

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ КОДОВОГО ЗАМКА ДЛЯ
СЕЙФА НА БАЗЕ ПЛАТЫ ARDUINO, С ИСПОЛЬЗОВАНИЕМ БЕСПРОВОДНОГО
МОДУЛЯ СВЯЗИ BLUETOOTH

Сулейманова Галина Николаевна¹, Бакиров Расим Гаязович²

¹Доцент кафедры "Иностранные языки" Ташкентского университета информационных технологий имени Мухаммад ал-Хоразмий

²Студент Ташкентского университета информационных технологий имени Мухаммад ал-Хоразмий

<https://doi.org/10.5281/zenodo.7856741>

Abstract. *This article discusses the development of a code lock based on Arduino, equipped with a Bluetooth wireless communication module. An application has been developed to control the lock, which allows the user to remotely manage access to the object.*

Keywords: *Arduino, code lock, mobile app development, Android, Bluetooth, algorithm programming, wireless communication.*

В настоящее время безопасность имеет особое значение, и многие люди стремятся защитить свои ценности и документы от кражи. Один из способов обеспечения безопасности – использование сейфа с кодовым замком. Однако, сейчас многие сейфы имеют ограниченную функциональность и не позволяют следить за условиями хранения ценностей внутри.

Используя платформу Arduino, можно создать собственный сейф с кодовым замком и датчиком температуры, который не только обеспечит безопасность, но и позволит следить за температурой внутри сейфа. Кодовый замок может быть установлен на двери сейфа, а датчик температуры – внутри, чтобы обеспечить мониторинг условий хранения ценностей.

Для сборки сейфа на базе Arduino нам потребуются следующие компоненты:

- Плата Arduino Uno;
- Макетная плата;
- Резисторы 100Ом;
- Клавиатура 4x3;
- Сервопривод;
- Провода;
- 4 красных светодиода;
- Кабель USB;
- Bluetooth HC-05.

Принципиальная схема данного электронного оборудования наглядно представлена на рисунке 1.

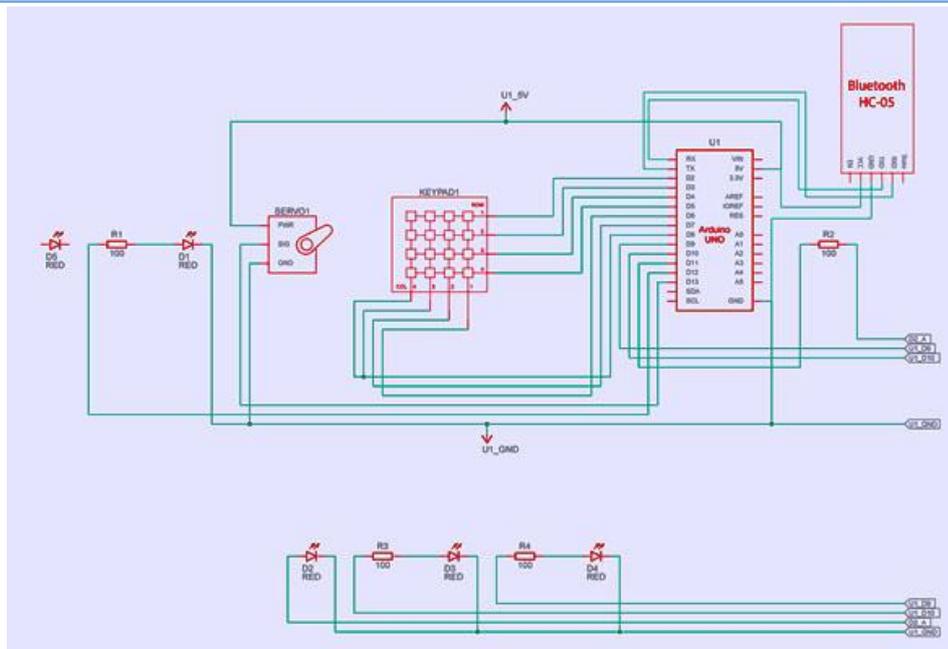


Рис.1. Принципиальная схема управления кодового замка сейфа на базе Arduino с использованием беспроводного модуля связи Bluetooth

Схема сборки подобного устройства довольно проста. Алгоритм работы устройства заключается в том, что пользователь вводит код доступа через клавиатуру Matrix 4x3. Код доступа будет сохранен в памяти Arduino и проверен на совпадение с введенным пользователем [1].

