



Cloud Based Electrical Device Control Middleware APIs for IoT

Prevesh Kumar Bishnoi¹, Dr. Dharmender Kumar²

¹Assistant Professor Computer Science and Engineering, SET Mody University, India, erpkbishnoi@gmail.com

²Professor Department of Computer Science and Engineering, GJUS&T Hisar, ²dharm_india@yahoo.com

ABSTRACT

Lots of programming is to be made to control the electrical devices from the web or Internet, due to it the lots of time is consumed for such programming. So in this research, framework to control the electrical devices like stepper and servo motor was developed. All APIs developed in Java and based on the proposed algorithm. Finally all APIs were tested and evaluated for the use in application programming. In this research, I will be focusing to control all the electrical devices via cloud. In electrical machines, there are two important devices, those are very popular in IOT, robotics and electrical engineering. Such devices are Stepper motor and Servo Motor. Finally a framework of API has been developed for such devices. This framework has electrical device control methods in Java with standard telemetry parameters.

Key words – IoT, Middleware, Electrical, APIs, Firmata

1. INTRODUCTION

A stepper motor is a direct current electric motor that splits the complete rotation into many stages. Then the motor rotator can move a least angle step at a time. According to different applications the different speed, torque motors are designed, developed and available for use. Controlling it by computation algorithm/program or computer has many advantages like interactive and automatic control. Generated pulsed are applied to the motor drive to control the motor rotation. Stepper motor is available in mainly three categories permanent magnet, hybrid synchronous stepper and variable reluctance stepper. All necessary parameters are written on the motor plate. On the bases of that it can be utilized in many different applications. The rated voltage will produce rated curved current in DC: but this is mostly a waste rating, because all modern drives are limited current and the drive voltage motor is much higher than the rated voltage. Motor torque can be changed by change the input current. That can be controlled by motor drive, microcontroller and APIs and user interface.

All necessary calculations can be made on the bases of the input required, sensor feedback and data analyses etc. If the motor is be controlled remotely, short distance controlling techniques are available like radio wave, infrared etc. but all for the short distance controlled only. To control the motor at

worldwide level a WWW network is required. This network is I-Way/ Information superhighway or Internet to day. To do it server side play important role. It can be done in PHP, ASP.Net or Advanced Java. But direct API are not available. So finally in this research a framework has been developed with proper algorithm design. The basic concept of the research is making the connection of motor with motor drive like L293D and then drive to microcontroller as standard practice. In the same series of development the framework also has the APIs to control the Servo motors. It is a direct current that is gear based that is equipped with multiple gears and works on servo mechanism. It moves very specific angle. The servo mechanism indicates about negative sensing response that is applied to improve the accuracy and performance enhancement of devices. Servo is used for exact position in Robotic Weapons, Legs, RC Airplanes, and Helicopters etc. Rotational nature of Servo motors not like other direct current motors. Shaft of servo motor is not rotating continuously. Servomotor makes the rotation between any two specified angles. That can be made via cloud using suggested APIs of the framework. Servo motor can be control via APIs between two angles like 90 to 180 and back. APIs are carriage command to the microcontroller and microcontroller to the motor drive or equivalent arrangement to drive the motor. Mainly it is the process of varying the voltage and current to control the motor for speed, torque etc. To design and develop of convert a servo motor, a DC motor with potentiometer and a control circuit is required. One microcontroller Node MCU will connect to motor in standard fashion. Interaction of a Microcontroller and motor is shown in Figure 1 via a drive.

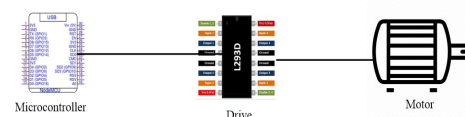


Figure 1: Microcontroller to Electrical Device

Node MCU will be connected to motor drive as shown in above diagram. On the other side it will be connected node MCU to Wi-Fi and Internet or intranet via 802. 11 network protocol. Figure 2 shows the detailed interaction of user and Electrical device via Internet and other necessary components.

