**IOSR Journal of Engineering (IOSRJEN)** **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Volume 10, PP 53-56

# Design of a Mini Robot Using Iot Through Ipv6 End To End Connectivity & High-Level System Design Using Wireless Sensor Networks

Dr. Bhavin Shah[1], Prof. Nilesh Gode[2], Prof. Mohan Kumar[3]
[1](Department Of Electronics & Telecommunication Engineering, Atharva College Of Engineering, India)
[2](Department Of Electronics & Telecommunication Engineering, Atharva College Of Engineering, India)
[3](Department Of Electronics & Telecommunication Engineering, Atharva College Of Engineering, India)

***Abstract :*** *The Proposed Paper Presents A Mini Robot Using Iot Through IPV6 End To End Connectivity, Assembled From Low-Cost Hardware Parts, And Open-Source Software Building Blocks. These Building Blocks Provide A Reusable & Extensible Base For Emerging Applications Mixing Robotics With The Internet Of Things And High-Level System Design Using Wireless Sensor Networks & Programming For Handling Traffic Using NS2.*
***Keywords -*** *Iot, Robotics, Cloud Computing, Wireless Sensor Networks*

## I. INTRODUCTION

In The Recent Years, Wide Interest And Innovation In The Field Of Internet Of Things (Iot) Have Been Observed, Triggered By Technological Advances In Embedded Systems Hardware, Software, And Connectivity. The Increasing Availability Of Compact, Cheap, Power-Efficient Micro-Controllers And Peripherals Has Spun A New Category Of Computers: Low-End Iot Devices. Even Though Such Devices Cannot Run Traditional Operating Systems (E.G. Linux And Equivalents) Due To Very Constrained Memory, CPU, Power Resources, Most Low-End Iot Devices Have Enough Resources To Run Newer Operating Systems [4] And Cross-Platform Application Code. Furthermore, Recent Network Technology And Protocol Standardization Efforts Have Enabled New Interconnection Capabilities For Such Devices, Such As Low-Power, End-To-End Ipv6 Based Networking. Simultaneously, Robotics Is Experiencing A Dramatic Growth, Not Only In Their Traditional Applications, Such As Industrial Automation, But Also In Other Domains Such As Self Driving Cars, And Personal Robots Such As Drones, Vacuum Cleaning Robots, And Other Types. The Smaller Robots Are Targeted By The Field Of Nano Robotics, Another Class Of Robots (And Applications) Is Expected To Consist Of Mini-Robots [10] Approximately Of The Size And Computing Capabilities Of Current Iot Devices. Mini-Robots Are Expected To Become Commodity And 1000 Times Cheaper Than Currently Available Robots [7]. Leveraging A Number Of Emerging Techniques, Such As 3D Printed Robots [8], And Network Connectivity Enabling New Paradigms Ranging From Fog Computing [3] To Cloud Robotics [6], Such Robots Are Likely To Be Massively Deployed In A Variety Of Application Domains In The Near Future. The Encounter Of Iot And Robotics Thus Promises To Open A Fascinating New Field [13] [14].

## II. IOT IN ROBOTICS

An Emerging Class Of Mini-Robots Will Inherit From The Same Constraints As Current Iot Devices (E.G. Actuators) Has Very Limited Memory, Finite Processing Power, And Strong Energy Limitations. In The Proposed Paper, We Focus On Three Important Aspects In Iot Robotics: Hardware Aspects, Software Aspects, And Network Aspects.

**Hardware Aspects:** From A Hardware Perspective, A Robot Consists In
(i) Structural And Mechanical Components, E.G. Carcass, Frame, Wheels
(ii) Sensor And Actuators, E.G. Motors, Distance Sensors
(iii) Computational Elements And Electronics, E.G. Micro-Controllers, Motor Controllers, And
(iv) Power Supply, E.G. Batteries

Recently, The Rise Of Open Source Hardware And The Maker Scene Lead To Increased Availability And Such A Significant Price Drop For These Components That Mini-Robots Under Rs. 1,000 To Rs. 3,000 Are Becoming A Reality [7]. Popular Examples Of Structural Components Include Mechanical Parts, Chassis, While 3D Printers Allow Virtually Anyone To Conveniently Create Custom Parts With A High Precision. The Arduino Community Lead To The Availability Of A Wide Range Of Affordable Sensors, From Inertial

Measurement Units To Full-Blown Laser Distance Scanners, And A Variety Of Actuators Became Available Due To The Scale Modeling Community. The Market Around Low-Power, Low-Cost Micro-Controllers Are Currently Booming. Basing Robots On These Low-Power Platforms Conveniently Allows The Use Of Standard Off-The-Shelf Batteries, Or, In Intermittent Activity Scenarios, Of Small Solar Panels And Other Means Of Energy Harvesting Solutions [13] [14].

