PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

Monitorování teplot pomocí čidel DS1822

2011 Roman Polanský

Anotace

Bakalářská práce Monitorování teplot pomocí čidel DS1822 se zabývá sběrem teplot z teplotních čidel DS1822 a následným zobrazením naměřených hodnot.

Prohlášení

Prohlašuji, že jsem vytvořil tuto bakalářskou práci samostatně za vedení Mgr. Martina Dostála Ph.D., a že jsem v seznamu použité literatury uvedl všechny zdroje použité při zpracování práce.

V Olomouci 10. 08. 2011

Obsah

Seznam použitých zkratek

Výraz	Popis
Master	Zařízení připojené na sběrnici (PC, mikroprocesor atd.)
Slave	Teplotní čidlo DS1822
1-Wire	Jednovodičové rozhraní pro komunikaci s připojenými zařízeními
DQ	Datový vstupně výstupní pin. Otevřený kanál rozhraní pinu 1-Wire sběrnice
VDD	Volitelný V_{DD} pin. V_{DD} pin, který musí být propojen s pinem GND v parazitním módu.
GND	Uzemnění(nulový vodič)
Scratchpad	Vnitřní paměť DS1822, která uchovává data
EEPROM	Elektricky mazatelná semipermanentí paměť typu ROM-RAM
SRAM	Statická polovodičová paměť
ROM	Typ elektronické paměti, kdy její obsah nelze měnit (je dán z výroby)
CRC	Cyklický redundantní součet, který se používá k detekci chyb
Read Time Slot	Čtení logických hodnot z časového slotu uvnitř DS1822
Write Time Slot	Zápis logických hodnot do časového slotu uvnitř DS1822
Release	Uvolnění sběrnice
High	1-Wire sběrnice, která je pod napětím
Low	1-Wire sběrnice, která není pod napětím
SingleDrop	Jediné zařízení DS1822, které je připojeno na sběrnici
Multidrop	Více zařízení DS1822 připojených na sběrnici

Tabulka 1. Seznam použitých zkratek

1. Úvod

Cílem této bakalářské práce je vytvořit webovou aplikaci pro monitorování hodnot teplotních čidel. Aplikace má umožňovat zobrazení aktuálních hodnot vybraných čidel, volbu mezi číselným a grafickým zobrazením zaznamenaných hodnot, vykreslování grafů teplotních dat zaznamenaných v databázi v zadaném časovém úseku a nalezení maxima a minima v tomto časovém úseku.

Nejprve se čtenář seznámí s inteligentním teplotním čidlem DS1822 a 1-Wire sběrnicovým systémem, pomocí kterého komunikuje teplotní čidlo se svým okolím. Detailním prozkoumáním popisu teplotního čidla a 1-Wire sběrnice získá čtenář základní představu o tom, jak komunikuje teplotní čidlo DS1822 se svým okolím.

Prvořadým úkolem této bakalářské práce je vytvoření webové apli- kace pro monitorování hodnot teplotních čidel, k čemuž byly vybrány dnes již běžně dostupné technologie (skriptovací jazyk PHP, webový server Apache, databáze MySQL, javascript a framework jQuery).

Dalším úkolem této bakalářské práce je vytvoření aplikace, která sbírá naměřené teploty a ukládá je do databáze. Pro vytvoření samostatné aplikace pro sběr naměřených hodnot byly použity následující technologie: Microsoft Framework 2.0, programovací jazyk C# a databáze MySQL.

Posledním úkolem této bakalářské práce je zpracovat uživatelskou a programátorskou dokumentaci.

Vývoj aplikace a následné testy byly prováděny na notebooku Acer Extensa 5220 s procesorem Intel Celeron o frekvenci 1.6 MHz s 2 GB paměti RAM.

2. Inteligentní teplotní čidlo DS1822

2.1. Popis teplotního čidla DS1822

DS1822 je digitální teploměr, pomocí kterého lze měřit teplotu s přesností na 9 až 12 bitů, a disponuje funkcemi, které zaznamenávají překročení stanoveného teplotního limitu tzv. poplachové funkce. Tyto poplachové funkce umožňují nastavení mezních limitů. **DS1822** komunikuje s okolím přes **1-Wire** sběrnici. **1-Wire** sběrnice vyžaduje připojení pouze přes jednu datovou linku (a uzemnění) a touto datovou linkou je spojena s centrálním mikroprocesorem (v našem případě PC). Teplotní rozsah je stanoven na -55 ˚C až +125 ˚C, s přesností + - 2 ˚C v rozsahu -10 ˚C až +85 ˚C. **DS1822** umožňuje tzv. parazitní napájení, což znamená, že je napájeno přes datovou linku, přičemž nepotřebuje externí napájení.

Každé **DS1822** má unikátní 64bitové sériové číslo, které umožňuje vícenásobné použití teplotních čidel na **1-Wire** sběrnici. Tento způsob identifikace usnadňuje práci mikroprocesoru při řízení a obsluze ostatních zařízení připojených na tutéž datovou linku. Digitální teploměr **DS1822** se používá pro měření teplot uvnitř i vně budov, ve strojírenství a při ostatních kontrolních činnostech.

