

Министерство образования Пензенской области

ГБОУ ПО «Академический лицей №14»

Проект
для участия в VI открытый региональный конкурс
исследовательских и проектных работ школьников
«Высший пилотаж - Пенза» 2023
на тему:

«Система автоматизированного контроля входных групп «FaceGate»

Выполнил:
Филимонов Денис Сергеевич

Научный руководитель:
Трофимов Юрий Александрович

Пенза, 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
ЧАСТЬ 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СОЗДАНИЯ ПРОЕКТА «FACEGATE».....	5
1.1. Особенности использования систем контроля и управления доступом	5
1.2. Сравнение возможных аналогов	6
1.3. Программно-аппаратная платформа Arduino	7
1.4. Язык программирования Python	7
1.5. Использование Serial-порта для передачи информации.....	9
ЧАСТЬ 2. СОЗДАНИЕ ПРОЕКТА «FACEGATE».....	10
2.1. Схема проекта «FaceGate»	10
2.2. Механизм работы «FaceGate»	10
2.3. Программная часть с использованием Python	11
2.4. Программная часть с использованием Arduino	11
2.5. Тестирование «FaceGate»	12
2.6. Оценка себестоимости проекта.....	13
ЗАКЛЮЧЕНИЕ	14
СПИСОК ЛИТЕРАТУРЫ.....	15
ПРИЛОЖЕНИЕ 1	16
ПРИЛОЖЕНИЕ 2	19

ВВЕДЕНИЕ

Проект «FaceGate» представляет собой инновационную систему автоматизированного контроля входных групп, разработанную для обеспечения безопасности и эффективного управления доступом в различные учреждения. Используя передовые технологии распознавания лиц, данная система в реальном времени анализирует и сравнивает лица с базой данных, интегрированной с Google Firebase. Результаты сопоставления отображаются на дисплее для оператора безопасности, предоставляя информацию о посетителе. Решение о разрешении или запрете доступа принимается программой автоматически, что позволяет автоматизировать работу турникета.

Проект написан на языке Python и объединяет в себе функционал библиотек OpenCV, face_recognition и интеграцию с Firebase Admin SDK, обеспечивая высокую степень точности и надежности в процессе распознавания. Графический интерфейс облегчает взаимодействие оператора с системой, предоставляя интуитивно понятные механизмы для мониторинга и управления.

Цель работы - создание и внедрение системы автоматизированного контроля входных групп с использованием технологии распознавания лиц.

Основные задачи включают в себя повышение уровня безопасности объекта за счет эффективного и точного определения личности, обеспечение оперативного принятия решений по управлению доступом на основе данных из базы Firebase, а также удобное взаимодействие оператора с системой через графический интерфейс.

Задачи проекта:

1. Изучить библиотеки OpenCV, face_recognition, firebase-admin.
2. Определить основные принципы работы с Google Firebase.
3. Выполнить анализ функционала и особенностей взаимодействия с Arduino.
4. Разработать и интегрировать графический интерфейс для оператора безопасности.
5. Разработать программу на Python для распознавания лиц.
6. Интегрировать код Arduino для физического управления турникетом.
7. Провести комплексное тестирование всей системы.
8. Провести анализ безопасности и оценку надежности работы системы в различных условиях.
9. Произвести экономическое обоснование проекта.

Объект исследования

Объектом исследования является система «FaceGate», представляющая собой комплексное решение для автоматизированного контроля входных групп.

Предмет исследования

Предметом исследования является технологический комплекс, охватывающий программную и аппаратную составляющие системы «FaceGate».

Практическая значимость проекта «FaceGate» проявляется в нескольких ключевых аспектах:

- повышение безопасности входных групп;
- эффективное управление доступом;
- интеграция с облачными технологиями;
- удобство интерфейса для оператора;

- гибкость архитектуры;
- применимость в различных сферах;
- развитие исследовательских инноваций;
- экономия ресурсов.

Целевая аудитория проекта «FaceGate» включает в себя организации и учреждения, ориентированные на обеспечение высокого уровня безопасности и эффективного управления доступом, в частности:

- бизнес-центры и офисы;
- образовательные учреждения;
- медицинские учреждения;
- транспортные узлы;
- безопасность в гостиничной индустрии;
- различные мероприятия и выставки.

ЧАСТЬ 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СОЗДАНИЯ ПРОЕКТА «FACEGATE»

1.1. Особенности использования систем контроля и управления доступом

В современном мире безопасности на объектах различной направленности уделяется особое внимание, вне зависимости от того, это крупное предприятие или небольшой офис, школа или торговый центр. Наряду с традиционным штатом сотрудников охраны, занимающихся выполнением этих функций, используются различные технические системы, расширяющие возможности и функциональность служб охраны. Система контроля и управления доступом (СКУД) – это элемент системы безопасности, созданный на основе технических устройств и электронных систем управления, объединенных в единую сеть, работающую на специально разработанном программном обеспечении (рис. 1). Кроме этого, система контроля доступа позволяет автоматизировать учет рабочего времени на предприятии и в офисе, тем самым облегчая работу кадровой службы организации [1].

