

**Podstawowe parametry:**

- liczba obsługiwanych kart RFID/NFC: 100,
- liczba obsługiwanych PIN-kodów: 100,
- zasilanie: 8...10 VDC, 150 mA,
- parametry styków przełącznika: 250 VAC, 125 VDC, AC: 10 A; DC: 8 A (szczegóły w dokumentacji przełącznika).

\* **Uwaga!** Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętność lutowania! Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wlutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wlutowane w płytkę PCB),
  - wersja [A] – płytka drukowana bez elementów i dokumentacji.
- Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- wersja [A+] – płytka drukowana [A] + zaprogramowany układ [UK] i dokumentacja,
  - wersja [UK] – zaprogramowany układ.

**Dodatkowe materiały do pobrania ze strony [www.ulubionykiosk.pl/media](http://www.ulubionykiosk.pl/media)**

- Simple Access System 2 (2) (EP 6/2022)
- Simple Access System 2 (1) (EP 5/2022)
- NFC Lock (EP 4/2022)
- AVT5186 Bezstykowy zamek RFID
- AVT969 Bezstykowy zamek RFID
- AVT886 System bezstykowej kontroli dostępu (EP 10/2000)

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik PDF! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! <http://sklep.avt.pl>

W przypadku braku dostępności na stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: [kity@avt.pl](mailto:kity@avt.pl)

W ofercie AVT\*

**AVT6009**

# multiLock

*multiLock to system kontroli dostępu, który umożliwia dostęp poprzez karty zbliżeniowe standardu RFID/NFC oraz zwykłe 4-cyfrowe PIN-kody. W celu obsługi wspomnianych kart zastosowano moduł czytnika RFID/NFC firmy NXP pod postacią peryferium oznaczonego symbolem PN532, który integruje w sobie specjalizowany układ o tej samej nazwie (mikrokontroler z rdzeniem 80C51 wyposażony w 40 kB pamięci ROM i 1 kB pamięci RAM) oraz cały interfejs RF.*

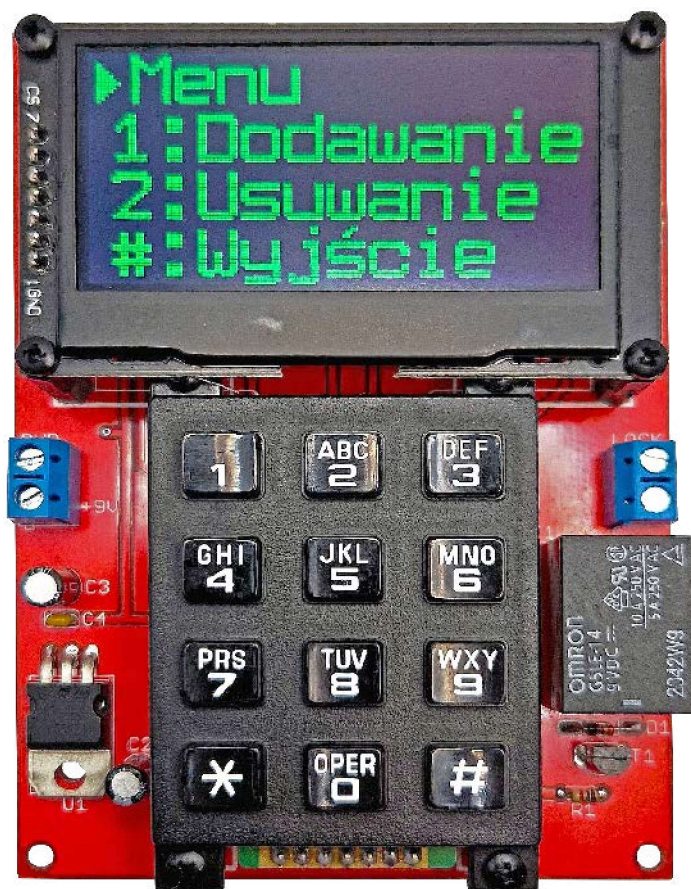
W portfolio moich projektów bez problemu znajdziecie różnego rodzaju systemy dostępu, w których medium autoryzacyjnym były pastylki iButton, karty RFID/NFC czy zwykłe PIN-kody, jednak przyznać muszę, że zawsze były to systemy, w implementacji których zdecydowałem się na zastosowanie wyłącznie jednej ze wspomnianych technologii. Nie powinno to specjalnie dziwić, gdyż dokładnie w ten sposób implementowane są zazwyczaj komercyjne rozwiązania tego typu, jednak musicie przyznać, że stanowi to swego rodzaju ograniczenie.

Podam prosty przykład. Załóżmy, że korzystamy z systemu kontroli dostępu, który stosuje karty RFID/NFC w celu autoryzacji dostępu. Czy w przypadku, gdy nie mamy przy sobie tego rodzaju karty, istnieje możliwość autoryzowanego dostępu? Oczywiście, że NIE! Czy tego rodzaju rozwiązanie jest satysfakcjonujące? Oczywiście, że NIE! Jak powinno się rozwiązać tego rodzaju zagadnienie? Moim zdaniem, każdy tego rodzaju system powinien zapewniać kilka mechanizmów kontrolowanego dostępu zawierających różne media autoryzacyjne. Nie ukrywam, że do wniosków takich doszedłem, użytkując jeden z nowszych, komercyjnych systemów kontroli dostępu, gdzie medium autoryzacyjnym były zarówno karty/breloki standardu RFID/NFC, jak i zwykłe PIN-kody...

