The standardization of device driver's implementation in distributed control systems is essential to reduce the development and integration effort. Nowadays, there are multiple distributed control systems and high-level applications used in the scientific community. Due to the lack of standardization in the interfaces with the device drivers, it is required to integrate each hardware device separately in each control system. However, a clear model of how different software applications should manage the device drivers could reduce development efforts and ease the maintainability of such complex systems. The Nominal Device Support initiative is an open source project, for the standardization of data acquisition and timing devices. The first implementations of NDS (version 1.x and 2.x) were oriented to simplify the implementation of EPICS devices supports. NDS v3 is more ambitious and is built as an abstraction layer for any control system or software application.

**SUMMARY**

**MAIN FEATURES OF NDS-CORE LAYER**

- Software layer implemented in C++ in charge of interfacing with the hardware using the device driver provided by the manufacturer.
- The driver manages the functionality using a hierarchical organization in nodes, called NDS-nodes. Each NDS-node manages an specific hardware block (i.e: firmware, ADC, DAC. Timing, clocking, triggering, timestamping, future timing event, etc.)
- An NDS-node provides a list of defined process variables in charge of the configuration and management of the specific hardware functionality.
- The NDS PVs are classified as Variablen/Out and Delegaten/Out. NDS PVs implement function to read and write its specific values. Additionally, DelegatePVs trigger the execution of specific methods when are written or read.
- NDS-core provides method to subscribe and replicate PVs. This option allows to share data among different NDS Device Drivers.
- An NDS device driver is implemented using NDS-core and is available for the user as a library (shared or static) to be used in any application.
- NDS-core layer has been tested using google test. A simplified test control system is provided with the test code.

**ACKNOWLEDGEMENTS**

This work was supported in part by the Spanish Ministry of Economy and Competitiveness, Projects Nº ENE2015-64914-C3-3-R and Madrid regional government (YEFI fund), Grant Nº PEID-2018-RE/IFIC-8571.

**RESULTS**

- Integrated in ITER CODAC Core System 6.0 (NDSv 3.2.0.0)
- Drivers available for PXIe Devices: PXI6683 (timing card), PXIe3636 (X-Series DAQ Devices) and FlexRIO with analog input module N15761
- Drivers available for MTCA devices: PTM1588 (timing card)
- Version with new functionalities in Q4 2019 (v3.3.0.0)
- Support for ITER Data Archiving Network and Synchronous Data Network