Čidlo DS18B20

DALLAS SEMICONDUCTOR

Toto je neoficiální překlad Datasheetu čidla DS18B20. Za případné chyby se omlouvám.
Tento překlad vzniknul za účelem vlastní potřeby a v žádném případě s ním nesmí být nakládáno za účelem obohacení sebe nebo dalších osob ☺.

J.B.

Přehled informací

Obr.1 ukazuje blokové schéma čidla DS18B20 a popisuje ho tab.1. V 64-bit ROM je uložen unikátní kód – sériové číslo. Paměť scratchpad obsahuje 2x 8bit.registry s výsledkem převodu teploty, které jsou uloženy pro výstup v digitální formě. Dále paměť scratchpad poskytuje spojení s 1-bajtem horním a 1-bajtem spodním registrem alarmu (TH a TL) a 1-bajtem konfiguračním registrem. Konfigurační registr dovolí uživateli nastavit rozlišení převodu teploty na digitální číslo a to v rozlišení 9,10,11 nebo 12 bitů. TH, TL a konfigurační registry jsou nevolatelní (EEPROM), takže si udrží data i při odpojení napájení. Čidlo DS18B20 používá Dallase exklusivní 1-Wire bus protokol, který sdružuje komunikaci zařízení pouze po jednom drátu. Ovládací linka potřebuje slabý pull rezistor, všechny zařízení jsou připojené ke sběrnici přes 3-stavový port nebo port s otevřeným kolektorem (DQ pin u součástky DS18B20). V tomto systému sběrnice microprocessor (master) identifikuje a adresuje zařízení na sběrnici pomocí unikátního 64-bit. kódu. Protože každé zařízení má unikátní kód, může se teoreticky komunikovat s nekonečným počtem zařízení na sběrnici. 1-Wire bus protokol obsahuje vysvětlení příkazů a časování, které je obsaženo v kapitole 1-WIRE BUS SYSTEM tohoto datasheetu čidlaDS18B20.

Další vlastností čidla DS18B20 je schopnost pracovat bez externího napájecího zdroje. Napájení je provedeno přes 1-Wire pull-up rezistor přes pin DQ, tehdy, když je sběrnice v log.1. Log.1 sběrnice také nabíjí interní kondenzátor CPP, který nahrazuje napájení při sběrnici v log.0. Tato metoda odvození napájení ze sběrnice se nazývá „parazitní napájení“. Jako druhá možnost může být čidlo napájeno externím zdrojem přes pin V_{DD}.

Operace měření teploty

Jádrem funkce čidla DS18B20 je přímá digitální komunikace snímače teploty s masterem. Výsledek převodu teploty je uživatelem nastavitelný na 9,10,11 nebo 12 bitů, a tomu odpovídá rozlišení 0,5°C (9bitů); 0, 25°C (10bitů); 0,125°C (11bitů); 0,0625°C (12bitů). Výchozí nastavení po zapnutí napájení je 12-bitový převod. DS18B20 je po zapnutí v neurčitém stavu. K zahájení měření teploty a převodu A/D musí master jednotka (většinou to bývá mikroprocesor) vyslat příkaz „Convert T (44h)“. Následuje převod a výsledek je uložen ve dvou „teplotních registrech“ v paměti „Scratchpad“ a DS18B20 se vrátí do neurčitého stavu. Pokud je DS18B20 napájeno z externího zdroje, master jednotka může vysílat „čtecí časové úseky (read time slots)“ (koukni do kapitoly *1-WIRE BUS SYSTEM)* po příkazu Convert T (44h) a DS18B20 bude odpovídat log.0 při probíhání převodu teploty a log.1 až bude převod dokončen. Pokud je čidlo napájeno přes sběrnici *1-WIRE BUS SYSTEM,* toto oznámení o dokončení převodu nemůže být použito a na sběrnici musí být od začátku převodu log.1. Požadavky na napájení přes *1-WIRE BUS SYSTEM* je detailně vysvětleno v kapitole *POWERING THE DS18B20* tohoto datasheetu. Výstupní data (teplota) jsou ve °C, pro aplikace s použitím °F musí být použit převod softwarově. Převedená teplota na data je uložená jako 16-bitový výsledek - rozšířený dvojkový doplněk – číslo v „teplotních registrech“ – koukněte na obr.2. Znaménkový bit (S) signalizuje, zda je teplota kladná nebo záporná. Pro kladné teploty je S=0, pro záporné teploty S=1. Pokud je DS18B20 nastaven na 12-bit převod – výsledek – všechny bity v registrech obsahují platná data. Při převodu 11bitovém není použit bit0, při převodu 10bitovém nejsou použity bity 1 a 0, při převodu 9bitovém nejsou použity bity 2, 1 a 0. Tabulka 2 ukazuje příklady digitálních výstupních dat a odpovídajících teplot při 12-bitovém převodu.

Formát teplotních registrů - obr. 2

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LSB	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
MSB	S	S	S	S	S	2^6	2^5	2^4

Vztah teplota/data (12BIT) - tabulka 2

Teplota	digitální výstup (Binary)	digitální výstup (Hex)
+125°C	0000 0111 1101 0000	07D0h
+85°C*	0000 0101 0101 0000	0550h
+25.0625°C	0000 0001 1001 0001	0191h
+10.125°C	0000 0000 1010 0010	00A2h
+0.5°C	0000 0000 0000 1000	0008h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1000	FFF8h
-10.125°C	1111 1111 0101 1110	FF5Eh
-25.0625°C	1111 1110 0110 1111	FE6Fh
-55°C	1111 1100 1001 0000	FC90h

*Po zapnutí napájení je hodnota v registrech odpovídající +85°C

Signalizace Alarmu

Po vykonání teplotního převodu se výsledek porovná s uživatelem definovanými hodnotami registrů TH a TL (viz.obr.3). Znaménkový bit (S) signalizuje, zda je teplota kladná nebo záporná (bit0-6 – značí celou hodnotu bez desetin atd). Pro kladné teploty je S=0, pro záporné teploty je S=1. Registry TH a TL jsou v paměti EEPROM, takže jejich obsah se i po vypnutí napájení neztratí. TH a TL mohou být přístupný přes 2. a 3. bajt „scratchpad" jak vysvětluje část *MEMORY-PAMĚŤ* tohoto datasheetu.

Formát registru T_H a T_L (obr.3)

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
S	2^6	2^5	2^5	2^5	2^2	2^1	2^0

Pouze bity 4-11 z teplotních registrů se používají k porovnání pro alarm v registrech TH a TL. Pokud je naměřená teplota nižší nebo stejná jako TL nebo vyšší než TH, je nastaven příznakový bit alarmu uvnitř DS18B20. Tento příznakový bit je aktualizován každé měření teploty. Proto, pokud přestane alarm být platný, příznakový bit je dalším převodem vynulován.

