Proc. of The Fifth Intl. Conf. On Advances in Computing, Electronics and Electrical Technology - CEET 2016
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-087-3 doi: 10.15224/ 978-1-63248-087-3-61

Intelligent Mailbox

An application of Internet of Things

Nabeel ur Rehman, Adeel Muhammad

***Abstract*— Internet of things (IoT) has wide range of application and it has potential to make things easier to monitor and manage in our day to day life. This research has developed a prototypic implementation of an intelligent mailbox using the concepts of internet of things to facilitate monitoring of physical mailbox installed at homes or work places. The successful demonstration of prototype suggests that low cost embedded platforms such as Raspberry Pi 2 can be used to develop an economical, smart and easy to manage IoT applications. This project can be scaled up to commercial level for offering services to masses with further modifications.**

***Keywords*—Internet of things, Intelligent Mailbox, Matlab Simulink, embedded solution, M2M**

I. Introduction

Home automation is becoming a necessity for every day life as semantic web is expanding. More than 60% of the energy consumed in buildings originate from household appliances [1]. If appliances are made intelligent and networked, then there is a possibility of reducing this huge amount of consumption. Reducing consumption is therefore a matter of managing and optimizing the utilization of electrical energy on a system level based on energy profiling. In order to incorporate this concept in home automation, a system wide standard has to be implemented where a network of small controllers communicate with one central controller which controls and actuates the rest of appliances. This is therefore mandatory that such a cluster of sensors, actuators and control units communicate over a given data model or ontology.

Keeping in view the future expandability opportunities of smart home or connected grid systems, it is necessary to have a standard which is used by system integrators, technology vendors, power and energy utility integrators, service providers and telecom companies. Based on fact that interoperability of new devices and integrating future appliances is required, we suggest to use a standard set by European Telecommunications Standards Institutes (ETSI) for Machine to Machine (M2M) communication. One of the primary reasons to incorporate oneM2M standard is that it addresses the need of common M2M service layer which can be readily embedded in different hardware and software products with ease. It can be relied to connect the myriad of devices and application servers across multiple vendors [2].

Nabeel ur Rehman
University Heilbronn
Germany

Adeel Muhammad
University of Stuttgart
Germany

Added advantage of using oneM2M defined by ETSI is that it already incorporates extended level of security which means direct usability with eHealth and telemedicine, industrial automation and home automation. This gives advantage of networking with pacemakers, fitness trackers and other medical facilities installed at hospitals or at homes for residential treatment facilities which are monitored remotely by team of doctors.

For the discussed implementation in this research paper, basic facility used is Raspberry Pi 2B (RP2B) which is an economical low cost, low power single board computer. It can be used to run resource intensive tasks which requires ordinary level of computation. One of important factors to choose Raspberry Pi 2B as a platform is direct integration of Windows 10 IoT core which comes with dashboard to directly integrate IoT applications and its core devices. This core is not yet used but will be integrated in intelligent mailbox as future development when implemented at commercial level. In the current conception and validation stage, an open source flavor of Linux is used which is known as Raspbian. Raspbian supports direct integration and support of MATLAB and Simulink. For the stated concept and prototype implementation, Simulink from Mathworks was used to create system level model.

Using the low level interfaces of RP2B through MATLAB, it is very much possible to simulate and model the real world environment inside Simulink. Physical connection with I/O pins of A/D converters, push buttons and flash light connections are possible on lowest level in RP2B which can be monitored and modeled within MATLAB and Simulink environment. Image and video toolboxes can then be used in order to get meaningful data in intelligent smart mailbox.

The main focus of this research is development of an intelligent mailbox which has the ability to communicate with residents of home as soon as a mail or parcel is delivered. The implementation is done using RP2B. Combination of C-programming language and shell scripts are used to implement the concept. Prototype is successfully tested and desired results are obtained.

