



Design Specification for QuickPost

November 12, 2015

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
8888 University Drive
Burnaby, BC V5A 1S6

Re: ENSC 440W Design Specification for QuickPost

Dear Dr. Rawicz:

The enclosed document is a design specification for QuickPost, a product that will allow users to retrieve their mail effectively. Our goal is to design and implement a mailbox system that sends notifications to recipients through e-mail. Additionally, we hope to improve security and convenience of large package deliveries with a parcel container that is part of the QuickPost system. The parcel container uses a locking mechanism which allows recipients to collect their packages securely. We highly believe that our product will help users save time and relieve worries about stolen packages in communities.

The objective of the enclosed design specification is to provide an overview of QuickPost's design procedures. More specifically, the justification of design choices and how they will meet functional requirements (provided in the functional specification) will be outlined in the enclosed document.

Smart Post Solutions consists of five passionate, detail-oriented, and talented senior engineering students: Anmol Bhullar, Lestley Gabo, Jay Kim, Paola Pilaspilas and Jinhong Min. If you have any questions or concerns about our proposal, please feel free to contact me at 604-653-7204 or by e-mail at asbhulla@sfu.ca.

Sincerely,

A handwritten signature in blue ink, appearing to read "Anmol Bhullar".

Anmol Bhullar
President and CEO

Enclosure: *Design Specification for QuickPost*



Design Specification 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 12, 2015

Revision Number:

1.0

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



Abstract

The following document contains design specifications for QuickPost, a smart mailbox and parcel container system. Detailed design choices for the current approach in implementing the proof-of-concept are outlined in the subsequent sections. More specifically, choices made in selecting appropriate hardware components and software methods will be justified. Preliminary design thoughts for future versions of QuickPost are also explained. Ideally, the design choices made should reflect the functional requirements drafted in [1].

Currently, the proof-of-concept is split up into two components: a device that goes inside mailboxes and an external parcel container. Both components aim to provide notifications to recipients through email. Arduinos are used to drive the proof-of-concept system and perform relevant I/O processing while compatible Shields are used for feedback to the mail recipient. Furthermore, cameras are used to either serve security or visual feedback purposes. Passive infrared (PIR) sensors and force sensitive resistors notify the microcontrollers that couriers have provided input to the system. Lastly, one of the goals of the parcel container is to provide security with package deliveries. As such, a security mechanism is designed and implemented in the hopes of lowering the number of package theft incidents.

The success of the proof-of-concept device will depend on the results of the included test plan. The functionalities of each hardware component as well as the performance of the system as a whole will be tested to ensure that desirable results are achieved. Testing procedures are defined as test cases may need to be repeated to establish confidence in the results.

Table of Contents

Abstract ii

1) Introduction 1

 a) Scope 1

 b) Intended Audience 2

2) System Overview 2

3) Parcel Container 2

 a) Force Sensitive Resistor 3

 b) Keypad 4

 c) Camera 5

 d) Alarm 6

 e) Solenoid Lock 8

 f) Push button 10

 g) Power 10

4) Mailbox Device 10

 a) OV7670 Camera Module 11

 b) PIR 12

 c) Power 14

5) Email notifications 14

6) Safety and sustainability 17

7) Test Plan 17

8) Conclusion 19

References 20

List of Figures

Figure 1: Visualization of QuickPost – mailbox device 1

Figure 2: Block diagram of QuickPost system 2

Figure 3: Hardware interface for parcel container 3

Figure 4: Logarithmic force curve of the FSR 406 (image taken from [2]) 4

Figure 5: Keypad functionality 5

Figure 6: Logitech C170 USB camera (image taken from [3]) 5

Figure 7: High level diagram of Arduino Yun (image taken from [4]) 6

Figure 8: Buzzer flow chart 7

Figure 9: dB vs. voltage applied (graph taken from [6]) 7

Figure 10: 12V Lock-style solenoid (image taken from [8]) 8

Figure 11: Suggested guide on driving solenoids with Arduino (image taken from [9]) 8

Figure 12: Schematic used for solenoid lock 9

Figure 13: Solenoid lock functionality 9

Figure 14: Hardware interface for mailbox device 11

Figure 15: Hardware Connection between Camera Module and Arduino 11

Figure 16: Motion sensor functionality (image taken from [15]) 12

Figure 17: Heat flow capacitor (image taken from [16]) 13



Figure 18: Emails from QuickPost to recipient 14
Figure 19: SMTP commands used in sending emails 15
Figure 20: Multipart MIME message for QuickPost 16

List of Tables

None

Glossary

ASCII	American Standard Code for Information Interchange, a common format for text files in the internet
Darlington Transistor	Current amplifier, a single transistor with a high current gain
dB	Decibel, measurement for sound
FSR	Force Sensitive Resistor, a material that measures the force applied on it
IMAP	Internet Message Access Protocol, users can log into different email clients or webmail, emails are kept on remote servers
IR	Infrared Radiation, an invisible electromagnetic radiation
ISP	Internet Service Provider, an organization that provides internet access
MIME	Multipurpose Internet Mail Extensions, internet standard of email format
OPKG	Open PacKaGe Management is a package manager used to download and install OpenWrt packages
PIR	Passive Infrared Sensor, a sensor that measure infrared light
POP3	Post Office Protocol version 3, a protocol of retrieving email information
Port	Endpoint of communication in networks
Pyroelectric	Material can generate temporary voltage when cooled or warmed
RFC	Request For Comments, official documents for internet specifications, communication protocols, procedures, and events
SMTP	Simple Mail Transfer Protocol, internet standard for electronic mail transmission



Design Specification for QuickPost

Spam	Annoying and unwanted emails
SSL/TLS	Secure Sockets Layer / Transport Layer Security, the protocols that provide secure SMTP transactions
TIP120	A Darlington transistor, used to control medium to high powered electronics
UVC	USB Video Class, a device capable of streaming video or taking still images

1) Introduction

For most apartments and office buildings, mailboxes tend to be placed on the ground level while for some residential neighbourhoods, there is a designated but inconvenient location for mailboxes. Receiving unwanted mail (such as fliers) are common occurrences and may frustrate recipients (especially seniors and the disabled as walking/driving are not trivial tasks for them). Furthermore, there is the issue of missing a package delivery as couriers tend to either leave the package on the ground or notify the recipient of a pickup location. The former presents a security issue while the latter presents an inefficient method of delivery (for both courier and recipient).

Smart Post Solutions hopes to bring an end to the preceding issues with QuickPost. QuickPost aims to notify the receiver that mail has been placed in their mailbox and provide visual feedback by sending images of the mailbox contents. A visualization of the QuickPost is provided below in Figure 1.

