

Rozdział 5

Projekt urządzenia

W rozdziale omówiony zostanie projekt fizyczny urządzenia (5.1) oraz projekt oprogramowania (5.2).

5.1. Projekt fizyczny

Urządzenie składa się z następujących komponentów:

1. ESP-WROOM-32 - mikrokontroler zamontowany na płytce rozwojowej zgodnej z ESP32-DevKit, jednostka główna.
2. SIM800L - moduł komunikacyjny GPRS.
3. MAX3232 - właściwie moduł z konwerterem napięć MAX3232 ze standardu TTL do RS232.
4. MP1584 - moduł z przetwornicą napięć step-down MP1584en.

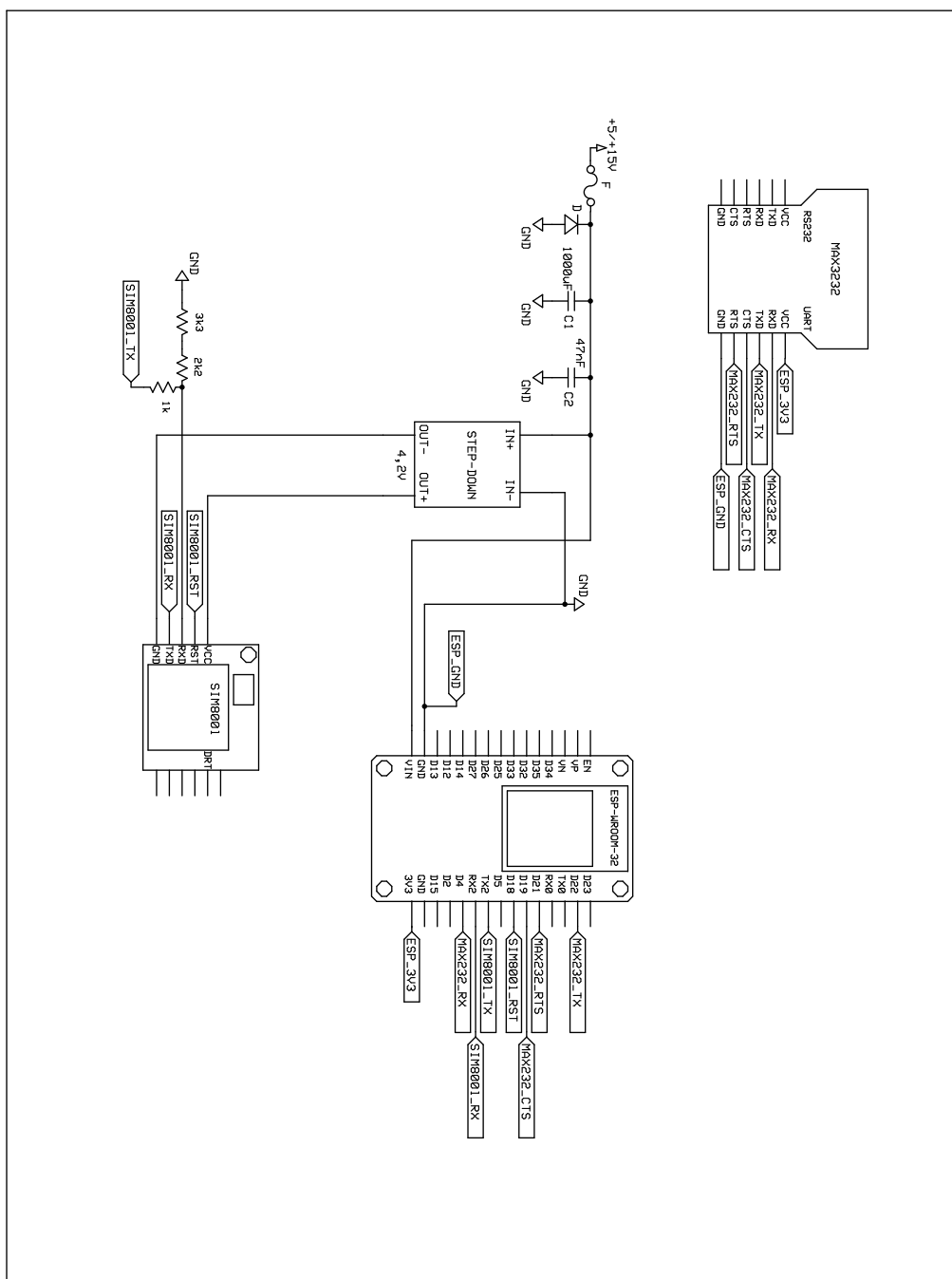
Rysunek 5.1 przedstawia schemat logiczny połączenia modułów. W urządzeniu wymagane było dodatkowej przetwornicy step-down z powodu innego napięcia zasilania poszczególnych podzespołów.

W celu zabezpieczenia układu przed zniszczeniem w razie odwrotnego podłączenia zasilania zastosowano bezpiecznik 3 A wraz z diodą o maksymalnym natężeniu 10 A [?]. Aby zapobiec zakłóceniom wysokiej częstotliwości użyto kondensator ceramiczny C2 o pojemności 47 nF, który stanowi filtr dolnoprzepustowy. Kondensator elektrolityczny C1 jest używany na wypadek wystąpienia obciążenia impulsowego. W razie wystąpienia impulsu np. w momencie logowania się do sieci komórkowej, ładunek zostanie pobrany z kondensatora, nie zakłócając działania całego urządzenia [?].

5.1.1. ESP-WROOM-32

ESP-WROOM-32 (ESP32) służy w projekcie jako jednostka główna sterująca pozostałymi modułami. Sam mikrokontroler zamontowany jest na płytce rozwojowej wzorowanej na (zgodnej z) ESP32-DevKit od oryginalnego producenta modułu (Espressif). Płytką rozwojową wprowadza kilka ułatwień dla „konstruktorów” urządzeń, na przykład:

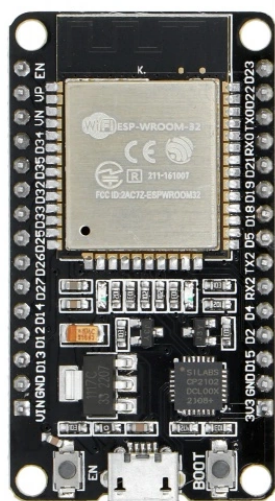
- wbudowany przetwornik napięć step-down (pin VIN, rysunek 5.2b), pozwala na zmianę napięć od 5 - 12 V na napięcie pozwalające na zasilanie mikrokontrolera,
- pin zasilania innych modułów napięciem 3,3 V (pin 3V3),
- piny pozwalające na podłączenie do (wspólnej) masy.



