Progress Report for QuickPost

Prepared for:

Dr. Andrew Rawicz – ENSC 440W
Steve Whitmore – ENSC 305W
Respected Staff of
School of Engineering Science at
Simon Fraser University

Project Members:

Anmolpreet Singh Bhullar
Jae (Jay) Kim
Jinhong Min
Lestley Gabo
Paola Pilaspilas

Issued:

November 29, 2015

Revision Number:

1.0

Contact: Anmol Bhullar – 604.653.7204 – asbhulla@sfu.ca.

1) Introduction

QuickPost is a smart mailbox and parcel container system that intends on providing feedback and security for mail and package deliveries. Currently, the proof-of-concept is split up into two components: a device that goes inside mailboxes, and an external parcel container. The former is intended for use in apartment complexes or neighbourhoods with community mailboxes, while the latter can be used for any place of residence.

The following sections outline details regarding scheduling and budget, as well as progress made on the software and hardware components of QuickPost. Arduinos are used to drive the system and perform I/O processing while compatible Shields are used to provide email notifications. Furthermore, cameras are used to either serve security or visual feedback purposes. Passive infrared (PIR) sensors and force sensitive resistors (FSR) notify the microcontrollers that couriers have provided input to the system.

2) Schedule

Progress on QuickPost's development can be found below in Figure 1. The tasks in green were completed by the original deadline while tasks in blue were completed with extensions. Tasks in orange are still ongoing. As shown, most tasks were completed on time or with short revisions to the original finish date. Currently, we are in the testing and debugging phase of development.

ID	Task Name	Duration	Start	Original Finish	Actual Finish
1	Research	31 days	Mon 15-09-07	Sun 15-10-18	Sun 15-10-18
2	Mailbox Firmware	38 days	Mon 15-10-05	Mon 15-11-16	Wed 15-11-25
3	Parcel Container Firmware	31 days	Mon 15-10-05	Mon 15-11-16	Mon 15-11-16
4	Mailbox Device Enclosure Model	40 days	Mon 15-10-05	Fri 15-10-16	Fri 15-11-27
5	Parcel Container Model	10 days	Mon 15-10-05	Fri 15-10-16	Fri 15-10-16
6	Mailbox Device Enclosure Built	31 days	Sat 15-10-17	Mon 15-11-16	Fri 15-11-27
7	Parcel Container Built	26 days	Sat 15-10-17	Mon 15-11-16	Fri 15-11-20
8	Parcel Locking Mechanism	17 days	Sat 15-10-24	Mon 15-11-16	Mon 15-11-16
9	System Testing/Debugging	16 days	Mon 15-11-16	Mon 15-11-30	Sun 15-12-06
10	Documentation	56 days	Wed 15-09-23	Wed 15-12-09	Wed 15-12-09
11	Project Proposal	4 days	Wed 15-09-23	Mon 15-09-28	Mon 15-09-28
12	Functional Specification	16 days	Mon 15-09-28	Mon 15-10-19	Mon 15-10-19
13	Design Specification	16 days	Mon 15-10-19	Mon 15-11-09	Mon 15-11-09
14	Progress Report	16 days	Mon 15-11-09	Sun 15-11-29	Sun 15-11-29
15	Presentation/Demo	9 days	Fri 15-11-27	Wed 15-12-09	Wed 15-12-09
16	Port-Mortem	9 days	Fri 15-11-27	Wed 15-12-09	Wed 15-12-09

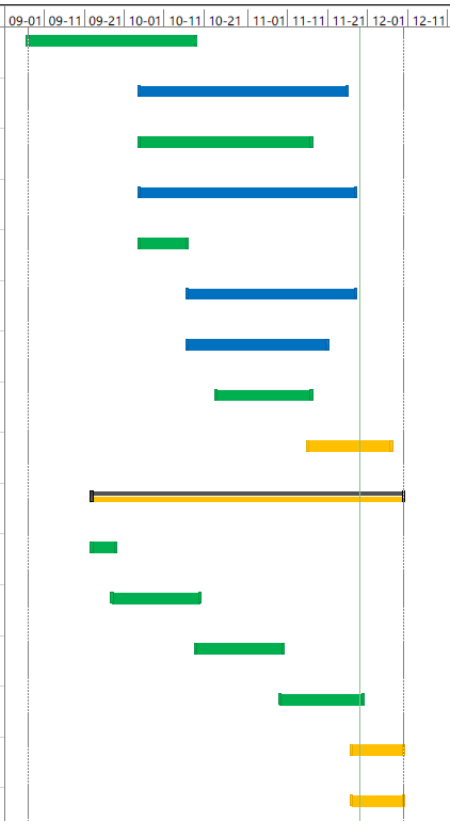

Figure 1: Current task scheduling

3) Progress

The mailbox device utilizes a PIR sensor, camera module, LEDs, and Wifi shield. The code used in incorporating the preceding components with an Arduino has been completed. Furthermore, we have used the SMTP/MIME protocol to send emails with image attachments, and have tested the code with Telus and SMTP2GO email servers. Also, code used to open image files off the SD card and encode in base64 scheme has been successfully implemented. The code used in interfacing the VC0706 camera module is currently being debugged. Moreover, the dimensions of the components were measured to create a SolidWorks model of an enclosure. The model has been completed and sent to SFU ENSC staff to produce a 3D printed enclosure.

The code used in incorporating the parcel container components (FSRs, USB camera, speakers, keypad, and solenoid lock) has been completed and tested separately. Images are captured successfully and emailed as attachments using the Arduino Yun's onboard Wifi module. Furthermore, the container itself has been built and the components are currently being integrated. Other clean-up tasks, such as code optimization and additional features, will be considered if time permits.

4) Remediation

Unfortunately, implementing the original camera (OV7670) bought for the mailbox device was left as the last firmware task and was started later than we would have liked. The main issue is that since the OV7670 outputs raw image data in a time sensitive fashion, code execution would have to be very fast. As a result, we would have needed to use low level port manipulation on the Arduino. Coding and debugging port manipulation code is time costly, and we did not feel confident in writing low level code with the remaining time. Lastly, since the OV7670 module does not have any image processing capability, an additional task would be to figure out how to perform JPEG compression on the raw image data. Because of time constraints, another camera module (VC0706) that is easier to interface with an Arduino was bought as we still would like to provide visual feedback to the mail receiver.

5) Financial

Smart Post Solutions was granted $400 from ESSEF. Any remaining expenses will be covered by the group members. Current expenditures are summarised below in Table 1.

Table 1: Expenditures for QuickPost

Item	Price (CDN)
VC0706 UART VGA Camera	$76.84
OV7670 Camera	$13.35
USB External Stereo	$10.99
Arduino Yun	$131.35

Force Sensitive Resistor	$8.50x2 = $17.00
Lock-style solenoid	$20.00
12V Battery	$30.00
Adafruit HUZZAH CC3000 Wifi Shield	$53.50
Round Tactile Button Switch	$7.80
9V Battery Holder	$5.25
USB Cable A-B	$5.25
4x4 Keypad	$15.00
USB Camera	$25.00
USB Speaker	$25.00
USB HUB	$10.00
Parcel container materials	$50.00
Physical Mailbox	$25.00
PIR sensor	$25.00
Mailbox Enclosure	$100.00
Total:	**$646.33**
Money Received From ESSEF:	**$400.00**
Total Money Spent:	**$246.33**

6) Conclusion

Overall, development for QuickPost has progressed on schedule with few exceptions. The coding for all components for both mailbox device and parcel container is completed and individually tested. We have depleted the funding provided by ESSEF but the team has agreed to cover the remaining costs. The remainder of development time will be used for integration and debugging. We feel confident in delivering a functional system by the demo date of December 9th.