizontal alignment: Left
Vertical alignment: Top
Background
Image: Element-Lamp-Lamp1-Green
Label Id: 37
Element name: GenElemInst_38
Type of element: Label
Tab Order: default
Static optimized: True
Text ID: 438
Texts
Text: Enable Active
Position
X: 484
Y: 187
Width: 150
Height: 30
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
Label Id: 39
Element name: GenElemInst_40
Type of element: Label
Tab Order: default
Static optimized: True
Text ID: 611
Texts
Text: Drive error
Position
X: 484
Y: 438
Width: 150
Height: 30
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255

Slider Id: 40
Element name: GenElemInst_41
Type of element: Slider
Tab Order: default
Static optimized: False
Position
X: 840
Y: 533
Width: 215
Height: 59
Variable: GVL.rBallPosCmd
Move to click: True
Scale
Show scale: True
Scale start: 0
Scale end: 550
Main scale: 100
Subscale: 50
Scale format (C syntax): %i
Scale proportion: 0
Bar
Diagram type: Bottom
Diagram type: Right
Orientation: Horizontal
Running direction: Left to right
Running direction: Top to bottom
Dip Switch Id: 42
Element name: GenElemInst_43
Type of element: Dip Switch
Tab Order: default
Static optimized: False
Position
X: 383
Y: 486
Width: 70
Height: 70
Variable: GVL.bHome_Preset
Image settings
Transparent: False
Transparent color: Color [A=0, R=255, G=255, B=255]
Isotropic type: Isotropic
Horizontal alignment: Left
Vertical alignment: Top
Element behavior: Image toggler
Tap FALSE: False
Background
Image: Element-Switch-DipSwitch-Red
Dip Switch Id: 44
Element name: GenElemInst_45

Type of element: Dip Switch
Tab Order: default
Static optimized: False
Position
X: 383
Y: 593
Width: 70
Height: 70
Variable: GVL.rSliderVal
Image settings
Transparent: False
Transparent color: Color [A=0, R=255, G=255, B=255]
Isotropic type: Isotropic
Horizontal alignment: Left
Vertical alignment: Top
Element behavior: Image toggler
Tap FALSE: False
Background
Image: Element-Switch-DipSwitch-Red
Label Id: 46
Element name: GenElemInst_47
Type of element: Label
Tab Order: default
Static optimized: True
Text ID: 117
Texts
Text: Horizontal
Position
X: 342
Y: 558
Width: 150
Height: 30
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
Label Id: 48
Element name: GenElemInst_49
Type of element: Label
Tab Order: default
Static optimized: True
Text ID: 750
Texts
Text: PID Pause
Position
X: 343
Y: 660

Width: 150
Height: 30
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
Slider Id: 51
Element name: GenElemInst_52
Type of element: Slider
Tab Order: default
Static optimized: False
Position
X: 518
Y: 496
Width: 191
Height: 61
Variable: GVL.rPGainPID
Move to click: False
Scale
Show scale: True
Scale start: 0
Scale end: 20
Main scale: 10
Subscale: 5
Scale format (C syntax): %i
Scale proportion: 0
Bar
Diagram type: Bottom
Diagram type: Right
Orientation: Horizontal
Running direction: Left to right
Running direction: Top to bottom
Label Id: 52
Element name: GenElemInst_53
Type of element: Label
Tab Order: default
Static optimized: True
Text ID: 97
Texts
Text: P-Gain
Position
X: 521
Y: 560
Width: 150
Height: 30
Text properties
Horizontal alignment: Centered

Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
Slider Id: 54
Element name: GenElemInst_55
Type of element: Slider
Tab Order: default
Static optimized: False
Position
X: 499
Y: 619
Width: 191
Height: 61
Variable: GVL.IDGain
Move to click: False
Scale
Show scale: True
Scale start: 0
Scale end: 2000
Main scale: 250
Subscale: 100
Scale format (C syntax): %i
Scale proportion: 0
Bar
Diagram type: Bottom
Diagram type: Right
Orientation: Horizontal
Running direction: Left to right
Running direction: Top to bottom
Label Id: 56
Element name: GenElemInst_57
Type of element: Label
Tab Order: default
Static optimized: True
Text ID: 429
Texts
Text: D-Gain
Position
X: 518
Y: 681
Width: 150
Height: 30
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255