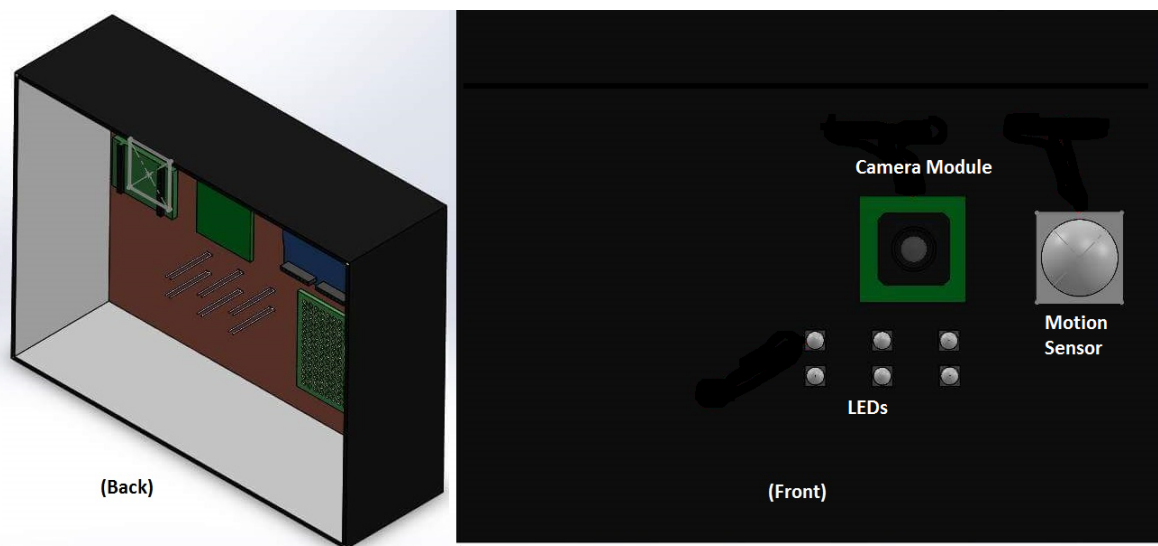


Figure 1: Visualization of QuickPost – mailbox device

The second component of QuickPost aims to provide a secure and hassle-free method of package delivery. With QuickPost's parcel container, couriers can drop packages off even if the recipient is not present. The container will remain locked until it is unlocked by the intended recipient. Additional security measures are implemented in case anyone who is unauthorized attempts to open the container.

a) Scope

The design procedures of QuickPost are outlined in the subsequent sections, and will be used to implement the proof-of-concept. Furthermore, preliminary design concerns are explained for the production version of QuickPost. The following procedures are meant to be coordinated with the requirements explained in the functional specification. Members of Smart Post Solutions should refer to the design procedures during the design and implementation phase.

b) Intended Audience

The design specification is intended for all members of Smart Post Solutions. Ideally, the functional requirements outlined previously should be clearly met in the resulting proof-of-concept device. Furthermore, the design specification is intended towards Canada Post and courier service companies (e.g. UPS, DHL) to explain the inner workings of QP and assist in the implementation of QP devices across Canada.

2) System Overview

As mentioned, the QuickPost system utilizes various hardware components to achieve security and timely feedback for recipients. A block diagram representation of QuickPost is provided below in Figure 2. Green blocks are exclusive to the mailbox device while yellow blocks are exclusive to the parcel container. Red blocks are relevant to both devices.

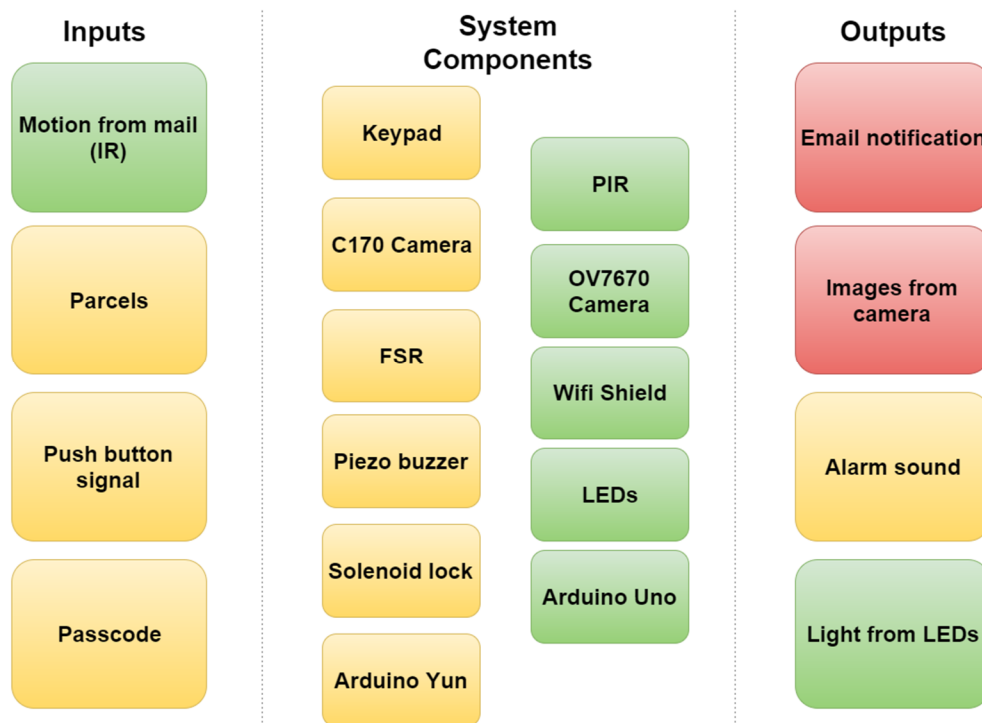


Figure 2: Block diagram of QuickPost system

3) Parcel Container

The hardware connections used to implement the functionalities of the parcel container is provided below in Figure 3. The breadboard will be replaced with a proto-board while each component will be placed appropriately inside/on the parcel container.

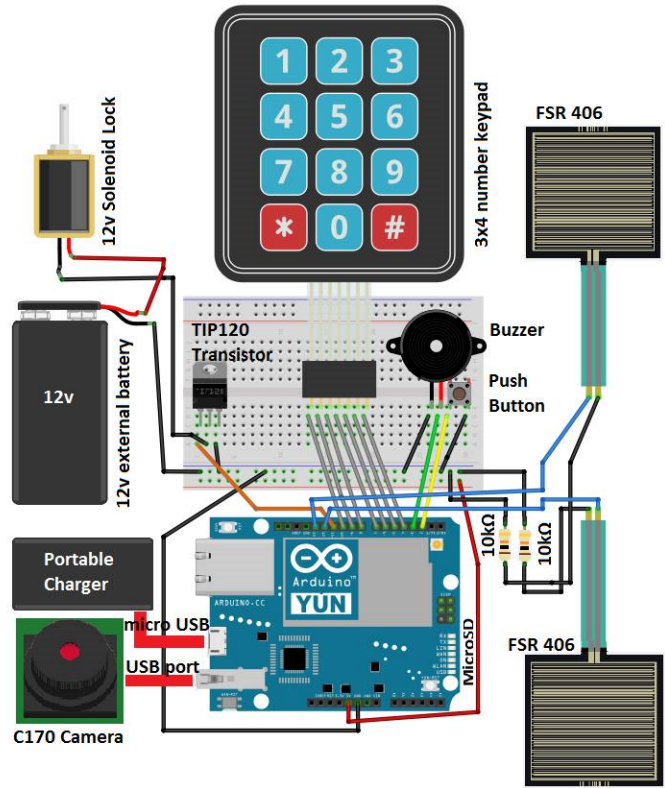


Figure 3: Hardware interface for parcel container

a) Force Sensitive Resistor

A force sensitive resistor (FSR) is a material that measures changes in resistance when pressure is applied. An FSR is typically used to sense and measure the force of the pressure. The sensor will allow the microcontroller to know if there are parcels inside the parcel box. Furthermore, the measurements will determine if the security mechanism is enabled or disabled. An image of the FSR 406 used in the proof-of-concept is provided above in Figure 3.