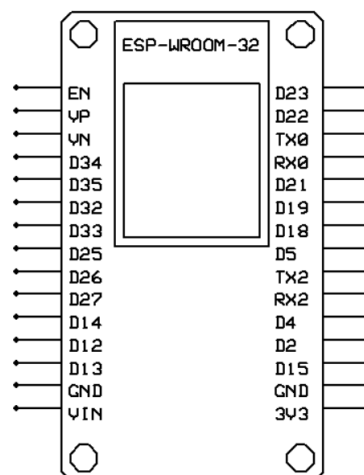
Rys. 5.1: Schemat połączenia modułów urządzenia

Piny od Dx (gdzie x to liczba całkowita) odpowiadają numerom pinów wyjścia/wejścia. Piny RXx są to domyślne piny stosowane do komunikacji UART. Piny VP oraz VN są pinami wejścia. EN jest wykorzystywany przy programowaniu mikrokontrolera.

Ponieważ aż ESP32 umożliwia komunikację poprzez UART na prawie wszystkich pinach we/wy, możliwe było podłączenie modułów GPRS oraz MAX3232 do dowolnych pinów. Piny oznaczone jako $RX0$ oraz $TX0$ (w programie mikrokontrolera domyślnie przypisane do interfejsu



(a)



(b)

Rys. 5.2: Zdjęcie używanego modułu (a) oraz jego uproszczony schemat (b)

UART0) stosowane są do programowania modułu oraz wyświetlania informacji o błędach. Z tego powodu pinów nie używano do komunikacji z innymi urządzeniami.

5.1.1.1. Charakterystyka napięciowa i prądowa

Rekomendowanym napięciem dostarczanym do układu ESP-WROOM-32 jest 3,3 V (tabela 5.1).

Tab. 5.1: Rekomendowane warunki operacyjne[?]

Symbol	Parameter	Min	Typical	Max	Unit
VDD33	Power supply voltage	3,0	3,3	3,6	V
IV DD	Current delivered by external power supply	0,5	-	-	A
T	Operating ambient temperature	-40	-	85	°C

Na podstawie dostarczanego napięcia możliwe jest obliczenie napięcia na pinach wy/we mikrokontrolera. Sposób obliczania jest opisany w tabeli 5.2, która została zapożyczona z dokumentacji technicznej układu[?].

Tab. 5.2: Napięcia na pinach wy/we

Symbol	Parameter	Min	Typ	Max	Unit
VIH	High-level input voltage	0.75xVDD	-	VDD+0.3	V
VIL	Low-level input voltage	-0.3	-	0.25xVDD	V
VOH	High-level output voltage	0.8xVDD	-	-	V
VOL	Low-level output voltage	-	-	0.1xVDD	V

Wartości napięć na pinach wy/we dla VDD = 3,3 V określa tabela 5.3.

Przetwornica dołączona do ESP32 DevKit pozwala na konwersję podanego na wejściu napięcia do 3,3 V, odpowiednich do zasilania ESP32. Maksymalne napięcie wejściowe przetwornicy

Tab. 5.3: Obliczone napięcia na pinach wy/we

Symbol	Parameter	Min	Typ	Max	Unit
VIH	High-level input voltage	2,48	-	3,60	V
VIL	Low-level input voltage	-0,30	-	0,83	V
VOH	High-level output voltage	2,64	-	-	V
VOL	Low-level output voltage	-	-	0,33	V

wynosi 15 V, minimalne natomiast 5 V [?]. Minimalny prąd wyjściowy przetwornicy wynosi 800 mA, natomiast maksymalny 1300 mA.

Zalecana wydajność prądowa źródła zasilającego ESP32 wynosi 500 mA lub więcej [?]. Największe pobory prądu wynikają z użycia funkcji WiFi oraz Bluetooth, szczególnie podczas transmisji danych (dla standardu 802.11b jest to nawet 240 mA; 802.11g - 190 mA).

5.1.1.2. Pozostałe specyfikacje

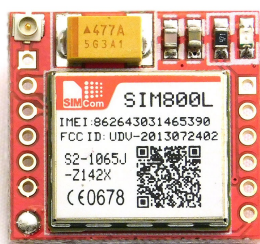
Układ ESP32 wyposażony jest w:

- dwa rdzenie o możliwym do regulacji taktowaniu 80 - 240 MHz,
- 520 KB pamięci SRAM (320 KB DRAM oraz 200 KB IRAM) [?],
- 16 KB pamięci RTC SRAM (dane zachowywane pomiędzy cyklami głębokiego snu),
- 448 KB pamięci ROM,

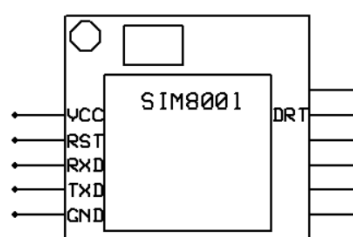
Dodatkowo moduł ESP-WROOM-32 rozszerza chip o 4 MB pamięci Flash, gdzie wgrywany jest m.in. kod programu.

5.1.2. SIM800l

Moduł SIM800l używany jest do komunikacji poprzez sieć komórkową GSM/GPRS. Wszystkie opisane tu informacje pochodzą ze specyfikacji producenta modułu[?].



(a)



(b)

Rys. 5.3: Zdjęcie używanego modułu (a) oraz jego uproszczony schemat (b)

Piny modułu, na którym zamontowany jest SIM800l opisane są w tabeli 5.4. Resetowanie modułu jest przeprowadzane poprzez ustawienie stanu niskiego na wyprowadzeniu RST.

5.1.2.1. Charakterystyka napięciowa i prądowa

Producent określa rekomendowane poziomy napięć zasilania modułu jako 3,4 - 4,4 V (tabela 5.5).

