Ovladač systému REXYGEN pro 1-Wire (modul OwsDrv)

Uživatelská příručka

REX Controls s.r.o.

Verze 3.0.4 Plzeň 27.3.2025

Obsah

1	Ovladač OwsDrv a systém REXYGEN	2							
	1.1 Úvod	2							
	1.2 Požadavky na systém	2							
	1.3 Instalace ovladače na vývojovém počítači	2							
	1.4 Instalace ovladače na cílovém počítači	3							
	1.4.1 Zprovoznění 1Wire serveru	3							
2	Zařazení ovladače do projektu aplikace	5							
	2.1 Přidání ovladače OwsDrv do projektu	5							
	2.2 Připojení vstupů a výstupů do řídicího algoritmu	6							
3	Konfigurace ovladače	9							
	3.1 Konfigurační dialogové okno	9							
	3.2 Využití alarmů programu owfs	11							
	3.3 Speciální signály	14							
4	4 Co dělat při problémech								
	Literatura	18							

Ovladač OwsDrv a systém REXYGEN

Úvod 1.1

V této příručce je popsáno používání ovladače OwsDrv pro výměnu dat s 1-Wire [1] čidly a zařízeními ze systému REXYGEN. Ovladač OwsDrv využívá programový balík OWFS [2], zejména jeho část owserver.

Pomocí tohoto ovladače je možné komunikovat se všemi zařízeními, která jsou v OWFS podporována.

1.2Požadavky na systém

Aby bylo možno ovladač využívat, musí být na vývojovém (konfiguračním) počítači a na cílovém zařízení (počítači) nainstalováno programové vybavení:

Vývojový počítač				
Operační systém	jeden ze systémů: Windows 10/11, GNU/Linux			
Vývojové nástroje	verze REXYGEN pro daný operační systém			
Cílové zařízení				
$\check{\mathrm{R}}$ ídicí systém REXY-	verze pro GNU/Linux			
GEN				
1-Wire ovladač	verze pro GNU/Linux			
OWFS	verze pro GNU/Linux			

Instalace ovladače na vývojovém počítači 1.3

Ovladač OwsDrv se instaluje jako balíček řídicího systému REXYGEN. Je obsažen v instalátoru vývojových nástrojů systému REXYGEN, pro jeho nainstalování je pouze nutné ho v instalačním programu systému REXYGEN zaškrtnout. Po typické instalaci se řídicí systém REXYGEN nainstaluje do cílového adresáře

C:\Program Files\REX Controls\REX_<version>, kde <version> označuje verzi systému REXYGEN.

Po úspěšné instalaci se do cílového adresáře zkopírují soubory:

bin\OwsDrv_H.dll - Konfigurační část ovladače OwsDrv.

DOC\CZECH\OwsDrv_CZ.pdf - Tato uživatelská příručka.

1.4 Instalace ovladače na cílovém počítači

Pokud ještě nemáte na cílovém zařízení (např. Raspberry Pi) nainstalovaný runtime modul **RexCore** řídicího systému **REXYGEN**, nainstalujte jej nejdříve podle příručky Začínáme se systémem **REXYGEN** pro příslušnou platformu [3].

Pro zpřístupnění dat z 1-Wire zařízení v systému REXYGEN a komunikaci s nimi je potřeba nainstalovat jednak moduly **owserver** a **ow-shell** systému OWFS a 1-Wire ovladač systému REXYGEN, což provedeme z příkazové řádky pomocí příkazu: Debian:

sudo apt-get install owserver ow-shell rex-owsdrvt

1.4.1 Zprovoznění 1Wire serveru

Modul owserver musí být nejdříve nakonfigurován dle typu použitého 1-Wire komunikačního rozhraní. Například pro I2C zařízení založené na čipu DS2482-100 nebo DS2482-800 by měl soubor /etc/owfs.conf vypadat následovně:

```
!server: server = localhost:4304
allow_other
server: port = localhost:4304
server: i2c = ALL:ALL
timeout_volatile = 2
```

Poznámka: soubor můžete editovat pomocí příkazu sudo nano /etc/owfs.conf. Při použití USB převodníku (např. DS9490R) použijte:

```
!server: server = localhost:4304
allow_other
server: port = localhost:4304
server: usb = all
timeout_volatile = 2
```

Při použití Raspberry Pi GPIO pinů použijte:

```
!server: server = localhost:4304
allow_other
server: port = localhost:4304
server: w1
timeout_volatile = 2
```

Je také nutné povolit I2C a /ow komunikace (sudo raspi-config -> Zapnout I2C a 1-Wire).

