IEEE 1451 HTTP Server Implementation for Marine Data

Joaquín del Río[1], Marc Martínez[1], Daniel Mihai Toma[1], Antoni Mànuel[1]

Helena G. Ramos[2]

[1]SARTI Research Group. Universitat Politècnica de Catalunya. Rambla Exposició 24, 08800, Vilanova i la Geltrú. Barcelona. Spain. +(34) 938 967 200. Joaquin.del.rio@upc.edu

[2]Instituto de Telecomunicaçoes, DEEC. Instituto Superior Tecnico, UTL. Lisboa, Portugal,

Abstract- *Accessing real time data from a marine sensor network (MSN) can be a challenge. Open access to real time data using interoperable internet technologies is one of the major demands. The IEEE 1451 Smart Transducer Interface Standards [1,2,7] specify a standard API (application processor interface) process to discover and access sensor data by using an HTTP connection. This paper presents an HTTP Server for Marine Data coming from OBSEA[3,5] cabled Observatory at the Western Mediterranean Sea. The implementation was using LabVIEW Web Services.*

Keywords- *IEEE1451, LabVIEW Web Services, Marine Sensor Network*

I. INTRODUCTION

Data Interoperability is one of the major issues nowadays, where internet technologies enable the access to sensors data in real time. Sensor Web Enable (SWE) [4] from OGC (Open Geospatial Consortium) and IEEE1451 Standard are two approaches to achieve these interoperability problems. Discover and accessing available sensor data from an Internet web server trough a standard is one objectives of this work. In this case, an HTTP 1451.0 Server has been implemented using LabVIEW Web Services. Previous implementations of a HTTP 1451.0 server were been done using different technologies or language programming like JAVA. In this case, LabVIEW graphical programming was used to facilitate the integration process for electronics or scientist who most times are more familiar with LabVIEW than other languages.

II. IMPLEMENTATION

The IEEE 1451 family of standards were created to address the standard digital documenting of sensors [1,2]. The main objective is to add a digital layer of memory, functionality and communication to sensors so as to make sensors controllable and measurements accessible through a network with sufficient information on the sensor characteristics and history.

Figure 1 depictures the main components of a full compliance IEEE1451 system: The Transducer Interface Module (TIM), a device that could be either a sensor or an actuator with transducer channels and a Network Capable Applications Processors (NCAP) where different instruments or transducers are connected through a TIM (Transducer Interface Module). Each NCAP inside the system will have a NCAP address and each TIM connected to NCAP his TIM addresses.

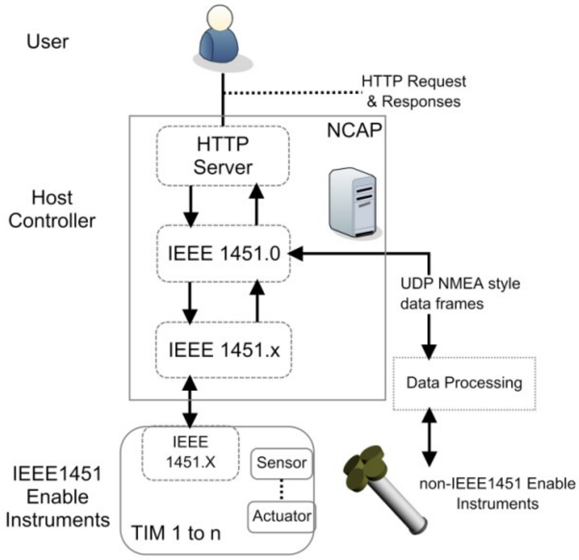


Figure 1. HTTP access to an IEEE 1451.0 NCAP and Non Standard Instruments Interface

For a non IEEE1451 compliance instruments, a data processing block implement the instrument manufacture-dependent protocol and translate the instrument data to a UDP NMEA-0183 style data frame that is sent to the IEEE1451.0 NCAP who interpret this data coming from a virtual TIM with virtual channels.

As an example, a non IEEE1451 instrument likes a marine CTD who measures current, temperature and depth will be interpreted by our NCAP as a virtual TIM with three different channels.