Tab. 5.4: Piny modułu i ich funkcje

Oznaczenie	Funkcja
VCC	Zasilanie modułu
RST	Reset modułu
RXD	Odbiór danych (UART)
TXD	Wysył danych (UART)
GND	Masa
DTR	Kontrola uśpienia modułu

Tab. 5.5: Charakterystyka napięciowa SIM8001

Symbol	Parameter	Min	Typ	Max	Unit
VBAT	Power supply voltage	3,4	4,0	4,4	V
TOPER	Operating temperature	-40	+25	+85	°C
TSTG	Storage temperature	-45		+90	°C

Zasilanie oprócz spełnienia wymagania w postaci specyficznego napięcia (4,0 V) musi potrafić wygenerować prąd o natężeniu co najmniej 2 A (tabela 5.6). Moduł pobiera prąd 2 A w trakcie łączenia się z siecią komórkową, zatem co najmniej raz przy włączaniu urządzenia.

Tab. 5.6: Charakterystyka prądowa SIM8001

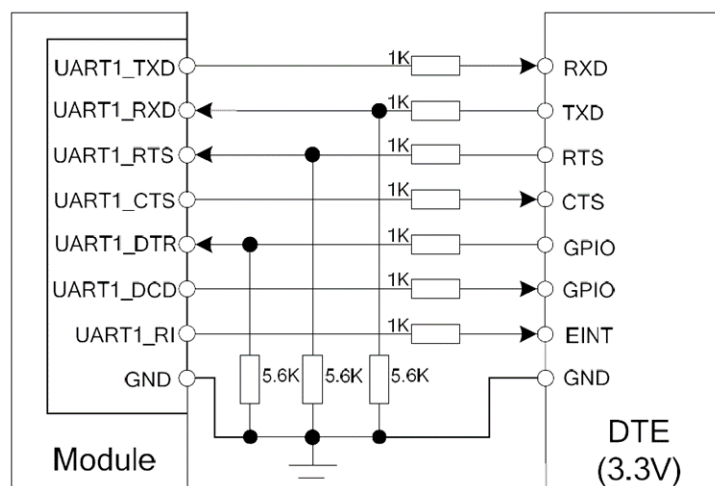
Symbol	Parameter	Conditions	Max	Unit
IMAX	Peak current	During TX burst	2,0	A

Urządzenie operuje różnymi poziomami napięć na interfejsach komunikacyjnych. Producent określa poziomy stanu logicznego niskiego oraz wysokiego dla wyjścia (VOL oraz VOH) i wejścia (VIL oraz VIH). Dane widoczne są w tabeli 5.7.

Tab. 5.7: Charakterystyka napięciowa interfejsu UART

Symbol	Min	Max	Unit
VIL	-0,3	0,7	V
VIH	2,1	3,1	V
VOL	-	0,4	V
VOH	2,4	-	V

Skorzystano zatem z rekomendacji producenta określającej sposób podłączenia modułu do urządzeń korzystających z logiki 3,3 V. Jak można zauważyć, rekomendacja zaleca użycie dzielnika napięcia. Zamiast rezystora 5,6 kΩ użyto dwóch rezystorów 2,2 kΩ oraz 3,3 kΩ.



Rys. 5.4: Rekomendowane podłączenie modułu do interfejsu UART urządzeń 3,3 V

5.1.2.2. Pozostałe specyfikacje

Moduł wspiera komunikację poprzez interfejs UART od 12000 bps do 460800 bps oraz użycie kontroli przepływu, zarówno sprzętową (hardware), jaki i programową (software). Prędkość połączenia nie musi być ustawiana jawnie - moduł sam określa prędkość przesyłu danych, ale tylko dla prędkości 12000 - 115200 bps.

SIM8001 obsługuje transmisję wieloszczelinową GPRS - klasa 12 (*GPRS multislot class 12*), zatem obsługuje do 4 szczelin przy transmisji w górę i w dół, przy czym przypisana liczba

szczelin nie może być większa niż 5. Wszystkie rodzaje kodowań używanych w GPRS mogą być obsługiwane (CS-1, CS-2, CS-3, CS-4). Maksymalna możliwa prędkość przesyłu danych poprzez sieć komórkową GPRS wynosi 85,6 kbps w dół (downlink) oraz w górę (uplink). Poniższa tabela przedstawia prędkość w górę/w dół w zależności od użycia danego kodowania oraz ilości szczelin.

Tab. 5.8: Prędkość przesyłu w zależności od kodowania oraz ilości szczelin

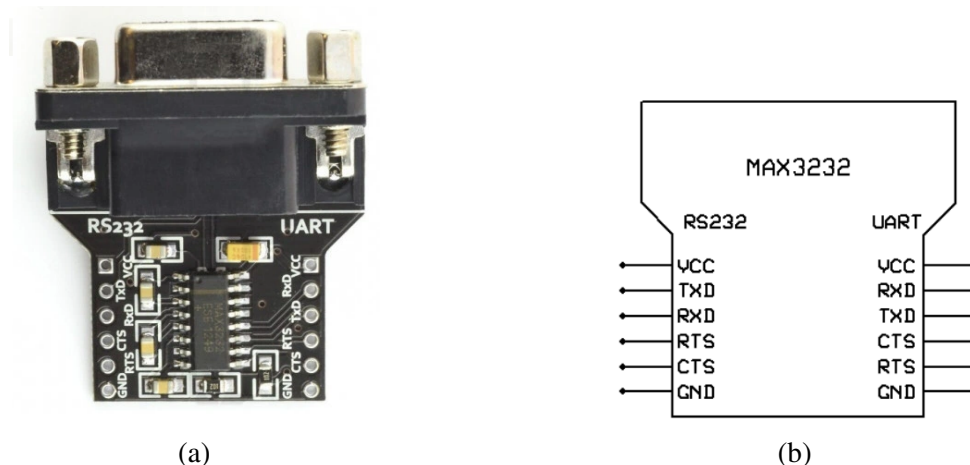
Coding scheme	1 timeslot	2 timeslot	4 timeslot
CS-1	9.05kbps	18.1kbps	36.2kbps
CS-2	13.4kbps	26.8kbps	53.6kbps
CS-3	15.6kbps	31.2kbps	62.4kbps
CS-4	21.4kbps	42.8kbps	85.6kbps

Moduł wspiera karty SIM działające na napięciach 3,0 oraz 1,8 V. Używana może być tylko karta Micro SIM (12x15 mm).