Restartujte owserver a vypište detekovaná 1-Wire zařízení pomocí příkazu owdir. Výstup může vypadat zhruba takto:

/28.551DDF030000
/bus.1
/bus.0
/uncached
/settings
/system
/statistics
/structure
/simultaneous
/alarm

První řádek je ID 1-Wire zařízení (v tomto případě teplotní čidlo DS18B20). Přečtěte teplotu pomocí příkazu :

owread /28.551DDF030000/temperature12

(ID změňte tak, aby odpovídalo vašemu čidlu).

Zařazení ovladače do projektu aplikace

Zařazení ovladače do projektu aplikace spočívá v přidání ovladače do hlavního souboru projektu a z připojení vstupů a výstupů ovladače v řídicích algoritmech.

2.1 Přidání ovladače OwsDrv do projektu

Přidání ovladače **OwsDrv** do hlavního souboru projektu je znázorněno na obr. 2.1. Pro zařazení ovladače do projektu slouží dva bloky. Nejprve je na výstup **Modules** bloku exekutivy **EXEC** připojen blok typu **MODULE** s názvem **OwsDrv**, který nemá žádné další parametry.

Druhý blok OWS typu IODRV, připojený na výstup exekutivy Drivers má tři nejdůležitější parametry:

- module Jméno modulu, ke kterému se ovladač váže, v tomto případě OwsDrv. POZOR, jméno rozlišuje velká a malá písmena!
- classname Jméno třídy ovladače, které je pro tento ovladač OwsDrv. POZOR, jméno rozlišuje velká a malá písmena!
- cfgname Jméno konfiguračního souboru ovladače (*.rio, REXYGEN Input/Output). Jedná se o textový soubor, který se v případě potřeby vytvoří při prvním spuštění konfiguračního dialogu. Pojmenovat jej můžete libovolně (zde ow_cfg.rio). Pro další informace viz kapitolu 3.

Jménem tohoto bloku, na obr. 2.1 zadaným jako OWS, začínají názvy všech vstupních a výstupních signálů poskytovaných tímto ovladačem.

Právě popsané parametry bloku IODRV se konfigurují v programu REXYGEN Studio v dialogovém okně, které je rovněž ukázáno na obrázku 2.1.

2.2 Připojení vstupů a výstupů do řídicího algoritmu

Vstupy a výstupy z ovladačů se připojují do souborů s příponou .mdl jednotlivých úloh. V hlavním souboru projektu jsou soubory úloh uvedeny pouze odkazem v blocích typu QTASK nebo TASK připojovaných na výstupy QTask, Level0,..., Level3 exekutivy. Pro připojení vstupů a výstupů z ovladače OwsDrv do řídicího systému REXYGEN lze použít bloky, znázorněné na obr. 2.2.

Blok typu From sloužící pro připojení jednoho vstupu má parametr Goto tag roven OWS__temperature, blok typu Goto používaný pro připojení jednoho výstupu by měl hodnotu parametru Goto tag rovnu OWS__jmeno. Všechny signály mají přímo na začátku svého jména prefix OWS následovaný dvěma znaky _ (podtržítko). Prefix (jméno bloku IODRV) může být libovolný, ale doporučuje se používat jméno modulu nebo třídy ovladače.

