

New programming framework for Wireless sensor network

FIHRI Mohammed¹, EZZATI Abdellah²

^{1,2} LAVETE laboratory Mathematics and Computer Science Dept Sciences and Tech Fac Settat, 26000, Morocco

Corresponding Author: FIHRI Mohammed

Abstract : *Wireless sensor networking is an innovating area. Where we use specific equipment in order to measure small rated data, which often have a major importance to the user such as temperature or input output state and share it wirelessly. In this paper we will try to make WSN simple to use by common developers that not necessary have the WSN networking knowledge; by creating a new programming framework that is specially created for Linux based operating systems. The use of wireless sensor network involves the use of a routing protocol, security mechanism, application communication layer... How we can provide those solutions and there applications without many coding and in easy way? It's the role of our new programming framework that will be describe in this paper, it.s concept of external calling system depending on the functionality, and its architecture. Finally we present the conformity of our framework with the SCADA System.*

Keywords: Framework, WSN, Programming

Date of Submission: 02-10-2017

Date of acceptance: 18-10-2017

I. Introduction

We can define WSN as an infrastructure-less network that use self-configurations algorithms to create a network in order to monitor physical data. Typically we use a lot of nodes, each node is constructed by one or several sensors that collects the physical or environmental data and transfer it into electrical signal, a microprocessor, an emitter/receiver radio card and a power unit.

A framework is a conceptual structure that deals with one or several specific problems. Usually is a set of programs, tools, materials or functionality that assured the right treatment of a programming problem. This set of programs avoids us to repeat the same part of code in order to solve the same problem, which increase the development speed of a project. There will be no need to re-develop a whole lot of functions. With the use of a framework we can choose which solution to implement between the provided ones by the framework. Not all frameworks respond to the same needs, and several frameworks can be used jointly in certain situations.

In order to improve the use of WSN in the real life, we will try to support in our framework the most common sensors like DHT22 or camera with keeping the framework open to any kind of sensor. Two routing protocols will be also supported Zigbee for one state (district input) sensors or low bandwidth and OLSR for high bandwidth use. Also the framework will present two types of sending data active and passive. in the active communication the sensor node receive

II. Wireless Sensor Network

Wireless network is type of computer networks that use the radio wave in order to connect various network's elements to each other. There are different types of wireless networks: Wireless Wide Area Networks, Wireless Local Area Networks, Wireless Metropolitan Area Networks and Wireless Personal Area Networks. Wireless Personal Area Networks (WPAN) as its name indicates is for personal use and can cover only a small area, like home where the television, garage door, fire alarm and phone are connected to the computer using Bluetooth or IEEE802.15.4. It's often infrastructure-less with small bandwidth but more energy efficient and less expensive.

We can say that WSNs belongs to WPAN category, and at first was only meant for military use. During the cold war the United States Army start to deploy a primitive but operational type of sensors in order to monitor submarine activities in north Americas. But the real history start in the 80's when DARPA try to create a distributed sensor network (DSN) that can perform distributed computing, signal processing and tracking using acoustic sensors; while maintaining a collaborative work, autonomy of nodes and cost-effective.

The major aspect of WSN is that the sensor nodes are densely deployed in an open space; on a battlefield in front of, or beyond, enemy lines; in the interior of industrial machinery; in a biologically and/or chemically contaminated field; in a commercial building; in a home; or in or on a human body. A sensor node typically has embedded processing capabilities and onboard storage, but it's limited in power, computational capacities, and memory.

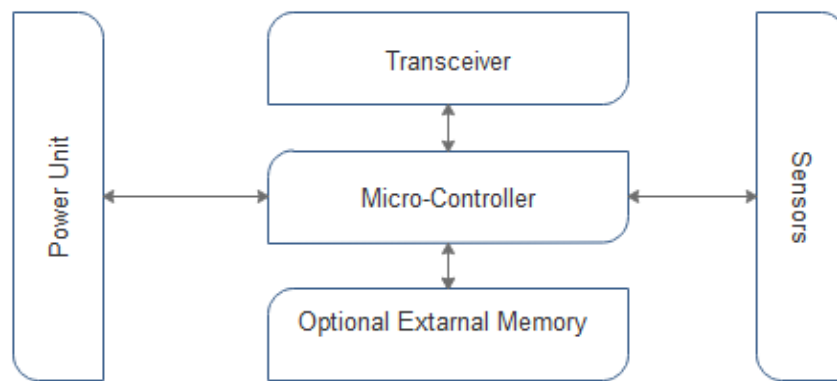


Figure 1. Wireless Sensor Node Architecture

In a sensor node, the main functionality of an embedded processor or microcontroller is to be the middleman between the other composites of the node such as sensor devices and transceiver. The microcontroller is also responsible on every computing or running algorithm in order to communicate with sensors with their specific communication protocol, store the measured information in the internal or external memory and process it if needed, and manage the power.

