Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 13 (14): 2725-2729, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.2725.2729

Interface Middleware Design and Implementation Based on Sensor Data Interface Standard

[1,2]Huajie Xu, [1]Lei Zheng, [1]Yunfei Li and [1]Zhongcheng Wu
[1]High Magnetic Field Laboratory, Chinese Academy of Sciences, Hefei, Anhui, 230031, China
[2]Department of Mechanical and Electronic Engineering, Chizhou College, Chizhou, Anhui, 247000, China

Abstract: With the development of sensor network and its application, there are large numbers of similar hardware and software to be designed and developed for sensor and sensor node, which are so inefficient and repeated. In this study, in order to achieve reusability and adaptability, the sensor data interface standard has been proposed, based on which, the interface middleware is designed. In middleware, the communication interface, interactive command and data acquisition function are provided with consistency of norm and adaptability across different platform. As a result, middleware has been applied in sensor and sensor network, meanwhile, the adaptability and reusability of middleware can be presented.

Key words: Sensor data interface standard, middleware, sensor, sensor node

INTRODUCTION

The development of sensor network and its application under the background of Internet of Things (IOT), which highlight the inefficient and duplication of design and development of hardware and software. A specification of hardware and software that follow standardization and unification is necessary from the sensor manufacturers to system integration applications, which improve the efficiency of sensor network application and development and realize the modularization and reusability of hardware and software (Zhou Hongbo, 2010; Ovidiu Vermesan, Peter Friess *et al*., 2011).

Therefore, a lot of efforts and studies have been made in industry and academic institutes. The National Institute of standards and Technology (NIST) developed a standard cluster of IEEE1451 with international organization named as IEEE. The standard cluster of IEEE1451 ranged from the IEEE1451.0 to the IEEE1451.7 (Lee, 2000, 2001), which illustrate the network accessing of actuator, transducer and sensor, communication protocols and Transducer Electronic Data Sheet (TEDS). The Open Geospatial Consortium (OGC) proposed SWE (Sensor Web Enablement) Web service middleware based on upper network application and development. PSI5 and AS-i proposed by Bosch (BOSCH) and international Association of AS-i made corresponding specification of sensor and actuator from electrical interface to communication interface. Although, these technical specifications and standards are used in a range of industries, these did not become a norm or standard that widely followed (Song, 2008; Lee, 2001).

In order to adapt to the development of IOT architecture, China Standardization Working Group on Sensor Networks (WGSN) was established in Beijing on September 11, 2009. It is a technical organization that approved and led by the National Standardization Management Committee, which engaged in sensor network standardization. Sensor network standard is an important part of the technical standard system of IOT in China. The standard is mainly relates to unity and standardization of the general specification, interface, communication and information exchange, service support, collaborative information processing, network management, information security and testing etc.. The data interface middleware are designed and developed under the guidance of the part II of sensor network interface named as data interface standard.

SENSOR DATA INTERFACE STANDARD

Sensor data interface standard is the second part of the sensor network interface, which presents specification that illustrates sensor data description information classification, data description format, information interaction mode. These realized the information interaction and adaptive configuration and recognition. Meanwhile, specification of interaction protocol is also detailed. Figure 1 shows that the sensor data interface standard interaction protocols cover with the digital communication sensor and upper sensor network

Corresponding Author: Huajie Xu, High Magnetic Field Laboratory, Chinese Academy of Sciences, Hefei, Anhui, 230031, China

Table 1: Command word code

Entries	Code	Description
Command word (0×00-0x1F, 0×00-0×01, 0×0C-0×1E reserved)	0×02(CMD_RMPROF)	Remove data description file
	0×03(CMD_RMPROF_ACK)	The node feedback after removing data description file
	0×04(CMD_WRPROF)	Configuration data description file
	0×05(CMD_WRPROF_ACK)	The node feedback after configuration data description file
	0×06(CMD_RDPROF)	Read the data description file
	0×07(CMD_RDPROF_ACK)	The node feedback after read data description file
	0×08(CMD_DATA)	Read data
	0×09(CMD_DATA_ACK)	The node return sensing data while data command is received
	0×0A(CMD_DATACTL)	Data control
	0×0B(CMD_DATACTL_ACK)	The node feedback after the data control command received,
	0x1F(CMD_ERR)	Error flag