W 0	w ow_exec.mdl - REXYGEN Studio - [ow_exec]														
<u>F</u> ile	<u>F</u> ile <u>E</u> dit <u>V</u> iew Project <u>T</u> arget Tools <u>S</u> ettings <u>W</u> indow <u>H</u> elp														
	d E		d b) 🛍 ⊃	C .	R	U			b	. B. B. C	5 ≣	NK NE		
P	ow_exe	c			• 8		***	r BI	ock pro	perties					
						1		Plac							
									k nomoj					Plack type	
	Modu	les>							k name;					everlib\TC	
	Driv	ers 🕂	prev nex	t prev	/ next>			Bloc	s k type de	escription:				exectio µo	
			OWS	R	PI			RE)	(YGEN in	put/output d	river				
	Archiv	/es>													69-1
	QT	ask >						Ope		ocumentation	<u>n</u>				l oggle quick r
				_				Para	ameters	Options	Style				
	Leve	2IO -	prev nex ow task	tÞ				-50	alar nar	ameters					
	Leve	112									1				
									Pi	arameter	Value		Ту	pe	Descri
	Leve	12>							modu	le	OwsDrv	4	string		Module name
	Leve	13>						4	classr	ame	OwsDrv		string		I/O driver class nar
									s crgna	me	ow_crg.rio		string		Configuration file
	EXE							-	t lactor		10240		long		Stack size [butes]
Ľ								-	i nri		2		long		Driver thread logic
									/ timer		J □ off		bool		Driver is a source o
								-	unici				5001		Direction
								mod	ule: Mod	lule name					
							500	<u>کې</u>	onfigu	e					(
						l									
Proje	a :									la	arget: Not con	nected	1		

Obrázek 2.1: Příklad zařazení ovladače OwsDrv do projektu aplikace



Obrázek 2.2: Měření a záznam teplot, čtení logických v
stupů a nastavování logických výstupů pomocí ovladače
 $\tt OwsDrv$

Konfigurace ovladače

V této kapitole je popsána konfigurace jednotlivých vstupních a výstupních signálů a jejich symbolické pojmenování. Signály jsou namapovány na jednotlivé proměnné OWFS serveru.

3.1 Konfigurační dialogové okno

Konfigurační dialogové okno znázorněné na obr. 3.1 je obsaženo v souboru OwsDrv_H.dll a aktivuje se v programu REXYGEN Studiostisknutím tlačítka Configure v parametrickém dialogu bloku typu IODRV s parametry ovladače OwsDrv (viz kap. 2).

V horní části dialogu se definuje napojení na owserver. Program owserver typicky běží na stejném stroji jako RexCore, ale není to podmínkou.

Ve spodní části okna jsou definovány jednotlivé signály, které mohou být následně využity ke čtení nebo k zápisu v řídicím algoritmu systému REXYGEN. Jednoduše přidejte signály, použijte device ID, které vypsal příkaz owdir.

Signály lze přidávat nebo editovat po dvojkliku na dané položce přímo v parametrickém dialogu na obr. 3.1 nebo po stisku tlačítka Add nebo Edit v malém dialogu znázorněném na obr. 3.2.

Pokud je daný signál výstupem (ve sloupci I/O je vybrána hodnota output), je po spuštění systému tento výstup jednorázově nastaven na hodnotu Value, pokud tato hodnota není přepsána z řídicího algoritmu.

Při běhu řídicího systému se pro jednotlivé výstupní signály, označené v konfiguračním dialogu ve sloupci I/O jako výstupy (Output), cyklicky prochází tabulka signálů v pořadí uvedeném v tomto dialogu a pokud se signál od posledního zápisu změnil, zapíše se jeho nová hodnota. Obdobně, jednotlivé vstupní signály, označené v konfiguračním dialogu ve sloupci I/O jako vstupy (Input), se cyklicky čtou v témže pořadí. Při velkém množství zkonfigurovaných vstupů může přečtení celé tabulky trvat dost dlouhou dobu. Proto program owserver umožňuje indikovat změny signálů jako tzv. alarmy, v adresáři /alarm, viz sekci 3.2. Tento ovladač umožňuje s alarmy pracovat od verze 2.50 systému REXYGEN.

***	OwsDrv configura	tion										
Та	rget computer name/	r name/IP address: localhost:4304										
Ma	ximum number of cor	nsecutively served alarms:	5	5								
Ma	ximum number of ret	ries after item error:	3	3								
Tin	neout for pending ope	erations (in seconds):	20	20								
	REX I/O Name	Sensor/Actuato	or Path	Туре	I/O	Init. Value	Readable	Writab				
1	room1	/28.551DDF030000/tem	perature12	double	input	22	🗹 true	🗌 false				
2	room2	/28.063BC8040000/tem	perature12	double	input	22	🗹 true	🗌 false				
3	room3	/28.B8BDBC020000/ten	nperature12	double	input	22	🗹 true	🗌 false				
4	outside	perature12	double	input	10	🗹 true	🗌 false					

Obrázek 3.1: Konfigurační dialog ovladače 1-Wire

Je-li navíc označena volba Uncached, bude daný signál vždy čten z připojeného obvodu (např. z teploměru), není-li tato volba vybrána, bude vracen z vyrovnávací paměti (cache) owserveru, která se typicky aktualizuje jednou za 15 vteřin. Upozornění: Čím více signálů má zaškrtnutou volbu Uncached, tím pomalejší bude odezva tohoto ovladače.

