

The image shows a complex industrial machine, possibly a robotic arm or a manufacturing component, illuminated with a strong blue light. The machine has various metal parts, wires, and a central component that looks like a motor or actuator. The background is dark, making the blue-lit machine stand out.

SIEMENS

SIMOTION Engineering

The SCOUT Engineering System

SIMOTION SCOUT

One system for all engineering tasks

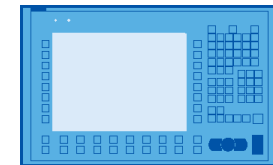
Overview

Task-oriented programming languages

Test support

Commissioning support

Engineering tasks



HMI

- Configuring

Open-loop control

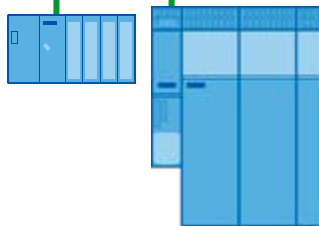
- Parameterization
- Programming
- Commissioning, diagnostics

Communications

- Configuration

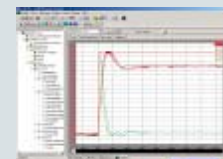
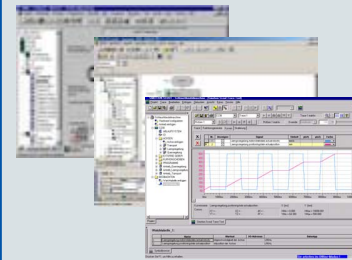
Drive

- Commissioning

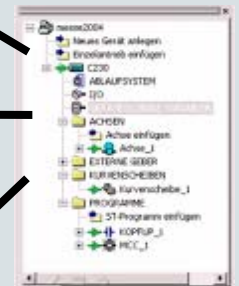


Engineering workbench

SIMOTION SCOUT



Centrally called from the project tree



Common data management

Standard, unified and consistent engineering for the complete system

SIMOTION SCOUT

Workbench with all of the tools integrated

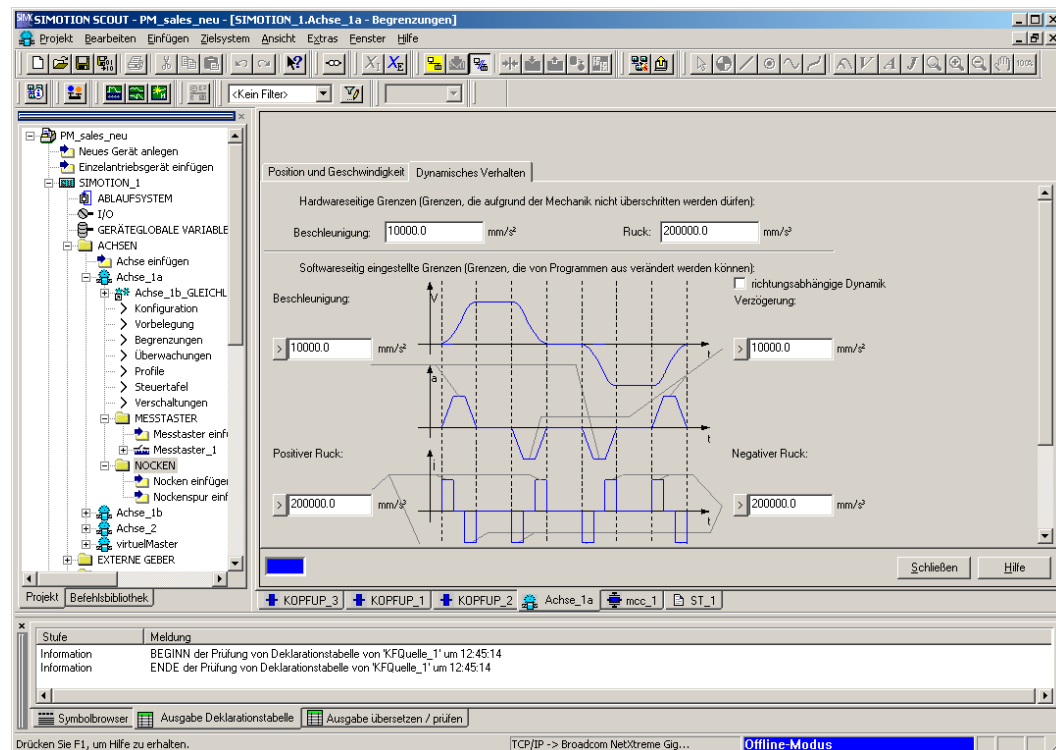
Overview

Task-oriented programming languages

Test support

Commissioning support

- SCOUT allows all SIMOTION objects to be configured
- Has all of the editors required for programming
- Commissioning and test tools are included in the workbench



Hardware is configured just the same as for TIA

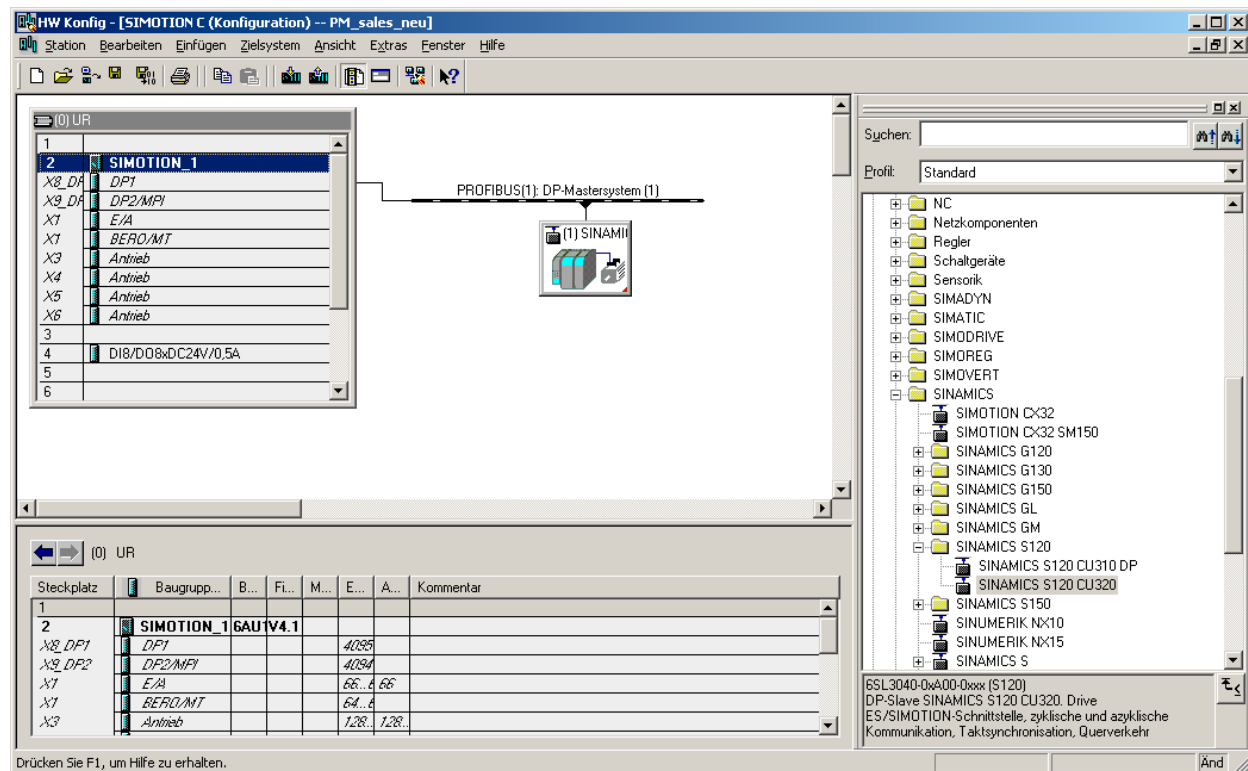
Overview

Task-oriented programming languages

Test support

Commissioning support

- The hardware components are configured exactly the same as for STEP7
- Configuration of the drives and I/O
- Configuration of bus systems



