

MS360-LPM

wersja 1.09 (**wersja robocza**)



Dokumentacja użytkownika

Białystok 2011

Podstawy

Komunikacja z multiczujnikiem MS360-LPM dostępna jest za pomocą transmisji szeregowej EIA-485 (wcześniej RS-485) przy wykorzystaniu protokołu Modbus w trybie RTU. Urządzenie działa jako Slave. Format podstawowej ramki komunikacyjnej przedstawiony jest w tabeli 1. Prędkość komunikacji może być programowo zmieniana i może wynieść 1200bps, 2400bps, 4800bps, 9600bps lub 19200bps.

parametr	ustawienie
bit parzystości	brak
ilość bitów danych	8
ilość bitów stopu	1

Tabela 1: Format podstawowej ramki komunikacyjnej

Modbus

Multiczujnik MS360-LPM używa protokołu komunikacyjnego MODBUS w trybie RTU. Funkcje, na jakie może odpowiadać, to:



kod funkcji	opis
1	Odczyt wielu zmiennych typu coil
2	Odczyt wielu zmiennych typu discrete input
3	Odczyt wielu zmiennych typu holding register
4	Odczyt wielu zmiennych typu input register
5	Zapis pojedynczej zmiennej typu coil
6	Zapis pojedynczej zmiennej typu holding register
15	Zapis wielu zmiennych typu coil
16	Zapis wielu zmiennych typu holding register

Tabela 2: Funkcje dostępne na multiczujniku MS360-LPM

MS360-LPM wykorzystuje wszystkie bloki danych, jakie oferuje MODBUS, czyli coils, discrete inputs, holding registers i input registers. Organizacja bloków danych przedstawiona jest w tabeli 3.

adres (zakres 1)	adres (zakres 2)	zmienna	wartość domyślna
HOLDING REGISTERS (funkcja 3, 6, 16):			
0x00	0x100	tempMaxSendTime	0
0x01	0x101	tempMinSendTime	1
0x02	0x102	tempMinDelta	0
0x03	0x103	tempScale	0
0x04	0x104	humMaxSendTime	0
0x05	0x105	humMinSendTime	1
0x06	0x106	humMinDelta	0
0x07	0x107	humScale	0
0x08	0x108	prMaxSendTime	0
0x09	0x109	prMinSendTime	1
0x0A	0x10A	prMinDelta	0
0x0B	0x10B	prScale	0
0x0C	0x10C	debounceTime	0
0x0D	0x10D	actSoftStat	65280
0x0E	0x10E	usartSpeed	960
0x0F	0x10F	tempMultiTime	1
0xFF	-	slaveAddress	1
COILS (funkcja 1, 5, 15):			
0x00	0x110[0]	stateDO1	0
0x01	0x110[1]	stateDO2	0
0x02	0x110[2]	stateDO3	0
0x10	0x110[3]	configDO1	0
0x11	0x110[4]	configDO2	0
0x12	0x110[5]	configDO3	0
od 0x50 do 0x8F	od 0x111 do 0x114	Multi DS: termometr o indeksie i do wymiany zakres 1: i=adres-0x50 zakres2: 0x111[0]->indeks 0, 0x111[15]->indeks 15, 0x112[0]->indeks 16, 0x112[15]->indeks 31, itd.	
DISCRETE INPUTS (funkcja 2):			
0x00	0x110[6]	stateDI1	0
0x01	0x110[7]	stateDI2	0
0x02	0x110[8]	statePresence	0
0x10	0x110[9]	risingEdgeDI1	0

