

ISWiFi – firmware

Table of Contents

- Changelog.....3
- Dokumenty powiązane.....4
- Hardware.....5
- Funkcje.....5
 - Repeater Modbus RS485.....5
 - Testy.....5
 - Podgląd stanu ISM 0-10.....5
 - Serwer REST API.....6
 - Logger.....6
 - Log stały.....6
 - Log wydarzeń.....7
 - Log błędów.....7
 - Historia Wartości.....7
 - Kontroler temperatury.....8

Changelog

Rev	Data	Opis	Autor
1.0	05.03.17	Init draft	M.R.

Dokumenty powiązane

ID	Tytuł	Wersja	Data	Plik	Author
[1]	Moduł komunikacji bezprzewodowej ISWiFi	3	02-03-2017	ISWiFi_moduł.pdf	R.Smaga
[2]	ISWiFi Communication	1.01	04.03.17	Komunikacja.pdf	M.Radziwon
[3]	Parametry	1.01	5.03.17	Parametry.pdf	M.Radziwon

Hardware

Oprogramowanie wbudowane będzie działało na procesorze STM32F105 będącym częścią systemu opisanego w [1].

Funkcje

Repeater Modbus RS485

~~Urządzenie powinno przekierowywać zapytania modbus kierowane do ISM010 (adres zgodny z /ISM010_26¹) pojawiające się na porcie MODBUS SLAVE [1] na port ISM0-10V MODBUS MASTER, odczekać maksymalny interwał (/Properties/ModbusDelay) i zwrócić odpowiedź jeśli nadeszła, jeśli nie to zwrócić błąd:~~

~~0x0B Gateway Target Device Failed to Respond~~

~~i nie przysyłać z powrotem ramki odpowiedzi jeśli się pojawi~~

~~Jeśli ISWiFi jest w stanie komunikacji z ISM010, żądanie zewnętrzne powinno zostać zakolejkowane, jeśli minie czas zdefiniowany w Properties/ModbusDelay powinien zwrócić błąd:~~

~~0x06 Slave Device Busy~~

~~Urządzenie powinno obsługiwać standard protokołu modbus RTU oraz ASCII (/Modbus/Type) na porcie MODBUS SLAVE. Port port ISM0-10V MODBUS MASTER ustawiony na sztywno ASCII~~

~~Po każdej transakcji ustawić /ISM010_available na true lub false~~

~~Po nieudanej transakcji dodać błąd do logu błędów~~

Testy

- ~~• Prawidłowo działający repeater ASCII i RTU w adresacji [1-127].~~
- ~~• Przekazanie błędu 0x0B i zaniechanie przesłania spóźnionej ramki odpowiedzi~~
- ~~• Przekazanie błędu 0x06~~

Podgląd stanu ISM 0-10

Urządzenie powinno odczytać wszystkie parametry ISM010 w momencie podania zasilania oraz po wymuszeniu resetu (/ISM010_restart). Parametry 1-16 oraz 501-634

~~Urządzenie powinno wykorzystywać komunikację przechodzącą przez Repeater Modbus RS485 w celu uaktualniania parametrów (w przypadku zapisu nowych wartości do ISM010 należy uaktualnić wartość po otrzymaniu potwierdzenia z ISM010).~~

Urządzenie powinno cyklicznie (co interwał zdefiniowany w /Properties/QueryInterval) odpytywać ISM010 o wartości parametrów 2-16.

1 Adres ISM010 powinien być sprawdzony raz po włączeniu zasilania

Zmiana każdego parametru w stosunku do ostatnio zapamiętanej wartości powinna być sprawdzana wg parametru /Query/SensRange (jeśli zmiana wartości jest większa niż zadana wartość procentowa poprzedniej wielkości parametr jest zapisywany). Jeśli zapisywany parametr może wymagać wpisu do jednego lub więcej logów (log zdarzeń, historii wartości, histogramu wartości, lista błędów).

