Spis treści

[1. CzytnikKart 1](#_Toc41614090)

[2. Wyswietlacz 3](#_Toc41614091)

[3. GSM 5](#_Toc41614092)

[4. Klawiatura 6](#_Toc41614093)

# CzytnikKart

Po uruchomieniu systemu, system przechodzi w stan czuwania, ustawia attempt (ilość blednych odczytow na 0), sprawdza czy nie została przylozona karta.

Jeżeli karta została przyłożona, do zmiennej typu bool CheckChard zostaje przypisana wartość z funkcji ReadCard() („pozytywny odczyt wartość 1, negatywny wartość 0).

1. Odczyt negatywny ->wzrost licznika błędnych odczytów o 1,

Wyświetlenie funkcja WarningRead komunikatu o błędzie,

Przekazanie do kontrolera funkcja WrongRead(), danych o błędnym odczycie karty.

Jeżeli ilość błędnych odczytów przekroczy wartość 3, funkcja BlockCardReader(), blokujemy możliwość odczytu karty, nadajemy komunikat o błędzie autoryzacji AutorisationERROR(), blokujemy klawiaturę (BlockPinPad(), informujemy dzwiekowym komunikatem o błędzie odczytu SoundError().

1. Odczyt pozytywny

Ustawiamy ilość błędnych odczytów na 0 (attempt=0;),

Blokujemy czytnik kart(BlockCardReader),

Sprawdzamy czy pin został wprowadzony, (CheckPin() i podajemy wartość z funkcji do zmiennej typu bool Check

1. Jeżeli Check ma wartość False

Zwiekszamy wartość licznika błędnych wprowadzen pinu o 1,

Informujemy o błędnym pinie WrongPin()

Oraz informujemy dzwiekowo o błędzie WarningPin()

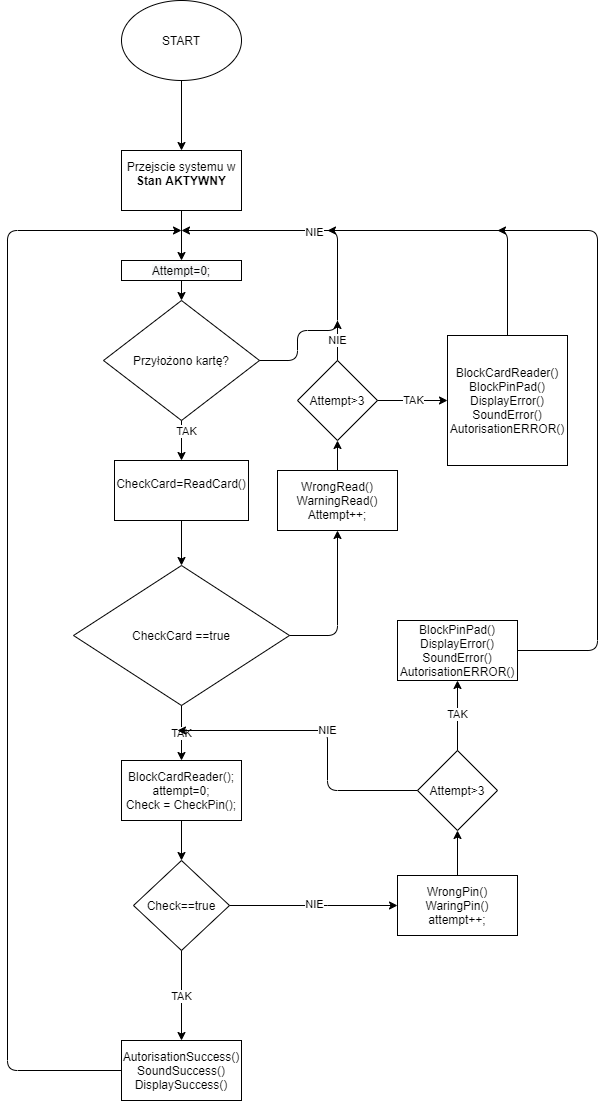
Jeżeli wartość attempt przekroczy wartość 3 wykona się lista poleceń analogicznie do odczytu karty -> błąd autoryzacji , komunikaty oraz blokada pin pada

1. Jeżeli Check ma wartość TRUE

Nadajemy komunikat o udanej autoryzacji -> AutorisationSuccess()

Wyswietlamy komunikat o poprawnej autoryzacji -> DisplaySuccess()

Inforumujemy dzwiekiem o udanej autoryacji -> SoundSuccess().