Master může kontrolovat příznakový bit všech DS18B20 po sběrnici vysláním příkazu *ALARM SEARCH (ECh).* Každé DS18B20 s nastaveným příznakem alarmu bude reagovat na vyslaný příkaz, takže master musí jednoznačně určit, které čidlo DS18B20 bude testovat na příznak alarmu. Jestliže nějaká podmínka alarmu existuje a TH nebo TL registrům by byly přednastavené hodnoty, měla by se provést nová konverze pro potvrzení alarmové podmínky.

Napájení DS18B20

DS18B20 může být napájeno přivedením U na pin Vdd nebo může pracovat v režimu „tzv. parazitního napájení", kdy DS18B20 funguje bez externího napájení. Parazitní napájení je velice užitečné pro aplikace, kde se měří teploty na vzdálených místech nebo jsou velice omezeny prostorem. Obr.1 ukazuje parazitní napájení čidla, které odebírá napětí z 1-WIRE SBĚRNICE přes pin DQ, když je v log.1. Odebrané napájecí napětí čidla , když je sběrnice v log.1, nabije kondenzátor Cpp a z něj je čidlo napájeno při log.0 na sběrnici. Pokud je čidlo zapojeno v parazitním módu, musí být pin Vdd připojen na GROUND (zem).

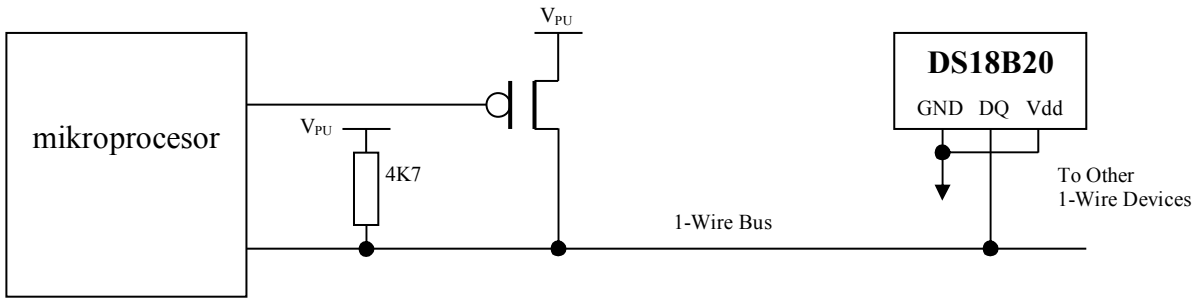
V parazitním módu sběrnice a kondenzátor Cpp mohou dostatečně napájet čidlo pro většinu operací tak dlouho, jak jsou specifikovány časové průběhy a požadavky jsou OK. (odkaz na DC a AC elektrické charakteristiky). Nicméně, když čidlo provádí konverzi teploty a kopíruje data z paměti SCRATCHPAD do EEPROM, napájení operace může být asi 1,5mA. Tento odběr může zapříčinit nepřijatelný pokles napětí přes slabý „1-WIRE pull-up" rezistor a spotřeba bude větší, než může kondenzátor Cpp dodat. K zajištění dostatečného napájení čidla je nezbytné poskytnout silný pull-up na sběrnici kdykoli se bude převádět teplota nebo se budou kopírovat data z SCRATCHPAD do EEPROM. Toho můžeme docílit použitím MOSFET připojeným přímo ke sběrnici viz. obr.4. Sběrnice musí být zapojená přes pull-up 10us (max) po vyslání příkazu převodu (příkaz Convert T (44h)) nebo kopírování SCRATCHPAD (příkaz Copy SCRATCHPAD (48h)) a sběrnice musí být „držena" v log.1 pull-upem po celou dobu převodu (t_{conv}) nebo přesunu dat (t_{wr}=10ms). Žádné další aktivity se nemůžou konat po sběrnici zatímco pull-up je povolen.

Čidlo může též být napájeno tradiční metodou připojení na externího napájení k Vdd pinu, jak ukazuje obr.5. Výhoda této metody je, že MOSFET pull-up není potřebný a sběrnice je volná pro přenášení ostatních komunikačních dat během převodu teploty.

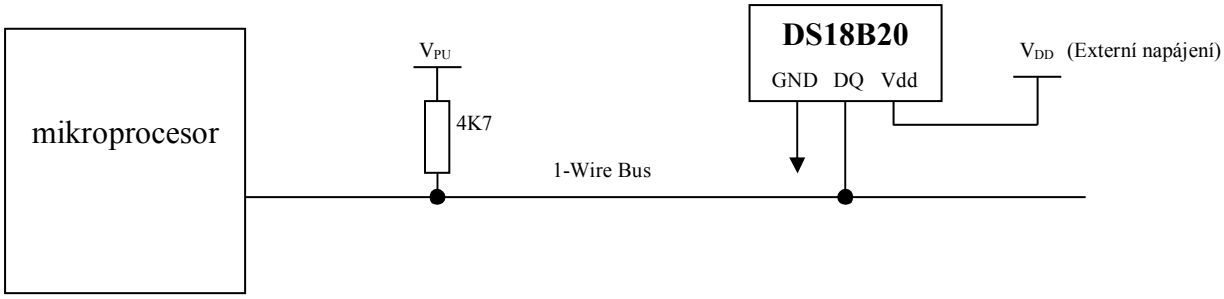
Použití parazitního napájení není doporučené pro teploty nad 100°C, ?protože není možné udržet požadavek komunikací k vyššímu prosakování proudu, které můžou existovat na těchto teplotách?. V aplikacích, ve kterých jsou tyto teploty pravděpodobné, je výrazně doporučeno napájení z externího napájení.

V některých situacích nemůže master vědět, zda čidla na sběrnici jsou napájena parazitně nebo externě. Master – uprocesor – potřebuje tuto informaci k určení, když bude použit pull-up během převodu teploty. K dostání této informace master může vyslat příkaz „Skip ROM (CCh)“, následně příkaz „Read Power Supply (B4h)“ a následně „číst stav sběrnice“. Výsledek čtení sběrnice – při parazitním napájení čidla je sběrnice v log.0 a při externím napájení čidla bude v log.1. Když je sběrnice v log.0, master „ví“, že musí sběrnici (čidlo) napájet pull-up na 1-Wire sběrnici během převodu teploty.