Slider Id: 59
Element name: GenElemInst_59
Type of element: Slider
Tab Order: default
Static optimized: False
Position
X: 502
Y: 717
Width: 191
Height: 61
Variable: GVL.IIGain
Move to click: False
Scale
Show scale: True
Scale start: 0
Scale end: 10000
Main scale: 1000
Subscale: 500
Scale format (C syntax): %i
Scale proportion: 0
Bar
Diagram type: Bottom
Diagram type: Right
Orientation: Horizontal
Running direction: Left to right
Running direction: Top to bottom
Label Id: 61
Element name: GenElemInst_61
Type of element: Label
Tab Order: default
Static optimized: True
Text ID: 739
Texts
Text: I-Gain
Position
X: 520
Y: 784
Width: 150
Height: 30
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
Dip Switch Id: 63
Element name: GenElemInst_63
Type of element: Dip Switch
Tab Order: default

Static optimized: False
Position
X: 642
Y: 244
Width: 70
Height: 70
Variable: GVL.bMeanFilter
Image settings
Transparent: False
Transparent color: Color [A=0, R=255, G=255, B=255]
Isotropic type: Isotropic
Horizontal alignment: Left
Vertical alignment: Top
Element behavior: Image toggler
Tap FALSE: False
Background
Image: Element-Switch-DipSwitch-Yellow
TLabel Id: 65
Element name: GenElemInst_65
Type of element: Label
Tab Order: default
Static optimized: True
Text ID: 193
Texts
Text: Mean Filter
Position
X: 600
Y: 314
Width: 150
Height: 30
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
TLabel Id: 67
Element name: GenElemInst_67
Type of element: Label
Tab Order: default
Static optimized: True
Text ID: 494
Texts
Text: Clear Light
Position
X: 606
Y: 437
Width: 150
Height: 30

Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default
Font
Transparency: 255
Slider Id: 69
Element name: GenElemInst_69
Type of element: Slider
Tab Order: default
Static optimized: False
Position
X: 500
Y: 832
Width: 191
Height: 61
Variable: GVL.ITGain
Move to click: False
Scale
Show scale: True
Scale start: 0
Scale end: 100
Main scale: 25
Subscale: 5
Scale format (C syntax): %i
Scale proportion: 0
Bar
Diagram type: Bottom
Diagram type: Right
Orientation: Horizontal
Running direction: Left to right
Running direction: Top to bottom
Label Id: 71
Element name: GenElemInst_71
Type of element: Label
Tab Order: default
Static optimized: True
Text ID: 422
Texts
Text: TControl
Position
X: 519
Y: 894
Width: 150
Height: 30
Text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Text format: Default

Font
Transparency: 255

* Dip Switch Id: 74
Element name: GenElemInst_74
Type of element: Dip Switch
Tab Order: default
Static optimized: False
Position
X: 643
Y: 362
Width: 70
Height: 70
Variable: GVL.blmcCtrlClearError
Image settings
Transparent: False
Transparent color: Color [A=0, R=255, G=255, B=255]
Isotropic type: Isotropic
Horizontal alignment: Left
Vertical alignment: Top
Element behavior: Image toggler
Tap FALSE: False
Background
Image: Element-Switch-DipSwitch-Yellow

→ Slider Id: 77
Element name: GenElemInst_76
Type of element: Slider
Tab Order: default
Static optimized: False
Position
X: 865
Y: 884
Width: 191
Height: 61
Variable: GVL.rLowWindowSet
Move to click: False
Scale
Show scale: True
Scale start: -1
Scale end: 0
Main scale: 0,5
Subscale: 0,25
Scale format (C syntax): %i
Scale proportion: 0

Bar
Diagram type: Bottom
Diagram type: Right
Orientation: Horizontal
Running direction: Left to right
Running direction: Top to bottom

└─ Label Id: 79
 Element name: GenElemInst_78
 Type of element: Label
 Tab Order: default
 Static optimized: True
 Text ID: 306
 Texts
 Text: LowWindow
 Position
 X: 884
 Y: 946
 Width: 150
 Height: 30
 Text properties
 Horizontal alignment: Centered
 Vertical alignment: Centered
 Text format: Default
 Font
 Transparency: 255
└─ Slider Id: 81
 Element name: GenElemInst_80
 Type of element: Slider
 Tab Order: default
 Static optimized: False
 Position
 X: 866
 Y: 996
 Width: 191
 Height: 61
 Variable: GVL.rHighWindowSet
 Move to click: False
 Scale
 Show scale: True
 Scale start: 0
 Scale end: 1
 Main scale: 0,5
 Subscale: 0,25
 Scale format (C syntax): %i
 Scale proportion: 0
 Bar
 Diagram type: Bottom
 Diagram type: Right
 Orientation: Horizontal
 Running direction: Left to right
 Running direction: Top to bottom
└─ Label Id: 83
 Element name: GenElemInst_82
 Type of element: Label
 Tab Order: default

