COMPARATIVE STUDY OF OPEN SOURCE TOOLS FOR INTERNET OF THINGS

Sumangali K¹, P Kiran²
Department of Information Technology, School of Information Technology and Engineering, VIT University, Vellore – 632 014, Tamilnadu, India.
Email: sumivenkata@gmail.com

Received on 25-10-2016
Accepted on 02-11-2016

Abstract:

This paper discusses about the open source tools of Internet of Things such as Contiki OS and Node Red in order to design error less network because IoT which connects the billions of devices and for specific applications the data delivered by the sensors must be maintained secure this paper discusses about design of a sensor network and communication between them by sending hello packet by using node red flow based simulation has been done in IoT platform.

Keywords: Contiki OS, Internet of Things, IoT, Node red, sensors.

1. Introduction

Open source tool is an expression used to define a project or tool that performs a particular task, in which the source code is readily available for utilization with its unique outline, free of charge. Open source tool are normally made as a collective effort in which developers enhance the code and offer the changes inside the group, and is normally accessible at no charge under a permit characterized by the Open Source Initiative. Open source tool may be feasible alternative options for prominent licensed source applications and some open source tools offers highlights or execution advantages that surpass their business rivals. The open source tool is synonymous with open source utility and like open source applications [1, 4]. Open source programming incorporates working protocols, applications, and projects in which the source code is distributed and made accessible to people in general, empowering anybody to duplicate, change and redistribute that code without paying expenses. Open source "items" commonly develop through group participation among individual software engineers and also extremely extensive organizations. An open source permit grants anyone in the group to study, change and distribute the product for nothing and for any reason.
Internet of Things (IoT) speaks about a general idea for the capacity of network devices to sense and gather information from our general surroundings, and impart that information over the Internet where it can be transformed and used for different fascinating purposes. Some likewise utilize the term modern Internet reciprocally with IoT. This points out fundamentally to business utilizations of IoT innovation in the world of assembling. The Internet of Things is not constrained to mechanical applications, alone. The Internet of Things (IoT) depicts a system of physical items that join with one another through the internet. Items, or "things" can exchange data remotely without human interaction. A "thing" can be any item that can be allocated by an IP address and furnished with the capacity to exchange information over a network. While the IoT is as yet to evolve completely, it has already became popular in the associated world in which individuals, devices, situations and virtual items are all joined and fit for connection. There's another wave of innovation stages focusing on the need to extend these refined correspondences, and in addition equipment producers are creating physical devices and sensors to power the IoT [4].

2. Opensource Tools for IOT: In this section, we discuss about some of the innovative open source tools serving the IoT. These tools provide various options to develop interesting IoT applications. These tools provide open source implementations of IoT guidelines, and to make it less complex for designers to make IoT applications.

A. Contiki OS: Contiki is an open source operating system for networked, memory constrained systems with a particular focus on low-power wireless Internet of Things devices. This was developed by Adam Dunkels in 2002 and further modifications were carried out by the world wide team of developers [2].

B. Node Red: Node-RED is an open source tool for flow design using local host i.e. browser based flow editor and even we can connect to the hardware devices such as the raspberry pi, Arduino Uno light weight development boards as well as in the cloud environment here we are using nodes.js which is also an open source that runs on chrome’s java script

<table>
<thead>
<tr>
<th>Table1. The role of open source in building Internet of things.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Applications</strong></td>
</tr>
<tr>
<td>Frameworks</td>
</tr>
<tr>
<td>Enabling M/W</td>
</tr>
<tr>
<td>OS</td>
</tr>
<tr>
<td>Firmware</td>
</tr>
<tr>
<td>Dev Tools</td>
</tr>
<tr>
<td>Hardware</td>
</tr>
</tbody>
</table>
3. Detailed Discussion about Contiki OS

A. Hardware Contiki is designed to run on classes of hardware devices that are severely constrained in terms of memory, power, processing power, and communication bandwidth. A typical Contiki system has memory on the order of kilobytes, a power budget on the order of milliwatts, processing speed measured in megahertz, and communication bandwidth on the order of hundreds of kilobits/second. This class of systems includes both various types of embedded systems as well as a number of old 8-bit computers [3].

B. Networking Contiki provides three network mechanisms i.e the TCP/IP stack, which provide IPv4 networking, the IPv6 stack, which provides IPv6 networking, and the Rime stack, which is a set of custom lightweight networking protocols designed specifically for low power wireless networks. The IPv6 stack was contributed by Cisco and was, at the time of release, the smallest IPv6 stack to receive the IPv6 Ready certification. The Rime stack is an alternative network stack that is intended to be used when the overhead of the IPv4 or IPv6 stacks is prohibitive. The Rime stack provides a set of communication primitives for low-power wireless systems. The default primitives are single-hop unicast, single-hop broadcast, multi-hop unicast, network flooding, and address-free data collection [4]. The primitives can be used on their own or combined to form more complex protocols and mechanisms.