Task-oriented programming languages

Overview

Task-oriented programming languages

Test support

Commissioning support

SIMOTION offers various programming languages – which can also be mixed in an application:

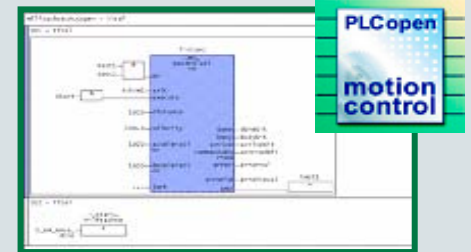
Structured Text (ST)

```

1000  VAR_GLOBAL VAR
1001  INT Counter := 0;
1002  END_VAR
1003
1004  FUNCTION_BLOCK FB1
1005  VAR
1006  INT Counter := 0;
1007  END_VAR
1008  FB1_SETPOINT : INT := 0;
1009  FB1_RESET : BOOL := FALSE;
1010  FB1_COUNTER : INT := 0;
1011  END_VAR
1012  FB1_SETPOINT := 0;
1013  FB1_RESET := FALSE;
1014  FB1_COUNTER := 0;
1015  IF FB1_RESET THEN
1016  Counter := 0;
1017  END_IF
1018  IF FB1_SETPOINT > 0 THEN
1019  Counter := Counter + 1;
1020  END_IF
1021  END_FUNCTION_BLOCK
1022
1023  FB1 FB1_COUNTER;
1024  FB1_COUNTER := 0;
1025  END_VAR
1026
1027  END_FUNCTION_BLOCK
1028
1029  END_FUNCTION_BLOCK
1030
1031  END_FUNCTION_BLOCK
1032
1033  END_FUNCTION_BLOCK
1034
1035  END_FUNCTION_BLOCK
1036
1037  END_FUNCTION_BLOCK
1038
1039  END_FUNCTION_BLOCK
1040
1041  END_FUNCTION_BLOCK
1042
1043  END_FUNCTION_BLOCK
1044
1045  END_FUNCTION_BLOCK
1046
1047  END_FUNCTION_BLOCK
1048
1049  END_FUNCTION_BLOCK
1050
1051  END_FUNCTION_BLOCK
1052
1053  END_FUNCTION_BLOCK
1054
1055  END_FUNCTION_BLOCK
1056
1057  END_FUNCTION_BLOCK
1058
1059  END_FUNCTION_BLOCK
1060
1061  END_FUNCTION_BLOCK
1062
1063  END_FUNCTION_BLOCK
1064
1065  END_FUNCTION_BLOCK
1066
1067  END_FUNCTION_BLOCK
1068
1069  END_FUNCTION_BLOCK
1070
1071  END_FUNCTION_BLOCK
1072
1073  END_FUNCTION_BLOCK
1074
1075  END_FUNCTION_BLOCK
1076
1077  END_FUNCTION_BLOCK
1078
1079  END_FUNCTION_BLOCK
1080
1081  END_FUNCTION_BLOCK
1082
1083  END_FUNCTION_BLOCK
1084
1085  END_FUNCTION_BLOCK
1086
1087  END_FUNCTION_BLOCK
1088
1089  END_FUNCTION_BLOCK
1090
1091  END_FUNCTION_BLOCK
1092
1093  END_FUNCTION_BLOCK
1094
1095  END_FUNCTION_BLOCK
1096
1097  END_FUNCTION_BLOCK
1098
1099  END_FUNCTION_BLOCK
1100
1101  END_FUNCTION_BLOCK
1102
1103  END_FUNCTION_BLOCK
1104
1105  END_FUNCTION_BLOCK
1106
1107  END_FUNCTION_BLOCK
1108
1109  END_FUNCTION_BLOCK
1110
1111  END_FUNCTION_BLOCK
1112
1113  END_FUNCTION_BLOCK
1114
1115  END_FUNCTION_BLOCK
1116
1117  END_FUNCTION_BLOCK
1118
1119  END_FUNCTION_BLOCK
1120
1121  END_FUNCTION_BLOCK
1122
1123  END_FUNCTION_BLOCK
1124
1125  END_FUNCTION_BLOCK
1126
1127  END_FUNCTION_BLOCK
1128
1129  END_FUNCTION_BLOCK
1130
1131  END_FUNCTION_BLOCK
1132
1133  END_FUNCTION_BLOCK
1134
1135  END_FUNCTION_BLOCK
1136
1137  END_FUNCTION_BLOCK
1138
1139  END_FUNCTION_BLOCK
1140
1141  END_FUNCTION_BLOCK
1142
1143  END_FUNCTION_BLOCK
1144
1145  END_FUNCTION_BLOCK
1146
1147  END_FUNCTION_BLOCK
1148
1149  END_FUNCTION_BLOCK
1150
1151  END_FUNCTION_BLOCK
1152
1153  END_FUNCTION_BLOCK
1154
1155  END_FUNCTION_BLOCK
1156
1157  END_FUNCTION_BLOCK
1158
1159  END_FUNCTION_BLOCK
1160
1161  END_FUNCTION_BLOCK
1162
1163  END_FUNCTION_BLOCK
1164
1165  END_FUNCTION_BLOCK
1166
1167  END_FUNCTION_BLOCK
1168
1169  END_FUNCTION_BLOCK
1170
1171  END_FUNCTION_BLOCK
1172
1173  END_FUNCTION_BLOCK
1174
1175  END_FUNCTION_BLOCK
1176
1177  END_FUNCTION_BLOCK
1178
1179  END_FUNCTION_BLOCK
1180
1181  END_FUNCTION_BLOCK
1182
1183  END_FUNCTION_BLOCK
1184
1185  END_FUNCTION_BLOCK
1186
1187  END_FUNCTION_BLOCK
1188
1189  END_FUNCTION_BLOCK
1190
1191  END_FUNCTION_BLOCK
1192
1193  END_FUNCTION_BLOCK
1194
1195  END_FUNCTION_BLOCK
1196
1197  END_FUNCTION_BLOCK
1198
1199  END_FUNCTION_BLOCK
1200
1201  END_FUNCTION_BLOCK
1202
1203  END_FUNCTION_BLOCK
1204
1205  END_FUNCTION_BLOCK
1206
1207  END_FUNCTION_BLOCK
1208
1209  END_FUNCTION_BLOCK
1210
1211  END_FUNCTION_BLOCK
1212
1213  END_FUNCTION_BLOCK
1214
1215  END_FUNCTION_BLOCK
1216
1217  END_FUNCTION_BLOCK
1218
1219  END_FUNCTION_BLOCK
1220
1221  END_FUNCTION_BLOCK
1222
1223  END_FUNCTION_BLOCK
1224
1225  END_FUNCTION_BLOCK
1226
1227  END_FUNCTION_BLOCK
1228
