

## AIS DT

# Protokoły i algorytmy zarządzania energią oraz monitorowania (AIS.PA)

Załącznik – Protokół AIS dla agentów do komunikacji z centrum monitorowania

2020.09.30

© 2020 Instytut Nauki i Techniki Stipendium. All rights reserved.

## Spis treści

1. Wstęp .....	2
2. Konfiguracja agenta komunikacyjnego .....	2
3. Statusy urządzeń .....	3
4. Wymagania.....	4
5. Komunikacja MQTT .....	5
6. Przykłady komunikatów .....	6
6.1. Przesłanie konfiguracji komunikacji.....	6
6.2. Zapytanie Modbus .....	6
6.3. Dane odczytowe .....	7
6.4. Dane odczytowe zagregowane .....	7

Fundusze  
Europejskie  
Program RegionalnyUnia Europejska  
Europejski Fundusz  
Rozwoju Regionalnego

## 1. Wstęp

Niniejszy dokument stanowi dokumentację techniczną protokołu komunikacyjnego AIS służącego do komunikacji z centrum monitorowania. Protokół umożliwia wykorzystanie centrum monitorowania do zbierania danych z zewnętrznych urządzeń (np. liczniki energii, liczniki mediów, stacje pogodowe, czujniki przemysłowe). Celem protokołu jest rozszerzenie zakresu działania centrum monitorowania poza sterownik AIS i zapewnienie możliwości jego integracji z innymi systemami.

## 2. Konfiguracja agenta komunikacyjnego

1. Konfiguracja ogólna
  - 1.1. Nazwa agenta (name text) - informacyjne
  - 1.2. Wersja konfiguracji (cfg\_ver text) – opcjonalny identyfikator wersji konfiguracji
2. Konfiguracja interfejsów (interfaces)
  - 2.1. Baud (baud\_rate): 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
  - 2.2. Parity (parity): 0-no parity, 1-odd parity, 2-even parity
  - 2.3. Stop bits (stop\_bits): 1-1bit, 2-2bits, 15-1.5bits
  - 2.4. Modbus TCP Gateway Enable (modbus\_gateway\_enable): true
  - 2.5. Modbus TCP Gateway Port (modbus\_gateway\_port): 502
3. Lista odczytywanych urządzeń (devices)
  - 3.1. Identyfikator (device\_id uuid)
  - 3.2. Nazwa (name text)
  - 3.3. Interfejs (opcjonalne, zgodnie z potrzebami urządzenia)
  - 3.4. Wyłącz urządzenie (disabled boolean) – urządzenie jest wyłączone z odczytów
  - 3.5. Adres Modbus (modbus\_address integer)
  - 3.6. Harmonogram odczytu (read\_schedule\_period): 100ms, 125ms, 200ms, 250ms, 333ms, 500ms, 1s, 2s, 3s, 4s, 5s, 6s, 10s, 12s, 15s, 20s, 30s, 1min, 2min, 3min, 4min, 5min, 6min, 10min, 12min, 15min, 20min, 30min, 1h
  - 3.7. Minimalny interwał wysyłania odczytów (min\_interval\_s) – wartość podawana w sekundach, domyślnie 59s
  - 3.8. Maksymalny interwał wysyłania odczytów (max\_interval\_s) – domyślnie 86400s
  - 3.9. Lista odczytów (reading)
    - 3.9.1. Numer funkcji (modbus\_func integer)
    - 3.9.2. Adres rejestru (modbus\_reg integer)
    - 3.9.3. Ilość rejestrów (modbus\_reg\_count integer)
  - 3.10. Lista parametrów (params)
    - 3.10.1. Kod parametru (code text)
    - 3.10.2. Jednostka (unit text)
    - 3.10.3. Rejestr Modbus (modbus\_reg): function/address/modbus\_type/factor
    - 3.10.4. Numer funkcji: 1,2,3,4
    - 3.10.5. Typy modbus:
      - 3.10.5.1. int16
      - 3.10.5.2. uint16
      - 3.10.5.3. int32
      - 3.10.5.4. uint32
      - 3.10.5.5. int64
      - 3.10.5.6. uint64
      - 3.10.5.7. float/float32/float32\_big/float32\_abcd
      - 3.10.5.8. float32

