Volume 5, Issue 4, 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Spark Core Based Wireless Remote Door Lock and Multiple Access Synchronization

Shirsankar Basu*, Md Arshad Feeroz, Diljit PR, Neha Firdaush Raun, Faraz Alam
CSE, GNIT, Kolkata,
West Bengal,India

Abstract — With the rapid advancement of technology, Internet usage and connectivity has grown manifold, things have become mostly mobile and innovations have taken a new turn with the seamless, ubiquitous computing accelerated by Internet of Things(IOT).The Internet of Things is the Network of Things wherein blended are several electronic modules, software packages, sensors, actuators alongside the internet connectivity to enable it to achieve greater value and service. Backed by the idea of IOT, this paper puts forward the concept of a door lock that can be controlled remotely from anywhere across the globe through an Android ADK based mobile application or a web application. In essence, the smart-lock-project concentrates on the replacement of keys with a smartphone or a computer, and also resolving the access requests issued by multiple users. For the existing hardware module, the project uses Spark Core Microcontroller equipped with the Arduino board on one side and Texas Instruments Wi-Fi Module on the other. The Spark-firmware mentioned above interleaves the background CPU activity associated with WiFi and cloud activity.

Keywords— Internet of Things, Spark-Core, Spark Cloud Server, Wi-Fi Module, Android ADK.

I. INTRODUCTION

In the traditional era of home automation, a digital door lock system is equipped with several digital information, encoded as in smartcards, secret codes, semi-conductors and finger prints, as the method of authentication, ripping through the very legacy of key system. But the next wave in the era of computing tides away the bounds of traditional workshop with the advent of Internet of Things [see Figure 1] paradigm.

In the Internet of Things(IoT) [1] paradigm, many of the objects that surround us, such as sensors and actuators, are already on the network in one form or another. Radio Frquency Identification(RFID)[2] and sensor network technologies shall definitely rise to face new challenges with information and communication systems are invisibly embedded in the environment around us. As a consequence, it generates enormous amounts of data to be stored, processed and presented in a seamless and efficient form, yielding services and commodities similar to the traditional ones. As noted by Gartner's IT Hype Cycle[1], Internet of Things(IoT) has a massive impact on emerging technologies and applications and will take about 5-10 years for market adoption [3].

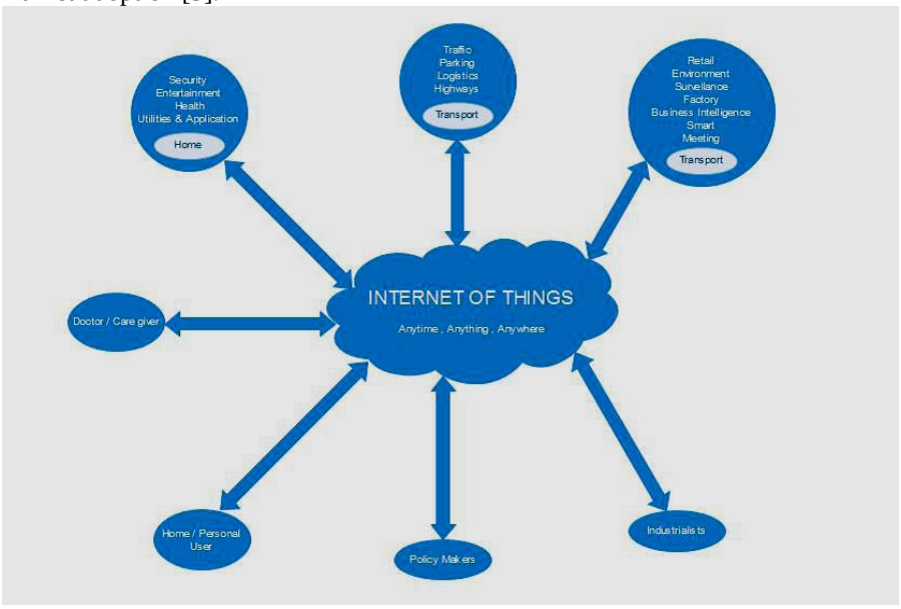

Fig 1: The Internet of Things Schematic

[1] Gartner's IT Hype Cycle is a way to represent the emergence, adoption, maturity, and impact on applications of technologies.

The taxonomy of several components required for the Internet of Things(IoT) are hereby listed. There are primarily three IoT components that facilitates seamless ubiquitous computing: (a)Hardware{made up of sensors, actuators and embedded communication hardware, (b)Middleware[2] {on demand storage and computing tools for data analytics, and (c)Presentation{easy-to-understand visualization and interpretation tools that can be easily accessed on different platforms.