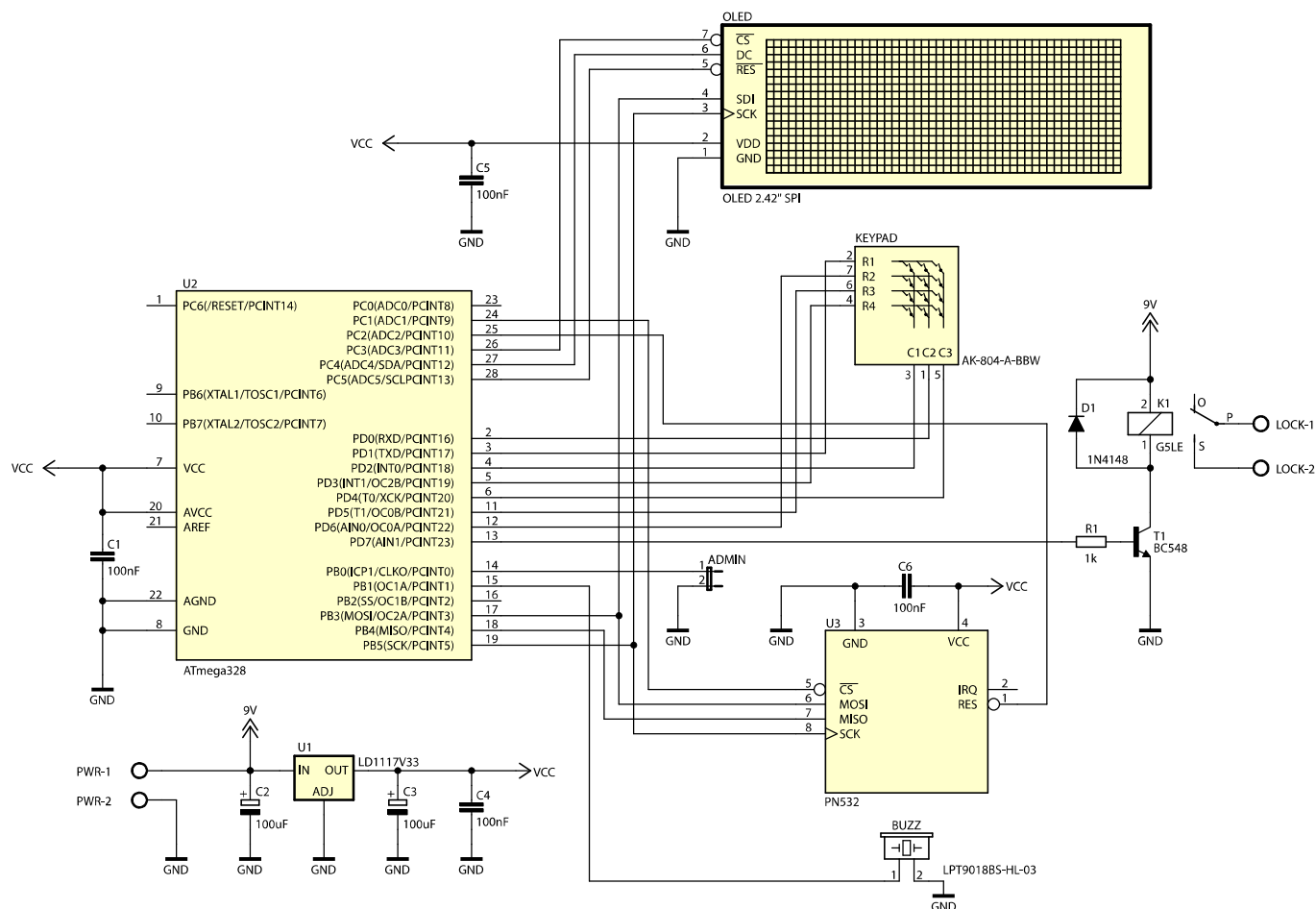
i właśnie tego rodzaju implementację chciałem Wam zaprezentować w ramach niniejszego projektu.

## Moduł PN532

Moduł, o którym już wspomniałem, to zespół pozwalający na wymianę danych (jednokierunkową lub dwukierunkową) z kartami i urządzeniami w standardzie RFID i NFC. Obsługiwane rodzaje kart to: Mifare 1k, 4k, Ultralight, DesFire, ISO/IEC 14443-4, Innovision Jewel (IRT5001) oraz FeliCa (RCS\_860, RCS\_854). Oprócz obsługi kart lub tagów RFID, moduł pozwala również na dwukierunkową komunikację z urządzeniami NFC – np. telefonem komórkowym, a także wymianę danych między dwoma modułami w trybie P2P.



Wspomniany podzespół wyposażono dodatkowo w wygodne interfejsy komunikacyjne, dzięki którym możliwa jest jego konfiguracja i wymiana danych z procesorem hosta. Interfejsy te to: FC, SPI lub HSU (*High Speed UART*). Wybór aktywnego interfejsu komunikacyjnego dokonywany jest za pomocą dedykowanych switchy SMD zintegrowanych na obwodzie drukowanym PCB. Aby przygotować moduł do pracy z interfejsem SPI, przełączamy przełącznik oznaczony jako ON w pozycję On (blisko znacznika „1”), zaś przełącznik oznaczony jako KE w pozycję Off (blisko znacznika „KE”). Na tym samym obwodzie drukowanym zintegrowano również antenę poprowadzoną na brzegach płytki, dzięki czemu dostajemy urządzenie gotowe do pracy. Tyle w kwestii samego modułu.



Rysunek 1. Schemat ideowy urządzenia multiLock

Kilka słów uwagi należy w tym miejscu poświęcić samej metodologii przesyłania danych RFID/NFC. Obie to dwie bardzo zbliżone do siebie technologie bezprzewodowej komunikacji, które używane są praktycznie na każdym kroku w różnych dziedzinach naszego życia – od weryfikacji tożsamości poprzez kontrolę dostępu aż do systemów bezprzewodowych płatności zbliżeniowych. Jakie są główne różnice pomiędzy RFID a NFC? RFID pozwala na jednokierunkową (w sensie inicjowania transferu), bezprzewodową komunikację pomiędzy niezasilanym znacznikiem a zasilanym czytnikiem RFID, przy czym odległość między tym czytnikiem a znacznikiem może dochodzić nawet do 200 m (w wybranych implementacjach) i zależy w dużej mierze od zastosowanych zakresów częstotliwości radiowych oraz protokołów komunikacji.

W odróżnieniu od RFID, technologia NFC pozwala natomiast na dwustronną komunikację pomiędzy dwoma urządzeniami NFC, realizując obsługę bardziej złożonych

mechanizmów, a także wymianę danych. Wynika to głównie z idei powstania tej technologii, która w zamierzeniu miała umożliwić komunikację na małe, bezpieczne odległości (poniżej 20 cm), przez co idealnie nadaje się do zastosowania w telefonach komórkowych do przeprowadzania bezpiecznych transakcji bezgotówkowych lub wymiany danych pomiędzy tymi urządzeniami.

W prezentowanym urządzeniu skupię się wyłącznie na obsłudze kart Mifare 1k oraz urządzeń NFC zgodnych z tym standardem, gdyż naszym zadaniem będzie wyłącznie odczytanie unikalnego numeru seryjnego karty/urządzenia RFID/NFC (tzw. UID). Należy mieć jednak świadomość, że karty tego rodzaju udostępniają dużo większą funkcjonalność. Dla przykładu wyposażono je w 1024 bajty pamięci EEPROM (16 sektorów po 4 bloki o rozmiarze 16 B), która może być zabezpieczona przed zapisem/ odczytem za pomocą zdefiniowanego klucza dostępowego (i to indywidualnie dla każdego z tychże sektorów).