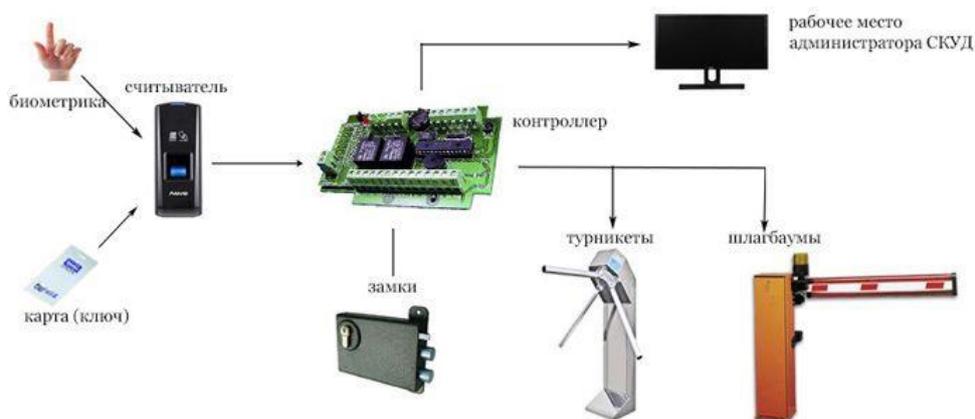


Рисунок 1. Схема основных элементов и конфигурации СКУД

По типу работы и способу передачи информации установки контроля доступа бывают: автономные, сетевые и беспроводные. Основными функциями, определяющим назначение СКУД, являются: разграничение и контроль доступа в определенную зону пространства (помещение), учёт рабочего времени работников и автоматизация работы службы охраны. В качестве дополнительных функций СКУД может быть совмещена с системами видеонаблюдения и охранно-пожарной сигнализации. Также в качестве дополнительной могут выступать способы передачи беспроводного сигнала, для чего используются каналы связи: Bluetooth или Wi-Fi, ZigBee или GSM.

Характерными особенностями СКУД, которые устанавливаются в школах, являются:

1. Эффективность работы:
 - a. наличие простых исполнительных устройств, обеспечивающих лишь недопущение прохода посторонних на территорию или в здание школы;
 - b. возможность прохода ученика без идентификатора с последующей отметкой охранника в ПО системы контроля школы;
 - c. фотоидентификация позволяет верифицировать личность проходящего;
 - d. контроль повторных проходов через исполнительное устройство.
2. Соответствие нормам безопасности в режиме ЧС, регламентированным Правилами пожарной безопасности подобных объектов.
3. Наличие удобных для детей идентификаторов.

4. Биометрия, идентификация без поднесения идентификатора (handsfree), а также информирование родителей.

1.2. Сравнение возможных аналогов

Стоимость оборудования и монтажа СКУД зависит от конфигурации системы, количества точек доступа, а также типов и марок используемого оборудования [1]:

В приведённой таблице указаны цены на различные системы СКУД (таблица 1).

Таблица 1.

Тип системы СКУД	Средняя стоимость, руб.
С доступом по электронному ТМ (TouchMemory) ключу с электромеханическим накладным замком для установки в помещении	4350
С доступом по телефону через интернет с электромагнитным замком для установки на уличную входную дверь	7900
С видеодомофоном и вызывной панелью с доступом по электронному ТМ (TouchMemory) ключу с электромагнитным замком	14000
С доступом по отпечатку пальца и карте с влагостойким электромагнитным замком для установки на калитку/ворота	25300
С онлайн-учётом рабочего времени и доступом по отпечатку пальца и карте с электромагнитным замком для установки в помещении	31800

Распознавание лиц в СКУД может быть организовано с помощью камеры видеонаблюдения, и даже турникетом может управлять сама камера, так как у некоторых камер есть реле для управления большинством преграждающих устройств. В таком случае само распознавание происходит на сервере, после успешного распознавания сервер отправляет на камеру команду открыть турникет.

Основные производители и стоимость СКУД с системой распознавания лиц приведены в таблице 2.

Таблица 2

Производитель СКУД	Средняя стоимость, руб.
ZKTeco (Китай)	от 3 000 до 172480
HikVision (Китай)	по запросу
HiWatch (Китай)	от 22190 до 29990
Dahua (Китай)	от 22290 до 28990
Suprema (Южная Корея)	от 55956 до 197312
ИНТЕМС (Россия)	от 30000 до 120000

СКУД с распознаванием лиц на турникет от компании АНТ [2]:

- на существующий турникет от 75000р. до 300 000р
- «с нуля» – от 165 000р. до 450 000 р.

Таким образом, мы видим очень большой разброс цен на предлагаемые решения. При этом более-менее подходящая система с автоматическим контролем входных групп стоит уже достаточно много.

1.3. Программно-аппаратная платформа Arduino

Arduino – это инструмент для проектирования электронных устройств более плотно взаимодействующих с окружающей физической средой, чем стандартные персональные компьютеры, которые фактически не выходят за рамки виртуальности. Данный микроконтроллер применяется для создания электронных устройств с возможностью приема сигналов от различных цифровых и аналоговых датчиков, которые могут быть подключены к нему, и управления различными исполнительными устройствами [3].