Po każdej pomyślnej transakcji ustawić /ISM010_available na true

Serwer REST API

Szczegóły komunikacji opisane są w dokumencie [2], lista parametrów w dokumencie [3].

Serwer powinien działać zarówno po WiFi jak i Ethernet.

Każde zapytanie o parametr ISM010 powinno skutkować wysłaniem zapytania MODBUS, a odpowiedź sprawdzona jak w Podgląd stanu ISM 0-10.

/ISM010_restart powinien wymusić wpis do logu wydarzeń i do logu stałego.

każda zmiana parametru powinna być wpisana do logu wydarzeń oraz powinna uaktualnić pole /Properties/LastChange oraz /Properties/Changes

Każde żądanie usunięcia pliku wymusza wpis do logu wydarzeń i logu stałego.

Logger

Logger powinien tworzyć pliki csv oddzielane znakami tabulacji z CRLF jako znakiem końca linii.

Każda linia powinna zawierać czas+datę tekstowo, czas unixowy, oraz string z opisem błędu.

Pliki loggera powinny mieć maksymalną wielkość wynikającą z rozmiaru karty, po którego przekroczeniu po dodaniu nowego wpisu, najstarszy powinien być kasowany. Moment przekroczenia maksymalnego rozmiaru powinien być odnotowany w logu stałym wraz z zapisaniem daty najstarszego wpisu.

Zapisywanie powinno być „bezpieczne” tzn odporne na zanik napięcia zasilania w chwili zapisu.

Rozważana jest opcja archiwizacji (kompresji) plików po przekroczeniu pewnej wartości i umieszczenie informacji o tym w logu.

Log stały

Plik niemożliwy do usunięcia przez API. Zapisywane do niego powinny być jedynie najważniejsze zdarzenia –:

- usuwanie/czyszczenie innych logów
- zmiana ustawień ISWiFi
- zmiana ustawień ISM010
- restartowanie poprzez API,
- kasowanie awarii

- zmiana ustawień czasu (/Time/Time)

Log wydarzeń

W logu wydarzeń powinny być odnotowane poniższe sytuacje:

- nawiązanie połączenia z aplikacji mobilnej bądź Ethernet (jeśli możliwe zapis adresu MAC łączącego się urządzenia)
- zaniki zasilania
- start systemu (pojawienie się zasilania)
- zmiana fazy z 00 na 01 (start systemu)
- zmiana fazy ISM010 (parametr 10)
- odczyt plików /Data/{filename} – moment dostępu oraz plik
- restart poprzez API
- usunięcie pliku poprzez API
- zapis parametru poprzez API
- wystąpienie błędu (kod błędu + numer kolejny błędu)

Log błędów

Log błędów powinien zawierać pojawianie się błędów w parametrze ISM010 13, oraz 9, jak również ich skasowanie (automatyczne dla 13 i wymagające kasowania awarii dla 9 - potrzebne rozróżnienie w opisie). Każdy pojawiający się błąd powinien mieć nadany numer kolejny dodawany do opisu.

Historia Wartości

Tu proponuję dwie opcje:

1. Jeden plik historii wartości.

Każda zmiana wartości sprawdzona jak w Podgląd stanu ISM 0-10 (/Query/SensRange) jest odnotowywana w jednym wspólnym pliku w postaci [czas unixowy],[numer parametr],[nowa wartość]. Przy przekroczeniu wielkości pliku, po dopisaniu nowego wpisu, wpis najstarszy jest usuwany.

2. Osobne pliki dla każdego parametru. Jeśli suma wielkości plików przekroczy ustaloną wartość, ostatni wpis z największego pliku jest usuwany po wpisaniu nowej wartości (możliwe, że do innego pliku)

Kontroler temperatury

W urządzeniu zakładane są dwa obwody sterowania temperaturą za pomocą algorytmu z histerezą. Jeśli $T < T_{0-H}$ wyjście sterujące przełączane jest na stan wysoki, jeśli $T > T_0$ – wyjście sterujące przełączane jest na stan niski.

Czujnikiem temperatury będzie układ KTY81