1229  END_FUNCTION_BLOCK
1230
1231  END_FUNCTION_BLOCK
1232
1233  END_FUNCTION_BLOCK
1234
1235  END_FUNCTION_BLOCK
1236
1237  END_FUNCTION_BLOCK
1238
1239  END_FUNCTION_BLOCK
1240
1241  END_FUNCTION_BLOCK
1242
1243  END_FUNCTION_BLOCK
1244
1245  END_FUNCTION_BLOCK
1246
1247  END_FUNCTION_BLOCK
1248
1249  END_FUNCTION_BLOCK
1250
1251  END_FUNCTION_BLOCK
1252
1253  END_FUNCTION_BLOCK
1254
1255  END_FUNCTION_BLOCK
1256
1257  END_FUNCTION_BLOCK
1258
1259  END_FUNCTION_BLOCK
1260
1261  END_FUNCTION_BLOCK
1262
1263  END_FUNCTION_BLOCK
1264
1265  END_FUNCTION_BLOCK
1266
1267  END_FUNCTION_BLOCK
1268
1269  END_FUNCTION_BLOCK
1270
1271  END_FUNCTION_BLOCK
1272
1273  END_FUNCTION_BLOCK
1274
1275  END_FUNCTION_BLOCK
1276
1277  END_FUNCTION_BLOCK
1278
1279  END_FUNCTION_BLOCK
1280
1281  END_FUNCTION_BLOCK
1282
1283  END_FUNCTION_BLOCK
1284
1285  END_FUNCTION_BLOCK
1286
1287  END_FUNCTION_BLOCK
1288
1289  END_FUNCTION_BLOCK
1290
1291  END_FUNCTION_BLOCK
1292
1293  END_FUNCTION_BLOCK
1294
1295  END_FUNCTION_BLOCK
1296
1297  END_FUNCTION_BLOCK
1298
1299  END_FUNCTION_BLOCK
1300
1301  END_FUNCTION_BLOCK
1302
1303  END_FUNCTION_BLOCK
1304
1305  END_FUNCTION_BLOCK
1306
1307  END_FUNCTION_BLOCK
1308
1309  END_FUNCTION_BLOCK
1310
1311  END_FUNCTION_BLOCK
1312
1313  END_FUNCTION_BLOCK
1314
1315  END_FUNCTION_BLOCK
1316
1317  END_FUNCTION_BLOCK
1318
1319  END_FUNCTION_BLOCK
1320
1321  END_FUNCTION_BLOCK
1322
1323  END_FUNCTION_BLOCK
1324
1325  END_FUNCTION_BLOCK
1326
1327  END_FUNCTION_BLOCK
1328
1329  END_FUNCTION_BLOCK
1330
1331  END_FUNCTION_BLOCK
1332
1333  END_FUNCTION_BLOCK
1334
1335  END_FUNCTION_BLOCK
1336
1337  END_FUNCTION_BLOCK
1338
1339  END_FUNCTION_BLOCK
1340
1341  END_FUNCTION_BLOCK
1342
1343  END_FUNCTION_BLOCK
1344
1345  END_FUNCTION_BLOCK
1346
1347  END_FUNCTION_BLOCK
1348
1349  END_FUNCTION_BLOCK
1350
1351  END_FUNCTION_BLOCK
1352
1353  END_FUNCTION_BLOCK
1354
1355  END_FUNCTION_BLOCK
1356
1357  END_FUNCTION_BLOCK
1358
1359  END_FUNCTION_BLOCK
1360
1361  END_FUNCTION_BLOCK
1362
1363  END_FUNCTION_BLOCK
1364
1365  END_FUNCTION_BLOCK
1366
1367  END_FUNCTION_BLOCK
1368
1369  END_FUNCTION_BLOCK
1370
1371  END_FUNCTION_BLOCK
1372
1373  END_FUNCTION_BLOCK
1374
1375  END_FUNCTION_BLOCK
1376
1377  END_FUNCTION_BLOCK
1378
1379  END_FUNCTION_BLOCK
1380
1381  END_FUNCTION_BLOCK
1382
1383  END_FUNCTION_BLOCK
1384
1385  END_FUNCTION_BLOCK
1386
1387  END_FUNCTION_BLOCK
1388
1389  END_FUNCTION_BLOCK
1390
1391  END_FUNCTION_BLOCK
1392
1393  END_FUNCTION_BLOCK
1394
1395  END_FUNCTION_BLOCK
1396
1397  END_FUNCTION_BLOCK
1398
1399  END_FUNCTION_BLOCK
1400
1401  END_FUNCTION_BLOCK
1402
1403  END_FUNCTION_BLOCK
1404
1405  END_FUNCTION_BLOCK
1406
1407  END_FUNCTION_BLOCK
1408
1409  END_FUNCTION_BLOCK
1410
1411  END_FUNCTION_BLOCK
1412
1413  END_FUNCTION_BLOCK
1414
1415  END_FUNCTION_BLOCK
1416
1417  END_FUNCTION_BLOCK
1418
1419  END_FUNCTION_BLOCK
1420
1421  END_FUNCTION_BLOCK
1422
1423  END_FUNCTION_BLOCK
1424
1425  END_FUNCTION_BLOCK
1426
1427  END_FUNCTION_BLOCK
1428
1429  END_FUNCTION_BLOCK
1430
1431  END_FUNCTION_BLOCK
1432
1433  END_FUNCTION_BLOCK
1434
1435  END_FUNCTION_BLOCK
1436
1437  END_FUNCTION_BLOCK
1438
1439  END_FUNCTION_BLOCK
1440
1441  END_FUNCTION_BLOCK
1442
1443  END_FUNCTION_BLOCK
1444
1445  END_FUNCTION_BLOCK
1446
1447  END_FUNCTION_BLOCK
1448
1449  END_FUNCTION_BLOCK
1450
1451  END_FUNCTION_BLOCK
1452
1453  END_FUNCTION_BLOCK
1454
1455  END_FUNCTION_BLOCK
1456
1457  END_FUNCTION_BLOCK
1458
1459  END_FUNCTION_BLOCK
1460
1461  END_FUNCTION_BLOCK
1462
1463  END_FUNCTION_BLOCK
1464
1465  END_FUNCTION_BLOCK
1466
1467  END_FUNCTION_BLOCK
1468
1469  END_FUNCTION_BLOCK
1470
1471  END_FUNCTION_BLOCK
1472
1473  END_FUNCTION_BLOCK
1474
1475  END_FUNCTION_BLOCK
1476
1477  END_FUNCTION_BLOCK
1478
1479  END_FUNCTION_BLOCK
1480
1481  END_FUNCTION_BLOCK
1482
1483  END_FUNCTION_BLOCK
1484
1485  END_FUNCTION_BLOCK
1486