Static optimized: True
Text ID: 58
Texts
 Text: HighWindow
Position
 X: 885
 Y: 1058
 Width: 150
 Height: 30
Text properties
 Horizontal alignment: Centered
 Vertical alignment: Centered
 Text format: Default
 Font
 Transparency: 255
■ Bar Display Id: 85
 Element name: GenElemInst_84
 Type of element: Bar Display
 Tab Order: default
 Static optimized: False
 Value: GVL.rFollowingError
Center
 X: 595
 Y: 973
Position
 X: 445
 Y: 948
 Width: 300
 Height: 80
Background
 Image color: Element-BarDisplayImage-Gray
 Own image
 Image: <default>
 Transparency color: Color [A=255, R=0, G=0, B=0]
 Optimized drawing: True
Bar
 Diagram type: Scale besides bar
 Orientation: Horizontal
 Running direction: Left to right
 Running direction: Bottom to top
 Optimum size for bar: False
Scale
 Scale start: 0
 Scale end: 2
 Main scale: 2
 Subscale: 0,1
 Scale line width: 1
 Scale color: Color [A=255, R=255, G=255, B=255]
 Scale in 3D: True

Element frame: False
Label
Unit: Following Error
Font: Not implemented yet
Scale format (C syntax): %.1f
Maximum text width of labels: 24
Text height of labels: 15
Positioning
Horizontal offset: 0
Vertical offset: 0
Horizontal scaling: 0
Vertical scaling: 0
Colors
Graph color: Color [A=255, R=0, G=255, B=0]
Bar background: False
Frame color: Color [A=255, R=212, G=208, B=200]
Switch whole color: False
Use gradient color for bar: False
Color range markers: No markers
☐ Combo Box, Integer Id: 87
Element name: GenElemInst_86
Type of element: Combo Box, Integer
Tab Order: default
Static optimized: False
Position
X: 336
Y: 737
Width: 150
Height: 30
Variable: GVL.iSignalType
Settings of the list
Number of rows: From style
Row height: -1
Height of image: -1
Width of image: -1
Offset of image: -1
Scroll Bar size: 20
Value range
Limit value range: False
Minimum value: 0
Maximum value: 4
Filter missing text entries: False
Text properties
Usage of: Default style values
Individual text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Font
Transparency: 255

Individual font variables
Size: , <pt>
Individual selection text properties
Horizontal alignment: Centered
Vertical alignment: Centered
Font
Transparency: 255
Individual selection font variables
Size: , <pt>

1.2 Device: Realtime_Data

Information

Name: DataLayer
Vendor: Bosch Rexroth AG
Categories:
Type: 42420
ID: 1001 0000
Version: 1.8.0.0
Description: DataLayer

1.2.1 Device: motion_axs_Axis_Y_cmd_input

DataLayerUser Parameters

Parameters:

Name:	Type:	Value:	Default Value:	Unit:	Description:
Root Uri	STRING	'motion/axs/Axis_Y/cmd-input'			Root Uri

Information

Name: DataLayerUser
Vendor: Bosch Rexroth AG
Categories:
Type: 42430
ID: 1001 0000
Version: 1.8.0.0
Description: DataLayerUser

1.2.1.1 Device: double_1

DataLayerNode Parameters

Parameters:

Name:	Type:	Value:	Default Value:	Unit:	Description:
Node	STRING	"			
No_Of_Buffers	INT	1			
Outputs	LINT	2684485632			
Buffer_Name	String	'double'			
No_Of_Variables	INT	1			
Variable	STRING(2)	'in'			

Information

Name: DataLayerNode
Vendor: Bosch Rexroth AG
Categories:
Type: 42450
ID: 1001 0000
Version: 1.8.0.0
Description: DataLayerNode

1.2.2 Device: motion_cmd_input

DataLayerUser Parameters

Parameters:

Name:	Type:	Value:	Default Value:	Unit:	Description:
Root Uri	STRING	'motion/cmd-input'			Root Uri

Information

Name: DataLayerUser
Vendor: Bosch Rexroth AG
Categories:
Type: 42430
ID: 1001 0000
Version: 1.8.0.0
Description: DataLayerUser