Когда пользователь вводит правильный код, сервопривод открывает дверь сейфа, и пользователь получает доступ к его содержимому. При этом, чтобы предоставить пользователю обратную связь, при вводе кода будут последовательно загораться светодиоды. Таким образом, пользователь будет получать визуальное подтверждение каждого введенного символа и знать, сколько символов осталось ввести.

Если же пользователь вводит неправильный код, светодиоды начинают мигать и потухать, чтобы предупредить пользователя об ошибке. Это поможет избежать блокировки доступа и упростить процесс ввода кода. Пользователь сможет быстро понять, что вводит неверный код, и исправить его.

При вводе правильного кода светодиоды продолжают гореть, пока сейф не будет закрыт. Это поможет пользователю ориентироваться в процессе ввода кода и убедиться, что он ввел все символы правильно. При закрытии сейфа, сервопривод закрывает дверь, и доступ к его содержимому снова становится недоступен без правильного кода.

Таким образом, данное устройство представляет собой надежное и удобное средство для контроля доступа к сейфу. Оно предоставляет пользователю обратную связь о правильности введенного кода, а также предупреждает его об ошибках при вводе [2].

Следующим этапом была разработка приложения для управления собранным сейфом через беспроводную связь Bluetooth. Для этого был составлен алгоритм подключения смартфона к модулю Bluetooth HC-05 и созданы кнопки управления на клавиатуру 4x3.

В данной конструкции платформа Android включает в себя сетевой стек Bluetooth, который обеспечивает беспроводный обмен данными между беспроводными устройствами. Доступ к функциям Bluetooth осуществляется через Bluetooth API,

предоставляемый платформой приложения. Эти API-интерфейсы обеспечивают возможность подключения приложений к другим устройствам Bluetooth для осуществления беспроводной связи типа "точка-точка" или "многоточка".

С использованием интерфейса программирования приложений (API) Bluetooth, возможно осуществить следующие операции:

- Поиск других устройств, поддерживающих протокол Bluetooth.
- Запрос состояния локального адаптера Bluetooth в отношении уже сопряженных устройств Bluetooth.
- Установление каналов RFCOMM.
- Подключение к другим устройствам с использованием сервиса обнаружения.
- Передача данных между устройствами, как с одного, так и на другое.
- Управление несколькими соединениями одновременно [3].

Для использования API Bluetooth необходимо объявить несколько разрешений в файле манифеста. Получив доступ к использованию Bluetooth, приложение должно осуществить доступ к BluetoothAdapter и определить наличие Bluetooth на устройстве. Если Bluetooth доступен, для установления соединения следует выполнить три шага:

1. Найти ближайшие Bluetooth-устройства, включая уже сопряженные и новые.
2. Установить соединение с выбранным Bluetooth-устройством.
3. Осуществить передачу данных с помощью подключенного устройства.

В Android-разработке существует пакет android.bluetooth, который содержит все API для работы с Bluetooth. Данный пакет предоставляет классы и интерфейсы, необходимые для установления соединений Bluetooth. Ниже представлены основные классы и интерфейсы:

- Один из основных классов в пакете android.bluetooth - это BluetoothAdapter. Он представляет локальный Bluetooth-адаптер (Bluetooth-радио), который является точкой входа для всех взаимодействий Bluetooth. С помощью BluetoothAdapter можно обнаруживать другие устройства Bluetooth, запрашивать список связанных (сопряженных) устройств, создавать экземпляры с помощью класса BluetoothDevice, используя известный MAC-адрес, и создавать BluetoothServerSocket для прослушивания сообщений от других устройств.

- Класс BluetoothDevice, в свою очередь, представляет удаленное устройство Bluetooth. Он используется для запроса соединения с удаленным устройством через BluetoothSocket или для запроса информации об устройстве, такой как его имя, адрес, класс и состояние связи.

- BluetoothSocket представляет интерфейс для сокета Bluetooth, аналогичного TCP Socket. Он является точкой подключения, которая позволяет приложению обмениваться данными с другим устройством Bluetooth с помощью InputStream и OutputStream.

- BluetoothServerSocket представляет открытый серверный сокет, который прослушивает входящие запросы, аналогично TCP ServerSocket. Для соединения двух устройств одно из устройств должно открыть серверный сокет с помощью этого класса. Когда удаленное устройство Bluetooth отправляет запрос на подключение к этому

устройству, устройство принимает соединение и затем возвращает подключенный файл BluetoothSocket.

- BluetoothClass описывает общие характеристики и возможности устройства Bluetooth. Он представляет доступный только для чтения набор свойств, определяющий классы и службы устройства. Хотя эта информация дает полезную информацию о типе устройства, атрибуты этого класса не обязательно описывают все профили и службы Bluetooth, поддерживаемые устройством [4].

