Raspberry Pi 3 as a Smart Stand-alone POS System in a Small Restaurant Business

Sidney Carlo Lopez, Sidney Sheldon Tan, Wellington Villasin, Aaron John Zarzoso and Engr. Mark Lorenze Torregaza
BS – Computer Engineering, Department of Electronics and Communications Engineering,
De La Salle University Manila, 2401 Taft Ave, Malate, Manila, 1004 Metro Manila.
sidney_lopez@dlsu.edu.ph

***Abstract*—Most retailers consider Point of Sale (POS) systems as the central component of their business. It is what keeps everything together and allows an efficient flow of service within a business. It is important to know what a POS system really is, it is not one entity. A POS system utilizes software and hardware components, every establishment uses a different combination of hardware and software. This paper aims to show a cost effective POS system that makes use of the raspberry pi 3. The system will be implemented on a small restaurant and will automate order and payment transactions. Notifications to the owner of each transaction will be possible via SMS and access to the inventory via database is also possible via the internet.**

***Index Terms*—Point of Sale; Raspberry Pi; Internet of Things; SMS; Python.**

I. INTRODUCTION

According to Yamerie Grullon [1], A POS system is an integration of hardware and software that allows vendors to take transactions and efficiently carry out day-to-day business operations. Most small restaurant businesses operate on simple POS equipment which does not alleviate much of the work in the restaurant. POS equipment are being constantly improved, most POS equipment today use touch-screen technology for ease of use and some are even internet of things (IoT) devices. This paper aims to show the integration of different hardware components that come together to create a smart stand-alone POS system for a small restaurant business.

Modern POS systems can perform a multitude of functions including the following [2]:

- Determine payment for each transaction
- Record the method of payment
- Note the amount of money in the register
- To create periodic reports within the day
- Allow hourly employees to clock in and out
- Determine labor and payroll data
- Record daily check averages for each employee
- Keep track of menu items sold
- Record information on repeat customers

There are two common software deployment methods; one is On-premise which is the traditional software model, the other is Cloud-based which is also known as Software-as-a-Service (Saas) POS solutions. Regardless of the software deployment method, the hardware components is the same. Crucial hardware components include cash registers, receipt printers, barcode scanners and debit/credit card readers. Ideal POS software features include [2]:

- Sales Reporting, to analyze sales data
- Customer Management for log purchases
- Inventory Management to manage quantity of stock
- Employee Manage to provide clock-in/clock-out tools

For this paper, a raspberry pi will be used as the central processing unit. Similar works have been done already. A Raspberry Pi is a credit card-sized computer which was originally designed for educational purposes (Figure 1).

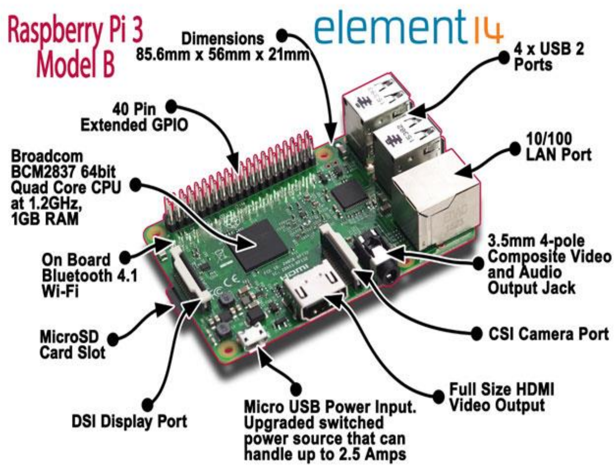


Figure 1: Description of the Raspberry Pi Board

The development of the Raspberry Pi has brought a surge of enthusiasts that bear great potential for applying computing in various areas. The Raspberry Pi brings a lot of advantages and features that allow for students and enthusiasts to tinker and develop different systems via the General Purpose Input/Output (GPIO) pins which is used to interface general purpose electronics [6].

