

# SYSTEMY WBUDOWANE - PROJEKT

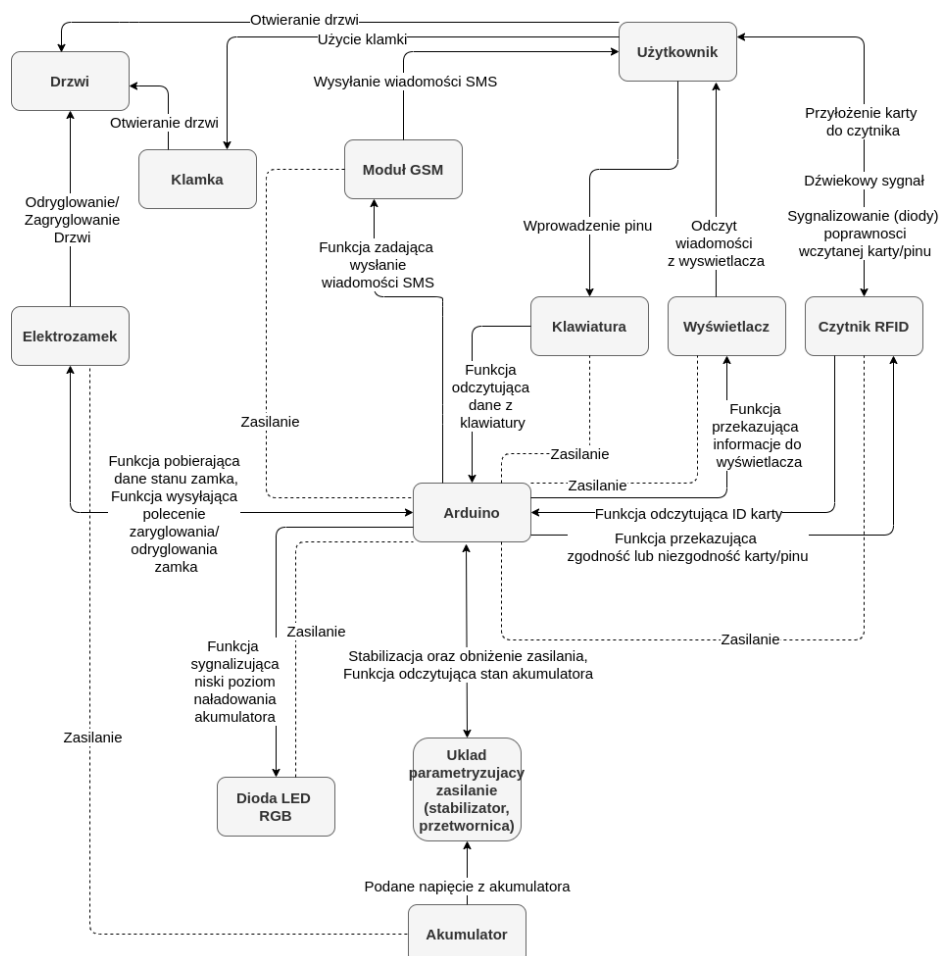
## System blokady drzwi i kontroli dostępu