Gotowy moduł z zamontowanym SIM800l udostępnia możliwość podłączenia anteny ze złączem U.FL lub anteny helicznej. Aby moduł mógł współpracować z siecią GPRS, dołączona antena musi pracować na częstotliwościach 800 - 1900 MHz.

5.1.3. MAX3232

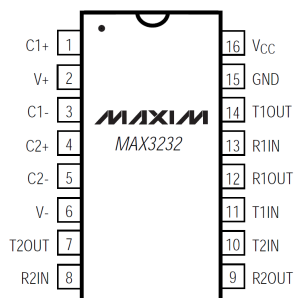
MAX3232 służy do zmiany napięć ze standardu TTL/CMOS do standardu RS232. Sama przetwornica jest częścią modułu z jednej części zakończonego żeńskim złączem D-SUB, a z drugiej listwą goldpin (męska). Wszelkie informacje o MAX3232 pochodzą ze specyfikacji producenta [?].



Rys. 5.5: Zdjęcie używanego modułu (a) oraz jego uproszczony schemat (b)

Poniżej opisano, powiązane z RS232, połączenia pomiędzy wyprowadzeniami modułu a przetwornicą. Dodatkowo w tabeli 5.9 opisano funkcje poszczególnych wyprowadzeń MAX3232 (opis pochodzi ze specyfikacji [?]).

1. T1IN połączone jest z TXD (prawa strona, UART).
2. T2IN połączone jest z RTS (*Ready To Send*) (prawa strona, UART).
3. T1OUT połączone jest z TXD (lewa strona, RS232) oraz do wyprowadzeń złącza D-SUB.
4. T2OUT połączone jest z RTS (lewa strona, RS232) oraz do wyprowadzeń złącza D-SUB.



Rys. 5.6: Schemat MAX3232

5. R1IN połączone jest z RXD (prawa strona, UART).
6. R2IN połączone jest z RTS (*Ready To Send*) (prawa strona, UART).
7. R1OUT połączone jest z RXD (lewa strona, RS232) oraz do wyprowadzeń złącza D-SUB.
8. R2OUT połączone jest z CTS (lewa strona, RS232) oraz do wyprowadzeń złącza D-SUB.

Tab. 5.9: Wyjścia przetwornicy MAX3232

PIN	NAME	FUNCTION
MAX3232		
1	C1+	Positive Terminal of Voltage-Doubler Charge-Pump Capacitor
2	V+	+5,5V Generated by the Charge Pump
3	C1-	Negative Terminal of Voltage-Doubler Charge-Pump Capacitor
4	C2+	Positive Terminal of Inverting Charge-Pump Capacitor
5	C2-	Negative Terminal of Inverting Charge-Pump Capacitor
6	V-	-5,5V Generated by the Charge Pump
7, 14	T_OUT	RS-232 Transmitter Outputs
8, 13	R_IN	RS-232 Receiver Inputs
9, 12	R_OUT	TTL/CMOS Receiver Outputs
10, 11	T_IN	TTL/CMOS Transmitter Inputs
15	GND	Ground
16	VCC	+3,0V to +5,5V Supply Voltage

5.1.3.1. Charakterystyka napięciowa i prądowa

Specyfikacja MAX3232 określa maksymalne (rysunek 5.7), minimalne oraz typowe (tabela 5.10), dopuszczalne na wejściach oraz generowane wyjściach, poziomy napięcie. Przy zasilaniu przetwornicy napięciem 3,3 V, można spodziewać się napięcia 3,2 V 5.10, natomiast napięcia maksymalnego 3,6 V. Napięcie jest zatem kompatybilne z jednostką główną, nawet przy zasilaniu modułu poprzez wbudowaną w ESP32 płytke rozwojową przetwornicą (maksymalne napięcie na wyprowadzeniach ESP32 wynosi 3,6 V).

Sygnał doprowadzany z jednostki głównej do MAX3232 (3,3 V) przetwarzany jest na standard RS232. Przetwornica wykonuje odwrócenie logiki sygnału, tj. sygnał o ujemnym napięciu oznacza logiczną 1, natomiast o napięciu dodatnim oznacza logiczne 0. Standard określa, że sygnał wysoki,

wysyłany przez sterownik, musi wynosić -5 - -15 V, sygnał niski natomiast 5 - 15 V [?]. Jak widać na rysunku 5.8a, moduł spełnia to wymaganie.

ABSOLUTE MAXIMUM RATINGS

VCC	-0.3V to +6V
V+ (Note 1)	-0.3V to +7V
V- (Note 1)	+0.3V to -7V
V+ + V- (Note 1)	+13V
Input Voltages	
T_IN, SHDN, EN	-0.3V to +6V
MBAUD	-0.3V to (VCC + 0.3V)
R_IN	±25V
Output Voltages	
T_OUT	±13.2V
R_OUT	-0.3V to (VCC + 0.3V)
Short-Circuit Duration	
T_OUT	Continuous

Rys. 5.7: Maksymalne dopuszczalne wartości napięć i natężeń MAX3232

Tab. 5.10: Charakterystyka napięciowa MAX3232

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
DC CHARACTERISTICS					
VCC Power-Supply Current	No load, VCC = 3,3V or 5,0V, TA = +25°C		0,3	1,0	mA
Input Logic Threshold High	VCC = 3,3V	2,0			V
	VCC = 5,0V	2,4			
Output Voltage Low	IOUT = 1,6mA			0,4	V
Output Voltage High	IOUT = -1,0mA	VCC - 0,6	VCC - 0,1		V

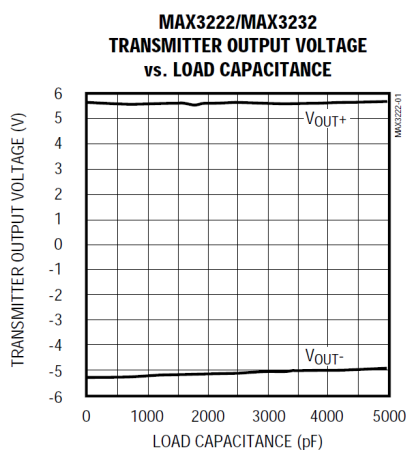
W zależności od prędkości komunikacji oraz pojemności obciążenia można spodziewać się różnego poziomu poboru prądu. Standard RS232 [?] określa maksymalną pojemność kabla, razem ze złączami jako 2500 pF, zatem można założyć, że urządzenie przy prawidłowym zastosowaniu zużywało będzie w przybliżeniu od 10 do 25 mA, w zależności od prędkości transmisji. Prąd wyjściowy przetwornicy dołączonej do ESP32 DevKit (1117C 33 D248), wynosi minimalnie 800 mA [?], zatem powinna ona być zdolna do zasilenia ESP32 oraz modułu MAX3232.