2.2. Detailní popis pinů

Číslo pinu IO	Pořadí	Označení	Popis
5	1	GND	Uzemnění
4	2	DQ	Datově vstupně výstupní pin. Zprostředkovává také napájení v parazitním módu.
3	3	V_{DD}	Volitelný V_{DD} pin. V_{DD} pin, musí být propojen s pinem GND v parazitním módu

Tabulka 2. Detailní popis pinů

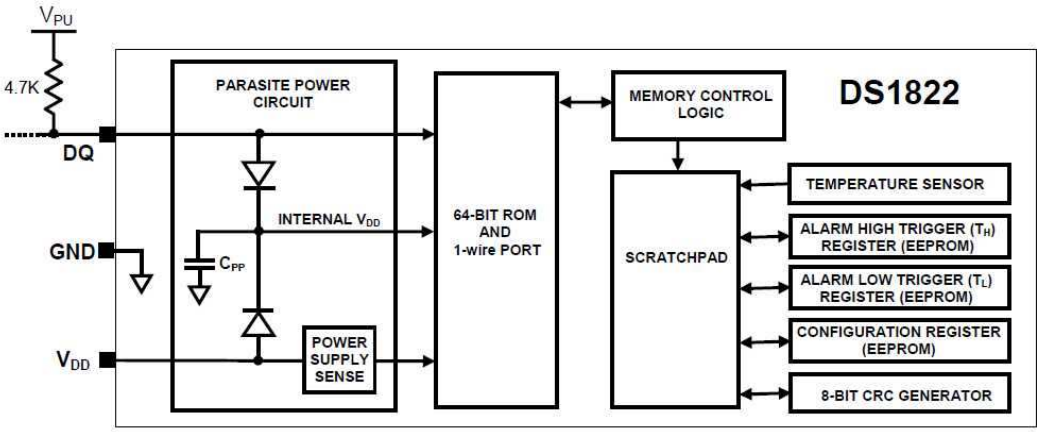
2.3. Celkový přehled

Obrázek 1 znázorňuje blokový diagram **DS1822** včetně popisu pinů, které jsou popsány v tabulce 1. 64bitová paměťová banka uchovává unikátní sériové číslo. Paměť **Scratchpad** obsahuje dvoubajtové teplotní registry, které obsahují naměřené teplotní hodnoty z teplotního čidla.

Dále pak paměť **Scratchpad** zprostředkovává přístup do 1. bajtu horního a dolního poplachového registru i do 1. bajtu konfiguračního registru. Konfigurační registr umožňuje nastavit úroveň přesnosti měřené teploty, a to v rozsahu 9-12 bitů. Do registrů T_H a T_L, jsou zapisována data, je-li digitální teploměr vypnut. **DS1822** využívá protokol **1-Wire** sběrnice vyvinutý firmou DALAS, která komunikuje prostřednictvím jednoho signálu. Připojená teplotní čidla využívají otevřený port na sběrnici nebo tří-stavový kanál (**DQ**). Na tomto sběrnicovém systému identifikuje **Master** adresy ostatních **Slave** zařízení, která jsou připojená na sběrnici a která obsahují unikátní 64bitový sériový kód. Díky jedinečnému sériovému kódu zařízení **Slave** lze na sběrnici připojit libovolné množství těchto zařízení.

Hlavním rysem **DS1822** je schopnost pracovat bez externího napájení. Toto napájení je zprostředkováno z **1-Wire** sběrnice prostřednictvím pinu datové linky (**DQ**). Je-li sběrnice ve stavu **High**, dochází k nabití kondenzátoru, který napájí teplotní čidlo v případě, že dojde k poklesu napětí na sběrnici. Tento způsob se nazývá tzv. parazitní napájení. Jako alternativní způsob napájení je možné zvolit napájení teplotního čidla prostřednictvím externího zdroje přes pin **VDD**.

2.4. Blokové schéma DS1822



Obrázek 1. Blokové schéma DS1822

2.5. Operace měření teploty

Hlavní funkcí **DS1822** je digitálně měřit teplotu. Přesnost měření teploty lze uživatelsky nastavit prostřednictvím 9. až 12. bitu, které korespondují s 0,5 ◦C; 0,25 ◦C; 0,125 ◦C a 0,0625 ◦C teplotními rozsahy. Výchozí nastavení teplotního senzoru je 12 bitů.

DS1822 má při nečinném stavu na sběrnici nízké napětí. K měření teploty se využívá AD převodník a **Master** k tomuto převodu používá příkaz Convert T [44h]. Po vyvolání tohoto příkazu se ukládají převedené teplotní hodnoty do 2bajtového teplotního registru v paměti **Scratchpad,** a **DS1822** se vrací nazpět do nečinného stavu. V případě, že je **DS1822** napájeno z externího zdroje, potom může Master přímo číst hodnoty z časového slotu. Jakmile je vyvolán příkaz pro převod teploty, vrací **DS1822** po celou dobu převodu hodnotu 0. Jakmile je teplotní převod dokončen, vrací hodnotu 1. Při parazitním způsobu napájení se tento způsob převodu nedá využít, protože je sběrnice ve stavu **High**.

Výstupní teplotní data jsou přímo kalibrovaná ve stupních Celsia. Pro převod na stupně Fahrenheita se musí použít konverzní funkce. Hodnoty naměřené teplotním senzorem jsou uloženy jako dvojice znaménkových bitů v teplotním registru. Znaménkový bit (S) indikuje stav teploty plus nebo minus. Pro teploty nad 0 ◦C je zde uložen stav 0 a pro teploty pod 0 ◦C je zde uložen stav 1. Je-li nastavena přesnost měření teploty na 12 bitů, budou všechny bity teplotního registru obsahovat platná data. Pro 11bitovou přesnost nebude 0. bit definován. Dále pak pro 10bitovou přesnost nebude definován 0. a 1. bit a pro 9bitovou přesnost nebude definován 0., 1. a 2. bit. V tabulce 2 je uveden příklad digitálního výstupu dat, při nastavené úrovni přesnosti měřených dat na 12 bitů.

