

Power Generation Using LiveCode

Sanjana K, Sindhu G, Siri M, Sripriya R ,Dr.Nandakumar A N*

Department of CSE, GSSSIETW, Mysuru, Karnataka, India

DOI: <https://doi.org/10.21467/proceedings.1.4>

* Corresponding author email: nandakumar@gsss.edu.in

Abstract

Power distribution helps in proper planning for expansion of the electrical networks in order to face the load growth and to supply the consumers properly. The results show that the voltage profile is improved, and losses are reduced, when a distribution of proper size is incorporated at proper location in the system. Live Code is a development environment that integrates user-interface design with writing and testing program code. In this paper we are using live code to distribute the electricity through different channels depending on the live requirement and availability of resources.

Keywords—LiveCode, Internet of Things, Generic editor, Voltage profile.

1 Introduction

The objective of this paper is to distribute electricity in an efficient manner by supplying electricity based on requirements of particular locations and availability. The major advantage of this paper is that this distribution can be controlled from remote location. In this paper we have built a generic editor for this purpose, where we write compile and upload our Livecode to the cloud (file server). The Raspberry pi on the other hand will continuously poll the cloud for new update, if new update is found the raspberry pi will kill its currently running image of the LiveCode and start running the new update retrieved from the cloud.

2 Literature Survey

Live coding is dynamic way for writing code in real time. [1]. A Lesson in LiveCodeThe LiveCode Lessons Book is a collection of over 400 separate lessons on how to achieve specific tasks in LiveCode.How do I connect to a database, how do I skin a button, how do I get my app onto my device, what's the best way to show progress of a download,send an email or take a photo... so manyquestions answered. [2]. Designing the Internet of Thingsby Adrian McEwen, Hakim CassimallyWhether it's called physical computing, ubiquitous computing, or



© 2018 Copyright held by the author(s). Published by AIJR Publisher in Proceedings of the 3rd National Conference on Image Processing, Computing, Communication, Networking and Data Analytics (NCICCND 2018), April 28, 2018. This is an open access article under [Creative Commons Attribution-NonCommercial 4.0 International](https://creativecommons.org/licenses/by-nc/4.0/) (CC BY-NC 4.0) license, which permits any non-commercial use, distribution, adaptation, and reproduction in any medium, as long as the original work is properly cited. ISBN: 978-81-936820-0-5

the Internet of Things, it's a hot topic in technology: how to channel your inner Steve Jobs and successfully combine hardware, embedded software, web services, electronics, and cool design to create. Helps software engineers, web designers, product designers, and electronics engineers start designing products using the Internet-of-Things approach Explains how to combine sensors, servos, robotics, Arduino chips, and more with various networks or the Internet, to create interactive, cutting-edge devices Provides an overview of the necessary steps to take your idea from concept through production .If you'd like to design for the future, Designing the Internet of Things is a great place to start.[3]. Electric Power Distribution by A. S. Pabla The Electricity Sector is currently experiencing many changes -impact of high-end technologies, privatization of the power utilities, rising tariffs, power shortages, etc. The sector is reinventing itself to overcome these challenges and is anticipating growth with the institution of the electricity reforms and the entry of private companies [4]. Cloud Computing: Concepts by Zaigham Mahmood, Thomas Erl, Ricardo Puttini Clouds are distributed technology platforms that leverage sophisticated technology innovations to provide highly scalable and resilient environments that can be remotely utilized by organizations in a multitude of powerful ways. To successfully build upon, integrate with, or even create a cloud environment requires an understanding of its common inner mechanics, architectural layers, and models, as well as an understanding of the business and economic factors that result from the adoption and real-world use of cloud-based services.