C. Low Power Operation

Contiki systems are severely power-constrained. Battery operated wireless sensors may need to provide years of unattended operation and with little means to recharge or replace its batteries. This provides a set of mechanisms for reduction of power consumption of the system on which it executes. The default mechanism for attaining low power operation of the radio is called ContikiMAC. With ContikiMAC, nodes can be running in low power mode and still be able to receive and relay radio messages.

D. Installation Procedure for Contiki OS

Contiki can be available in a single download file which contains an entire Contiki development environment. This works on Linux virtual machine environment that runs in VMWare player and having Contiki and all development tools, simulators and compilers already installed in Contiki development. The present version of the Contiki is 2.7. Steps for installation of Contiki os
1. Install the VMware player
2. Get the Contiki virtual image of size 2.5GB
3. Open the VMware virtual machine and Contiki then wait for the login screen to appear
4. Provide the password as “USER”
**E. Simulation** The Contiki system has many prebuilt modules which can readily run on cooja simulator or on the real hardware platform. Two ways to open the cooja simulator. 1. Double click on the cooja icon which is present on desktop. 2. Opening through terminal and enter into the cooja directory.

**F. Adding sensor motes**

When the simulation window is generated, add motes to the simulation window.

i.e Menu => Motes => Add Motes.

The type of motes should specify after adding motes to the simulation window. We do have more than ten different types of motes. Some of them are MicaZ, Sky, Trxb1120, Trxb2520, cc430, ESB, eth11, Exp2420, Exp1101, Exp1120, WisMote and Z1.

The contiki operating system will generate an object code for these motes on real hardware and also on simulator when the hardware platform is not available.

**Processing Steps for the simulation**

1. For adding a mote, select Add Motes Select any of the motes given then you will get the screen shown in Fig. 2.

2. Cooja opens the Create Mote Type dialogue box, which gives the name of the mote type as well as the Contiki application that the mote type will run.

3. Once it is compiled without errors then click Create (Fig. 2).

4. Enter the number of motes that has to be created and in which position they has to place. In this example, we created ten motes and placed in some particular positions. Click the Start button in the Simulation Control window and enable
the mote’s Log Output: printf() statements in the View menu of the Network window. The Network window shows the output Hello World in the sensors. This is a simple output of the Network window.

Fig 2: Log output in motes.

Fig 3: Simulation results in Contiki.

The entire simulation result is shown in Fig. 3. The result of the above Hello World application can also be run using the terminal. To compile and test the program, go into the Hello World directory:

```
sumi@localhost$ cd /home/user/Contiki/examples/hello-world
sumi@localhost$ make
```

This will compile the Hello World Hello World application to be compiled into a single program that can be run by typing the following command:

```
sumi@localhost$./hello-world
```

This will print out the following text:

Contiki initiated, now starting process scheduling Hello world.

The program will then appear to hang, and must be stopped by pressing Control + C.

4. Node Red:
Node red is a tool for designing and connecting hardware devices for internet of things Node-RED is creation of IBM. Node-RED is an open source tool for flow design using local host i.e. browser based flow editor and even we can connect to the hardware devices such as the raspberry pi, Arduino Uno light weight development boards as well as in the cloud environment here we are using nodes.js which is also an open source that runs on chrome’s java script [5].

G. Flow Based Programming (FBP)

Exchange of messages via the edges which describes a graphs of nodes.

![Node Red Diagram](image)

**Fig 4:** Twitter application design using node red.

UNIX PIPE is the best example for the Flow based programming while a stream of data is running and in the parallel small applications will run.

**Table 2:** Comparison between Node Red and Contiki OS

<table>
<thead>
<tr>
<th>Comparisons</th>
<th>C</th>
<th>java</th>
<th>Multi-threading</th>
<th>MCU</th>
<th>Modularity</th>
<th>Real time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contiki os</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Node Red</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

5. Conclusion:

The open source community has provided a huge contribution towards the development of IoT. The data accumulated by computers in IOT has been utilized to keep track and measure everything. An open source tool along with IOT has helped tremendously to reduce the cost and also proper utilization of resource. It has the potential to redesign and replace the world. The open source community is also growing with the development of IOT simultaneously. This has helped
the world to adopt to the gadgets with new technology easily. The IOT is a new trend across a wide range of industries and its effect will last for very longer period of time. To ensure this success, the IOT needs to utilize open source standards and open source tools.

References


