# Internet of Things using RIOT and KAA software platform on AT9186 Radio Interface

**Saminath.V**

Team Lead

***Abstract***-The Internet of Things (IoT) supports from low-power 8-bit microcontrollers to processors designed for smartphones. There are huge number of IoT supporting platforms are readily available to interface with Hardware and easy to make End products. This paper explains the IoT connectivity between Hardware platform and Application development using open software source framework like KAA and RIOT. The ATMEL based AT91 Device has connected with radio interface and make 6LowPAN networks, border router configuration it exposes IP information to external world. The consumer IoT products will be remotely monitored, control, event-driven operations and user data route from network to web applications.

***Index Terms***- IOT: Internet of Things, SLIP: Serial Line Interface Protocol, RPL: Routing Protocol Algorithm, OS; operating system, protocol, IPv6, wireless, Radio.

## I. Introduction

An open source project which provides a universal software framework and core set of system services that enable interoperability among connected products and software applications across manufacturers to create dynamic proximal networks. The products for the home, automotive and the enterprise are connected globally and update data to cloud.It can interact in new, exciting and useful ways that will engage and delightful to users. The emerging Internet ofThings (IoT) concept is considered to be the next technological revolution, one that realizes communication between many types of object.

The protocols can be described in this framework are:
1) MQTT: a protocol for collecting device data and communicating it to servers (D2S)
2) XMPP: a protocol best for connecting devices to people, a special case of the D2S pattern, since people are connected to the servers
3) DDS: a fast bus for integrating intelligent machines (D2D)
4) AMQP: a queuing system designed to connect servers to each other (S2S)

Node communication has lot of constraint in terms of computing power and available memory. In Radio interface has possibility to save energy, little bandwidth and limited packet sizes. IoT creates unique things (Device) and virtually mapped to internally. Radio-frequency identification (RFID) is mandatory for the Internet of Things to create and communication establish.IoT is designed for Heterogeneous platforms with multiple network composition.

RIOT is an open source operating system deliberately designed for Internet of Things (IoT) developments. RIOT based on microkernel, and cryptographic libraries, protocol support (wiselib, IPV6, UDP,6LowPan, NHDP), flexible memory management. It supports for various microcontrollers, radio drivers, sensors. These requirements comprise a low memory, high energy efficiency, real-time capabilities, high degree of modularity and configurable communication stack, and support for a wide range of low-power devices. RIOT supports C/C++ language for development, multithread programming, real time kernel. It supports various protocol. Each process runs in separate thread with its own memory configuration.

The system is based on a microkernel and offers real time multi-threading to supports maxi-mum modularity. RIOT implements the micro-kernel architecture inherited from FireKernel. RIOT kernel designed to fulfill strong real-time requirements for high priority processing.Zero-latency on interrupt handler so the Context-switching has minimum time combined with thread priorities.Timer tickles scheduler function implemented on RIOT, it schedules timers to wake up periodically and check running process state.

The 6LoWPAN is header compression mechanism it allows IPv6 data routed into general networks. The relationship between 6LoWPAN and IPv6. TheIPv6 allows several address on the device for local and global communication. The IPv6 addresses contain several zeros that can be shortened in several forms. The 128 bits of IPv6 addresses are divided in two parts: the network prefix (64 bits) and the host address (64 bits). Edge Router performs the header compression in 6LoWPAN and IPv6 network.The loopback address (::1) is a similar address than 127.0.0.0 in IPv4 used to check the status of the interface and local-link addresses (FE80::/10) is used to communicate with devices in the same link