0x11	0x110[10]	risingEdgeDI2	0
0x20	0x110[11]	fallingEdgeDI1	0
0x21	0x110[12]	fallingEdgeDI2	0
0x30	0x110[13]	errorDIO1	0
0x31	0x110[14]	errorDIO1	0
0x32	0x110[15]	errorPresence	0
0x40	0x115[0]	Temp: brak odpowiedzi	
0x41	0x115[1]	Temp: przekroczone czas pomiaru	
0x42	0x115[2]	Temp: błąd sumy kontrolnej (CRC)	
0x43	0x115[3]	Wilg: brak odpowiedzi	
0x44	0x115[4]	Wilg: przekroczone czas pomiaru	
0x45	0x115[5]	Wilg: błąd sumy kontrolnej (CRC)	
0x46-0x4F	0x115[6-15]	Przeźródź wolna (np. dla flag dla czujnika ciśnienia)	
od 0x50 do 0x8F	od 0x116 do 0x119	Multi DS: brak odpowiedzi dla termometru który zapisany jest pod indeksem i, gdzie: zakres 1: i=adres-0x50 zakres2: 0x116[0]->indeks 0, 0x116[15]->indeks 15, 0x117[0]->indeks 16, 0x117[15]->indeks 31, itd.	
INPUT REGISTERS (funkcja 4):			
0x00	0x11A	tempReg	0
0x01	0x11B	humReg	0
0x02	0x11C	pressReg	0
0x03	0x11D	debounceReg	0
0x04	0x11E	avSensorsReg	0
0x05	0x11F	windSpeed	0
0x0D	0x120	bootVer	x
0x0E	0x121	softVer	x
0x10	0x122	allDataSize	27
0x11	0x123	dsCount	0
od 0x100 do 0x13F	od 0x124 do (0x124+dsCount-1)	tempDsReg	x
od 0x140 do 0x1FF	od (0x124+dsCount) do (0x124+dsCount*4-1)	dsRomCode	x

Tabela 3: Organizacja danych na MS360-LPM

Opis szczegółowy danych

___MaxSendTime

Do zmiennych tego typu zaliczamy tempMaxSendTime, humMaxSendTime oraz prMaxSendTime. Zmienne te określają maksymalny czas, po którym następuje aktualizacja i zatwierdzenie wartości z pomiaru z czujnika temperatury, wilgotności lub ciśnienia, po której w odpowiednim rejestrze zapisywana jest aktualna wartość.

Jeśli zmienna ta ma wartość 0, oznacza to, że jest ona nieaktywna. Wartość tej zmiennej odpowiada wielokrotności 0,1 sekundy.

Przykład: humMaxSendTime=20, tzn. że aktualizacja wilgotności nastąpi maksymalnie po 2s od poprzedniej aktualizacji.

___MinSendTime

Do zmiennych tego typu zaliczamy tempMinSendTime, humMinSendTime oraz prMinSendTime. Zmienne te określają minimalny czas, po jakim może nastąpić aktualizacja wartości pomiaru. Gdy minie ten czas, następuje pomiar, którego wynik porównywany jest z ostatnio zatwierzoną wartością pomiaru, uwzględniając przy tym zmienną ___MinDelta. W przypadku gdy wartość ___MinSendTime jest większa niż ___MaxSendTime, pomiar aktualizacja i zatwierdzenie pomiaru nastąpi po czasie wskazanym przez ___MaxSendTime.

Zmienna przyjmuje wartość większą od 0. Przy próbie zapisu wartości 0, zapisywane jest 1. Wartość tej zmiennej odpowiada wielokrotności 0,1 sekundy.

___MinDelta

Do zmiennych tego typu zaliczamy tempMinDelta, humMinDelta oraz prMinDelta. Zmienne te określają minimalną różnicę między poprzednio zatwierdzoną wartością pomiaru a obecnym pomiarem. Jeśli faktyczna różnica jest równa lub większa od zapisanej w zmiennej ___MinDelta i minął czas wskazany w ___MinSendTime, następuje aktualizacja i zatwierdzenie temperatury, wilgotności lub ciśnienia (zależnie dla której wielkości zaistniała taka sytuacja).

Wartość tej zmiennej jest wielokrotnością 0,1°C dla temperatury lub 0,1% dla wilgotności.

Przykład: tempMinDelta=35, tzn. że zmiana wartości, odczytywanej z rejestru tempReg, nastąpi wówczas, kiedy minie czas minimalny wskazany w tempMinSendTime i temperatura zmieni się o co najmniej 3,5°C. Wcześniejsza zmiana temperatury może nastąpić tylko na skutek upływu czasu maksymalnego wskazanego przez tempMaxSendTime.

___Scale

Do zmiennych tego typu zaliczamy tempScale, humScale oraz prScale. Zmienne te określają przesunięcie (korektę) dla wykonywanych pomiarów. Wartość jest wielokrotnością 0,1°C dla temperatury lub 0,1% dla wilgotności. Możliwa jest korekta ujemna, polegająca na odjęciu pewnej

wartości. W takim przypadku $___Scale=65536-x$, gdzie x jest wartością bezwzględną korekty.

Przykład: tempScale=50, tzn. korekta pomiaru wynosi +5° C. Jeśli faktyczna temperatura wynosi 27,5°C, to po uwzględnieniu korekty będzie wynosić 32,5°C.