Backed by IoT, our work is, in the main, based on the resource called Spark Core, which is a Wi-Fi development kit for internet-connected hardware. The core has on board a microcontroller [4] and Wi-Fi module. The Spark Cloud API, needed here, has got a uniform resource locator that is used to get variables, post a function call or put new firmware. All the requests to Spark Core come through the API server using certain security system. Each Spark Core device is supplied with a unique device id using which, a particular Spark Core connects to the Spark Cloud Server. To control a core, one xmust define and expose functions in the Core Firmware, and thereafter these functions can be called remotely using the Spark Cloud API.

The use of Spark Cloud API in conjunction with Internet of Things(IoT) [5] paradigm definitely cuts short the several glitches associated with wired or wireless digital door lock systems, [6] and also cuts down on the installation charges of several required hardware devices to do the same.

II. MICROCONTROLLER AND WI-FI MODULE

The Microcontroller is a small, low-cost, low power arduino board based processor that runs a single application .It runs the software and tells the core what to do without having an operating system. Instead, it runs a single application, often called a firmware or an embedded application [7] having a few lines of code, or can be at times very complex, depending on the objective of work.

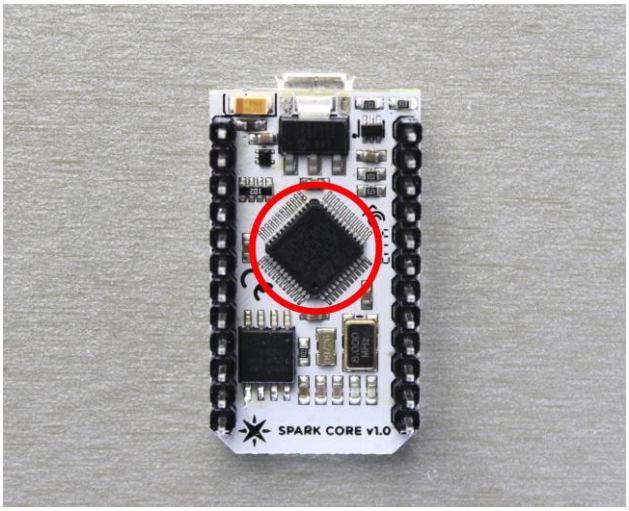

Fig 2: The Spark Core Diagram

The microcontroller[see Figure 2], good at controlling things, has a set of "pins" that are called GPIO(General purpose Input Output) pins or I/O pins, that can be hooked, to sensors or buttons to listen to, or, to lights and motors to act upon the world. These microcontroller's pins are directly connected to the headers on one side of the core so that one can easily access them; to be specific, the pins labelled D0 to D7 and A0 to A7 are hooked directly to microcontroller's GPIO pins.

A microcontroller can also communicate with other chips using common protols like serial (also called UART), SPI or IPC .It can be made more powerful by connecting it to the special purpose chips like motor drivers or shift registers. Sometimes these chips can be wrapped on a shield as an accessory to extend the core. Here the microcontroller uses the arduino language with it's custom libraries.

Some of its key features are as follows.

- ARM 32-bit Cortex M3 CPU Core.
- 72 MHz operating frequency.
- 128 KB of Flash Memory.
- 20 KB of SRAM.
- 12 bit ADC.
- USB 2.0 full speed interface.
- USART, SPI and 12C interfaces.
- JTAG Debugmode.

The core also has a Wi-Fi module that connects it to the local Wi-Fi network in then same way a computer or smart phone does to a Wi-Fi network. The core is programmed to stay connected to the internet by default.

[2] The middleware is also sometimes mentioned as Firmware: eg, Spark firmware.

III. CONNECTION AND SYSTEM ARCHITECTURE

A. Connection to Spark Cloud

The core can be connected to the Wi-Fi router through a Spark Android Appon our phone. This might take a little while; but the result reflects through the following colours.

- **Blinking Blue:** Listening for Wi-Fi channels.
- **Solid Blue:** Getting Wi-Fi information from the application.
- **Blinking Green:** Connecting to the Wi-Fi network.
- **Blinking Cyan:** Connecting to the Spark
- **Blinking Magenta:** Updating to the newest firmware.
- **Breaking Cyan:** The status is *Connected*.

Thus when the core connects to the Internet, it establishes a connection to the Spark cloud: by connecting to the cloud, the core becomes accessible from anywhere through a simple RESET API(application programming interface)[see Figure3].The API is designed to interface with the core through a web application or a mobile application in a secure and private way, so that only authenticated and trusted users can get to access the core. Thereafter the code on the icrocontroller can be burnt through the Spark Web IDE over the cloud.

B. Design and System Architecture: Spark Lock

The system architecture of our work is comprised of the following central components.

- Mobile Application/Web Application.
- A Router.
- The Internet of Things(IoT) Model
- The Spark core Microcontroller
- A Door Based Mechanical Lock.

