25th IEEE Real Time Conference
May 25-29, 2026

A common framework for real-time structured data communication in fusion devices

N. Ferron[1], C. Galperti[2], G. Manduchi[1,3], A. Neto[4], A. Segovia[2], F. Felici[2]

[1] *Consorzio RFX (CNR, ENEA, INFN, Università di Padova, Acciaierie Venete SpA), 35127 Padova, Italy*
[2] *EPFL-SPC, 1015 Lausanne, Switzerland*
[3] *CNR-ISTP, 35127 Padova, Italy*
[4] *Fusion for Energy, 08019 Barcelona, Spain*

Goal of this work

A modular and flexible set of tools to handle the **exchange of structured data** in a popular framework configuration in **fusion devices**, composed by MARTe2, MDSplus, and MATLAB/Simulink.

```
static DataTypeMap rtDataTypeMap[] = {
  /* cName, mwName, numElements, elemMapIndex, dataSize, slDataId, isComplex, isPointer */
  { "unsigned short", "uint16_T", 0, 0, sizeof(uint16_T), uint8_T, 0, 0, 0 },
  { "double", "real_T", 0, 0, sizeof(real_T), uint8_T, 0, 0, 0 },
  { "struct", "Lambda_T", 2, 1, sizeof(Lambda_T), uint8_T, 0, 0, 0 }};
```

A popular fusion PCS

MARTe2

widely used in fusion research and other time-critical domains [2, 4]

1. high-performance, real-time execution framework
2. deterministic control of tasks in demanding experimental settings
3. component-based architecture for modular development and high configurability
4. portable

MDSplus

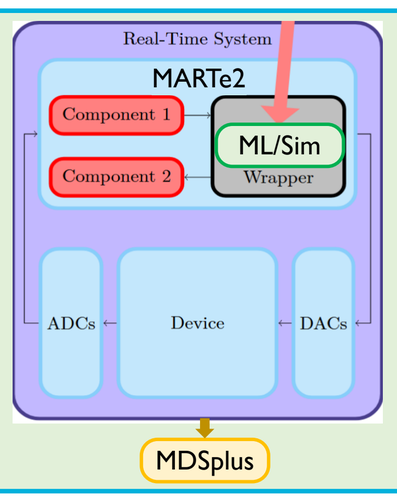
robust data management system tailored for scientific experiments [5]

1. hierarchical data storage
2. data versioning
3. distributed data access
4. event-driven action dispatching functionalities
5. rich tool ecosystem

MATLAB/Simulink®

powerful modelling and simulation layer to prototype control algorithms and deploy them into real-time systems [3, 6]

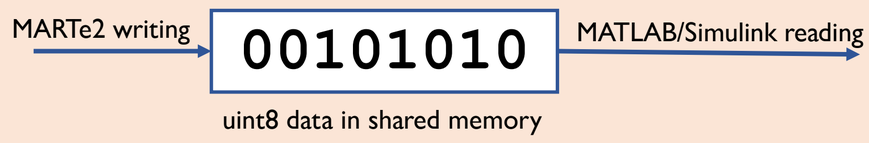
1. extensive and well-maintained libraries
2. intuitive graphical modelling environment
3. advanced code generation capabilities
4. *de-facto* standard in fusion science



Structured signals and parameters

Communication between tools is managed via:

1. **signals**: array of multiple timestamped data whose value can change each real-time loop
2. **parameters**: an immutable value that is determined at the real-time system bootstrap