As is specified in the Std. 1451.0 2007 [7] the IEEE 1451.0 HTTP APIs shown at Table 1 focuses mainly on accessing transducer data and TEDS using the HTTP 1.1 protocol. This API contains operations to read and write TransducerChannels, read and write TEDS, and send configuration, control, and operation commands to the TIMs. The API contains five interfaces: TransducerAccess, TransducerManager, TimDiscovery, TEDSManager, and AppCallback. The first four interfaces are implemented by IEEE 1451.0 and are called by the measurement applications. If the application desires advanced optional features, it will need to implement the "AppCallback" interface, which IEEE 1451.0 will invoke.

Table 1. HTTP API

Interface	commands	path
Discovery API	TIMDiscovery	1451/Discovery/TIMDiscovery
	TransducerDiscovery	1451/Discovery/TransducerDiscovery
Transducer Access API	ReadData	1451/TransducerAccess/ReadData
	StartReadData	1451/TransducerAccess/StartReadData
	MeasurementUpdate	1451/TransducerAccess/MeasurementUpdate
	WriteData	1451/TransducerAccess/WriteData
	StartWriteData	1451/TransducerAccess/StartWriteData
TEDS Manager API	ReadTeds	1451/TEDSManager/ReadTeds
	ReadRawTeds	1451/TEDSManager/ReadRawTeds
	WriteTeds	1451/TEDSManager/WriteTeds
	WriteRawTeds	1451/TEDSManager/WriteRawTeds
	UpdateTedsCache	1451/TEDSManager/UpdateTedsCache
Transducer Manager API	SendCommand	1451/TransducerManager/SendCommand
	StartCommand	1451/TransducerManager/StartCommand
	CommandComplete	1451/TransducerManager/CommandComplete
	Trigger	1451/TransducerManager/Trigger
	StartTrigger	1451/TransducerManager/StartTrigger

Some of the API commands implemented by the LabVIEW Web Service are shown in Table 2

Table 2. Implemented Commands by the LabVIEW HTTP 1451 web Service

•TimDiscovery (TimDiscoveryHTTPRequest, TimDiscoveryHTTPResponse)
•TransducerDiscovery (TransducerDiscoveryHTTPRequest, TransducerDiscoveryHTTPResponse)
•ReadTransducerData (ReadTransducerDataHTTPRequest, ReadTransducerDataHTTPResponse)
•ReadMetaIDTEDS (ReadMetaIDTEDSHTTPRequest, ReadMetaIDTEDSHTTPResponse)
•ReadTIMGeoLocationTEDS(ReadTIMGeoLocationTEDSHTTPRequest, ReadTIMGeoLocationTEDSHTTPResponse)

A remote client use the "http" protocol to send messages to the IEEE 1451.0 NCAP, which acts like a server, serving transducer data to the remote client. The HTTP message from the remote client unit will be transmitted across a network connecting the remote client with the IEEE 1451.X transducer nodes, or in this case to the non IEEE1451 instruments. The message will adhere to the HTTP URL syntax (RFC 2616) as http://<host>:<port>/<path>?<parameters>.

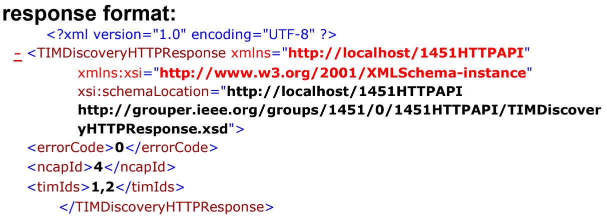
For example, an http request to our 1451 server could be:

http://esonet.epsevg.upc.es:1451/1451/Discovery/TIMDiscovery?ncapId=4&responseFormat=xml

Where it specify a NCAP server hosted at esonet.epsevg.upc.edu at port 1451. The LabVIEW web service path for the discovery messages is /1451/Discovery and in this example a TIMDiscovery request to our NCAP is done asking for an xml formatted reply.

The LabVIEW IEEE1451 HTTP web service reply to the previous command will be:

XML response format:



```
<?xml version="1.0" encoding="UTF-8" ?>
- <TIMDiscoveryHTTPResponse xmlns="http://localhost/1451HTTPAPI"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://localhost/1451HTTPAPI
      http://grouper.ieee.org/groups/1451/0/1451HTTPAPI/TIMDiscover
      yHTTPResponse.xsd">
  <errorCode>0</errorCode>
  <ncapId>4</ncapId>
  <timIds>1,2</timIds>
  </TIMDiscoveryHTTPResponse>
```

A full XML schema is available at http://grouper.ieee.org/groups/1451/0/1451HTTPAPI/. The file name is SmartTransducerHTTPResponse.xsd.