5.1.3.2. Pozostałe specyfikacje

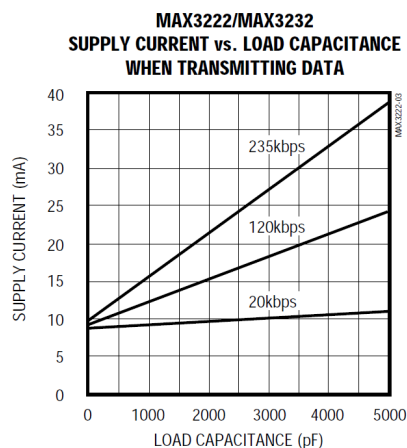
Maksymalna prędkość danych, możliwa do przetworzenia jest minimalnie 120 kbps, nie jest określona górna granica, jednak szybkość może osiągnąć typowo 235 kbps.

Tab. 5.11: Prędkość przesyłu danych MAX3232

PARAMETER	MIN	TYP	MAX	UNITS
Maximum Data Rate	120	235		kbps



(a) Napięcie wyjściowe (TOUT) (V) w zależności od pojemności obciążenia (pF)



(b) Pobór prądu (mA) w zależności od pojemności obciążenia (pF)

5.1.4. MP1584

Jako układ regulujący oraz stabilizujący napięcie zasilania modułu SIM800l użyto modułu przetwornicy step-down MP1584. Napięcie dostarczane na wejście modułu może wahać się od 4,5 - 28,0 V. Układ posiada regulację napięcia wyjściowego w postaci potencjometru. Możliwe jest uzyskanie napięcia wyjściowego równego 0,8 - 20,0 V. Maksymalne natężenie jakie może dostarczyć układ wynosi 3 A, jest to zatem wartość wystarczająca dla modułu SIM800l (5.1.2.1).

5.1.5. Źródło zasilania

Dla układu przedstawionego na rysunku 5.1, jako źródło zasilania może być użyta dowolna bateria/ladowarka spełniająca poniższe wymagania:

- prąd stały wytwarzany na wyjściu,
- napięcie wyjściowe 5 - 15 V,
- moc na poziomie 11,0 W.

Minimalna i maksymalna wartość napięcia podyktowana jest napięciami, z jakimi działa przetwornica ESP32 DevKit (5.1.1.1).

Moc wynika głównie ze zużycia prądu modułu SIM800l w trakcie łączenia z siecią (2 A przy 4,2 V (8,4 W), 5.1.2.1), jak również ze zużycia pozostałych modułów, choć te w porównaniu z SIM800 są niewielkie, rzędu setek mA. Minimalnym rekomendowanym natężeniem, które musi zapewniać źródło modułowi ESP32 wynosi 500 mA przy 3,3 V (1,65 W) (5.1.1.1). Wysokie zużycie ESP32 w większości przypadków wynika z użycia wbudowanej funkcji WiFi oraz Bluetooth (prądy rzędu setek mA); w projekcie te funkcje nie są używane. Prąd pobierany przez MAX3232 może wynieść maksymalnie 0,25 A przy 3,3 V (0,825 W).

5.2. Projekt oprogramowania

Urządzenie (ESP32) programowano przy użyciu frameworku *ESP-IDF*, stworzonym, rozwijanym i zalecanym przez producenta układu *Espressif*. Framework używa systemu czasu rzeczywistego dla urządzeń embedded *FreeRTOS*, które pozwala na wygodne tworzenie wątków i udostępnia proste API do komunikacji ich między sobą. Jest to obecnie jedyny możliwy sposób programowania układu.

podczas programowania układu skorzystano z dwóch bibliotek:

- *MbedTLS* - implementującej szyfrowanie, w tym w standardzie TLS1.3,
- *Eclipse Paho MQTT C/C++ client for Embedded platforms* - implementującą komunikację w standardzie MQTT, w tym MQTT v5.0.

Chociaż moduł SIM800l potrafi korzystać z szyfrowania, to używa on starszych standardów. Po wykonaniu najnowszej aktualizacji urządzenia, wspiera ono standard TLS1.2. Programowa implementacja szyfrowania pozwala również na bardziej elastyczne zmiany w razie stosowanych w przyszłości standardów. Użyto wbudowanej w ESP-IDF, zmodyfikowanej (wsparcie hardware'owe [?]) biblioteki *MbedTLS*, wersji v3.x. Podczas testów napotkano problemy z biblioteką w ESP-IDF v5.0. Po migracji do ESP-IDF v5.1, problemy nie pojawiły się ponownie.

Do obsługi MQTT wybrano *Eclipse Paho*, głównie z powodu braku innych bibliotek wspierających MQTT v5.0. Wykorzystano jedynie najprostszą część biblioteki „*MQTTPacket*”, która implementuje jedynie parsowanie/deparsowanie pakietów MQTT. Niskopoziomą obsługę zgodną ze standardem MQTT v5.0 zaimplementowano w ramach projektu.