Ciekawe, nieprawdaż? I na tym zakończę opis samego modułu i zagadnień związanych z obsługą kart RFID/NFC, mimo że jest to naprawdę dość obszerne i niezmiernie ciekawe zagadnienie. Dlaczego? Otóż dlatego, że tematyka ta została przeze mnie szczegółowo opisana w ramach projektu o nazwie „NFC Lock”, który opublikowano na łamach naszego miesięcznika w wydaniu EP 04/22 (KIT AVT 5926) [1], w związku z czym zainteresowanych Czytelników odsyłam do tych materiałów. Jeśli zaś chodzi o nasz niniejszy projekt, to, jak napisałem na wstępie, w celu autoryzacji dostępu używa on alternatywnego sposobu w postaci zwykłego, 4-cyfrowego kodu PIN, którego wprowadzenie możliwe jest dzięki zastosowaniu standardowej klawiatury matrycowej. Tyle tytułem wstępu.

### Budowa i działanie

Przejdźmy do schematu ideowego naszego urządzenia, który pokazano na **rysunku 1**. Jak widać, zaprojektowano bardzo prosty system mikroprocesorowy,

**Wykaz elementów:** (kupuj na stronie sklep.avt.pl lub osobiście: Warszawa, ul. Leszczyńska 11, tel. +48222578451, e-mail: handlowy@avt.pl)

**Półprzewodniki:**

- U1: LD1117V33 (TO-220)
- U2: ATmega328 (DIP28)
- U3: moduł RFID/NFC typu PN532
- OLED: wyświetlacz OLED 2.42", kontroler SSD1309, interfejs SPI
- D1: 1N4148 (DO-35)
- T1: BC548 (TO-92)

**Rezystory:**

- R1: 1 kΩ (miniaturowy 1/8 W, raster 0,2")

**Kondensatory:**

- C1, C4, C5, C6: 100 nF ceramiczny (raster 0,1")
- C2, C3: 100 µF/16 V elektrolityczny (raster 0,1")

**Pozostałe:**

- K1: przekaźnik G5LE-14-9
- PWR, LOCK: złącze śrubowe AK500/2 (raster 0,1")
- KEYPAD: klawiatura numeryczna typu AK-804-A-BBW
- BUZZ: przetwornik piezoelektryczny SMD typu LPT9018BS-HL-03-4.0-12-R
- ADMIN: jumper THT

którego sercem jest bardzo popularny mikrokontroler firmy Microchip (dawniej Atmel) o oznaczeniu ATmega328 taktowany wewnętrznym, wysokostabilnym generatorem RC o częstotliwości 1 MHz. Mikrokontroler ten odpowiedzialny jest za sprzętową obsługę interfejsu SPI, dzięki któremu realizuje współpracę zarówno ze wspomnianym wcześniej modulem RFID/NFC, jak i bardzo efektywnym wyświetlaczem OLED o przekątnej 2,42" i rozdzielczości 128×64 piksele wyposażonym w sterownik ekranu typu SSD1309 (zbliżony do SSD1306) stanowiącym element graficznego interfejsu użytkownika.

I już w tym miejscu należy się kilka słów wyjaśnienia. Jak widzicie, oba peryferia (moduł RFID/NFC i wyświetlacz OLED) korzystają z tego samego, szeregowego interfejsu

SPI, zaś wybór aktywnego układu, dla którego przeznaczone są transmitowane dane (lub z którego odczytywane są dane – w przypadku modułu RFID/NFC), odbywa się tradycyjnie dzięki odrębnym wyprowadzeniom CS obu układów. Jest jednak pewien istotny haczyk. Otóż moduł RFID/NFC komunikuje się z mikrokontrolerem dzięki transmisji szeregowej SPI w kolejności od bitu najmłodszego (LSB), zaś wyświetlacz OLED w kolejności od bitu najstarszego (MSB). W takim wypadku każdorazowa zmiana docelowego peryferium wymaga reinicjalizacji modułu SPI po stronie mikrokontrolera. To bardzo istotny „drobiazg”, który łatwo pominąć, jeśli nie studujemy dokładnie dokumentacji producenta, co z kolei uniemożliwiłoby nam nawiązanie poprawnej komunikacji ze wspomnianymi wcześniej podzespołami.

Przejdźmy zatem do naszego modułu OLED. Zastosowanie wyświetlacza tego typu wynikało z chęci wyposażenia naszego urządzenia w efektywny, a zarazem bardzo czytelny interfejs użytkownika, co nie jest bez znaczenia w przypadku obsługi urządzeń tego typu. Co istotne, wyświetlacz, o którym mowa, ma bardzo przystępną cenę na dalekowschodnim portalu aukcyjnym, która oscyluje w granicach 10 \$, a dodatkowo dostępny jest w kilku możliwych kolorach, przez co łatwo dopasować docelowy design urządzenia do naszych upodobań czy wymagań klienta. Jedyne, na co należy zwrócić uwagę, to położenie i rozkład wyprowadzeń, gdyż w tym zakresie mogą występować nieznaczące różnice wynikające z kolejnych wersji wspomnianego peryferium.

Zastosowanie mikrokontrolera tego rodzaju wyposażonego w dość sporą wielkość pamięci Flash (jak na skromne AVR-y) nie wynikało z objętości programu obsługi aplikacji (który de facto zajmuje jedynie około 30% tej pamięci, z czego zdecydowana większość to tablice czcionek ekranowych i stosowanych piktogramów), a z niezbędnej wielkości zintegrowanej pamięci EEPROM, w której będziemy przechowywać dane autoryzacyjne użytkowników (ich kody PIN i numery seryjne kart RFID/NFC).

Wróćmy zatem ponownie do naszego schematu ideowego. Mikrokontroler poza zadaniami wspomnianymi powyżej realizuje obsługę klawiatury matrycowej stanowiącej element interfejsu użytkownika, jak i obsługę buzzera piezoelektrycznego niewyposażonego w zintegrowany generator, a to dzięki wykorzystaniu wbudowanego w mikrokontroler układu czasowo-licznikowego Timer1 pracującego w trybie CTC i generującego przebieg o częstotliwości 4 kHz na wyprowadzeniu PB1 (OC1A) układu. Ponadto serce naszego systemu steruje pracą przekaźnika K1 (poprzez stopień tranzystorowy T1), który w zamierzeniach sterować ma elektromechanicznym rygłem pozwalającym na dostęp do chronionego obiektu, jak i obsługuje zworkę ADMIN, zadaniem której jest wprowadzenie systemu multiLock w tryb wprowadzania kodu/karty RFID/NFC administratora.