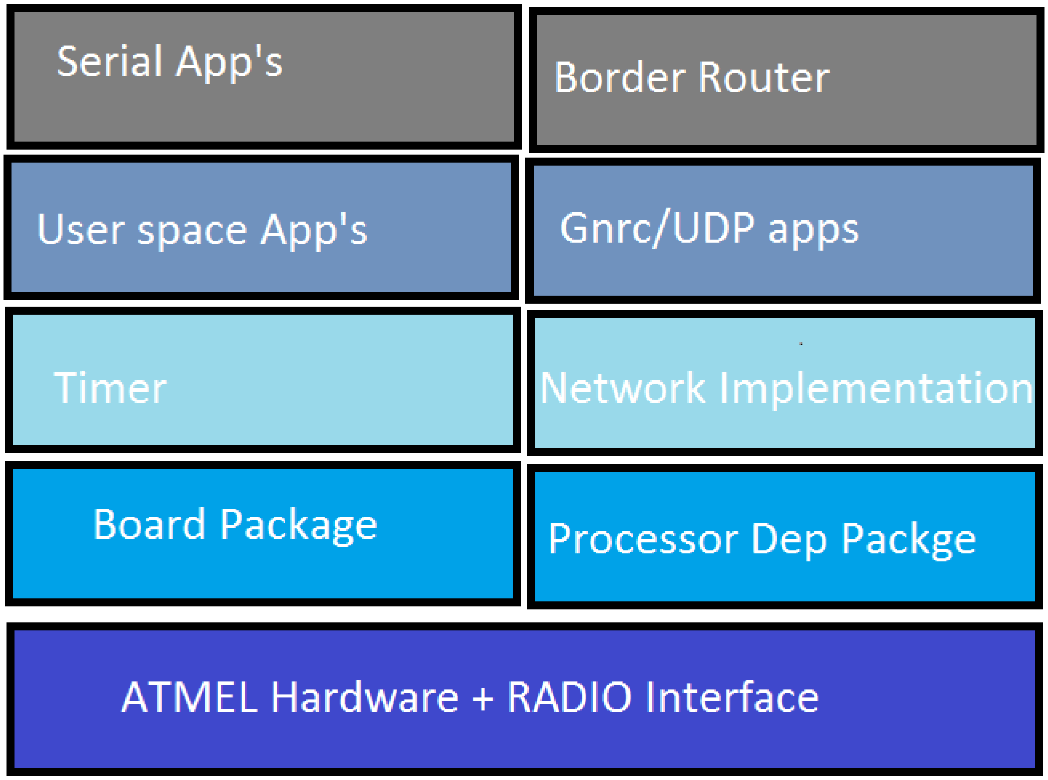


Figure-:1 RIOT Operating System Internal

This experimental study we used ATMEL development platform which SoC includes a SAMD21 ARM Cortex-M0+ micro-controller bundled with Atmel's AT86RF233, a 2.4GHz IEEE802.15.4 compatible radio. For programming the MCU comes with 32Kb of RAM and 256Kb of flash memory. The SAMR21 SoC includes an on-chip AT86RF233 radio. It is internally connected via SPI and some GPIO pins and behaves in exact the same way as externally connected SPI devices. Atmel Route Under MAC (RUM) supports both IPv6 and 6LoWPAN stack solution for the low cost applications.

Route Under Mac (RUM) is a small 802.15.4 protocol developed by Atmel. This protocol routes packets at the MAC layer, as opposed to the application or IPv6 layer.The main components of the stack include RUM, and IPv6 / 6LoWPAN. The stack has following features like build code was very minimal 6KB on AVR end node. Self-forming network, during power up find a network, and associate to it. Self-healing network, Nodes re-associate upon a failure to communicate. Multi-hop routing, Nodes can hop from the coordinator to another controller.

The main advantages of 6LoWPAN being an IP is that it doesn't require additional software and uses a widely known L3 protocol. The only addition is that between a 6LoWPAN and an IPv6 network there is an edge router (ER) that performs the header compression,
Kaa is a open-source middleware platform connected with various IoT devices. Kaa provides SDKs in C, C++, and Java languages. These SDKs are integrated with a number of popular platforms, such as: Android, iOS, Intel Edison, Raspberry PI, BeagleBone Black, Econais EC19D, STM32 based microcontrollers. KAA client application connected with KAA server. There are two methods of assigning a transport channel to a service. The first is to assign a separate transport channel to each service. The other method is to group all or some of the services and assign them a common transport channel. However, it is important to remember that one channel can maintain only one open session at a time.

Each transport channel is capable of transferring data in one of the following modes.