IV. IMPLEMENTATION AND ALGORITHM

The biggest challenge of this project is to glue the different pieces together and maintain user synchronization for this application. The challenge was first met with creating a cross platform application in html5 but later on shifted to create two different applications for android[3] and the web. The user with a basic web application, as shown here, has to enter user credentials[see Figure 4] and submit the same to the server for processing. If the user credentials match, they are redirected to a different page wherein they need to check the current status[see Figure 5] of their lock, send a virtual key to their friends or families or, for that matter, to the users requesting access, view logs or, if necessary, change his/her keys. The minimal user interface lets users lock or unlock the system through a simple click. The intelligent user synchronization algorithm prevents other users from accessing the application and helps to maintain the integrity.

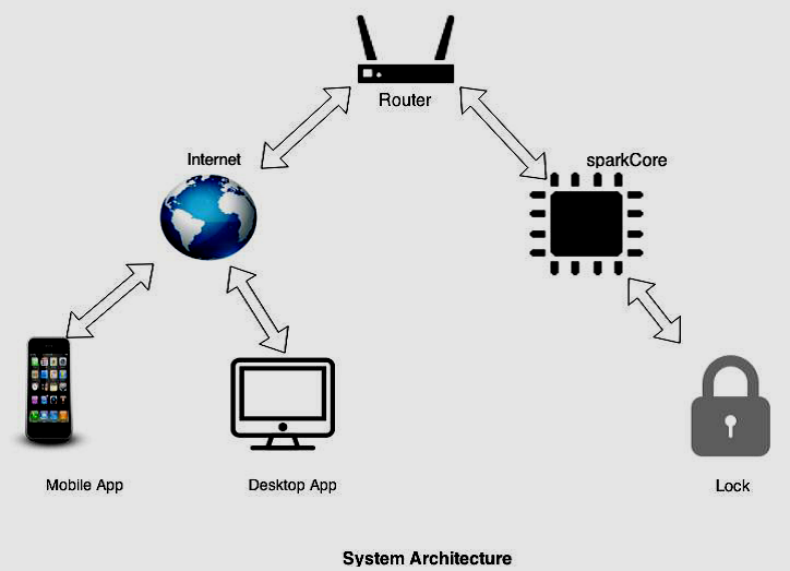

Fig 3: System Architecture Schematic

The feature called send invites [see Figure 6] lets user sends virtual keys to the authenticated users requesting accesses.

[3] Android is a software stack for mobile devices that includes an operating system, middleware, key applications, made in Java-like language, running on Dalvik created by Google. The android OS is based on Linux.

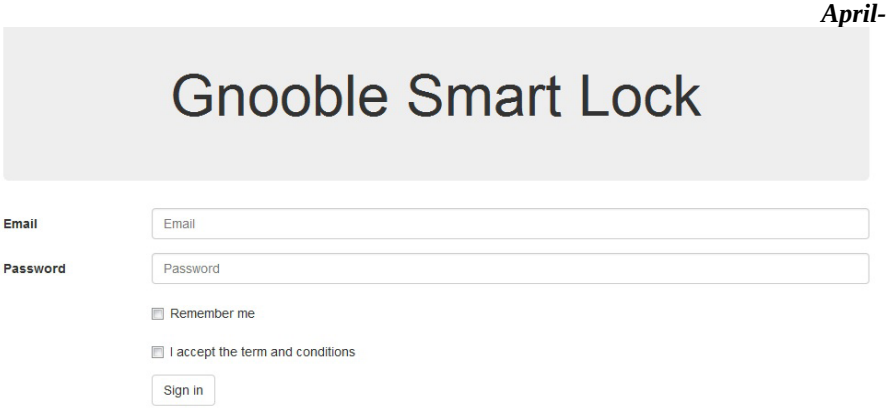

Fig 4: Spark Lock: Asking for User-name and Password

The heart of the application underlies the locking/unlocking of the physical lock. Prior to the user having the option "lock/unlock" his or her lock, the algorithm checks whether there already exists any user using the application. In essence, when the user tries manipulating the lock, he or she enters the critical section of the code, that is ensured by the synchronization[4] part of our algorithm.