Implementacja takiej sprzętowej zwory była intencjonalna a wynikała z chęci zabezpieczenia systemu przed przypadkowym wejściem w tryb wprowadzania kodu/karty RFID/NFC administratora, wszak tego rodzaju media autoryzacyjne umożliwiają w dalszej kolejności pełną kontrolę nad listą dostępnych użytkowników i ich danych autoryzacyjnych. Co więcej, aktywna zworka ADMIN umożliwia wykasowanie całej pamięci użytkowników (poza danymi autoryzacyjnymi Administratora).

Tyle w kwestii schematu ideowego, w związku z czym przejdźmy do zagadnień programistycznych. Tak jak wspomniano wcześniej, nie będę tutaj powielał informacji

Listing 1. Plik nagłówkowy modułu obsługi klawiatury matrycowej

```
//Definicje portu klawiatury
#define KEYPAD_PORT PORTD
#define KEYPAD_PIN PIND
#define KEYPAD_DDR DDRD
#define COL1 (1<<PD2)
#define COL2 (1<<PD0)
#define COL3 (1<<PD4)
#define ROW1 (1<<PD1)
#define ROW2 (1<<PD6)
#define ROW3 (1<<PD5)
#define ROW4 (1<<PD3)

//Definicja stałej nieprzyciśniętego klawisza
#define NOT_PRESSED 255

//Definicja znaków specjalnych keypad-a
#define CODE_GWIAZDKA 10
#define CODE_HASH 11
```

Listing 2. Funkcja odpowiedzialna za inicjalizację obsługi klawiatury matrycowej

```
void keypadInit(void) {
    //Podciągnięcie wierszy klawiatury pod VCC
    KEYPAD_PORT |= ROW1|ROW2|ROW3|ROW4;
}
```

Listing 3. Definicje stałych modułu obsługi klawiatury matrycowej

```
//Tablica kolumn
const uint8_t Cols[3] = {COL1, COL2, COL3};
//Tablica wierszy
const uint8_t Rows[4] = {ROW1, ROW2, ROW3, ROW4};
//Tablica zwracanych kodów
const uint8_t Codes[12] = {1, 2, 3, 4, 5, 6, 7, 8, 9, CODE_GWIAZDKA, 0, CODE_HASH};
```

Listing 4. Funkcja odpowiedzialna za odczyt kodu wciśniętego klawisza klawiatury matrycowej

```
uint8_t keypadRead(void) {
    uint8_t Key = NOT_PRESSED;

    //Ustalamy numer wciśniętego klawisza
    for(uint8_t Col=0; Col<3; Col++) {
        //Wybraną kolumnę ustawiamy, jako wyjściową ze stanem 0
        KEYPAD_DDR |= Cols[Col];
        //Szukamy czy nie zawarto jej do wybranego wiersza
        for(uint8_t Row=0; Row<4; Row++) {
            if(!(KEYPAD_PIN & Rows[Row])) {
                Key = Col + (Row*3);
                break;
            }
        }
        //Wybraną kolumnę ustawiamy, jako wejściową
        KEYPAD_DDR &= ~Cols[Col];
    }
    _delay_ms(30);

    //Wszystkie kolumny ustawiamy jako wyjścia ze stanem 0,
    //żeby sprawdzić czy nadal naciskamy jakiś klawisz
    KEYPAD_DDR |= COL1|COL2|COL3;
    //Sprawdzamy czy jakkolwiek klawisz nadal jest naciśnięty
    if((KEYPAD_PIN & ROW1) && (KEYPAD_PIN & ROW2)
        && (KEYPAD_PIN & ROW3)
        && (KEYPAD_PIN & ROW4)) Key = NOT_PRESSED;
    else if(Key != NOT_PRESSED) Key = Codes[Key];
    //Wszystkie kolumny ustawiamy jako wejścia
    KEYPAD_DDR &= ~(COL1|COL2|COL3);

    return Key;
}
```

Listing 5. Funkcja inicjalizacyjna sterownika ekranu OLED o przekątnej 2.42" wyposażonego w sterownik ekranu SSD1309

```

void OLEDinit(void) {
    //wszystkie porty jako wyjściowe
    //CS w stanie "1" (inactive), a RES w stanie "0" (reset)
    OLED_PORT |= (1<<OLED_CS);
    OLED_DDR |= (1<<OLED_DC)|(1<<OLED_CS)|(1<<OLED_RES);

    _delay_ms(10);
    OLED_SET_RES;
    _delay_ms(100);

    SSD1309writeCmd(DISPLAY_OFF_CMD);
    SSD1309writeCmd(SET_START_LINE_CMD | 0x00);

    SSD1309writeCmd(SET_CONTRAST_CMD);
    SSD1309writeCmd(0x32); //Contrast

    SSD1309writeCmd(SET_SEG_REMAPING_CMD | 0x01);
    SSD1309writeCmd(SET_COM_SCAN_REMAPPED_CMD);
    SSD1309writeCmd(NORMAL_DISPLAY_CMD);

    SSD1309writeCmd(SET_MULTIPLEX_RATIO_CMD);
    SSD1309writeCmd(0x3F); //1/64 duty

    SSD1309writeCmd(SET_DISPLAY_OFFSET_CMD);
    SSD1309writeCmd(0x00); //Not offset

    SSD1309writeCmd(SET_DISPLAY_CLOCK_DIV_CMD);
    SSD1309writeCmd(0xA0);

    SSD1309writeCmd(SET_PRECHARGE_PERIOD_CMD);
    //Set Pre-Charge as 15 Clocks & Discharge as 1 Clock
    SSD1309writeCmd(0xF1);

    SSD1309writeCmd(SET_COM_PINS_CMD);
    SSD1309writeCmd(0x12);

    SSD1309writeCmd(SET_VCOMH_DESELECT_CMD);
    SSD1309writeCmd(0x30); //Set VCOM Deselect Level

    SSD1309writeCmd(MEMORY_MODE_CMD);
    SSD1309writeCmd(0x00); //Horizontal addressing mode

    SSD1309writeCmd(CHARGE_PUMP_CMD);
    SSD1309writeCmd(0x14); //Disable

    SSD1309writeCmd(DISPLAY_ALLON_RESUME_CMD);
    SSD1309writeCmd(NORMAL_DISPLAY_CMD);
    SSD1309writeCmd(DISPLAY_ON_CMD);

    OLEDcls();
}