2.6. Formát teplotního registru

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LS Byte	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
MS Byte	S	S	S	S	S	2^6	2^5	2^4

Obrázek 2. Formát teplotního registru

2.7. Vazba teploty a dat

TEPLOTA	DIGITÁLNÍ VÝSTUP (Binárně)	DIGITÁLNÍ VÝSTUP (Hexadecimálně)
+125.0000˚C	0000 0111 1101 0000	07D0h
+85.0000˚C	0000 0101 0101 0000	0550h
+25.0625˚C	0000 0001 1001 0001	0191h
+10.1250˚C	0000 0000 1010 0010	00A2h
+0.5000˚C	0000 0000 0000 1000	0008h
0.0000˚C	0000 0000 0000 0000	0000h
-0.5000˚C	1111 1111 1111 1000	FFF8h
-10.1250˚C	1111 1111 0101 1110	FF5Eh
-25.0625˚C	1111 1110 0110 1111	FE6Fh
-55.0000˚C	1111 1100 1001 0000	FC90h

Tabulka 3. Vazba teploty a dat

2.8. Operace signalizace alarmu

Pro operaci převodu teploty jsou naměřené hodnoty porovnávané s uživatelsky definovanými hodnotami, které jsou uloženy v 1. bajtu poplachového registru T_H a T_L. Znaménkový bit (S) obsahuje příznak „0“ pro teplotu vyšší než 0 ◦C a příznak „1“ pro hodnotu nižší než 0 ◦C. Registry T_H a T_L, jsou uloženy v paměti NV (EEPROM). Tyto registry lze nastavit prostřednictvím 2. a 3. bajtu v paměti **Scratchpad**.

Pouze 11 bitů ze čtyř teplotních registrů je použito v registrech T_H a T_L a jsou porovnávány jako 8bitový registr. Je-li naměřená teplota nižší nebo popřípadě vyšší než hranice uživatelského poplachu, potom je nastaven příznak poplachu. Při každém dalším měření teploty je tento příznak znovu nastavován. **Master** může příkazem **Alarm Search** [ECh] zjistit, která teplotní čidla **DS1822** mají nastavena příznak poplachu. **Slave**, který má nastaven příznak poplachu, odpovídá na toto volání.

2.9. Formát registru T_H a T_L

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

Obrázek 3. Formát teplotního registru T_H a T_L

2.10. Napájení čidla

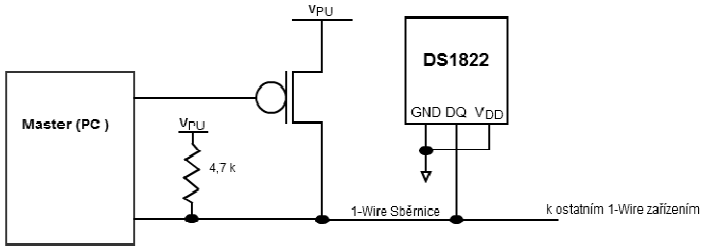
DS1822 může být napájeno z externího zdroje prostřednictvím pinu **VDD** nebo také tzv. parazitním módem. Parazitní mód je velice užitečná věc pro měření teplot na delší vzdálenost nebo v místech, kde není dostatek místa pro externí napájení. Obrázek 1 ukazuje parazitní způsob napájení **DS1822** prostřednictvím pinu **DQ** připojeným na sběrnici **1-Wire**. V případě, že je **DS1822** v parazitním módu, musí být pin **VDD** připojen na pin **GND**.

DS1822 vyžaduje, aby pro operace konverze teploty a převodu dat byla sběrnice ve stavu **High** maximálně 10 μs. Ostatní operace nemají stanovenou hranici.

DS1822 může být také napájeno konvenční metodou, jak je znázorněno na obrázku 5. Použití **DS1822** v parazitním módu se nedoporučuje pro teploty přesahující 100 ◦C, jelikož **DS1822** není schopno plnohodnotně komunikovat po sběrnici. Při teplotách nad 100 °C se doporučuje použít externí napájení.

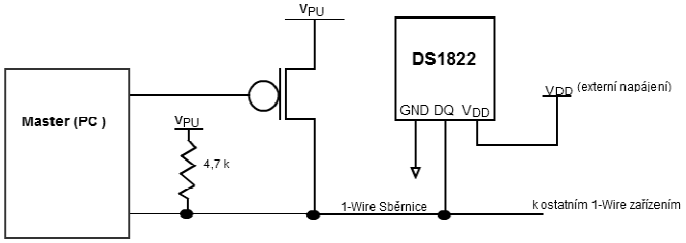
Při teplotní konverzi musí **Master** vědět, jakým způsobem je zařízení **Slave** napájeno. Během čtení dat z časového slotu při parazitním způsobu napájení je sběrnice ve stavu **Low,** a jiné zařízení **Slave,** které je externě napájeno, může využít tuto sběrnici. V opačném případě, je-li sběrnice ve stavu **Low** a **Master** má informaci o tom, že je zařízení **Slave** napájeno z datové linky, nelze do doby teplotní konverze komunikovat s jiným zařízením **Slave**.

2.10.1. Způsoby napájení DS1822



Obrázek 4. Nahrazení parazitního napájení během teplotní konverze

2.10.2. Napájení externím zdrojem

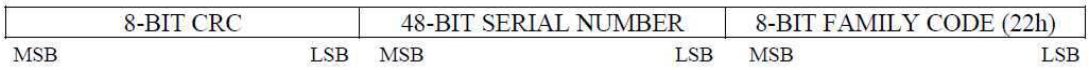


Obrázek 5. Napájení externím zdrojem

2.11. 64bitový identifikační kód

Každé zařízení **DS1822** obsahuje unikátní 64bitový kód (znázorněn na obrázku 6), který je uložen v paměti **ROM**. Posledních 8 bitů tohoto kódu se označuje za tzv. rodinný kód (skupina teplotních čidel stejného druhu) - v našem případě 22h. Dalších 48 bitů obsahuje unikátní sériové číslo. Prvních 8 bitů obsahuje cyklický redundantní součet (**CRC**) bajtů, který je vypočítáván z prvních 56 bitů paměti **ROM**. 64bitový **ROM** kód a přidružené kontrolní logické funkce umožňují, aby zařízení **DS1822** pracovalo přes **1-Wire** sběrnicový protokol.