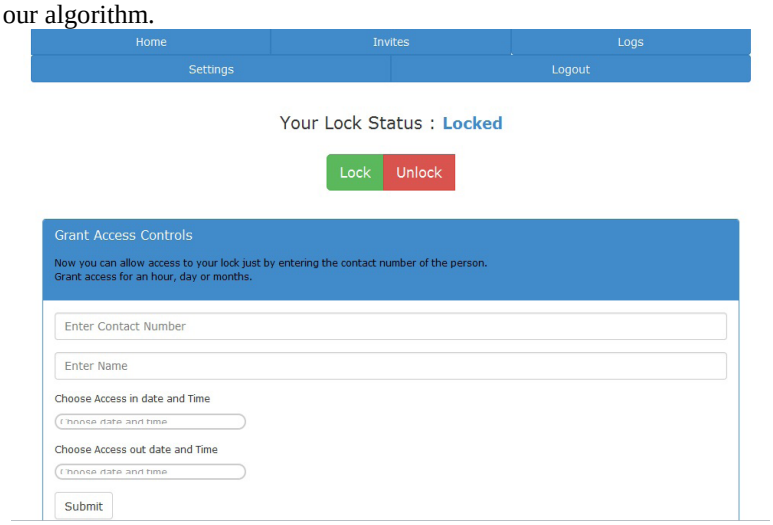

Fig 5: View current Status of the Lock

Algorithm 1 The Smart-Lock Algorithm

Step1: START.
Step2: Get your user id and password.
Step3: Enter user interface and validate Login.
Step4: If login = successful, go to Step5.
Step5: Available choice(s).

- If (choice = 1) Go to Step6: Lock/Unlock
- If (choice = 2) Go to Step8: Check lock status
- If (choice = 3) Go to Step9: Send invitees.
- If (choice = 4) Go to Step14: View Log of users.
- If (choice = 5) Go to Step15: Logout. End of Algorithm.

Step:6 Press Lock/Unlock button.
Step:7 Opens/closes the lock remotely and go to Step:16.
Step:8 If (locked), Display status Locked Else Unlocked
Step:9 Do send invitees/grant access.
Step:10 Enter contact number, access-in- time, access-out-time, send invite.
Step:11 Guest user receives a sms with a url and hash key.
Step:12 Guest user clicks on the url and gets a guest interface.
Step:13 Guest user enters contact number and hash key, access the Lock between in-time and out-time and go to Step:16.
Step:14 View logs of the guest users; go to Step:16.
Step:15 Logout.
Step:16 END.

[4] Synchronization part reflects in Step:6 through Step:8

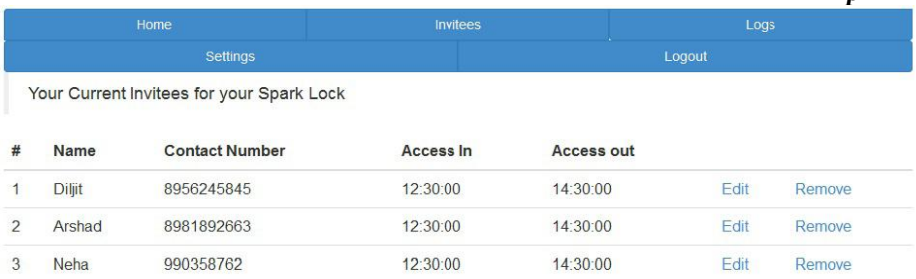

Figure 6: Current Invitees of Spark Lock

V. CONCLUSION AND FUTURE SCOPE

Thus the Spark Core Based Wireless Remote Door Lock and Multiple Access Synchronization has been designed and tested successfully. It has been developed integrating all the features of the hardware components used. Novelties in this work include the concept and use of Internet of Things (objects) and Spark Core Microcontroller with a Wi-Fi module as to connect to the Spark Cloud Server. It has been made sure that, given proper user credentials, multiple users can operate the door lock remotely through a synchronization algorithm of the critical section of the code. The use of Android ADK based mobile application makes it a hardware-cum-software project as to broaden the space of work. Further work on this project can concentrate on enhancing the security issues using more advanced encryption technology.

REFERENCES

[1] Ashton K. 2009. 'That "Internet of Things" thing', *RFID Journal*.

[2] Gartnerv Inc. 2012. Gartner's hype cycle special report for 2011.Available at:http://www.gartner.com/technology/rese-arch/hype-cycles/.

[3] Buckley, J. (Ed.). 2006. The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems. New York: Auerbach Publications.

[4] Mazidi, Muhammad Ali, Janice Gillispie Mazidi and Rolin D. McKinlay. *The 8051 Micro-controller and Embedded Systems*.

[5] Zorzi, M., A. Gluhak, S. Lange and A. Bassi. 2010. ' From today's Intranet of Things to future Internet of Things: A Wireless and Mobility-related View', *IEEE Wireless Communications* 17: 43–51.

[6] Alkar, A. and U. Buhur. 2005. 'An Internet based wireless home automation system for multifunctional devices', *IEEE Transactions on consumer Electronics* 51: 1169–1174.

[7] Li, X., R.X. Lu, X. H. Liang, X. M. Shen, J. M. Chen and X. D. Lin. 2011. 'Smart community:an Internet of Things Application', *IEEE Communications Magazine* 49: 68–75.