The parcel container utilizes a “fake floor” where two FSR 406s are evenly spaced on the bottom of the parcel box. The floor needs to be as thin as 1/6” to enable the FSRs to sense the pressure changes. When a parcel is then placed on top of the fake floor, the resistors will sense the pressure from the parcel and relay the measurement to the microcontroller. From that, the microcontroller sums the force measurement from the two FSRs.

When a parcel is taken out of the box, the sum of the forces measured will decrease. Then, the microcontroller checks to see if the security system is on. By default, the security system is ‘ON’ and can only be turned off if the correct passcode was entered on the keypad. If the pressure decreases while the security system is on, the alarm turns on for 60 seconds. Furthermore, the USB camera captures an image (of whoever caused the decrease in pressure), and sends the image to the intended recipient.

The relationship between resistance and force is visualized below in Figure 4.

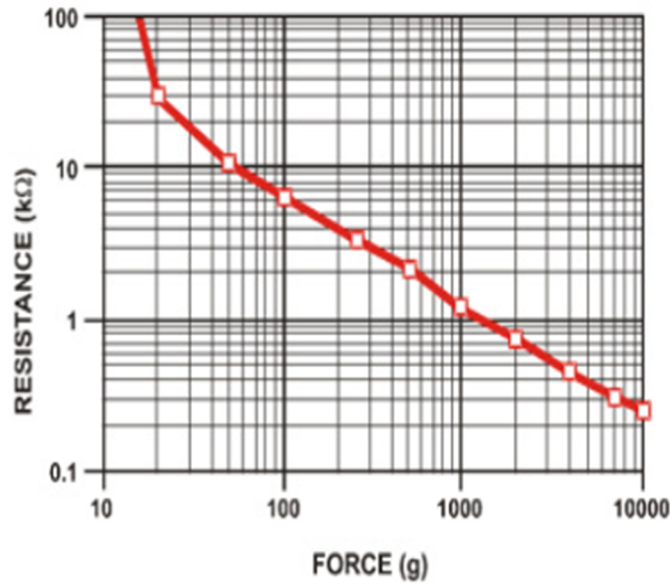


Figure 4: Logarithmic force curve of the FSR 406 (image taken from [2])

As shown, the FSR 406 is able to handle forces ranging from 100g to 10kg. When there is more than one package inside the container, the microcontroller will keep adding the total pressure sensed by the FSRs. From Figure 4, the additional forces will not be linear but that is acceptable as accuracy is not the goal of the FSRs. The function of the FSR is to carefully measure pressure changes, which is enough for the purposes of QuickPost. In other words, the objective is to sense the pressure of parcels coming in and out of the container; parcels that are stacked on top of one another would still be noticed by the FSR. As a result, QuickPost does not need to measure the accurate weight of the parcels.

b) Keypad

The keypad (shown in Figure 3) is a numbered 3x4 flexible and thin membrane with an adhesive back. Users are able to input a password (with the keypad) that unlocks the parcel box. For the proof-of-concept, the password is intended to be 4 digits. Furthermore, the keypad will be attached on the front of the parcel box. A flowchart describing the keypad functionality is provided below in Figure 5.

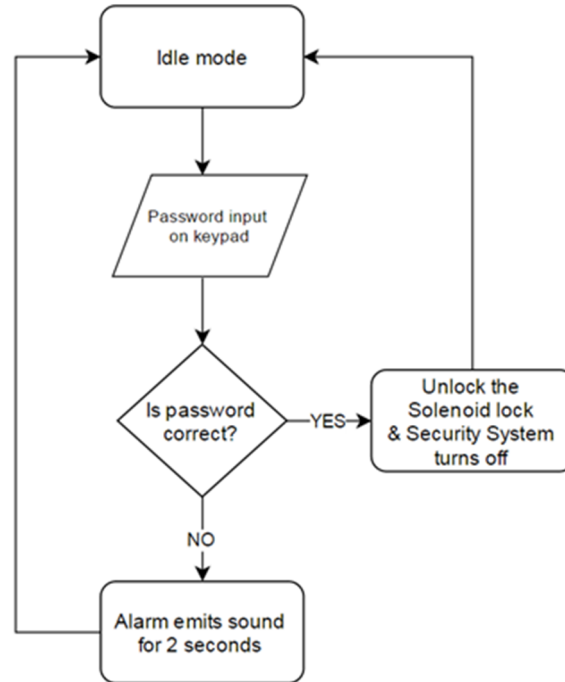


Figure 5: Keypad functionality

As shown, the correct password will unlock the solenoid lock and disabled the security system. The security system will only be off while the lock remains open (the cover of the parcel box is still open). Subsequently, only the authorized users are able to take parcels out of the box without triggering the alarm.

c) Camera

The camera used in taking images of unauthorized users is provided below in Figure 6.



Figure 6: Logitech C170 USB camera (image taken from [3])

For the proof-of-concept, the Logitech C170 was chosen because of its UVC compatibility. UVC compatibility implies that the device is capable of streaming video and taking still images. Another reason for using the C170 (along with the Arduino Yun) is that the UVC package and camera software is easily installed via OPKG (Open PacKaGe Management). A high level diagram of the Arduino Yun is provided below in Figure 7.

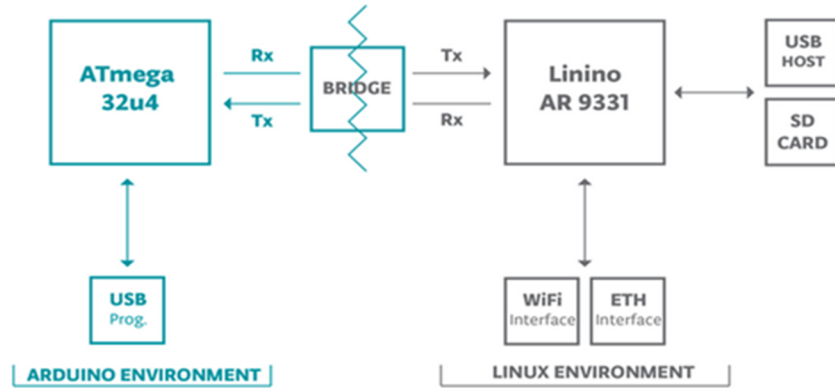


Figure 7: High level diagram of Arduino Yun (image taken from [4])

As shown, the Yun has a built in Linux distribution running on the Linino AR9331 processor. Within the Linux environment, users are able to run Python or shell scripts. Furthermore, there are onboard WiFi and Ethernet transceivers as well as a microSD card slot. The ATmega32u4 microcontroller is another resource for further data processing (if needed). All of the preceding features makes the Yun compatible with the requirements of the parcel box.