2.11.1. Popis 64bitového identifikačního kódu



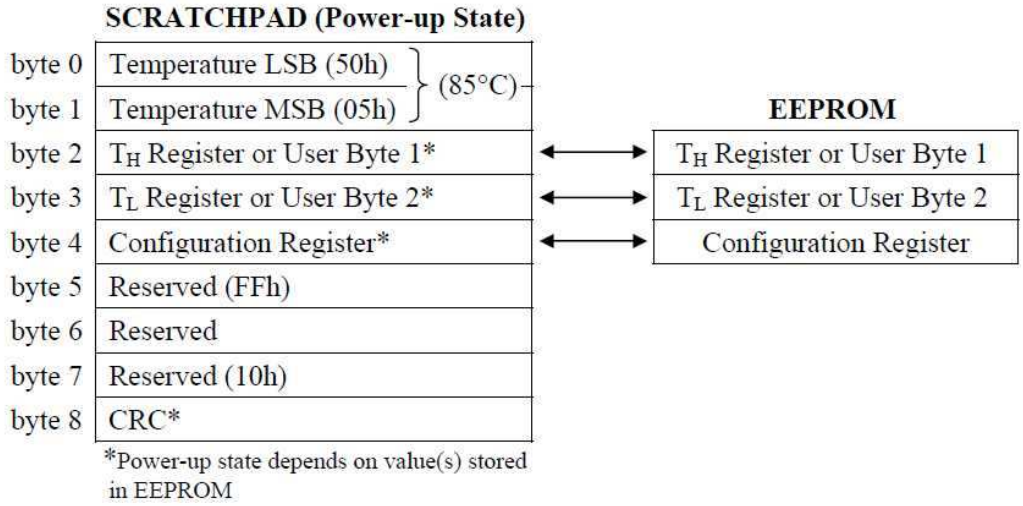
Obrázek 6. Popis 64bitového identifikačního kódu

2.12. Paměť

DS1822 má organizovanou paměť, jak je znázorněno na obrázku 7. Tato paměť je složena ze **SRAM Scratchpad,** který obsahuje **UV EEPROM -** úložiště poplachových registrů T_H a T_L a konfiguračního registru. V případě že nejsou poplachové funkce využity, může být tato paměť použita jako univerzální paměť. Bajty 0 a 1 ve vnitřní paměti **Scratchpad** obsahují příslušné **LSB** a **MSB** teplotní registry. Tyto bajty jsou pouze pro čtení. Bajty 2 a 3 zprostředkovávají přístup do T_H a T_L registrů. 4. bajt obsahuje data konfiguračního registru. Bajty 5, 6 a 7 jsou rezervované pro vnitřní použití zařízením a nelze je přepisovat. 8. bajt je pouze pro čtení a obsahuje **CRC** kód složený z bajtů 0 až 7.

Data jsou zapisována do 2. až 4. bajtu v paměti Scratchpad, skrze operaci **Write Scratchpad** [4Eh]. Data se zapisují od 2. bajtu do 4. bajtu. Datová integrita je při zápisu zajištěna a data je možné pouze číst prostřednictvím operace **Read Scratchpad** [BEh]. Při čtení dat z paměti **Scratchpad** jsou data zasílána prostřednictvím sběrnice **1-Wire**, počínaje nejmenším bitem 0. bajtu. Přenos (T_H a T_L registru a konfiguračního registru) dat z paměti **Scratchpad** do paměti **EEPROM** vyvolává **Master** prostřednictvím příkazu **Copy Scratchpad** [48h].

2.12.1. Mapa paměti teplotního čidla



Obrázek 7. Mapa paměti teplotního čidla

2.12.2. Konfigurační registr

4. bajt ve vnitřní paměti **Scratchpad** obsahuje konfigurační registr, který je znázorněn na obrázku 8. Přesnost měření může **Master** nastavit prostřednictvím tohoto registru. Používají se k tomu bity R0 a R1, jak je to vidět v tabulce 3. Při zapnutí teplotního čidla jsou nastaveny automaticky tyto hodnoty v registrech. Bity R0 a R1 jsou shodně nastaveny na logickou hodnotu 1 (což odpovídá 12bitovému rozlišení). Je to z důvodu přesnosti měření a časem konverze. Bit 7 a bity 0 až 4 v konfiguračním registru jsou vyhrazeny pouze pro vnitřní potřebu **DS1822** a nelze je měnit.

2.12.3. Popis konfiguračního registru

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

Obrázek 8. Popis konfiguračního registru

2.12.4. Popis nastavení teplotního rozlišení

R1	R0	rozlišení	Max. čas konverze
0	0	9 bitů	93.75ms
0	1	10 bitů	187.75ms
1	0	11 bitů	375.75ms
1	1	12 bitů	750.75ms

Tabulka 4. Popis nastavení teplotního rozlišení

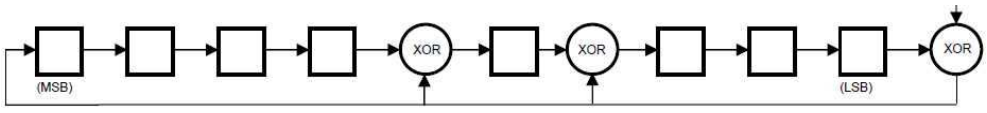
2.12.5. CRC – kontrolní součet

Kontrolní součet **CRC** je uložen v 9. bajtu paměti **Scratchpad** a obsahuje 64bitové unikátní číslo (**ROM** kód). **CRC** se vypočítá z prvních 56 bitů **ROM** kódu a je uložen v nejvyšším bajtu **ROM**. **CRC** paměti **Scratchpad** se vypočítává z dat uložených v téže paměti. Je to proto, že při změně dat se zároveň mění i tento kód. **CRC** kód, který obdrží **Master**, je vlastně potvrzením, že data která čte, jsou validní. Pro ověření správnosti dat **Master** opětovně vypočítá **CRC** kód z přijatých dat a poté porovná výsledek s **CRC ROM** pamětí (pro čtení z **ROM** paměti) nebo s pamětí **Scratchpad** (pro čtení ze **Scratchpad** paměti). Jsou-li shodné **CRC** kódy, potom jsou přijatá data bez chyb. **Master** jako jediný porovnává **CRC** kódy. V případě že se neshodují, pokračuje v činnosti nebo tuto činnost přeruší. Žádné schéma uvnitř **DS1822** neporovnává rovnost **CRC.** Výpočet **CRC** probíhá dle této polynomické funkce:

$$CRC = X_8 + X_5 + X_4 + 1$$

Master může přepočítávat **CRC** kód a porovnávat ho s hodnotou **CRC** v **DS1822**. Využívá přitom polynomického generátoru, jak je tomu na obrázku 9. Tento obvod se skládá z posunu registru a XOR brány. Tento registr je při inicializaci nastaven na 0. Přičemž se začíná nejmenším bitem paměti **ROM** nebo nejmenším bitem z bajtu 0. paměti **Scratchpad**. V jeden okamžik smí být vložen jeden bit do tohoto registru, kde jsou ukládány tyto hodnoty. Po posunutí 56 bitů paměti **ROM,** nebo po posunutí posledního bitu 7. bajtu v paměti **Scratchpad**, bude generátor **CRC** obsahovat výsledný **CRC** kód. Dalších 8 bitů paměti **ROM** nebo **Scratchpad** z **DS1822** musí být posunuto zpět do okruhu. V tomto bodě, je-li výpočet **CRC** korektní, bude registr, který uchovává posunuté bity, obsahovat pouze samé nuly.

2.12.6. CRC generátor



Obrázek 9. CRC generátor

3. 1-Wire sběrnicový systém

1-Wire sběrnicový systém je používán jako prostředek, kterým zařízení **Master** ovládá jedno/více **Slave** zařízení. **DS1822** je vždy **Slave** zařízení. Pokud je připojeno pouze jedno **Slave** zařízení na sběrnici, tak jej systém označuje výrazem **SingleDrop**. V případě připojení více zařízení **Slave** na sběrnici systém tato zařízení označuje výrazem **Multidrop**.

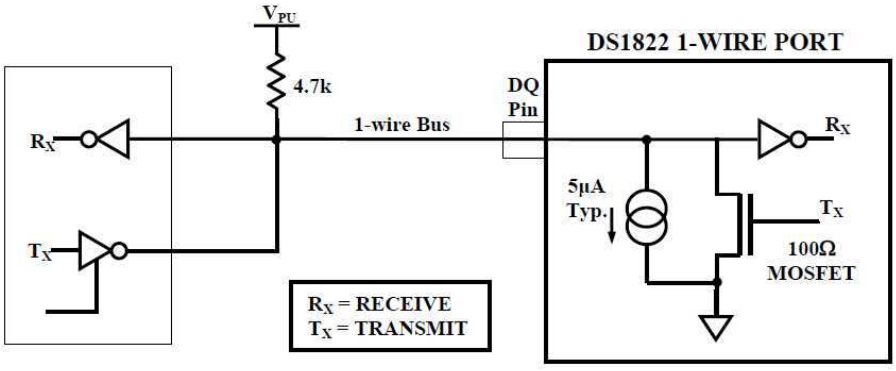
Všechna data a příkazy jsou přenášeny nejmenším prvním bitem přes **1-Wire** sběrnici.

3.1. Hardwarová konfigurace

1-Wire sběrnice má definovanou pouze jednu datovou linku. Každé zařízení **Master**/**Slave** je připojeno rozhraním k datové lince přes otevřený kanál nebo tří-stavový port. Tento způsob připojení umožňuje každému zařízení uvolnit datovou linku pro ostatní zařízení v případě, že se neprovádí přenos dat a sběrnice je volná pro připojení dalšího zařízení. **1-Wire** port **DS1822** (pin **DQ**) je otevřený kanál s vnitřním okruhem, jak je zobrazeno na obrázku 10.

1-Wire sběrnice je vybavena externím odporem. Je to z toho důvodu, že při nečinnosti **1-Wire** sběrnice, by byla ve stavu **High.** V případě že dojde k přerušení transakce, sběrnice opětovně použije tento odpor ke snížení napětí. Nekonečný obnovovací čas mezi bity může trvat tak dlouho, že **1-Wire** sběrnice bude v nečinném stavu během obnovovací periody. Používá-li se zařízení **Slave** déle než 480 μs, bude všem zařízením **Slave**, která jsou připojena na sběrnici, zaslán reset pulz.

3.1.1. Hardwarová konfigurace



Obrázek 10. Popis hardwarové konfigurace

3.2. Posloupnost kroků transakce

Posloupnost kroků transakce pro přístup k **DS1822** je následující:

- 1. inicializace
- 2. příkaz **ROM** (následují všechny požadované výměny dat)
- 3. **DS1822** funkční příkaz (následují všechny požadované výměny dat)

Je velmi důležité dodržovat pokaždé tento postup pro přístup k **DS1822**. V případě že nebude dodržen uvedený postup pro přístup k **DS1822,** a uvedené kroky budou zpracovány v jiném pořadí, nebo nebudou-li vůbec zpracovány všechny výše uvedené kroky, bude zařízení mimo provoz. Výjimku z těchto pravidel tvoří příkaz **Search ROM** [F0h] a **Alarm Search** [ECh]. Je-li vyvolán příkaz **ROM**, který je uveden ve výjimce, **Master** musí vrátit odpověď z kroku 1.

3.2.1. Inicializace

Všechny transakce na **1-Wire** sběrnici musí začínat inicializačním krokem. Inicializační krok se skládá z reset pulzu, který zasílá **Master** po sběrnici zařízením **Slave**, která vracejí prezenční pulz. Přítomnost prezenčních pulzů oznamuje, že jsou na sběrnici připojeny zařízení **Slave**, která jsou připravena k provozu.