A POS system based on the raspberry pi by Hiram Villareal called “IoTPOS a point of sale for Raspberry Pi” has been posted in Hackaday.io. His POS system includes a serial receipt thermal printer with a function to print barcode product labels without a need to buy a dedicated printer. He redesigned the interface to work with low resolution screens as 800x480px, now compatible with the official 7″ touchscreen monitor for the Pi. The IoT functions includes e-mails with: receipt copies, status reports, user logins, offers, CSVs or others important notifications for you and your clients [3]. This project is cost-effective as it eliminates most of the expensive parts and functions of a propriety POS.

To achieve the stand-alone function of the POS system, a bill and coin acceptor and a coin changer module will be used to handle payment transactions. Two thermal printers will be utilized, one for the front of house that provides the customer's copy and the other is for the kitchen to prepare orders. The software will be implemented in python using Kivy as the user interface (UI). Kivy is an open source Python library for the development of interactive user interfaces of multi-touch applications [5]. The POS device will house a 7-inch touch-screen display for customer order input.

II. System Design

A. *Proposed Restaurant Layout*

The set-up would be a dining area that implements a pick-up and claygo (clean as you go) setting (Figure 2). Since the POS system is capable of completing payment transactions on its own and forwarding orders to the kitchen, the dining area can function without supervision. Customers can line up in front of the POS device and enter their orders, a receipt will come out after the customer has paid. The kitchen will receive the order with a corresponding number, after the food is ready the number is called and the customer may now pick-up the food from the Order Window. After eating the customer will deposit the dirty dinner ware into the Dirty Dish Repository. For surveillance and monitoring a camera can be attached to the raspberry pi and placed on top of the POS device to maximize the field of vision (Figure 3). Floor plans are made using Roomsketcher.com.

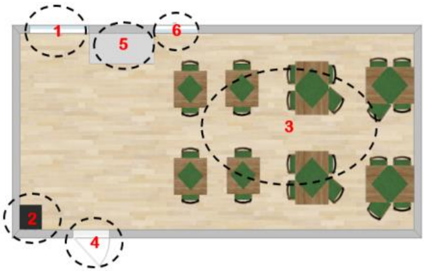


Figure 2: Proposed restaurant layout

Legends:
- [1] Order Window
- [2] POS Device
- [3] Dining Area
- [4] Door
- [5] Condiments and Utensils Area
- [6] Dirty Dish Repository

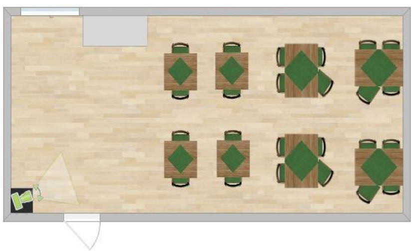


Figure 3: Proposed camera placement

B. *System Flowchart*

The customer can choose between dine-in or take-out, after choosing either dine-in or take-out the customer is directed to a menu screen. The customer can choose his/her desired dish, after choosing the desired dish the machine will check its database of the availability of the dish, if available the customer will be directed to the drinks selection, if the dish is not available the customer is shown a message that the dish is not available and the customer can go back to the menu selection. Now that the customer is sure of his/her orders the customer can insert money into the bill and coin acceptor, if he/she is unsure the customer can go back to the menu selection. Upon payment the machine will determine the amount of money inserted and provide change when needed, if the money is not sufficient the program will not continue until cancelled or provided sufficient cash. Receipt will be supplied upon sufficient payment and the database will be updated and an SMS will be sent to the owner/administrator of the order's information.