Parazitní napájení DS18B20 během převodu teploty – obr. 4



Napájení DS18B20 externím zdrojem – obr. 5



64-BIT LASERED ROM KÓD

Každé čidlo DS18B20 obsahuje unikátní 64–bit kód (viz.obr.6) uložený v ROM. Posledních 8 bitů z ROM kódu obsahuje čidlo 1-Wire „rodinný kód“: DS18B20 je 28h. Následujících 48 bitů obsahuje unikátní sériové číslo. Nejvyšších 8 bitů obsahují „kruhovou kontrolu načtených dat“- cyclic redundancy check (CRC) bajt , který je počítán z prvních 56 bitů ROM kódu. Detailní vysvětlení CRC bitů je poskytnuto v kapitole *CRC GENERATION.* 64-bit ROM kód a přidružená ROM funkční kontrolní logiky dovolí čidlu pracovat jako 1-Wire zařízení používající protokol popsaný v kapitole *1-WIRE BUS SYSTÉM*

64-BIT ROM KÓD – obr. 6

8-bitů CRC	48 bitů sériové číslo	8-bit „rodinný kód“
MSB LSB	MSB LSB	MSB LSB

Paměť

Paměť čidla je organizovaná viz. obr.7. Paměť se stává z SRAM „scratchpad“ – paměť zápisník s EEPROM pamětí pro ukládání registrů horní a spodní hranice alarmu teploty (TH a TL) a konfigurační registr. Poznámka - když funkce alarmu čidla DS18B20 není využita, TH a TL registry mohou sloužit jako general-purpose memory (hlavní účelná paměť). Všechny příkazy pro paměť jsou popsány detailně v kapitole FUNCTION COMMANDS (Funkční příkazy).

Bajt 0 a bajt 1 z „scratchpad“ obsahuje LSB a MSB registry přiřazené pro převod teploty. Tyto bajty jsou pouze čtecí. Bajty 2 a 3 jsou TH a TL registry. Bajty 4 obsahuje konfigurační registr dat, který je detailně popsán v kapitole CONFIGURATION
REGISTR. . Bajty 5, 6, a 7 jsou rezervovány pro interní použití zařízením a nemohou být přepsány. Tyto bajty budou při čtení vždy log.1. Bajt 8 z scratchpadu je pouze pro čtení a obsahuje „cyclic redundancy check“ (CRC) - kód pro bajty 0-7 scratchpadu. DS18B20 – čidlo – generuje tento CRC použitím metody popsané v kapitole CRC GENERATION.
Data jsou psána z bajtů 2,3 a 4 scratchpadu použitím příkazu Write Scratchpad [4Eh]; data musí být poslána do čidla startem s LSB z bajtu 2. K ověření úplnosti dat může být scratchpad přečten (použitím příkazu Read Scratchpad [BEh]) po zapsání. Když čteme scratchpad, data jsou přenášena po 1-Wire sběrnici s začátkem LSB z bajtu 0. K přenosu TH, TL a konfiguračního registru z scratchpadu do EEPROM, procesor musí vyslat příkaz Copy Scratchpad [48h].
Data v EEPROM registrech jsou uloženy i po výpadku Ucc a po zapnutí jsou data opět načtena do příslušných registrů do scratchpadu. Data mohou tedy být načteny z EEPROM do scratchpadu kdykoli příkazem Recall E^2 [B8h]. Procesor (master) může číst stav sběrnice (Read time slot) po příkazu Recall E^2 a čidlo DS18B20 bude indikovat stav vysíláním log.0.pokud bude pracovat a log.1 pokud bude hotové.

Rozložení paměti DS18B20 – obr.7

SCRATCHPAD	**(Stav po zapnutí)**		**EEPROM**
bajt 0	Teplotní reg. LSB (50h) °		
bajt 1	Teplotní reg. MSB (05h) °		
bajt 2	T_H reg. nebo uživ. bajt 1*	$\longleftrightarrow$	T_H reg. nebo uživ. bajt 1
bajt 3	T_L reg. nebo uživ. bajt 2*	$\longleftrightarrow$	T_L reg. nebo uživ. bajt 2
bajt 4	Konfigurační reg.*	$\longleftrightarrow$	Konfigurační reg.
bajt 5	Reservováno pro čidlo (FFh)		
bajt 6	Reservováno pro čidlo (0Ch)		
bajt 7	Reservováno pro čidlo (10h)		
bajt 8	CRC*		

* Stav po zapnutí závisí na hodnotě uložené v EEPROM
° po zapnutí je hodnota těchto registrů odpovídající teplotě 85°C

Konfigurační registr

Bajt 4 scratchpadu obsahuje konfigurační registr s rozložením bitů viz.obr. 8. Uživatel může nastavit rozlišení převodu teploty čidla použitím bitů R0 a R1 viz.tabulka 3. Po zapnutí Ucc jsou tyto bity nastaveny na log.1 = 12bit rozlišení. Poznámka – rozlišení je přímo závislé na časové délce převodu. Bit 7 a bit 0-4 v konfiguračním registru jsou rezervovány pro interní použití a nemůžou být zapsány; tyto bity jsou čteny jako log.1.

Konfigurační registr – obr.8

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	R1	R0	1	1	1	1	1

Závislost rozlišení převodu na času - tabulka 3

R1	R0	Resolution	Max Conversion Time
0	0	9-bit	93.75 ms ($t_{CONV}/8$)
0	1	10-bit	187.5 ms ($t_{CONV}/4$)
1	0	11-bit	375 ms ($t_{CONV}/2$)
1	1	12-bit	750 ms (t_{CONV})

Generování CRC

CRC kód je vytvořen buď jako část ze 64-bit ROM kódu čidla a nebo se nachází v 9. bajtu scratchpadu (bajt 8). Jsou to dva rozdílné CRC bajty, které se ale vytváří a počítají zcela stejně (ovšem z jiných dat). Kód CRC jako část ROM kódu je vypočítán z prvních 56 bitů ROM kódu a je obsažen v MSB bajtu ROM kódu. Kód CRC ve scratchpadu je vypočítán z dat uložených ve scratchpadu a ten je proto změněn při každé změně dat ve scratchpadu. CRC poskytuje procesoru kontrolu správného načtení dat z čidla. K ověření správnosti dat musí procesor vypočítat CRC z obdržených dat a porovnat výsledek s hodnotou na 8 bajtu ROM-kódu (při čtení ROM kódu) nebo porovnat výsledek s hodnotou na 8 bajtu Scratchpadu (bajt CRC) (při čtení dat ze Scratchpad paměti čidla). Pokud vypočítaná hodnota CRC souhlasí (viz. níže), data jsou přijatá bez chyby. Porovnání načteného bajtu CRC a vypočítaného CRC masterem a rozhodnutí o pokračování v operaci jsou určena pouze masterem. V čidle DS18B20 není žádné vnitřní zapojení, které zabrání komunikovat čidlu s masterem, když se CRC načtené z čidla nerovná hodnotě vypočítané masterem.