1487  END_FUNCTION_BLOCK
1488
1489  END_FUNCTION_BLOCK
1490
1491  END_FUNCTION_BLOCK
1492
1493  END_FUNCTION_BLOCK
1494
1495  END_FUNCTION_BLOCK
1496
1497  END_FUNCTION_BLOCK
1498
1499  END_FUNCTION_BLOCK
1500
1501  END_FUNCTION_BLOCK
1502
1503  END_FUNCTION_BLOCK
1504
1505  END_FUNCTION_BLOCK
1506
1507  END_FUNCTION_BLOCK
1508
1509  END_FUNCTION_BLOCK
1510
1511  END_FUNCTION_BLOCK
1512
1513  END_FUNCTION_BLOCK
1514
1515  END_FUNCTION_BLOCK
1516
1517  END_FUNCTION_BLOCK
1518
1519  END_FUNCTION_BLOCK
1520
1521  END_FUNCTION_BLOCK
1522
1523  END_FUNCTION_BLOCK
1524
1525  END_FUNCTION_BLOCK
1526
1527  END_FUNCTION_BLOCK
1528
1529  END_FUNCTION_BLOCK
1530
1531  END_FUNCTION_BLOCK
1532
1533  END_FUNCTION_BLOCK
1534
1535  END_FUNCTION_BLOCK
1536
1537  END_FUNCTION_BLOCK
1538
1539  END_FUNCTION_BLOCK
1540
1541  END_FUNCTION_BLOCK
1542
1543  END_FUNCTION_BLOCK
1544
1545  END_FUNCTION_BLOCK
1546
1547  END_FUNCTION_BLOCK
1548
1549  END_FUNCTION_BLOCK
1550
1551  END_FUNCTION_BLOCK
1552
1553  END_FUNCTION_BLOCK
1554
1555  END_FUNCTION_BLOCK
1556
1557  END_FUNCTION_BLOCK
1558
1559  END_FUNCTION_BLOCK
1560
1561  END_FUNCTION_BLOCK
1562
1563  END_FUNCTION_BLOCK
1564
1565  END_FUNCTION_BLOCK
1566
1567  END_FUNCTION_BLOCK
1568
1569  END_FUNCTION_BLOCK
1570
1571  END_FUNCTION_BLOCK
1572
1573  END_FUNCTION_BLOCK
1574
1575  END_FUNCTION_BLOCK
1576
1577  END_FUNCTION_BLOCK
1578
1579  END_FUNCTION_BLOCK
1580
1581  END_FUNCTION_BLOCK
1582
1583  END_FUNCTION_BLOCK
1584
1585  END_FUNCTION_BLOCK
1586
1587  END_FUNCTION_BLOCK
1588
1589  END_FUNCTION_BLOCK
1590
1591  END_FUNCTION_BLOCK
1592
1593  END_FUNCTION_BLOCK
1594
1595  END_FUNCTION_BLOCK
1596
1597  END_FUNCTION_BLOCK
1598
1599  END_FUNCTION_BLOCK
1600
1601  END_FUNCTION_BLOCK
1602
1603  END_FUNCTION_BLOCK
1604
1605  END_FUNCTION_BLOCK
1606
1607  END_FUNCTION_BLOCK
1608
1609  END_FUNCTION_BLOCK
1610
1611  END_FUNCTION_BLOCK
1612
1613  END_FUNCTION_BLOCK
1614
1615  END_FUNCTION_BLOCK
1616
1617  END_FUNCTION_BLOCK
1618
1619  END_FUNCTION_BLOCK
1620
1621  END_FUNCTION_BLOCK
1622
1623  END_FUNCTION_BLOCK
1624
1625  END_FUNCTION_BLOCK
1626
1627  END_FUNCTION_BLOCK
1628
1629  END_FUNCTION_BLOCK
1630
1631  END_FUNCTION_BLOCK
1632
1633  END_FUNCTION_BLOCK
1634
1635  END_FUNCTION_BLOCK
1636
1637  END_FUNCTION_BLOCK
1638
1639  END_FUNCTION_BLOCK
1640
1641  END_FUNCTION_BLOCK
1642
1643  END_FUNCTION_BLOCK
1644
1645  END_FUNCTION_BLOCK
1646
1647  END_FUNCTION_BLOCK
1648
1649  END_FUNCTION_BLOCK
1650
1651  END_FUNCTION_BLOCK
1652
1653  END_FUNCTION_BLOCK
1654
1655  END_FUNCTION_BLOCK
1656
1657  END_FUNCTION_BLOCK
1658
1659  END_FUNCTION_BLOCK
1660
1661  END_FUNCTION_BLOCK
1662
1663  END_FUNCTION_BLOCK
1664
1665  END_FUNCTION_BLOCK
1666
1667  END_FUNCTION_BLOCK
1668
1669  END_FUNCTION_BLOCK
1670
1671  END_FUNCTION_BLOCK
1672
1673  END_FUNCTION_BLOCK
1674
1675  END_FUNCTION_BLOCK
1676
1677  END_FUNCTION_BLOCK
1678
1679  END_FUNCTION_BLOCK
1680
1681  END_FUNCTION_BLOCK
1682
1683  END_FUNCTION_BLOCK
1684
1685  END_FUNCTION_BLOCK
1686
1687  END_FUNCTION_BLOCK
1688
1689  END_FUNCTION_BLOCK
1690
1691  END_FUNCTION_BLOCK
1692
1693  END_FUNCTION_BLOCK
1694
1695  END_FUNCTION_BLOCK
1696
1697  END_FUNCTION_BLOCK
1698
1699  END_FUNCTION_BLOCK
1700
1701  END_FUNCTION_BLOCK
1702
1703  END_FUNCTION_BLOCK
1704
1705  END_FUNCTION_BLOCK
1706
1707  END_FUNCTION_BLOCK
1708
1709  END_FUNCTION_BLOCK
1710
1711  END_FUNCTION_BLOCK
1712
1713  END_FUNCTION_BLOCK
1714
1715  END_FUNCTION_BLOCK
1716
1717  END_FUNCTION_BLOCK
1718
1719  END_FUNCTION_BLOCK
1720
1721  END_FUNCTION_BLOCK
1722
1723  END_FUNCTION_BLOCK
1724
1725  END_FUNCTION_BLOCK
1726
1727  END_FUNCTION_BLOCK
1728
1729  END_FUNCTION_BLOCK
1730
1731  END_FUNCTION_BLOCK
1732
1733  END_FUNCTION_BLOCK
1734
1735  END_FUNCTION_BLOCK
1736
1737  END_FUNCTION_BLOCK
1738
1739  END_FUNCTION_BLOCK
1740
1741  END_FUNCTION_BLOCK
1742
1743  END_FUNCTION_BLOCK
1744