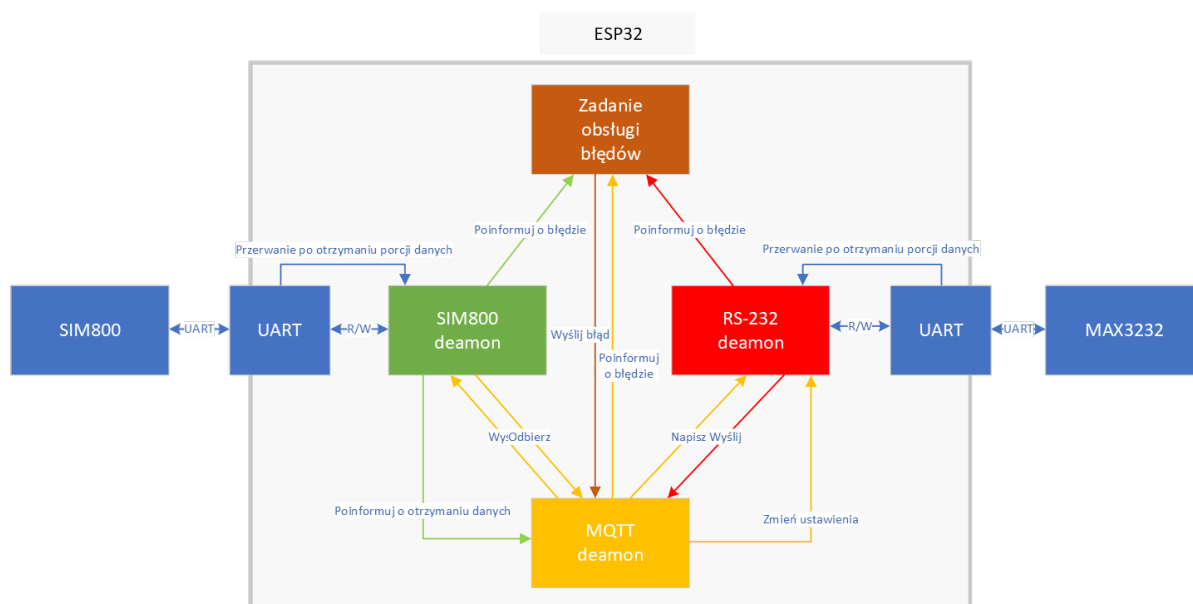
Taktowanie procesora ustawiono na 160 MHz.

5.2.1. Wątki

Oprogramowanie mikrokontrolera wykorzystuje cztery główne wątki:

1. Zadanie obsługi błędów - odbiera błędy wysyłane od innych zadań i na ich podstawie decyduje o uruchomieniu zadania ponownie lub restarcie całego urządzenia.
2. SIM800 daemon - niskopoziomowy sterownik modułu SIM800.
3. MQTT daemon - zadanie odpowiadające za obsługę komunikacji z serwerem za pomocą protokołu MQTT oraz wywoływanie odebranych komend.
4. RS323 demaon - obsługujące komunikację poprzez port RS232.

Rysunek 5.9 przedstawia w uproszczony sposób komunikację między zadaniami oraz interfejsami komunikacyjnymi.



Rys. 5.9: Uproszczona komunikacja między wątkami

Program mikrokontrolera rozpoczyna się od pobrania z pamięci urządzenia następujących danych:

- adres IP/DNS serwera,
- port serwera,

- nazwa użytkownika używana do logowania do systemu (adres MAC urządzenia),
- hasło,
- łańcuch certyfikatów.

Nazwa użytkownika jest pobierana poprzez użycie wbudowanej w ESP-IDF funkcji `esp_base_mac_addr_get()`, która odczytuje adres MAC z pamięci EFUSE mikrokontrolera. Pozostałe informacje są pobierane z partycji *Conf* o systemie plików SPIFFS i mogą zawsze zostać zmienione przez użytkownika.

Następnie uruchamiane są wątki. Procedura widoczna jest na rysunku 5.10.

5.2.2. Inicjalizacja modułu SIM800

Schemat blokowy widoczny na rysunku 5.11 przedstawia inicjalizację modułu SIM800.

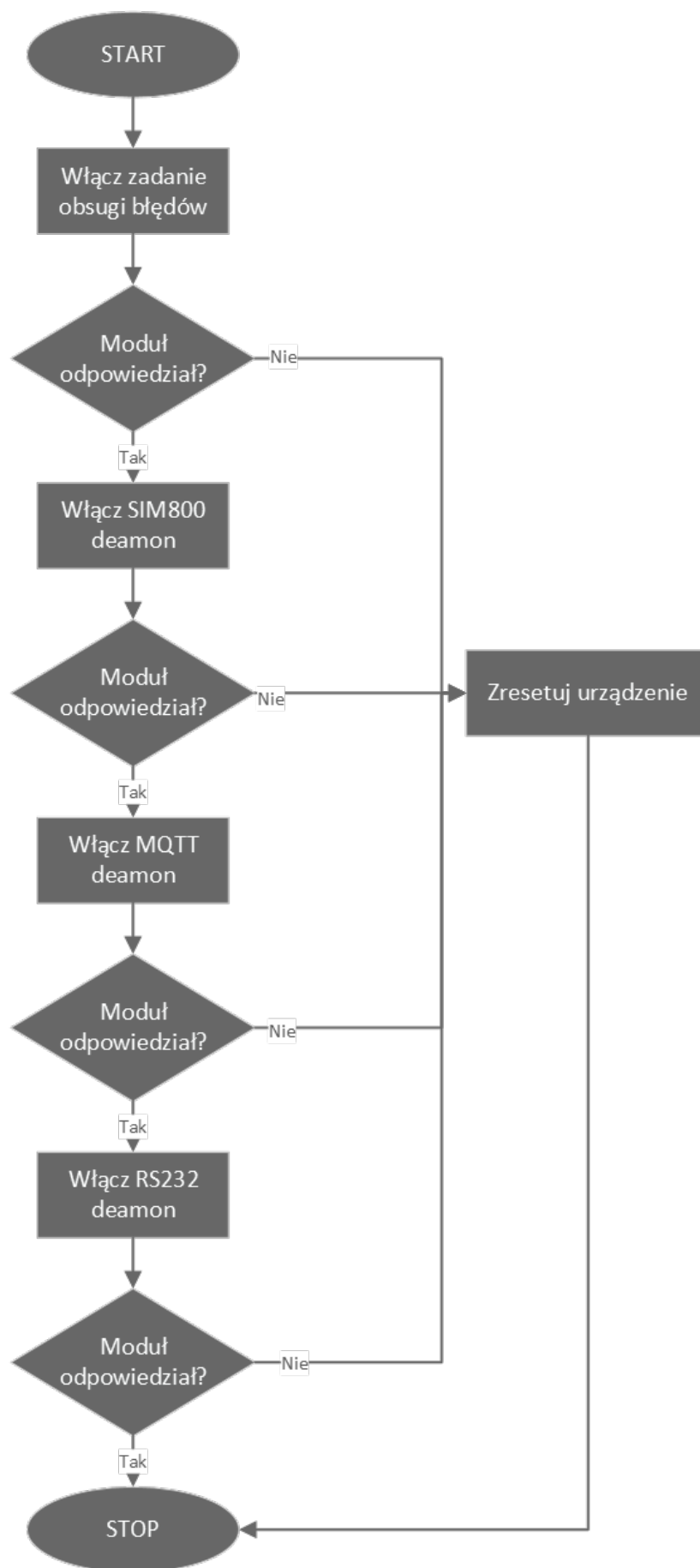
W tabeli 5.12 widoczne są komendy użyte podczas inicjalizacji modułu. Użyte komendy służą sprawdzeniu czy możliwe jest połączenie z serwerem oraz ustawieniu wymaganych parametrów połączenia z siecią GPRS. Oprócz pobierany jest również aktualny czas, dzięki czemu możliwa jest późniejsza weryfikacja ważności certyfikatu serwera.