```

Listing 6. Plik nagłówkowy modułu obsługi wyświetlacza OLED 2,42"

```

//OLED RESOLUTION
#define OLED_WIDTH 128
#define OLED_HEIGHT 64

//OLED port definitions
#define OLED_PORT PORTC
#define OLED_DDR DDRC

#define OLED_CS PC3 //CHIP SELECT
#define OLED_DC PC4 //DATA(1)/COMMAND(0)
#define OLED_RES PC5 //RESET

#define OLED_SET_CS OLED_PORT |= (1<<OLED_CS)
#define OLED_RESET_CS OLED_PORT &= ~(1<<OLED_CS)

//DDRAM MODE
#define OLED_SET_DC OLED_PORT |= (1<<OLED_DC)
//COMMAND MODE
#define OLED_RESET_DC OLED_PORT &= ~(1<<OLED_DC)

#define OLED_SET_RES OLED_PORT |= (1<<OLED_RES)
#define OLED_RESET_RES OLED_PORT &= ~(1<<OLED_RES)

//OLED Commands
#define SET_CONTRAST_CMD 0x81
#define DISPLAY_ALLON_RESUME_CMD 0xA4
#define DISPLAY_ALLON_CMD 0xA5
#define NORMAL_DISPLAY_CMD 0xA6
#define INVERSE_DISPLAY_CMD 0xA7
#define DISPLAY_OFF_CMD 0xAE
#define DISPLAY_ON_CMD 0xAF

#define SET_DISPLAY_OFFSET_CMD 0xD3
#define SET_COM_PINS_CMD 0xDA

#define SET_VCOMH_DESELECT_CMD 0xDB

#define SET_DISPLAY_CLOCK_DIV_CMD 0xD5
#define SET_PRECHARGE_PERIOD_CMD 0xD9

#define SET_MULTIPLEX_RATIO_CMD 0xA8

#define SET_START_LINE_CMD 0x40

#define MEMORY_MODE_CMD 0x20
#define SET_COLUMN_ADDR_CMD 0x21
#define SET_PAGE_ADDR_CMD 0x22

#define SET_COM_SCAN_NORMAL_CMD 0xC0
#define SET_COM_SCAN_REMAPPED_CMD 0xC8

#define SET_SEG_REMAPING_CMD 0xA0
#define CHARGE_PUMP_CMD 0x8D

```

na temat obsługi modułu RFID/NFC, gdyż takowe są bardzo obszerne i zebrano je w ramach mojego wcześniejszego artykułu, o którym wspomniałem powyżej, a skupię się na elementach nowych.

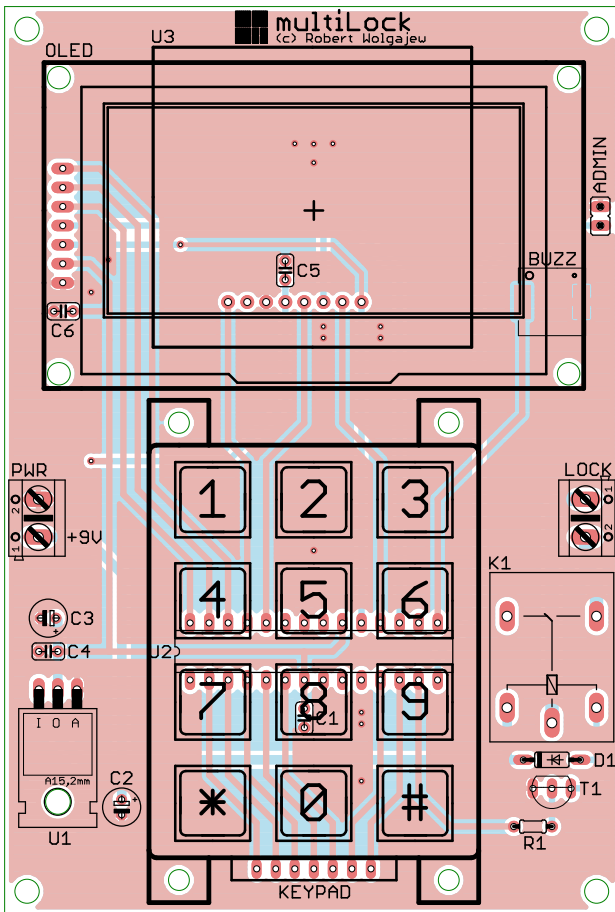
Tym nowym elementem jest obsługa klawiatury matrycowej. W sumie to nic trudnego, a już na pewno nic odkrywczego, ale pokażę Wam, jak w prosty sposób rozwiązałem tego rodzaju zagadnienie, zaznaczając, że nie jest to rozwiązanie uniwersalne, a szyte na miarę niniejszego projektu, gdzie nie ma potrzeby detekcji faktu wciśnięcia kilku klawiszy klawiatury w jednym czasie. Zaczniemy od pliku nagłówkowego do obsługi tegoż peryferium upraszczającego i porządkującego późniejszy kod, który to plik pokazano na **listingu 1**.

Dalej, na **listingu 2**, pokazano ciało funkcji odpowiedzialnej za inicjalizację obsługi klawiatury matrycowej, której jedynym zadaniem jest podciągnięcie stosownych portów wejściowych do plusa zasilania (przez rezystory podciągające zlokalizowane wewnątrz struktury mikrokontrolera).