Rysunek 1 Czytnik Kart

# Wyswietlacz

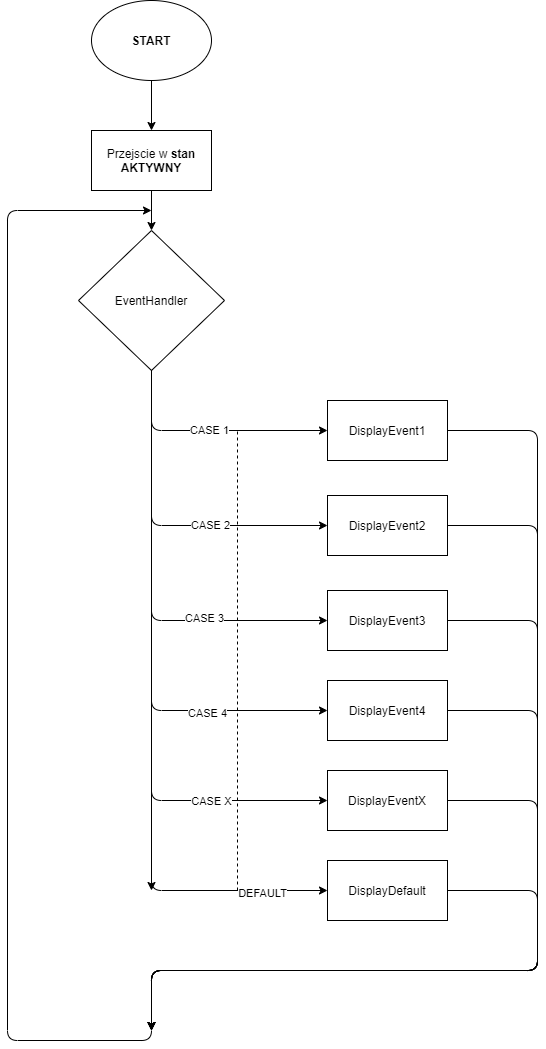
Wyswietlacz działa na zasadzie przelacznika (w kodzie – switch).

Na podstawie zmiennej EventHandler, przechwytuje zadana wartość wydazenia które ma obsluzyc,

Na podstawie tej zmiennej wyświetlacz posiada przelacznik, mogący obsłużyć X rożnych wydażen do wyświetlenia jak np. błędny pin, informacje o pinie który właśnie jest wprowadzany itp…

Posiada również domyślny stan (Default), który będzie wyświetlany w czasie czuwania systemu.

Po obsłudze zdażenia, wyświetlacz prechodzi w stan czuwania.



Rysunek 2 Wyświetlacz

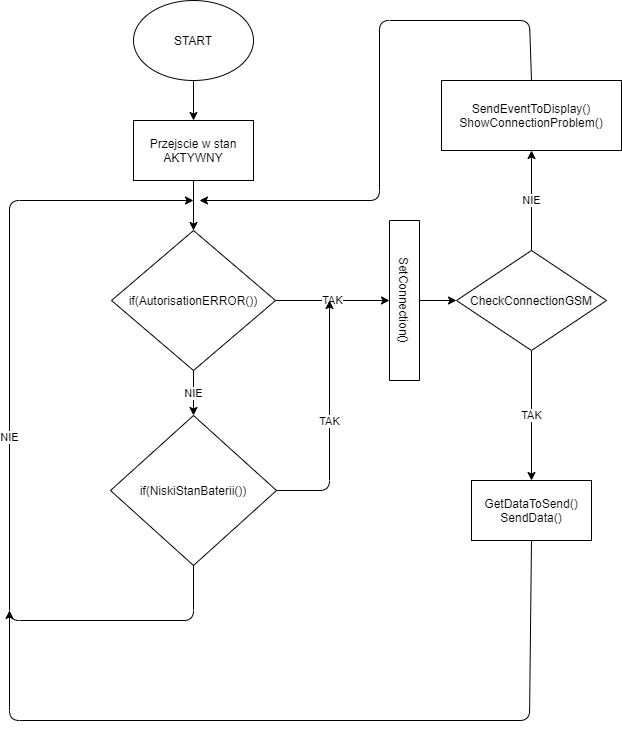
# GSM

Po przejściu do stanu AKTYWNY algorytm sprawdza czy funkcja AutorisationERROR() – zwrocila wartość true, jeżeli nie sprawdza czy została zwrocona wartość przez funkcje NiskiStanBaterii(),

Jeżeli żaden z warunków nie zostanie sprawdzony powtarzaj sprawdzanie.

Jeżeli ktorys z warunków zwrocil wartość TRUE, Wtedy nastepuje ustawienie połączenia GSM z siecia właściciela, jeżeli połączenie zostanie utworzone, program pobierze dane o błędzie GetDataToSend(), a następnie przesle poprzez GSM SendData() wiadomość SMS do użytkownika.

Natomiast jeżeli nie udało się utworzyć połącznia, program przekaże zdażenie do wyświetlacza o błędzie połączenia GSM, które na podstawie odpowiedniego nr. Event wyda polecenie do wyświetlenia odpowiednich informacji. Dodatkowo GSM zacznie sygnalizowac dioda/dźwiękiem komunikat o błędzie ShowConnectionProblem(), po czym ponownie sprobuje uzystak połączenie.



