Building EPICS IOCs for Eurotherm 3504/3508 with ModBus TCP

Ji Li

This document describes how to build EPICS IOCs for Eurotherm 3504/3508.

# Prerequisite

The IOC relies on ModBus. It uses absolute address to communicate with Eurotherm, which is supported in ModBus R2-9 or later. The source code of ModBus is located at:

<https://github.com/epics-modules/modbus>

# Get source code of the IOC

The IOC was developed at Diamond light source, and the source code is located at:

<https://github.com/dls-controls/eurothermModbus>

# Create EurothermModbus support

Run command:

makeBaseApp -t support EurothermModbus

Copy the following files in the git repository to the corresponding directories:

* configure/RELEASE
* eurothermModbusApp/Db/\*
* eurothermModbusApp/src/Makefile

Make necessary changes to the environment variables.

# Create IOC

Run commands to create a basic IOC:

makeBaseApp -t ioc eurotherm3k

makeBaseApp -t ioc -i -a linux-x86\_64 eurotherm3k

Copy the following files from the Git repository to the corresponding directories:

* iocs/examples/RELEASE
* iocs/examples/exampleApp/Db/\*
* iocs/examples/exampleApp/src/Makefile

Make necessary changes to the environment variables. Make necessary changes to the Makefile(s) according to the IOC name used (eurotherm3k for this example).

Edit st.cmd with the following content:

|  |
| --- |
| #!../../bin/linux-x86\_64/eurotherm3k  ## You may have to change eurotherm3k to something else  ## everywhere it appears in this file  < envPaths  epicsEnvSet("ENGINEER", "Ji Li x7636")  epicsEnvSet("LOCATION", "28ID1")  epicsEnvSet("EPICS\_CA\_AUTO\_ADDR\_LIST", "NO")  epicsEnvSet("EPICS\_CA\_ADDR\_LIST", "10.28.32.255")  cd ${TOP}  #=============================  ## Register all support components  dbLoadDatabase("dbd/eurotherm3k.dbd",0,0)  eurotherm3k\_registerRecordDeviceDriver(pdbbase)  #=============================  # Params:  # . portName  # . hostInfo  # . priority  # . noAutoConnect  # . noProcessEos  drvAsynIPPortConfigure( "EUROTHERM", "192.168.111.222:502", 100, 0, 1 )  #=============================  # Params:  # . portName  # . linkType  # . timeoutMsec  # . writeDelayMsec  modbusInterposeConfig("EUROTHERM", 0, 2000, 0 )  #=============================  # Params:  # . portName  # . tcpPortName  # . slaveAddr  # . modbusFunction - 3 (read)  # . modbusStartAddr - -1 (absolute addressing)  # . modbusLength - (in 16-bit words)  # . modbusDataType - 0 (UINT16)  # . pollMsec  # . plcType  drvModbusAsynConfigure("EURTHM\_MB\_1\_MASTER\_RX", "EUROTHERM", 1, 3, -1, 1, 0, 1000, "" )  #=============================  # Params:  # . portName  # . tcpPortName  # . slaveAddr  # . modbusFunction - 6 (write)  # . modbusStartAddr - -1 (absolute addressing)  # . modbusLength - (in 16-bit words)  # . modbusDataType - 0 (UINT16)  # . pollMsec  # . plcType  drvModbusAsynConfigure("EURTHM\_MB\_1\_MASTER\_TX", "EUROTHERM", 1, 6, -1, 1, 0, 0, "" )  #=============================  asynSetTraceMask("EUROTHERM",-1,0x09)  asynSetTraceIOMask("EUROTHERM",-1,0x4)  asynSetOption("EUROTHERM", 0, "disconnectOnReadTimeout", "Y")  #=============================  ## Load record instances  dbLoadRecords("db/eurotherm3k\_expanded.db","user=softioc")  #=============================  iocInit()  dbpf XF:28ID1-ES:1{Env:03}LOOP1:RR.DRVH 0.2  dbpf XF:28ID1-ES:1{Env:03}LOOP2:RR.DRVH 0.2 |

# Reference

* Driver Support for Modbus Protocol under EPICS, <https://cars9.uchicago.edu/software/epics/modbusDoc.html>
* Exception Responses, <http://www.simplymodbus.ca/exceptions.htm>