One API on server will read the state of electrical device and store it one central database with transparent information of database connectivity and middleware like JDBC/ ODBC / JDBC-ODBC bridge. Other API will update the status of the motor for electrical device as per the action of the user. Now node MCU will be requesting the status for user command and act accordingly

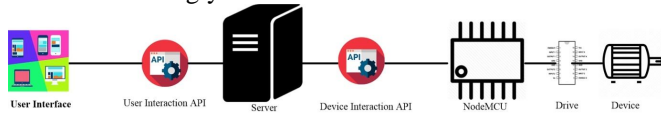


Figure 2: Access to Electrical Device Via Internet

For the connectivity, HTTP protocol with firmata over wireless TCP-IP is used. The necessary modification of firmata with library of stepper motor and Servo Motor has been made.

2. REVIEW

W N Hussein, et al. published a paper entitled A Methodology for Big Data Analytics and IoT-Oriented Transportation System for future implementation. This research suggests a model to apply of Internet of Things and Big-Data in transport system for the customer response. Finally the author suggested a process flow to apply the IoT and Big-Data to improve the customer satisfaction. The result of this research will be the integrated implementation framework for BDA and Internet of Things based transportation systems to achieve customer satisfaction.

[25] Mohammed, Faiz, Ghufan and Mahameda proposed a process in their research. In this process, mostly the roof of buildings are spotted in a satellite pictures on the bases of color of the roof, and a search decomposition method is applied to arear after division in the grids. The designated track is directed to the independent robot via Wi-Fi communication as a sequence of points demarcated for actual application in the external environment. The movable robot makes its way to every point in the path using the Global Positioning System and digital scope. It uses the five infrared sensors to avoid and find any unwanted object in the path. To position the robot to move at a straight line and to reach to the target with minimum error or with full accuracy a proportional-integration-derivative controller technique is used. Finally after getting the findings, the path of the moveable robot without the PID controller strayed from the designated path by more than two meter, but the after using PID it has been reduced to one meter only.. The moveable robot reached fewer than three meters from the concluding goal, an error for GPS correctness. [23] Mwila M. K. and others researchers proposed in a paper on designing and applications of a Node.js framework based communication framework for an man less independent ground automobile. In this research they used a approach to develop a communication platform and improves interoperability between diverse types of components irrespective of the producers and of the software platform. It was deployed on credit card computer such as Raspberry Pi and BeagleBone Black to access and connecting several sensors on the platform and to control hardware such as stepper motors. Because Pi and BeagleBone both are linux

base platform so the messaging design can also be used for Windows operating systems or traditional laptops have Linux that run algorithms for localization, terrain mapping, and track planning. Initially considered as a few selected languages, the true nature and power of JavaScript have only newly been deeply appreciated, with a major move underway to apply this language in new and engaging contexts. Finally the researchers proved, "The decisive goal of the framework was the construction of communication and inter-operation of UGV constituents and a sensor suite within a network. The framework was useful to G-Bat, a UGV platform developed at CSIR DPSS Landward Sciences to test and simulate the communication part of an independent direction finding system. The test was a successful step that covered the way for more robust implementation of the framework in future works". [4]. EC Nelson and two other researchers concluded in their research related to embedded architectural framework for communication between automobiles and user devices. The last few decades have seen quick development in the amount and importance of software in vehicle industry. Such type of development is induced the modification to over all development that software as a part of hardware to control the application of a set of services, APIs etc. that arbitrate between hardware and user-observed functionality. Finally the researchers found and designed, "An vehicle computing platform based on a software service platform to offer not only entertainment but information functions implemented in two model vehicles of Ford Motor Company". [7] Raharjo, Burns, Venugopalan and MD Wang abstracted in a research related to application and development of a easy to use interface and collaborating data collection system for cerebral palsy that the Cerebral palsy is a permanent motor problem that appears in early age and it requires multiple tests to assess the physical and mental competences of the patients. Present medical chronicles data collection systems, e.g., EPIC, applied for Cerebral palsy is very common, problematic to navigate, and there are few chances of error and mistakes. Data can not be extracted in a simple way, that bounds data analysis to this unlikely source of information. To overawed these confines, he intended and prototyped a database with a graphical user interface precisely towards clinical research in Cerebral palsy. Framework developed using oracle database MySQL and Java language is very reliable, secure, platform independent may be easily integrated with other programming languages for data analysis like MATLAB. All researchers Finally abstracted, "The developed a Graphical user interface design is a intuitive tool for data gathering and which can be applied to many different fields aside from Cerebral palsy to infer useful information out of the vast amount of data being collected". [10]. Simonds discussed in his paper that is to framework development of vehicle e.g. Software for the next-generation automobile. They reviewed that the automotive electronic, software and embedded systems are ever-changing very fast in terms of necessitating connectivity of all kinds. One Ford's company model in the area of APIs framework permits producers to modify and handle varied communication tools very fast. Developed