II. Methods and Equipments

This project is a combination of both hardware and software components which are communicating together for successful execution of intelligent mailbox. Components have been selected to minimize the setup efforts and costs. This section identifies hardware, electronics circuit components and software packages used in this project.

A. *Hardware Components*

Raspberry Pi 2 is working as a core computation device for this project. Model used for this particular implementation is Raspberry Pi 2B (RP 2B) which is a second generation model with 900 MHz quad-core ARM

Proc. of The Fifth Intl. Conf. On Advances in Computing, Electronics and Electrical Technology - CEET 2016
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-087-3 doi: 10.15224/ 978-1-63248-087-3-61

Cortex A7 CPU and RAM of 1.0 GB. It incorporates 40 GPIO pins, 4 USB ports along with full HDMI and ethernet ports. A camera can be attached to camera interface (CSI). A micro SD card slot is available for installation of operating system which also acts as storage unit. It has capability to run full range ARM GNU/Linux distribution or other operating systems like Window 10 IoT. This project is using ARM GNU / Linux distribution due to reason discussed in section I. 3D layout of Raspberry Pi 2 Model B is shown in Fig. 1.

RP2B camera is another hardware used which is a five megapixel fixed focused camera which supports 1080p30, 720p60 and VGA90 video modes. It can be attached with Raspberry Pi on CSI with 15cm ribbon cable. For internet connectivity, TP-Link wireless router model TL-WN722N is used which has 150Mbits/s data rate. Wireless router configurations required to run it with RP2B are mentioned in section III. With help of WLAN connection, the solution can be made as portable as possible with minimum number of wired connections needed.

Conventional LCD Screen, PS2 mouse and keyboard are attached with USB ports of Raspberry Pi for initial setup. Once the setup has been done, they can be removed to make it standalone system. Other electronics components include push button, proximity sensors, resistors and connecting wires to connect these components. This project assumes that testing facility has access to wireless internet connection and mailbox. In future real product implementation, GSM module can be integrated to remove this dependency.

The postbox used is a standard lid postbox which is installed with required proximity sensor, pushbutton and flash light along with standard 3.3volts power connection. Standard wire connections can be made

Table 1 summaries the hardware and electronic circuit components that are used in this project along with the comments about their usages.

B. *Software Packages*

Four different software packages are used in this project. bcm2835, mentioned at [4], is used to enable reading and writing on the Raspberry Pi 2 pins for input and output operations. After installation of bcm2835, necessary configurations are stored with the project code for its usage. These are explained in section III.

Figure 1. Raspberry Pi 2 Model B [3]

TABLE I. HARDWARE REQUIREMENTS

Components	***Model***	***Usage Comment***
Raspberry Pi 2	B	Single Board Computer
Camera	RP2B Camera Module	Imaging Tool
WiFi Adapter	TP-Link TL-WN722N	WiFi connection
LCD, PS2 Mouse and Keyboard	Any	For initial setup
Bumper Switch and proximity sensor		Take input from mailbox lid
LED and Wires		Camera flash and to make connections
Mailbox		Receive Mails

Geany, a third party IDE is used as main programming environment on Raspberry pi. This IDE supports Phtyon, Java and C++ codes as well. It is chosen due to its less number of dependencies requirement. Added advantage of using Geany is independence of a particular desktop environment like KDE or GNOME. C-programming language is used to implement the basic functionality of I/O in RP2B. Rest of the model is loaded from MATLAB Simulink.

Camera and email modules has been implemented using shell scripts in Linux. These scripts can be called from Simulink or can be made to run directly from C language implementation. Raspberry Pi 2 Linux also requires SSMTP and MPACK to be installed for sending emails and making attachments in email respectively.

In order to encapsulate the data in specific ontology to be used by other resources, an open source ontology developer, Protégé is used

III. Control Algorithm And Implementation

This section explains the implementation of hardware components, software algorithms, work flows and configurations which are required to set up this prototype.