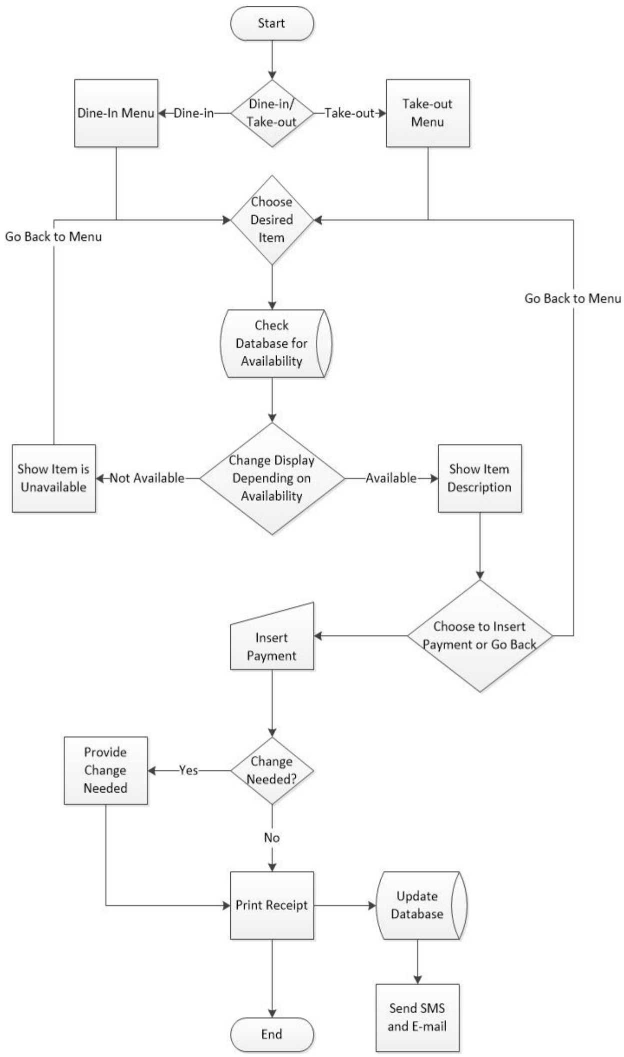


Figure 4: System flowchart

C. *System Configuration*

The whole system will utilize a variety of hardware components including the Raspberry Pi, Arduino, Bill and Coin Acceptor, Coin Changer, GSM module, 7-inch Touch Display, Alarm System, Printer and Camera.

The block diagram shown in Figure 5 shows the interacting components of the system. The Raspberry Pi is the main computer of the whole machine. All of the outputs of the Arduino would be sent to the Raspberry Pi. The Coin changer, touch screen, laser detector alarm, and sensors would all be connected to the Raspberry Pi. The Arduino is the microcontroller responsible for running the Bill Acceptor, Coin Acceptor, and GSM Module. The output signals of both the Bill and Coin Acceptor are passed to the Raspberry Pi 3. The GSM module is responsible for sending messages with every transaction.

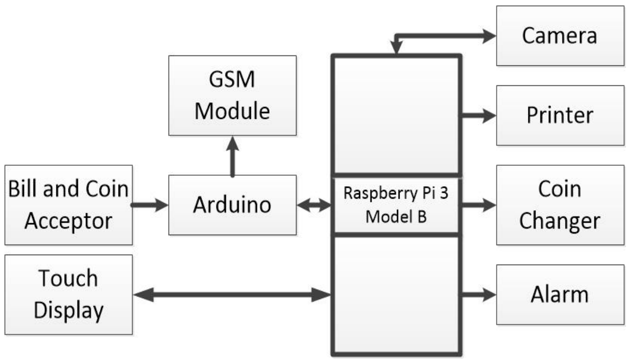


Figure 5: Interacting Components of the System

III. HARDWARE IMPLEMENTATION

A. Electronics

a. Currency Acceptor

The Bill and Coin Acceptors will be responsible for accepting the payments within the transactions. The Bill Acceptor is capable of accepting Philippine currency where other currencies will automatic/ally be voided by the machine. Only 20, 50 and 100 Pesos are programmed to the acceptor since the food would have a low price range thus P200, P500, and P1000 bills are not necessary. The Coin Acceptor only accepts 1, 5, and 10 pesos respectively. Centavos and other currencies will not be accepted by the machine and therefore be automatically voided. Both of the acceptors would be connected to the microcontroller where it would read the signals of the given data. Both the acceptors can be reprogrammed to accept other denominations and currencies when needed.

