Using FOCAS Libraries in the TMA55OP Flexible Manufacturing Cell

Csokmai Lehel Szabolcs[1]

[1] University of Oradea, lcsokmai@uoradea.ro

Keywords: FOCAS, LAN, network.

Abstract: FOCAS libraries are used for exchanging data and information between a personal computer and the CNC/PMC through the HSSB or Ethernet. The FOCAS libraries are used with the Visual Studio Integrated Development Environment. The software was written in the C# programming language using the .NET Framework 2.0 .

1. Introduction

FOCAS libraries are used for exchanging data and information between a personal computer and the CNC/PMC through the HSSB or Ethernet. The FOCAS libraries are used with the Visual Studio Integrated Development Environment. The software was written in the C# programming language using the .NET Framework 2.0 .

The main features of the FOCAS libraries are:

- read/write data realted to axes/spindle, absolute positions, machine positions, remaining distance to go
- operations related to CNC program, upload/download, search, delete
- read/write CNC file data, tool offset data, workpiece reference point offset data, parameters
- read/write tool life data
- read history data, operation history and alarm history
- read/write data related to PMC

2. Presentation

Every application built with the help of FOCAS libraries have to establish the communication with the CNC. It si necessary to get the library handle number before calling any function. The library handle number can be obtained with the use of cnc_allclibhndl function. The following code is used to get the library handle number from the CNC:

```
Focas1.focas_ret ret = (Focas1.focas_ret)Focas1.cnc_allclibhndl(out cncHandle);
```

The cncHandle variable contains the library handle number. The type of the cncHandle variable is uShort. The ret variable contains the status message received from the CNC. If ret is EW_OK then the function is executed successfully, otherwise the following major error codes are returned:

- EW_HSSB, communication error
- EW_HANDLE, acquiring of handle is failed
- EW_BUSY, initialization between MMC function and CNC/PMC is in progress

Once the library handle number is aquired, it must be held until the termination of the application.

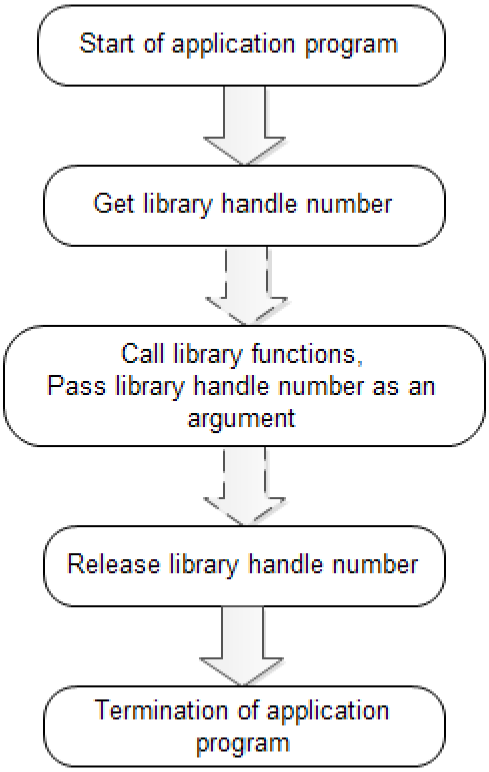
Start of application program
Get library handle number
Call library functions,
Pass library handle number as an argument
Release library handle number
Termination of application program

Figure 1. Processing sequence

When the application terminates, the library handle must be released by using the cnc_freelibhndl function.

```
if (cncHandle != 0) Focas1.cnc_freelibhndl(cncHandle);
```

Using the cnc_sysinfo, we can get various system informations such as:

- Maximum controlled axes
- Kind of CN
- Kind of M/T/TT
- Series number
- Version number
- Current controlled axes
-

```
Focas1.ODBSYS sysinfo_01 = new Focas1.ODBSYS();
Focas1.cnc_sysinfo(cncHandle, sysinfo_01);
string str_cnc_type = new string(sysinfo_01.cnc_type);
```

The cnc_rdposition function reads the positions from first axis to the specified axis number. The absolute, machine, relative and distance positions can be showed with the help of this function.

```
Focas1.ODBPOS pos = new Focas1.ODBPOS();
Focas1.cnc_rdposition(cncHandle, 0, ref 32, pos);
```

The pos variable contains the requested positions and can be used to show the following data:

- data, position data (*pos.p1.abs.data*)
- dec, decimal place (pos.p1.abs.dec)
- unit (pos.p1.abs.unit)
- name (*pos.p1.abs.name*)

The pmc_rdpmcrng function reads the PMC data of the specified PMC address/range. This function is used to exchange data between the application on MMC function and LADDER software on PMC.

```
Focas1.IODBPMC0 pmcData = new Focas1.IODBPMC0();
Focas1.focas_ret ret = (Focas1.focas_ret)Focas1.pmc_rdpmcrng(cncHandle, 5, 0, 16, 19,
12, pmcData);
```

The above functions will read byte data from the addresses R0016 - R0019.

- adr_type is relay (5)
- data_type is byte (0)
- s_number, start PMC address number
- e_number, end PMC address number
- length, 8 + addresses

The results are posted with the help of the following lines:

```
string str_0_bits = Convert.ToString(pmcData.cdata[0], 2);
```

This line converts the R0016 value binary. The maximum number which can be retunred is 255 (11111111).

3. Conclusions

With the help of the FOCAS libraries we can build custom applications for the TMA55OP flexible manufacturing cell. Adding a simple socket client and server modules to this application will make the application communicate over a simple ethernet (TCP/IP) network.

This work was partially supported by the strategic grant POSDRU/107/1.5/S/80272, inside POSDRU Romania 2007-2013 co-financed by the European Social Fund – Investing in People.

References:

1. Riccardo De Asmundis, Labview - Modeling, Programming and Simulations, 2011, ISBN: 9789533075211
2. Markus Feilner, Openvpn: Building And Integrating Virtual Private Networks, 2006, ISBN: 190481185X
3. Joseph Steinberg, SSL VPN : Understanding, evaluating and planning secure, web-based remote access, 2005, ISBN: 1904811078
4. FOCAS ½ documentation, A02B-0207-K737 Edition 03.8
5. Ganea, Macedon. Maşini unelte şi sisteme flexibile. Oradea : Editura Universitatii din Oradea, 2010. ISBN: 978-606-10-0020-3.
6. Ganea, Macedon.. Masini unelte flexibile si echipamente tehnologice pentru prelucrarea pieselor prismatice. Oradea : Editura Universitatii din Oradea, 2009. ISBN:978-973-759-884
7. Ganea, C. Particularitati tehnologice privind prelucrarea pieselor . Universitatea Tehnica Cluj-Napoca : s.n., 1998.
8. Andrea, Matta. Design of Advanced Manufacturing Systems. Torino : Springer, 2005. ISBN-10 1-4020-2930-6 200.
9. Altintas, Yusuf. Manufacturing Automation. s.l. : Cambridge University Press, 2000. ISBM:0-521-65973-6.
10. Barabas T., Vesselenyi T. Maşini unelte şi agregate. Oradea : Editura Universitatii din Oradea, 1997.
11. Abrudan, I. Sisteme flexibile de fabricatie. Concepte de proiectare si management. Cluj-Napoca: Editura Dacia, 1996.