1.2.2.1 Device: double

DataLayerNode Parameters

Parameters:

Name:	Type:	Value:	Default Value:	Unit:	Description:
Node	STRING	"			
No_Of_Buffers	INT	1			
Outputs	LINT	2684485632			
Buffer_Name	String	'double'			
No_Of_Variables	INT	1			
Variable	STRING(2)	'in'			

Information

Name: DataLayerNode
Vendor: Bosch Rexroth AG
Categories:
Type: 42450
ID: 1001 0000
Version: 1.8.0.0
Description: DataLayerNode

1.2.3 Device: motion_state_functions_somo_rt_data

DataLayerUser Parameters

Parameters:

Name:	Type:	Value:	Default Value:	Unit:	Description:
Root Uri	STRING	'motion/state/functions/somo/rt-data'			Root Uri

Information

Name: DataLayerUser
Vendor: Bosch Rexroth AG
Categories:
Type: 42430
ID: 1001 0000
Version: 1.8.0.0
Description: DataLayerUser

1.2.3.1 Device: signals_in

DataLayerNode Parameters

Parameters:

Name:	Type:	Value:	Default Value:	Unit:	Description:
Node	STRING	"			
No_Of_Buffers	INT	1			
Inputs	LINT	2684420096			
Buffer_Name	String	'signals_in'			
No_Of_Variables	INT	100			
Variable	STRING(9)	'signal_00'			
Variable	STRING(9)	'signal_01'			
Variable	STRING(9)	'signal_02'			
Variable	STRING(9)	'signal_03'			
Variable	STRING(9)	'signal_04'			
Variable	STRING(9)	'signal_05'			
Variable	STRING(9)	'signal_06'			
Variable	STRING(9)	'signal_07'			
Variable	STRING(9)	'signal_08'			
Variable	STRING(9)	'signal_09'			
Variable	STRING(9)	'signal_10'			
Variable	STRING(9)	'signal_11'			
Variable	STRING(9)	'signal_12'			
Variable	STRING(9)	'signal_13'			
Variable	STRING(9)	'signal_14'			
Variable	STRING(9)	'signal_15'			
Variable	STRING(9)	'signal_16'			
Variable	STRING(9)	'signal_17'			
Variable	STRING(9)	'signal_18'			
Variable	STRING(9)	'signal_19'			
Variable	STRING(9)	'signal_20'			
Variable	STRING(9)	'signal_21'			
Variable	STRING(9)	'signal_22'			
Variable	STRING(9)	'signal_23'			
Variable	STRING(9)	'signal_24'			
Variable	STRING(9)	'signal_25'			
Variable	STRING(9)	'signal_26'			
Variable	STRING(9)	'signal_27'			
Variable	STRING(9)	'signal_28'			
Variable	STRING(9)	'signal_29'			
Variable	STRING(9)	'signal_30'			
Variable	STRING(9)	'signal_31'			
Variable	STRING(9)	'signal_32'			
Variable	STRING(9)	'signal_33'			
Variable	STRING(9)	'signal_34'			
Variable	STRING(9)	'signal_35'			
Variable	STRING(9)	'signal_36'			
Variable	STRING(9)	'signal_37'			
Variable	STRING(9)	'signal_38'			
Variable	STRING(9)	'signal_39'			