1745  END_FUNCTION_BLOCK
1746
1747  END_FUNCTION_BLOCK
1748
1749  END_FUNCTION_BLOCK
1750
1751  END_FUNCTION_BLOCK
1752
1753  END_FUNCTION_BLOCK
1754
1755  END_FUNCTION_BLOCK
1756
1757  END_FUNCTION_BLOCK
1758
1759  END_FUNCTION_BLOCK
1760
1761  END_FUNCTION_BLOCK
1762
1763  END_FUNCTION_BLOCK
1764
1765  END_FUNCTION_BLOCK
1766
1767  END_FUNCTION_BLOCK
1768
1769  END_FUNCTION_BLOCK
1770
1771  END_FUNCTION_BLOCK
1772
1773  END_FUNCTION_BLOCK
1774
1775  END_FUNCTION_BLOCK
1776
1777  END_FUNCTION_BLOCK
1778
1779  END_FUNCTION_BLOCK
1780
1781  END_FUNCTION_BLOCK
1782
1783  END_FUNCTION_BLOCK
1784
1785  END_FUNCTION_BLOCK
1786
1787  END_FUNCTION_BLOCK
1788
1789  END_FUNCTION_BLOCK
1790
1791  END_FUNCTION_BLOCK
1792
1793  END_FUNCTION_BLOCK
1794
1795  END_FUNCTION_BLOCK
1796
1797  END_FUNCTION_BLOCK
1798
1799  END_FUNCTION_BLOCK
1800
1801  END_FUNCTION_BLOCK
1802
1803  END_FUNCTION_BLOCK
1804
1805  END_FUNCTION_BLOCK
1806
1807  END_FUNCTION_BLOCK
1808
1809  END_FUNCTION_BLOCK
1810
1811  END_FUNCTION_BLOCK
1812
1813  END_FUNCTION_BLOCK
1814
1815  END_FUNCTION_BLOCK
1816
1817  END_FUNCTION_BLOCK
1818
1819  END_FUNCTION_BLOCK
1820
1821  END_FUNCTION_BLOCK
1822
1823  END_FUNCTION_BLOCK
1824
1825  END_FUNCTION_BLOCK
1826
1827  END_FUNCTION_BLOCK
1828
1829  END_FUNCTION_BLOCK
1830
1831  END_FUNCTION_BLOCK
1832
1833  END_FUNCTION_BLOCK
1834
1835  END_FUNCTION_BLOCK
1836
1837  END_FUNCTION_BLOCK
1838
1839  END_FUNCTION_BLOCK
1840
1841  END_FUNCTION_BLOCK
1842
1843  END_FUNCTION_BLOCK
1844
1845  END_FUNCTION_BLOCK
1846
1847  END_FUNCTION_BLOCK
1848
1849  END_FUNCTION_BLOCK
1850
1851  END_FUNCTION_BLOCK
1852
1853  END_FUNCTION_BLOCK
1854
1855  END_FUNCTION_BLOCK
1856
1857  END_FUNCTION_BLOCK
1858
1859  END_FUNCTION_BLOCK
1860
1861  END_FUNCTION_BLOCK
1862
1863  END_FUNCTION_BLOCK
1864
1865  END_FUNCTION_BLOCK
1866
1867  END_FUNCTION_BLOCK
1868
1869  END_FUNCTION_BLOCK
1870
1871  END_FUNCTION_BLOCK
1872
1873  END_FUNCTION_BLOCK
1874
1875  END_FUNCTION_BLOCK
1876
1877  END_FUNCTION_BLOCK
1878
1879  END_FUNCTION_BLOCK
1880
1881  END_FUNCTION_BLOCK
1882
1883  END_FUNCTION_BLOCK
1884
1885  END_FUNCTION_BLOCK
1886
1887  END_FUNCTION_BLOCK
1888
1889  END_FUNCTION_BLOCK
1890
1891  END_FUNCTION_BLOCK
1892
1893  END_FUNCTION_BLOCK
1894
1895  END_FUNCTION_BLOCK
1896
1897  END_FUNCTION_BLOCK
1898
1899  END_FUNCTION_BLOCK
1900
1901  END_FUNCTION_BLOCK
1902
1903  END_FUNCTION_BLOCK
1904
1905  END_FUNCTION_BLOCK
1906
1907  END_FUNCTION_BLOCK
1908
1909  END_FUNCTION_BLOCK
1910
1911  END_FUNCTION_BLOCK
1912
1913  END_FUNCTION_BLOCK
1914
1915  END_FUNCTION_BLOCK
1916
1917  END_FUNCTION_BLOCK
1918
1919  END_FUNCTION_BLOCK
1920
1921  END_FUNCTION_BLOCK
1922
1923  END_FUNCTION_BLOCK
1924
1925  END_FUNCTION_BLOCK
1926
1927  END_FUNCTION_BLOCK
1928
1929  END_FUNCTION_BLOCK
1930
1931  END_FUNCTION_BLOCK
1932
1933  END_FUNCTION_BLOCK
1934
1935  END_FUNCTION_BLOCK
1936
1937  END_FUNCTION_BLOCK
1938
1939  END_FUNCTION_BLOCK
1940
1941  END_FUNCTION_BLOCK
1942
1943  END_FUNCTION_BLOCK
1944
1945  END_FUNCTION_BLOCK
1946
1947  END_FUNCTION_BLOCK
1948
1949  END_FUNCTION_BLOCK
1950
1951  END_FUNCTION_BLOCK
1952
1953  END_FUNCTION_BLOCK
1954
1955  END_FUNCTION_BLOCK
1956
1957  END_FUNCTION_BLOCK
1958
1959  END_FUNCTION_BLOCK
1960
1961  END_FUNCTION_BLOCK
1962
1963  END_FUNCTION_BLOCK
1964
1965  END_FUNCTION_BLOCK
1966
1967  END_FUNCTION_BLOCK
1968
1969  END_FUNCTION_BLOCK
1970
1971  END_FUNCTION_BLOCK
1972
1973  END_FUNCTION_BLOCK
1974
1975  END_FUNCTION_BLOCK
1976
1977  END_FUNCTION_BLOCK
1978
1979  END_FUNCTION_BLOCK
1980
1981  END_FUNCTION_BLOCK
1982
1983  END_FUNCTION_BLOCK
1984
1985  END_FUNCTION_BLOCK
1986
1987  END_FUNCTION_BLOCK
1988
1989  END_FUNCTION_BLOCK
1990
1991  END_FUNCTION_BLOCK
1992
1993  END_FUNCTION_BLOCK
1994
1995  END_FUNCTION_BLOCK
1996
1997  END_FUNCTION_BLOCK
1998
1999  END_FUNCTION_BLOCK
2000  END_FUNCTION_BLOCK
    
```