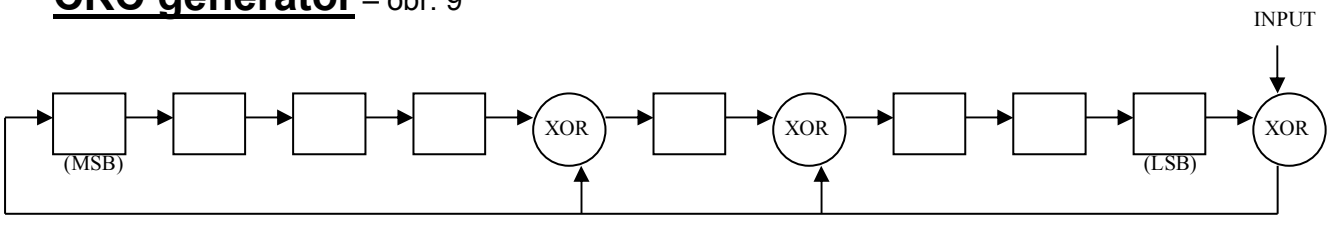
Vzorec výpočtu CRC je:

$$CRC = X^8 + X^5 + X^4 + 1$$

Procesor může přepočítat CRC a porovnat výsledek s hodnotou z čidla (DS18B20) použitím vzorce viz.obr. 9. Toto zapojení se skládá z posuvného registru a XOR hradla. Před začátkem výpočtu musí být registr vynulován! Začíná se nejnižším bitem (LSB) ROM kódu při ověření dat při čtení ROM kódu nebo LSB bajtu 0 ze scratchpadu při ověření dat při čtení paměti scratchpad, jeden bit za druhým se posouvá do posuvného registru. Po posunutí 56. bitu z ROM kódu nebo MSB bajtu 7 ze scratchpadu, bude obsahovat generátor přepočítanou hodnotu CRC. Nyní následuje výpočet s načtenou hodnotou CRC bajtu ROM kódu nebo bajtu CRC ze scratchpadu. Výsledek po úplném výpočtu CRC musí být 0. Pokud není, data nebyla načtena správně.

Další informace o Dallas 1-Wire CRC DS18B20 je k dispozici v Aplikačním notesu *27: Pochopení a použití CRC s Dallas Semiconductor Touch Memory Products*.

CRC generátor – obr. 9



1-WIRE BUS SYSTEM

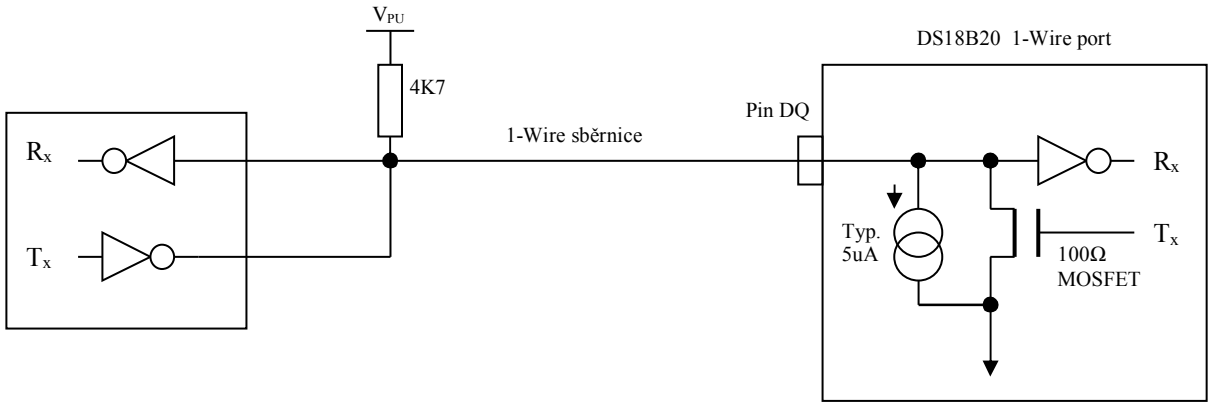
1-Wire sběrnice používá jednu sběrnici od mastera (procesoru) k jednomu nebo více zařízením. DS18B20 je vždy jako „slave“. Když je pouze jedno zařízení na sběrnici, systém je nazýván jako “single-drop” systém; pokud je na sběrnici více zařízení, nazýváme tento systém jako “multidrop”. Všechny data a příkazy jsou posílány nejprve LSB po 1-Wire sběrnici. Následující komunikace po 1-Wire sběrnici je rozdělena do tří částí – sekvencí a to: konfigurační sekvence, přenášecí sekvence a 1-Wire signály (signály druhové a časové).

Hardwarová konfigurace

1-Wire sběrnice má samozřejmě pouze jeden signálový vodič. Každé zařízení (master nebo slave) se připojí na tento vodič přes „open-drain“ (otevřený kolektor) nebo 3-stavový port. Toto dovolí každému zařízení uvolnit sběrnici, když zařízení nezasílá data a sběrnice je možná použít pro ostatní zařízení. 1-Wire port čidla DS18B20 (pin DQ) je otevřený kolektor s vnitřním zapojením viz.obr. 10.

1-Wire sběrnice požaduje externí pull-up rezistor o hodnotě asi 5kΩ; to pro stav 1-Wire sběrnice do log.1. Když bude z nějaké příčiny přenos pozastaven, sběrnice musí být ponechána v klidovém stavu (log.1), když přenos má znovu začít. Neurčitý obnovovací čas může nastat mezi bity pokud 1-Wire bus je v nečinném (1) stavu během obnovovací periody. Když je sběrnice držena na log.0 více jak 480µs, všechny komponenty na sběrnici budou resetovány.

Hardwarová konfigurace – obr.10



Transakční sekvence

Transakční sekvence pro přístup do čidla je následující:
Krok 1. Inicializace
Krok 2. Příkaz ROM (následovaný nějakou požadovanou výměnou dat)
Krok 3. DS18B20 funkční příkaz (následovaný nějakou požadovanou výměnou dat)

Je velice důležité dodržet tuto sekvenci vždy při komunikaci, jinak čidlo nebude odpovídat, pokud nějaký krok bude vynechán. Výjimku tvoří příkazy *Search ROM [F0h]* a *Alarm Search [ECh]*. Po vyslání jednoho z těchto příkazů master musí zpět ke kroku 1 v sekvenci.

Krok 1. - Inicializace

Všechny transakce na -Wire sběrnici začínají s inicializační sekvencí. Inicializační sekvence obsahuje RESET puls od masteru a následovaným presentačním pulsem (pulsy) zaslaným čidlem (čidly) (DS18B20). Presentační puls dává masteru vědět o zařízeních na sběrnici (najité jako DS18B20) připravené k práci. Časování RESET a present pulsů jsou detailně popsány v kapitole *1-WIRE SIGNALING*.