3 Proposed System

- The aim is to build a power distribution system that can adapt to the changing environments and requirements.
- Based on the requirement we change the code in the generic editor ..
- In the generic editor, we type the program compile it and we debug all the error to generate class file.
- The class file is uploaded which is nothing but the live code
- To upload the class file, we first establish connection with File server in the cloud through FTP server by providing user authentication details
- At the raspberry pi end will be constantly polling the cloud for new class file, if found it kills the currently running image of the LiveCode and start running the new update retrieved from the cloud.

4 System Architecture

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. Our System has modules which have been listed below. Before that we have the system, architecture which will give the overall understanding how the system actually is implemented.

4.1 Data Flow Diagram

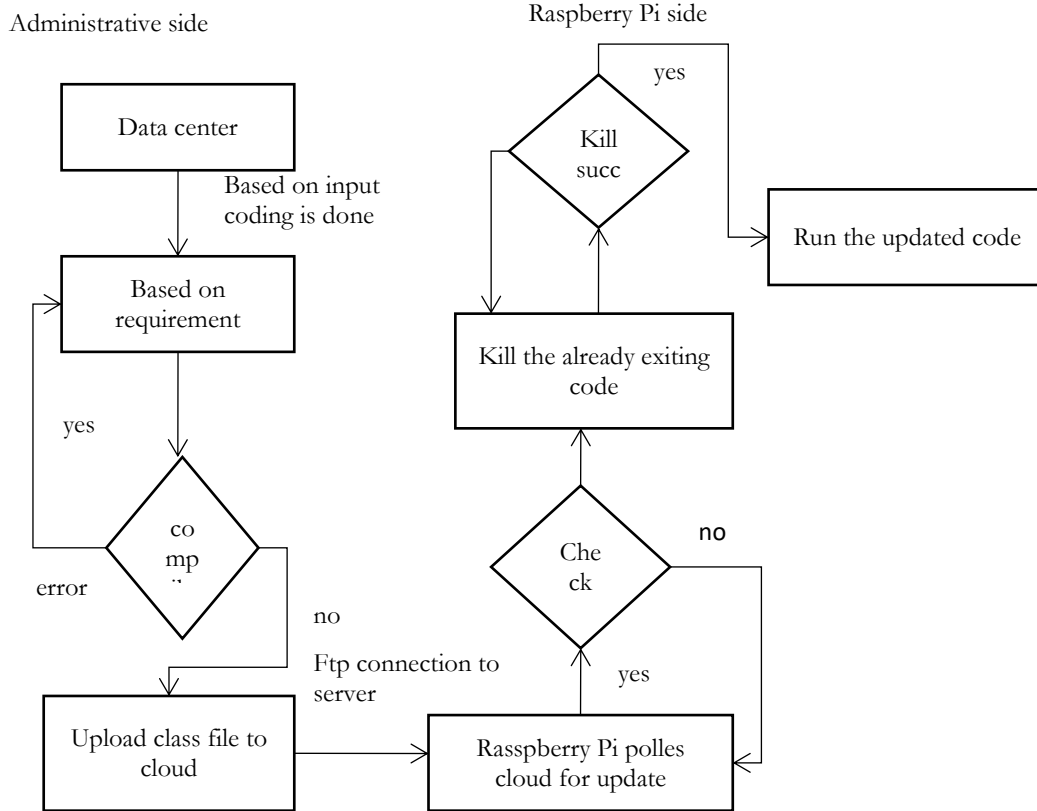


Fig.1. Data Flow Diagram

Based on the requirement we change the code in the generic editor ..

- In the generic editor, we type the program compile it and we debug all the error to generate class file.
- The class file is uploaded which is nothing but the live code
- To upload the class file, we first establish connection with File server in the cloud through FTP server by providing user authentication details
- At the raspberry pi end will be constantly polling the cloud for new class file, if found it kills the currently running image of the LiveCode and start running the new update retrieved from the cloud.
- The high-level design is given in Fig.2.