A. *Model Description and Proof of Concept*

MATLAB Simulink model can be made as shown in figure 2. This model is illustrated only to proof the concept developed. Once its is successful, a real prototypic implementation can be done. In order to run such a model, RP2B needs to be connected with computer running MATLAB using wired LAN connection. Sample inputs are generated on programmed pins i.e. Pin 09 and Pin. 15. "RP Model" block runs the state machine which is described in section III-B. In order to take a snapshot, commands are sent to RP in Snap_shot block. Post processing of captured image is also done in order to convert captured image from YCbCr to RGB, noise is reduced using averaging filter and image quality is reduced to decrease the size of attached image. This image is then stored before "Send_email" block sends and raises the statusFlag which marks the successful completion of task.

SEEK DIGITAL LIBRARY

Proc. of The Fifth Intl. Conf. On Advances in Computing, Electronics and Electrical Technology - CEET 2016
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-087-3 doi: 10.15224/ 978-1-63248-087-3-61

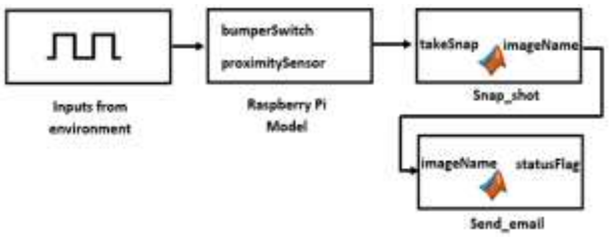


Figure 2. Matlab model diagram showing work and dataflow

B. *Hardware Assembly and Workflow*

In order to realize Simulink model as discussed in Fig. 2 in real prototype, the basic hardware connections and structure is shown in Fig. 3. Bumper switch is attached directly under the lid of mailbox which is in OFF state once lid is closed. Once the lid is open, the bumper switch goes into ON state which is regarded as first indication that the mailbox is open. An input from proximity sensor is also taken which confirms the open lid state of letter box. Lid is constantly monitored using input of switch and sensor. As proximity sensor is calibrated to respond to even slightest of movements, it can be observed if letter is added or not. A delay is added till lid is open. An interrupt is generated to snap an image of mailbox interior once mailbox lid closes and bumper switch is in OFF state. It is assumed that lid is properly closed. In case bumper switch doesn't go to OFF state for considerable period of time, its input is ignored as fault tolerance strategy and only proximity sensor input is used to monitor both lid state and new letter. A full recovery alert is generated and sent to owner as email with high priority alert.

Before an image is taken, LED flashlight will be switched on in order to eliminate darkness. Once an image is captured, it is post processed before sending to owner of mailbox. Open computer vision library (OpenCV) is used to average out the image in order to reduce the noise generated by flashlight in image. A simple implementation of edge detection algorithm like Canny Edge detection can be used to find the edges [5]. Edge detection algorithm enables to crop out the parts of image which do not belong to received mail [6]. Once image is finalized, LED flashlight is turned off which is easily operated using output pin of RP2B. Before attempting to send email, onboard USB port is powered on which was kept in OFF mode along with other peripherals to minimize the used power. USB port with connected WLAN port is initialized and connection is established as discussed in configuration management section. After successful completion of this task, USB port is again shutdown.

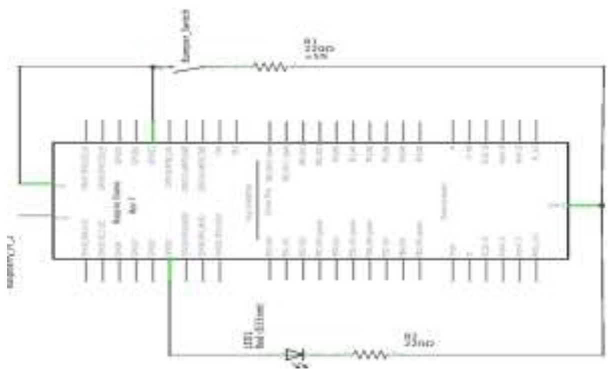


Figure3: Hardware assembly and electronic circuits

C-programming language and shell script are used to develop the work flow of this project. This project has two major work flows. The main program thread which continuously monitors and controls the actual program and a back ground process / back ground worker which snaps and image and perform image processing before sending an email to desired list of recipents. Main process described in this section is summarized in Fig. 4.

