ДОДАТКИ

ДОДАТОК А

ПУБЛІКАЦІЯ ЗА ТЕМОЮ РОБОТИ

```
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <Servo.h>
#include <MFRC522.h>
#include "SafeState.h"
#include "icons.h"

#define SERVO_PIN 6
#define SERVO_LOCK_POS   20
#define SERVO_UNLOCK_POS 90

#define BUZZER_PIN 13

#define RGB_RED_PIN 7
#define RGB_GREEN_PIN 8
#define RGB_BLUE_PIN 9

#define RFID_SS_PIN 10
#define RFID_RST_PIN 9

char* keys[]={"133 193 216 101"};
Servo lockServo;

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

const byte KEYPAD_ROWS = 4;
const byte KEYPAD_COLS = 4;
byte rowPins[KEYPAD_ROWS] = {A3, A2, A1, A0};
byte colPins[KEYPAD_COLS] = {A4, A5, A6, A7};
char keys[KEYPAD_ROWS][KEYPAD_COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, KEYPAD_ROWS,
KEYPAD_COLS);

SafeState safeState;
MFRC522 rfid(RFID_SS_PIN, RFID_RST_PIN);

void lock() {
  lockServo.write(SERVO_LOCK_POS);
  safeState.lock();
  setRGBColor(255, 0, 0);

void unlock() {
  lockServo.write(SERVO_UNLOCK_POS);
```

```
  setRGBColor(0, 255, 0);
}

void setRGBColor(int red, int green, int blue) {
  analogWrite(RGB_RED_PIN, red);
  analogWrite(RGB_GREEN_PIN, green);
  analogWrite(RGB_BLUE_PIN, blue);
}

void showStartupMessage() {
  lcd.setCursor(4, 0);
  lcd.print("Welcome!");
  delay(1000);
}

String inputSecretCode() {
  lcd.setCursor(5, 1);
  lcd.print("[____]");
  lcd.setCursor(6, 1);
  String result = "";
  while (result.length() < 4) {
    char key = keypad.getKey();
    if (key >= '0' && key <= '9') {
      lcd.print('*');
      result += key;
    }
  }
  return result;
}

void showWaitScreen(int delayMillis) {
  lcd.setCursor(2, 1);
  lcd.print("[.........]");
  lcd.setCursor(3, 1);
  for (byte i = 0; i < 10; i++) {
    delay(delayMillis);
    lcd.print("=");
  }
}

bool setNewCode() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Enter new code:");
  String newCode = inputSecretCode();

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Confirm new code");
```

```
  String confirmCode = inputSecretCode();

  if (newCode.equals(confirmCode)) {
    safeState.setCode(newCode);
    return true;
  } else {
    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print("Code mismatch");
    lcd.setCursor(0, 1);
    lcd.print("Safe not locked!");
    delay(2000);
    return false;
  }
}

void showUnlockMessage() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.write(ICON_UNLOCKED_CHAR);
  lcd.setCursor(4, 0);
  lcd.print("Unlocked!");
  lcd.setCursor(15, 0);
  lcd.write(ICON_UNLOCKED_CHAR);
  delay(1000);
}

void safeUnlockedLogic() {
  lcd.clear();

  lcd.setCursor(0, 0);
  lcd.write(ICON_UNLOCKED_CHAR);
  lcd.setCursor(2, 0);
  lcd.print(" # to lock");
  lcd.setCursor(15, 0);
  lcd.write(ICON_UNLOCKED_CHAR);

  bool newCodeNeeded = true;

  if (safeState.hasCode()) {
    lcd.setCursor(0, 1);
    lcd.print("  A = new code");
    newCodeNeeded = false;
  }

  auto key = keypad.getKey();
  while (key != 'A' && key != '#') {
    key = keypad.getKey();
  }
```

```
  bool readyToLock = true;
  if (key == 'A' || newCodeNeeded) {
    readyToLock = setNewCode();
  }

  if (readyToLock) {
    lcd.clear();
    lcd.setCursor(5, 0);
    lcd.write(ICON_UNLOCKED_CHAR);
    lcd.print(" ");
    lcd.write(ICON_RIGHT_ARROW);
    lcd.print(" ");
    lcd.write(ICON_LOCKED_CHAR);

    safeState.lock();
    lock();
    showWaitScreen(100);
  }
}

void safeLockedLogic() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.write(ICON_LOCKED_CHAR);
  lcd.print(" Safe Locked! ");
  lcd.write(ICON_LOCKED_CHAR);

  String userCode = inputSecretCode();
  bool unlockedSuccessfully = safeState.unlock(userCode);
  showWaitScreen(200);

  if (unlockedSuccessfully) {
    showUnlockMessage();
    unlock();
  } else {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Access Denied!");
    showWaitScreen(1000);
    tone(BUZZER_PIN, 1000, 500);
  }
}

void checkRFID() {
  if (!rfid.PICC_IsNewCardPresent()) {
    return;
  }
  if (!rfid.PICC_ReadCardSerial()) {
```

```
    return;
  }

  String rfidTag = "";
  for (byte i = 0; i < rfid.uid.size; i++) {
    rfidTag += String(rfid.uid.uidByte[i] < 0x10 ? "0" : "");
    rfidTag += String(rfid.uid.uidByte[i], HEX);
  }

  rfidTag.toUpperCase();

  if (safeState.unlockWithRFID(rfidTag)) {
    showUnlockMessage();
    unlock();
  } else {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Access Denied!");
    showWaitScreen(1000);
    tone(BUZZER_PIN, 1000, 500);
  }

  rfid.PICC_HaltA();
}

void setup() {
  lcd.begin(16, 2);
  init_icons(lcd);

  lockServo.attach(SERVO_PIN);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(RGB_RED_PIN, OUTPUT);
  pinMode(RGB_GREEN_PIN, OUTPUT);
  pinMode(RGB_BLUE_PIN, OUTPUT);

  SPI.begin();
  rfid.PCD_Init();

  Serial.begin(115200);
  unlock();

  showStartupMessage();
}

void loop() {
  if (safeState.locked()) {
    checkRFID();
    safeLockedLogic();
  } else {
```

```
    safeUnlockedLogic();
  }
}
```

```
#ifndef SAFESTATE_H
#define SAFESTATE_H

class SafeState {
  public:
    SafeState();
    void lock();
    bool unlock(String code);
    bool locked();
    bool hasCode();
    void setCode(String newCode);

  private:
    void setLock(bool locked);
    bool _locked;
};

#endif
```

```
#ifndef ICONS_H
#define ICONS_H

#include <LiquidCrystal.h>

#define ICON_LOCKED_CHAR   (byte)0
#define ICON_UNLOCKED_CHAR (byte)1

#define ICON_RIGHT_ARROW   (byte)126

void init_icons(LiquidCrystal &lcd);

#endif
```

