Key Master (New)

Team: Zerui An (zeruian2), Tingfeng Yan (ty7), Celine Chung (mwchung2)
ECE445 Project Proposal - Spring 2020
TA: Ruomu Hao

1 Introduction

1.1 Objective

How many keys do you carry around? One probably wouldn't feel any inconvenience if there are only a few keys to manage. However, it would be a pain to memorize all the key-lock pairs if one owns many keys. To help these key owners match a key to a lock easily, we plan to design a key holder that is able to recognize a lock and find the key to it.

1.2 Background

According to several polls hosted on online forums[1][2], more than 30% of people own 6 or more keys. Managing a lot of keys is not easy, so these people would surely benefit from a good key organizer. In fact, some companies, like Orbitkey, have already noticed this problem and developed their own solution[3]. However, their products are just high-quality key holders which are unable to match a key to a specific lock.

Our project is inspired by Sp17 group 14's Key Master project. The original project uses RFIDs (attached to locks) to identify each key-lock pair. We noticed a potential problem with this method: it would be hard to replace an existing key-lock pair with a new pair, since one has to remove the RFID from the old lock and attach it to the new one. To simplify this process, we plan to use QR codes instead of RFIDs to identify key-lock pairs. We assume that a random QR code generator is available, which prints out QR codes on small pieces of paper that can be attached to locks. Once a QR code is scanned, the correct key would be selected.

We also made a few changes to the user interface. Instead of having one LED per slot (which may not be a good idea when there are lots of keys to store), we plan to physically label each slot, and have a 7-segment LED display the selected slot number. We will also add a few buttons to support functions like add and delete key-lock pairs.

1.3 High-level Requirements

- The camera should take no more than 5 seconds to scan a QR code
- The correct number should be displayed within 2 seconds after a registered QR code is scanned.
- The accuracy should be at least 90%