Проекты устройств, основанные на Arduino, могут работать самостоятельно или взаимодействовать с программным обеспечением на компьютере (напр.: Flash, Processing, MaxMSP).

Платы Arduino относительно дешевы по сравнению с другими платформами, их программное обеспечение работает под ОС Windows, Macintosh OSX и Linux. Большинство микроконтроллеров ограничивается ОС Windows. Среда подходит как для начинающих пользователей, так и для опытных и основывается на среде программирования Processing.

Сфера использования данной платформы на современном этапе практически безгранична. С помощью неё можно спроектировать множество различных систем, которые смогут помочь человеку оптимизировать затраты природных ресурсов и уменьшить стоимость потребляемых услуг. Так на базе микроконтроллеров работает система «умный дом», автоматические вентиляторы и светофоры, миниметеостанции, квадрокоптеры и т.д. Спектр применения широк.

Основные преимущества платформы Arduino: открытая схема оборудования, открытый код программы, простая и удобная среда программирования, возможность функционирования на различных видах систем, программирование, подключение и питание может осуществляться одним USB-кабелем, приемлемая цена оборудования.

В нашем проекте будет использоваться плата Arduino Uno, т.к. она обладает необходимой надежностью и функционалом (рис. 2).



Рисунок 2. Плата Arduino UNO

1.4. Язык программирования Python

Язык программирования Python является, пожалуй, одним из самых популярных. Используется в веб-разработке, DataScience, системах автоматизации, приложениях – сферы его применения поистине безграничны. Но главное преимущество Питона – это его низкий порог входа [4]. Иными словами, обучиться программированию на этом языке может практически каждый. Python обладает простым и понятным синтаксисом. Даже далекому от программирования человеку будет несложно разобраться в строках кода. Запуск кода на Python возможен на любом ПК, интерпретаторы для него есть везде. А это значит, что разработчик

может сразу видеть результат. Однако, будучи интерпретируемым языком, Python уступает по скорости компилируемым языкам (поскольку не работает непосредственно на компьютере).

Интерпретируемость сильно упрощает тестирование и перемещение с платформы на платформу небольших блоков кода. Совместимость с большинством существующих ОС делает Python универсальным языком программирования. Он прекрасно подходит для тех, кто только начинает программировать. Данный высокоуровневый язык позволяет разработчику сосредоточиться на сути и целях, а не конкретных шагах и процедурах. По этой причине писать скрипты на Python оказывается быстрее, чем на многих других языках. Его сходство с естественным языком (английским) сильно облегчает его изучение, если сравнивать Python с другими языками программирования. Разработчику легко запомнить синтаксис «питона» и начать читать код.

Помимо небольших прикладных скриптов, на Python можно писать и крупные коммерческие приложения, думая только о том, как обеспечить им необходимый функционал (все мелкие задачи решаются «под капотом», и программисту не нужно беспокоиться о них).

Надежность делает язык программирования Python очень популярным. Ключевым фактором при его выборе является гибкость этого языка программирования и его соответствие парадигме объектно-ориентированного программирования. Поэтому его активно применяют в Data Science и машинном обучении.

Преимущества языка программирования Python:

1. Простота изучения. Язык программирования Python легко выучить с нуля, поскольку он ясен, логичен и читабелен. Сложный синтаксис и запутанные правила — это не про “питон”. Если вы владеете английским, то у вас всё получится.
2. Высокая востребованность языка. В 2020 г. в рейтинге IEEE Spectrum Python занял первое место. Он не только широко распространён, но и применим в разных видах программных продуктов. Зарботки Python-программистов очень неплохи, особенно в веб-разработке, машинном обучении и Data Science.
3. Использование Python в технологиях искусственного интеллекта и машинного обучения. С помощью Python удобно производить сложные расчеты, поэтому он и пригодился в машинном обучении. А библиотеки, отвечающие за нейронные системы, пришлись очень кстати в ИИ.
4. На языке программирования Python можно написать программу для ПК, мобильное и веб-приложение (сегодня в вебе и мобильной разработке очень много таких продуктов).
5. У языка Python имеется множество расширений для создания микро- и макропроектов.
6. Открытый исходный код и бесплатная доступность языка на официальном сайта. Кроме того, каждый может внести свой вклад в развитие Python и добавить новые фичи в него.
7. Язык независим от ОС и с одинаковой эффективностью работает на главных платформах - Linux, Windows, macOS.

Недостатки Python:

1. Медленная работа. “Питон” плохо подходит для создания высокопроизводительных приложений, лучше сразу выбрать другой язык.
2. Крайне мало возможностей для процессорных оптимизаций: их сводит на нет особая модель памяти языка.

Для нашего проекта данные недостатки являются несущественными, поэтому язык Python был выбран в качестве основного для написания приложения для распознавания лиц и взаимодействия с базой данных.