ДОДАТОК Б

СЛАЙДИ ПРЕЗЕНТАЦІЇ

Харківський національний університет радіоелектроніки
Кафедра Радіотехнологій інформаційно-комунікаційних систем

Кваліфікаційна робота бакалавра на тему:

Цифровий замок для сейфа на базі Arduino

Студент: Балюк Дмитро Олександрович
Група: ТРРТу-21-1
Керівник: ст. викл. Ганшин Дмитро Геннадійович

Харків 2024

ВСТУП

У сучасному світі безпека є однією з найважливіших складових нашого життя. Система безпеки повинна бути надійною, зручною у використанні та доступною. Саме тому я обрав тему цифрових замків для своєї дипломної роботи.

Метою цього проекту було створення ефективної системи цифрового замка, яка забезпечує високий рівень захисту та зручність для користувачів. Для реалізації цієї задачі я використав популярну платформу Arduino, яка є ідеальним інструментом для створення прототипів завдяки своїй гнучкості, доступності та широкій підтримці з боку спільноти розробників.

У своїй роботі я досліджував різні методи ідентифікації користувачів, такі як використання клавіатур, RFID-карт та інші технології. Особливу увагу було приділено безпеці зберігання даних та захисту від несанкціонованого доступу. У ході проекту я розробив програмне забезпечення, яке включає різноманітні функції для управління замком, обробки введення з клавіатури, відображення інформації на дисплеї та роботи з RFID-картами.

АНАЛІЗ ТЕХНОЛОГІЙ ЦИФРОВИХ ЗАМКІВ

Цифрові замки – це складні системи, які поєднують електронні та механічні компоненти для забезпечення високого рівня безпеки та зручності. Вони постійно вдосконалюються , щоб відповідати сучасним вимогам безпеки і потребам користувачів.

Існує кілька видів цифрових замків, які класифікуються за різними критеріями, такими як методи аутентифікації , призначення, спосіб встановлення та технології підключення і за рівнем безпеки.

Цифрові замки класифікуються за різними критеріями, що дозволяє підібрати оптимальне рішення для конкретних потреб і вимог. Незалежно від типу, всі цифрові замки мають забезпечувати високий рівень безпеки та зручності для користувачів.