A Wireless Sensor network can be homogeneous or heterogeneous. In the first case the nodes are similar to each other in every aspect construction, purpose and functions but it can be different in the sensed data type. In the second one the nodes are different in their construction, and it depends on their functionality or the role that they play. Generally speaking, it possible to divide the nodes to three types, end-device, router or internal-gateway and network coordinator or base-station (also can be called root-access-point or external-gateway).

The end-device is a sensor node with one main functionality, which is to sense and sent the results to it network parent. It should not in any case to be a part of networking aspects.

The router is the intermediate device between the end-device and the base station. Often it's responsible on the routing aspect and coordination in a part of the deployment area, and sometimes gathering and processing data before send it to the base station. It can be also equipped with sensors like the end-device.

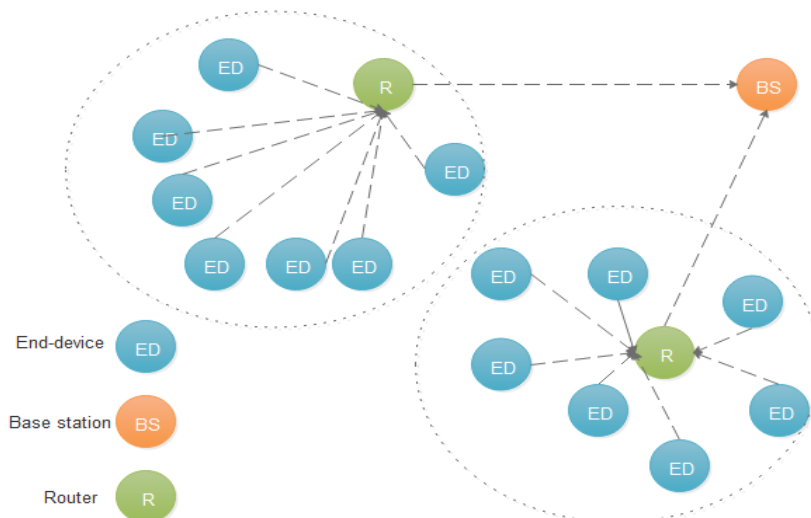


Figure 2. Types of Wireless Sensor Network Nodes

The base station take the hand on the coordination of the network like synchronizing the sending of data and topology control. Also it responsible on sending the sensed data to the other type of network where the storage server is located. Usually is provided by a good and long range transceiver, high level power source if not lined, powerful computing unit and connected to another network such us internet or satellite.

The topology of a sensor networks changes very frequently, due to the death or the sleep mode of some node and sometimes due to the mobility of the sensors nodes. There are several type of topologies in WSN Star, Tree, Mesh, Hybrid and even sometimes a customized topology that depends on the specific use but we can classified it under hybrid. WSNs support a wide range of applications, such as radiation and nuclear-threat

detection systems; weapon sensors for ships; battlefield reconnaissance and surveillance; military command, control, communications, intelligence, and targeting systems; biomedical applications; habitat sensing; seismic monitoring; physical security; process control; air traffic control; traffic surveillance; video surveillance; industrial manufacturing automation, distributed robotics; weather sensing, environment monitoring, and building and structure monitoring.

III. Concept

We can divide the frameworks in two types. The Application frameworks, such as web application frameworks (such as Django, Ruby on Rails, Symphony or Catalyst) or desktop application development frameworks (Cocoa on Mac, Qt in cross-platform...). We also find frameworks specifically addressing some development issues, such as logging frameworks (eg Log4J in Java), persistence frameworks and ORM frameworks (eg Hibernate in Java, SQLAlchemy in Python, Propel In PHP ...), Web content presentation frameworks (Bootstrap, Foundation ...). Our Framework will be specially designed to deal with every aspect of WSN communications (both network and sensor), storage, notification and OS using. Also the framework will present an easy use interface which make it an applications framework.

The main concept of our framework is the combination of several programs. Where every program is coded with the adequate language(c, python, perl and php...) and can ensure one and only functionality with no forking concept. Each program use another code via an external system call when a functionality is needed. Each program does not know when it's been calling and by who, the only present coupling is that each program except a specific parameters and return even a state return or results. We can say that our framework is a collection of codes (sensor's driver, linux libraries, routing protocols code, spi communication support, AT command interpreter, different database support code, collection of open source software...). This collection of codes is organized in deferent directories depending on its function (will be describe in the next section) and it's scheduled using intermediate programs that read and write in configuration's files.