The Bill and Coin acceptor will run with the use of the PWM pins of the Arduino as the output of both acceptors are pulses. C code will be used to process the data for the Raspberry Pi to read.

Pseudocode:

```
Read payment input
Store payment into variable
    If payment is less than price
        Then Print insufficient
    Else if payment is equal to price
        Then Proceed to next page
    Else if payment is greater than price
        Then Supply change and proceed to next page
    EndIf
```

b. GSM Module

The GSM module will be used to notify the owner or administrator of every transaction of the machine. A GSM SIM800 module will be used and will be run using the Arduino, since existing libraries for running a GSM SIM800 is not easily available for the Raspberry Pi. The GSM module is connected to the TX and RX pins of the Arduino.

c. 7-inch Touch Screen Monitor for Raspberry Pi

The 7-inch touch screen display has an 800 x 400 resolution and connects via an adapter board that handles power and signal conversion. A ribbon cable connects to the DSI port of the raspberry Pi.

d. Coin Changer

A self-made, 3D printed coin changer mechanism is used to supply change. A servo motor will be responsible for driving the mechanism to dispense coins. To control multiple servo motors and better timing control, hardware implementation was used. A servo motor breakout board for the Raspberry Pi is used to correct the motor timings incurred by software implementation on the available PWM pins of the Raspberry Pi. A coin sorter can be used to recycle the coins from the coin acceptor and sort these to the individual coin banks of the coin changer. A pre-set number of change could also be placed in the coin changer banks.

e. Mini Thermal Receipt Printer

The printer only requires a 3.3V – 5V TTL serial output to print. The printer connects to the raspberry pi via the RX pin, since there is only one RX pin available a USB to TTL serial adapter will be used.

f. Laser Tripwire Alarm

The laser tripwire makes use of a light dependent resistor (LDR) to detect the varying light levels. A capacitor is connected in series to the LDR. The capacitor will charge at different speeds depending on the light levels. A laser is fixed on the LDR and any attempt to open the device will trigger the alarm. A buzzer is connected to any available GPIO pin to output sound when the alarm is triggered. A pseudocode is shown below to show how the alarm works.

Pseudocode:

```
Read LDR value
If LDR value is less than light value threshold
    Then turn on alarm
    Else alarm is off
```

g. USB Webcam

A USB webcam can be used in-sync with the laser alarm, the camera can snap a photo of the perpetrator when the alarm is triggered. The camera can also be accessed via its IP by the owner/administrator to view the present activity within the restaurant. The camera can also record video if needed.

IV. SOFTWARE IMPLEMENTATION

The user interface (UI) is run using Kivy and Python. Kivy supports multi-touch for apps.

A. GUI Design

The UI features options for take-out and dine-in and displays a twenty-five item menu. Information of each menu item is available after each order, a drink menu is shown then check-out follows shortly.

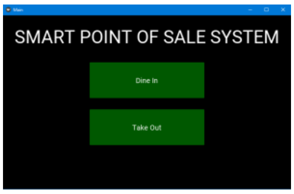


Figure 8: Start

Figure 9: Item menu

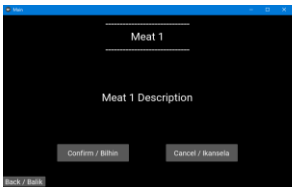


Figure 10: Item information

Figure 11: Drink menu

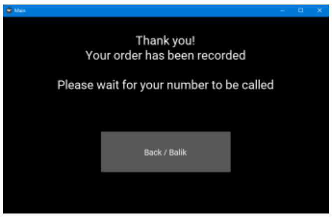


Figure 12: Check-out

B. Database

A local database is used to store inventory details. This can be uploaded to the internet if needed. MySQL is used with python. The database is accessed to determine item availability and is updated after every transaction. E-mail notifications for inventory information can also be enabled as the library for this is easily available.