**Software Perspective:** The Software Running On Iot Mini-Robots Consists In
(i) Hardware Abstraction And Device Drivers
(ii) Control Software
(iii) Communication Software, And
(Iv) A System Layer That Glues Together All These Elements

The Most Popular Software Base For Robots Is The Robot Operating System (ROS) [9] A Set Of Libraries And Tools Running On Top Of A Host Operating System (I.E. A Traditional OS Such As Linux, Windows). ROS Is Thus Not Intended To Run On Mini-Robot Hardware, Whose Constrained Resources (Memory, CPU, Power) Won't Match Traditional OS Resource Requirements. Instead, Newer And More Compact Operating Systems [4] Must Be Used As Base On Such Hardware. For Instance, RIOT [2] Provides Real-Time Capabilities, Hardware Abstraction, Multi-Threading, And Full Ipv6 Networking While Fitting The Tight Memory Constraints Of Micro-Controllers Typically
Found On Low-End Iot Devices. However, Contrary To ROS, RIOT Does Not Provide Specific Libraries Targeting

Robotics. The Shortcoming Could Be Overcome By Using Light Robotics Libraries [1] To RIOT Using BSD Socket Or POSIX [13] [14].

**Network Challenges:** On The Network Side, Iot Mini Robots Need
(i) Enhanced Algorithms And Protocols, And
(ii) Novel/Holistic Network Architectures

Afore Mentioned Constraints On Software And Hardware Translate Into Challenges For Network Technologies, Which Are Expected To Operate With Low Memory Foot-Print, Low Energy Consumption, And High Reliability Over Wireless, And To Interoperate With The Internet. For Instance, The IETF Is Currently Standardizing The Use Of Ipv6 With Protocols As 6lowpan, RPL, Coap, Over Low-Power Wireless Link Layers In Iot, E.G. BLE, Or IEEE 802.15.4 Using TDMA And Frequency Hopping To Increase Reliability. But These Are Not Designed To Accommodate Mobility, Temporal Loss Of Connectivity And Topological Changes, In Addition To The Classical Radio Interference, Multipath Fading: They Should Be Extended And Adapted [11]. Also, Robotics Using Iot Combines Embedded System Logic And Includes Complexity Of Some Tasks Iot Robot May Have To Carry Out For E.G. Grasping An Unknown Objec; Thus, It Will Be Necessary To Deport Some Of The Logic And/Or Processing For Robot Control To Remote Servers I.E., The Cloud. Elements Of Such Architecture Already Exist [5]. But Convergence/Adaptation Is Needed Between Such Elements And Standard Iot Protocols In The Making, Such As CBOR, COAP, MQTT, Or ICN [13][14]. The Goal Being To Provide Fully Integrated Communication Architecture, From Iot Mini-Robots Up To The Cloud.

## III. TECHNICAL DESCRIPTION

The Proposed Paper Presents Illustrates A High-Level Architecture As Shown In Fig. 1. The Figure Illustrates Assembling Low-Cost, Off-The-Shelf Components Including A Low-Power MCU, DC Drive Motor, Power Stage, Steering Server, And Ultrasonic Distance Sensor. The Behavior Of The Mini-Robot Will Be
(i) Reprogrammable On The Fly From The Cloud
(ii) Simultaneously Subject To Local And Cloud-Based Control Loops

For Local Control The Mini-Robot Will Run RIOT [13] [14], An Open Source Real-Time Operating System Which Fits Resource Constrained And Low-Cost Micro-Controller Platforms. For Communication With The Cloud, The Mini-Robot Will Combine Low-Power Wireless And IP Protocols, Providing End To End Connectivity Via Standard Access Points, Using RIOT's Default Network Stack. The Cloud Component Consists In A Simple REST-Based Daemon Using Coap To Communicate With The Robot, And The Web For User Interaction (E.G. Changing The Robot's Behavior). For Cloud Based Connectivity Using Other Devices & High Level Programming Using NS2, The Architecture Is As Shown In Fig. 2. The Code To Handle Traffic Is Also Illustrated Below.