IV. Architecture

In the next figure we will see the architecture of our framework. The PHP&CGI interfaces are used as IHM that allow to the user to communicate with our framework (even though some of the main functionality still existed in file configuration more work must be done). The php interface pass the collected parameters to the configuration builder (in case of configuration) or resources demander. The configuration builder is used to create a configuration file from the collected parameters and pass to the resource demander that call the main resource responsible on file storage. As for the resources demander its only job is to detect where the request is coming from and depending on the source and parameters a main resource is defined. The main resource is where all the systems calls are located, each main resource is a sequence of system call of other codes depending on the parameters we can say that it's the responsible of automatizing the refactor. Each functionality of the system has its own main resource. Data collector scripts are executed by the main resource and call the appropriate sensor driver in sensor communications layer in order to get a new data, and can also in some cases to call the data storage scripts to get some needed information about the concerned sensor. The data storage scripts support until now two data base Mysql and DB2 but it can support more just by adding a new script. The data to file generator is a three scripts program (one for XML generation, the second for log, and configuration files) where we enter the information, then we formatted those information depending on the out parameter, or simply to update an already existing file. The other devices communications layer is a set of independent codes that take the information parameters and send it to the chosen device (SMS Modem, Vsat, 3g dongle...) in order to be sent. In case of a specific communications for the destination system like routing protocol compilation the OS scripts schedules are called in order to put or take things from the operation system.

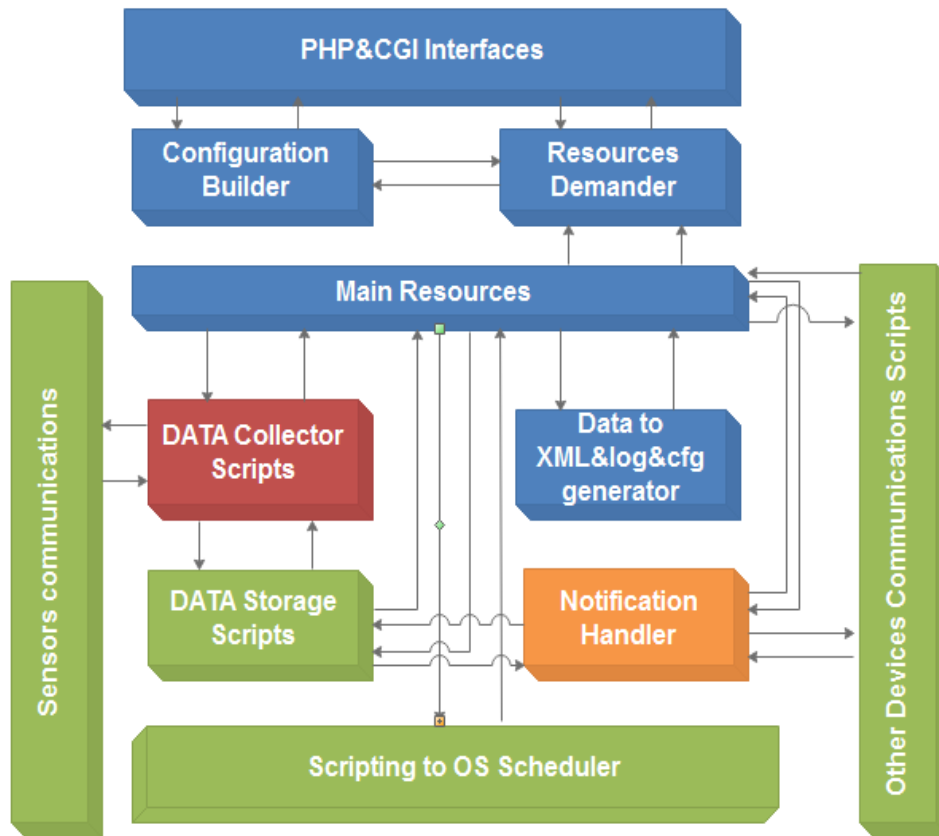


Figure 3. Framework Architecture

Until the redaction of this article the framework can run in any linux distribution, however only two types of nodes can be programmed via our framework. The first one is any node that use ARM processor with 2.6 and higher until 3 linux kernel, and the second one is any Atmega328 based node. In the case of ARM node like raspberry framework upload the files using SSH or local configuration, and in case of Atemega node the files are uploaded via USB to serial communication using an arduino IDE, the framework generate only the codes in this case.