Variable	STRING(9)	'signal_40'
Variable	STRING(9)	'signal_41'
Variable	STRING(9)	'signal_42'
Variable	STRING(9)	'signal_43'
Variable	STRING(9)	'signal_44'
Variable	STRING(9)	'signal_45'
Variable	STRING(9)	'signal_46'
Variable	STRING(9)	'signal_47'
Variable	STRING(9)	'signal_48'
Variable	STRING(9)	'signal_49'
Variable	STRING(9)	'signal_50'
Variable	STRING(9)	'signal_51'
Variable	STRING(9)	'signal_52'
Variable	STRING(9)	'signal_53'
Variable	STRING(9)	'signal_54'
Variable	STRING(9)	'signal_55'
Variable	STRING(9)	'signal_56'
Variable	STRING(9)	'signal_57'
Variable	STRING(9)	'signal_58'
Variable	STRING(9)	'signal_59'
Variable	STRING(9)	'signal_60'
Variable	STRING(9)	'signal_61'
Variable	STRING(9)	'signal_62'
Variable	STRING(9)	'signal_63'
Variable	STRING(9)	'signal_64'
Variable	STRING(9)	'signal_65'
Variable	STRING(9)	'signal_66'
Variable	STRING(9)	'signal_67'
Variable	STRING(9)	'signal_68'
Variable	STRING(9)	'signal_69'
Variable	STRING(9)	'signal_70'
Variable	STRING(9)	'signal_71'
Variable	STRING(9)	'signal_72'
Variable	STRING(9)	'signal_73'
Variable	STRING(9)	'signal_74'
Variable	STRING(9)	'signal_75'
Variable	STRING(9)	'signal_76'
Variable	STRING(9)	'signal_77'
Variable	STRING(9)	'signal_78'
Variable	STRING(9)	'signal_79'
Variable	STRING(9)	'signal_80'
Variable	STRING(9)	'signal_81'
Variable	STRING(9)	'signal_82'
Variable	STRING(9)	'signal_83'
Variable	STRING(9)	'signal_84'
Variable	STRING(9)	'signal_85'
Variable	STRING(9)	'signal_86'
Variable	STRING(9)	'signal_87'
Variable	STRING(9)	'signal_88'
Variable	STRING(9)	'signal_89'
Variable	STRING(9)	'signal_90'
Variable	STRING(9)	'signal_91'
Variable	STRING(9)	'signal_92'

1.2.3.1 Device: signals_in

Variable	STRING(9)	'signal_93'
Variable	STRING(9)	'signal_94'
Variable	STRING(9)	'signal_95'
Variable	STRING(9)	'signal_96'
Variable	STRING(9)	'signal_97'
Variable	STRING(9)	'signal_98'
Variable	STRING(9)	'signal_99'

Information

Name: DataLayerNode
Vendor: Bosch Rexroth AG
Categories:
Type: 42450
ID: 1001 0000
Version: 1.8.0.0
Description: DataLayerNode

1.2.3.2 Device: signals_out

DataLayerNode Parameters

Parameters:

Name:	Type:	Value:	Default Value:	Unit:	Description:
Node	STRING	"			
No_Of_Buffers	INT	1			
Outputs	LINT	2684485632			
Buffer_Name	String	'signals_out'			
No_Of_Variables	INT	100			
Variable	STRING(9)	'signal_00'			
Variable	STRING(9)	'signal_01'			
Variable	STRING(9)	'signal_02'			
Variable	STRING(9)	'signal_03'			
Variable	STRING(9)	'signal_04'			
Variable	STRING(9)	'signal_05'			
Variable	STRING(9)	'signal_06'			
Variable	STRING(9)	'signal_07'			
Variable	STRING(9)	'signal_08'			
Variable	STRING(9)	'signal_09'			
Variable	STRING(9)	'signal_10'			
Variable	STRING(9)	'signal_11'			
Variable	STRING(9)	'signal_12'			
Variable	STRING(9)	'signal_13'			
Variable	STRING(9)	'signal_14'			
Variable	STRING(9)	'signal_15'			
Variable	STRING(9)	'signal_16'			
Variable	STRING(9)	'signal_17'			
Variable	STRING(9)	'signal_18'			
Variable	STRING(9)	'signal_19'			
Variable	STRING(9)	'signal_20'			

1.2.3.2 Device: signals_out

Variable	STRING(9)	'signal_21'
Variable	STRING(9)	'signal_22'
Variable	STRING(9)	'signal_23'
Variable	STRING(9)	'signal_24'
Variable	STRING(9)	'signal_25'
Variable	STRING(9)	'signal_26'
Variable	STRING(9)	'signal_27'
Variable	STRING(9)	'signal_28'
Variable	STRING(9)	'signal_29'
Variable	STRING(9)	'signal_30'
Variable	STRING(9)	'signal_31'
Variable	STRING(9)	'signal_32'
Variable	STRING(9)	'signal_33'
Variable	STRING(9)	'signal_34'
Variable	STRING(9)	'signal_35'
Variable	STRING(9)	'signal_36'
Variable	STRING(9)	'signal_37'
Variable	STRING(9)	'signal_38'
Variable	STRING(9)	'signal_39'
Variable	STRING(9)	'signal_40'
Variable	STRING(9)	'signal_41'
Variable	STRING(9)	'signal_42'
Variable	STRING(9)	'signal_43'
Variable	STRING(9)	'signal_44'
Variable	STRING(9)	'signal_45'
Variable	STRING(9)	'signal_46'
Variable	STRING(9)	'signal_47'
Variable	STRING(9)	'signal_48'
Variable	STRING(9)	'signal_49'
Variable	STRING(9)	'signal_50'
Variable	STRING(9)	'signal_51'
Variable	STRING(9)	'signal_52'
Variable	STRING(9)	'signal_53'
Variable	STRING(9)	'signal_54'
Variable	STRING(9)	'signal_55'
Variable	STRING(9)	'signal_56'
Variable	STRING(9)	'signal_57'
Variable	STRING(9)	'signal_58'
Variable	STRING(9)	'signal_59'
Variable	STRING(9)	'signal_60'
Variable	STRING(9)	'signal_61'
Variable	STRING(9)	'signal_62'
Variable	STRING(9)	'signal_63'
Variable	STRING(9)	'signal_64'
Variable	STRING(9)	'signal_65'
Variable	STRING(9)	'signal_66'
Variable	STRING(9)	'signal_67'
Variable	STRING(9)	'signal_68'
Variable	STRING(9)	'signal_69'
Variable	STRING(9)	'signal_70'
Variable	STRING(9)	'signal_71'
Variable	STRING(9)	'signal_72'
Variable	STRING(9)	'signal_73'