3.2.2. Příkazy ROM

Poté co **Master** obdrží prezenční pulzy, může použít některý z příkazu **ROM**. Tyto příkazy pracují s 64bitovým unikátním kódem (identifikátorem), kterým je každé zařízení **Slave** vybaveno a umožňuje jednoznačně identifikovat konkrétní zařízení **Slave** mezi všemi, která jsou připojena na sběrnici **1-Wire**. Tyto příkazy umožňují zařízení **Master** určit, kolik je přítomných zařízení **Slave** a jakého typu jsou tyto zařízení **Slave** na sběrnici, nebo zařízení **Slave**, které překročilo programovatelnou mez teplotního alarmu. Příkazů **ROM** je 5 a každý příkaz se skládá z 8 bitů. Zařízení **Master** musí vyvolat vhodný příkaz **ROM** před použitím **DS1822** funkčního příkazu.

Popis příkazů ROM:

Vyhledej - **SEARCH ROM:** Je-li systém uveden do provozu, musí poté **Master** identifikovat **ROM** kódy všech zařízení typu **Slave**, která jsou připojena na sběrnici. **Master** tak získá informaci o počtu a typu zařízení, která jsou připojena na sběrnici. **Master** zná **ROM** kódy (které získal prostřednictvím eliminačního procesu), které zprostředkovává **SEARCH ROM** cyklus, kdykoliv je nutné identifikovat všechny zařízení typu **Slave**. Po každém cyklu, ve kterém se volá příkaz **SEARCH ROM**, musí zařízení **Master** vrátit krok 1 (inicializace).

Čti paměti - **READ ROM:** Tento příkaz smí být použit pouze v případě, že je na sběrnici připojeno pouze jedno zařízení typu **Slave**. **Master** bude vždy používat 64bitové číslo **ROM** zařízení **Slave** a nikdy nepoužije příkaz **SEARCH ROM**. Je-li tento příkaz použit, když bude na sběrnici připojeno více zařízení typu **Slave**, dojde ke kolizi, která bude mít za následek, že všechna zařízení budou odpovídat ve stejný čas.

Porovnej paměti - **MATCH ROM:** Příkaz **MATCH ROM**, kterému je předán 64bitový identifikační kód **Slave** zařízení, umožňuje zařízení **Master** získat přímou adresu **Slave** zařízení na **Single-Drop** nebo **Multidrop** na sběrnici. Pouze zařízení **Slave**, které má stejný **ROM** kód, odpoví zařízení **Master** a všechna ostatní čekají na reset pulz.

Přeskoč paměť - **SKIP ROM:** **Master** může tento příkaz použít současně pro všechny zařízení typu **Slave**, aniž by zasílal **ROM** kódové informace. Tento příkaz slouží pro odvolání transakce.

Vyhledej poplach - **ALARM SEARCH:** Operace tohoto příkazu je stejná jako operace **SEARCH ROM**, ale pouze zařízení typu **Slave**, která mají nastavený příznak **ALARM**, odpoví prezenčním pulzem. Ostatní zařízení očekávají reset pulz. Tento příkaz umožňuje zařízení **Master** získat informaci o tom, u kterých zařízení typu **Slave** došlo k překročení mezního limitu, stanoveného jako vyvolání poplachu. Po každém **ALARM SEARCH** cyklu vrátí **Master** krok 1 (inicializace) z posloupnosti kroků transakce.

3.2.3. DS1822 funkční příkazy

Poté co **Master** použil příkaz **ROM** a získal adresu zařízení, se kterým chce komunikovat, smí použít jeden z funkčních příkazů. Tyto příkazy umožňují zařízení **Master** zapisovat a číst z vnitřní paměti (**Scratchpad**) zařízení **DS1822**. **Master** smí také spouštět převod teploty a číst údaj o stavu napájení zařízení **Slave**.

Popis DS1822 funkčních příkazů:

Konverze: Tento příkaz spustí jeden převod teploty. Po konverzi jsou výsledná data uložena ve dvou bajtovém teplotním registru uvnitř vnitřní paměti **Scratchpad,** a **DS1822** se vrátí do svého nízkonapěťového módu tj. nečinnosti. Je-li zařízení provozováno v parazitním módu, a provedení převodu trvá déle než 10 μs, musí **Master** zastavit jakákoliv zpracování dalších příkazů, dokud nebude převod ukončen. Je-li zařízení napájeno z externího zdroje, potom může **Master** pokračovat ve čtení časového slotu a zařízení **DS1822** bude po dobu provádění konverze odpovídat stavem 0. Stavem 1 odpoví, jakmile bude konverze dokončena.

Zápis do paměti – **Scratchpad:** Tento příkaz umožňuje zařízení **Master** zapisovat trojici bajtů dat do vnitřní paměti **Scratchpad** zařízení **DS1822**. První datový bajt je zapsán do registru T_H (2. bajt). Druhý bajt je zapsán do registru T_L (3. bajt) a třetí bajt je zapsán do konfiguračního registru (4. bajt). Všechny tři bajty musí být zapsány dříve, než zařízení **Master** vyvolá reset pulz, jelikož by data mohla být znehodnocena.

Čtení paměti – **Scratchpad:** Tento příkaz umožňuje číst obsah vnitřní paměti **Scratchpad**. Přenos data začíná nejmenším bitem z bajtu 0 a končí 9. bajtem (bajt 8 je vyhrazen pro **CRC**). Zařízení **Master** může kdykoliv vyvolat reset pulz, aniž by přečetl celý obsah vnitřní paměti **Scratchpad**, pokud požaduje pouze některá data z paměti **Scratchpad**.

Kopírování z paměti **Scratchpad** do **EEPROM:** Tento příkaz zkopíruje obsah registrů T_H , T_L a konfigurační registry (bajty 2, 3 a 4) v paměti **Scratchpad** do paměti **EEPROM**.