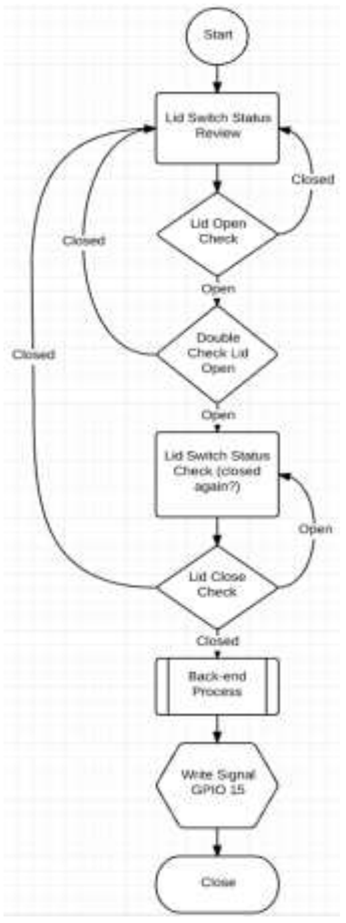


Figure 4. Main program workflow

C. *Mapping on IoT Ontology*

In order to share the data of Raspberry Pi and mail box with central controller or other smart devices in home, the data is encapsulated according to SAREF oneM2M ontology. The required data to share includes the captured image, time stamp when the image was captured, state of bumper switch and proximity sensor, wireless router status and power utilization profile. Although power utilization is in microwatts for our setup, it is still included to completely incorporate the SAREF ontology standard. According to the ontology, a device is a tangible object designed to accomplish a particular task in households, common public buildings or offices [2]. In order to accomplish this task, the device performs one or more functions. In the setup described in Fig. 3 - Raspberry Pi, proximity sensor, bumper switch, LED, mailbox and wireless router are all considered as saref:Device. These are all the hardware components listed in Table 1 except LCD, mouse and keyboard which are not included because they are only required in setup phase.

Proc. of The Fifth Intl. Conf. On Advances in Computing, Electronics and Electrical Technology - CEET 2016
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-087-3 doi: 10.15224/ 978-1-63248-087-3-61

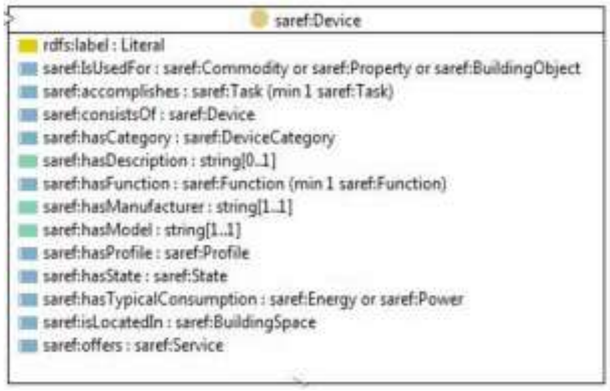
All the devices in given ontology can be mapped in three major categories which are saref:FunctionRelated, saref:EnergyRelated and saref:BuidlingRelated. In the given scenario, mailbox is categorized as building related and energy related while all the rest are categorized both as function related as well as energy related. Function related devices can further be categorized as saref:Senors (proximity sensor), saref:Actuator (bumper switch), saref:Network (wireless router) and saref:Lightining (LED). Once devices are rightly categorized, they can be further categorized by their functions. This means that saref:Sensors is defined by its function and command. It accomplishes a saref:SensingFunction and it can raise an event which is defined as saref:EventFunction or saref:Command. This event function is actually the input which is recorded as change of state as described in section III-B. A saref:Actuator accomplishes function of saref:OnOffFunction. These all saref:Functions can be mapped and linked with original C code to know the real status of every saref:Device.

