FANUC PROGRAM NAME DECODING

This file describes some of the Fanuc Focas functions used to fetch the program O# and the contents of the file. The actual program name (or part id) is different than the Fanuc convention of O numbers (i.e., O#). The program name can emebedded in a comment be at the top or after the O# in a program. This file contains the code to attempt to find this program name. The basic program logic if as follows:

1. Read program O# save into prognum
2. Attempt to find program name or part id. Search code for comment (1st line) or line after O# to extract program name. Save progname.
3. Write to MTConnect. If progname exists, write it. If not, write O# prognum. If none, write O99.

# Fanac Focas Funtions

## FWLIBAPI short WINAPI cnc\_rdseqnum(unsigned short FlibHndl, ODBSEQ \*seqnum);

**Description** Reads the sequence number of the NC program which is being currently executed in CNC. If the NC program has no sequence numbers in its all blocks, the sequence number of the last executed block is read. This function is used for watch the block being executed or the current process by the application program, or only displaying the current sequence number. The sequence number is stored in "buf.data" with unsigned binary format.

## FWLIBAPI short WINAPI cnc\_rdprgnum(unsigned short FlibHndl, ODBPRO \*prgnum);

**Description** Reads program number of the program which is being currently selected in CNC. As for Series 16/18/21/0, Power Mate i, if CNC exetues the sub-program, this function can read also the main program number. In this case, that main program number is the root program number which was selected on the CNC for the execution. If the program being executed is not a sub-program, the same program number is set to both 'Running program number' and 'Main program number'. This function is used for management of NC programs in CNC by the application program, etc. The program numbers are stored in "buf.data" and 'buf.mdata" with unsigned binary format. It is possible to use this function for the program number 8 digits, however it is necessary to switch API to the one for the program number 8 digits. In series 15, it is necessary to switch API to the one for the program number 8 digits. See "[Program number 8 digits](file:///C:\Program%20Files\NIST\proj\MTConnect\Documentation\Fanuc\GeFanuc%20FOcas1\GeFanuc%20FOcas1\general.htm#O8)" for details.

# MTConnect Adapter code using Focas to read program information

## Reading Program O# or Program Name

The code below uses the Fanuc Focas library function cnc\_rdseqnum to retrieve the current block number. This is saved into MTconect agent, but unused .

Next, the Fanuc conventional way to signify the program is by an O#. Without an O# the program cannot be loaded. However, the O# is limited in that it only provides a number and not a part identification. Thus, there is code to read the leading comments, if an CONFIGURATION FLAG is set (FirstComment) This

int CiSeries::getLine()

{

std::string prognum,progname;

short ret;

if (!\_adapter->mConnected)

return -1;

ODBSEQ block;

ret = cnc\_rdseqnum(\_adapter->mFlibhndl, &block ); // 15,16,18,21,0,powermate

if (ret == EW\_OK)

{

\_adapter->SetMTCTagValue("line", StdStringFormat("%d", block.data));

}

else

{

// Note GLogger does timestamp error logging into file debug.txt!

GLogger.Info(StdStringFormat("Error iSeries::cnc\_rdseqnum=%d\n", ret));

}

ODBPRO buf ;

ret = cnc\_rdprgnum( \_adapter->mFlibhndl, &buf ) ; // 15,16,18,21,0,powermate

if (ret == EW\_OK)

{

// mdata - Main program number. data- Running program number.

prognum=StdStringFormat("O%04hd", buf.data); }

else

{

LogErrorMessage(" CiSeries::cnc\_rdprgnum FAILED\n", ret );

}

// Now try to extract program name from comments. Read 1st lines of file,

// look for “(“, if found, extract comment then program name

if(Globals.ProgramLogic == "FirstComment")

{

PRGDIR2 prg[BUFSIZE+1];

short i, num;

short top = 0;

std::string sLine;

do {

num = BUFSIZE;

ret = cnc\_rdprogdir2( \_adapter->mFlibhndl, 1, &top, &num, prg );

if ( ret == EW\_NUMBER )

break;

if ( ret )

{

LogErrorMessage("iSeries::cnc\_rdprogdir2 ERROR\n", ret );

break;

}

for ( i = 0 ; i < num ; i++ )

{

sLine = prg[i].comment;

int n = sLine.find("(");

int m= sLine.find(")");

if(std::string::npos !=n && std::string::npos !=m)

sLine= sLine.substr(n+1,m-n-1);

if(!sLine.empty() && sLine != "()" )

progname=sLine;

if(!sLine.empty())

break;

}

top = prg[num-1].number + 1;

if(!sLine.empty())

break;

} while ( num >= BUFSIZE );

}

if(!progname.empty())

{

\_adapter->SetMTCTagValue("program", progname);

}

else if(!prognum.empty())

{

\_adapter->SetMTCTagValue("program", prognum);

}

else

{

\_adapter->SetMTCTagValue("program", "O1");

}

return EW\_OK;

}

Below is historic code that looked for a comment on the line after the O#. If found extracted the comment as the program name. NO LONGER USED.

std::string FanucShdrAdapter::getProgramName(char \* buf)

{

GLogger.LogMessage("FanucMTConnectAdapter::getProgramName Enter\n", 3);

int prognum;

std::string progname;

std::stringstream ss(buf);

std::string sLine;

while(std::getline(ss, sLine, '\n'))

{

// look for O # which must be there

if(1 != sscanf(sLine.c\_str(), "O%d", &prognum))

continue;

progname=StdStringFormat("O%d", prognum);

// if find comment inside (...) return as program name

int n = sLine.find("(");

int m= sLine.find(")");

if(std::string::npos !=n && std::string::npos !=m)

return sLine.substr(n+1,m-n-1);

// else could be standalone on next line

if(std::getline(ss, sLine, '\n'))

{

n = sLine.find("(");

m= sLine.find(")");

if(std::string::npos !=n && std::string::npos !=m)

return sLine.substr(n+1,m-n-1);

}

return progname;

}

GLogger.LogMessage("FanucMTConnectAdapter::getProgramName Leave\n", 3);

return "";

}