2 Design

2.1 Block Diagram

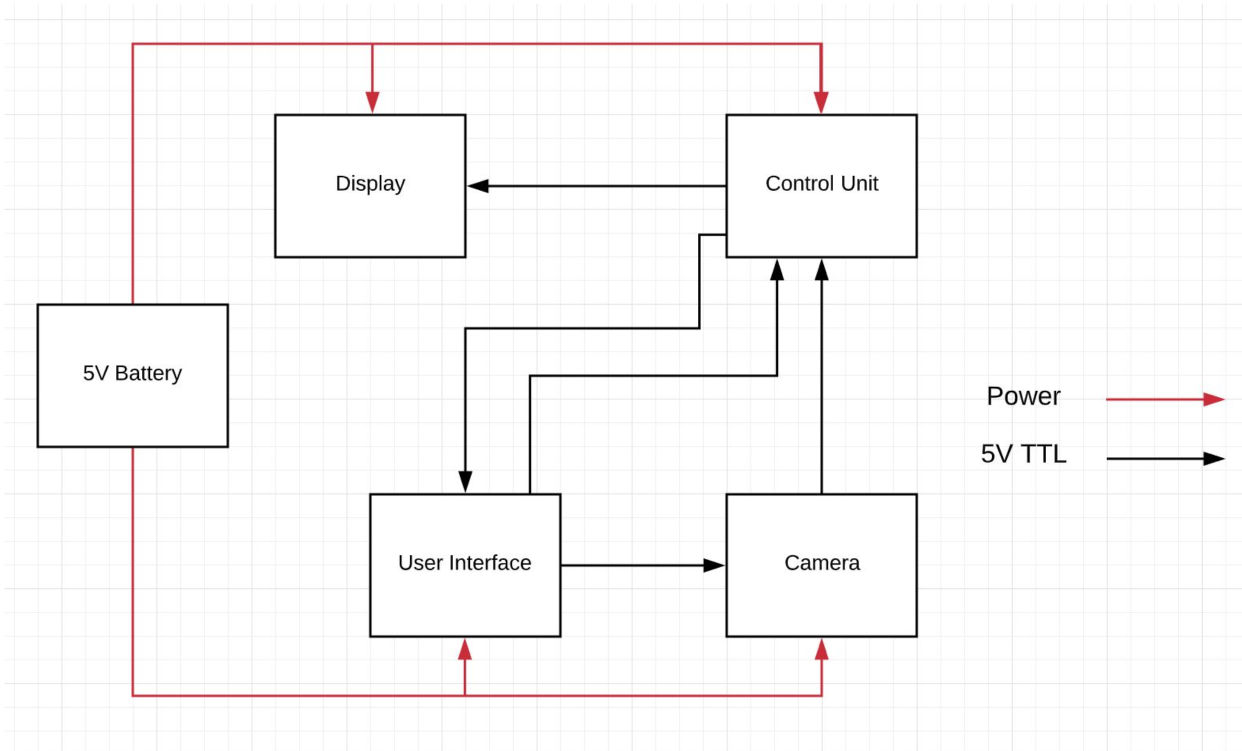

Figure 1. Block Diagram

We plan to use alkaline batteries (with regulator) to supply 5V. Signals in between the blocks will be implemented as 5V TTL.
Key storage block is not shown in the figure because it requires no electric signals.

2.2 Physical Design

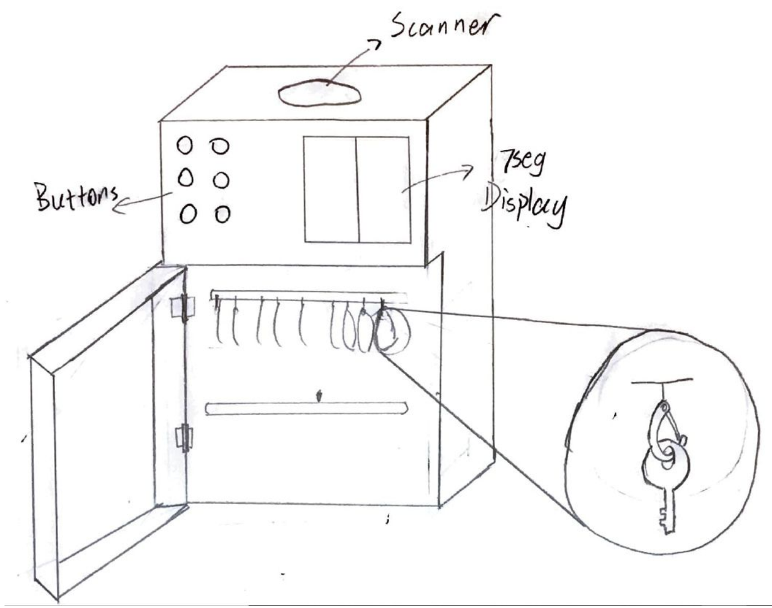

Figure 2. Physical Design

Our product is box-shaped, with a scanner on the top and user interface (including buttons and display) on the front. Keys are kept inside the box, and a number is labeled on each slot. Batteries are installed in the back (not shown in the figure).
Since we are designing a hand-held device, the size should be roughly the same as a wallet.

2.3 Functional Overview

2.3.1 Control Unit

The control unit takes processed image data from the camera, and then check with existing data in the key storage which the user saved in advance. The control unit will send proper signals to the display according to whether it finds the right key or not.

The control unit is also able to receive orders like “New”, “Delete” from the user interface and communicate with the key storage to do the right work. Once the control unit switches its working state, it will also give feedback to the user interface, changing the LED color. For example, blue while it’s finding keys, green while it’s adding a new key, and red while it’s deleting an existing key.

Meantime, the control unit should send the key position it’s working on to the display so that the user can receive correct information about changes.

2.3.2 Camera

The camera is responsible for reading the QR codes.

2.3.3 Key Storage

The key storage is where the keys are stored. Every slot has its unique index.

2.3.4 User Interface

The user interface is supposed to have the following features:
Select1: by pressing Select1 once, the selected key position index will increment by 1. The index starts from 01 and ends with 24. 01 would be selected upon pressing Select1 at 24.
Select2: by pressing Select2 once, the selected key position index will increment by 10. 10 would be selected upon pressing Select2 at 24.
Find: checking image with all existing data to find the right key.
New: Register a new key/lock pair at the selected position.
Delete: Overwrite the selected key/lock pair to 0's.

The user interface also has LEDs indicating the current mode.

2.3.5 Display

This is a 2-bit 7-seg display and it shows the index of the key position that is currently worked on. If the box has a maximum load of 24 keys, the display will be able to present number 01 to 24.
It should also be capable of displaying special characters in special circumstances (e.g. error).

2.4 Block Requirements

2.4.1 Control Unit

During Select mode, the control unit should be able to switch the key position it's working on and send the correct index to the display within 1s.

Upon receiving New from the user interface, the control unit should be able to pair a QR code with an unused key position whose index will be delivered to the display at the same time. If the selected slot is already occupied, the display should become 88 and no new key would be registered.

During Find mode, the control unit should be able to check the bit string which it just received from the camera with all existing index/string pairs. Once the control unit finds the same bit string, it will send the corresponding index to the display. In case it goes through all 24 key positions but finds no matched bit string, it will send 88 to the display.

Upon receiving Delete from the user interface, the control unit should be able to mark the corresponding slot as “unused”.

Upon receiving Photo from the user interface, the control unit should command the camera to take a picture.

Notice that only New and Delete rely on what is selected.

2.4.2 Camera

The camera should be able to correctly interpret any version 1 QR code within 5 seconds.

2.4.3 Key Storage

The key storage must be able to contain 24 keys firmly and have each key position labeled physically.

2.4.4 User Interface

The two Select buttons should not be able to set any number outside the [1,24] range.
The tri-color LED on the user interface should correctly indicate the current state: green for New (add a new pair), blue for Find (find the right key), and red for Delete (delete a pair).
Notice that LED color change indicates the completion of a process.

2.4.5 Display

The display is supposed to show whichever index that the control unit sends to it and the contents will not change until new user requests.

2.5 Risk Analysis

Memory Usage - ATMega itself may not have enough memory to process image data. (Solution: Use external QR code scanner that sends only a string to MCU)
Ordinary microcontrollers like ATMega328 have only a few KBs of RAM, which is not big enough to do image processing. Therefore, we cannot simply connect a camera to the microcontroller and feed all the data to it.

We now have two possible solutions. Instead of using a simple camera, we can use a QR code scanner that sends only a string to the microcontroller (as described in the block requirement for the camera module). The challenging part of this approach lies in the communication between microcontroller and scanner. Most of the scanners we found are not designed for Arduino, which means we need to handle the communication without relying on existing libraries.

Alternatively we can use external SRAM and process the image data on the microcontroller. The extra memory space would make it possible for the microcontroller to process image data. However, The challenge now lies in processing speed. A 8-bit microcontroller using off-chip RAM may end up taking too long to read a QR code.

3 Ethics and Safety

The general safety of our design can be guaranteed because no hazardous or volatile material is used in our design. The mechanical design of our product is also user-friendly. The only safety concern might come from the overheating of some electronic components.

But, in the case when users lose their key-master, it would be worse than just losing a simple keychain. So, from the #7 of IEEE Code of Ethics, “seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors”, we will try our best to solve this problem with our teammates. If we can’t solve the problem by ourselves, we will ask for help from the professor and TA.

We believe that our design is in compliance with the IEEE Code of Ethics and the ACM Code of Ethics and Professional Conduct. By following the code #1 from IEEE Code of Ethics, we will make our design process and the final product as safe as we can and try to prevent any possible harm done to the user and society. Additionally, as stated in the ACM Code of Ethics code 2.9, “Design and implement systems that are robustly and usably secure”, our camera will be intuitive and easy to use for users. This will make sure that our key detecting system is made to make the user's life better.

References

[1] Kenny, “How many keys do you carry?”, *Pricescope*, 08-Oct-2012. [Online]. Available: https://www.pricescope.com/community/threads/how-many-keys-do-you-carry.180672/ [Accessed: 01-Apr-2020]

[2] Mediocrebot, “How many keys are you usually carrying?”, *Meh*, 07-Dec-2018. [Online]. Available: https://meh.com/forum/topics/how-many-keys-are-you-usually-carrying?sort=most-likes [Accessed: 01-Apr-2020]

[3] “Orbitkey - Our Story”, *Orbitkey*. [Online]. Available: https://www.orbitkey.com/pages/about [Accessed: 01-Apr-2020]

[4]”IEEE Code of Ethics”, *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed: 02-Apr-2020]