Table 2: Data interaction frame format

Frame header	Data length	Command word	Data length of command	Data	CRC8	Frame end

Table 3: Bytes allocation of data interaction frame segment

Entries	Bytes	Description
Frame header	2	Bounded as 0×00AA
Data length	2	Represent the length of [command word+ data length of command area +data]
Command word	1	Represent the type of command
Data length of command area	2	Data length of command area
Data	N	Command ancillary data, such as configuration data description file, command, must attach the data content of the data description file, the data area also requires the use of the TLV format description?
CRC8	1	8bits CRC redundancy code of data that besides the header and the CRC code.The CRC polynomial provisions: C(x) = x^8 + x^2 + x + 1 (CRC-8-CCITT)
Frame end	2	Bounded as 0xAAFF

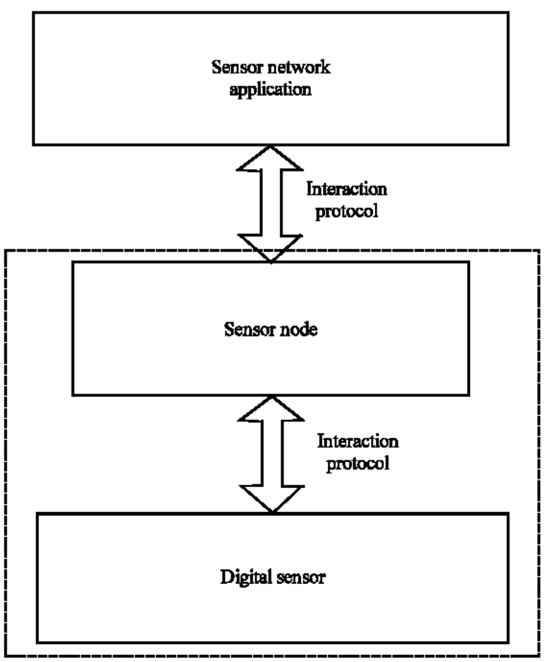

Fig. 1: Architecture of interaction protocol

applications. The position and role of sensor data interface standard are depicted as shown in Fig. 1. Sensor node can configure digital sensor and read data acquired from digital sensor by interactive protocol, by which, network application can configure and manage sensor nodes and digital sensors.

The interactive protocol command word and function are shown in Table 1. Sensor data can be exchanged based on the unified command word, which applied among the network application, the sensor node and digital sensor. The middleware in this paper follows the specification in the communication interface.

Data exchanging is usually realized in frame format as shown in Table 2. A frame is consist of header, data length, command word, command length, data, CRC8 and frame end. Meanwhile, sensor data is encapsulated in TLV format.

Allocation of bytes in interaction frame is shown in Table 3, in which, every segment has fixed bytes except data segment. Data segment in TLV format has a variant length, so that the bytes of whole frame are uncertain.

INTERFACE MIDDLEWARE DESIGN AND IMPLEMENTATION

During the development of Internet of Things (IOT), middleware is an important technology (Sun Qibo, Liu Jie *et al*., 2010; Shao Huagang, Cheng Jin *et al*., 2010; Heinzelman *et al*., 2004), which can suit to the layer architecture of IOT, follow standard, modularization, loose coupling and reusability, so that the efficiency of application and development can be improved significantly.