3

ПОРІВНЯННЯ ЗАМКІВ РІЗНИХ ВИРОБНИКІВ

Виробники	Технології	Додатковіфункції	Ціни ($)
August (Smart Lock, Smart Lock Pro, Wi-Fi Smart Lock)	підтримка Bluetooth, Wi-Fi, сумісність з HomeKit, Google Assistant, Amazon Alexa	можливість віддаленого доступу, журнал доступу,автоматичне замикання і відмикання при наближенні користувача	200-250
Schlage (Encode, Connect, Sense)	підтримка Z-Wave, Wi-Fi, HomeKit, Amazon Alexa	вбудованийWi-Fi, можливість встановлення кодів доступу,сумісність з іншими системами розумногобудинку	150-300
Yale (Assure Lock, Real Living, Nest x Yale)	підтримка Bluetooth, Z-Wave, Zigbee, Wi-Fi, HomeKit	можливість встановлення кодів доступу, інтеграція з системами розумногобудинку, аварійне відкриття за допомогою фізичного ключа.	200-350
Kwikset (Kevo, Halo, Premis)	підтримка Bluetooth, Wi-Fi, Z-Wave, HomeKit.	можливість встановлення кодів доступу, інтеграція з іншими системами розумного будинку, можливість віддаленого керування	150-250
Samsung (SHP-DP609, SHS-2920, SHP-DS705)	підтримка Bluetooth, Wi-Fi, інтеграція з системами розумного будинку	біометричний доступ, цифрові коди, можливість дистанційного керування, тривожні сповіщення	200-400

4

ПОРІВНЯННЯ ЗАМКІВ РІЗНИХ ВИРОБНИКІВ

Виробники	Переваги	Недоліки
August (Smart Lock, Smart Lock Pro, Wi-Fi Smart Lock)	Простота встановлення і використання. Широка сумісність з іншими розумними пристроями Можливість віддаленого керування	Вимагає окремого Wi-Fi мосту для деяких функцій. Відносно висока ціна.
Schlage (Encode, Connect, Sense)	Висока надійність і безпека. Широкий вибір моделей для різних потреб. Інтеграція з багатьма системами безпеки та розумного будинку.	Складніше встановлення у порівнянні з деякими іншими брендами. Деякі моделі дорожчі за аналоги.
Yale (Assure Lock, Real Living, Nest x Yale)	Великий вибір технологій підключення. Висока якість і надійність. Підтримка численних платформ розумного будинку.	Складніше встановлення деяких моделей. Відносно висока ціна.
Kwikset (Kevo, Halo, Premis)	Доступні ціни. Легкість встановлення і налаштування. Хороший вибір технологій підключення.	Деякі моделі можуть бути менш надійними порівняно з іншими брендами. Обмежені функції у деяких моделях.
Samsung (SHP-DP609, SHS-2920, SHP-DS705)	Високотехнологічні функції, такі як біометричний доступ. Сучасний і стильний дизайн. Надійність і якість від відомого бренду.	Складніше встановлення. Відносно висока ціна.

5

ТЕХНІЧНІ ВИМОГИ ДО ЦИФРОВОГО ЗАМКА

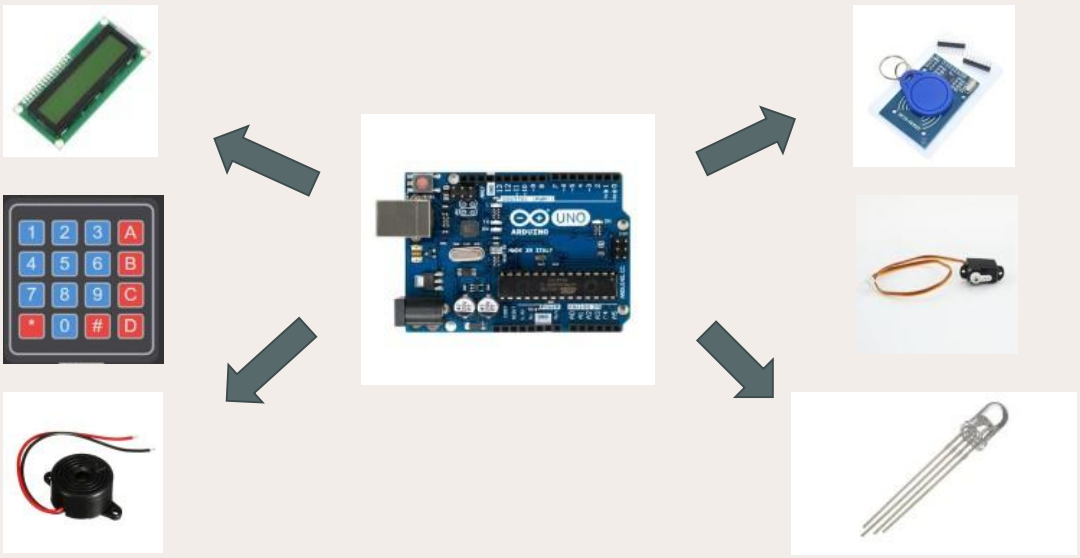
Основні технічні характеристики цифрового замка повинні забезпечувати високу надійність, безпеку та зручність використання, що робить його ефективним рішенням для захисту сейфів та інших об'єктів.