V. EVALUATION

A. Bill and Coin Accuracy

Bill and coin acceptor accuracy at recognizing and processing denomination values were tested with 3 series of 10 trial tests, a total of 30 trials was done for each denomination. An accuracy of 97% was recorded for processing 50, 10 and 5 and 100% for 20 and 1 peso denominations. The accuracy can be improved with rewiring the data and power cables properly. Electromagnetic interference caused by stray electric signals can cause discrepancies with the output values of the bill and coin acceptor. An improved accuracy of 100% was recorded for all denominations, see Figures 13 and 14 for comparison.

B. Coin Changer Dispense Accuracy

The self-made coin changer's accuracy at dispensing 5 and 1 peso denominations were tested with 3 series of 30 trials, totalling at 90 trials for each denomination. An accuracy of 100% was recorded for dispensing 1 peso denomination and a 98.9% was recorded for dispensing 5 peso denomination. The use of a continuous servo was suspected to be affecting the accuracy. Since continuous servo motors lose their position control, timing issues may arise whenever the servo would start after a recent stop since it does not know where it was. Although the error is minimal, correcting the accuracy is still needed, a stepper motor was used and tested. The stepper motor was able to dispense accurately at 100% for 90 trials. The stepper motor is found to be more accurate although it is more expensive but timing a full 360 degree rotation is easier. For the testing results of the servo motor, refer to Figure 15.

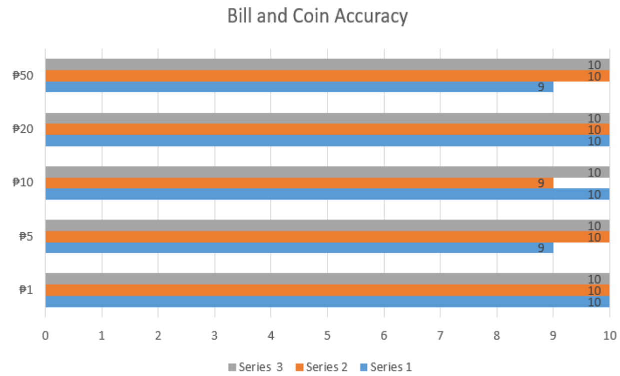


Figure 13: Bill and coin testing results prior rewiring

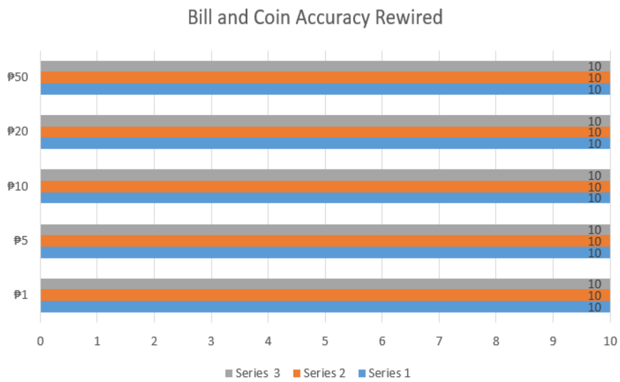


Figure 14: Bill and coin testing results post rewiring

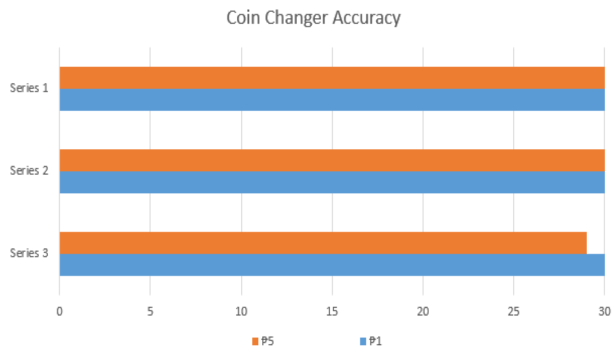


Figure 15: Coin changer dispense testing results

C. GSM Module – e-Gizmo SIM800 GSM/GPRS Dev Kit

The GSM module's efficiency at sending SMS text was tested. After every transaction, the time it would take for owner's phone to receive the text was recorded. The average latency over a 30 trial test was calculated. The SMS module had an average delay of 10 seconds before the message is received by the owner. A sample snippet of the text message is shown in Figure 16.