framework or system, the Vehicle Consumer Services Interface (VCSI provides the necessary level of security for mobile services and the ability to personalize and upgrade flexibly with a portability approach to device in term of connectivity. To communicate with portable devices and off-board systems the XML is being used for this purpose, by that the highest degree of freedom and levels of branding and personalization are achieved. Researchers said, “In a Extensive Markup Language based interface of Ford's proprietary that is VUML means vehicle user-interface mark-up language split up the MI from the core application. Using this, the developers uses many HMI component categories like from knobs, switches, and buttons for touch screen displays and a conversational voice-activation systems”. So the manufacturers may avail large number of skins and facilities in their final products by choosing the HMI category to enhance the brand qualities, look, vision and their customers' favorites. It is based on Java, so the developer and user will all advantages of the platform independence and portability of Java properties. With the CDC, the basis profile (one of the standard profiles for handheld devices using Java) allows a VCSI-based solution to achieve very well, providing all the anticipated roles and services. Ford has recognized a set of OS norms that emphasis on robustness and real-time performance specifications. A key parameter in the overall system is about the start and stop the driver and application without restart of Operating Systems. This is particularly very significant in a automobile because user requires the devices and off-board connectivity at different and changeable times. [13] P. Casals-Torrens worked on the topic 'Virtual Laboratory and review on the European Higher Education Area framework covers the acquisition of skills such as the ability to learn independently, that needs students to dedicate much of their time to individual level and group work to enhance and further accompaniment the information acquired in the classroom. To reinforce the findings obtained from classroom happenings, lecturers must create tools to encourage teaching learning process and facilitate independent learning. The objective of this research is doing the simulation and testing of electrical machines using virtual laboratories. This simulation was based on Java technology.[18]

3. PROPOSED ALGORITHM

Figure 3 is representing the electronic diagram of the whole experimental arrangement of realization of the algorithms and mathematical concept. At the lowest level the motor(either stepper or servo) will be available. Motors will be connected to the motor drive to control it. Motor drive will be getting the signal and power to run the motor in terms of command by microcontroller NodeMCU. NodeMCU or ESP8206 will communicate HTTP server and database on cloud via Internet/Intranet. To provide Internet to NodeMCU standard arrangement of router/ Switch or Access Point (AP) with ISP is made.

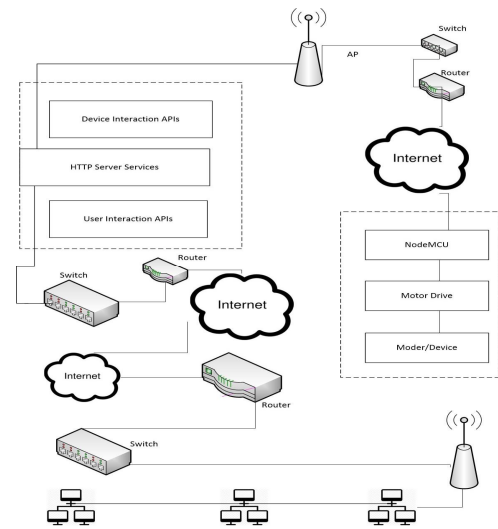


Figure 3: Electronic Diagram

One firmware will be continually observing the status of motor on off on cloud database with the help of Device Interaction API. As the status on the cloud is changed Node MCU accesses it and informs it to the motor. As per the Instructions of NodeMCU motor will act accordingly.