- 3.10.6. Typ Parametru (param\_type):
    - 3.10.6.1. 1 – wartości bieżące
    - 3.10.6.2. 2 – wartości nastaw
    - 3.10.6.3. 3 – wartości narastające
  - 3.10.7. Formuła (opcjonalna)
    - 3.10.7.1.  $((value * 12) - 232)$
    - 3.10.7.2.  $((value+3) / 34)$
    - 3.10.7.3.  $(param1\_code + 2.5 * param2\_code)$
  - 3.10.8. Minimalny interwał wysyłania (min\_interval\_s) – wartość podawana w sekundach, jeżeli nie podana to pobierana z urządzenia, parametr zabezpiecza przed zbyt dużą liczbą danych
  - 3.10.9. Maksymalny interwał wysyłania (max\_interval\_s) – wartość podawana w sekundach, jeżeli nie podana to pobierana z urządzenia, parametr zabezpiecza przed zbyt rzadkim wysyłaniem danych
  - 3.10.10. Deadzone (deadzone) – domyślnie brak, wartości ujemne traktujemy jak brak dedzone, 0 oznacza pomijanie tych samych wartości, wartości dodatnie to deadzone z dodatkowym tresholdem.
  - 3.10.11. Wysyłaj odczyty (send\_value)
  - 3.10.12. Wysyłaj agregaty (send\_agg)
  - 3.10.13. Okres czasu zbierania agregatów w minutach (agg\_size): 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60
  - 3.10.14. Wersja konfiguracji (cfg\_ver text) – opcjonalny identyfikator wersji konfiguracji
4. Lista harmonogramów (schedules)
- 4.1. Identyfikator (schedule\_id uuid)
  - 4.2. Typ (schedule\_type): event, state
  - 4.3. Reguły (rules):
    - 4.3.1. ts\_begin timestamp
    - 4.3.2. ts\_end timestamp
    - 4.3.3. dow[]
    - 4.3.4. dom[]
    - 4.3.5. doy[]
    - 4.3.6. week[]
    - 4.3.7. time\_begin time
    - 4.3.8. time\_end time
    - 4.3.9. value

### 3. Statusy urządzeń

Każde urządzenie ma swój status. Problemy podczas odczytów obniżają status urządzenia. Gdy odczyty zaczynają działać, to po pewnym czasie urządzenie podnosi swój status.

Szczegóły w poniższej tabeli.

Status	Running status	Flow up ↑↑	Flow down ↓↓	Retries per cycle
OK	1 Flawless		Dowolny błąd odczytu danych ⇒ Good	3
OK	2 Good	Brak błędów przez 10 odczytów ⇒ Flawless	3 cykle odczytu z błędem ⇒ Poor + notyfikacja	3
Unstable	3 Poor	Brak błędów przez 10 odczytów ⇒ Good + notyfikacja	3 cykle odczytu z błędem ⇒ Bad	2
Unstable	4 Bad	Wykonanie 5 odczytów ⇒ Poor	3 cykle odczytu z błędem ⇒ Down + notyfikacja	1

Down	5 Down	Dobry dowolny odczyt ⇒ Bad + notyfikacja		1
------	--------	---	--	---