Tab. 5.12: Komendy inicjalizacyjne

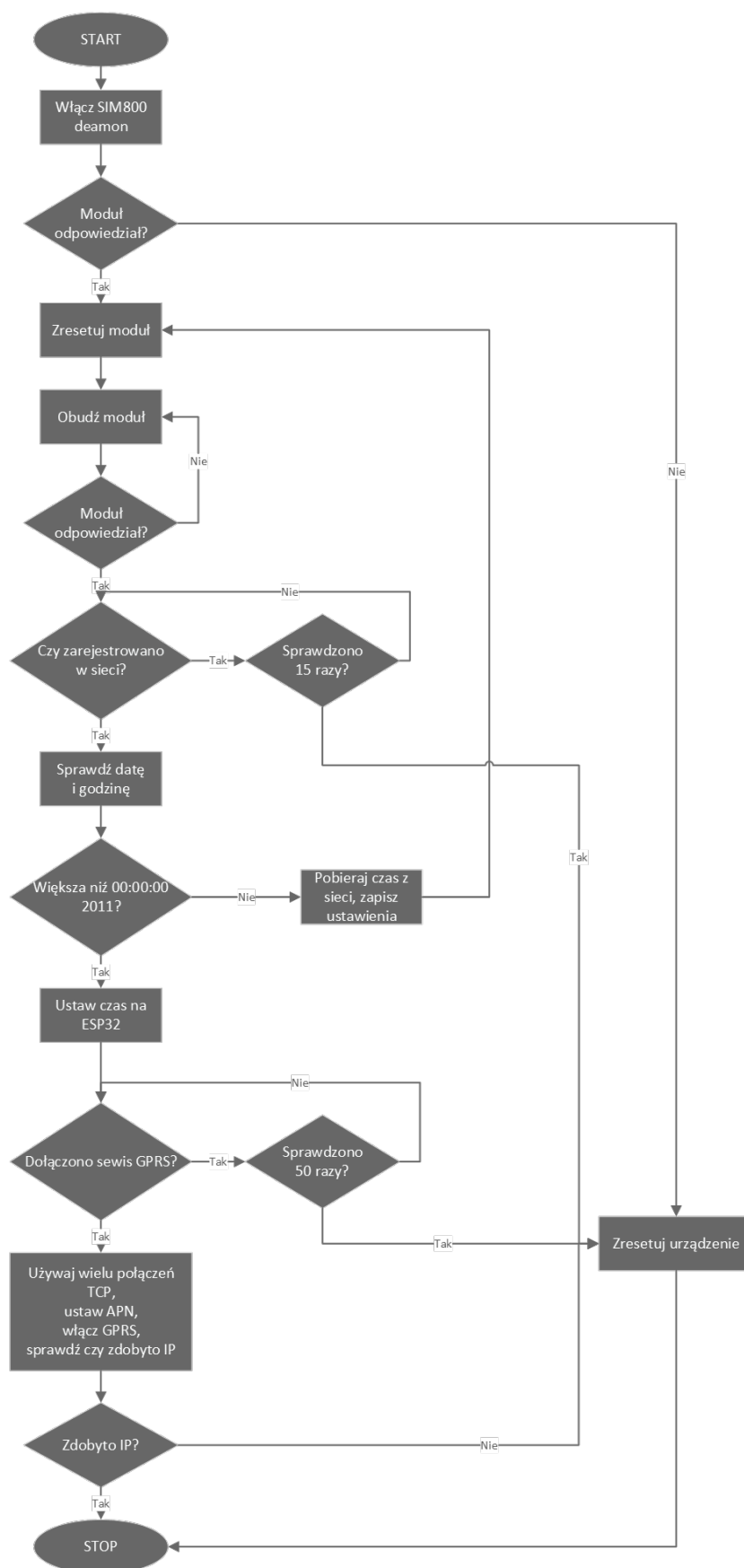
Komenda	Znaczenie
AT	Obudź moduł
AT+CREG?	Czy zarejestrowano w sieci?
AT+CCLK?	Sprawdź datę i godzinę
AT+CLTS=1	Pobieraj czas i godzinę z sieci
AT+W	Zapisz ustawienia
AT+CGATT?	Sprawdź czy dołączono serwis GPRS
AT+CIPMUX=1	Używaj wielu połączeń TCP
AT+CSTT=<apn>,<user name>,<password>	Ustaw APN
AT+CIICR	Włącz GPRS
AT+CISFR	Sprawdź adres IP

APN (punkt dostępowy, ang. *Access Point Name*) sieci komórkowej jest ustawiany przy użyciu komendy `AT+CSTT=<apn>,<user name>,<password>` (gdzie <apn>, <user name> i <password> to ciąg znaków). Nazwa punktu (<apn>), nazwa użytkownika (<user name>) oraz hasło (<password>) są odczytywane z pliku konfiguracyjnego urządzenia (5.2.4). W razie niepodania nazwy użytkownika APN lub hasła, parametry te nie będą używane (`AT+CSTT=<apn>`).

Po pomyślnym wykonaniu wymienionych komend możliwe jest połączenie się do serwera poprzez sieć komórkową.