Rysunek 3 GSM

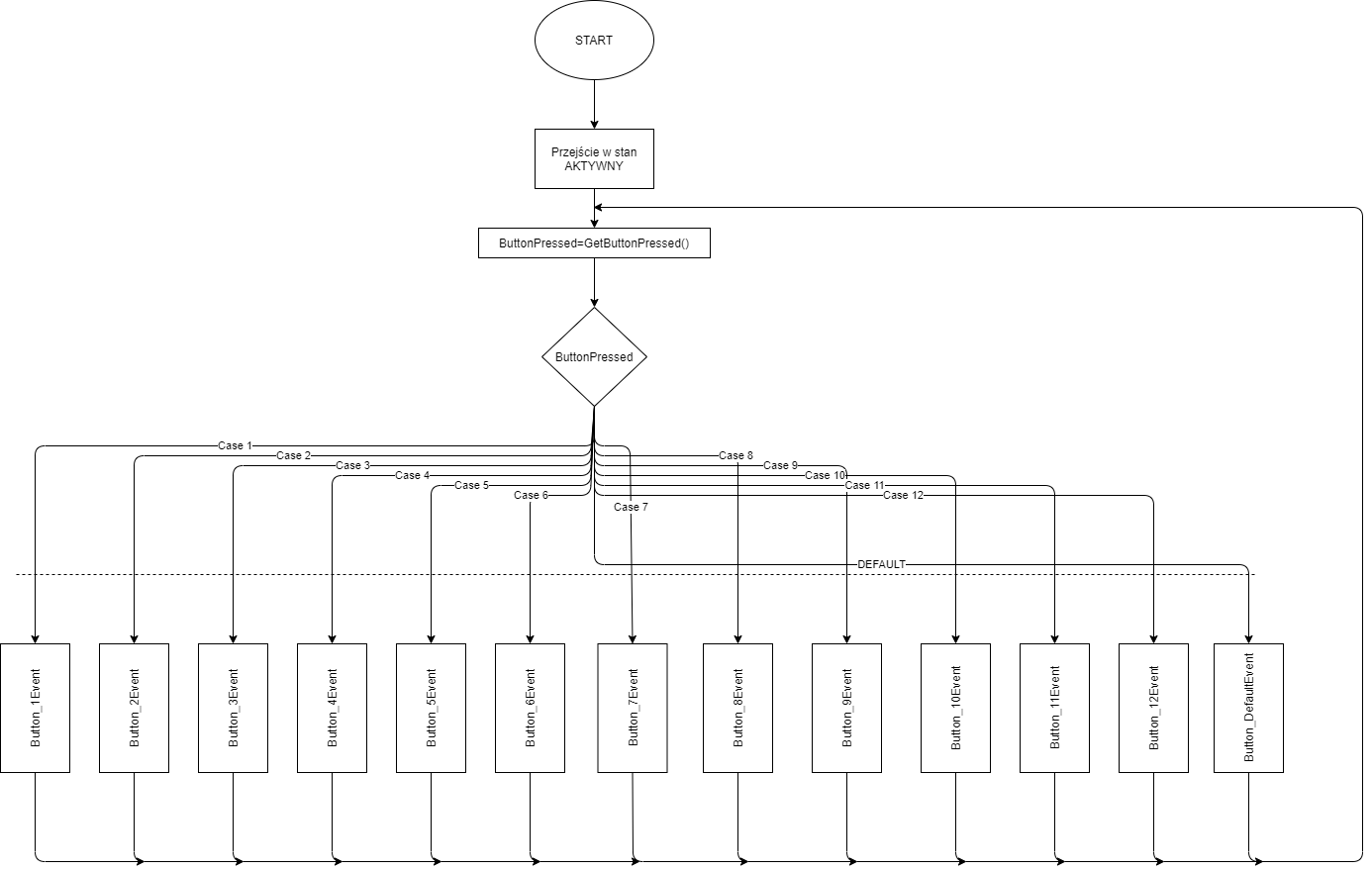
# Klawiatura

Klawiatura działa na bardzo prostym algorytmie składającym się z dwóch głownych cześć.

1. Funkcji pobierającej kliknięty przycisk GetButtonPressed() oraz przypisujaca wartość przycisku do zmiennej ButtonPressed.
2. Druga częścią tego algorytmu jest switch obsługujący napiśniecie każdego z przycisków.

Dla każdego przycisku jest przypisana inna funkcja regulujące działanie całego programu.

Przełącznik posiada również opcje domyślną DEFAULT, która posiada również swoja funkcje w raznie nie kliknięcia żadnego przycisku z klawiatury (stan czuwania).

Po każdorazowym nacisnieciu przycisku i obsłużeniu nacisnietego przycisku, algorytm będzie czekał na kolejny przycisk celem powtorzenia operacji obsługi przycisku.

Rysunek 4 Klawiatura