Krok 2. – Příkazy ROM

Po detekci presentačního pulsu může master vyslat příkazy ROM. Tyto příkazy vyvolají unikátní 64-bit ROM kód z každého zařízení na sběrnici a povolí masteru jednoznačně specifikovat zařízení, pokud jich

je na sběrnici více. Tyto příkazy tedy dovolí masteru určit kolik a jaké typy zařízení jsou nalezeny na sběrnici nebo kolik zařízení má nastavenou podmínku alarmu. Je pět příkazů ROM a každý příkaz je 8 bitový. Mikroprocesor musí vyslat příslušný příkaz ROM před vysláním funkčního příkazu. Vývojový diagram pro operaci příkazů ROM je na obr.11

SEARCH ROM (F0h)

Když je systém po zapnutí napájení inicializovaný, mikroprocesor musí identifikovat kódy ROM všech SLAVE zařízení na sběrnici, které dovolí mikroprocesoru určit číslo čidla a jeho typ. Master se naučí kódy ROM během procesu vyloučení (eliminace), který požaduje k vykonání SearchROM cyklu, tj. příkaz *SEARCH ROM* a následná výměna dat bude tolikrát,kolikrát je potřeba k identifikaci všech čidel. Když je na sběrnici pouze jedno čidlo, může být použit místo příkazu SearchROM příkaz ReadROM (viz.níže). Detailní vysvětlení příkazu SearchROM a jeho procedury - viz. ***iButton® Book of Standards*** **at www.ibutton.com/ibuttons/standard.pdf.**
Po každém cyklu SearchROM se musí mikroprocesor vrátit na krok 1 inicializace transakční sekvenci

READ ROM (33h)

Tento příkaz může být použit pouze pokud je na sběrnici pouze jedno zařízení – čidlo. To dovolí mikroprocesoru číst 64-bit ROM kód čidla bez použití procedury SearchROM. Pokud je tento příkaz použit, když je na sběrnici víc jak jedno zařízení, nastane kolize dat, protože všechna čidla se pokusí vyslat data současně.

MATCH ROM (55h)

Příkaz MatchROM a následné vyslání 64-bitového ROM kódu dovolí masteru komunikaci s požadovaným slave-zařízení na sběrnici multidrop nebo singledrop. Pouze slave zařízení, kterému odpovídá 64-bitový ROM kód, bude odpovídat na funkční příkazy vyslané masterem. Všechny ostatní slave (čidla) zařízení na sběrnici budou čekat na resetovací impuls.

SKIP ROM (CCh)

Mikroprocesor může použít tento příkaz na sběrnici i bez vyslání informačních ROM kódů. Např. mikroprocesor může na sběrnici vyvolat převod teploty vysláním příkazu *SKIP ROM* a následně příkazem *ConvertT(44h).*
Pozn. Příkaz ReadScratchpad může následovat za příkazem SkipROM pouze pokud je pouze jedno zařízení na sběrnici. V tomto případě je ušetřeno to, že mikroprocesor může číst ze slave bez zaslání 64 bitového ROMkódu zařízení. Pokud by bylo na sběrnici víc než jedno slave (čidlo), příkaz SkipROM a následný příkaz *Read Scratchpad* bude příčinou kolize dat na sběrnici, protože se budou pokoušet posílat data všechny slave současně.

ALARM SEARCH (ECh)

Operace tímto příkazem je stejná jak operace příkazem Search ROM kromě toho, že budou odpovídat pouze čidla s nastaveným příznakem alarm. Tento příkaz umožní masteru zjistit, zda nějaké zařízení DS18B20 splnilo podmínku „alarmu“ v posledním teplotním převodu. Po každém cyklu *ALARM SEARCH* (tj. Příkaz Alarm Search a následná výměna dat) se musí master vrátit na krok 1 – Inicializace v transakční sekvenci. V kapitole *OPERATION - ALARM SEARCH* je vysvětlena operace s příznakem alarmu.

Krok 3. – Funkční příkazy DS18B20

Po použití příkazu ROM k adresaci čidla DS18B20, se kterým budeme komunikovat, master může vyslat jeden funkční příkaz pro čidlo DS18B20. Tyto příkazy dovolí masteru zapisovat do a číst z paměti Scratchpad, zahájit převod teploty a zjistit druh napájení. Funkční příkazy čidla DS18B20, které jsou níže napsány jsou seřazeny v tabulce 4 a jsou představeny vývojovým diagramem na obr. 12.

CONVERT T (44h) – převod teploty

Tento příkaz zahájí jeden převod teploty. Převod teploty - výsledná „teplotní“ data se uložení do dvou 8bit registrů ve scratchpad paměti a čidlo DS18B20 se vrátí do „klidového stavu“ (bude čekat na inicializační sekvenci). Když je zařízení zapojeno v parazitním módu, master musí během 10µs po tomto příkazu uvolnit 1-Wire sběrnici, aby se na sběrnici „udělala“ log.1 a mohlo se napájet čidlo po dobu konání převodu t_{conv}, jak je popsáno v kapitole *NAPÁJENÍ (POWERING THE) DS18B20*. Když je čidlo DS18B20 napájeno externím zdrojem, master může po příkazu *ConvertT* vysílat „čtecí časové úseky“ a čidlo DS18B20 bude odpovídat log.0 během převodu teploty a log.1 po jeho ukončení. V parazitním módu tato informace o průběhu převodu není možná, protože sběrnice musí být v log.1 během celého převodu.

WRITE SCRATCHPAD [4Eh] – zápis do paměti

Tento příkaz dovolí masteru zapsat 3 bajty dat do paměti scratchpad čidla. První bajt je zapsán do registru TH (bajt 2 v paměti scratchpadu), druhý bajt do registru TL (bajt 3) a třetí do Konfiguračního registru (bajt 4). Data musí být poslána - první bit je LSB. Všechny tři bajt MUSÍ být zapsány před vykováním resetu masterem, neboť data by se nezapsala správně!!!

READ SCRATCHPAD [BEh] – čtení paměti

Tento příkaz dovolí masteru číst obsah paměti scratchpad. Přenášená data se začínají číst LSB bajtem 0 a pokračují až do devátého bajtu (bajt 8 – CRC). Master smí vyslat RESET pro ukončení čtení kdykoli, pokud chce pouze část dat z paměti.

COPY SCRATCHPAD [48h] – kopírování paměti

Tento příkaz zkopíruje obsah paměti scratchpad TH, TL a konfigurační registry (bajty 2, 3 a 4) do EEPROM v čidlu. Když je slave používán v parazitním módu, během 10µs (max) po tomto příkazu musí master uvolnit 1-WIRE sběrnici (aby se na sběrnici „udělala“ log.1 a mohlo se napájet čidlo) nejméně na 10ms jak je napsáno v kapitole POWERING THE DS18B20 – (napájení DS18B20)

RECALL E^2 [B8h]

Tento příkaz vyvolá hodnoty TH, TL a konfigurační reg. z EEPROM čidla a uloží je na místa bajtů 2, 3, a 4 v paměti scratchpad. Master může po příkazu *Recall E^2* vysílat „čtecí časové úseky“ a čidlo DS18B20 bude odpovídat log.0, pokud bude proces probíhat a log.1, pokud bude proces dokončen. *Recall E^2* operace proběhne automaticky po zapnutí napájení, aby byly připraveny pro práci aktuální data.