1.5. Использование Serial-порта для передачи информации

Serial port (серийный порт, последовательный порт или СОМ-порт -, communications port) – это последовательный интерфейс с двойной направленностью. Вся информация по этому порту передается с шагом равным одному биту, т.е. данные передаются бит за битом, в отличие от параллельного порта. Несмотря на то, что в некоторых других интерфейсах как, например, в Ethernet, FireWire и USB, применяется последовательный обмен данными, название «последовательный порт» закрепилось за Serial-портом [5].

Данный порт, в сравнении с другими «последовательными» технологиями, обладает отличительной особенностью: в нем отсутствует какое-либо временное требование между 2 байтами. Временные требования существуют только между битами одного байта. Величина, обратная временной паузе между битами одного байта, носит название «baud rate» (скорость передачи). Кроме того, в данной технологии нет такого понятия, как "пакет". Другие технологии "последовательной" передачи данных (X.25, USB или Ethernet), используют "пакеты", также в них существуют и жесткие временные требования между битами одного пакета.

Ранее последовательный порт применялся для подключения терминала, позднее его использовали для подключения модема или мыши. В настоящее время его применяют в качестве средства соединения с источниками бесперебойного питания, а также в качестве средства связи с аппаратными средствами разработки встраиваемых вычислительных систем, спутниковыми ресиверами, кассовыми аппаратами и приборами систем безопасности. Несмотря на морально устаревший формат, Serial-порт еще нередко используется в промышленном и узкоспециальном оборудовании, а также на некоторых современных компьютерах.

Последовательный порт активно вытесняется интерфейсом USB и FireWire. Однако имеются специальные стандарты эмуляции последовательного порта над USB и над Bluetooth. Программная эмуляция порта широко используется и по сей день. Так, практически все мобильные телефоны сегодня эмулируют внутри себя СОМ-порт и модем, с целью реализации доступа компьютера к сети Интернет.

Главное преимущество данной технологии состоит в простоте подключения. Главными недостатками данного порта являются его низкая скорость, большие размеры разъемов, а также высокие требования к времени отклика операционной системы. Также, в данном стандарте наблюдается высокое количество прерываний (одно прерывание на каждые 8 байт).

Для нашего проекта основным каналом связи между аппаратной частью FaceGate и программной является именно Serial-порт т.к.:

- Arduino эмулирует именно Serial-порт;
- он прост в реализации и программировании;
- его недостатки несущественны для проекта.

ЧАСТЬ 2. СОЗДАНИЕ ПРОЕКТА «FACEGATE»

2.1. Схема проекта «FaceGate»

Визуализация структуры проекта включает взаимосвязь между компонентами, такими как камера, ПК, Arduino и турникет, демонстрируя поток данных и команд (рис. 3).

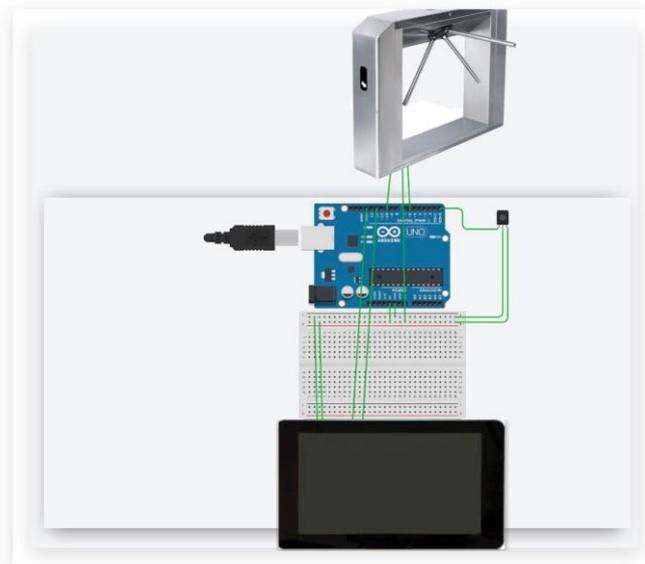


Рисунок 3. Схема проекта «FaceGate»

Для обеспечения взаимодействия между программной и аппаратной частями проекта «FaceGate» используется Serial-порт. Python-код, ответственный за обработку изображений и управление системой, отправляет команды и данные на Arduino через Serial-порт. Ардуино, в свою очередь, принимает эти данные и управляет турникетом в соответствии с предоставленными командами. Этот метод обеспечивает стабильную передачу информации между программой и аппаратурой, обеспечивая эффективное управление системой.

2.2. Механизм работы «FaceGate»

Когда в поле зрения камеры попадает лицо, программа захватывает его и проверяет, есть ли соответствующее изображение в базе данных. Захват лица происходит благодаря библиотеке cvzone, в частности, библиотеке opencv.

Программа сравнивает лицо с базой данных, используя библиотеку firebase_admin. Если лицо совпадает с изображением в базе данных (что проверяется библиотекой face_recognition), то на дисплей выводится информация о пользователе, такая как количество посещений, имя, фамилия, должность и так далее.

Затем, с помощью библиотеки pyserial, подается сигнал на Arduino, открывающий турникет. Срабатывает задержка, не позволяющая человеку отметить дважды за один промежуток времени (например, дважды за день).