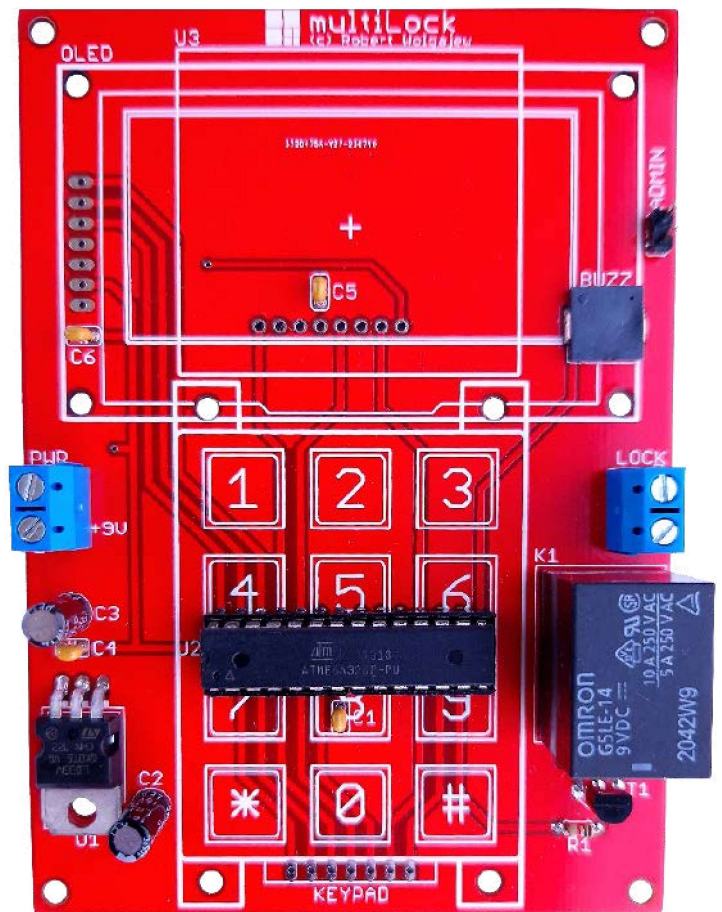
Zanim przejdę do funkcji pozwalającej na odczyt kodu przyciśniętego przycisku, niezbędne jest wprowadzenie kilku zmiennych (a w zasadzie stałych) upraszczających ten mechanizm. Stałe te zadeklarowano w pamięci RAM, co może wydawać się pewnym marnotrawstwem tej pamięci, ale nasze urządzenie korzysta z niewielkiej ilości pamięci RAM, w którą wyposażony jest mikrokontroler, przez co zadeklarowanie tych stałych w tej pamięci (zamiast w pamięci Flash) nie wydaje się jakimś kardynalnym błędem, a z pewnością przyspiesza dostęp do ich wartości. Definicje wspomnianych stałych pokazano na **listingu 3**.

I na koniec zagadnień implementacyjnych związanych z obsługą klawiatury matrycowej funkcja odpowiedzialna za odczyt kodu wciśniętego klawisza klawiatury matrycowej, której ciało pokazano na **listingu 4**.

Tyle w kwestii samej klawiatury. Zanim jednak przejdę do zagadnień montażowych, kilka niezbędnych słów uwagi na temat zastosowanego wyświetlacza OLED. Zgodnie z tym, co napisałem wcześniej, wyświetlacz nasz wyposażono w sterownik ekranu typu SSD1309 produkcji firmy Solomon Systech Limited, który w swojej konstrukcji jest bardzo zbliżony do dobrze znanego i chętnie używanego przez amatorów (jak i naszych chińskich przyjaciół) sterownika SSD1306. Pisząc bardzo zbliżony, mam na myśli, że sposób obsługi obu układów jest identyczny, ale różnią się one nieco możliwościami sprzętowymi, jak i procedurą inicjalizacji. Jako że w Internecie znajduje się mnóstwo przykładów obsługi obu sterowników, ograniczę się wyłącznie do prezentacji kodu funkcji inicjalizacyjnej, która jest charakterystyczna (w sensie wysyłanych ustawień)



Rysunek 2. Schemat montażowy urządzenia multiLock



Fotografia 1. Wygląd obwodu drukowanego urządzenia multiLock tuż przed przyłutowaniem modułu RFID, wyświetlacza OLED i klawiatury matrycowej (w wersji prototypowej)

wyłącznie dla naszego modułu o przekątnej 2,42". Ciało wspomnianej funkcji pokazano na **listingu 5**. Aby zrozumieć działanie funkcji inicjalizacyjnej, niezbędna jest znajomość pliku nagłówkowego, którego treść pokazano na **listingu 6**. Tyle w kwestiach implementacyjnych.

## Montaż i uruchomienie

Przejdźmy do schematu montażowego urządzenia, który został pokazany na **rysunku 2**. Jak widać, zaprojektowano bardzo zwartą konstrukcję obwodu drukowanego z wyłącznym zastosowaniem elementów THT (poza buzerem), po to, by całe urządzenie swoimi wymiarami nie przekraczało niezbędnego, minimalnego obszaru dla wykonania interfejsu użytkownika a jednocześnie było proste w konstrukcji. W tym celu w wybranych miejscach zastosowano montaż „piętrowy”, w związku z czym kolejność implementacji ma tutaj szczególne znaczenie.

Warto również podkreślić, że dla zminimalizowania zakłóceń, na płytce urządzenia poprowadzono obszerne pola masy po obu stronach obwodu drukowanego oraz zastosowano szereg przelotek pomiędzy nimi w celu zmniejszenia pojemności pasożytniczych.

Montaż urządzenia rozpoczynamy od przyłutowania mikrokontrolera (najlepiej w odpowiedniej podstawie), stabilizatora napięcia

(U1), pozostałych elementów półprzewodnikowych, a na koniec elementów pasywnych (w tym przekaźnika) i złącza PWR, LOCK oraz jumpersa ADMIN. W dalszej kolejności montujemy klawiaturę matrycową, korzystając z długiego złącza GOLDPIN, które zapewnia jej niezbędne połączenie elektryczne oraz 4 tulejek dystansowych o wysokości 13 mm (wraz ze stosownymi śrubami/nakrętkami), które zapewniają jej konieczną stabilizację mechaniczną oraz pozycjonują ją w taki sposób, by znajdujące się na jej powierzchni klawisze umiejscowione były powyżej korpusu przekaźnika.