READ POWER SUPPLY [B4h]

Master vyšle tento příkaz pro určení druhu napájení čidla. To zjistíme tak, že po vyslání příkazu *Read Power Supply* vyšle master „čtecí časový úsek“ – při parazitním napájení bude „čtený časový úsek“ log.0 a při externím napájení bude log.1. V kapitole Napájení - POWERING THE DS18B20 jsou informace k použití pro tento příkaz.

DS18B20 Popis funkčních příkazů – tab.4

Příkaz	Popis	Protokol (kód)	Stav 1-Wire sběrnice po použití příkazu	Pozn.
Příkazy převodu teploty				
Convert T	Zahájení převodu teploty	44h	DS18B20 vysílá informaci o stavu převodu pro mastera (pouze při externím napájení čidla)	1
Příkazy pro paměť				
Read Scratchpad	Čte veškerý Scratchpad obsahující i bajt CRC	BEh	DS18B20 posílá až všech 9 bajtů Scratchpadu do mastru	2
Write Scratchpad	Zapisuje data bajtů 2,3,4 (T_H, T_L, konfig.reg.) do Scratchpadu	4Eh	Master posílá 3 datové bajty do čidla DS18B20	3
Copy Scratchpad	Kopíruje T_H, T_L, konfig.reg. ze Scratchpadu do EEPROM čidla	48h	Není	1
Recall E^2	Vybírá T_H, T_L, konfig.reg. z EEPROM čidla do Scratchpadu	B8h	Čidlo DS18B20 posílá informace o stav transakce Recall E^2	
Read Power supply	Signály pro master ke zjištění módu napájení	B4h	DS18B20 posílá druh napájení pro mastera	

Pozn.:

1. Pro parazitní napájení čidel master musí povolit pull-up na sběrnici během převodu teploty a kopírování ze Scratchpadu do EEPROM čidla. Žádné další (činnosti) příkazy toto nepotřebují.
2. Master může přerušit přenos dat kdykoliv pomocí resetu
3. Všechny tři bajty musí být zapsány před použitím resetu.

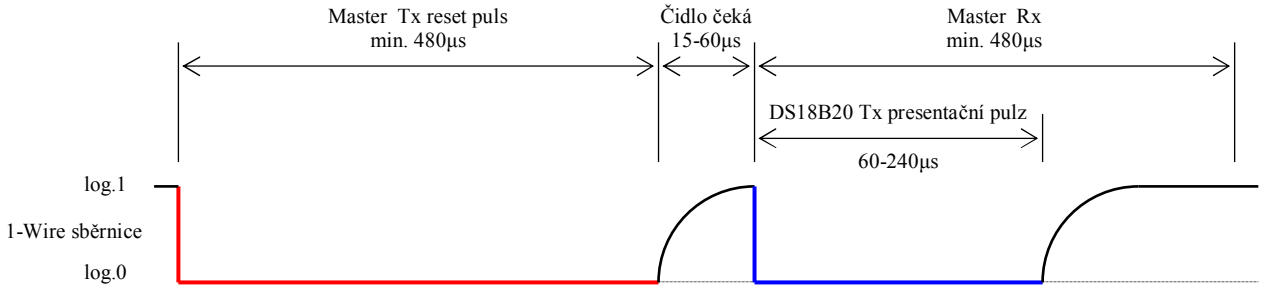
1-WIRE SIGNALING – časování signálů

DS18B20 používá komunikační protokol „1-Wire communication“ k zabezpečení správného přenosu dat. Několik typů signálů z tohoto protokolu je tu popsáno. Jsou to: Reset puls, presentační puls, zápis 0, zápis 1 , čtení 0 a čtení 1. Master začíná všechny tyto signály (pulsy) s výjimkou presentačního pulsu.

Inicializace: Reset a Presentační puls

Všechna komunikace s DS18B20 začíná inicializační sekvencí, která obsahuje Reset puls a za ním následuje presentační puls z čidla DS18B20 (viz.obr.13). Když DS18B20 pošle inicializační puls na základě Reset pulsu je to potvrzení správného připojení a připravenosti čidla DS18B20 komunikovat s masterem. Během inicializační sekvence master zasílá (Tx) Reset puls - 1-Wire sběrnice v log.0 minimálně 480µs. Master poté uvolní sběrnici a přejde do režimu přijímače (Rx). Když je sběrnice uvolněna, 5k pull-up rezistor uvede 1-Wire sběrnici do log.1. Když DS18B20 detekuje náběžnou hranu, počká ještě 15µs až 60µs a poté pošle presentační impuls - 1-Wire sběrnice do log.0 na 60µs to 240µs.

Časování inicializační sekvence – obr.13



Signál od masteru
Signál od čidla DS18B20
Navrácení do log.1 Pull-up rezistorem

READ/WRITE (čtecí/zapisovací) časový průběh

Master zapisuje/čte data do/z čidla DS18B20 během časového úseku. Jeden datový bit je posílán přes 1-Wire sběrnici za jeden časový úsek.

WRITE (zapisovací) časový úsek

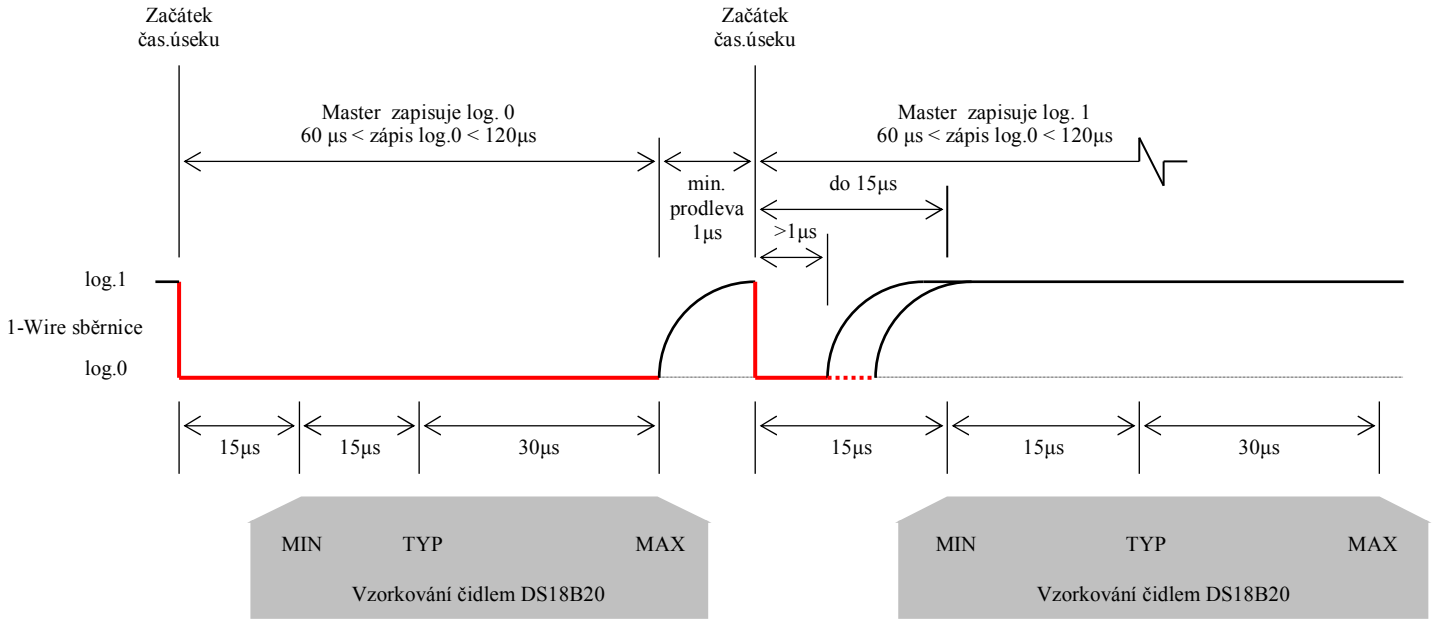
Zde máme dva typy: “Zápis 1” a “Zápis 0”. Master používá logiku „Zápis 1= log.1“ a „Zápis 0=log.0“. Všechny zapisovací časové průběhy musí být minimálně dlouhé 60µs s minimálně 1µs prodlevou pro navrácení sběrnice do log.1. mezi jednotlivými časovými úseky. Oba dva úseky zápisu začínají tak, že master udělá log.0 na 1-Wire sběrnici (viz.obr. 14).