2.3. Программная часть с использованием Python

Разработка Python-скрипта для обработки изображений, распознавания лиц с использованием OpenCV и библиотеки face_recognition, а также отправки команд на Arduino через Serial-порт (рис. 4).

```
import pickle
import time
import cv2
import os
import face_recognition
import numpy as np
import cvzone
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
from firebase_admin import storage
from datetime import datetime

cred = credentials.Certificate('serviceAccKey.json')
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://faceattendance-87f81-default-rtdb.firebaseio.com/',
    'storageBucket': 'faceattendance-87f81.appspot.com'
})

bucket = storage.bucket()

cap = cv2.VideoCapture(0)
```

Рисунок 4. Фрагмент Python-скрипта для распознавания

Полный код Python-скрипта представлен в Приложении 1.

2.4. Программная часть с использованием Arduino

Разработка Arduino-скетча, который принимает данные от Python-скрипта через Serial-порт и управляет турникетом в соответствии с полученными командами (рис. 5).

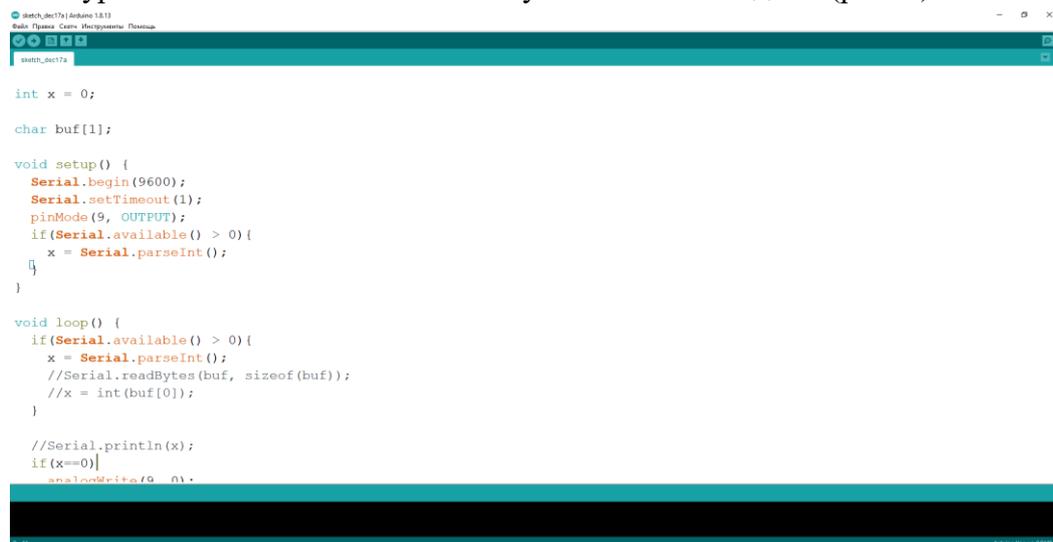


Рисунок 5. Фрагмент Arduino-скетча

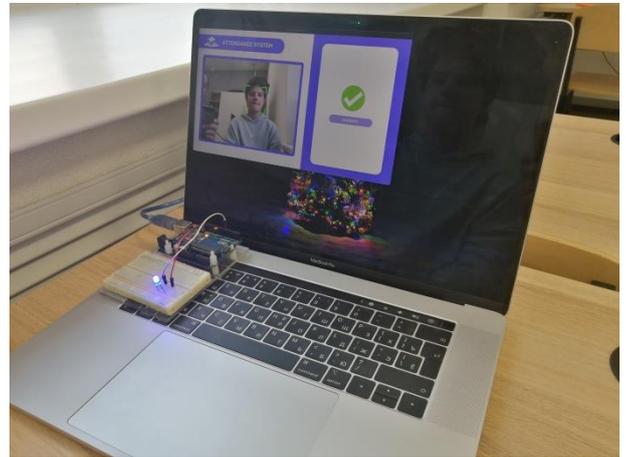
Для эмуляции работы турникета на данном этапе проекта используется светодиод, который загорается синим цветом при допуске посетителя (турникет открыт), или остается в выключенном состоянии (турникет закрыт). В дальнейшем к работе светодиода будет добавлен механизм передачи управляющего сигнала на турникет (зависит от конкретной модели турникета).

Полный код Arduino-скетча представлен в Приложении 2.