- suitable for all tasks

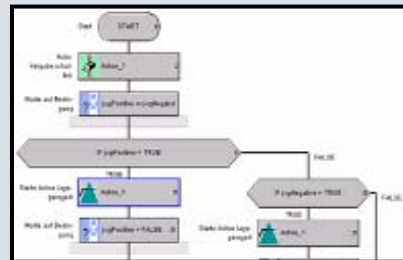
Ladder diagram (LAD)



Function Block Diagram (FBD)

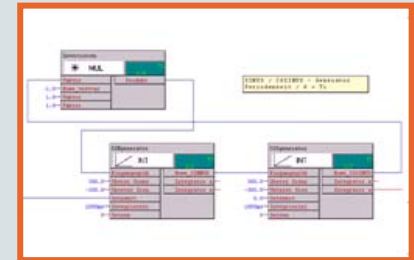
- especially for cyclic tasks (e.g. logic)

Motion Control Chart (MCC)



- especially for sequential tasks (e.g. Motion Control)

Drive Control Chart (DCC)



- especially for continuous tasks (e.g. closed-loop control & arithmetic functions)

The optimum programming language for each and every task

Text-based high-level language programming

Overview

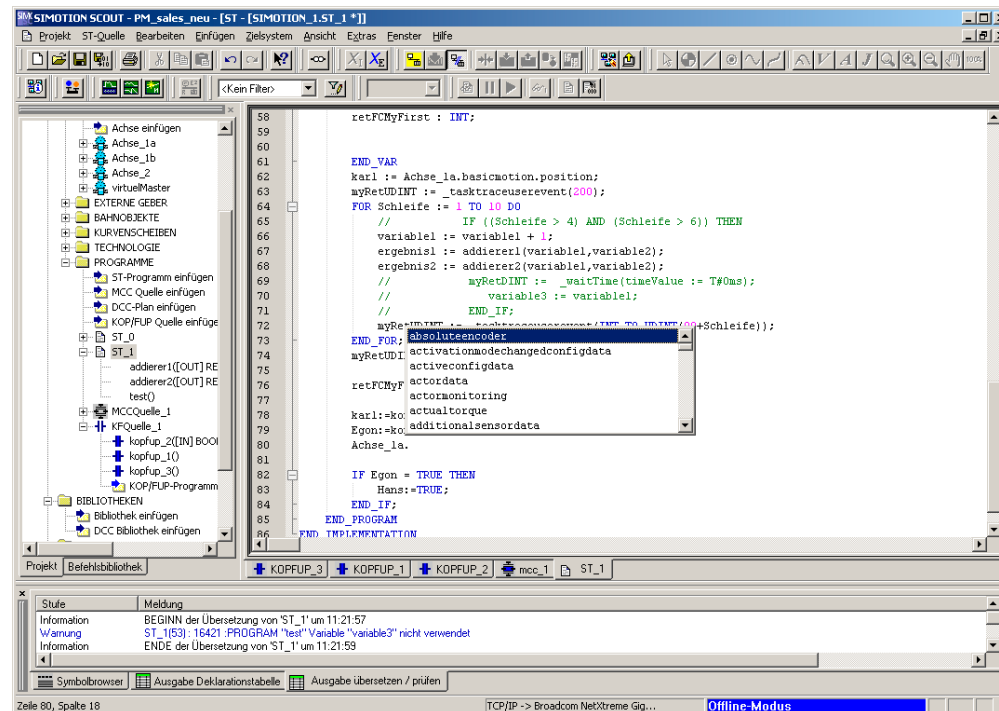
Task-oriented programming languages

Test support

Commissioning support

- User-friendly editor with auto complete – automatic indentation, folding, syntax coloring, ...
- Suitable for all tasks

Programming in a high-level language



Ladder diagram and function chart to program logic

Overview

Task-oriented programming languages

Test support

Commissioning support

- LAD/FBD Editor with input support when calling function blocks
- Especially suitable for programming logic functions

Programming in LAD/FBD

Parameterization is supported

The screenshot shows the SIMOTION SCOUT software interface. On the left, a project tree displays the file structure. The main window shows a ladder diagram with a function block call for 'mc_jog'. A dialog box titled 'Aufrufparameter Eingeben' (Call Parameters Input) is open, showing the function block name and instance name. Below the dialog, a table lists the parameters for the 'mc_jog' function block.

Index	Name	Ein/Aus	Datentyp	Wert	Defaultwert
1	axis	VAR_INPUT	POSAXIS	Achse_1b	
2	enable	VAR_INPUT	BOOL	...	0
3	forward	VAR_INPUT	BOOL	...	0
4	backward	VAR_INPUT	BOOL	...	0
5	modeinc	VAR_INPUT	BOOL	...	0
6	movement	VAR_INPUT	LREAL	...	0.0
7	velocity	VAR_INPUT	LREAL	...	-1.0
8	acceleration	VAR_INPUT	LREAL	...	-1.0
9	deceleration	VAR_INPUT	LREAL	...	-1.0
10	jerk	VAR_INPUT	LREAL	...	-1.0
11	done	VAR_OUTPUT	BOOL	...	
12	busy	VAR_OUTPUT	BOOL	...	
13	active	VAR_OUTPUT	BOOL	...	
14	commandabort	VAR_OUTPUT	BOOL	...	

The dialog box also includes fields for 'Funktionsbaustein' (Function Block) set to '_mc_jog' and 'Instanz' (Instance) set to '???'. Buttons for 'OK', 'Abbrechen' (Cancel), and 'Hilfe' (Help) are visible at the bottom of the dialog.

Graphic motion control programming

Overview

Task-oriented programming languages

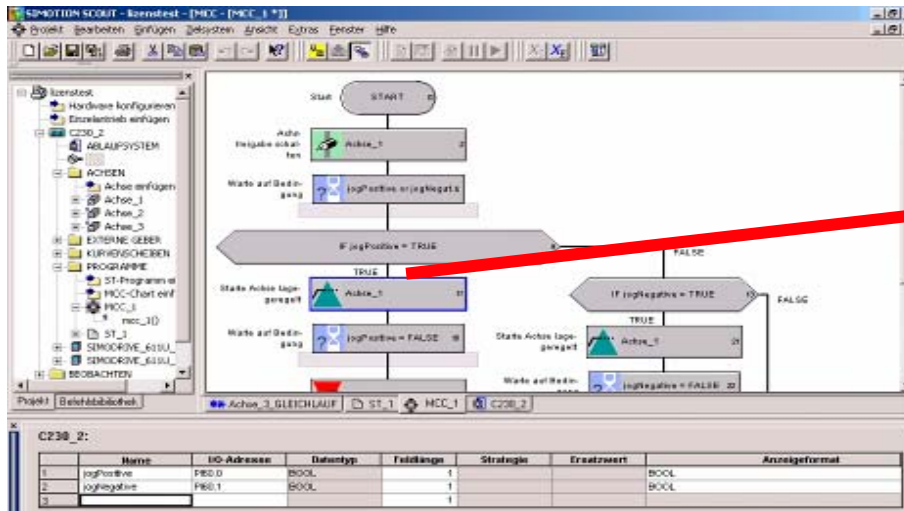
Test support

Commissioning support

- Graphic programming as **flow diagram**
- Parameters are entered in function-specific **parameterizing screen forms**
- Can be structured and are transparent by forming modules, with zoom-in & zoom-out
- Diagnostics using graphic step tracing and single-step mode