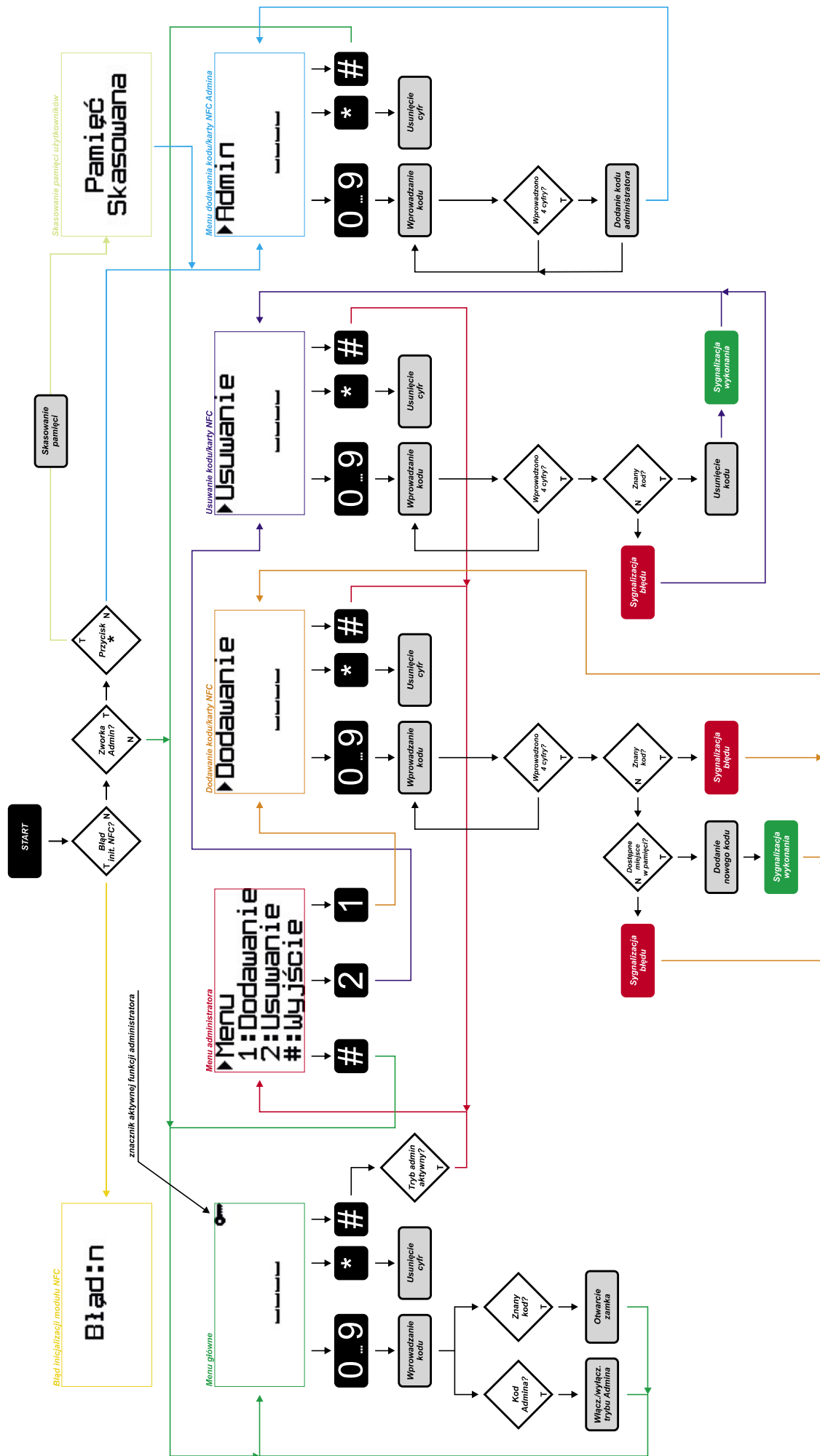
Kolejnym krokiem procesu montażowego jest montaż modułu RFID/NFC, co wykonujemy w taki sposób, by znalazł się on najbliżej, jak to się da, od dolnej powierzchni montowanego w dalszym kroku wyświetlacza OLED (także korzystając z dedykowanego złącza GOLDPIN, tym razem krótkiego). W tym momencie ustawiamy w odpowiednich pozycjach przełączniki SMD umieszczone na module RFID, dzięki którym wybieramy aktywny interfejs komunikacyjny (w naszym wypadku SPI). Aby przygotować moduł do pracy z interfejsem SPI, przełączamy przełącznik oznaczony jako ON w pozycję On (blisko znacznika „1”), zaś przełącznik oznaczony jako KE w pozycję Off (blisko znacznika „KE”).

W ostatnim kroku montujemy wyświetlacz OLED, korzystając, jak wcześniej, z długiego złącza GOLDPIN zapewniającego mu niezbędne połączenie elektryczne oraz 4 tulejek dystansowych o wysokości 12 mm (wraz ze stosownymi śrubami/nakrętkami), które zapewniają mu konieczną stabilizację mechaniczną. Sam wyświetlacz przysłoni co prawda moduł RFID/NFC, lecz testy praktyczne wykazały, iż nie zawiera on elementów, które w sposób znaczący mogłyby zaburzyć funkcjonowanie anteny umieszczonej na tym peryferium.

Poprawnie zmontowany układ nie wymaga żadnych regulacji (poza koniecznością ustalenia kodu PIN/karty RFID/NFC administratora) i powinien działać tuż po włączeniu zasilania.

W przypadku zastosowania przekaźnika innego typu niż przewidziany w projekcie, należy użyć tulejki dystansowej o innej wysokości, niż podano powyżej dostosowanej, co oczywiste, do wysokości korpusu wspomnianego elementu.

Na **fotografii 1** pokazano wygląd obwodu drukowanego urządzenia multiLock tuż przed przyłutowaniem modułu RFID, wyświetlacza OLED i klawiatury matrycowej (w wersji prototypowej, przed wprowadzeniem drobnych poprawek w zakresie położenia elementów).



Rysunek 3. Kompletny diagram prezentujący sposób obsługi urządzenia multiLock, jak i rodzaje dostępnych ekranów menu

**Ustawienia Fuse-bitów:**

CKSEL3...0: 0010  
 SUT1...0: 10  
 CKOUT: 1  
 CKDIV8: 0  
 EESAVE: 0

**Obsługa**

Konstruując algorytm obsługi urządzenia multiLock oraz wygląd graficznego interfejsu użytkownika, chciałem maksymalnie uprościć niezbędne czynności po stronie użytkownika systemu, zachowując jednocześnie spójność, jak i atrakcyjność wizualną poszczególnych ekranów funkcyjnych. Tuż po włączeniu zasilania urządzenie multiLock przechodzi do niezmiernie prostego ekranu głównego, w zakresie którego możemy wprowadzić PIN-kod użytkownika (lub administratora), jak też obsłużyć kartę RFID/NFC (odczytać jej numer seryjny). Co oczywiste, wprowadzenie poprawnego kodu PIN lub zbliżenie obsługiwanej karty RFID/NFC powoduje chwilowe załączenie wbudowanego przekaźnika, który w swoich zamierzeniach ma sterować ryglem elektromechanicznym.

Klawisze „0...9” służą w tym przypadku do wprowadzania kodu użytkownika, klawisz „\*” kasuje wszystkie wprowadzone cyfry, zaś klawisz „#” wywołuje menu administratora, ale tylko wtedy, gdy aktywny jest tryb administratora. Wprowadzane cyfry kasowane są również automatycznie po czasie około 15 s bezczynności po stronie użytkownika.