Kopírování z paměti **EEPROM** do **Scratchpad** (**Recall**): Tento příkaz kopíruje obsah registrů T_H , T_L a konfigurační registry z paměti **EEPROM** a nahrazuje data v bajtech číslo 2, 3 a 4 příslušné paměti **Scratchpad**. Zařízení **Master** může číst časový slot během operace **Recall**, avšak zařízení **Slave** bude po dobu kopírování vracet příznak 0, dokud překopírování nebude ukončeno. Poté vrátí příznak 1. Operace **Recall** se automaticky vyvolá při každém připojení zařízení **Slave** ke sběrnici.

Stav napájení **Master,** po vyvolání tohoto příkazu, provádí čtení časového slotu u zařízení **Slave**, které je v parazitním módu.

3.2.4. Přehled funkčních příkazů

Command	Description	Protocol	1-Wire Bus Activity After Command is Issued	Notes
TEMPERATURE CONVERSION COMMANDS				
Convert T	Initiates temperature conversion.	44h	DS1822 transmits conversion status to master (not applicable for parasite-powered DS1822s).	1
MEMORY COMMANDS				
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS1822 transmits up to 9 data bytes to master.	2
Write Scratchpad	Writes data into scratchpad bytes 2, 3, and 4 (T_H, T_L, and configuration registers).	4Eh	Master transmits 3 data bytes to DS1822.	3
Copy Scratchpad	Copies T_H, T_L, and configuration register data from the scratchpad to EEPROM.	48h	None	1
Recall E^2	Recalls T_H, T_L, and configuration register data from EEPROM to the scratchpad.	B8h	DS1822 transmits recall status to master.	
Read Power Supply	Signals DS1822 power supply mode to the master.	B4h	DS1822 transmits supply status to master.	

NOTES:

1. For parasite-powered DS1822s, the master must enable a strong pullup on the 1-Wire bus during temperature conversions and copies from the scratchpad to EEPROM. No other bus activity may take place during this time.
2. The master can interrupt the transmission of data at any time by issuing a reset.
3. All three bytes must be written before a reset is issued.

Obrázek 11. Přehled funkčních příkazů

3.3. Signalizace na 1-Wire sběrnici

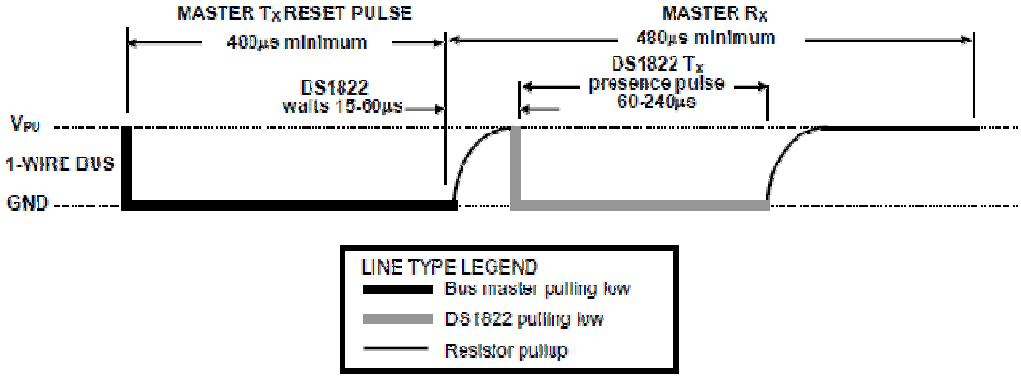
DS1822 používá striktně pouze **1-Wire** komunikační protokol, aby si byl jist datovou integritou. Existuje několik dohodnutých signálních typů daných protokolem. Mezi nimi jsou: reset pulz (nulování), presence pulz (oznámení přítomnosti), write 0 (zápis), write 1 (zápis), read 0 (čtení), read 1 (čtení). Všechny tyto signály (s výjimkou presence pulzu) inicializuje **Master**.

3.3.1. Inicializační procedura: nulování přítomnostních pulzů

Veškerá komunikace s **DS1822** začíná inicializační sekvencí, která seskládá ze dvou pulzů. Nulovacího (reset) pulzu od zařízení **Master** a prezenčního (presence) pulzu jako odpověď od **DS1822**. (Jak je tomu znázorněno na obrázku 12.) Když **DS1822** pošle prezenční (reset) pulz jako odpověď na nulovací (reset) pulz, pro zařízení **Master** je to jednoznačný signál, že zařízení **Slave** je připraveno pracovat.

Během inicializační sekvence **Master** zasílá (*Tx*) nulovací (reset) pulz tak, že je sběrnice ve stavu **Low** po dobu minimálně 480 μs. **Master** poté uvolní (**Release**) sběrnici a přejde do tzv. přijímacího módu (*Rx*).
Když **DS1822** detekuje tuto prudkou změnu napětí, posečká ještě 15 až 60 μs a poté odpoví přítomnostním pulzem tak, že uvede sběrnici do stavu **Low** po dobu 60 až 240 μs.

3.3.2. Časování inicializační procedury



Obrázek 12. Časování inicializační procedury

3.3.3. Čtení a zápis časového slotu

Master zapisuje data do **DS1822** skrze zapisovací časový slot (**Write time slot**) a čte data z **DS1822** přes čtecí časový slot (**Read time slot**). Jeden bit dat je poslán přes **1-Wire** sběrnici přes časový slot.

3.3.4. Zápis do časového slotu

Existují dva typy časových slotů. **Write 1** časový slot a **Write 0** časový slot. **Master** používá **Write 1** časový slot k zapsání logické 1 a **Write 0** časový slot k zapsání logické 0 v **DS1822**. Všechny zápisové časové sloty musí být minimálně 60 μs dlouhé s minimální 1 μs mezerou mezi zapisováním do časového slotu. Oba typy zapisovacích slotů jsou inicializované poklesem napětí na sběrnici.