To utilize the USB camera, the UVC and webcam software packages must be installed using OPKG. OPKG is a package manager used to download and install OpenWrt packages; installation procedures can be found in [5].

For the proof-of-concept, the specific webcam software installed for the C170 was fswebcam. Next, arguments (such as filename and resolution) can be passed when running fswebcam to take an image. The following command (which will save the image to the microSD card) can be placed inside the Arduino sketch:

```
fswebcam -i 0 -d v4l2:/dev/video0 --jpeg 95 --save test.jpg -S 20 -r 640x480
```

d) Alarm

A buzzer is used for the alarm which provides audio feedback in the security system. For the proof-of-concept, the buzzer with a wide range of operating voltage (3-24V) was chosen. Therefore, the Arduinos are able to power the buzzers with the 5V output pin. A flowchart outlining the buzzer's role in the security system is provided below in Figure 8.

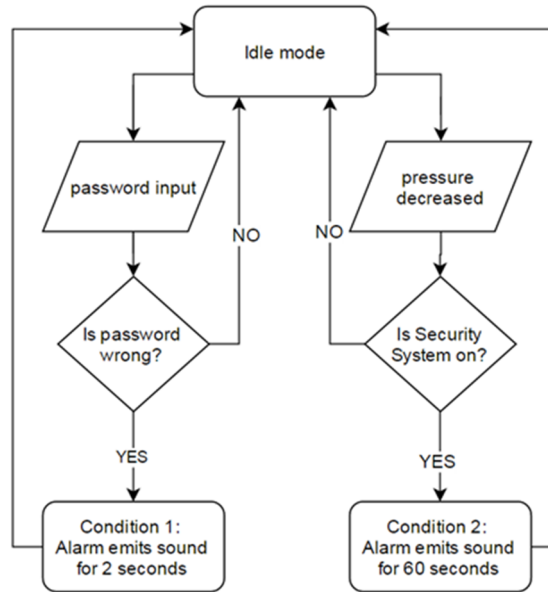


Figure 8: Buzzer flow chart

The buzzer’s default mode is idle, and will only turn on given two situations. If the wrong password is provided, the alarm turns on for 2 seconds as a warning. The second situation occurs when the FSRs notice a pressure decrease while the security system is enabled. Given that the parcel container will be placed outdoors, the output sounds levels should be loud enough to be heard but not cause hearing damage. The relationship between voltage and sound levels of the buzzer is provided below in Figure 9.

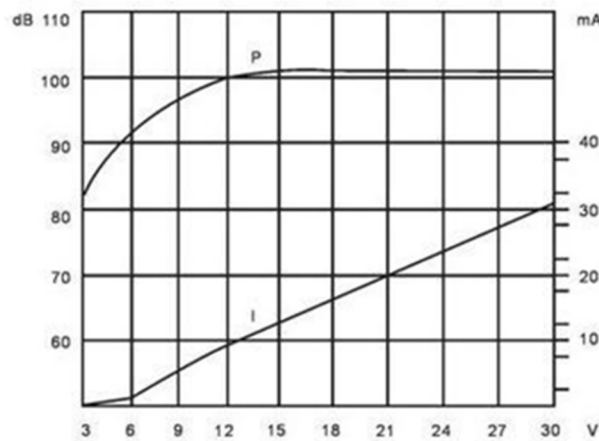


Figure 9: dB vs. voltage applied (graph taken from [6])

Figure 9 shows an increasing relationship between loudness and voltage applied. Powering the buzzer with 5 volts is sufficient as the buzzer could output up to 88 dB, which is significantly louder than a buzzer powered by 3V. Sounds that reach 85 dB or higher are considered “loud” and can cause permanent damage after prolonged exposure [7].

e) Solenoid Lock

A solenoid lock is used to lock the parcel container, and was chosen such that the Arduino is able to electrically control the lock. An example of a solenoid lock is provided below in Figure 10.

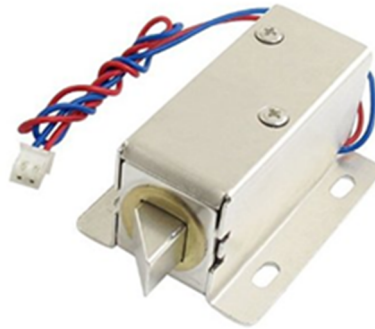


Figure 10: 12V Lock-style solenoid (image taken from [8])

The solenoid lock used is a 12V Lock-style solenoid. As the lock must be powered with 12V, the 5V supplied by the Arduino Yun is not enough. Therefore, an external battery is used to power the lock. Driving the lock requires a high current draw, which is produced by a Darlington transistor. A suggested schematic on interfacing the lock with an Arduino is provided below in Figure 11.

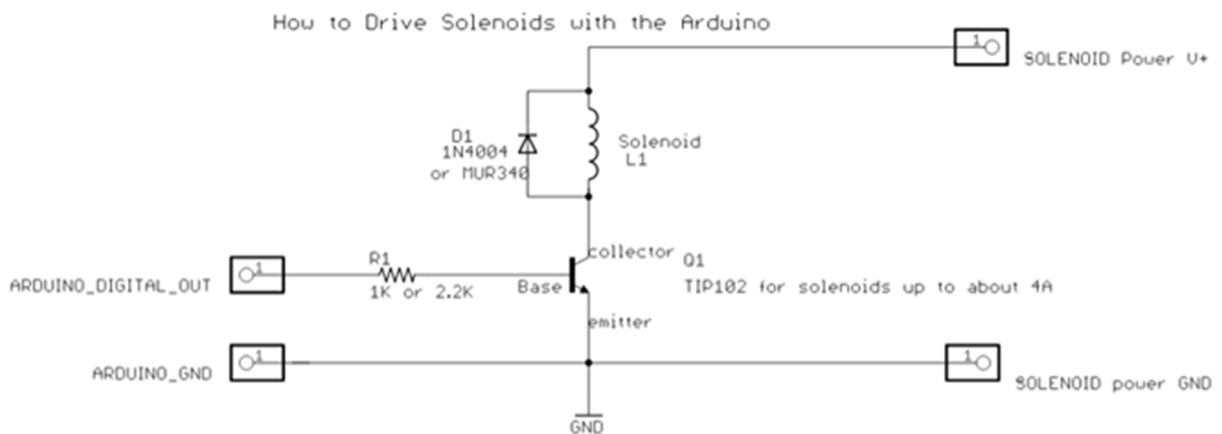


Figure 11: Suggested guide on driving solenoids with Arduino (image taken from [9])

Using Figure 11, the resulting circuit used for the proof-of-concept device is provided below in Figure 12.