Organization of such data is not trivial (MARTe2 configuration file):

```
+IOGAM_MDSReader2 = {
  Class = IOGAM
  InputSignals = {
   // ================== triaxial pickups (poloidal) set 2 ====================
   BP_1102      = { Type = float64 NumberOfElements = 1 DataSource = MDSReader2 }
   BP_1102_Qual = { Type = uint8   NumberOfElements = 1 DataSource = MDSReader2 }
   BP_1106      = { Type = float64 NumberOfElements = 1 DataSource = MDSReader2 }
   BP_1106_Qual = { Type = uint8   NumberOfElements = 1 DataSource = MDSReader2 }
   BP_1110      = { Type = float64 NumberOfElements = 1 DataSource = MDSReader2 }
   BP_1110_Qual = { Type = uint8   NumberOfElements = 1 DataSource = MDSReader2 }
   BP_1114      = { Type = float64 NumberOfElements = 1 DataSource = MDSReader2 }
   BP_1114_Qual = { Type = uint8   NumberOfElements = 1 DataSource = MDSReader2 }
   BP_1118      = { Type = float64 NumberOfElements = 1 DataSource = MDSReader2 }
   BP_1118_Qual = { Type = uint8   NumberOfElements = 1 DataSource = MDSReader2 }
   BP_1122      = { Type = float64 NumberOfElements = 1 DataSource = MDSReader2 }
   BP_1122_Qual = { Type = uint8   NumberOfElements = 1 DataSource = MDSReader2 }
   BP_1126      = { Type = float64 NumberOfElements = 1 DataSource = MDSReader2 }
   BP_1126_Qual = { Type = uint8   NumberOfElements = 1 DataSource = MDSReader2 }
   // ... to be continued (this configuration files is 2600 lines long)
```

The same file with structured data is much tidier:

```
+Types = {
  Class = ReferenceContainer
  +Triaxi_Signals_t = {
    Class = IntrospectionStructure
    BP_Signal = { NumberOfElements = {1} Type = float64 }
    BP_Qual   = { NumberOfElements = {1} Type = uint8   }
  }
}
+IOGAM_MDSReader2 = {
  Class = IOGAM
  InputSignals = {
    // ==================== triaxial pickups (poloidal) set 2 ======================
    BP_Signals_Struct = { Type = Triax_Signals_t NumberOfElements = 72 DataSource = MDSReader2 }
```

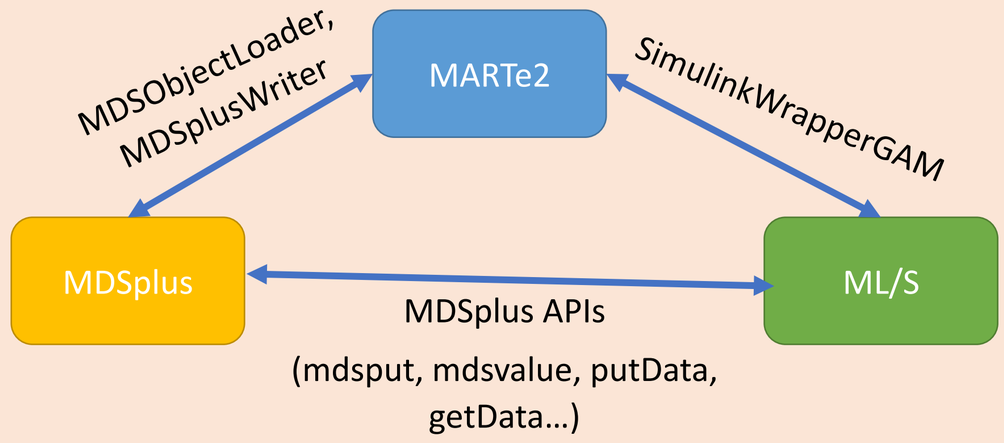
With **structures**, the experimental data:

1. remains tidy, easy and fast to consult
2. is less error-prone
3. implicitly carries additional information about each dataset

MARTe2, MDSplus and MATLAB/Simulink organize structures in different ways:

1. **MARTe2** stores structures in a flattened fashion
2. **MDSplus** stores structures in an internal binary format
3. **MATLAB/Simulink** generates standard C++ structures

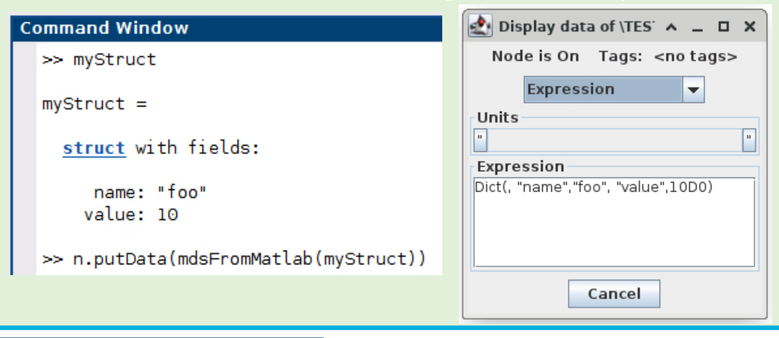
We propose a modular and flexible set of tools to provide structured data communication in a popular PCS configuration in fusion devices:



MDSplus to/from MATLAB

MDSplus offers extensive APIs to read and write structured data in its internal binary format.