Each saref:Device identified earlier has properties as given in Fig 5. The saref:hasProfile property can provide us with necessary information such as energy and power ratings, price and time information. Likewise. saref:hasState property provides us to store the current state of bumper switch and proximity sensor. With help of such hierarchical definition of each device, whole system can be presented in a tree like structure.

In our case where sensors and bumper switch etc. are not real smart devices but we are incorporating off the shelf devices in our system, the central unit of our mailbox i.e. RP2B can store the data. With help of an abstraction layer between RP and rest of smart home controller, we can accomplish the same task as with help of real IoT enabled devices. This means that in our implementation, for example, a bumper switch is queried for its state by any other appliance in smart home, its response is not generated by bumper switch in mailbox rather it is done by RP which has information stored about all devices which are connected to it [7] [8]. Such a paradigm is adopted because vendors are yet to produce low cost IoT enabled sensors and actuators or other devices.

D. *Configuration Management*

Implementation discussed in this research requires the configuration in Linux/Raspian for multiple modules for example Wi-Fi connection, bcm2835 software package, `SSMTP` & `MPACK` installations and cronjob settings.



saref:Device

rdfs:label : Literal
saref:IsUsedFor : saref:Commodity or saref:Property or saref:BuildingObject
saref:accomplishes : saref:Task (min 1 saref:Task)
saref:consistsOf : saref:Device
saref:hasCategory : saref:DeviceCategory
saref:hasDescription : string[0..1]
saref:hasFunction : saref:Function (min 1 saref:Function)
saref:hasManufacturer : string[1..1]
saref:hasModel : string[1..1]
saref:hasProfile : saref:Profile
saref:hasState : saref:State
saref:hasTypicalConsumption : saref:Energy or saref:Power
saref:isLocatedIn : saref:BuildingSpace
saref:offers : saref:Service

Figure 5. Device properties according to SAREF ontology

Wi-Fi connection setting needs to be adjusted in network interfaces configuration file of Linux which is normally located at path: "`/etc/network/interfaces`". The settings are detailed as:

```
auto lo
iface lo inet loopback

iface eth0 inet dhcp
auto wlan0

allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-ssid "<YOUR NETWORK SSID>"
wpa-psk "<NETWORK SECURITY KEY>"
```

In order for RP2B to support e-mail service and attachment service, `ssmtp` and `mpack` services have to be installed. To install services, root user access is required and can be easily obtained using wget or apt-get. To keep the program running as soon as Raspberry Pi boots, following set of commands need to be added in rc.local located at path "`/etc/rc.local`"

```
cd /projectLocation/
sudo ./output_file_name &
```

To gain low level direct access to input output pins of Raspberry Pi, bcm2835 package installation is required. Package "bcm2835-1.49.tar.gz" can be downloaded from its resources and quickly installed using:

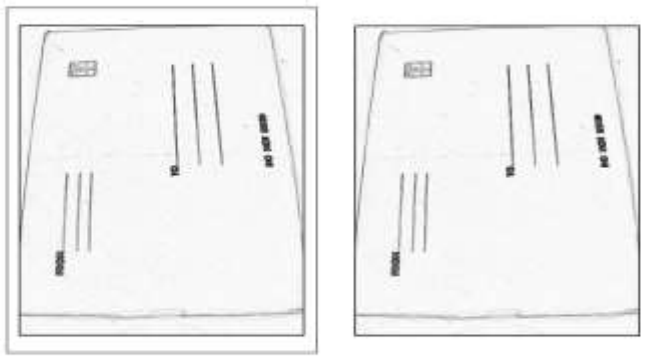
```
tar zxvf bcm2835-1.xx.tar.gz
cd bcm2835-1.xx
./configure
make
sudo make check
sudo make install
```