1.2.3.2 Device: signals_out

Variable	STRING(9)	'signal_74'
Variable	STRING(9)	'signal_75'
Variable	STRING(9)	'signal_76'
Variable	STRING(9)	'signal_77'
Variable	STRING(9)	'signal_78'
Variable	STRING(9)	'signal_79'
Variable	STRING(9)	'signal_80'
Variable	STRING(9)	'signal_81'
Variable	STRING(9)	'signal_82'
Variable	STRING(9)	'signal_83'
Variable	STRING(9)	'signal_84'
Variable	STRING(9)	'signal_85'
Variable	STRING(9)	'signal_86'
Variable	STRING(9)	'signal_87'
Variable	STRING(9)	'signal_88'
Variable	STRING(9)	'signal_89'
Variable	STRING(9)	'signal_90'
Variable	STRING(9)	'signal_91'
Variable	STRING(9)	'signal_92'
Variable	STRING(9)	'signal_93'
Variable	STRING(9)	'signal_94'
Variable	STRING(9)	'signal_95'
Variable	STRING(9)	'signal_96'
Variable	STRING(9)	'signal_97'
Variable	STRING(9)	'signal_98'
Variable	STRING(9)	'signal_99'

Information

Name: DataLayerNode
Vendor: Bosch Rexroth AG
Categories:
Type: 42450
ID: 1001 0000
Version: 1.8.0.0
Description: DataLayerNode

1.2.4 Device: ethercat_master_instances_ethercatmaster

DataLayerUser Parameters

Parameters:

Name:	Type:	Value:	Default Value:	Unit:	Description:
Root Uri	STRING	'fieldbuses/ethercat/master/instances/ethercatmaster/realtime_d ata'			Root Uri

Information

Name: DataLayerUser
Vendor: Bosch Rexroth AG
Categories:
Type: 42430
ID: 1001 0000
Version: 1.8.0.0
Description: DataLayerUser

1.2.4.1 Device: ctrlX_DRIVE_XMS_SoE_

DataLayerNode Parameters

Parameters:

Name:	Type:	Value:	Default Value:	Unit:	Description:
Node	STRING	'ctrlX_DRIVE_XMS_SoE_'			
No_Of_Buffers	INT	2			
Inputs	LINT	2684420096			
Outputs	LINT	2684485632			
Buffer_Name	String	'input'			
No_Of_Variables	INT	6			
Variable	STRING(20)	'AT.Drive_status_word'			
Variable	STRING(17)	'AT.Analog_input_1'			
Variable	STRING(27)	'AT.Power_supply_status_word'			
Variable	STRING(28)	'AT.Position_feedback_value_1'			
Variable	STRING(20)	'AT.Motor_temperature'			
Variable	STRING(39)	'AT.Effective_torque_force_command_value'			
Buffer_Name	String	'double'			
No_Of_Variables	INT	3			
Variable	STRING(23)	'MDT.Master_control_word'			
Variable	STRING(24)	'MDT.DC_Bus_Power_Control'			
Variable	STRING(26)	'MDT.Position_command_value'			

Information

Name: DataLayerNode
Vendor: Bosch Rexroth AG
Categories:
Type: 42450
ID: 1001 0000
Version: 1.8.0.0
Description: DataLayerNode

2 GlobalTextList: GlobalTextList