Programming as flow diagram in MCC

Parameterizing screen for positioning commands



Simple and transparent programming

Drive Control Chart (DCC) for closed-loop control and arithmetic functions

Overview

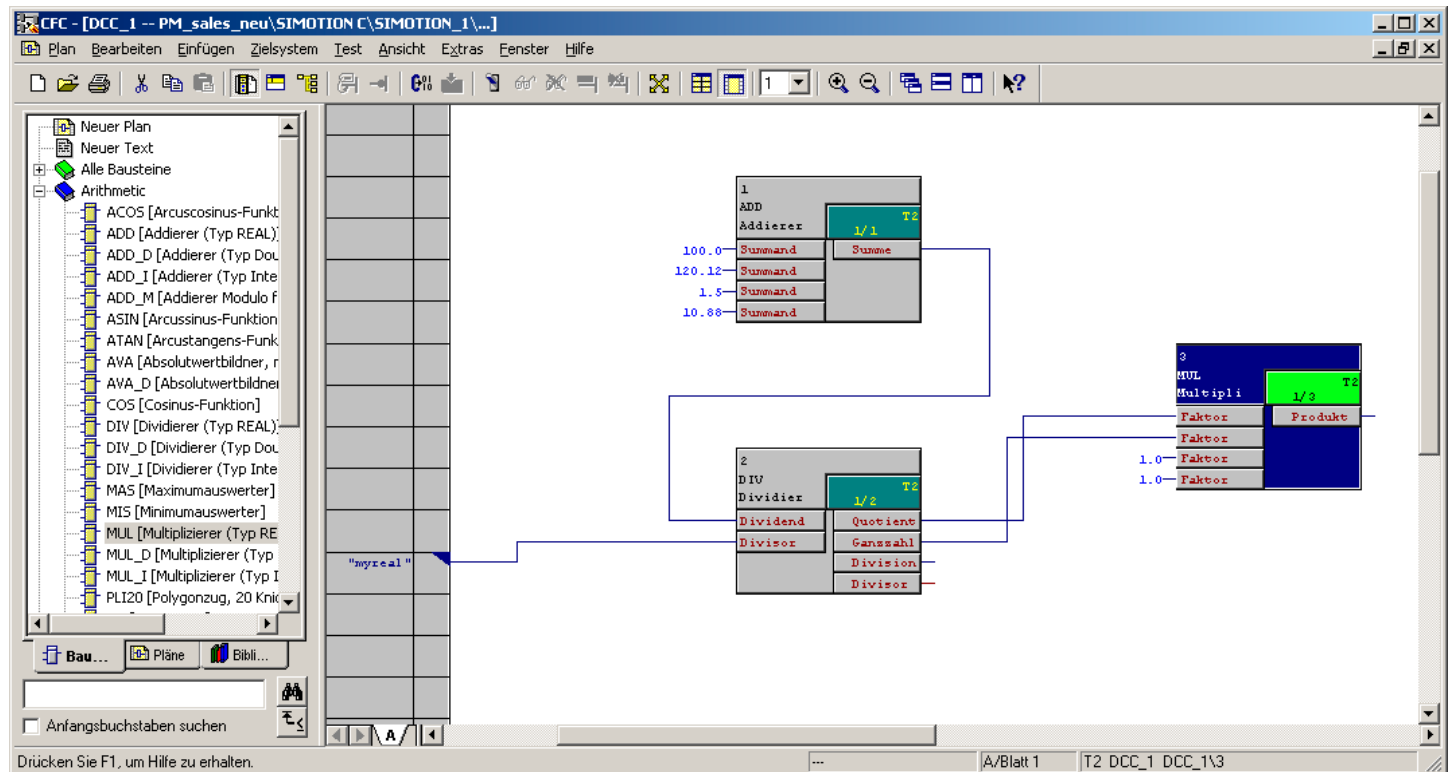
Task-oriented programming languages

Test support

Commissioning support

- DCC Editor based on CFC
- Especially suitable when programming closed-loop control functions

Programming with DCC



Program test

Overview

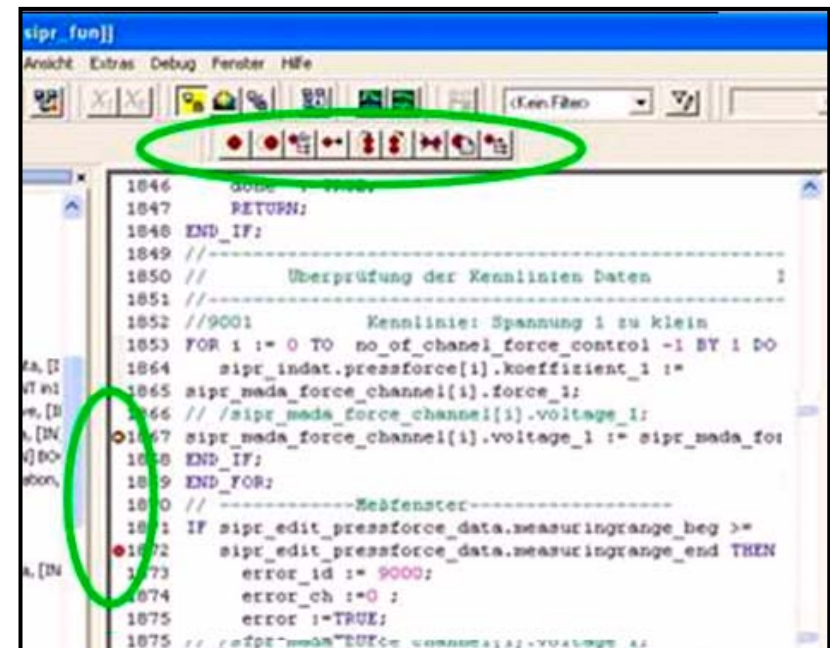
Task-oriented programming languages

Test support

Commissioning support

Program test:

- Debug functions with holding points
- Graphic step or signal tracing
- Single step mode



Time saving using specific and efficient tests

System information

Overview

Task-oriented
programming
languages

Test support

Commissioning
support

System information:

- Status display for system and user variables
- System utilization
- Task runtimes

Time saving using specific and efficient tests

Trace functions for the application test

Overview

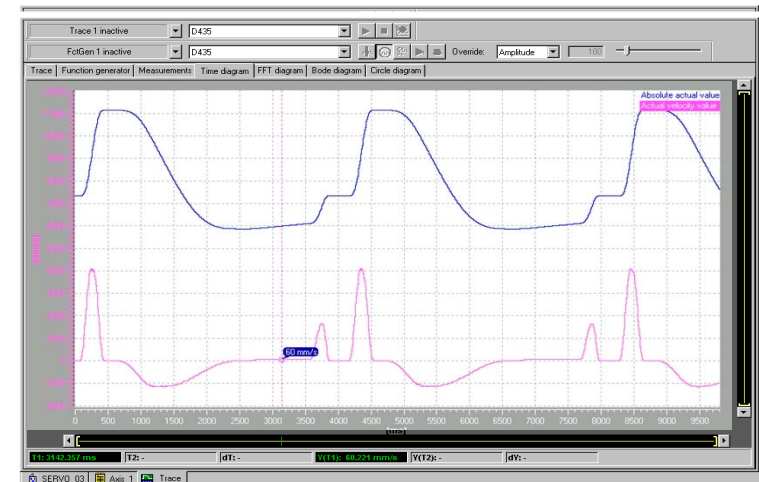
Task-oriented
programming
languages

Test support

Commissioning
support

Trace function for the application test:

- All variables are recorded in one trace (logic, motion control, technology, system, ...)
- Trigger, sample rate, time period can be freely set
- Displayed as:
 - list, with export, e.g. to Excel
 - diagram, where signals are superimposed for comparison purposes



Time saving using specific and efficient tests

Commissioning - support

Overview

Task-oriented programming languages

Test support

Commissioning support

Axis control panel:

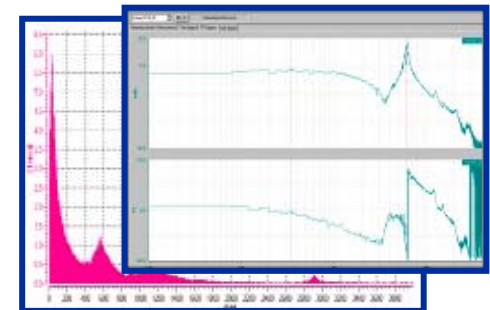
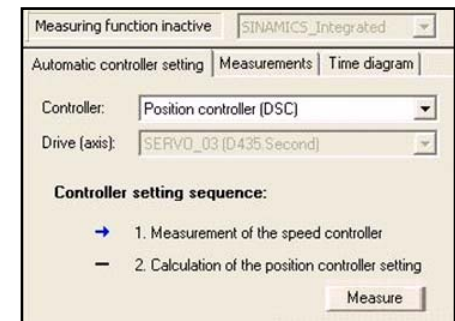
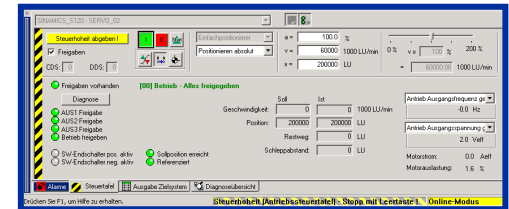
- To manually move axes and manually set and reset signals (input signals, enable signals, ...)

Automatic control optimization:

- For position controllers
- For speed controllers (only with Sinamics S120)
- Incl. automatic filter setting

Frequency response analysis:

- The function generator can be connected to any input quantity
- Any output quantities can be recorded/traced
- Evaluated as
 - **Bode diagram** or
 - **Frequency spectrum**



User-friendly machine commissioning and axis optimization

Axis control panel

Overview

Task-oriented programming languages

Test support

Commissioning support

Axis control panel:

- To manually move the axes and manually set and reset signals (input signals, enable signals, ...)

SINAMICS_S120 - SERVO_02

Steuerhoheit abgeben!

Freigaben

CDS: 0 DDS: 0

Einfachpositionierer

Positionieren absolut

a = 100.0 %

v = 60000 1000 LU/min

x = 200000 LU

0 % v x 100 % 200 %

= 60000.00 1000 LU/min

Freigaben vorhanden [00] Betrieb - Alles freigegeben

Diagnose

	Soll	Ist	
Geschwindigkeit:	0	0	1000 LU/min
Position:	200000	200000	LU
Restweg:	0		LU
Schleppabstand:	0		LU

SW-Endschalter pos. aktiv

SW-Endschalter neg. aktiv

Sollposition erreicht

Referenziert

Antrieb Ausgangsfrequenz ge

-0.0 Hz

Antrieb Ausgangsspannung c

2.0 Veff

Motorstrom: 0.0 Aeff

Motorauslastung: 1.6 %

Alarme

Steuertafel

Ausgabe Zielsystem

Diagnoseübersicht

Drücken Sie F1, um Hilfe zu erhalten.

Steuerhoheit [Antriebssteuertafel] - Stopp mit Leertaste!

Online-Modus

User-friendly machine commissioning and axis optimization

Automatic controller optimization

Overview

Task-oriented
programming
languages

Test support

Commissioning
support

Automatic controller optimization:

- For position controllers
- For speed controllers (only with Sinamics S120)
- Including automatic filter setting

The screenshot shows a software window titled "Automatic controller optimization". At the top, it says "Measuring function inactive" and "SINAMICS_Integrated" in a dropdown menu. Below this are three tabs: "Automatic controller setting" (selected), "Measurements", and "Time diagram". Under the "Automatic controller setting" tab, there are two dropdown menus: "Controller:" set to "Position controller (DSC)" and "Drive (axis):" set to "SERVO_03 (D435.Second)". Below these is a section titled "Controller setting sequence:" with two steps: "1. Measurement of the speed controller" (indicated by a blue arrow) and "2. Calculation of the position controller setting" (indicated by a minus sign). At the bottom right of the window is a "Measure" button.

User-friendly machine commissioning and axis optimization

Frequency response analysis

Overview

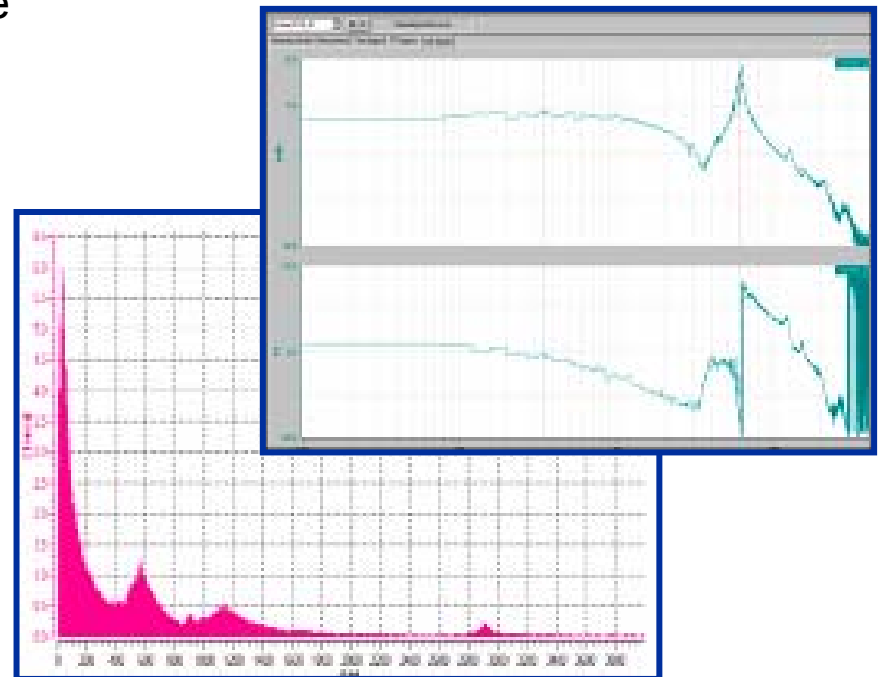
Task-oriented
programming
languages

Test support

Commissioning
support

Frequency response analysis:

- The function generator can be switched to any input quantity
- Any output quantities can be recorded/traced
- Evaluated as
 - **Bode diagram**, or
 - **Frequency spectrum**



User-friendly machine commissioning and axis optimization



SIEMENS



Many Thanks

Name: Michael Braun

Department: I DT MC PM SP

Telephone: +49 (9131) 98-2679

Mail: mb.braun@siemens.com