Ключові технічні параметри, що визначають функціональність і надійність замка:

- Мікроконтролер;
- Живлення;
- Вхідні та вихідні порти;
- Безпека;
- Надійність та довговічність;

6

РОЗРОБКА ЦИФРОВОГО ЗАМКУ ДЛЯ СЕЙФА



7

Програмування цифрового коду

Бібліотеки

```
#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <Servo.h>
#include "SafeState.h"
#include "icons.h"
```

```
#ifndef ICONS_H
#define ICONS_H

#include <LiquidCrystal.h>


#define ICON_LOCKED_CHAR    (byte)0
#define ICON_UNLOCKED_CHAR (byte)1


#define ICON_RIGHT_ARROW    (byte)126

void init_icons(LiquidCrystal &lcd);

#endif /* ICONS_H */
```

```
#ifndef SAFESTATE_H
#define SAFESTATE_H

class SafeState {
  public:
    SafeState();
    void lock();
    bool unlock(String code);
    bool locked();
    bool hasCode();
    void setCode(String newCode);

  private:
    void setLock(bool locked);
    bool _locked;
};

#endif /* SAFESTATE_H */
```

```
void setup() {
  lcd.begin(16, 2);
  init_icons(lcd);

  lockServo.attach(SERVO_PIN);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(RGB_RED_PIN, OUTPUT);
  pinMode(RGB_GREEN_PIN, OUTPUT);
  pinMode(RGB_BLUE_PIN, OUTPUT);

  SPI.begin();
  rfid.PCD_Init();

  Serial.begin(115200);
  unlock();

  showStartupMessage();
}
```

```
void showStartupMessage() {
  lcd.setCursor(4, 0);
  lcd.print("Welcome!");
  delay(1000);
}
```

8

Головний цикл

```
void loop() {
  if (safeState.locked()) {
    checkRFID();
    safeLockedLogic();
  } else {
    safeUnlockedLogic();
  }
}
```

9

10

Висновки

У ході виконання дипломної роботи було розглянуто й проаналізовано різні аспекти розробки цифрових замків на базі платформи Arduino. Проведено огляд існуючих технологій цифрових замків, що включає використання клавіатур, RFID-технологій та інших засобів ідентифікації.

Розроблено програмне забезпечення для цифрового замка, яке включає функції управління серводвигуном, обробки введення з клавіатури, відображення інформації на LCD-дисплеї та роботу з RFID-картами. Було реалізовано функцію додавання нових карток для можливості розширення списку користувачів, що мають доступ до замка.

У процесі роботи було приділено увагу безпеці зберігання паролів та ідентифікаційних даних. Використання EEPROM для зберігання секретних кодів забезпечує надійність та стійкість системи до втрати живлення. Крім того, розроблена система включає захисні механізми від спроби підбору пароля, такі як затримка у випадку неправильного введення.

Практична частина роботи продемонструвала ефективність створеної системи. Цифровий замок на базі Arduino успішно виконує поставлені завдання, забезпечуючи надійний захист приміщення та зручність використання. Отримані результати підтверджують доцільність використання платформи Arduino для розробки подібних систем, завдяки її гнучкості, доступності та широким можливостям для інтеграції з іншими компонентами.

Ця дипломна робота демонструє можливості сучасних цифрових замків, їх переваги та області застосування. Розроблене рішення може бути використано як основа для подальших досліджень та вдосконалень у сфері безпеки та автоматизації, що відкриває нові перспективи для створення ще більш надійних та зручних систем захисту.

11

ДЯКУЮ ЗА УВАГУ!

ДОДАТОК В

ВІДОМІСТЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ

№ п/п	Позначення	Найменування	Додаткові відомості
		Текстові документи	
1	ГЮІК.ХХХХХХ.000ПЗ	Пояснювальна записка	с.

Змін.	Арк.	Номер докум.	Підпис	Дата	ГЮІК.ХХХХХХ.000Д2		
Розроб.		Балюк			Цифровой замок для сейфу на базі Arduino Відомість кваліфікаційної роботи	Літ.	Аркуш 1 / Аркушів 1
Перевір.		Ганшин				У	
						ХНУРЕ кафедра РТІКС	
Н. контр.		Бітченко					
Затв.		Зарудний					