- From the endpoint to the server (upstream)
- From the server to the endpoint (downstream)

## II. NETWORK FORMATION

All nodes are connected via the tapbr0 bridge;each node is reachable within a single hop. Therefore, it is not possible to test real multi-hop communication with this method. Riot-networkscreates native nodes it can be deployed with in a network topology using tap interfaces. Atmel SAM R21 create setup a 6LoWPAN border router on RIOT, the board has capable of IPv6 network interface, also connection over serial interface to Linux Host and use SLIP interface.In RIOT, tunslip supports 6LoWPAN border router over SLIP runs on the Linux host. this tool will be compile for Linux host and wait message from board.

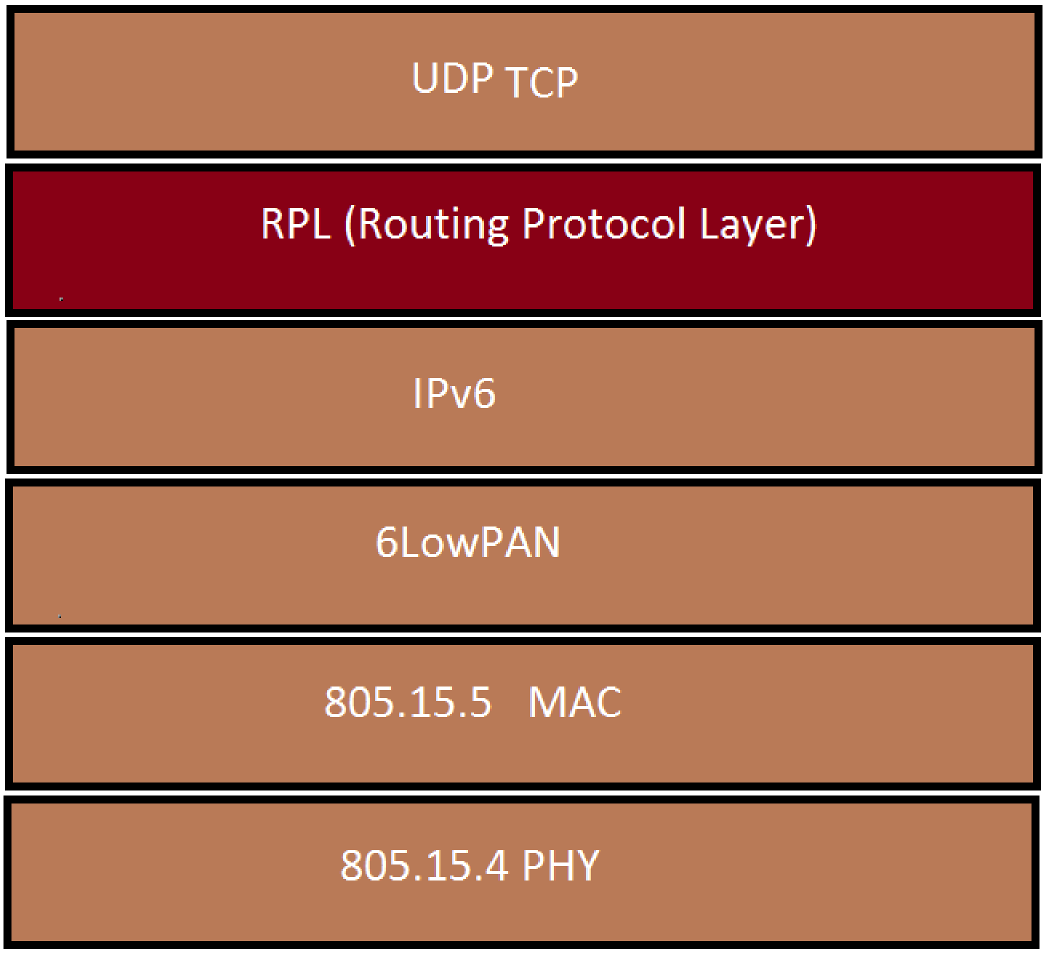


Figure:2 RIOT Networking Stack

Major requirement in Wireless IoT communication is power efficiency, reliability and Internet connectivity. Low power efficient IEEE 802.15.PHY is reliable, support for application & transport layer.6LoWPAN is an acronym for IPv6 over low power wireless personal area networks. Encapsulation and compression mechanisms on IPv6 makes, the IP protocol suitable for constrained devices and known as 6LoWPAN.LoWPANs are characterized by encapsulation and header compression that allows IPv6 packets to be communicated on the 802.15.4 physical layer.