Внешний вид разработанного приложения наглядно представлен на рисунке 2.



Рис.2. Внешний вид разработанного приложения

Для мониторинга температуры внутри сейфа, в данной конструкции планируется использовать датчик температуры DS18B20, который можно подключить к плате Arduino. Используемый датчик имеет высокую точность и может измерять температуру в диапазоне от -55°C до $+125^{\circ}\text{C}$. Мы можем использовать библиотеку OneWire и DallasTemperature, чтобы облегчить работу с датчиком температуры.

Когда температура внутри сейфа превысит заданный порог, Arduino может отправить сообщение через Bluetooth модуль HC-05 на смартфон пользователя, чтобы уведомить его о возможной угрозе для ценностей внутри сейфа.

Также можно настроить систему предупреждения о попытке несанкционированного доступа к сейфу. Для этого можно использовать дополнительные датчики, такие как датчик движения или датчик звука, которые будут отслеживать любую активность возле сейфа, и при обнаружении подобной активности Arduino будет отправлять сообщение на смартфон пользователя. Для обеспечения дополнительной безопасности можно использовать шифрование сообщений, передаваемых между Arduino и смартфоном, чтобы предотвратить возможные попытки несанкционированного доступа к системе управления сейфом.

На момент написания статьи уже собран прототип сейфа и разработано приложение для его управления. В ближайшем будущем в сейф будет добавлен датчик температуры, а также в приложение будет добавлена возможность мониторинга температуры. Так же в

данном проекте был использован модуль Bluetooth HC-05, который обладает возможностью передачи данных только по Bluetooth Classic. Однако, в случае необходимости, данный модуль может быть заменен на модуль Bluetooth HC-08 или HM-10, которые поддерживают передачу данных по протоколу Bluetooth Low Energy (BTLE).

Внешний вид собранной конструкции сейфа с электронным управлением кодового замка наглядно представлен на рисунке 3.

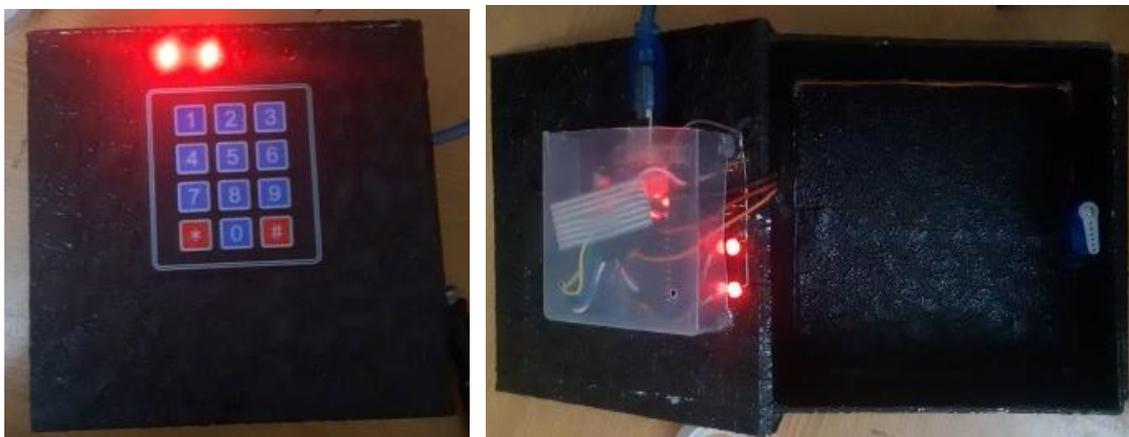


Рис.3. Внешний вид сейфа с электронным управлением кодового замка

.В заключении хотелось бы отметить, что система на базе Arduino обеспечивает не только безопасность и мониторинг температуры внутри сейфа, но также позволяет настраивать различные сценарии взаимодействия, в зависимости от индивидуальных потребностей пользователя.

REFERENCES

1. Аппаратная платформа Arduino [Электронный ресурс] // Arduino.ru. – Режим доступа: <http://arduino.ru>
2. Блум Джереми Б. Изучаем Arduino: инструменты и методы технического волшебства: Пер. с англ. - СПб.: БХВ-Петербург, 2015. - 336 с.: ил.
3. Документация Android [Электронный ресурс] // developer.android.com – Режим доступа: <https://developer.android.com/>
4. Beginning Android ADK with Arduino by Mario Bohmer, 2012 г.