Page d'accueil Monitoring Map Configuration Statistiques Journalisation Déconnecter

Config du box

Site : Rabat_Massira

Box : Rabat_Massira_IO_1

Liste des pins de Rabat_Massira_IO_1

Numero	statut	niveau	Equipement	Lieu	Description	Action	Mise à jour
1	1	Critique	Onduleur1	lieu1	Alarme generale	SMS/mail	✓
2	1	Warning	Onduleur1	lieu1	DC default	0	✓
3	1	Majeur	Onduleur1	lieu1	DC Defaut Tention	SMS	✓
4	1	Critique	Onduleur1	lieu1	Shutdown Actif	SMS/mail	✓
5	1	Warning	Onduleur2	lieu2	Test	Mail	✓
6	1	Critique	Onduleur2	lieu2	DC Defaut Batterie	SMS/mail	✓
7	1	Critique	détecteur d'incendie	lieu1	alarme feu	SMS/mail	✓
8	1	Critique	Redresseur1	lieu1	Defaut general Redresseur	SMS/mail	✓
9	1	Majeur	Redresseur1	lieu1	Alarme generale	SMS	✓
10	1	Critique	climatiseur	lieu1	Alarme generale	SMS/mail	✓
11	1	Critique	Climatiseur2	lieu2	Alarme generale	SMS/mail	✓
12	2	0			0	0	✓

Figure 4. IHM example of distrect I/O node configuration

APPLICATIONS “CASE STUDY SUPERVISORY CONTROL AND DATA ACQUISITION”

SCADA is not a specific technology, but a type of application. SCADA stands for Supervisory Control and Data Acquisition — any application that gets data about a system in order to control that system is a SCADA application. A SCADA application has two elements:

1. The process/system/machinery we want to monitor a control — this can be a power plant, a water system, a network, a system of traffic lights, or anything else. So our framework must be open to integrate any new driver[7].
2. A network of intelligent devices that interfaces with the first system through sensors and control outputs. This network, which is the SCADA system, gives the ability to measure and control specific elements of the first system. We can build a SCADA system using several different kinds of technologies and protocols. In the case of the presented framework we have two types of networks, an internal network that use the principal of wsn and external that can be any type of normal network. SCADA is used around the world to control all kinds of industrial processes. SCADA can help to increase efficiency, lower costs and increase the profitability of your operations. In terms of detect current flow and line voltage, to monitor the operation of circuit breakers, and to take sections of the power grid online or offline.

The previous aspects can be configured by our framework; until now our framework can ensure all the SCADA functionality. If more comparison must be done in terms of functionalities [7] would be a good reference.

V. Conclusion

The use of the framework is easy and does not require a solid coding skills, but more work must be done like supporting more sensors, sensor's platform and adapting the code to the new kernels. We must say that our work is not finished yet but it's only an opening for more related research such as implementing a real leach protocol and more supported platform for the framework.

References

- [1] CARVALLO, A. AND COOPER, J. THE ADVANCED SMART GRID. BOSTON: ARTECH HOUSE, 2015. PRINT.
- [2] THOMAS, M. AND MCDONALD, J. POWER SYSTEM SCADA AND SMART GRIDS. CRC PRESS, 2015.PRINT
- [3] MCCRADY, S. DESIGNING SCADA APPLICATION SOFTWARE: A PRACTICAL APPROACH. ELSEVIER, 2013. PRINT.
- [4] Z. ZHAO, G.-H. YANG, Q. LIU, V.O.K. LI AND L. CUI. IMPLEMENTATION AND APPLICATION OF A MULTI-RADIO WIRELESS SENSOR NETWORKS TESTBED. WIRELESS SENSOR SYSTEMS, IET, VOL. 1, NO. 4, PAGES 191–199, 2011. 28
- [5] G. BARRENETXEA, F. INGELREST, G. SCHAEFER, AND M. VETTERLI, “SENSORSCOPE: OUT-OF-THE-BOX ENVIRONMENTAL MONITORING,” IN ACM/IEEE IPSN , 2008 BAYDERE, S.,SAFKAN, Y., AND DURMAZ, O. 2005.
- [6] Robert Faludi.Building Wireless Sensor Networks: With Zigbee, Xbee, Arduino, And Processing. O'Reilly Media, 2011. Print.
- [7] DPS Solution, Dpstelecom/Com

FIHRI Mohammed New programming framework for Wireless sensor network.” IOSR Journal of Computer Engineering (IOSR-JCE) , vol. 19, no. 5, 2017, pp. 46-50.