On the other hand to control the motor the client make connection to Internet/Intranet and access the HTTP server and database server on cloud to change the status of the motor, to switch on or switch off the motor or to change the speed of the motor or rotate it backward or forward. For this purpose the client may use the web application, desktop application or a mobile android application. But everyone will go through user interaction APIs framework.

By accessing User interaction APIs, a user can give command to motor by involving the Device Interaction APIs. By using user interaction APIs, an application program initiate the controlling of motor in different terms like stop, start, forward, backward or changing of speed of motor. Even a user can rotate the motor by any permitted desired angle. The user applications can be developed in any technology like android, java, or any other. Because the whole concept is Internet based, so the concept is ubiquitous i.e. available anywhere everywhere.

NodeMCU has a limitation of Ethernet so always wireless connection is be required. But wireless connectivity has lots of advantage over wired connectivity in the area of networking and Internet of Things, it provide the mobility to the device and system.

3.1 API to read from:

Specific API will be used to make the communication with motor of electrical machine. Such API are responsible to read and write to Node MCU. Node MCU is making physical communication with the electrical device directly or via motor drive. Mainly there are two prototypes of methods read and write with different arguments. Prototype and Class diagram of both class is and methods. Algorithm to read is given in Figure 4.

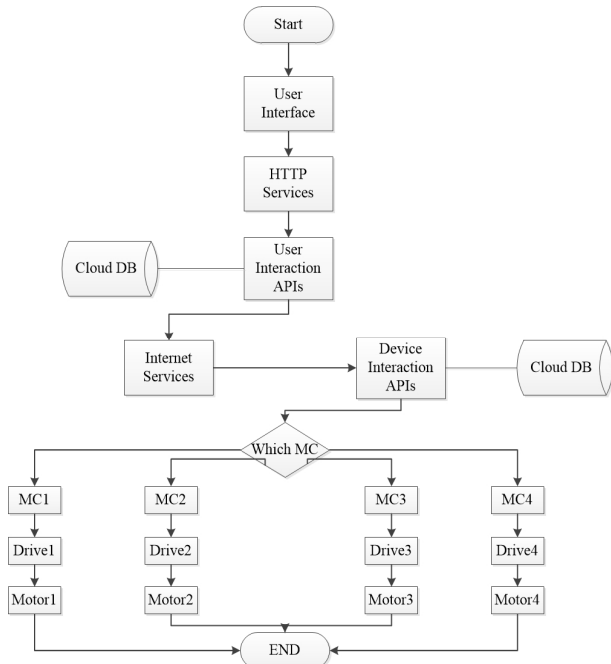


Figure 4: API to Read

L293D Necessary connection with the microcontroller is given in the Figure 5.

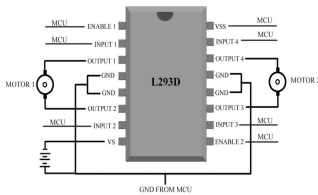


Figure 5: L293D Connection

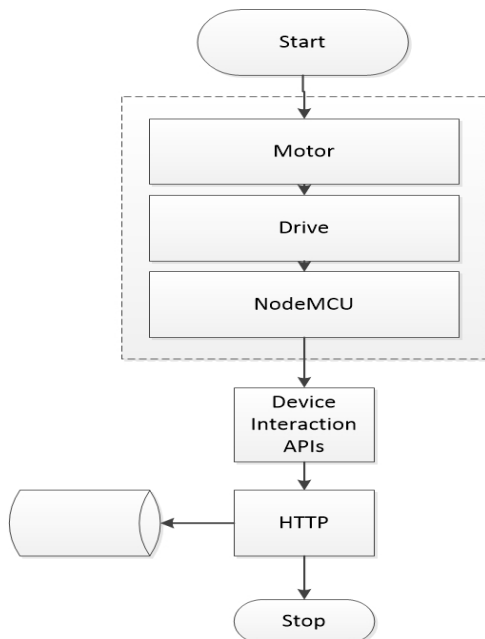


Figure 6: Algorithm to get command from user

3.2 API to get commands from user:

As we discussed one API is to make the connection with Node MCU and reading the status from Central database about the electrical device for example the device this on off or not connected. Exactly in the same way there is one method of the framework that is getting commands from user. This method will take the command from user to change the state of electrical devices and make necessary changes in Central database in fully authorized and authenticated way. This method of the framework is also having different signatures for prototype Figure 6 representing the algorithm for firing the command.