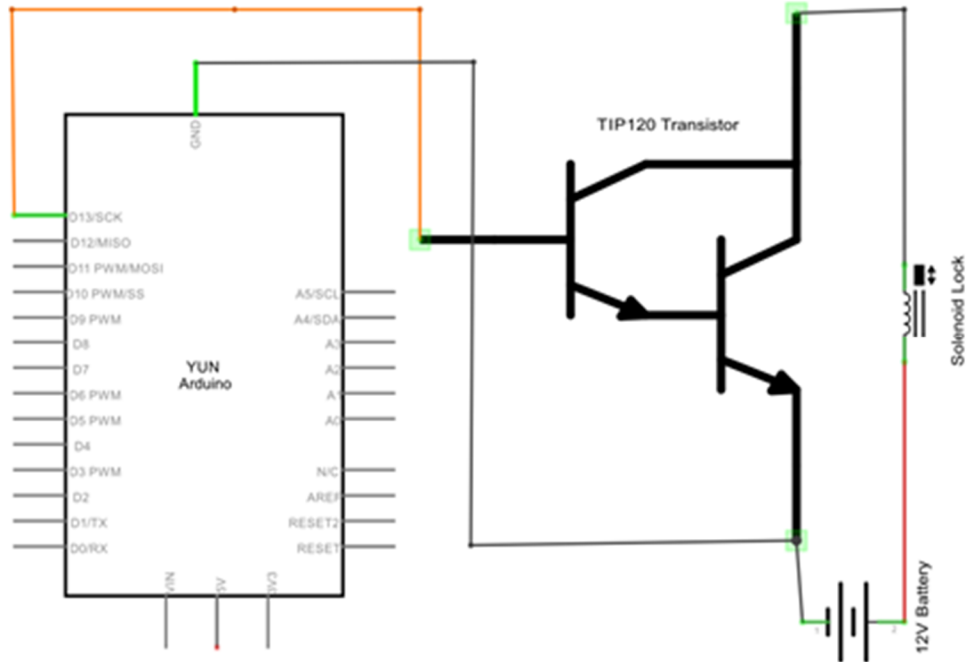


Figure 12: Schematic used for solenoid lock

The difference between the two schematics is that a TIP120 transistor is used instead of a TIP102. According to their datasheets, the difference between the two transistors lies in the collector to emitter breakdown voltage: 100V (TIP102) vs. 60V (TIP120). Another difference is that the TIP102 has collector current of 8-15A while the TIP120 is capable of 5-8A [10, 11]. The TIP120 fails sooner than the TIP102 due to the lower collector current rating. However, for the current version of QuickPost, the TIP120 drives the solenoid lock as needed.

A flowchart describing the solenoid lock logic is provided below in Figure 13.

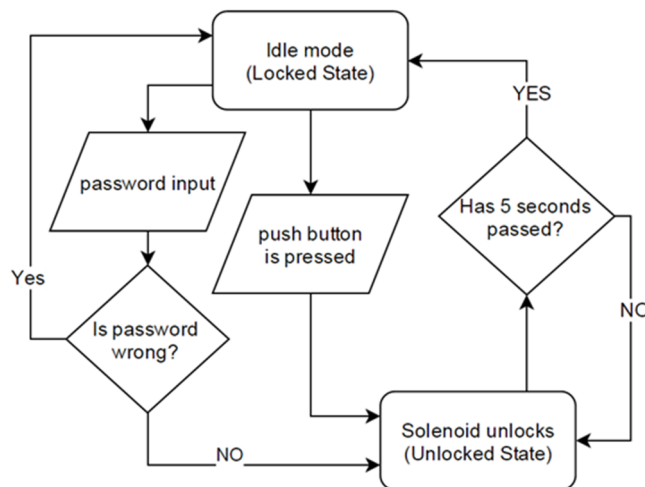


Figure 13: Solenoid lock functionality

As shown in the figure above, the solenoid lock has two states: locked and unlocked. The two states are dependent on the keypad and the pushbutton. By default, the solenoid lock is in the “locked” state and will remain so until it is unlocked by the microcontroller. Lastly, two situations can occur to unlock the parcel container: the correct password is input through the keypad, or the push button is pressed.

f) Push button

The push button is placed on the outside corner of the parcel container and is used to open the cover. When the button is pressed, the switch is closed and a HIGH signal is read by the microcontroller. Conversely, a LOW signal is read when the button is not being pressed. When the button is used, the security system remains enabled. As a result, parcels may be placed inside the parcel box, but not taken out without triggering the alarm. The alarm will activate if the container is unlocked by only using the push button and the FSR detects a decrease in pressure (i.e. parcels were taken out).

g) Power

For the current version of QuickPost, a portable charger is used for the Arduino Yun. A micro USB connection is used to supply the Arduino Yun with up to 2A and 5 volts. Therefore, the Yun will not be underpowered at all while using the portable power supply. The charger used in the proof-of-concept provides the option of using a micro USB connection, which is ideal for the Arduino Yun. Other than that, any portable battery pack that could supply 5V would work for the purposes of QuickPost.

Unfortunately, the solenoid lock used in the proof-of-concept requires 12 volts of power. As mentioned, the lock cannot share power with the Arduino (as the Arduino cannot supply more than 5V). Therefore, an external battery pack is used. The particular pack used was a rechargeable SLA lead-acid 12V battery.

4) Mailbox Device

The hardware connections used to implement the functionalities of the mailbox device is provided below in Figure 14. Again, the breadboard will be replaced with a proto-board while each component will be placed inside an enclosure. The enclosure intends to be attached to the inside wall of the mailbox; orientation will depend on the size and shape of the mailbox.

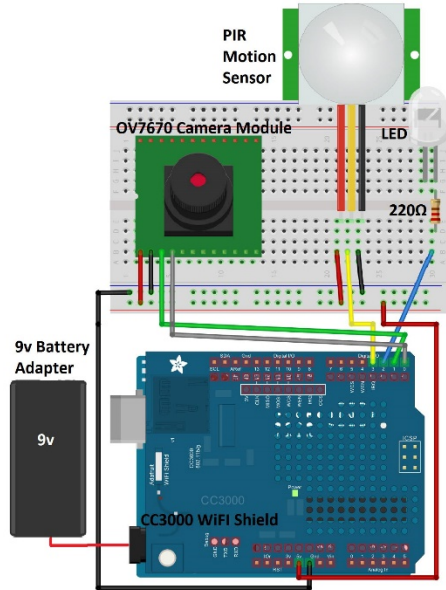


Figure 14: Hardware interface for mailbox device

a) OV7670 Camera Module

The mailbox device utilizes a camera module to capture images of the contents when the motion sensors are activated. The hardware connections between the module and Arduino UNO is provided below in Figure 15.

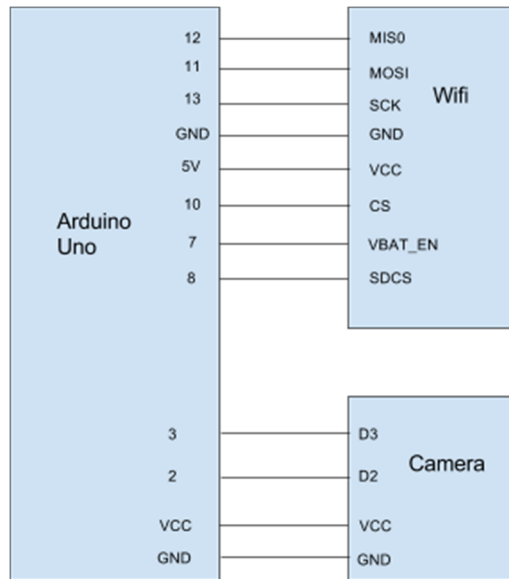


Figure 15: Hardware Connection between Camera Module and Arduino

From the above figure, the SPI pins from the Wifi shield are MOSI, MISO, SCK and VBAT_EN, which goes to pins 11, 12, 13, and 7 (respectively). Once the camera has taken a picture of the contents of the mailbox, the image is saved to the SD card located on the Wifi shield. The chip

select for the SD card (SDCS) is connected to pin 8 of the UNO. The camera uses two serial pins (D3, D2) for sending and receiving pixel data (respectively).

b) PIR

The sensor used to detect any incoming mail is a passive infrared sensor (PIR). PIRs function by “measuring infrared light emitted from objects that generate heat, and therefore, infrared radiation, in its field of view” [14]. A typical application of PIRs is sensing motion from humans, which is applicable to the expected scenario of a courier placing mail inside mailboxes. PIRs, rather than microwave or ultrasonic sensors (which can also be used as motion sensors), are used as they are relatively cheap, physically small, and easy to interface with a microcontroller. A high level diagram of PIR functionality is provided below in Figure 16.