Przykład: tempScale=65486, tzn. korekta pomiaru wynosi -5° C. Jeśli faktyczna temperatura wynosi 27,5°C, to po uwzględnieniu korekty będzie wynosić 22,5°C.

actSoftStat

Zmienna wykorzystywana jest przy aktualizacji oprogramowania.

usartSpeed

Zmienna określa prędkość komunikacji z jaką działa MS360-LPM, gdzie $usartSpeed=prędkość/10$. Wartości, jakie może przyjąć ta zmienna to 120, 240, 480, 960, 1920, co oznacza komunikację z prędkością kolejno 1200bps, 2400bps, 4800bps, 9600bps i 19200bps. Przy próbie zapisu innej wartości, jest ona pomijana. Przy zapisie poprawnej wartości innej niż dotychczas, po wysłaniu ewentualnej odpowiedzi, multiczujnik restartuje się i zaczyna komunikację z nową prędkością.

tempMultiTime

Zmienna określa odstęp czasu między kolejnymi aktualizacjami temperatury dla czujników podłączonych pod wspólną linię.

Zmienna przyjmuje wartość większą od 0. Przy próbie zapisu wartości 0, zapisywane jest 1. Wartość tej zmiennej odpowiada wielokrotności 0,1 sekundy.

slaveAddress

Zmienna ta określa adres własny multiczujnika MS360-LPM, wykorzystywany w protokole komunikacyjnym Modbus, który jest aktualnym adresem multiczujnika gdy na dipswitchu ustawiona jest wartość 255. Przy takim ustawieniu dipswitcha rejestr ten można odczytywać (funkcja 3) oraz zapisywać tylko funkcją 6. Gdy na dipswitchu

Dostęp do tego rejestru możliwy jest za pomocą funkcji 3 lub 6, tylko gdy na dipswitchu jest wartość 0 lub 255. Do rejestru można zapisać wartości ≤ 255 . Po zmianie wartości zmiennej następuje restart multiczujnika.

stateDO1, stateDO2

Zmienne określają poziom napięcia które jest wymuszane na DO1 lub DO2, w przypadku kiedy są one ustawione jako wyjście.

Wartość 0 oznacza niskie napięcie, wartość 1-wysokie.

configDO1, configDO2

Konfiguracja linii DO/DI.

Wartość 0 oznacza że linia pracuje jako wejście, wartość 1 oznacza pracę jako wejście i wyjście.

stateDI1, stateDI2

Zmienne określają poziom napięcia na DI1 lub DI2.

Wejścia są zwierne do masy, w związku z czym wartość 0 oznacza że na linii jest wysokie napięcie, wartość 1 oznacza niskie napięcie na linii.

statePresence

Stan czujnika obecności.

Wartość 0 oznacza brak ruchu, wartość 1 oznacza wykrycie ruchu.

risingEdgeDI1, risingEdgeDI

Pola te określają, czy wykryto narastające zbocze od czasu poprzedniego odczytu. Po odczytaniu tej wartości, pola te są zerowane.

Wartość 0 oznacza brak zbocza narastającego od poprzedniego odczytu, wartość 1 oznacza wykrycie zbocza narastającego.

fallingEdgeDI1, fallingEdgeDI2

Pola te określają, czy wykryto opadające zbocze od czasu poprzedniego odczytu. Po odczytaniu tej wartości, pola te są zerowane.

Wartość 0 oznacza brak zbocza opadającego od poprzedniego odczytu, wartość 1 oznacza wykrycie zbocza opadającego.

errorDIO1, errorDIO2

W przypadku konfiguracji linii jako wyjście (DO), testowane jest, czy żądany poziom napięcia jest zgodny z faktycznym. Jeśli nie jest, zgłaszany jest błąd poprzez zapis wartości 1 na polu errorDIO1 lub errorDIO2.

tempReg

Pomiar temperatury.

Dokładność 0,1°C.

humReg

Pomiar wilgotności.

Dokładność 0,1%.

pressReg

Pomiar ciśnienia.

debounceReg

Pomiar nasłonecznienia.

Wartość w luksach.

avSensorsReg

Zmienna jest zestawem flag, określających dostępność czujników. Bit o wartości 1 oznacza że czujnik jest dostępny, 0-czujnik niedostępny.