Node MCU is connected to Internet. And all the user interfaces and servers are also connected to Internet. All three different may be at different places in the world wherever Internet is available. Due to the availability of Internet all can communicate with each other. That is electrical devices can be controlled from any place in the world in ubiquitous way. The L293D is a classic motor driver or integrated circuit to moto control that allows DC motor to rotate in any desired direction like clockwise or anticlockwise. i.e. the stepper motor can be rotated any direction. Motor control is based on the principal of H Bridge. That is just like a manual commutator to change the direction of the current. Drive is used not only for stepper motor, some other types of motors like DC motor may also be controlled by this drive. Notion of H-bridge permits the circuit to stream the current in forward or backward direction and commutate it in the computational controlled way. As we know that the direction of voltage is to be changed to change the direction of the motor rotation via a H-Bridge integrated circuit. A single L293D integrated circuit includes double H-bridge circuits inside the it and capable to control two DC motors at a time. Figure 7 showing the pin diagram for it. Total number of pins in the drive is sixteen.

3.3 Framework Class diagram with Methods prototype:

As I discussed earlier also to design any component with or without logic the design or modelling is required to implement the concept. It is a UML class diagrams to implement the logic or communication model in the UML class diagram. As above, we discussed the there are two types of motors and each one will be having two category of APIs. One type of API will be the bridge between user and the Node MCU and other type will be Node MCU and the electrical device. Class diagram are given in following Figure 7.

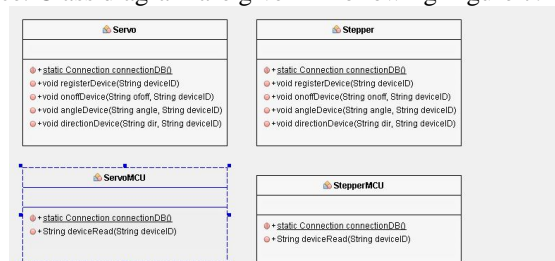


Figure 7: Class Diagram

4. RESULTS

4.1 Starting of Tomcat Telemetry

In my previous research related to it, I took the snapshot of when the tomcat webapplication server at startd. For proper comparison tomcat telemetry snap was taken before starting also. But in this one I took the snapshot, when Tomcat Server starting was in very beginning. So it can be seen very easily that every curve growing up in term of CPU Utilization, Memory Heap utilization, GC and Thread and class loaded in Figure 8. When the Tomcat server started the large number of classes, thread loaded and high memory heap consumed.

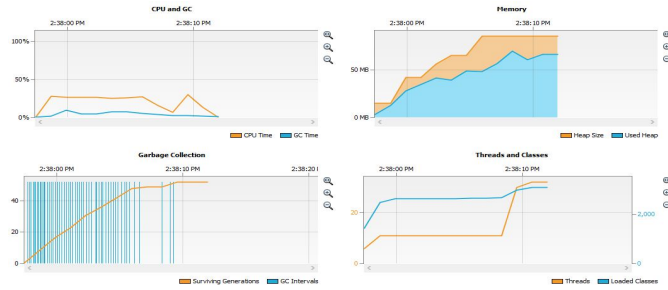


Figure 8: Tomcat Telemetry Class Loading

. Perpendicular lines in the GC curve/ graph shows the regular intervals of the Garbage Collection. Lines are very near to each other represents GC interval is very shorts. Because large number of classes objects are created and destroyed. In memory graph two different colour lines shows the available and used heap. But the used heap is always lesser than the available heap.

After saturation the Tomcat telemetry is given below in Figure 9.

