Use of Python as Replacement Magnet Power Supply Remote Interface

Brian Eng, Mary Ann Antonioli, Peter Bonneau, Aaron Brown, Pablo Campero, George Jacobs, Mindy Leffel,
Tyler Lemon, Marc McMullen, and Amrit Yegneswaran
Physics Division, Thomas Jefferson National Accelerator Facility, Newport News, VA 23606
March 11, 2024

This note presents the reason for replacing the magnet power supply's (MPS) remote interface with a Python program and the subsequent issue found.

The MPS for the Hall A Moller experiment uses a fixed port to respond to ASCII commands over TCP/IP. During normal operations, the response to the ASCII commands is either done when the MPS has its main AC power (480 VAC 3-phase) applied or when control power (120 VAC 1-phase) is enabled and supplied.

After the site acceptance test was completed, the input AC power was removed to prevent powering the test coil; to enable control power requires manually moving a switch on the MPS, along with supplying the 120 VAC from a wall outlet—a procedure not conducive to quick changes, as features are being added, to the Programmable Logic Controller (PLC) code that oversees control and monitoring.

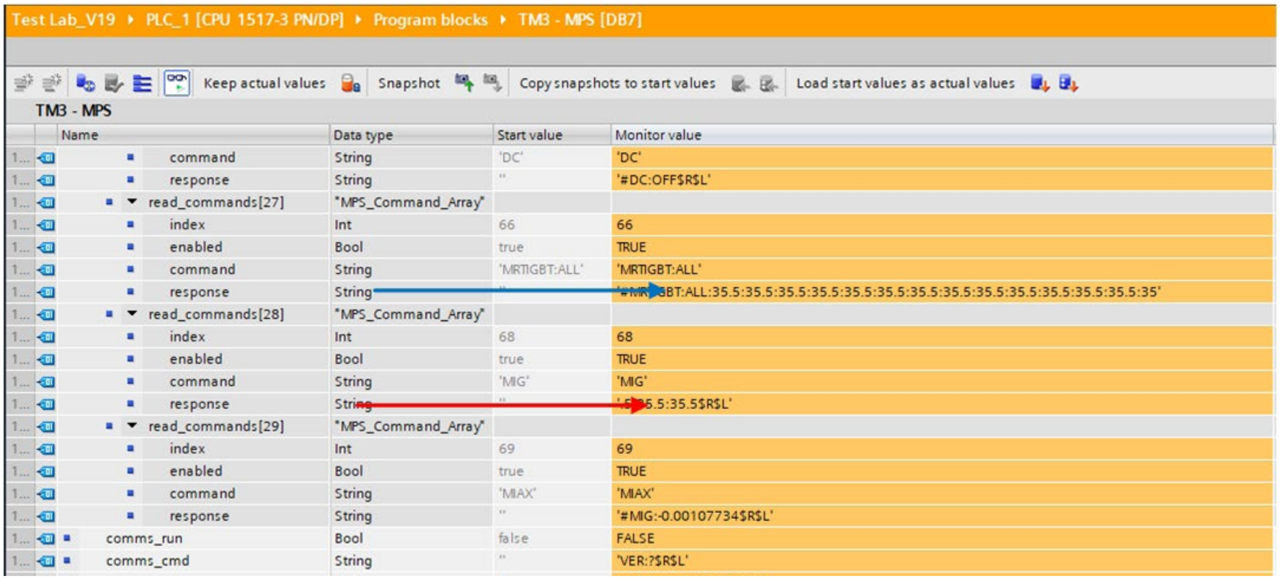
To resolve the issue, a Python program was written to act as the TCP/IP server in place of the MPS unit's server. The PLC modification required was to change the IP address from that of the actual MPS to the IP address of the computer running the Python code.

Currently, the Python code responds to the PLC command queries with fixed responses based on the MPS being in an off-and-non-faulted state. However, it is trivial to modify the responses given to the PLC.

It was found that running the PLC without any delays results in ~10 ms between commands, significantly faster than will be needed in the final system.

An issue found was that in using the command MRTIGBT:ALL, which returns a list of all insulated-gate bipolar transistor (IGBT) temperatures from the four power modules (each containing three IGBTs), along with the maximum temperature of each power module—16 temperatures in all—the PLC was splitting the response command between successive calls, Fig 1. This issue is currently being investigated. As the first response appears to be limited to 80 characters, it is speculated that a length is not being updated correctly or a character array is not the correct size, as not all PLC functions can use strings and require character arrays instead.

The Python replacement for the MPS remote interface has allowed for quick iteration on the PLC code while additional capabilities are added, without the need to physically access the MPS.


Test Lab_V19 ▸ PLC_1 [CPU 1517-3 PN/DP] ▸ Program blocks ▸ TM3 - MPS [DB7]
Keep actual values
Snapshot
Copy snapshots to start values
Load start values as actual values
TM3 - MPS

Name	Data type	Start value	Monitor value
command	String	'DC'	'DC'
response	String	''	'#DC:OFFRL'
▾ read_commands[27]	"MPS_Command_Array"		
index	Int	66	66
enabled	Bool	true	TRUE
command	String	'MRTIGBT:ALL'	'MRTIGBT:ALL'
response	String	''	'#MRTIGBT:ALL:35.5:35.5:35.5:35.5:35.5:35.5:35.5:35.5:35.5:35.5:35.5:35.5:35.5:35.5:35'
▾ read_commands[28]	"MPS_Command_Array"		
index	Int	68	68
enabled	Bool	true	TRUE
command	String	'MIG'	'MIG'
response	String	''	'.5:35.5:35.5RL'
▾ read_commands[29]	"MPS_Command_Array"		
index	Int	69	69
enabled	Bool	true	TRUE
command	String	'MIAX'	'MIAX'
response	String	''	'#MIG:-0.00107734RL'
comms_run	Bool	false	FALSE
comms_cmd	String	''	'VER:?RL'

FIG. 1. MRTIGBT:ALL response split among two commands, blue and red arrows.