	0	...	0	PSN	FLO	LHT	WND	MUL	HUM	TMP	
bit	15	...	7	6	5	4	3	2	1	0	0x04 (0x115)

TMP - temperatura (główna)

HUM - wilgotność

MUL - temperatura (na wspólnej linii)

WND - prędkość wiatru

LHT - nasłonecznienie

FLO - czujnik zalania

PSN - czujnik obecności

windSpeed

Zmienna przedstawia prędkość, z jaką wykrywane są zbrocza opadające na linii DI1 i prezentuje je jako ilość spadków napięcia na 10s. Pomiar dokonywany jest między dwoma kolejnymi spadkami. Przy braku spadku w czasie 10s, następuje wyzerowanie wartości.

bootVer

Wersja programu bootloader.



softVer

Wersja programu głównego.

inteligentne systemy automatyki

allDataSize

Ilość możliwych do odczytania rejestrów przy adresacji z zakresu 2.

dsCount

Ilość czujników temperatury podłączonych do wspólnej linii.

tempDsReg

Tablica zawierająca pomiary temperatur z termometrów podłączonych do wspólnej linii.

dDsRomCode

Tablica zawierająca numery seryjne termometrów podłączonych do wspólnej linii. Każdy numer seryjny składa się z 3 rejestrów (6 bajtów). Identyfikacja termometru, którego dotyczy dana wartość odbywa się na podstawie nr indeksu w tablicy.

Przykład: Dostępnych jest 5 czujników na wspólnej linii. Termometr, którego

numer seryjny zapisany jest jako pierwszy (pierwsze 3 rejestry) odpowiada pierwszej temperaturze z tablicy tempDsReg. Termometr, którego numer seryjny zapisany jest jako drugi (kolejne 3 rejestry) odpowiada drugiej temperaturze z tablicy tempDsReg, itd.



Adresacja za pomocą zakresu 2

Prawie wszystkie zmienne (oprócz slaveAddress) dostępne są na dwóch różnych adresach. Adres z zakresu 1 (pierwsza kolumna z tabeli 3) odpowiada adresowi pod jakim dostępna jest zmienna dla funkcji odpowiedniej dla typu danej zmiennej, np. stateDO2 jest dostępna pod adresem 0x02 dla funkcji odpowiadających za zapis i odczyt zmiennych typu coil.

Kolumna druga w tabeli 3 przedstawia inną adresację, którą można wykorzystać pracując ze zmiennymi. Wszystkie dane traktowane są tak, jakby leżały w jednym, wspólnym bloku, dzięki czemu można je razem odczytać i zapisać. Odczyt możliwy jest za pomocą funkcji 3, a zapis za pomocą funkcji 16 (nie dotyczy zmiennej actSoftState). Dzięki takiemu rozwiązaniu możliwy jest odczyt i zapis wszystkich zmiennych za pomocą jednego komunikatu. W przypadku podłączenia większej ilości czujników temperatury, całkowity odczyt pamięci zabierze maksymalnie 3 komunikaty.

Zmienne typu coil i discrete input są zawarte w jednym rejestrze o adresie 0x110, gdzie każdej zmiennej przypisany jest kolejny bit. Tabela 3 opisuje w nawiasach kwadratowych numer bitu, odpowiadającego kolejnej zmiennej. Bity liczone są od prawej strony, czyli bit numer 0 odpowiada bitowi najmniej ważnemu (LSB).

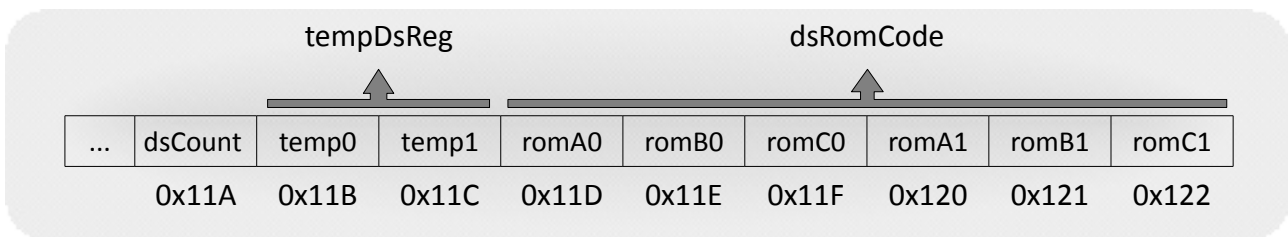
Przykład: Jeśli pod adresem 0x110 odczytaliśmy wartość 41498, czyli 0b1010001000011010 w systemie binarnym. Dla takiej wartości bit najmniej znaczący (LSB) ma wartość 0, a najbardziej znaczący (MSB) 1:

	1	0	1	0	0	0	1	0	0	0	0	1	1	0	1	0	
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0x110

Dla takich danych, zmienne, które przyjmują wartość 1 to stateDO2, configDO1, configDO2, risingEdgeDI1, errorDIO1, errorPresence.