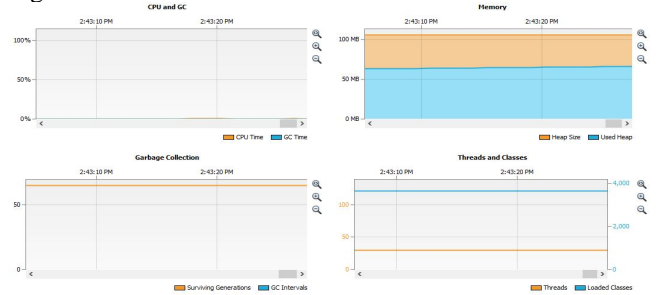


Figure 9: Tomcat Telemetry Class Loaded

4.2 Servo motor MCU device Read method Telemetry for 100 Number of threads: The following curve/ graph has been created for a hundred number of threads. There are four graphs CPU, GC, memory and thread and classes.

The CPU and GC you can see there is no blue line but approximately for few points, 1 percent to 2% or less than that. CPU utilization is being indicated this is between the time 2 2:44:20 and 2:44:30. As we discussed in previous graph interpretation also the height of the graph is showing the CPU utilization of the API. Length is showing the time to write that data to database. Figure 10 is created for writing the command to the cloud database from a user interactive panel. due to it, you are looking the curve is broken in between because first the http connection will be made and after firing

the command the connection way be broken and after that connection will be re-established and that data will be written and CPU utilization come down smoothly and become zero.

Similarly in memory you can see the biggest positive thing of the memory graph memory utilization is not increasing continuously. When the CPU utilization come up, heap used also come up. Now about the garbage collection, because there are very less number of objects are created due to it the garbage collection play less important role in this. But in thread and class graph we are seeing, there are two colored lines one is perfectly straight and parallel to the x-axis another one showing a page and this started exactly when the CPU utilization increased and the peak come down when the CPU utilization becomes zero. But why this peak is there. This peak is available here because we created 100 number of threads and everyone is trying to write to database for the purpose of testing.

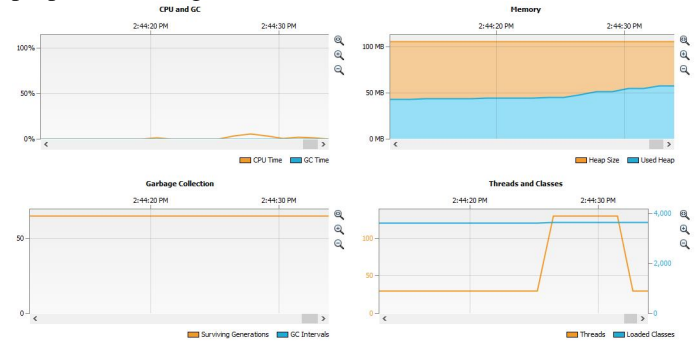


Figure 10: Tomcat Telemetry Read Method

4.3 Thread Creation and Timing:

The next diagram of the result is creation of number of threads and timing diagram. In Figure 11, there are four different possible colours for the thread states. The running thread is represented by green colour bar, sleeping by violet colour, waiting thread by yellow colour.

So at front of name of the thread a bra is representing the progress of the tread and time of progress with total time.

So in this diagram only three threads are running. But this diagram is only for starting of Tomcat.

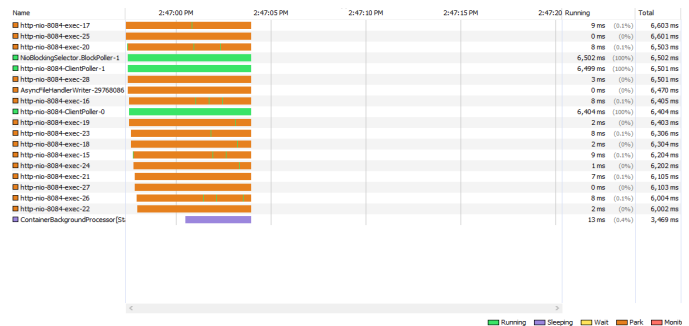


Figure 11: Thread Timing

4.4 Data base Query:

In the next Figure 12 the utilisation of CPU by the SQL Query is shown here. The JDBC connection is taking total time of execution 3889ms and utilisation of CPU is 0.1%.