Xplained Pro Kits are low cost MCU for developing IoT product, it has 32-bit ARM Cortex-M0 based and an 8 bit AVR core microcontroller with class 2.4 GHz RF transceiver. It equipped with two antennas and integrated ceramic antenna, the SMA connector.Two antennas has Jumper use to switch between them.Hardware abstraction layer handles both reception and transmission of IEEE 802.15.4 standard compliant data frames, allowing low energy communications between the radio nodes involved into the range measurement.Advanced timer system and hardware abstraction allows platform independent kernel and higher layers' implementations. RIOT also supports software suites with or without dynamic memory management,the kernel itself uses only static memory allocation

## III. Hardware Implementation

In this experiment we are creating small network with 2 samr21 boards, one board (tap0) used borderrouter connected to with Linux PC via SLIP (using second serial connection). Enable the gnrc_border_router with CoAP servers (usingmicrocoap_server example).

Step:1 Create a TAP interface, it will create network connect tothe tap0 interface
sudo ip tuntap add tap0 mode tap user
sudo ip link set tap0 up

Step:2 check the connectivity between RIOT and Linux HOST system. RIOT always transmit and receive in tap0 interface.
#ifconfig Iface 7 HWaddr: ce:f5:e1:c5:f7:5a
inet6 addr: ff02::1/128 scope: local [multicast]
inet6 addr: fe80::ccf5:e1ff:fec5:f75a/64 scope: local
inet6 addr: ff02::1:ffc5:f75a/128 scope: local [multicast]

Step:3 Connect the Linux Host to RIOT via UDP server. The 6 number is IPv6 used in netcat connection with more verbose infromation
nc -6uv fe80::ccf5:e1ff:fec5:f75a%tap0 8808

Step:4
For now, I succeeded creating a RPL network with the BR and the twoboards (ping6 is working on all boards, the BR as the root). I followedthe steps in [1] to configure RPL and [2] for the SLIP interface. I alsoadded a route:

Step:5: provide UART interface of the Board, and configure IPv6 address. This will allow your network from outside world.
:IOT/RPL/RIOT/dist/tools/tunslip$ sudo ./tunslip6 affe::1/64 -t tun0 -s /dev/ttyUSB0
[sudo] password for abcde:
********SLIP started on ``/dev/ttyUSB0"
opened tun device ``/dev/tun0"
ifconfig tun0 inet `hostname` up
ifconfig tun0 add affe::1/64
ifconfig tun0 add fe80::0:0:0:1/64
ifconfig tun0

```
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:127.0.1.1  P-t-P:127.0.1.1  Mask:255.255.255.255
          inet6 addr: fe80::1/64 Scope:Link
          inet6 addr: affe::1/64 Scope:Global
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

## IV. CONCLUSION

As a highly customizable and scalable platform, RIOT also provides you with a feature-rich toolset for building your own IoT applications for sport & fitness solutions of any scale - ranging from activity tracker apps to infrastructure management systems for sports venues.Improved efficiency, security and manageability for devices using a standards-based and IoT-optimized approach. Reduced complexity of integrating IoT capabilities into existing cloud services. Unified device and application data management using the same technology foundation and eliminating the need for duplicate infrastructure for clients and servers. Access to a large unified market of devices through open standards and the widely deployed ARM partner ecosystem.

## References

[1] http://electronicdesign.com/iot/understanding-protocols-behind-internet-things
[2] https://www.mbed.com/en/development/cloud/mbed-device-server/
[3] https://hal.inria.fr/hal-00768685v1/document
[4] https://github.com/RIOT-OS/RIOT/wiki/Introduction

## Authors

**First Author** – Saminath.V, M.E (Drives and Embedded controls), Accenture and saminathmail@gmail.com.