1. MATLAB structures are mapped to native MDSplus structured types (Dictionary, APD)
2. MDSplus APIs are updated, both procedural and object-oriented ones, to handle MATLAB structured objects (natively Java objects)
3. SCDDS (deployment framework at TCV [1]) is updated to use structures using new MDSplus APIs



MARTe2 to/from MATLAB

MARTe2 stores structures in a flattened fashion, retaining only an internal map of how the structure should be organized.

1. Structured parameters can be read using the new MDSObjectLoader component [7], that reads flat and structured parameters from MDSplus, text files and MATLAB files
2. The MARTe2 SimulinkGAM (Generic Application Module) is now able to exchange structured data in real-time with MATLAB/Simulink generated code, converting native C++ structures to MARTe2 structured on-the-fly

```
+Loader1 = {
    Class = ObjectLoader
    Shot = -1
    Tree = scddstest
    Server = pluto
    ClientType = Distributed
    +CDBConnection1 = {
        Class = ConfigurationDatabaseConnection
        // ...
    }
    +Connection1 = {
        Class = MDSObjectConnection
        Parameters = {
            matrix = { Path = "\\SCDDSTEST::TOP:GAIN1" DataOrientation = "RowMajor" }
            structMixed = {
                error = (uint32) 1
                mat = { Path = "LONG_UNSIGNED(\\SCDDSTEST::TOP:MATRET)" DataOrientation = "RowMajor" }
                structParamMatrix = { Path = "\SCDDSTEST::TOP:STRUCTMATRIX" }
            }
            myStruct = {
                param1 = { Path = "\\SCDDSTEST::TOP:GAIN1" }
                param3 = { Path = "\SCDDSTEST::TOP:STRUCT1" }
                param4 = { Path = "\SCDDSTEST::TOP:STRUCTARRAY" }
                param6 = { Path = "\SCDDSTEST::TOP:PROVE2" DataOrientation = "RowMajor" }            }
        }
    }
}
```

MARTe2 to/from MDSplus

MARTe2 has been provided with components to instantiate structured data in its startup configuration file, to read and write structured parameters and signals to and from MDSplus.

1. The MARTe2 MDSplusWriter component is updated to handle structured data, and is now able to write real-time signals directly from MARTe2 in the MDSplus database [7]
2. Structured parameters can be read using the new MDSObjectLoader component, that reads flat and structured parameters from MDSplus, text files and MATLAB files

Conclusions

This work:

1. bridges the architectural gaps between MARTe2, MDSplus, and MATLAB/Simulink
2. establishes a unified framework for structured data exchange
3. enhances data integrity and metadata retention without sacrificing real-time performance
4. provides a robust, scalable foundation essential for handling the increasing data complexity of next-generation fusion devices.

References

1. C. Galperti et al., Overview of the TCV digital real-time plasma control system and its applications *Fus. Eng. Des.* **2024** 208 114640
2. N. Ferron et al., Implementation and Review of the Axisymmetric Equilibrium System of RFX-mod2 within the MARTe2 Framework *Electronics* **2022** 11 2751
3. S. Dubbioso et al., Hardware-in-the-loop validation of an Extremum Seeking-based system for vertical stabilization of tokamak plasmas *IFAC-PapersOnLine* **2025** 59
4. D. Alves et al., A New Generation of Real-Time Systems in the JET *Tokamak IEEE Transactions on Nuclear Science* **2014** 61
5. G. Manduchi et al., Control and Data Acquisition System upgrade in RFX-mod2 *Fus. Eng. Des.* **2024** 112329
6. L. Piron et al., Error Field Detection and Correction Studies Towards ITER Operation *Nucl. Fusion* **2024** 64 066029
7. A. Neto et al., MARTe2, https://vcis-gitlab.f4e.europa.eu/aneto/MARTe2