Architecture of interface middleware: Sensor data interface middleware is based on sensor data interface

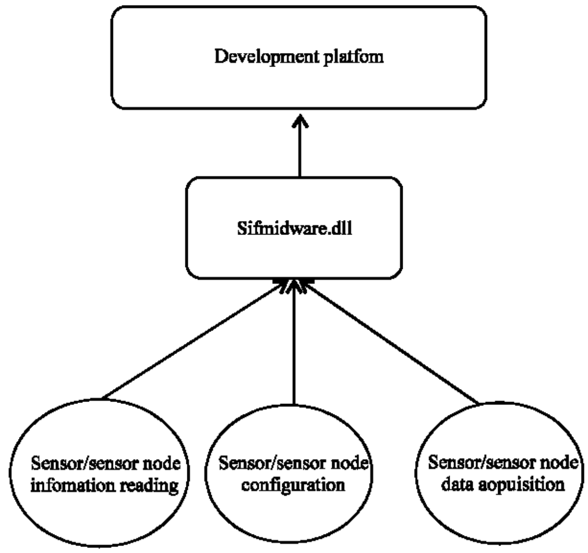

Fig. 2: Interface middleware architecture

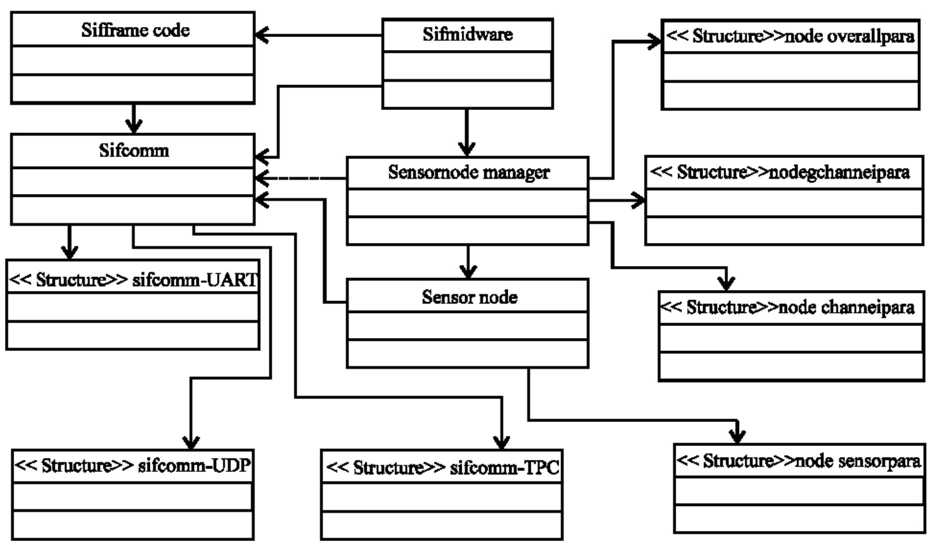

Fig. 3: Sensor interface middleware implementation and relation

standard, meets the application of sensor network mutual adaptation and interaction and is developed follow by the command word and its format in sensor data interface standard. Meanwhile, this middleware take account of the using of multi platform development and implementation. The interface middleware function hierarchy is shown in Fig. 2. The middleware is developed in the form of dynamic library. Reading and modifying of binary XML configuration information and data collection are the main function of middleware. The upper application development platform loads this middleware to develop applications for sensor network, so as to achieve the reusability and transparent operation.

Interface middleware design: SifMidWare class mainly contains SensorNodeManager, SifFrameCode and SifComm. SensorNodeManager can manage sensor and sensor node, scan docking sensor, read configuration information of sensor or sensor node and acquire sensor data. SifFrameCode can decode frame. SifComm is the basic class of interface communication, which provides serial, TCP/UDP communication. The architecture and implementation of these classes are shown as in Fig. 3.