Tryb administratora włączamy/wyłączamy (cyklicznie) z kolei za każdym razem wprowadzenia kodu administratora lub wczytania karty administratora, co sygnalizowane jest pojawieniem się ikonki klucza na ekranie graficznego interfejsu użytkownika. Po wejściu w tryb administratora dostępne mamy następujące opcje: dodawanie kodu/karty RFID/NFC użytkownika (wywoływane poprzez naciśnięcie przycisku „1”), usuwanie kodu/karty RFID/NFC użytkownika (wywoływane poprzez naciśnięcie przycisku „2”) oraz wyjście do ekranu głównego (wywoływane poprzez naciśnięcie przycisku „#”).

Po wejściu w menu dodawania kodu użytkownika ukazuje się prosty interfejs graficzny umożliwiający wprowadzenie nowo obsługiwanego kodu użytkownika czy też nowej karty RFID/NFC. Wprowadzanie i usuwanie cyfr przebiega analogicznie jak w przypadku ekranu głównego. Poprawne wykonanie operacji dodania kodu/karty użytkownika sygnalizowane jest dwukrotnym krótkim dźwiękiem wbudowanego buzzera piezoelektrycznego, zaś błędne wykonanie tejże operacji długim pojedynczym dźwiękiem wspomnianego buzzera.

Błąd w trakcie dodawania nowego kodu/karty RFID/NFC użytkownika może się pojawić wyłącznie w dwóch przypadkach: gdy nowo wprowadzany kod/karta został już

**Tabela 1. Rodzaje błędów modułu PN532 i ich znaczenie funkcjonalne**

Numer błędu	Rodzaj błędu
1	Błąd ramki potwierdzenia (ACK) modułu PN532
2	Przekroczenie czasu (1 s) odpowiedzi modułu PN532
3	Błąd nagłówka ramki wersji modułu PN532
4	Błąd konfiguracji RF
5	Błąd konfiguracji modułu SAM (Security Access Module)
6	Błąd odczytu tagu NFC
7	Błąd autoryzacji
8	Błąd odczytu bloku danych

wcześniej zapisany w pamięci urządzenia lub gdy wyczerpano limit miejsc w pamięci urządzenia przeznaczonych na tego rodzaju dane. Limit ten to odpowiednio (i niezależnie od siebie): 100 kodów PIN oraz 100 numerów seryjnych kart RFID/NFC. Wyjście z menu dodawania nowego kodu/karty użytkownika następuje poprzez naciśnięcie przycisku „#”.

Jeśli chodzi o drugą z opcji menu administratora, a mianowicie możliwość usuwania kodów/kart RFID/NFC użytkownika, to jest ona bliźniaczo podobna do dodawania tychże kart/kodów, przy czym w tym wypadku wystąpienie błędu towarzyszącego procesowi usuwania możliwie jest wyłącznie w przypadku, gdy usuwany kod/karta nie jest znany przez nasz system mikroprocesorowy, w związku z czym nie może zostać usunięty z pamięci.

Nie wspominałem jeszcze o specjalnym menu umożliwiającym wprowadzenie kodu/karty administratora, które wywoływane może być wyłącznie w trakcie włączania urządzenia, a następuje to na skutek założenia zworki ADMIN na stosownym goldpinie umieszczonym na obwodzie drukowanym. Po wejściu w menu dodawania kodu/karty RFID/NFC administratora postępujemy analogicznie, jak przy dodawaniu normalnego kodu PIN czy też karty użytkownika, przy czym w tym przypadku nie są generowane żadne sygnały błędów, gdyż wprowadzane dane nadpisują każdorazowo istniejące dane tego rodzaju. Wyjście z menu dodawania nowego kodu/karty administratora następuje poprzez naciśnięcie przycisku „#”.

Kolejną opcją dostępną w przypadku obecności zworki ADMIN na stosownym goldpinie umieszczonym na obwodzie drukowanym jest możliwość wykasowania całej pamięci kodów/numerów seryjnych kart, którą to opcję wywołujemy, wciskając i przytrzymując klawisz „\*” podczas włączania urządzenia. W takim przypadku cała pamięć PIN-kodów i numerów seryjnych kart RFID/NFC (poza danymi autoryzacyjnymi administratora) zostanie wykasowana, na ekranie urządzenia pojawi się stosowna informacja, po czym system przejdzie do menu dodawania kodu/karty administratora, z którego, jak

wcześniej, wychodzimy, wciskając klawisz „#” (jeśli nie chcemy zmieniać danych autoryzacyjnych administratora).

Na koniec kilka słów uwagi należy się potencjalnej możliwości obsługi urządzeń NFC w rodzaju telefonu komórkowego. Jak wiadomo, niektóre z tych urządzeń wyposażono w interfejs NFC umożliwiający wymianę danych lub obsługę płatności zbliżeniowych. Czy nasz prosty system będzie w stanie odczytać numer seryjny takiego urządzenia? Oczywiście, że tak, bo musi ono spełniać standardy protokołu NFC. Problem jednak w tym, że systemy operacyjne telefonów komórkowych generują losowe numery seryjne (tzw. RID od Random ID), w związku z czym ich zapamiętywanie nie ma większego sensu, gdyż podczas kolejnego połączenia z takim urządzeniem wygeneruje ono zupełnie inny numer seryjny.

Zachowanie to można zmienić, ale przynajmniej w Androidzie nie jest to takie proste i wymaga modyfikacji firmware lub użycia specjalnych aplikacji. Ponoć zdarzają się telefony ze statycznym numerem UID (takie informacje odnalazłem na specjalistycznych forach), lecz ja na takie nie natrafiłem, więc należy założyć, że standardowo generują one losowe numery RID.

I na koniec jeszcze jedna, istotna informacja. Uruchamianiu urządzenia multiLock towarzyszy inicjalizacja wbudowanego modułu czytnika kart RFID/NFC. Jest to proces dość skomplikowany i jeśli zakończy się z jakiegoś powodu niepowodzeniem (np. nie ustawiliśmy na module odpowiedniego interfejsu komunikacyjnego), to stosowne informacje wraz z kodem błędu pokazane zostaną w ramach graficznego interfejsu użytkownika, zaś samo urządzenie przejdzie do trybu bezczynności (nie będzie można go używać). Obsługiwane kody błędów wraz z ich znaczeniem opisano w **tabeli 1**.

Na **rysunku 3** pokazano z kolei kompletny diagram prezentujący sposób obsługi urządzenia multiLock i rodzaje dostępnych ekranów menu.

**Robert Wołgajew, EP**

[1] <https://ep.com.pl/projekty/projekty-ep/15235-nfc-lock>