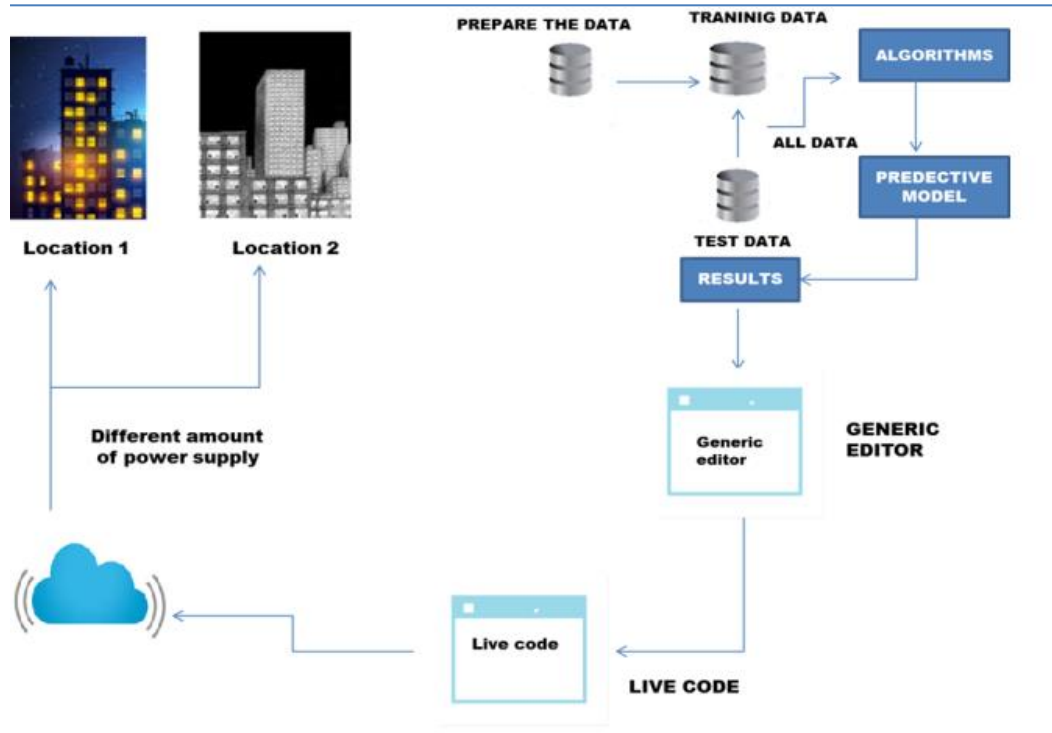


Fig.2. High Level Design

5 Implementation

The proposed method involves the following implementation modules:

- I. LiveCode
- II. Cloud
- III. Generic editor
- IV. Raspberry Pi

5.1 LiveCode:

Based on the requirements we change the image/rewrite the code already running on the raspberry pi . We compile this code in the generic editor, if there are no errors the 'class file' for the LiveCode is generated and it is ready to be run on the raspberry pi

5.2 Cloud:

From the generic editor the newly generated 'class file'Will be uploaded to the file server in the cloud first by establishing a connection with the 'FTP server' with proper authentication details of the user

5.3 Generic Editor.

The generic editor is built with functionalities required to write, compile, debug and upload the LiveCode for our paper

5.4 Raspberry Pi:

The Raspberry Pi will be constantly polling the ftp server for a new program parallel to running the already existing image of the LiveCode, if a new ‘.class file’ for a changed LiveCode is encountered it kills the current image of the LiveCode and executes the newly found code.