Similarly, for other methods like angleUpdate, deviceConnect etc. are shown in Figure 12.

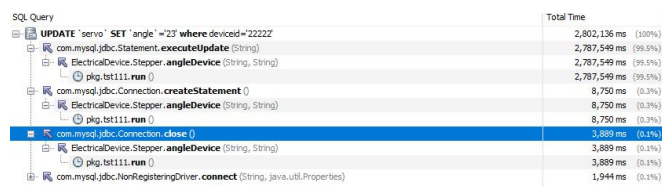


Figure 12: Database Query Execution

4.5 Stepper class for angle update method Telemetry for 100 Number of threads:

The Figure 13 snap shot taken for elementary 100 number of threads for stepper motor for specifically for a method angleUpdate.

It can be seen that the CPU utilisation is very less. But because hundred number of threads are there, due to this the CPU is being utilised less but for long time. Because 100 number of threads are needed and deleted stopped so GC timeline it is also above zero.

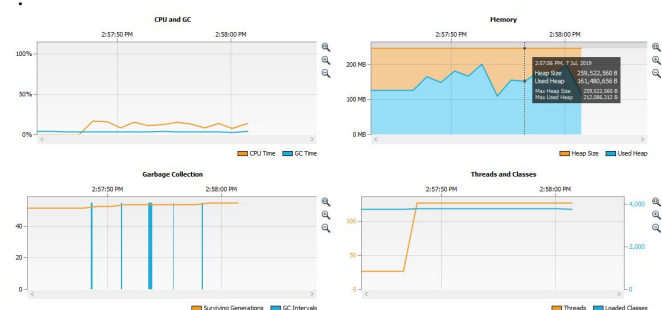


Figure 13: Tomcat Telemetry Angle Update

In the same figure you are seen so many pics in memory is utilization such peaks are available there, due to large number of thread creation and destroying. In garbage collection graph perpendicular lines are overlapping each other. This is also due to large number of thread working. In Thread and class diagram you can see the number of threads created or loaded close the limit of loaded classes

5. CONCLUSION

Research on the middleware APIs development of electrical devices specifically stepper and servo motor concluded that it is very useful in the controlling the devices by computer, microcontroller and microprocessor. Use and take the control of devices by user intuitive interface. In the same way the developer can develop bigger applications in a very short time. The knowledge of electronics and electrical devices and engineering will be transparent for the developer. All APIs were tested for the best performance in terms of memory, time and load. All APIs were tested with the piolet project of standalone and web applications.

ACKNOWLEDGEMENT

Authors do thanks to the CSE Departments of Mody University and CPU University and GJUS&T Hisar for providing the research facilities, environment and necessary platform for this research.

REFERENCES

- Hung-Chieh Yu; Shu-Ming Tseng, **AFlexible Framework for Wireless-Based Intelligent Sensor with Reconfigurability , Dynamic adding , and Web interface**, Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'06), 2006
- S. M. Kalyankar; S. D. Sawarkar, **Design of multinode reconfigurable multiprocessor network for embedded systems**, 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDs), 2017
- E. Karbab; D. Djenouri; S. Boukaboul; A. Bagula, **Car park management with networked wireless sensors and active RFID**, 2015 IEEE International Conference on Electro/Information Technology (EIT), 2015
<https://doi.org/10.1109/EIT.2015.7293372>
- 35 M. K. Mwila; P. Mbeve, **Design and implementation of a Node.js based communication framework for an unmanned autonomous ground vehicle**, 2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech), 2017
- Meads A., Roughton A., Warren I., Weerasinghe T., **Mobile service provisioning middleware for multihomed devices**, 5th IEEE International Conference on Wireless and Mobile Computing Networking and Communication, WiMob 2009, 12 October 2009 through 14 October 2009, Marrakech
- Huang G., Wang R., Tang C., **Data aggregation algorithm based on mobile agents middleware in WSN**, ,
- 36E. C. Nelson; K. V. Prasad; V. Rasin; C. J. Simonds, **An embedded architectural framework for interaction between automobiles and consumer devices**, Proceedings. RTAS 2004. 10th IEEE Real-Time and Embedded Technology and Applications Symposium, 2004., 2004
- Lee, Dong-Kyu; Kim, Tae-Hyon; Jeong, Seol-Young & Kang, Soon-Ju, **A three-tier middleware architecture supporting bidirectional location tracking of numerous mobile nodes under legacy \WSN\ environment** ,