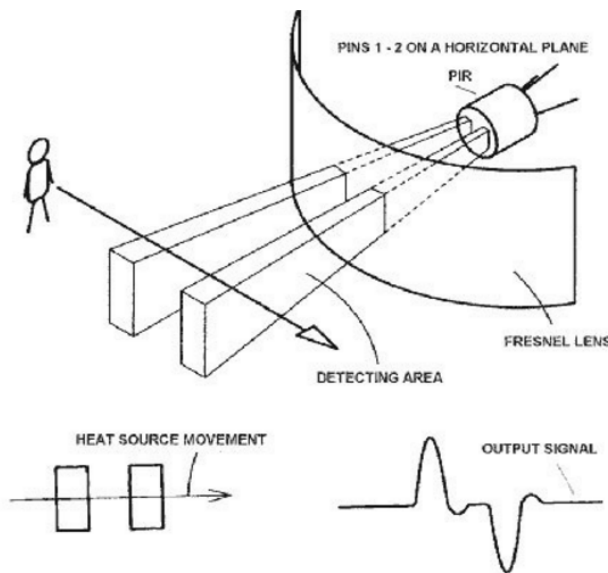


Figure 16: Motion sensor functionality (image taken from [15])

Generally, PIRs have two IR sensitive slots which allows for differential measurements rather than absolute. In other words, the sensor detects the change in IR rather than the current levels in the environment (as shown in Figure 16). As the person walks through the detecting area, the first slot detects the radiated heat and produces a positive pulse. Conversely, a negative pulse is produced when the person is out of the detecting area. Detected movement is considered as the change in measured pulses. For the purposes of QuickPost, the person walking through the detecting area could be analogous to a courier placing their hand inside the mailbox and depositing mail.

The sensor itself consists of a thin film of pyroelectric material (material which is able to produce temporary charges pending temperature changes). As electrodes are used to collect the temporary charges, the sensor is “essentially a capacitor that can be charged by an influx of heat” [16]. A circuit representation of the preceding concept is provided below in Figure 17.

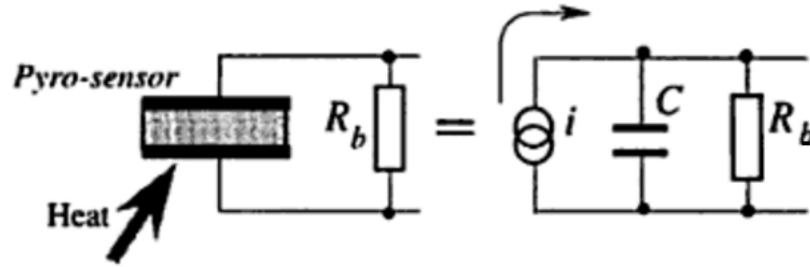


Figure 17: Heat flow capacitor (image taken from [16])

Simply put, all that is left to complete the sensor is a device used to measure the charges and produce a digital signal depending on the measurements. The Adafruit PIR uses a BISS0001 IC to perform the preceding tasks. R_b in Figure 17 represents the input resistance to the BISS0001 IC.

The Adafruit PIR sensor was chosen for the proof-of-concept version specifically for its small size (as the enclosure should be kept as small as possible) and the adjustable potentiometers to change sensitivity and timing. The Adafruit PIR is capable of sensing up to 6 meters with angles up to 110° , which is unnecessary for the purposes of QuickPost. As QuickPost will be placed inside a mailbox facing downwards, being able to adjust the sensitivity towards 15-20 centimeters may be useful in reducing false readings and power consumption.

Furthermore, the length of the output signal (once movement has been detected) can be adjusted accordingly with the onboard potentiometer. Equation 1 below provides the “time duration (T_x) during which the output pin remains high after triggering” [17].

$$T_x \approx 24576 * R_{10} * C_6 \tag{1}$$

According to the BISS0001 datasheet, R_{10} is $10K\Omega$ while C_6 is $0.01\mu F$. Adafruit has added a $1M\Omega$ potentiometer in series with R_{10} , resulting in equation 2 [15].

$$T_x \approx 24576 * (R_{pot} + R_{10}) * C_6 \tag{2}$$

The addition of R_{pot} results in a range of 2.5-250 seconds of trigger time. A longer trigger time may allow for more accurate readings as to when the courier has placed mail inside and when the mailbox door has closed. The production version of QuickPost should have an appropriate R_{pot} value (rather than using onboard potentiometers), as well as a fixed sensitivity that works for general mailbox sizes.

Interfacing the PIR with Arduinos requires one of the digital I/O pins to read the output signal from the PIR. Since the remaining functions of QuickPost (e.g. email, image capture) are predicated on the motion sensor, `digitalRead()` should be called continuously (to read the output signal pin) at the beginning of the software execution until motion has been detected.

Once mail has been placed inside the mailbox, functions to turn on the LED, capture image, and send email notifications should be called.

c) Power

Similar to the parcel container, a 9V battery pack will be used to power the mailbox device. As QuickPost intends to be used in outside environments, portable power supplies are a must. The production version of QuickPost should use lithium-ion or nickel-metal hydride batteries as they are more cost-efficient than alkaline batteries, as well as having a wider operating temperature range.

5) Email notifications

As QuickPost intends to notify mail recipients through email, the Simple Mail Transfer Protocol (SMTP) must be analyzed when considering the software design. QuickPost acts as an email client and uses port 25 to send messages to SMTP servers. A simplified overview of how email is sent from QuickPost to recipient is provided below in Figure 18.

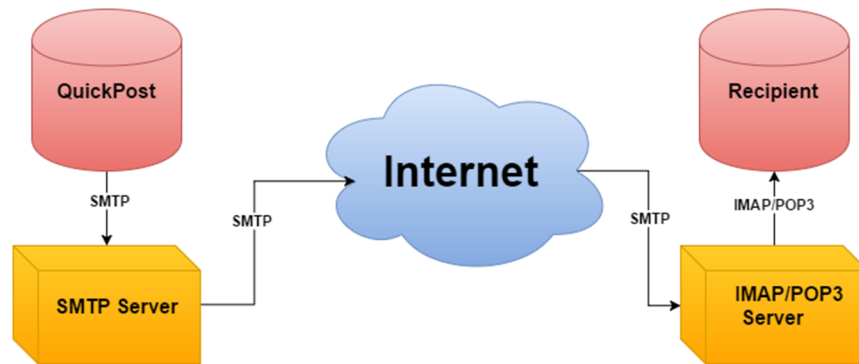


Figure 18: Emails from QuickPost to recipient

To send an email, clients connect to a local SMTP server and sends a formatted message (such that the server can recognize and process the content) to the server. Once the message has been received, it is sent through the Internet where the message hops from server to server until the message’s destination is reached. Once the message reaches the recipient’s local IMAP/POP3 server, the email can be retrieved by the recipient using IMAP/POP3.