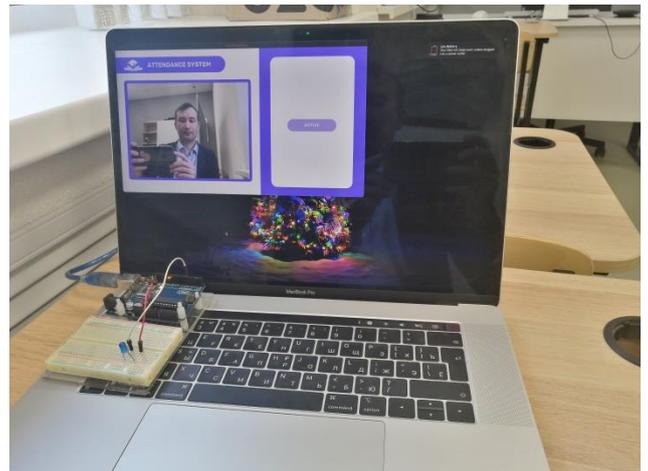
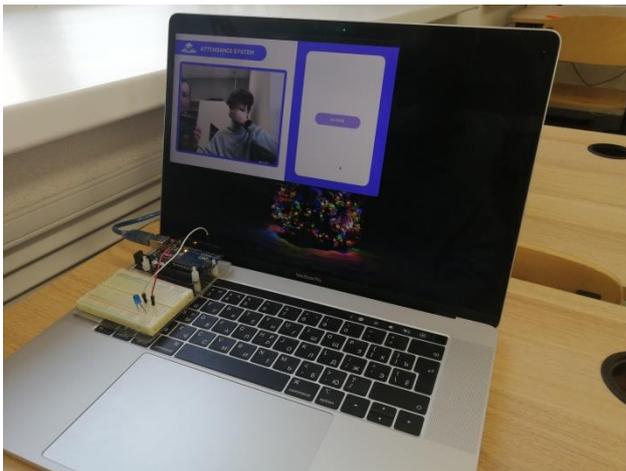
2.5. Тестирование «FaceGate»

Тестирование проекта «FaceGate» проводилось в различных условиях, включая проверку точности распознавания лиц, стабильность работы Serial-порта, адекватное управление турникетом и обработку возможных ошибок.

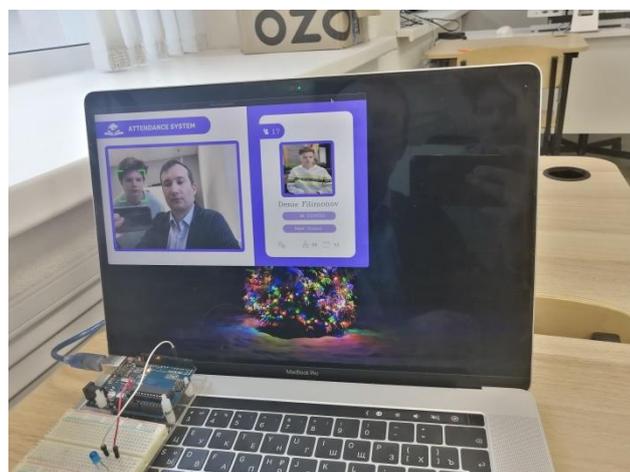
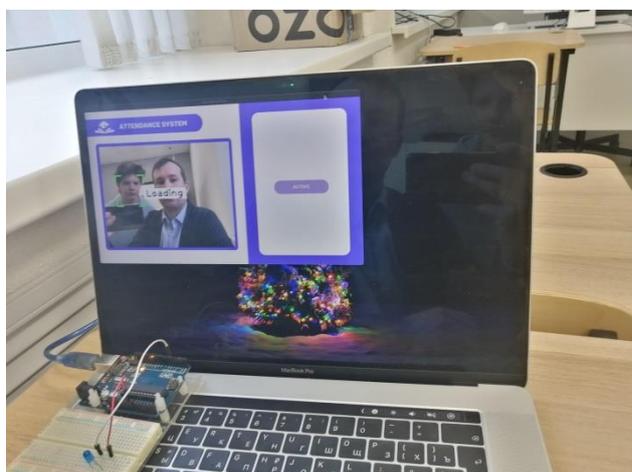
- 1) Тест с одним посетителем, доступ разрешен (происходит срабатывание турникета, отображается информация о посетителе из базы данных, посетитель помечается как вошедший).



- 2) Тест с одним посетителем, доступ запрещен (срабатывание турникета не происходит, система находится в активном состоянии и продолжает отслеживать посетителей).



- 3) Тест с несколькими посетителями (система верно определяет посетителя с допуском на объект).



Таким образом, система работает стабильно, заявленные функции работают верно. В перспективе планируется доработать функционал защиты от ложных срабатываний и обмана.

2.6. Оценка себестоимости проекта

Для определения общей себестоимости проекта FaceGate необходимо произвести расчет затрат на оборудование, программное обеспечение, материалы и рабочие часы.

С учетом всего вышеперечисленного, средняя себестоимость проекта при установке на готовые системы входных групп (турникеты и пр.) не будет превышать 10 000 р., что, гораздо ниже стоимости всех предлагаемых на рынке подобных решений.

ЗАКЛЮЧЕНИЕ

В результате реализации проекта была разработана интегрированная, надежная и интуитивно понятная система, способная эффективно контролировать доступ к объекту и предоставлять оператору всю необходимую информацию для принятия обоснованных решений.

Проект «FaceGate» не только обеспечивает повышение уровня безопасности, но и открывает перспективы для эффективного и технологичного управления доступом в различных сферах личной, общественной и корпоративной жизни.

Ожидаемые результаты проекта «FaceGate» включают в себя:

- повышение уровня безопасности;
- эффективное управление доступом;
- точность распознавания лиц;
- стабильная интеграция с firebase;
- эффективный графический интерфейс;
- надежное управление турникетом;
- высокая производительность в реальном времени;
- доступность и работоспособность;
- положительная обратная связь пользователей;
- план развития и обновлений.