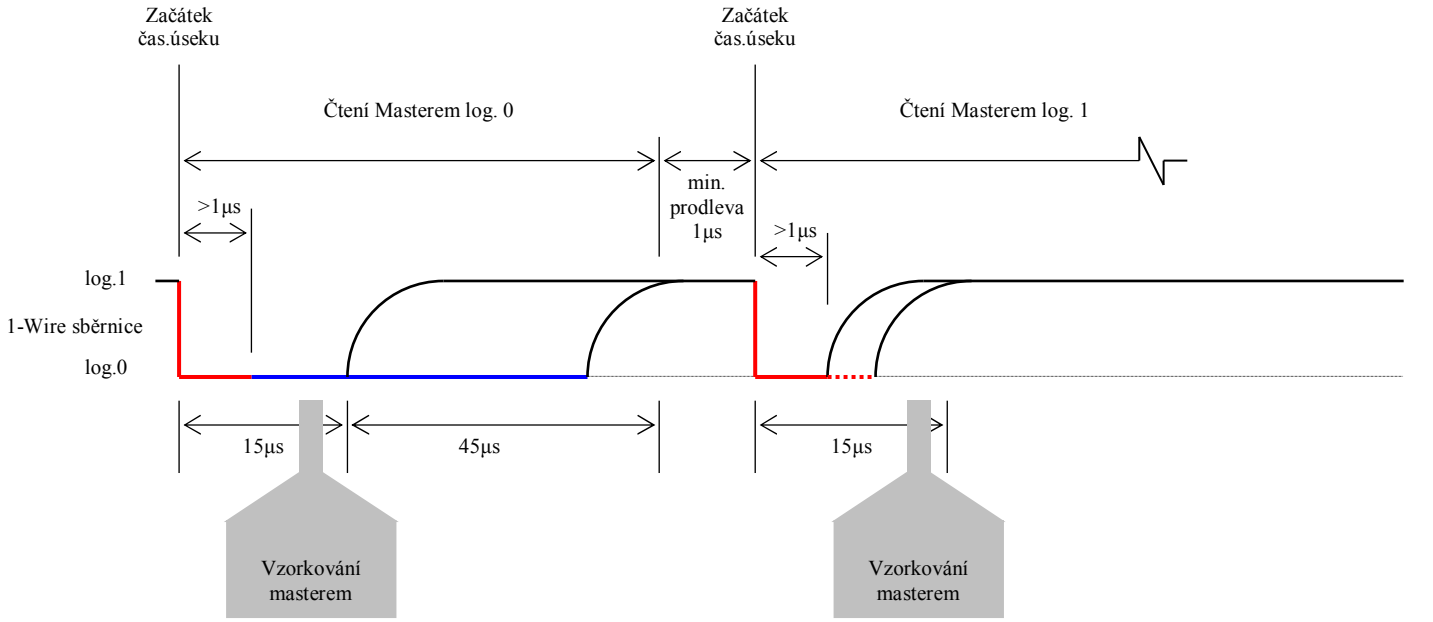
Pro vytvoření Zápisu log.1 puls začíná tak, že master uvede sběrnici do log.0 a během 15 µs po začátku časového úseku - musí sběrnici uvést do log.1.

Pro vytvoření Zápisu log.0 puls začíná tak, že master uvede sběrnici do log.0 a tato úroveň musí na sběrnici trvat více jak 60µs a méně než 120µs.

Čidlo DS18B20 vzorkuje 1-Wire sběrnici a podle poslední zjištěné hodnoty mezi 15µs až 60µs po zahajovací log.0 mastrem vyhodnotí hodnotu bitu. Když nadetekuji 1, tuto informaci bere čidlo jako log.1., když nadetekuji 0, tuto informaci bere jako log.0.

READ/WRITE průběh časového úseku – obr.14





Signál od masteru
Signál od čidla DS18B20
Navrácení do log.1 Pull-up rezistorem

READ (čtecí) časový úsek

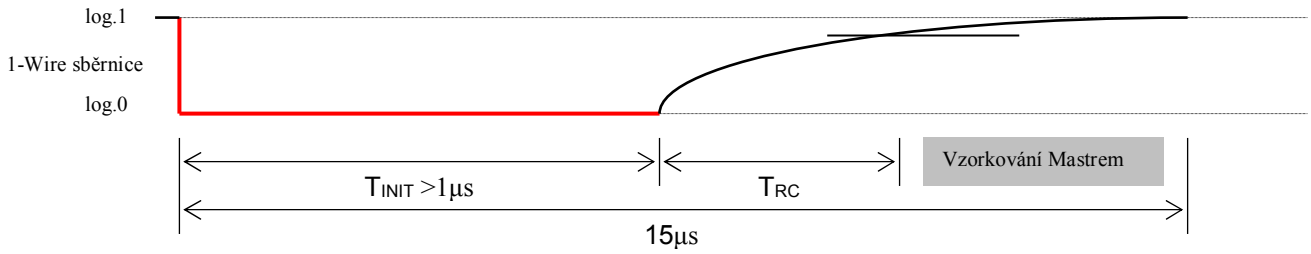
Čidlo DS18B20 může posílat data do masteru pouze, když master vyšle „Čtení časového úseku“. Proto master musí generovat čtecí časový úsek ihned po vyslání příkazu *Read Scratchpad [BEh]* nebo *Read Power Supply [B4h]*, takže DS18B20 může poskytnout potřebná data. Navíc master může generovat čtecí časové úseky po vyslání příkazu *Convert T [44h]* nebo *Recall E² [B8h]* pro zjištění stavu operace jak je vysvětleno v kapitole *DS18B20 FUNCTION COMMAND*.