## 4. Wymagania

- Konfiguracja może zawierać nadmiarowe (nieobsługiwane) pola. Nie jest to traktowane jako błąd konfiguracji. Konfiguracja musi być tylko prawidłowym json'em.
- W zakresie obsługiwanych pól konfiguracji nieprawidłowe wartości są sygnalizowane jako błąd w konfiguracji.
- Urządzenie w ciągły sposób odczytuje urządzenia. W przypadku błędów w odczycie Modbus RTU dany odczyt jest ponawiany (zgodnie z parametrami w tabeli statusów).
- Po każdym cyklu odczytów aktualizowany jest status urządzenia wg algorytmu w rozdziale 1.1. Zmiana statusu urządzenia kolejkuje wysłanie komunikatu zdarzenia o awarii komunikacji. W zasadzie są to zdarzenia zmiany statusu urządzenia.
- Dla każdego urządzenia przechowywane są statystyki pracy (ilość odczytów, ilość błędów). Statystyki można zdalnie wyresetować. Czas resetu statystyk jest dostępny do odczytu.
- Przez protokół MQTT urządzenie można zapytać o aktualny stan odczytów dla dowolnego urządzenia.
  - W konfiguracji możemy określić czy komunikacja jest w formacie JSON czy MessagePack.
  - W konfiguracji możemy określić format uuid: hex (standardowy 36 znakowy) lub base64 (22 znaki).
- Samo z siebie urządzenie kolejkuje wysyłkę komunikatów z aktualnymi odczytami (dla parametrów z send\_value = true oraz po odpowiednim spełnieniu warunków deadzone, min\_interval\_s, max\_interval\_s)
- Samo z siebie urządzenie kolejkuje wysyłkę komunikatów z agregatami (dla parametrów z send\_agg = true). Dla param\_type = 1 wartości to value\_count, value\_avg, value\_min, value\_max, value\_stddev. Dla param\_type = 3 wartości to begin\_value, delta\_value.
- Skolejkowane dane są wysyłane w zależności od rodzaju urządzenia (modemu komunikacyjnego). Po nawiązaniu połączenia wszystkie skolejkowane dane powinny zostać wysłane.
- Komunikaty zdarzeń mają zapamiętywany stan, aby nie „spamować” komunikatami o długotrwałych błędach. Ponadto w konfiguracji można określić min. czas pomiędzy zdarzeniami, aby zabezpieczyć się przed problemem flip/flop.
- Komunikacja MQTT jest zabezpieczona szyfrowaniem TLS min. w wersji 1.2.
- Urządzenie nasłuchuje na porcie modbus\_gateway\_port jako bramka Modbus TCP. Port zapewnia bezpośredni dostęp do urządzeń na magistrali i nie koliduje z bieżącymi odczytami przesyłanymi na MQTT.
- Urządzenie weryfikuje autentyczność serwera MQTT sprawdzając jego certyfikat. Jeżeli certyfikat nie jest na liście zaufanych certyfikatów to połączenie jest przerywane. Lista zaufanych certyfikatów powinna być min. 8 pozycyjna.
- Urządzenie może autoryzować się w serwerze MQTT przy pomocy użytkownika i hasła.
- Urządzenie może autoryzować się w serwerze MQTT przy pomocy klucza i certyfikatu urządzenia.
- Urządzenie przechowuje informacje, kiedy miało zsynchronizowany czas.
- Każdy odczyt jest znaczony jakością odczytu (quality):
  - 0 - OK cartain / pewna
  - 1 - T uncertain time / niepewny czas (np. RTC ustawiany ponad 36h temu)
  - 2 - D uncertain date / niepewna data (np. RTC ustawiany ponad 356 dni temu)
  - 4 - I interpolated / interpolowana
  - 8 - V uncertain value / niepewna wartość
  - 16 - W wrong value / zła wartość
  - Powyższe wartości dodajemy jako niezależne flagi.
- Adres serwera MQTT można określić jako listę adresów IP i nazw DNS. Nazwa DNS może się resolvować do kilku adresów IP. Wtedy taka nazwa zamieniana jest na kilka wpisów IP.
- Określamy politykę nawiązywania połączenia z serwerami MQTT:
  - lastsuccess – z ostatnim działającym, jeżeli nie działa to przepinamy się na kolejny.
  - failover – zawsze najpierw z pierwszym, potem z kolejnymi jak poprzedni nie odpowiada.

19.3. roundrobin – za każdym razem z kolejnym z listy.

## 5. Komunikacja MQTT

Komunikaty skierowane do urządzenia są wysyłane na topic:

```
{mqtt_topic_prefix}/{device_id}/in/+
```

Dodatkowo urządzenie nasłuchuje na topicu broadcastowym:

```
{mqtt_topic_prefix}00000000-00000000-00000000-00000000/in/+
```

Urządzenie wysyła dane na topic:

```
{mqtt_topic_prefix}/{device_id}/out/{message_type}
```

Przykłady komunikatów (ich treści) w formacie JSON znajdują się w załączniku ZIP. Dla niektórych rodzajów komunikatów (message\_type) jest kilka przykładów.

*in/sysconfig* – Konfiguracja systemowa urządzenia przesyłana do urządzenia.

Aktualizowane są tylko przesłane parametry. Pozostałe są na urządzeniu pozostawiane bez zmian. Często będzie tym kanałem przesyłany fragment konfiguracji, np. ograniczony do parametrów dotyczących komunikacji.

*in/appconfig* – Konfiguracja aplikacyjna przesyłana do urządzenia.

Aktualizowana jest konfiguracja w całości. Cała dotychczasowa zawartość jest wyrzucana i zastępowana nową.

*in/request-action* – Żądanie wykonania określonej akcji pochodzące z systemu i skierowane do urządzenia.