Pro optimalizaci výkonnosti tohoto ovladače je dobré vědět, jak ovladač interně funguje. Hlavní smyčka tohoto ovladače spouštěná každou periodu ovladače vždy zpracovává nejvýše jeden požadavek na program **owserver** a po vyslání požadavku na **owserver**, nečeká na jeho okamžitou odpověď (tj. pokud data odpovědi nejsou k dispozici, snaží se je získat při dalším spuštění smyčky). Po inicializaci ovladače při spuštění exekutivy reálného času pracuje hlavní smyčka následovně:

- Zkontroluje, zda byl dokončen právě rozpracovaný požadavek (z předchozího volání této smyčky).
- Pokud ano, začne zpracovávat alarmy (detaily viz následující sekci).
- Pokud není žádný alarm zpracováván, snaží se zapsat jednu hodnotu výstupu z algoritmu.
- Pokud není zpracováván zápis hodnoty, snaží se přečíst jednu hodnotu vstupu do algoritmu.

₩ Edit Item		?	×
REX I/O Name:	outside		
Sensor/Actuator path:	/28.3F11C8040000/temperatur	re12	
Type:	double		•
I/O:	input		•
Init. Value:	10		
Readable: Writable:			
Uncached: Alarm:			
Latch Alarm:			
Power-on Reset Alarm:			
Set Alarm:			
Reset Latch:			
	ОК	Cancel	

Obrázek 3.2: Konfigurační jednoho signálu ovladače 1-Wire

Z uvedeného postupu je patrné, že nejvyšší důležitost (prioritu) má zpracování alarmů, potom zápis výstupních hodnot z algoritmu a naposledy čtení signálů. Při častém výskytu alarmů (což nemusí být normální stav) by se mohlo stát, že se nedostanou na řadu zápisy výstupů z algoritmu ani čtení ostatních hodnot. Proto lze v konfiguraci ovladače na obr. 3.1 nastavit nejvyšší počet po sobě obsloužených alarmů (Maximum number of consecutively served alarms), po němž se provede první z ostatních čekajících úkolů (zápis nebo čtení položky).

3.2 Využití alarmů programu owfs

Práce s alarmy patří mezi pokročilé techniky a vyžaduje dobrou znalost **owfs** a programu **owserver**. Alarmy doporučujeme použít teprve tehdy, když je odezva ovladače příliš dlouhá (pomalá).

Konfigurace jednoho alarmu je pro případ 1-Wire zařízení založeného na obvodu DS2408 patrná z obr. 3.3. Cesta k signálu (Sensor/Actuator path) se zadává bez počá-

Edit Item	? ×
REX I/O Name:	extDI_06
Sensor/Actuator path:	/29.066418000000/sensed.BYTE
Type:	byte 🔹
I/O:	input 🔹
Init. Value:	0
Readable:	
Writable:	
Uncached:	
Alarm:	
Latch Alarm:	latch.BYTE
Power-on Reset Alarm:	por
Set Alarm:	set_alarm=133333333
Reset Latch:	latch.BYTE=0
	OK Cancel

Obrázek 3.3: Příklad vyplnění konfiguračního dialogu pro alarm

teční složky **/alarm**. Po volbě **Alarm** se zadávají další řetězce. Před spuštěním ovladače se zadané hodnoty předzpracují a uloží do pracovních proměnných pro každý alarm:

- sPath cesta k zařízení, zde: /29.066418000000. Pro čtení nebo zápis hodnot se před tuto cestu může přidat adresář /alarm a za ni znak / a hodnota některého z řetězců uvedených v následujících položkách
- sSensed soubor se čtenou hodnotou, zde: sensed.BYTE
- sLatch soubor s příznakem změny, zde: latch.BYTE
- sAlarmPor soubor indikující zapnutí napájení daného zařízení, zde: por
- sSet soubor, do kterého se má zapsat hodnota, určující, kdy se bude alarm generovat (první část položky Set Alarm až ke znaku =), zde: set_alarm
- sSetVal hodnota, která se má zapsat do souboru sSet (druhá část položky Set Alarm za znakem =), zde: 133333333
- sLatchRes soubor, do kterého se má zapsat hodnota, určující, že byl alarm obsloužen (první část položky Reset Latch až ke znaku =), zde: latch.BYTE
- sLatchResVal hodnota, která se má zapsat do souboru sLatchRes (druhá část položky Reset Latch za znakem =), zde: 0

Pro zpracování alarmů je v ovladači **OwsDrv** použit stavový automat s následujícími stavy:

- NOT_USED V konfiguraci ovladače není uveden žádný alarm.
- INIT Počáteční stav automatu.
- ALARM_DIR Zjišťování obsahu adresáře /alarm.
- ALARM_PROCESS Začátek zpracování každého alarmu.
- ALARM_POR_READ Zjišťování zda dané zařízení neprovedlo svou inicializaci po zapnutí napájení (power-on reset) pomocí čtení souboru, jehož jméno je uloženo v řetězci sAlarmPor.
- ALARM_POR_READ_WAIT Čekání na dokončení čtení zahájeného ve stavu ALARM_POR_READ.
- ALARM_SET Nastavení generování alarmu na daném zařízení po zapnutí napájení. Do souboru, jehož jméno je určeno řetězcem sSet se zapíše hodnota řetězce sSetVal.
- ALARM_SET_WAIT Čekání na dokončení zápisu zahájeného ve stavu ALARM_SET. Poté se začnou procházet všechny výstupy (Output). Pokud se najde výstup, jehož cesta začíná řetězcem sPath, zapíše se do příslušného souboru hodnota specifikovaná v položce Init. Value (viz obr. 3.3).
- ALARM_INIT_WRITE_WAIT Čekání na dokončení každého jednotlivého zápisu počáteční hodnoty z předchozího stavu.
- ALARM_POR_RESET Smazání příznaku inicializace po zapnutí napájení (power-on reset). Do souboru, jehož jméno je určeno řetězcem sAlarmPor, se zapíše hodnota 0.
- ALARM_POR_RESET_WAIT Čekání na dokončení smazání příznaku inicializace po zapnutí napájení.
- ALARM_LATCH Zjištění, zda dané zařízení indikuje výskyt alarmu. V tomto stavu se pošle příkaz na čtení obsahu souboru, jehož jméno je určeno řetězcem sLatch. Pokud je obsah nenulový nebo se v seznamu položek vyskytuje alespoň jedna nenulová, je zdetekován výskyt alarmu od posledního čtení.
- ALARM_LATCH_WAIT Čekání na dokončení čtení zahájeného ve stavu ALARM_LATCH.
- ALARM_SENSED Čtení hodnoty signálu po výskytu alarmu. Pokud byl zdetekován výskyt alarmu ve stavu ALARM_LATCH, zahájí se čtení souboru v adresáři /alarm, jehož jméno je určeno řetězcem sSensed.
- ALARM_SENSED_WAIT Čekání na dokončení čtení zahájeného ve stavu ALARM_SENSED.
- ALARM_LATCH_RESET Smazání příznaku výskytu alarmu. V tomto stavu se do souboru ve složce /alarm, jehož jméno je určeno řetězcem sLatchRes zapíše hodnota uložená v řetězci sLatchResVal.

- ALARM_LATCH_RESET_WAIT Čekání na dokončení smazání příznaku výskytu alarmu zahájeného ve stavu ALARM_LATCH_RESET.
- SENSED Čtení hodnoty signálu, který se mohl změnit ještě před smazáním příznaku výskytu alarmu ve stavu ALARM_LATCH_RESET. V tomto stavu se zahájí čtení obsahu souboru, jehož jméno je určeno řetězcem sSensed.
- SENSED_WAIT Čekání na dokončení čtení zahájeného ve stavu SENSED.
- ALARM_BYPASS Stav umožňující mezi obsluhou alarmů provést jeden zápis nebo čtení jiného signálu.