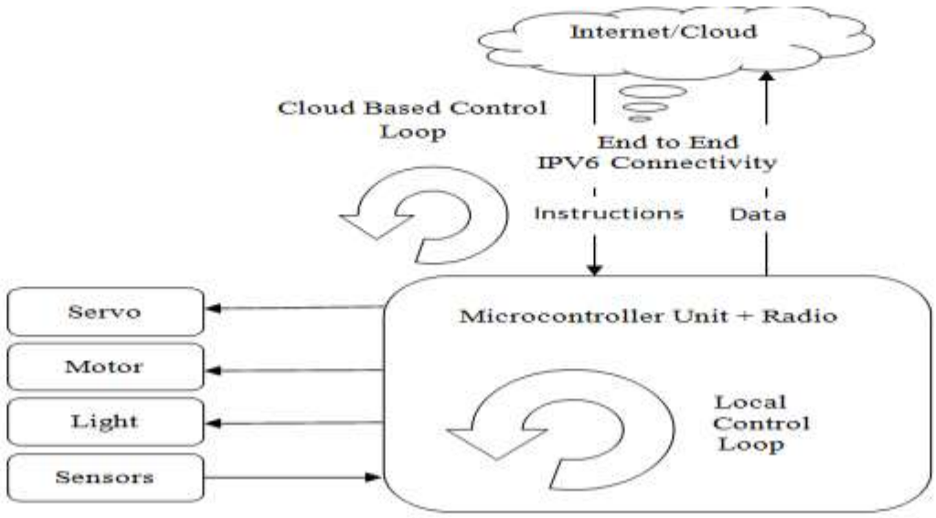


**Fig. 1: High-Level Architecture Using Low-Power Wireless, Cloud-Controlled & End To End Ipv6 Connectivity**

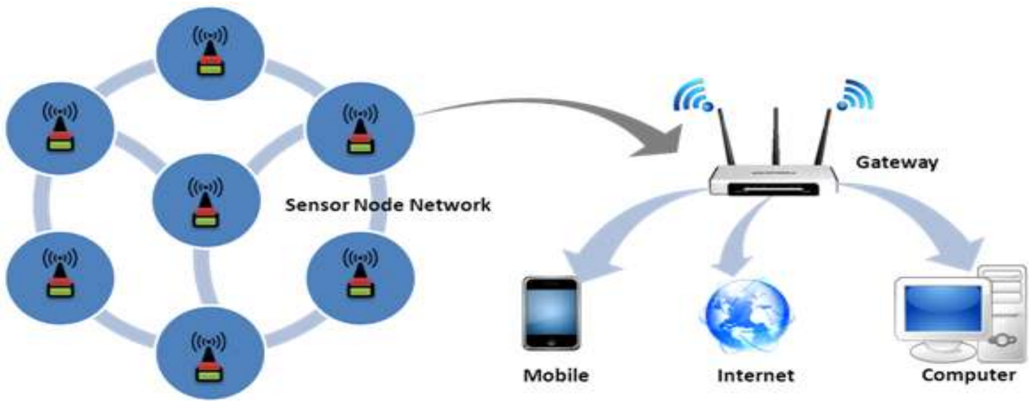


**Fig. 2: Wireless Sensor Network**

**Code For Handling Traffic In Wireless Sensor Networks:**

```
proc poissontraffic { src dst interval starttime } {
global ns_ node_
set udp($src) [new Agent/UDP] eval $ns_ attach-agent \$node_($src) \$udp($src)
set null($dst) [new Agent/Null] eval $ns_ attach-agent \$node_($dst) \$null($dst)
set expl($src) [new Application/Traffic/Exponential] eval \$expl($src) set packetSize_ 70
eval \$expl($src) set burst_time_ 0
eval \$expl($src) set idle_time_ [expr
$interval*1000.0-70.0*8/250]ms ;# idle_time + pkt_tx_time = interval
eval \$expl($src) set rate_ 250k
eval \$expl($src) attach-agent \$udp($src)
eval $ns_ connect \$udp($src) \$null($dst)
$ns_ at $starttime "$expl($src) start"
}
if { ("$val(traffic)" == "cbr") || ("$val(traffic)" == "poisson") } {
puts "\nTraffic: $val(traffic)"
#Mac/802_15_4 wpanCmd ack4data on
puts [format "Acknowledgement for data: %s" [Mac/802_15_4 wpanCmd
ack4data]] set lowSpeed 0.5ms
set highSpeed 1.5ms
Mac/802_15_4 wpanNam PlaybackRate $lowSpeed
$ns_ at [expr $appTime1+0.1] "Mac/802_15_4 wpanNam PlaybackRate
$highSpeed"
$ns_ at $appTime2 "Mac/802_15_4 wpanNam PlaybackRate $lowSpeed"
$ns_ at [expr $appTime2+0.1] "Mac/802_15_4 wpanNam PlaybackRate
```

```
$highSpeed”
$ns_ at $appTime3 “Mac/802_15_4 wpanNam PlaybackRate $lowSpeed”
$ns_ at [expr $appTime3+0.1] “Mac/802_15_4 wpanNam PlaybackRate
$highSpeed”
eval $val(traffic)traffic 19 6 0.2 $appTime1
eval $val(traffic)traffic 10 4 0.2 $appTime2
eval $val(traffic)traffic 3 2 0.2 $appTime3
Mac/802_15_4 wpanNam FlowClr -p AODV -c tomato
Mac/802_15_4 wpanNam FlowClr -p ARP -c green
if { “$val(traffic)” == “cbr” } {
set pktType cbr
} else {
set pktType exp
} [12]
```