Adresacja drugiego typu uwzględnia zmienną ilość danych, jaka wynika z możliwości podłączenia różnej ilości czujników temperatury. Pomiary z tych czujników zawarte są w tablicy tempDsReg, a w tablicy dsRomCode zapisane są ich numery seryjne. W sytuacji, gdy podłączono mniej niż maksymalna ilość czujników temperatury, czyli mniej niż 64 czujniki, adres pod którym rozpoczyna się tablica dsRomCode jest zmienny i zależy od ilości czujników. Tablica ta umieszczana jest bezpośrednio za pomiarami temperatury.

Przykład: Jeśli do multiczujnika podłączone są 2 dodatkowe czujniki temperatury, to przy wykorzystaniu drugiego zakresu adresacji odczytujemy 2 rejestry zawierające pomiary temperatury oraz leżące bezpośrednio za nimi 6 rejestrów, zawierających numery seryjne tych termometrów.



Wyjątki

Adresacja drugiego typu ma dwa wyjątki, wynikające ze specyfiki zmiennych. Dotyczą one zmiennej actSoftStat i slaveAddress. Zmienna actSoftStat może być tylko odczytana, przy zapisie wartość odpowiadająca tej zmiennej jest ignorowana. Zmienna slaveAddress jest z kolei w ogóle niedostępna. Pełna kontrola nad tymi zmiennymi jest wyłącznie za pomocą adresacji pierwszego typu, i dodatkowo przy zapisie należy wykorzystać funkcję zapisu jednego rejestru (funkcja 6).



przerwa (t_s) po której znowu dokonywany jest pomiar. Jeśli czas maksymalny (t_{max}) skończy się, to po zakończonym pomiarze aktualizowana jest wartość w rejestrze i moment ten (T_1) jest traktowany jako początek dla kolejnej ramki pomiaru.

W przypadku gdy czas maksymalny jest nieaktywny, aktualizacja pomiaru dokona się dopiero wówczas, gdy wynik pomiaru będzie większy (lub mniejszy) o co najmniej wartość $\min\Delta$ (T_2). Aktualizacja wartości pomiaru ma również miejsce w sytuacji, kiedy aktywowany jest maksymalny czas pomiaru ale przed jego upływem został osiągnięty lub przekroczony przedział wahań wyznaczony przez Δd .



Dane dodatkowe

	czas trwania	uwagi
t_{\min}	0,1s - 6553,5s	zależne od użytkownika
t_{\max}	0,1s - 6553,5s	zależne od użytkownika, może być nieaktywne (nie jest wówczas liczone)
t_{conv}	20ms	wilgotność
	80ms	temperatura SHT
	400ms	temperatura DS
t_s	100ms	

Tabela 4: Wartość odstępów czasowych

	rozdzielczość	jednostka
temperatura DS	0,125	°C
temperatura SHT	0,04	°C
wilgotność	0,4	%

Tabela 5: Rozdzielczość pomiarów

Spis treści

Podstawy.....	2
Modbus.....	2
Opis szczegółowy danych.....	5
___MaxSendTime.....	5
___MinSendTime.....	5
___MinDelta.....	5
___Scale.....	6
actSoftStat.....	6
usartSpeed.....	6
tempMultiTime.....	6
slaveAddress.....	6
stateDO1, stateDO2.....	6
configDO1, configDO2.....	7
stateDI1, stateDI2.....	7
statePresence.....	7
risingEdgeDI1, risingEdgeDI.....	7
fallingEdgeDI1, fallingEdgeDI2.....	7
errorDIO1, errorDIO2.....	7
tempReg.....	7
humReg.....	7
pressReg.....	8
debounceReg.....	8
avSensorsReg.....	8
windSpeed.....	8
bootVer.....	8
softVer.....	8
allDataSize.....	8
dsCount.....	8
tempDsReg.....	9
dDsRomCode.....	9
Adresacja za pomocą zakresu 2.....	10
Wyjątki.....	11
Zależności czasowe.....	12
Dane dodatkowe	14