HTML response format:

errorCode: 0<br />NCAP ID: 4<br />TIM IDs: <br />1,2

In this case, the response shall be formatted as a valid webpage in HTTP 1.1 format.

TEXT response format:

0
4
1,2

For text response, individual parameters are returned and shall be terminated with a carriage return/line feed sequence (CR/LF)(ASCII 13,10)

At this example two TIMs with TimIds 1 and 2 are present in our system.

Our system is composed by one NCAP server implemented in LabVIEW as a 1451 Web Service and for the first test two TIMs who represents one CTD instrument with three and a meteorological Data Station with 40 channels as is shown in Figure 2

- CTD Instrument TIMid=1. Three Channels: conductivity, temperature and depth.
- Meteo Station TIMid=2. Forty Channels: stationDate, stationTime, outsideTemp, wind10Avg, windDir, barometer, outsideHumidity, solarRad ,uv, hiOutsideTemp, lowOutsideTemp, hiHumidity, lowHumidity, hiBarometer, lowBarometer, hiWindSpeed, hiSolarRad, hiUV, dailyRain, wind10Avg, hiMonthlyOutsideTemp, lowMonthlyOutsideTemp, hiMonthlyHumidity, lowMonthlyHumidity, hiMonthlyBarometer,

lowMonthlyBarometer, hiMonthlyWindSpeed, hiMonthlySolarRad, hiMonthlyUV, monthlyRain, hiYearlyOutsideTemp, lowYearlyOutsideTemp, hiYearlyHumidity, lowYearlyHumidity, hiYearlyBarometer, lowYearlyBarometer, hiYearlyWindSpeed, hiYearlySolarRad, hiYearlyUV, totalRain

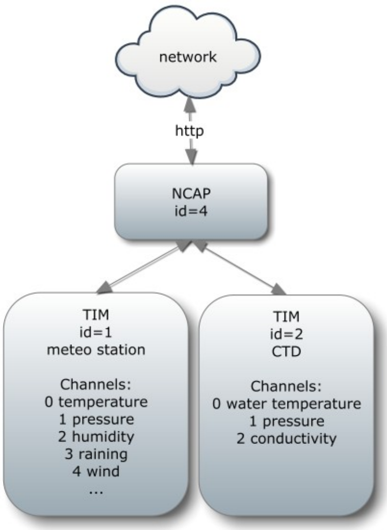


Figure 2. NCAP, TIMs and Channels of the implemented IEEE1451 system

III. CONCLUSIONS

In order to facilitate the interoperable access to the Obsea instruments data, a LabVIEW IEEE1451 HTTP web server has been developed. This development allows the OBSEA technicians to share instrument data trough a standard interface easily and quickly.

IV. REFERENCES

[1] L. Kang, "IEEE 1451: A standard in support of smart transducer networking," presented at Instrumentation and Measurement Technology Conference, 2000. IMTC 2000. Proceedings of the 17th IEEE, 2000.

[2] E. Y. Song and L. Kang, “Understanding IEEE 1451-Networked smart transducer interface standard – What is a smart transducer?,” Instrumentation & Measurement Magazine, IEEE, vol. 11, pp. 11-17, 2008.

[3] Mànuel, A.; Nogueras, M.; Del Rio, J. “OBSEA: an expandable seafloor observatory” Sea technology, ISSN: 0093-3651

[4] M. Botts, G. Percivall, C. Reed, and J. Davidson, "OGC Sensor Web Enablement: Overview and High Level Architecture."

[5] Aguzzi, Jacopo; Mànuel, Antoni; Condal, Fernando; Guillén, Jorge; Nogueras, Marc; del Rio, Joaquin; Costa, Corrado; Menesatti, Paolo; Puig, Pere; Sardà, Francesc; Toma, Daniel; Palanques, Albert. 2011. "The New Seafloor Observatory (OBSEA) for Remote and Long-Term Coastal Ecosystem Monitoring." Sensors 11, no. 6: 5850-5872.

[6] EsoNews Volume 4 Issue 1 and 2. ESONET NOE: http://www.esonet-noe.org/content/download/34923/475602/file/ESONEWS_7F-1.pdf

[7] Std. IEEE P1451.0 2007. Standard for a Smart Transducer Interface for Sensors and Actuators – Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats