Smart Lock Security System

CA&OS Project - University of Basel

Simon Kwik, Syarif Hidayatullah

22-01-2018

Table of Contents

Abstract

In this project we have built a smart lock security system using face recognition provided by open CV library. The system can be installed in any electronically controlled door.

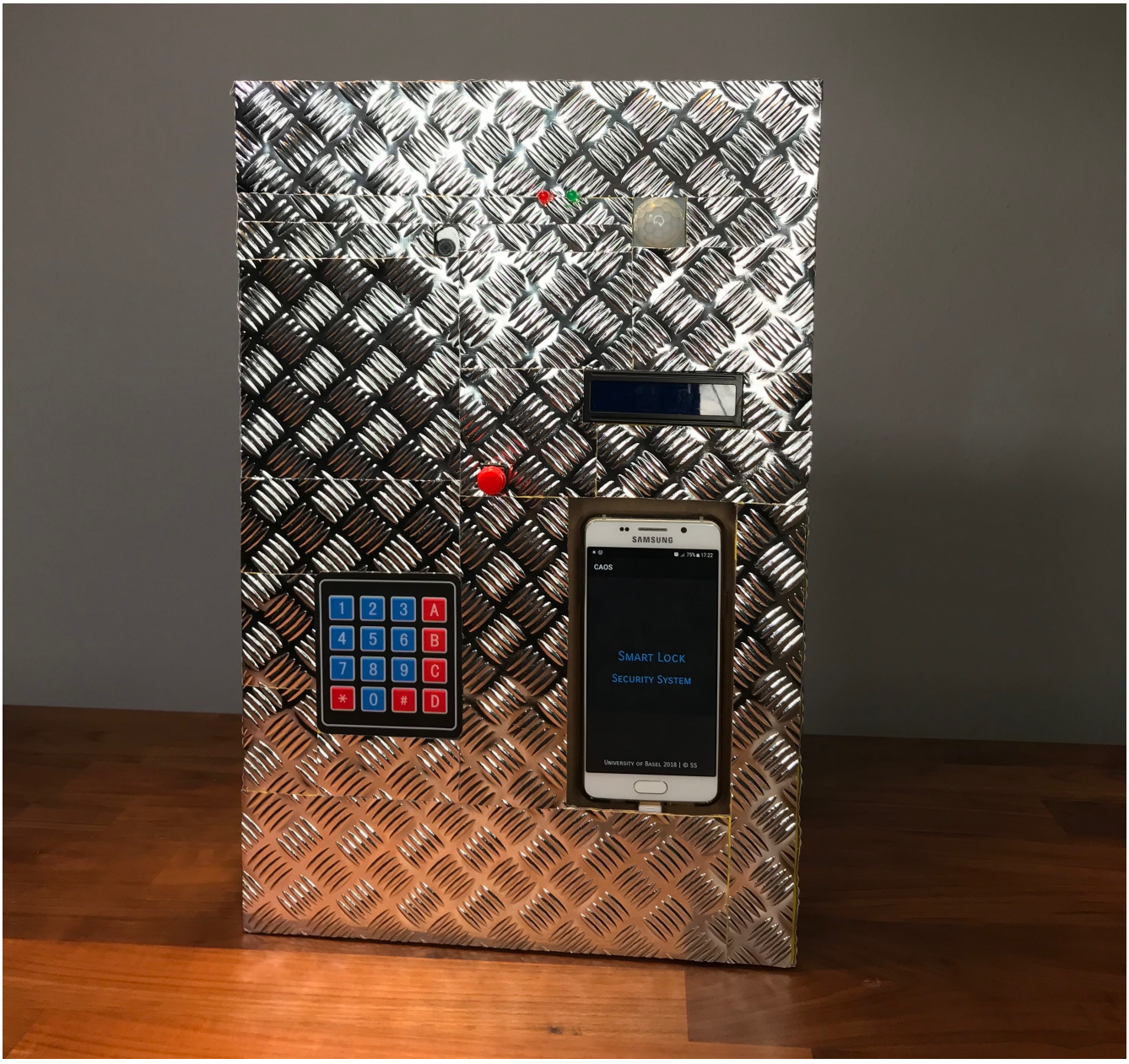

Figure 1. Frontside of Smart Lock Security System.

Introduction

This project is part of the computer architecture and operating system lecture at the University of Basel. The idea behind this project was inspired by our confusion of the swiss mailing system. As you might know, in Switzerland every house has a mailing box called “Milchkästchen” which is used for small package delivery. What stroke us as interesting, is that this box can be accessed by everyone and if a package was stolen out of it, the post service can not be held responsible for the liability of the theft. Due to this unlogical system, we had the idea of creating a smart lock security system.

Our goal was to build a security system, which is easy to maintain, robust and applicable in real life situation. The security system should be able to differ between registered and unregistered faces and depending on that it will decide independently which action to perform.

It is important to mention that the network security is not the focus of our project. Also we only had a budget of 140 CHF that limited the realisation of our project.

If the light is good enough, the system will be able to recognize faces within two seconds on average. When an unregistered face is detected the system will then activate the button and if the button is pressed then a call to the admin will be proceeded. The admin can decide to open the door via an Android app. If the system detects a registered face, the person can then unlock the door by entering the correct password through the keypad. The admin can gain access at any time by using the app.

We have created a flowchart for a better understanding of how our system works. See Appendix.

Background

For the face recognition we are using the open CV library for Python. It is actually written in C but wrapped in Python. We basically get the performance of C with Python syntax. For the face recognition we have to register at least one portrait image of a person and train the program on how to compare registered and unregistered faces. It is basically a supervised machine learning. The face recognition works astonishingly fast. It recognizes faces within one or two seconds if the light conditions are appropriate. We are using a Pi camera without IR filter this is because of its better performance in low light conditions. The camera is triggered by a motion detector attached to the GPIO pins on the Raspberry Pi. As soon as a motion is detected, the camera turns on and captures images in a loop. When a face is detected, the system will then proceed with the actual face recognition. It compares the eigenface of the person standing in front of the camera, with the eigenfaces registered in the system.

If the system finds a match, it will then ask for a password. We used a keypad module attached to an Arduino to receive the input of the user. Arduino and Raspberry Pi are connected via a USB cable. Communication between the two is realized using the nanpy and nanpy-firmware library. The nanpy library is installed on the Raspberry Pi and allows Python to stream commands to the Arduino on which the nanpy-firmware is uploaded. So the Arduino can receive streaming communications from the Raspberry Pi.

Methods

In this section we are presenting the methods which we have used to solve our main problems during our project.

Hardware Architecture

The core of our hardware consists of an Raspberry Pi 3, an Arduino UNO and an Arduino Wemos D1 mini which are working together. Furthermore the hardware includes multiple modules such as:

- LCD
- green LED
- red LED
- button
- 4x4 keypad
- motion detector
- camera
- relay
- simple smartphone

Since the hardware is composed of many pieces which are connected to each other, we needed a case which allowed us to perfectly manage those modules thus, are perfectly organized. We took a parcel box and designed it for our hardware components.

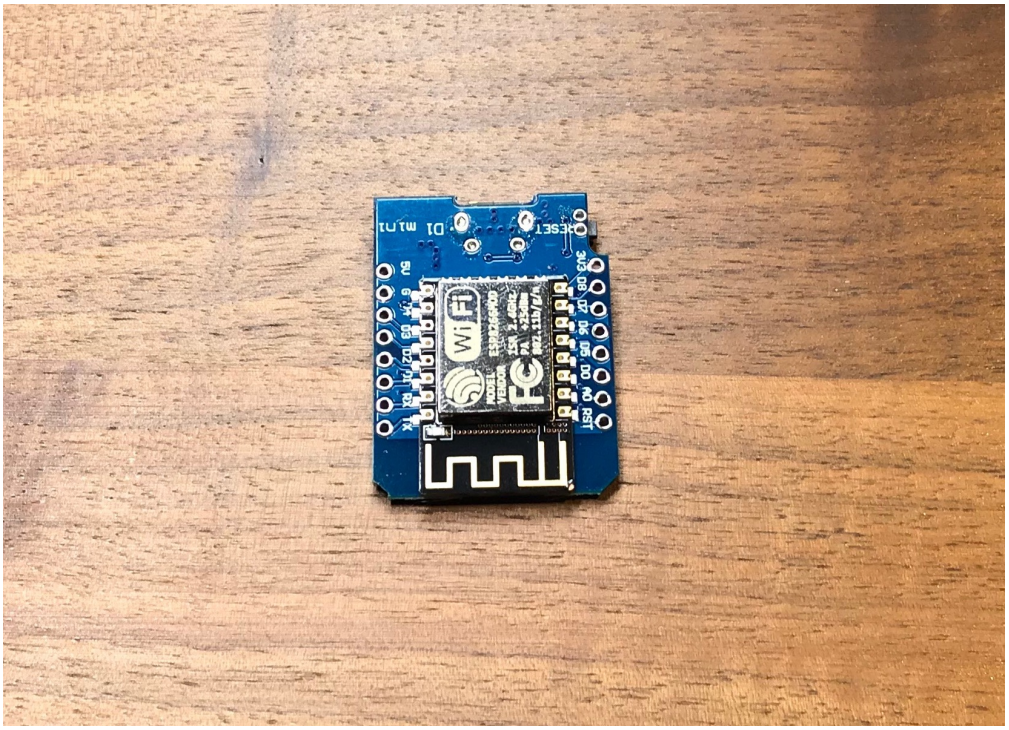

Figure 2. Arduino Wemos D1 mini.

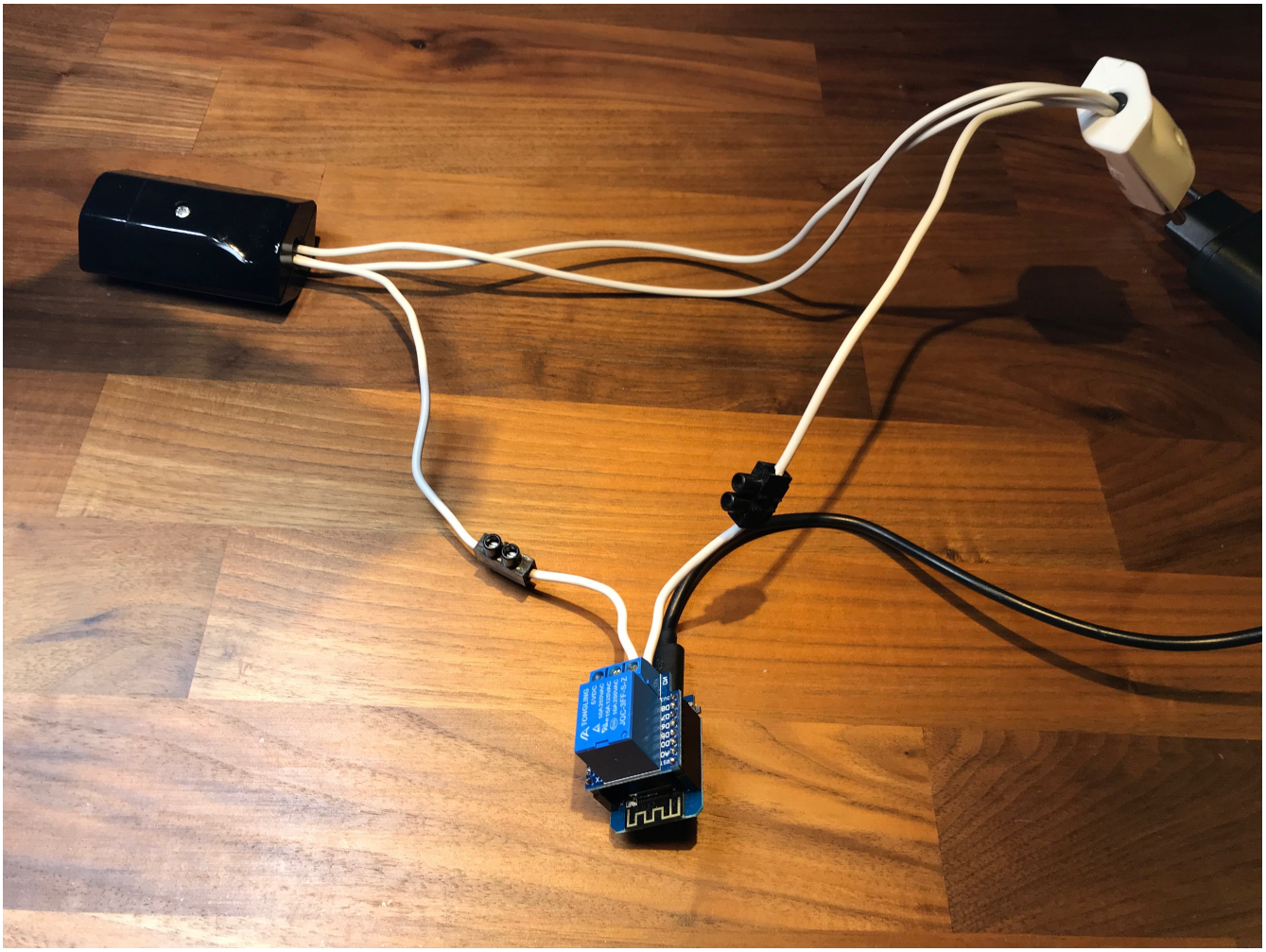

Figure 3. Arduino Wemos connected to relay.

Figure 4. Inside of Smart Lock Security System.

Communication between Raspberry Pi and Arduino

Thanks to the nanpy and nanpy-firmware library, it is possible that the Raspberry Pi can use the Arduino as a slave. Thus we can distribute simple tasks to the Arduino such as receiving commands from the Raspberry Pi and sending back the user's input. Particularly in our project, the Arduino takes the inputs from the keypad module and sends it to the Raspberry Pi, which displays it on the LCD.

Server

The communication between the face recognition system, controller app, and the relay is realized via a remote server written in javascript (Node.js) and Socket IO library. Please notice that in our system, we did not lay our focus on network security. In further semesters we are planning to improve our system with more advanced security features, especially when we have more fundamental knowledge of network security.

The most challenging part in setting up our server client communication, was to find a stable and "ready to use"-socket IO library for the Arduino board (Wemos D1 mini). There are not so many libraries available on the internet. Most of them need extra tweaking to make it work properly.

Controller App

For this project we have built a simple android application which allows users to control the relay remotely. The app is secured with a pin lock. Each time a connection to the server is established, the server will send the state of the relay to the app. In this way, the house owner has an overview on which electric devices are currently on or off.

Intercom

The first idea was to use a GSM shield to allow remote communication between the guest and the house owner. We have conducted some experiments using the shield we bought from Amazon for 20 Euros. We noticed that the quality of the shield was very poor. A stable connection and good quality call was not possible most of the time. We decided to use an android smartphone as an intercom. The difference to an ordinary intercom is the possibility to reach the house owner even though he/she is not in the house.

If a stranger is detected, the call button (red button) is activated. By pressing the call button, a request is sent to the server. The server sends a call command to the intercom which is also connected to the server via socket IO. A phone call is proceeded. The house owner can decide either to open the door or to leave it locked via the app on her/his smartphone.

Results

We tested our project in regards of the following aspects:

- speed
- accuracy
- overall usability

For the speed and accuracy we checked how long it takes until a person's face is detected as stranger or admin and how accurate the results were. Out of ten attempts we observed an average detection time of less than 2 seconds until the face is detected with a 100 percent rate of accuracy under good standard conditions. These includes that the admin's face is captured under good light conditions at the same location where the security system is installed. To test the overall usability we had 10 people testing our system. We observed that even though the testing persons were not familiar on how to use such a system all of them succeeded to use it independently.

That being said, we have achieved both predefined functional and non-functional requirements of the project.

Discussion

Even though the project is only a prototype of a marketable product, we have learned that nowadays such a system can be produced with a reasonable amount of money.

Further improvements are to be made, especially concerning the network security. Every client should get a unique token before it can be connected to the server. Moreover, we think that the user passwords have to be encrypted and saved on a remote database.

In this project we have learned how to combine and customize third party libraries to build a new product. Furthermore we learned how to establish a communication between multiple hardwares/modules.

We have both developed deeper interest on how face recognition works and we hope to gain more fundamental knowledge in this field during our studies.

Appendix

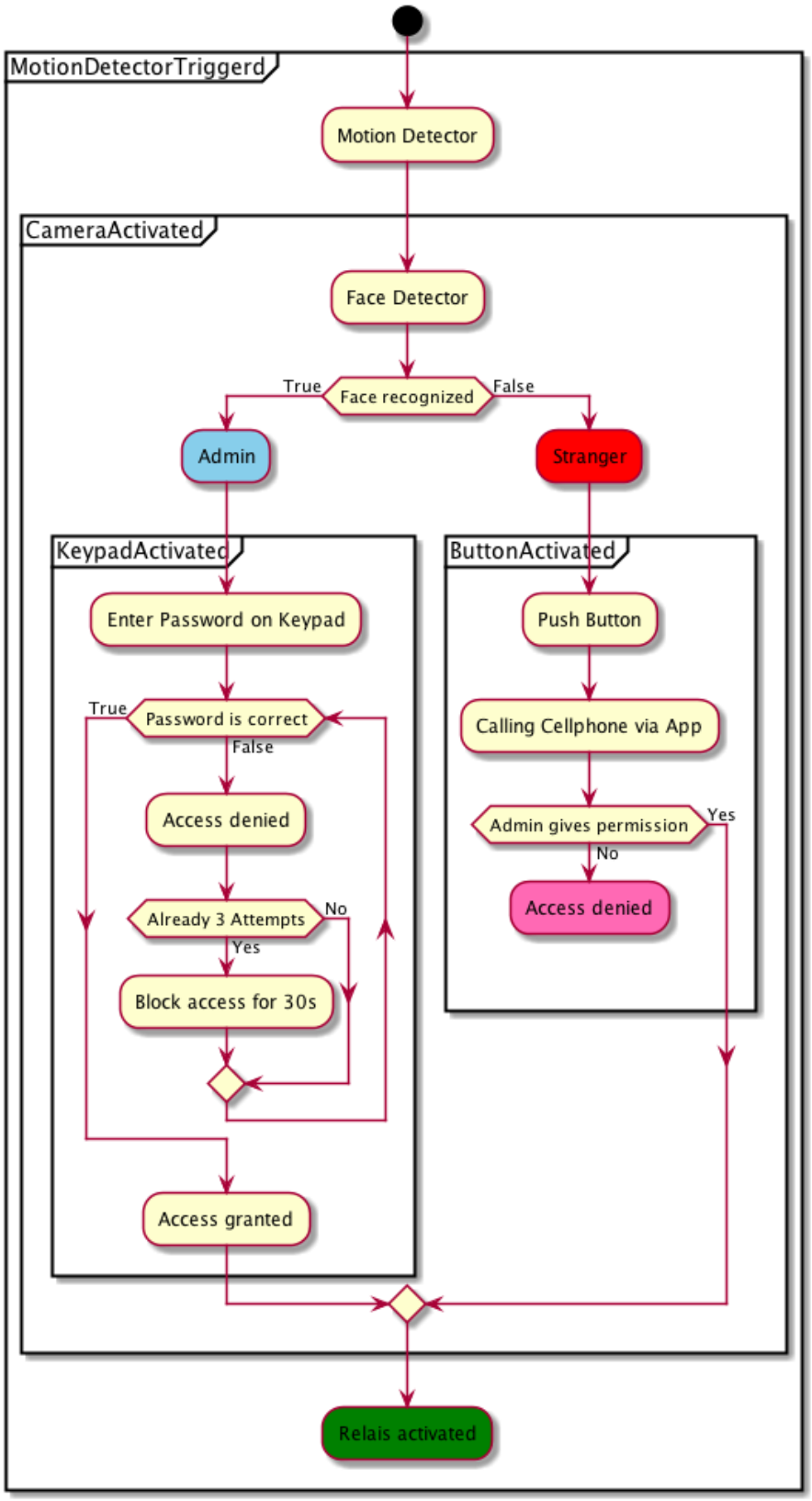

Figure 5. Flowchart of our Smart Lock Security System