Project Documentation

Project Title: Cashless Campus

Team Members: Avi Singh, Kevin Jose, Vineet Anand, Prawaal Pandey

Team Mentor: Swapnil Upadhyaye

Basic Aim:

To create a device capable of making payments without involving the use of physical money.

The payments made would be secured via biometric authentication i.e. a fingerprint scanner. The details of the transactions would be stored onto an SD card and also transferred to our servers hosted on the Intranet of IIT Kanpur. The receipt of the transaction would be printed using a thermal printer.

Our version of this device has been customized to purchase coupons at the Mess or items at the Canteen.

Motivation:

Here in the campus, the accounts running in canteen and mess store transactions in pen and paper and there is almost no authentication in making such payments, as the student making the purchase just uses his roll number as identity which can very easily faked by someone else. Also this involves a lot of paperwork on the side of the mess/canteen worker, and may also lead to

some calculation errors on their part due to the human involvement in making bills.

So we decided to use technology to solve these two problems at once, and the idea for the project was born. Initially, just limited to purchasing mess coupons, we saw its potential and extended it to be a cashless payment system, to be used anywhere in our Campus.

Theory:

For building a device capable of making secure payments without cash, we first needed an authentication system, an input/output interface to allow the user to confirm identity, enquire costs and availability, and finalize purchase. Also, we wanted the user to have a printed receipt of the transaction taking place for future reference and to be used as a coupon if need be. The details of the transaction needed to be stored such that they are easily accessible to both the parties involved in the transaction.

And most importantly, we needed a microcontroller to control all these devices and establish coordination between them.

Technologies and Devices Used

Microcontroller:

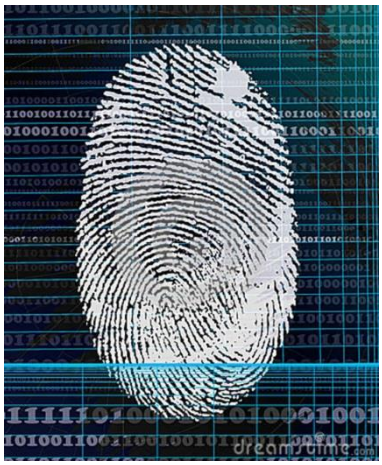
To control so many devices using a vast array of communication methods, we needed a micro controller with a lot of pins, and ample RAM and clock speed for powering the resource intensive components, particularly the LCD.

The micro controller thus selected was the **Atmega256** housed in an **Arduino Mega 2560** for easy programming.

For specifications of the Arduino Mega 2560, visit:

http://arduino.cc/en/Main/arduinoBoardMega2560

Authentication:



We zeroed in on a biometric authentication system since this would allow the user to make payments even when he is not carrying any card with himself.

The only biometric authentication system which could meet our demands without the costs going over the top was **Fingerprint Scanning**.

Fingerprint modules are small, easy to use and quite cheap. The user would only need to press his finger against the scanner and we could identify him.

The fingerprint scanning module that we used was R305, manufactured by the security devices company SFG.

The device uses serial communication via UART to communicate with the micro-controller.

Operation Principle

Fingerprint Enrolling

R305 Module

1. Taking image of a fingerprint and storing it into the image buffer.
2. Converting this image from the image buffer into an array of 256 bytes each that is stored onto a character buffer.
3. Using two such arrays and making a template file of 512 bytes.
4. Storing this template file onto the flash of scanner itself and assigning an ID to each of the template stored.

Fingerprint Matching

1. Taking an image of the finger pressed against the scanner and storing it on the image buffer.

2. Converting the image from the buffer into an array and matching it against the templates stored onto the device.
3. On finding a match, the device returns the ID of the fingerprint template that the finger matches against. Otherwise it returns an acknowledgment package indicating that no match was found.

For more details of the fingerprint scanner:

http://www.vegarobokit.com/index.php?route=product/product&product_id=505

User Interface:

In order to make the user interface as intuitive as possible, we decided to go with one of the most popular technology available in this area: **Touchscreen**.

The device uses a Colour LCD to display information to the user and a touchscreen on top of it for taking in the input from the user, for further processing.

The touch and display device used here is an Arduino Shield manufactured by Itead Studio. It is 3.2 inches (diagonally) long and has a resolution of 320x240 (also known as QVGA). It also has a slot for an SD card that uses SPI to communicate with the thermal printer.

More details can be found at:

http://imall.iteadstudio.com/im120417021.html

The shield is driven by using UTFT, UTouch and some other libraries written by Henning Karlsen. The link for which these libraries is:
http://www.henningkarlsen.com/electronics/library.php

The user manuals provided with the libraries make them very simple to use.

Printing Invoice:

The obvious choice in this area was to use a **Thermal Printer** which is small, efficient and widely used for printing bills. This printer does not

require any ink, but requires a special thermal paper for printing invoices. The device uses serial communication over UART for communicating with the microcontroller.

We have used the Mini Thermal Receipt Printer sold by Adafruit, and they have also written a library to operate the printer with an Arduino.

For details of the printer and links to the library

https://www.adafruit.com/products/597

Storage and Viewing of Transactions:

In order to keep record of the transactions so that a payment could be made on a monthly basis, we decided to use an **SD card** for the offline version of the device and also stored the same data on the **Cloud** i.e. our own servers hosted on the campus Intranet, which could be accessed via Ethernet ports. This also required setting up some **PHP** scripts and **HTML** pages to store the details and to later make them available for viewing.

The SD card reader is mounted onto the LCD shield itself and uses SPI to communicate with the micro controller. The library needed for communicating with the Arduino is included in the official IDE itself. For details of the Ethernet shield and the Ethernet library, visit:

http://arduino.cc/en/Main/ArduinoEthernetShield

To update the databases stored on our servers, we used the official Arduino Ethernet Shield, which also housed a slot for a microSD card. The shield takes data from the Arduino and then sends HTTP requests over Ethernet to out server in order to update the Databases.

To setup a server with a database, we have used a WAMP system i.e. Windows, Apache, MySQL and PHP.

For setting up a Wamp server:

http://www.wampserver.com/en/

We have also made a website which can access the database and allow individuals to check the bills for their running accounts by entering their roll numbers. They can also check which coupons are available in the Mess. Also, the admin user can modify the coupons, their availability and price, while also being capable of looking at the monthly bills of all their users.

Overview:

Once the device is switched on, it connects to our servers hosting the database of items, coupons and bills. From the server, it pulls the items available in the menu of that particular day and meal (For example coupons available in mess on a Monday dinner). It needs to pull these details just once, when the device is switched on.

Now, the device becomes ready for the user to press his finger against the fingerprint scanner. After successfully identifying the identity of the user using a fingerprint scanner, it presents the user with a welcome screen and then takes him to the menu where the user can select whatever coupons/items he needs to purchase.

After successfully selecting the items from the menu shown, the user then proceeds to see his grand total and confirm his order. After confirmation, the invoice is printed via the thermal printer and the details of the transaction are sent via an Ethernet cable to our servers which then make it available to the admin user for billing and to the customer to check his bill.

Our Approach:

- Hooked up the numerous devices together, connecting them to the central Arduino.
- Obtained Arduino libraries for the different devices we were going to use and familiarized ourselves with coding in them.

- Wrote down the codes for using the various devices, and then put them all together in one large program.
- Setup a WAMP system for the online features.

Utility:

- As mentioned earlier, can be used for purchasing coupons at the mess.
- Can be used for purchasing items at the canteen.
- Could be extended into making purchases anywhere, cashless and secure.

Limitations:

- **No Encryptions:** Neither the data on the SD card, or the data on our servers is encrypted yet.
- **Slow:** The device takes about 30 seconds to connect to the servers when switched on for the first time. Also the printer takes over 10 seconds to print a receipt.

Further Developments and Future Scope:

- Adding encryption on servers and SD card to strengthen security.
- Extending the device to be used for payment at any outlet, not just canteens and messes.

Some Useful Links

Arduino.cc

To familiarize yourself with the Arduino, Arduino shields, libraries and coding in the Arduino IDE.

Henningkarlsen.com

To get UTFT, UTouch libraries which are probably the best for LCD shields

Ladyada.net

For tutorials on Arduino programming, and interfacing with other components.

W3schools.org

For tutorials on HTML, PHP, MySQL.

A word of thanks

We would like to thank our mentor Swapnil Upadhyaye, and other co-ordinators Shivendu Bhushan and Sonu Agarwal for guiding us throughout the project and also for keeping us on the track when we were down or stuck.

Most of all, we would like to thank the Electronics Club in general for providing us with this great opportunity to learn in our summers.