*send-status* wymusza bezzwłoczne wysłanie statusu urządzenia, czyli wysłania komunikatu *out/status*.

*read-device* wymusza bezzwłoczne odpytanie urządzenia i wysłania wszystkich bieżących wartości komunikatem *out/entry-value*.

*in/request-modbus* – Zapytanie adhoc na magistrali RS485. Urządzenie w odpowiedzi odeśle *out/response-modbus*.

*in/set-value* – Żądanie zmiany wartości dla parametrów nastaw (param\_value=2).

Po ustawieniu podanych parametrów dla zmodyfikowanych urządzeń generowany jest odczyt adhoc i wysyłana odpowiedź *out/entry-value*.

Funkcje zapisu dla param\_type=2 są dobierane na podstawie funkcji odczytu: jak 1 to 5 lub 15, jak 3 to 6 lub 16. Można ustawić wyjątek w *modbus\_write* (niektóre urządzenia mają inne rejestry do zapisu lub nie obsługują 15 i 16).

*out/status* – Informacje o pracy urządzenia. Komunikat wysyłany po starcie urządzenia, przed jego wyłączeniem oraz cyklicznie (zgodnie z konfiguracją).

*out/entry-value* – Dane z odczytami dla poszczególnych urządzeń. Wysyłane są tylko parametry oznaczone jako *send\_value = true* i po uwzględnieniu reguł *min\_interval\_d*, *max\_interval\_s*, *deadzone\_value*.

*out/entry-agg* – Dane zagregowane dla poszczególnych urządzeń. Wysyłane są tylko parametry oznaczone jako *send\_agg = true*.

*out/response-modbus* – Odpowiedź na komunikat *in/request-modbus*.

*out/request-action* – Żądanie wykonania określonej akcji pochodzące z urządzenia i skierowane do systemu.

*send-appconfig* pozwala urządzeniu zażądać przestania bieżącej konfiguracji.

*device\_no* – to indeks urządzenia liczony od 1 dla całej konfiguracji

*param\_no* – to indeks parametru liczony od 1 w ramach każdego urządzenia



### 6.3. Dane odczytowe

```
{
  "request_id": "to co w request, jak podano...",
  "cfg_ver": "to co w konfigu, jak podano...",
  "value_list": [
    ["2020-04-23T18:25:43", 0, 1, 1, 0, 12764.32, 23987, 21.4],
    ["2020-04-23T18:30:43", 0, 1, 1, 0, 12764.32, 23987, 21.8],
    ["2020-04-23T18:35:43", 0, 1, 1, 0, 12764.32, 23987, 22.4],
    ["2020-04-23T18:40:43", 0, 1, 1, 0, 12764.32, 23987, 23.4]
  ]
}
```

### 6.4. Dane odczytowe zagregowane

```
{
  "request_id": "to co w request, jak podano...",
  "cfg_ver": "to co w konfigu, jak podano...",
  "value_list": [
    ["2012-04-23T18:00:00", 15, 1, 1, 0, [12764.32, 123.2], [23987, 100.2], [5, 21.4, 21, 21.8, 0.7]],
    ["2012-04-23T18:15:00", 15, 1, 1, 0, [12864.52, 123.2], [24087.2, 100.2], [5, 21.8, 21.5, 22.2, 0.7]],
    ["2012-04-23T18:30:00", 15, 1, 1, 0, [12964.72, 123.2], [24187.4, 100.2], [5, 22.2, 22, 22.6, 0.7]],
    ["2012-04-23T18:45:00", 15, 1, 1, 0, [13064.92, 123.2], [24287.6, 100.2], [5, 23.4, 23, 24.9, 0.7]],
    ["2012-04-23T18:00:00", 15, 2, 1, 0, [12764.32, 123.2], [23987, 100.2], [5, 21.4, 21, 21.8, 0.7]],
    ["2012-04-23T18:15:00", 15, 2, 1, 0, [12864.52, 123.2], [24087.2, 100.2], [5, 21.8, 21.5, 22.2, 0.7]],
    ["2012-04-23T18:30:00", 15, 2, 1, 0, [12964.72, 123.2], [24187.4, 100.2], [5, 22.2, 22, 22.6, 0.7]],
    ["2012-04-23T18:45:00", 15, 2, 1, 0, [13064.92, 123.2], [24287.6, 100.2], [5, 23.4, 23, 24.9, 0.7]]
  ]
}
```

Koniec