### Schemat komunikacji

## 1 Wstęp

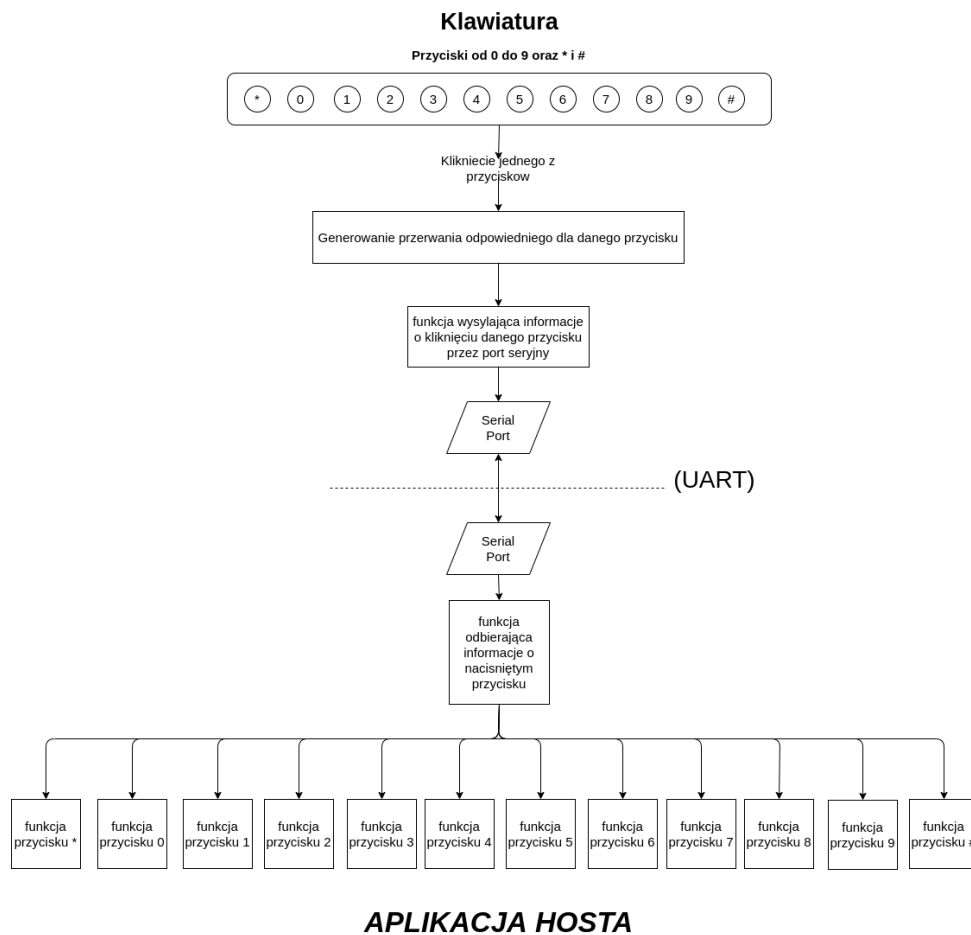
W tym pliku przedstawiony jest schemat połączeń komunikacyjnych i zasilania, określony protokół, jakim będzie się odbywać wymiana informacji, oraz jakie to będą informacje.

## 2 Ogólny schemat działania systemu



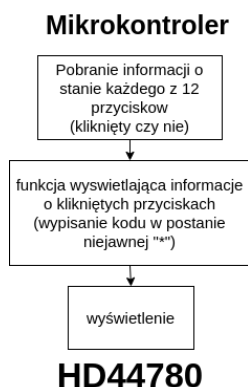
Rysunek 1: Schemat przedstawiający ogólne funkcjonowanie systemu

### 3 Schemat połączenia klawiatury z Arduino



## 4 Reakcja mikrokontrolera na klikane przyciski

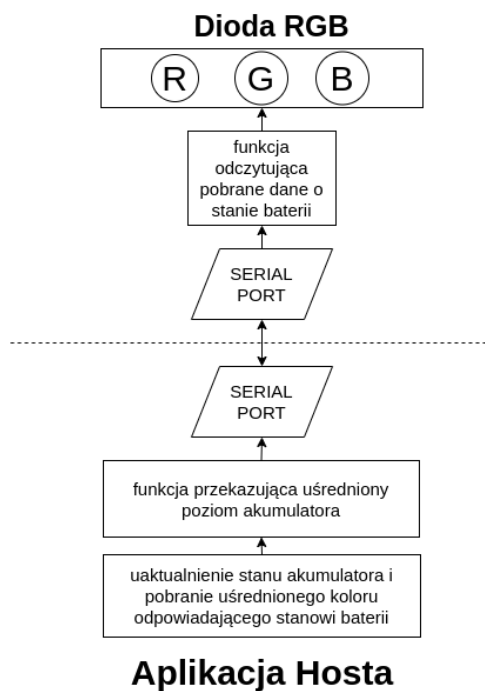
Następuje wypisywanie kodu w postaci niejawnej, czyli np. "\*\*\*\*\*".



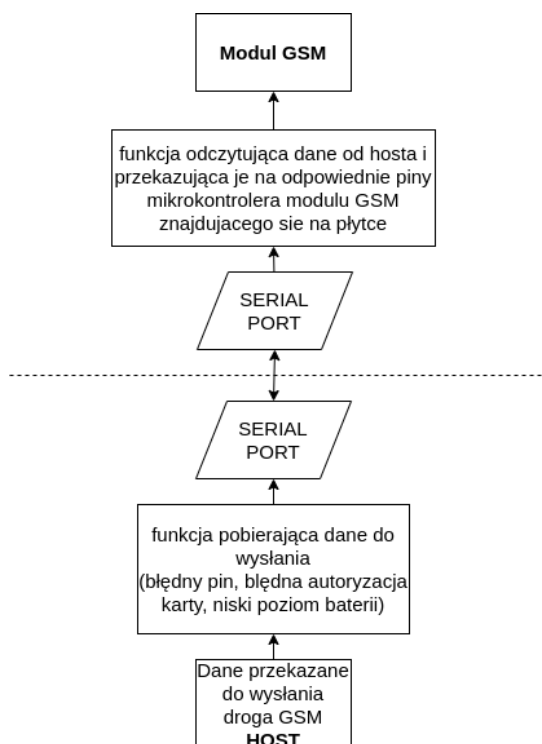
## 5 Sygnalizowanie diodą poziomu naładowania akumulatora