Перспективы проекта «FaceGate»:

- интеграция с sql-базой данных;
- оптимизация алгоритмов распознавания лиц;
- развитие функциональности графического интерфейса;
- внедрение машинного обучения;
- расширение совместимости;
- внедрение биометрических технологий;
- анализ и реакция на результаты тестирования;
- обучение и обновление персонала;
- улучшение энергоэффективности;
- партнерство и сотрудничество.

СПИСОК ЛИТЕРАТУРЫ

1. СКУД – система контроля и управления доступом: нюансы использования и основные функции: [Электронный ресурс]. URL: <https://homemyhome.ru/skud-sistema-kontrolya-i-upravleniya-dostupom.html>. (Дата обращения: 17.10.2023)
2. 7 вариантов СКУД с распознаванием лица: [Электронный ресурс]. URL: <https://ant-parsec.ru/uslugi/tipovye-resheniya/7-variantov-skud-s-raspoznavaniem-litsa?ysclid=lpwaa02r4s941520926>. (Дата обращения: 20.10.2023)
3. Самые маленькие Arduino для ваших мини-проектов + примеры самих проектов: [Электронный ресурс]. URL: <https://habr.com/ru/company/selectel/blog/529998/>. (Дата обращения: 23.10.2023)
4. Язык программирования Python: сферы применения, методы и этапы изучения: [Электронный ресурс]. URL: <https://gb.ru/blog/yazyk-programmirovaniya-python/>. (Дата обращения: 10.11.2023)
5. СОМ-порт (последовательный порт): [Электронный ресурс]. URL: <https://www.alterbit.ru/glossary168.html>. (Дата обращения: 15.11.2023)

Листинг программного кода Python-скрипта

```

import pickle
import time
import cv2
import os
import face_recognition
import numpy as np
import cvzone
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
from firebase_admin import storage
from datetime import datetime

cred = credentials.Certificate('serviseAccKey.json')
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://faceattendance-87f81-default-
rtddb.firebaseio.com/',
    'storageBucket': 'faceattendance-87f81.appspot.com'
})
bucket = storage.bucket()
cap = cv2.VideoCapture(0)
#####
cap.set(3, 640)
cap.set(4, 480)
#####
imgBg = cv2.imread('Resources/background.png')
folderModePath = 'Resources/Modes'
modePathList = os.listdir(folderModePath)
imgModeList = []
k = -1
for repath in range(len(modePathList) // 2):
    modePathList[k], modePathList[repath] = modePathList[repath],
modePathList[k]
    k -= 1
modePathList[1], modePathList[2] = modePathList[2], modePathList[1]
print(modePathList)
for path in modePathList:
    imgModeList.append(cv2.imread(os.path.join(folderModePath, path)))
# load the encoding file
print('Loading Encode File...')
file = open('EncodeFile.p', 'rb')
encodeListKnownWithIds = pickle.load(file)
file.close()
encodeListKnown, peopleIds = encodeListKnownWithIds
print('Encode File Loaded Completely')
modeType = 0
counter = 0
id = -1
imgStudent = []
while True:
    success, img = cap.read()

```

```

imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)
faceCFrame = face_recognition.face_locations(imgS)
encodeCFrame = face_recognition.face_encodings(imgS, faceCFrame)
imgBg[162:162 + 480, 55:55 + 640] = img
imgBg[44:44 + 633, 808:808 + 414] = imgModeList[modeType]
if faceCFrame:
    for encodeFace, faceLocation in zip(encodeCFrame, faceCFrame):
        matches = face_recognition.compare_faces(encodeListKnown,
encodeFace)
        faceDis = face_recognition.face_distance(encodeListKnown,
encodeFace)
        # print('matches', matches)
        # print('faceDis', faceDis)
        matchIndex = np.argmin(faceDis)
        if matches[matchIndex]:
            # print('Known Face Detected')
            # print(peopleIds[matchIndex])
            y1, x2, y2, x1 = faceLocation
            y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
            bbox = 55 + x1, 162 + y1, x2 - x1, y2 - y1
            imgBg = cvzone.cornerRect(imgBg, bbox, rt=0)
            id = peopleIds[matchIndex]
            if counter == 0:
                cvzone.putTextRect(imgBg, 'Loading', (275, 400),
colorR=(255, 255, 255), colorT=(0, 0, 0))
                cv2.imshow('FaceGate', imgBg)
                cv2.waitKey(1)
                counter = 1
                modeType = 1
            if counter != 0:
                if counter == 1:
                    # get the data
                    peopleInf = db.reference(f'Images/{id}').get()
                    print(peopleInf)
                    # get the img
                    blob = bucket.get_blob(f'Images/{id}.png')
                    array = np.frombuffer(blob.download_as_string(), np.uint8)
                    imgStudent = cv2.imdecode(array, cv2.COLOR_BGRA2RGB)
                    # update data
                    datetimeObject =
datetime.strptime(peopleInf['last_attendance'], "%Y-%m-%d %H:%M:%S")
                    secondsElapsed = (datetime.now() -
datetimeObject).total_seconds()
                    print(secondsElapsed)
                    if secondsElapsed > 30:
                        ref = db.reference(f'Images/{id}')
                        peopleInf['total_attendance'] += 1

ref.child('total_attendance').set(peopleInf['total_attendance'])

ref.child('last_attendance').set(datetime.now().strftime('%Y-%m-%d %H:%M:%S'))
                else:
                    modeType = 3
                    counter = 0