## IV. Conclusion & Future Work

The Proposed Paper Presents Decomposed Clean & Separated Building Blocks, Using Open-Source Software And Off-The Shelf Hardware & Also Its Connectivity With High Level Devices Using Wireless Sensor Networks. Care Was Taken To Make It Straight Forward To

(i) Substitute The Local Real-Time Control Loop On The Mini-Robot With More Advanced Motor Control, Local Sensor Data Fusion And Short-Term Decision Making

(ii) Relocate And/Or Enhance The Cloud-Based Control Loop With More Advanced Computational Offloading For Sensor Data Processing, Remote Decision Making, And Mid- To Long-Term Planning,

Thus, The Proposed Work Can Be Extended For A Wide Range Of Emerging Applications Using Iot And Robotics. The Future Work Will Focus On Efficient Portable Software Running On Iot Mini Robots, Mixing Of High Level Devices With Low Level Devices Using Wireless Sensor Networks, Computation Offloading Schemes, And Optimizing Standard Iot Protocols For Reliability In Face Of Mobility And Multi Hop Over Low-Power Wireless.

## References

**Journal Papers:**

[1] Aversive C++ Library: Http://Aversiveplusplus.Com.
[2] E. Baccelli Et Al. RIOT OS: Towards An OS For The Internet Of Things, IEEE INFOCOM, 2013.
[3] F. Bonomi Et Al. Fog Computing And Its Role In The Internet Of Things, In ACM Mobile Cloud Computing Workshop, 2012.
[4] O. Hahm Et Al. Operating Systems For Low-End Devices In The Internet Of Things: A Survey. IEEE Internet Of Things Journal, 2016.
[5] A. S. Huang Et Al. LCM: Lightweight Communications And Marshalling, In IEEE IROS, 2010.
[6] B. Kehoe Et Al. A Survey Of Research On Cloud Robotics And Automation. IEEE Trans On Automation Science And Engineering, 2015.
[7] G. A. Korash Et Al. African Robotics Network And The 10 Dollar Robot Design Challenge, IEEE Mag. On Robotics & Automation, 2013.
[8] M. Lapeyre Et Al. Poppy: Open Source 3D-Printed Robot For Experiments In Developmental Robotics. In IEEE ICDL, 2014.
[9] M. Quigley Et Al. ROS: An Open-Source Robot Operating System In ICRA Workshop On Open-Source Software, 2009.
[10] M. Rubenstein Et Al. Programmable Self-Assembly In A Thousand- Robot Swarm. Science, 2014.
[11] A. Tinka Et Al. A Decentralized Scheduling Algorithm For Time Synchronized Channel Hopping In Ad Hoc Networks, Springer. 2010.
[12] NS2 Library : Https://Ns2projects.Org/Ns2-Simulation-Code-For-Wireless-Sensor-Network/
[13] Hauke Petersen, Et Al. Demo: Iot Meets Robotics - First Steps, RIOT Car, And Perspectives. ACM International Conference On Embedded Wireless Systems And Networks (EWSN), Feb 2016, Graz, Austria. 2016.
[14] Dinesh Reddy, Et Al. Iot Meets Robotics - First Steps, RIOT Car, And Perspectives, IJSER, Volume 8, Issue 4, April 2017.