Sygnalizowanie diodą poziomu naładowania akumulatora odbywa się nie w postaci migiającej na niebiesko diody (jak to było założone w poprzednim szkicu projektu), lecz w postaci:

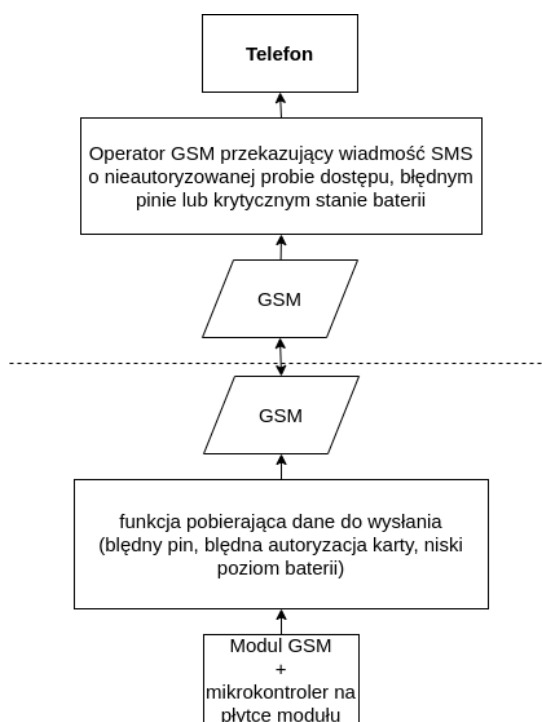
- koloru zielonego - oznacza naładowanie
- koloru pomarańczowego - oznacza częściowe rozładowanie
- koloru czerwonego - oznacza bardzo niski poziom baterii (sugerowana wymiana lub naładowanie akumulatora).



## 6 Komunikacja mikrokontrolera z modułem GSM



## 7 Przesyłanie informacji z modułu GSM na telefon



## 8 Parametry przesyłu danych przez UART



- **Baud Rate** - prędkość transmisji. Wartość domyślna, a więc 115200 będzie w zupełności wystarczająca dla wszystkich naszych wymagań (o ile urządzenie po drugiej stronie transmisji obsługuje taką wartość).
- **Word Length** - ilość bitów danych wraz z bitem parzystości przesyłanych w ramce. Zostawiamy domyślną wartość 8.
- **Parity** - obecność bitu parzystości. Nie będziemy próbować obsługiwać błędów zakomunikowanych przez niezgodność bitu parzystości
- **Stop Bits** - standardową wartością używaną w popularnych zastosowaniach jest 1 stop bitu, dlatego tu również zostawiamy wartość domyślną.
- **Data Direction** - jeżeli z jakiegoś powodu potrzebujemy jedynie możliwości nadawania lub odbierania, w tym miejscu możemy to skonfigurować. My będziemy korzystać z obydwu form, dlatego pozostawiamy domyślne Receive and Transmit.
- **Over Sampling** - w celu redukcji błędów transmisji wynikających chociażby z zakłóceń, stan każdego bitu pobierany jest kilku lub kilkunastokrotnie. W naszym wypadku może to być 8 lub 16 razy. Oczywiście większa wartość gwarantuje większą pewność poprawnej transmisji, natomiast mniejsza pozwala osiągnąć wyższe prędkości transmisji. Dostępne prędkości są dla nas wystarczające, pozostawiamy więc ten parametr z domyślną wartością, czyli 16.  
W razie potrzeby przeniesienia wprowadzenia TX oraz RX zastosujemy zmianę wyprowadzeń, czyli pin remapping w STM32 F4.

## 9 Informacje dodatkowe

Wszelkie połączenia między poszczególnymi komponentami są wykonane z miedzianych przewodów.

Połączenie między modułem GSM a telefonem wykorzystuje fale radiowe o częstotliwościach zgodnych z częstotliwościami modułu GSM tj. 850/900/1800/1900 Mhz.

Uwzględniamy, że host jako mikrokontroler Arduino, który steruje całym systemem, jest wyposażony w szereg funkcji odpowiedzialnych m.in. za zamykanie drzwi po określonym czasie (zliczającym cykle procesora i ustalonym przez osobę programującą urządzenie), zabezpieczenie, dzięki któremu nie ma możliwości, aby osoba uprawniona zatrzęsnęła się wewnątrz obiektu chronionego - w takim celu zastosowany jest elektrozamek oraz czujnik rygła, który pozwoli otworzyć drzwi od środka bez ponownej autoryzacji. Czyni to system bezpiecznym dla osoby uprawnionej.