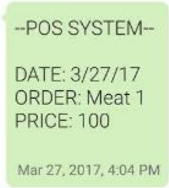
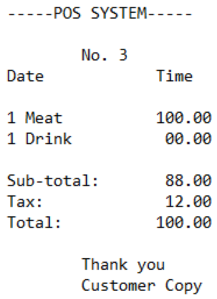


Figure 16: Screen-shot of a received message via GSM module

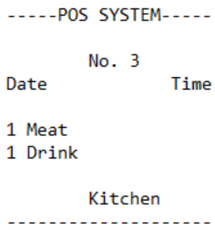
D. Mini Thermal Printer

Two printers are to be used, one for the front of house and the other is for the kitchen. A customer will receive a receipt and a similar receipt will also be printed in the kitchen. Sample snippets are shown below for customer receipt and kitchen receipt respectively, Figures 17 and 18.



```
-----POS SYSTEM-----

        No. 3
Date            Time

1 Meat          100.00
1 Drink          00.00

Sub-total:       88.00
Tax:             12.00
Total:          100.00

        Thank you
        Customer Copy
```

Figure 17: Snippet of customer receipt



```
-----POS SYSTEM-----

        No. 3
Date            Time

1 Meat
1 Drink

        Kitchen
--------------------
```

Figure 18: Snippet of kitchen receipt

VI. CONCLUSION

There is an abundance of POS systems in the world today. Most large scale businesses make use of these POS systems while the small businesses settle for single POS equipment. With the smart stand-alone Raspberry Pi POS system, even small restaurants and other businesses can make use of this technology. The stand-alone feature of the POS system was achieved through the use of the bill and coin acceptor along with the coin changer, these remove the payment process done by personnel running the cashier. Although testing in a real restaurant has not been done, individual testing and integration of the components has been achieved with convincing results.

VII. FUTURE SCOPE

Since the devices used do not have the propriety POS applications and are using relatively affordable peripherals, the system can be adaptable by small scale businesses. The system is easy to set up and manipulate depending on the needs of the owner. Connecting the Raspberry Pi to the internet via WiFI allows for more features to be added and enables IoT capabilities. This POS system is not limited to restaurant use only, depending on the set-up this POS system can be used in salons, retails shops, bakery, etc.

REFERENCES

[1] G. Justin, "What Is A Point of Sale System? A Guide to POS Features", *Software Advice*, 2016. [Online]. Available: http://www.softwareadvice.com/resources/what-is-a-point-of-sale-system/.

[2] S. Preizler, "The Importance of the Point of Sale (POS) System", *linkedin*, 2015. [Online]. Available: https://www.linkedin.com/pulse/importance-point-sale-pos-system-shmuly-preizler.

[3] H. Villareal, "IotPOS a Point of Sale for Raspberry Pi", Hackaday, 2016. [Online]. Available: https://hackaday.io/project/9798-iotpos-a-point-of-sale-for-raspberry-pi.

[4] K. Hensen, "Raspberry Pi 3 Model B Technical Specifications", *Element14.com*, 2016. [Online]. Available:https://www.element14.com/community/docs/DOC-80899/l/raspberry-pi-3-model-b-technical-specifications.

[5] "Kivy: Cross-platform Python Framework for NUI", *Kivy.org*, 2016. [Online]. Available: https://kivy.org/#home.

[6] R. Suehle and T. Callaway, *Raspberry Pi Hacks*, 1st ed. O'Reilly Media, 2013, p. Foreword.

[7] "adafruit/Python-Thermal-Printer", *GitHub*, 2016. [Online]. Available: https://github.com/adafruit/Python-Thermal-Printer.