6 Algorithm

```

Livecode part:
Class IOTServer{
Public static void main(String... v){

//create FTP object
FTPClientftpc = null;

//get process Id of Iotserver program
Process myprocess=Runtime.getRuntime().exec("jps");
BufferedReaderinp =new BufferedReader(new
InputStreamReader(myprocess.getInputStream()));
String myid = " ";pr;
While((pr=inp.readLine())!=null){
If(pr.contains("IOTServer"))
Myid=pr.split(" ")[0].trim(); }

//create ftpc object for establishing connection
Ftpc = new FTPClient();

//connect to remote ftp server
Ftpc.connect("servername.com");
//login with username and password
Ftpc.login("username","password");
//change directory to our space on the server
Ftpc.changeDirectory("iotspace");
//get list of all class files in ftp server
FTPFILE[] all_files = ftpc.list();
Boolean conatins_new = false;

//check if all_files contain livecode.class or done.txt
For(FTPFILE ff : all_files){
If("done.txt", equals(ff.getName())){

```

```
Contains_new = true; }
}
//now download remote file and copy to local file
Ftpc.download("livecode.class",new File("livecode.class"));

//after retrieving all process id running in the system into java_process_id
//kill the old livecode that is running
If(!process.split(" ")[0].trim().equals(myid))
Runtime.getRuntime().exec("kill"+java_process_id);}
}

//finally close the ftp connection
Ftpc.disconnect(true);

}
}
```

7 Conclusion

In today's world electricity plays a major role. Storage and distribution must be done in an efficient way. Our paper concentrates on how distribution can be done efficiently using the concept of Livecoding and IOT. We present here some live examples of technology and process interventions which have the potential to significantly improve the distribution of Power. Livecode is compile-free language so you can run and edit your application live. We remotely control the distribution of Power using IOT devices such as Raspberry Pi and distribute the Power to different location efficiently based on the requirement.

8 Future WORK

This paper is applied for two locations but it can be extended to more location or entire city, The method of automation can be applied to our paper and make human intervention even limited. The concept of live code can be applied to different domain and the amount of current used can be send to individual users on a daily basics and let the user keep track of their usage.

References

- [1] A Lesson in LiveCode From Theory to Algorithms By: Shai Shalev-Shwartz and Shai Ben-David Designing the Internet of Things: By: Adrian McEwen, Hakim Cassimally
- [2] Electric Power Distribution By: A. S. Pabla
- [3] Cloud Computing: Concepts By: Zaigham Mahmood, Thomas Erl, Ricardo Puttini
- [4] An Introduction to Statistical Learning with Applications. By: Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani.
- [5] Building Internet of Things with the raspberry pi By: Charalampos Doukas.
- [6] Getting Started with the Internet of Things By: Cuno Pfister.

- [7] Internet of Things with the raspberry pi By: Marco Schwartz MahbubulAlam,
- [8] Mining of Massive Datasets By :Jure Leskovec, AnandRajaraman and Jeff Ullman
- [9] T. Gonen (1986), "Electric power distribution system engineering". New York: McGraw Hill
- [10] R.E.Brown and M.Marshall(2000), "Budget constrained planning to optimize power system reliability". IEEE Transactions on Power Systems, v.15, n. 2, pp. 887-892.
- [11] F. Schweppe and D. Rom, Power system static-state estimation: Parts I, II, & III, Power Apparatus & Sys. IEEE Trans., vol. 89, no. 1, p. 125–130, Jan 1970.
- [12] Z. Zhi, L. Min, and X. Shiyu, A survey on state estimation algorithm of distribution grid, in Proc. Spring Congress on Engineering and Technology (S-CET), 2012 IEEE, May 2012, p. 1–4.
- [13] F. F. Wu, Power system state estimation: a survey, Int. Journal of Elec. Power & Energy Sys., vol. 12,no.2,p.80–87,1990.
- [14] W.-H. Liu and S.-L. Lim, Parameter error identification and estimation in power system state estimation, Power Sys., IEEE Trans., vol. 10,no. 1, p. 200–209, 1995.
- [15] E. Castillo, A. Conejo, R. Pruneda, and C. Solares, State estimation observability based on the null space of the measurement Jacobian matrix, Power Sys., IEEE Trans., vol. 20, no. 3, p. 1656–1658, Aug 2005.