IV. Results, Conclusion and Future Work

Successful implementation of this prototypic project suggests that RP2B can help in providing an economic, smart and efficient solution for the implementation of intelligent mailbox or any other similar small applications. With current implementation, it is also possible that user sends a request from its mobile or any internet enabled device to get the current status and picture of mailbox via email. This strategy is very useful in the case when we have a fault in system which is reported by mailbox controller such as lid is not closed properly by postman or due to any failure in mechanical part of mailbox. Under such circumstances, the owner of mailbox can generate a request in form of email which is read by RP2B and responded with current picture attached in an email.

An important factor in this implementation is decreasing the image quality along with efficient cropping which does not crop out required information but cuts out sufficient to reduce the image size for easy sending of email with attachement. With certain trails, certain image quality is maintained, edge detection is performed before cropping out the image as shown in Fig. 6.

SEEK DIGITAL LIBRARY



(a) (b)

Figure 6. (a) shows the original image (b) shows cropped image

As shown in Fig. 6, approximately 12% image reduction has been achieved from top and bottom while 6% of image size is reduced from sides. For a standard 640x480 pixels image, size is reduced to 488x424 which is significant decrease in size if RGB image is sent to mailing list.

Successful implementation of SAREF/oneM2M ontology and mapping non IoT devices to specified ontology provides a positive impression that by abstracting the used sensors and actuators, IoT based applications can still be produced. This implies that available low cost devices can be utilized straight away rather waiting for economical IoT enabled sensors or actuators.

This project can be scaled up to facilitate elder people of the community and / or out of home people to keep an eye on their mailbox without physically checking it. Integration of this solution with wireless GSM connection can also enable to receive the SMS notification or image in MMS. In addition to this, use of better button like magnetic switch can help in providing better input. Improvements in the switch can also make this product more robust in terms of its usage. Developed prototype involve basic checks to monitor the opening and closing of lid. More and more checks can be developed to cater of exceptions like improper closure of lid, multiple opening and closing of lid, to name a few. Automation of daily appliance can help in setting up smart homes easily.

From technical aspect, future improvements in this application can include but not limited to controlled access to shared data between multiple resources using software locks and data synchronization. In future, Windows 10 IoT core can also be incorporated on Raspberry Pi in order to explore new opportunities and decreasing the setup effort.

V. Acknowledgment

We would like thank Mr. Alexander Faul from University of Stuttgart in Germany, Miss Sussane Mai and Mr. Lucas Steinmetz from Heilbronn University in Germany for their contribution in providing implementation guidelines, support and resources for implementation of this innovative idea.

VI. References

[1] Laura Daniele, Frank den Hartog, Jasper Roes "Study on Semantic Assets for Smart Appliances Interoperability" D-S4 Final Report, Ver. 1.0, pp. 07, March 2015

[2] Laura Daniele, Frank den Hartog, Jasper Roes "Study on Semantic Assets for Smart Appliances Interoperability" D-S3 Third Interim Report, Ver. 1.0, January 2015

[3] Raspberry Pi 2, Model B, https://www.raspberrypi.org/products/raspberry-pi-2-model-b/.

[4] Mike McCauley, "bcm2835 documentation" Ver 1.49

[5] Sunil Shah, "Real-time Image Processing on Low Cost Embedded Computers", May 2014

[6] Tomas Toss "Automatic identification and cropping of rectangular objects in digital images" Department of Information Technology, September 2012

[7] McBride, B "The Resource Description Framwork (RDF) and its Vocabulary Description Language RDFS. In: Staab, S." Studer, R. (eds.) Handbook on Ontologies, pp. 51-66, Berlin, 2004

[8] Tomas Toss "Enabling query technologies for the semantic sensor web" International Journal on Semantic Web and Information Systems (IJSWIS) 8, pp. 43-63, 2012