Rys. 5.10: Rozpoczęcie programu



Rys. 5.11: Uruchomienie modułu SIM800

5.2.3. Ustanowienie połączenia z brokerem MQTT

Ustanowienie połączenia z brokerem odbywa się podczas tworzenia wątku „MQTT daemon” i składa się z kilku faz:

1. Ustanowienie połączenia TCP pomiędzy stronami.
2. Ustanowienie szyfrowanego połączenia, zgodnie z protokołem TLS1.3.
3. „Zalogowanie się” do brokera, zgodnie z protokołem MQTT v5.0.

Schemat na rysunku 5.12 przedstawia proces łączenia z brokerem MQTT.

Przed rozpoczęciem procesu łączenia z serwerem, inicjowana jest struktura, zawierająca najważniejsze informacje o wątku, np.:

- struktury biblioteki szyfrującej,
- bufor służący do parsowania wysyłanych pakietów,
- wskaźnik do funkcji callback, wywoływanej po otrzymaniu danych z kanału danych.

Ustanawiane jest połączenia TCP. Jeżeli uzyskano połączenie, wówczas inicjalizowana jest biblioteka *MbedTLS*:

- inicjalizowane są wymagane struktury,
- ustawiane są funkcje wejścia/wyjścia, z których korzystać będzie biblioteka do komunikacji z serwerem (tj. funkcje do komunikacji poprzez SIM800),
- weryfikowany jest wcześniej wczytany łańcuch certyfikatów.

Dodatkowo w bibliotece SIM800 ustawiany jest callback, który będzie „budził” wątek *MQTT daemon* po odebraniu danych od serwera.

Po inicjalizacji, MbedTLS przeprowadza „TLS handshake” z reverse proxy servera. Następnie wysyłana jest do brokera wiadomość CONNACK, parsowana przez bibliotekę *Eclipse Paho MQTT C/C++*. Ostatecznie uruchamiany jest wątek *MQTT daemon*, który będzie sterował komunikacją MQTT.

5.2.4. Ustawienia użytkownika

Zmienne informacje, dotyczące połączenia urządzenia z serwerem, są zaciągane z określonej partycji urządzenia o nazwie *Conf* i wielkości 65,536 kB, sformatowanej do systemu plików SPIFFS. System plików SPIFFS nie obsługuje folderów [?]. Partycja może zostać zawsze nadpisana przez użytkownika.

Tab. 5.13: Partycje w pamięci urządzenia

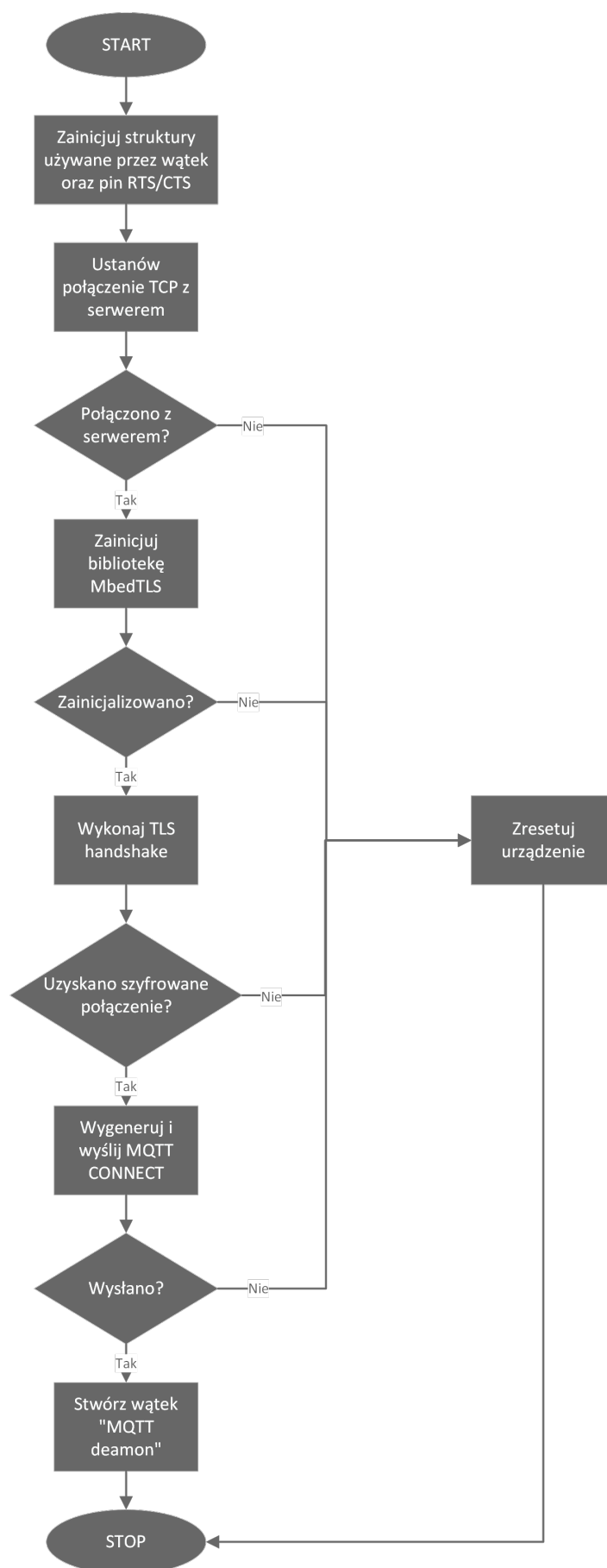
Nazwa	Typ	Subtyp	Offset	Wielkość
nvs	data	nvs	0x9000	0x6000
phy_init	data	phy	0xf000	0x1000
factory	app	factory	0x10000	1M
conf	data	spiffs		0x10000

Ustawienia są odczytywane z pliku *dev.conf*. Opcje muszą określać:

- password - hasło dostępu do brokera,
- server - IP/DNS serwera,
- port - port serwera,
- chain_size - długość łańcucha certyfikatów serwera,
- apn - punkt dostępowy sieci komórkowej,

- `apn_username` - nazwa użytkownika punktu dostępowego,
- `apn_password` - hasło punktu dostępowego.

Dodatkowo partycja musi zawierać pliki o nazwie *cert_x.crt*, gdzie *x* to numer certyfikatu w łańcuchu. Łańcuchem o numerze 0 jest *Root CA*. Kolejne są już *certyfikatami pośrednimi*.



Rys. 5.12: Inicjalizacja połączenia z serwerem