Všechny čtecí časové úseky musí být dlouhé minimálně 60µs s minimální prodlevou mezi pulsy 1µs. Čtecí časové úseky začínají vysláním log.0 na sběrnici na dobu min. 1µs a poté musí master sběrnici uvolnit (viz.obr.14). Po začátku čtecího časového úseku čidlo DS18B20 začne posílat 1 nebo 0 po sběrnici.DS18B20 posílá 1 tak, že pustí sběrnici do log.1 a 0 posílá držením sběrnice v log.0. Když je posílaná 0, čidlo DS18B20 uvolní sběrnici po ukončení časového úseku a sběrnice bude navrácena do log.1pull-up rezistorem. Výstupní data z čidla DS18B20 jsou platná po 15µs sestupné hrany generované masterem (začátek čtení). Proto master musí uvolnit sběrnici a vzorkovat její úroveň (stav) po 15µs od začátku čtecího časového úseku (odstartovaným masterem).

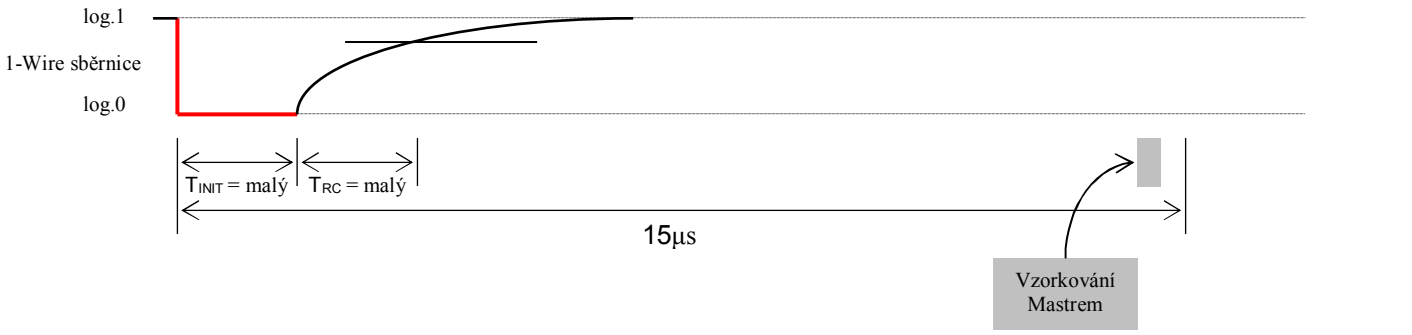
Z obr.15 je patrné, že součet T_{INIT}, T_{RC}, a $T_{VZORKOVANÍ}$ musí být menší, než 15µs pro čtení časového úseku.

?Obr. 16 ukazuje, že systémové časování okraje je maximalizovaný s držením T_{INIT} a T_{RC} tak krátce, jak je to možné a umístění vzorkování mastrem během čtecího časového úseku je ke konci 15µsekondové periody.?

Detailní časování čtení masteru – obr.15



Doporučené časování čtení log.1 – obr.16



Signál od masteru
Navrácení do log.1 Pull-up rezistorem

Vysvětlivky:

Scratchpad – paměť ROM v čidle, kde se nachází potřebné registry pro práci
„1-WIRE pull-up“ rezistor – rezistor zajišťující log.1 na sběrnici (viz.obr.4)
1-WIRE SBĚRNICE – sběrnice firmy Dallas SEMICONDUCTOR – komunikace po 1 drátu
DS18B20 – čidlo o kterém toto celé je ☺
Cpp – vnitřní kondenzátor, který se využívá při parazitním napájení
Master – řídící zařízení 1-WIRE SBĚRNICE – většinou µprocesor
„rodinný kód“ – kód druhu čidla (např. řada čidel DS18xXX)
MSB - více významný bit – u 8bit. registrů je to bit 7
LSB - poslední významný bit - u 8bit. registrů je to bit 0

Příkazy ROM

- SEARCH ROM (F0h)
- READ ROM (33h)
- MATCH ROM (55h)
- SKIP ROM (CCh)
- ALARM SEARCH (ECh)

Funkční příkazy DS18B20

- CONVERT T (44h) – převod teploty
- WRITE SCRATCHPAD [4Eh] – zápis do paměti
- READ SCRATCHPAD [BEh] – čtení paměti
- COPY SCRATCHPAD [48h] – kopírování paměti
- RECALL E2 [B8h]
- READ POWER SUPPLY [B4h]

Obslužná rutina: (pohled od mastera)

A. Zjištění přítomnosti a správné funkce čidla
(po zapnutí Ucc je v registrech hodnota +85°C

1. Vyslat Reset
2. Přijmout Presentační puls
3. Zjistit přijmutí presentačního pulsu – ANO –pokračuj // NE – chyba čidla
4. Vyslat SKIP ROM (CCh)
5. Vyslat READ SCRATCHPAD (BEh) (bajt 0 a bajt 1 Scratchpadu)
6. Přijmout 2 bajty z čidla, výsledek uložit do registrů TEMP_LSB a TEMP_MSB
7. Master kontroluje načtenou hodnotu, zda je +85°C (0550h) – ANO – pokračuj // NE – chyba hodnoty

8. Vyslat Reset
9. Přijmout Presentační puls
10. Vyslat SKIP ROM (CCh)
11. Vyslat WRITE SCRATCHPAD – Zapíše T_H, T_L, a konfig.reg. = rozlišení převodu
12. Vyslat požadovaná data do čidla T_H – 00h , T_L - 00h , konfig.reg. – 00h (=rozlišení 9bit)

13. Vyslat Reset
14. Přijmout Presentační puls
15. Vyslat SKIP ROM (CCh)
16. Vyslat ConvertT (44h)
17. ?Master kontroluje stav převodu? (nepoužívá se při parazitní napájení) – (převod trvá **95ms**)

18. Vyslat Reset
19. Přijmout Presentační puls
20. Vyslat SKIP ROM (CCh)
21. Vyslat READ SCRATCHPAD (BEh) (bajt 0 a bajt 1 Scratchpadu)
22. Přijmout 2 bajty z čidla, výsledek uložit do registrů TEMP_LSB a TEMP_MSB

Ax_RESET_C – Presentační část komunikace master / čidlo

Ax_WR_DO_C - Zápis do čidla, zaslání příkazu – 8bitů (**před** zavoláním procedury musí být v **W** zapisovaná hodnota)

Ax_CTI_Z_C - Čtení dat z čidla – 8bitů (**po** ukončení procedury jsou načtená data v reg. **TMPO**).