Přechody mezi jednotlivými stavy se řídí tabulkou 3.1. V prvním sloupci je uveden aktuální stav, ve druhém sloupci může být pro každý aktuální stav uvedena jedna nebo několik podmínek, ve třetím sloupci pak je uveden stav, do kterého automat přejde, pokud je splněna příslušná podmínka z druhého sloupce. Pro daný aktuální stav jsou podmínky vyhodnocovány shora dolů.

3.3 Speciální signály

V některých speciálních případech se ukazuje jako užitečné/nutné mít přístup k stavovým nebo konfiguračním proměnným driveru. Níže popsané signály označené písmenem R (W) jsou určeny pro čtení (zápis), tj. jedná se o vstupy (výstupy) řídicího systému.

Vlastní ovladač má tyto speciální signály:

_DGNRESET W reset (smazání) akumulovaných diagnostických informací

_TRANSACTIONS R celkový počet transakcí s programem owserver

_RECONNECTS R počet opakovaných navázání spojení (po chybě komunikace)

Všechny globální signály začínají znakem _ (podtržítko). Vzhledem k oddělení označení ovladače od názvu signálu pomocí dvou znaků _, budou se v tomto případě vyskytovat za sebou tři znaky _, např. OWS___DGNRESET.

Dále ke každému signálu lze přidat za jméno speciální text, který značí, že se nepracuje s vlastní hodnotou, ale s jejím atributem. Texty jsou následující (všechny začínají znakem _):

_Value	RW	vlastní hodnota signálu (tj. stejná, jako název bez speciální přípony)
_DGNRESET	W	reset diagnostických informací pro daný signál
_TRANSACTIONS	R	počet transakcí s programem owserver pro daný signál
_ReadEnable	RW	povolení čtení signálu; ekvivalent: _RE
_WriteEnable	RW	povolení zápisu signálu; ekvivalent: _WE
_WriteOneShot	W	jednorázový zápis signálu; ekvivalent: _WOS
_Alarm	R	příznak vzniku alarmu na daném signálu; přečtením se smaže
_PerFactor	R	násobek periody ovladače pro aktualizaci signálu
_PerCount	R	počet period ovladače od poslední aktualizace signálu
_PerMax	R	maximální počet period od aktualizace signálu
_PendCount	R	aktuální počet cyklů po které se čeká na vrácení hodnoty z
		programu owserver
_PendLast	R	poslední počet cyklů po které se čekalo na vrácení hodnoty
		z programu owserver
_PendMax	R	maximální počet cyklů po které se čekalo na vrácení hodnoty
		z programu owserver
_Period	R	perioda aktualizace signálu v sekundách
_Age	R	doba uplynulá od poslední aktualizace signálu (stáří signálu)
_AgeMax	R	nejvyšší stáří signálu od posledního resetu diagnostických
		informací