- Journal of Systems Architecture , 2011
- 9 Martin, Sergio; Diaz, Gabriel; Plaza, Inmaculada; Ruiz, Elena; Castro, Manuel & Peire, Juan, **State of the art of frameworks and middleware for facilitating mobile and ubiquitous learning development** , Journal of Systems and Software , 2011
<https://doi.org/10.1016/j.jss.2011.06.042>
 - 10 37I. Raharjo; T. G. Burns; J. Venugopalan; M. D. Wang, **Development of user-friendly and interactive data collection system for cerebral palsy**, 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), 2016
 - 11 M. Matijevic; V. Cvjetkovic, **Overview of architectures with Arduino boards as building blocks for data acquisition and control systems**, IEEE Conferences 2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV), 2016
 - 12 G. P. Silva; J. A. d. S. Borges, **A Didactic Processor and Simulator for IoT**, IEEE Conferences 2018 3rd International Conference of the Portuguese Society for Engineering Education (CISPEE), 2018
 - 13 38C. Simonds, **Software for the next-generation automobile**, IT Professional, 2003
 - 14 Gao J., Lan Y., Jin M., **A model of third-party integration testing process management for foundation software platform**, 2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008, 12 October 2008 through 14 October 2008, Dalian
 - 15 Houyou A.M., Stenzer A., De Meer H., **Performance evaluation of overlay-based range queries in mobile systems**, 4th International Workshop of the EuroNGI/EuroFGI Network of Excellence on Wireless and Mobility, 16 January 2008 through 18 January 2008, Barcelona
 - 16 Nickschas M., Brinkschulte U., **Guiding organic management in a service-oriented real-time middleware architecture**, 6th IFIP WG 10.2 International Workshop on Software Technologies for Embedded and Ubiquitous Systems, SEUS 2008, 1 October 2008 through 3 October 2008, Anacapri, Capri Island
 - 17 Paroux G., Demeure I., Reynaud L., **A power-aware middleware for mobile ad-hoc networks**, 8th International Conference on New Technologies in Distributed Systems, NOTERE 2008, 23 June 2008 through 27 June 2008, Lyon
 - 18 39P. Casals-Torrens, **Virtual Laboratory for Learning Asynchronous Motors in Engineering Degrees**, IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, 2013
 - 19 Chu W., Li W., **Study of integrating RFID middleware with enterprise applications based on SOA**, 2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008, 12 October 2008 through 14 October 2008, Dalian
<https://doi.org/10.1109/WiCom.2008.1631>
 - 20 Fangzhou Z., Yingying N., **Secure encapsulation of non-secure middleware**, 2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008, 12 October 2008 through 14 October 2008, Dalian
 - 21 Zhang N., Bao H., **Study on grid technology and its application**, 2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008, 12 October 2008 through 14 October 2008, Dalian
 - 22 Zhang Z., Li L., **Compositive test of a platform architecture for E-government**, 2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008, 12 October 2008 through 14 October 2008, Dalian
 - 23 Mohammed Z. Al-Faiz , Ghufuran E. Mahamed, **GPS-based Navigated Autonomous Robot**, 2015 International Journal of Emerging Trends in Engineering Research, Volume 3, No.4, April 2015
 - 24 Vijaykumar N Chalwa, Shilpa S Gundagi **Mechatronics Based Remote Controlled Agricultural Robot**, 2014 International Journal of Emerging Trends in Engineering Research, Volume 2, No.7, July 2014
 - 25 W N Hussein, L M Kamarudin, M R Hamzah, H N.Hussain and K J Jadaa, **A Methodology for Big Data Analytics and IoT-Oriented Transportation System for future implementation**, 2019 International Journal of Emerging Trends in Engineering Research, Volume 7, No.11, November 2019
<https://doi.org/10.30534/ijeter/2019/087112019>