Once a mail client is connected to an email server, the server listens for SMTP commands. A flowchart of client-server transactions is provided below in Figure 19. The actual SMTP commands used are capitalised in the square boxes, and were implemented with reference to RFC 2821 [18].

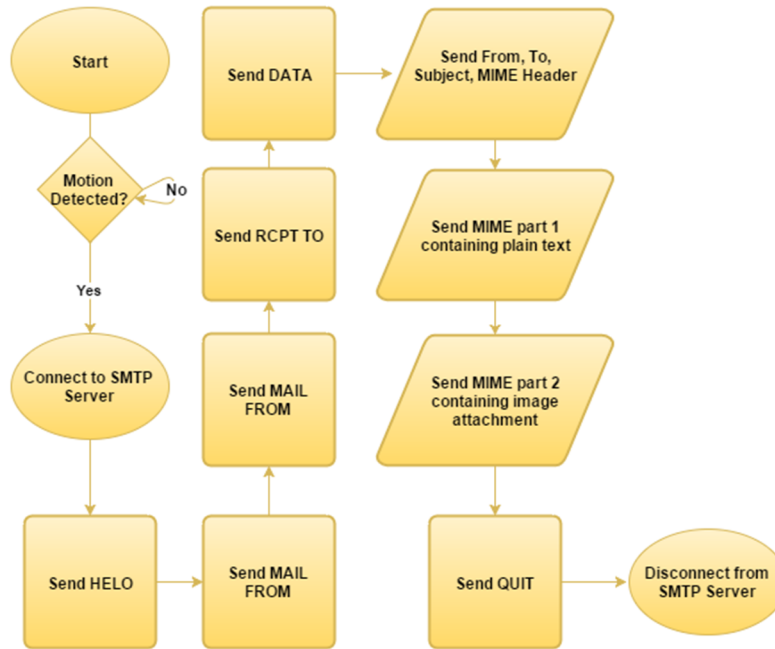


Figure 19: SMTP commands used in sending emails

The server provides a response (consisting of reply code and message) after every SMTP command. Ideally, the server responses should be processed and errors handled appropriately (processing server replies was omitted from the flowchart for ease of reading). The proof-of-concept device assumes that all SMTP commands are acknowledged without error by the server as Arduinos are not capable of handling all possible error codes. However, the production version of QuickPost should have error handling functions implemented. For example, the device should notify Smart Post solutions if the recipient’s email client classifies QuickPost notifications as spam.

As SMTP was originally introduced to handle plain text emails, Multipurpose Internet Mail Extensions (MIME) was created to support non-text attachments (e.g. images) and messages with multiple parts. The multipart message is still delivered using SMTP as MIME is simply an extension of the original protocol. A multipart MIME message containing plain text and an image attachment can be implemented with reference to RFC 2045, and is shown below in Figure 20 [19].



Figure 20: Multipart MIME message for QuickPost

The first 3 lines of Figure 20 are typical of any email, regardless if it is an MIME message or not. As shown, the MIME header tells the SMTP server and the recipient’s email client that the incoming message should be treated as an MIME message. In this case, the Content-Type field notifies that the message has multiple parts with mixed types. Furthermore, the boundary field signifies where each MIME part starts as the server looks for the boundary text prepended with 2 dashes. In Part 1, the Content-Type field tells the server that the contents will be in ASCII text. Conversely, the contents of Part 2 is recognized as an image file in JPEG format. Lastly, the Content-Disposition field notifies that the image should be kept as an attachment rather than an inline image.

For the parcel container, the Arduino has a full installation of Python 2.7. As such, a Python script was used to send an email to the recipient. The script encodes the image located on the SD card and sends it to the receiver’s email if unauthorized access to the parcel container occurs.

Using the Adafruit CC3000 Wifi Shield, sending SMTP commands is implemented by using the provided CC3000 library. Once the Arduino is connected to the SMTP server (by calling connect()), each SMTP command can be sent by calling the println() function. Since the println() function takes a string argument, each SMTP command in Figure 19 (as well as each line in Figure 20) can be passed as strings to println(). The server response from each SMTP command should be parsed for error codes. Lastly, once the email has been sent successfully, the connection to the server is dropped by calling stop(). The Adafruit_CC3000_Client class contains all aforementioned function calls.



One thing to note about port 25 is that it is blocked by most ISPs to prevent users from setting up personal SMTP servers and spreading spam. However, users are allowed to connect through port 25 as long as the connection is to their own ISP's SMTP server as ISP managed servers have spam/virus checkers built in. On top of that, almost all ISPs require SSL/TLS sessions to be started before any SMTP commands are sent.

Arduinos generally do not have enough memory or processing power to handle SSL/TLS. For the proof-of-concept, a solution to both aforementioned issues is to use an SMTP service provider (e.g. SMTP2GO). Services like SMTP2GO use numerous ports for SMTP, which solves the issue of port 25 blocking. Furthermore, SSL/TLS sessions are optional when connecting to SMTP2GO servers. The production version should use a processor that is capable of handling the security protocols.

Lastly, the proof-of-concept has the email server domain name hardcoded. Assuming the production version does not use an external SMTP provider, QuickPost should be able to recognize which ISP it has connected to and use the appropriate domain name when calling connect(). Being able to connect to various SMTP servers without having to manually configure each device would be an efficient way of implementing QuickPost across numerous cities and provinces.

6) Safety and sustainability

Various cases need to be considered when attempting to make QuickPost as safe as possible towards users (e.g. couriers, recipients, setup technicians). Since users will come in close proximity to QuickPost, the device should be properly enclosed to prevent any circuit contact. Even if QuickPost operates at relatively low voltages, exposed PCB contacts is still a significant precaution. Furthermore, use cases of QuickPost will generally take place outdoors. Therefore, special care must be taken to prevent any unwanted circuit behavior due to temperature changes.

Asides from safety concerns regarding circuit failures, security around neighbourhoods where QuickPost is used may also be an issue. Given the value of reclaiming metals inside electronic components, ICs, and PCBs, there may be increased incentive to break into mailboxes and steal devices placed inside. Additionally, theft may be more common around neighbourhoods that use cluster mailboxes as there will be numerous devices in a small area. Lastly, the parcel container could be stolen as the raw material used in building the container may be valuable.

7) Test Plan

Described below is a systematic test plan for QuickPost. The individual hardware and software components will be initially tested. After individual part testing is complete, integration of the two systems (mailbox and the parcel box) will be evaluated.



a. Hardware

Activation of Motion Sensor

Precondition: Microcontroller unit is powered with an operating voltage of 5V.