Č.	Stav	Podmínky přechodu	Nový stav
-1	NOT_USED	Při otvírání ovladače nalezen alespoň jeden alarm	INIT
0	INIT	Zahájení čtení adresáře /alarm	ALARM_DIR
1	ALARM_DIR	Při čtení adresáře /alarm se dojde na jeho konec	ALARM_PROCESS
		Je-li již souvisle přečteno nMaxConsAlarms, pak	ALARM_BYPASS
2	ALARM_PROCESS	Je-li souvisle přečteno méně než nMaxConsAlarms, pak	ALARM_POR_READ
		Na konci cyklu alarmů se přiřadí iAlarmPos = -1. Pak	ALARM_BYPASS
		Není-li sAlarmPor definován, pak další alarm	ALARM_PROCESS
3	ALARM_POR_READ	Je-li sAlarmPor prázdný, pak	ALARM_LATCH
		Po úspěšném přečtení sAlarmPor	ALARM_POR_READ_WAIT
	ALADM DOD DEAD WATT	Je-li proměnná por různá od nuly	ALARM_SET
1 ·+	ALANM_FOR_READ_WAIT	Není-li proměnná por různá od nuly	ALARM_LATCH
		Není-li sSet nebo sSetVal definován, pak další alarm	ALARM_PROCESS
5	ALARM_SET	Je-li sSet nebo sSetVal prázdný, pak	ALARM_POR_RESET
		iAlarmInitPos = -1; Po úspěšném zápisu pak	ALARM_SET_WAIT
6	ALADM CET WATT	Procházení iAlarmInitPos. Pro nalezené zápisy pak	ALARM_INIT_WRITE_WAIT
0	ALARM_SEI_WAII	Na konci iAlarmInitPos = -1; pak	ALARM_POR_RESET
7	ALADM THIT UDITE UATT	Pokud je iAlarmInitPos < 0, pak	ALARM_POR_RESET
'	ALARM_INII_WRIIE_WAII	Jinak	ALARM_SET_WAIT
	ALARM_POR_RESET	Není-li sAlarmPor definován, pak další alarm	ALARM_PROCESS
8		Je-li sAlarmPor prázdný, pak	ALARM_LATCH
		Po úspěšném zápisu	ALARM_POR_RESET_WAIT
9	ALARM_POR_RESET_WAIT	Po dokončení požadavku	ALARM_LATCH
10	ALARM LATCH	Není-li sLatch definován nebo je prázdný, pak další alarm	ALARM_PROCESS
10	KERIM_ERION	Po úspěšném přečtení	ALARM_LATCH_WAIT
11	ALARM LATCH WATT	Je-li proměnná latch různá od nuly, pak	ALARM_SENSED
	KERRITER TOIL #KIT	Jinak další alarm	ALARM_PROCESS
		Není-li sSensed definován, pak další alarm	ALARM_PROCESS
12	ALARM_SENSED	Je-li sSensed prázdný, pak	ALARM_LATCH_RESET
		Po úspěšném přečtení	ALARM_SENSED_WAIT
13	ALARM_SENSED_WAIT	Po dokončení požadavku	ALARM_LATCH_RESET
		Není-li sLatchRes nebo sLatchResVal definován, pak	ALARM_PROCESS
14	ALARM_LATCH_RESET	Je-li sLatchRes nebo sLatchResVal prázdný, pak	SENSED
		Po úspěšném zápisu	ALARM_LATCH_RESET_WAIT
15	ALARM_LATCH_RESET_WAIT	Po dokončení požadavku	SENSED
16	SENSED	Není-li sSensed definován nebo je prázdný, pak další alarm	ALARM_PROCESS
		Po úspěšném přečtení	SENSED_WAIT
17	SENSED_WAIT	Po dokončení požadavku	ALARM_PROCESS
18	ALARM BYPASS	Je-li iAlarmPos >= 0, pak další alarm	ALARM_PROCESS
18		Jinak pokračuj od začátku	INIT

Tabulka 3.1: Stavový automat zpracování alarmů

Co dělat při problémech

V případě, že v diagnostických prostředcích systému REXYGEN (např. Watch režimu programu REXYGEN Studio) jsou neočekávané nebo nesprávné hodnoty vstupů nebo výstupů, je vhodné nejdříve ověřit jejich funkci nezávisle na systému REXYGEN. Dále je nutné překontrolovat konfiguraci. Nejčastější chyby jsou:

Chyba v hardware - špatné zapojení

Nejsou nataženy moduly jádra pro I2C nebo USB zařízení

Nesprávné device ID

Příliš dlouhá perioda úlohy, která čte signály z OwsDrv (v systémovém logu se pravidelně objevuje chyba Socket Error). Pak je třeba zvětšit timeout programu owserver parametrem příkazového řádku --timeout_server=60 (timeout zvětšen na 60 vteřin)

V případě, že daný vstup či výstup funguje pomocí jiných softwarových nástrojů správně a při shodném zapojení v systému REXYGEN nefunguje, prosíme o zaslání informace o problému (nejlépe elektronickou cestou) na adresu dodavatele. Pro co nejrychlejší vyřešení problému by informace by měla obsahovat:

- Identifikační údaje Vaší instalace vyexportované pomocí programu REXYGEN Studio (Target → Licensing → Export).
- Stručný a výstižný popis problému.
- Co možná nejvíce zjednodušenou konfiguraci řídicího systému REXYGEN, ve které se problém vyskytuje (ve formátu souboru s příponou .mdl).

Literatura

- Maxim Integrated. 1-Wire Application Notes. http://www.maximintegrated.com, 2013.
- [2] Paul Alfille. OWFS 1-Wire Filesystem. http://www.owfs.org, 2013.
- [3] REX Controls s.r.o.. Začínáme se systémem REXYGEN na platformě Raspberry Pi, 2020. →.

Referenční číslo dokumentace: 17159