```

```

        imgBg[44:44 + 633, 808:808 + 414] =
imgModeList[modeType]
        if modeType != 3:
            if 70 < counter < 80:
                modeType = 2
                imgBg[44:44 + 633, 808:808 + 414] = imgModeList[modeType]
            if counter <= 70:
                cv2.putText(imgBg, str(peopleInf['total_attendance']),
(861, 125), cv2.FONT_HERSHEY_COMPLEX, 1,
                    (255, 255, 255), 1)
                cv2.putText(imgBg, str(peopleInf['major']), (1006, 550),
cv2.FONT_HERSHEY_COMPLEX, 0.5,
                    (255, 255, 255), 1)
                cv2.putText(imgBg, str(id), (1006, 493),
cv2.FONT_HERSHEY_COMPLEX, 0.6, (255, 255, 255), 1)
                cv2.putText(imgBg, str(peopleInf['birth']), (1025, 625),
cv2.FONT_HERSHEY_COMPLEX, 0.6,
                    (100, 100, 100), 1)
                cv2.putText(imgBg, str(peopleInf['age']), (1125, 625),
cv2.FONT_HERSHEY_COMPLEX, 0.6,
                    (100, 100, 100), 1)
                (w, h), _ = cv2.getTextSize(peopleInf['name'],
cv2.FONT_HERSHEY_COMPLEX, 1, 1)
                offset = (414 - w) // 2
                cv2.putText(imgBg, str(peopleInf['name']), (808 +
offset, 445), cv2.FONT_HERSHEY_COMPLEX, 1,
                    (0, 0, 0), 1)
                imgBg[175:175 + 216, 909:909 + 216] = imgStudent
            counter += 1
            if counter >= 100:
                counter = 0
                modeType = 0
                peopleInf = []
                imgStudent = []
                imgBg[44:44 + 633, 808:808 + 414] = imgModeList[modeType]
        else:
            modeType = 0
            counter = 0
            # cv2.imshow('WebCam', img)
            cv2.imshow('FaceGate', imgBg)
            key = cv2.waitKey(1)
            if key == ord('q'):
                break

```

Листинг программного кода для Arduino-скетча

```
int x = 0;

char buf[1];

void setup() {
  Serial.begin(9600);
  Serial.setTimeout(1);
  pinMode(9, OUTPUT);
  if(Serial.available() > 0){
    x = Serial.parseInt();
  }
}

void loop() {
  if(Serial.available() > 0){
    x = Serial.parseInt();
    //Serial.readBytes(buf, sizeof(buf));
    //x = int(buf[0]);
  }

  //Serial.println(x);
  if(x==0)
    digitalWrite(9, 0);
  else if(x == 1)
    digitalWrite(9, 1);
  // delay(100);
}
```

**Рецензия научного руководителя
на исследовательскую работу
«Система автоматизированного контроля входных групп «FaceGate»
(автор: Филимонов Денис Сергеевич)**

**VI открытого регионального конкурса исследовательских и
проектных работ школьников «Высший пилотаж - Пенза» 2024**

Система автоматизированного контроля FaceGate представляет собой инновационное решение для обеспечения безопасности и контроля доступа на различные объекты, такие как офисные здания, торговые центры, кинотеатры и другие общественные места. Система использует технологию распознавания лиц для идентификации посетителей и автоматической регистрации их входа и выхода из объекта. Это позволяет существенно сократить время, затрачиваемое на проверку пропусков, и повышает уровень безопасности объекта.

В своей работе автор предлагает простое, надежное и эффективное решение. Система «FaceGate» разработана с учетом удобства пользователей - все, что требуется от посетителя, это просто взглянуть на камеру системы, чтобы пройти идентификацию. Это позволяет сократить время пребывания людей в очередях и тем самым снижает риски передачи вирусных инфекций, особенно в периоды пандемий. Система легко интегрируется с различными системами управления доступом и может быть настроена для работы с большим количеством входных групп.

Отличительными особенностями системы «FaceGate» (по сравнению с аналогами) являются: низкая стоимость, универсальность применения, удобство использования (минимум действий, бесконтактный режим для посетителей), автоматический режим работы.

Система автоматизированного контроля «FaceGate» является перспективным и инновационным решением для обеспечения безопасности на различных объектах. Использование передовых технологий распознавания лиц и удобство использования делают данную систему привлекательной для многих организаций. Несмотря на некоторые ограничения, развитие технологий распознавания лиц, включая «FaceGate», будет продолжать развиваться, что делает этот проект весьма актуальным и перспективным.

Научный руководитель



Ю.А. Трофимов