Sensor interface middleware is mainly developed in C++ and C that can be called by advanced language such as Java, C#, VB, VC, Delphi. In order to call interface middleware on platform of LabVIEW, Matlab etc., The

```
#define SIF_API    extern "C" _declspec(dllexport)
int InitNodeComm(void * Para,char *ParaType)
{
        return SifMidWareInstance.InitNodeComm(pSifMyNode,Para,ParaType);
}
int CloseNodeComm()
{
        return SifMidWareInstance.CloseNodeComm(pSifMyNode);
}
//Remove node parameter cluster
int RMNodeOverallPara()
{
        return SifMidWareInstance.RMNodeOverallPara(pSifUesr,pSifMyNode);
}
int RMNodeGChannelPara(uint8 GChannelNO)
{
        return SifMidWareInstance.RMNodeGChannelPara(pSifUesr,pSifMyNode,GChannelNO);
}
int RMNodeChannelPara(uint8 ChannelNO)
{
        return SifMidWareInstance.RMNodeChannelPara(pSifUesr,pSifMyNode,ChannelNO);
}
int RMNodeSensorPara(uint8 ChannelNO)
{
        return SifMidWareInstance.RMNodeSensorPara(pSifUesr,pSifMyNode,ChannelNO);
}
int RMNodeSignalDescPara(uint8 ChannelNO)
{
        return SifMidWareInstance.RMNodeSignalDescPara(pSifUesr,pSifMyNode,ChannelNO);
}
```

Fig. 4: Part of encapsulated code

extern interface is encapsulated again so that it can be called in pure C language. Some encapsulated codes are shown in Fig. 4.

SENSOR INTERFACE MIDDLEWARE APPLICATION

Sensor standard interface module configuration platform is developed in C# based on Microsoft. NET. There are two modules that consists of scanning module which sending inspection order through serial port or Ethernet to identify docking sensor or sensor node. Sensor configuration module is the second module that is realized by calling interface middleware in dynamic library form such as SifMidWare.dll. Sensor information configuration is to transform the parameter to binary xml file and write into EEPROM in sensor.

Sensor data acquisition is implemented on sensor calibrating, sensor testing and verifying platform based on the LabVIEW platform by calling the interface middleware.

Sensor data acquisition can acquire data through serial port or Ethernet, through which, sensor information reading and configuration also implemented.

CONCLUSIONS

Interface middleware is developed based on sensor data standard and called in dynamic library form, by which, interface module configuring platform and sensor calculating, testing and verifying platform are achieved. Application result indicates that sensor network development and design can be achieved on different platform. Data interface standard function and reusability and efficiency of software development have been illustrated well.

Communication method of sensor interface middleware should be improved in next step, which developed in pure C language, unrelated with system function and can be loaded in embedded system platform for sensor network.

ACKNOWLEDGMENT

This stuudy is supported by a grant from Natural Science Foundation of the Education Department of Anhui Province (NO. KJ2012A207).

REFERENCES

Heinzelman, W.B., A.L. Murphy, H.S. Carvalho and M.A. Perillo, 2004. Middleware to support sensor network applications. IEEE Network, 18: 6-14.

Lee, K., 2000. IEEE 1451: A standard in support of smart transducer networking. Proceedings of the 17th IEEE Conference on Instrumentation and Measurement Technology, Volume 2, May 1-4, 2000, Baltimore, MD., pp: 525-528.

Lee, K., 2001. Sensor networking and interface standardization. Proceedings of the 18th IEEE Conference on Instrumentation and Measurement Technology, Volume 1, May 21-23, 2001, Budapest, Hungary, pp: 147-152.

Shao, H.G., J. Cheng, H. Wang and Z. Li, 2010. The system based on internet of things and its middleware. Comput. Eng., 36: 84-86.

Song, E.Y. and K. Lee, 2008. Understanding IEEE 1451-Networked smart transducer interface standard-What is a smart transducer? IEEE Instrument. Measur. Mag., 11: 17-10.1109/MIM. 2008.4483728.

Sun, Q., J. Liu and S. Li, 2010. The internet of things: concept, architecture and key technology research. J. Beijing Univ. Posts Telecommun., 33: 1-9.

Vermesan, O., P. Friess, P. Guillemin, S. Gusmeroli and H. Sundmaeker et al., 2011. Internet of things strategic research roadmap. http://www.sintef.no/home/Publications/Publication/?pubid=SINTEF+S19802.

Zhou, H., 2010. The Internet of Things: Technology, Applications, Standards and Business Mode. 1st Edn., The Electronic Industry Press, China.