Step(s):

i) Courier puts mail in the mailbox

Expected Outcome: Motion sensor is activated

Activation of LEDs of the Mailbox

Precondition: Microcontroller unit is powered with an operating voltage of 5V.

Step(s):

i) Courier places mail in the mailbox.

ii) Motion sensors detects the new mail and it is activated.

Expected Outcome: LEDs of the mailbox turns on.

Disabled Lock Mechanism of the Parcel Box

Precondition: The parcel box is locked.

Step(s):

i) User enters the correct passcode on the keypad.

ii) User removes contents from the parcel box.

Expected Outcome: User is able to remove the contents from the parcel box without the activation of the alarm system.

Activated Lock Mechanism of the Parcel Box and Red LED

Precondition: The parcel box is locked

Step(s):

i) User enters the incorrect passcode on the keypad

Expected Outcome: Red LED on the outside of the parcel box will turn on and parcel box will remain unlocked.

Activation of the Load Sensor

Precondition: Pushbutton on the mailbox is activated

Step(s):

i) Courier opens the parcel box.

ii) Courier places new parcel in the parcel box.

Expected Outcome: FSRs are activated and detects the pressure being applied to the FSR of the box and checks for positive weight discrepancies. Alarm is not activated because pressure is applied and not removed due to the new parcel.

b. Software

Activation of the Alarm System of the Parcel Box

Precondition: The parcel box is locked and pushbutton is not activated.

Step(s):



- i) Courier opens the parcel box.

Expected Outcome: Parcel box alarms, takes a picture of the outside, notifies user of activated alarm and sends it via email.

WiFi Connection of the Mailbox of Previously Known Networks

Precondition: Wifi connection is available.

Step(s):

- i) Courier drops in mail in the mailbox.
- ii) Motion sensors are activated.

Expected Outcome: System is able to connect to previously known network, takes picture of the contents of the mailbox and sends it via email.

c. Unit Testing

1. Activate the motion sensors and push button for the mailbox.
2. Verify that the LED turns on in the mailbox.
3. Verify that parcel box can be opened without the need of entering the passcode.
4. Place parcel in the parcel box.
5. Verify that when pressure is applied on FSRs of the parcel box that the alarm does not get activated.
6. Verify that there is a WiFi connection and email server is on for both the mailbox and the parcel box.
7. Monitor if data sent is to camera module of the mailbox and picture is taken of the contents of the mailbox.
8. Verify that email is sent to user with the picture that was taken.
9. Verify that once email has been sent that the motion sensors and LEDs turns off until they are activated again.

8) Conclusion

The design specification is meant to explain the design choices taken in implementing the proof-of-concept of QuickPost. Furthermore, the justifications behind the design choices as well as differentiations between the proof-of-concept and later stages of QuickPost have been outlined. Ideally, the resulting design should satisfy all functional requirements summarised in [1].

The objective of the mailbox device is to detect when couriers deliver mail to the receiver's mailbox. PIRs detect the infrared radiation (e.g. heat given off the courier's hand) and signals the microcontroller to call the appropriate functions. Emails will provide the feedback to the recipient such that the recipients can retrieve their mail in an efficient manner. Furthermore, considerations must be made regarding SMTP server accessibility. For example, some SMTP servers may place a limit on how many emails can be sent from a single client per day. 3rd party SMTP providers can alleviate some accessibility issues as ports other than 25 may be used, as well as higher sent email limits (with additional costs).



Lastly, the goal of the parcel container is to provide a secure and efficient method of handling package deliveries. The current security mechanism consisting of a solenoid lock, alarm, keypad, and camera is designed in a way to increase the chance of getting caught while attempting to steal packages. Similarly, email feedback is provided to alert the recipient their delivery has arrived.

References

- [1] Smart Post Solutions, "ENSC 305W/440W: 2015 Projects," 19 October 2015. [Online]. Available:
<http://www2.ensc.sfu.ca/~whitmore/courses/ensc305/projects/2015/ofunc.pdf>. [Accessed 10 November 2015].
- [2] Trossen Robotics, "FSR specifications," [Online]. Available:
<http://www.trossenrobotics.com/productdocs/2010-10-26-DataSheet-FSR406-Layout2.pdf>. [Accessed 11 November 2015].
- [3] Logitech, "Logitech C170," [Online]. Available:
<https://secure.logitech.com/assets/36165/webcam-c110-gallery-2.png>. [Accessed 11 November 2015].
- [4] Arduino, "Arduino Yun and Linux Environments," [Online]. Available:
<https://www.arduino.cc/en/Main/ArduinoBoardYun>. [Accessed 11 November 2015].
- [5] OpenWRT, "USB Webcam Installation," [Online]. Available:
<https://wiki.openwrt.org/doc/techref/opkg>. [Accessed 11 November 2015].
- [6] Taydae Electronics, "Alarm/buzzer specifications," [Online]. Available:
<http://www.taydaelectronics.com/datasheets/A-875.pdf>. [Accessed 11 November 2015].
- [7] Dangerous Decibels, "Hearing loss decibel levels," [Online]. Available:
<http://www.dangerousdecibels.org/education/information-center/noise-induced-hearing-loss/>. [Accessed 11 November 2015].
- [8] Amazon, "Solenoid lock image," [Online]. Available: http://ecx.images-amazon.com/images/I/41q7nQMnihL._SX342_.jpg. [Accessed 11 November 2015].
- [9] Arduino, "Solenoid lock wiring," [Online]. Available:
http://playground.arduino.cc/uploads/Learning/solenoid_driver.pdf. [Accessed 11 November 2015].
- [10] Fairchild Semiconductors, "TIP102 transistor datasheet," [Online]. Available:
<https://www.fairchildsemi.com/datasheets/TI/TIP102.pdf>. [Accessed 11 November 2015].



- [11] Learn-c, "TIP120 transistor datasheet," [Online]. Available: <http://www.learn-c.com/tip120.pdf>. [Accessed 11 November 2015].
- [12] C. Mathas, "Sensing Motion with Passive Infrared (PIR) Sensors | DigiKey," 28 June 2012. [Online]. Available: <http://www.digikey.com/en/articles/techzone/2012/jun/sensing-motion-with-passive-infrared-pir-sensors>. [Accessed 6th November 2015].
- [13] Adafruit Industries, "Adafruit Learning Systems," [Online]. Available: <https://learn.adafruit.com/assets/511>. [Accessed 9 November 2015].
- [14] J. Webster and H. Eren, "Pyroelectric Materials," in *Measurement, Instrumentation, and Sensors Handbook*, Boca Raton, CRC Press LLC, 1999, pp. 32-110-32-113.
- [15] Shenzhen Haiwang Sensors, "Seeed Studio," [Online]. Available: http://www.seeedstudio.com/wiki/images/2/2f/Twig_-_BISS0001.pdf. [Accessed 10 November 2015].
- [16] IETF, "Internet Engineering Task Force (IETF)," April 2001. [Online]. Available: <https://www.ietf.org/rfc/rfc2821.txt>. [Accessed 9 November 2015].
- [17] IETF, "Internet Engineering Task Force," November 1996. [Online]. Available: <https://tools.ietf.org/html/rfc2045>. [Accessed 9 November 2015].