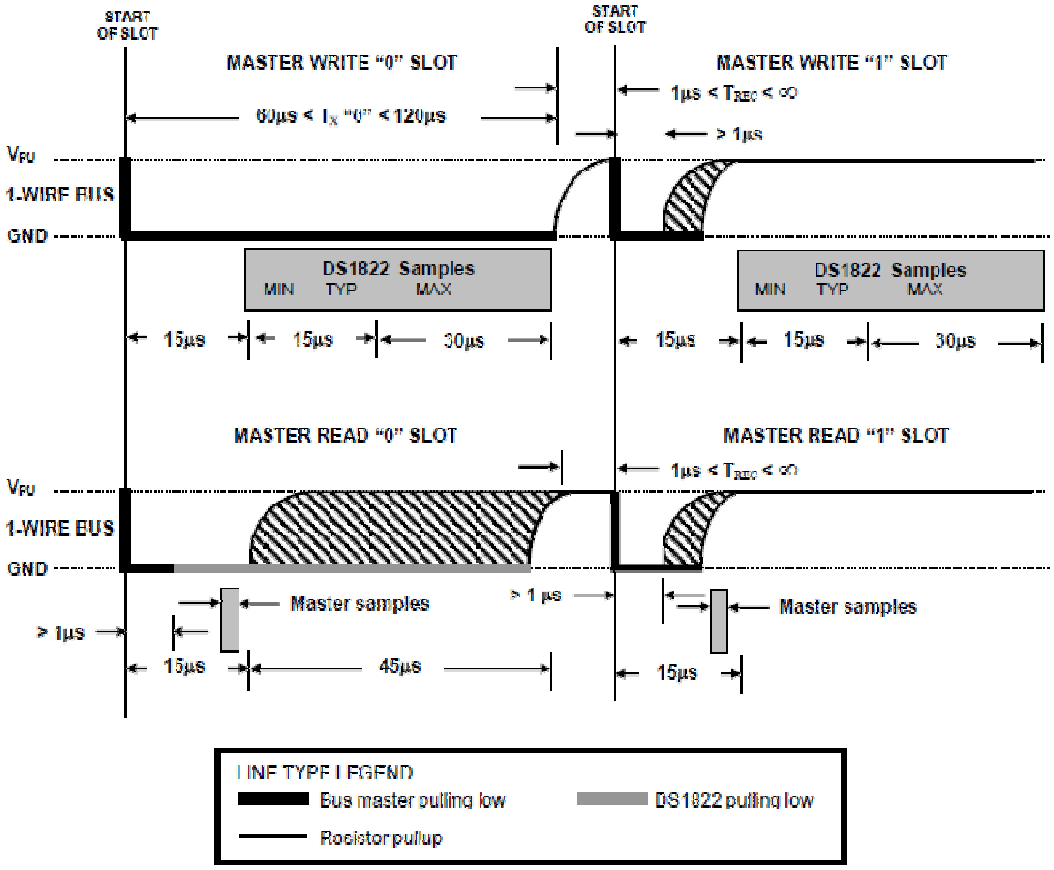
Pro zápis **Write 1** do časového slotu se sběrnice přepne do stavu, kdy je ve stavu **High,** a to 15 μs po inicializaci. V případě že **Master** uvolní sběrnici, poté odpor podrží sběrnici ve stavu **High**. Pro zápis **Write 0** do časového slotu musí **Master** podržet sběrnice ve stavu **Low** nejméně dalších 60 μs.

3.3.5. Čtení časového slotu

Master přijímá data od zařízení **Slave** pouze v případě, že čte časový slot. **Master** smí číst časový slot pouze v případě, že použil příkaz **Read Scratchpad** [BEh], **Read Power Supply** [B4h], **Convert** [44h] nebo **Recall E2** [B8h] (jestliže není v parazitním módu).

Všechny časové sloty musí být minimálně 60 μs dlouhé s minimální 1 μs mezerou mezi čtením časového slotu. Čtení slotu inicializuje **Master** tak, že na dobu 1 μs uvede sběrnici do stavu **Low,** a poté ji opětovně uvolní. (Jak je znázorněno na obrázku 13.) Poté co **Master** inicializuje časový slot, bude **DS1822** po sběrnici posílat 1 nebo 0. 1 zašle tím, že nechá sběrnici ve stavu **High,** a 0 tak, že nechá sběrnici ve stavu **Low.** Jestliže **DS1822** posílá 0, tak se na konci časového slotu uvolní (**Release**) sběrnice, čímž bude opětovně ve stavu **High**. Výstup dat z **DS1822** je validní po uplynutí 15 μs od změny stavu sběrnice po zahájení čtení časového slotu.

3.3.6. Diagram čtení a zápisu časového slotu



Obrázek 13. Časování inicializační procedury

3.3.7. Příklady operací s teplotním čidlem

Příklad 1: Na sběrnici je připojeno více **DS1822** teplotních senzorů, které jsou napájeny parazitním způsobem. **Master** inicializuje převod teploty u specifického čidla **DS1822** a provede přepočet **CRC** k ověření platnosti dat. Viz tabulka 4.

MASTER MÓD	DATA (LSB PRVNÍ)	KOMENTÁŘ
TX	Reset	Master vyšle Reset pulz
RX	Presence	DS1822 vrátí odpověď
TX	55h	Master vyvolává příkaz Match ROM
TX	64-bit ROM code	Master posílá DS1822 ROM kód
TX	44h	Master vyvolává příkaz Convert T
TX	DQ drží sběrnici pod napětím	Master drží sběrnici pod napětím po dobu konverze
TX	Reset	Master vyšle Reset pulz
RX	Presence	DS1822 vrátí odpověď
TX	55h	Master vyvolá příkaz Macth ROM
TX	64-bit ROM code	Master posílá DS1822 ROM kód
TX	BEh	Master vyvolá příkaz Read Scratchpad
RX	9 bajtů dat	Master přečte CRC ze Scratchpad, přepočítá CRC z prvních 8 datových bajtů a porovná s přečtenými 9 bajty. Jestliže se CRC rovnají, poté se pokračuje, v jiném případě se provede opětovné čtení CRC

Tabulka 5. Příklad 1 - operace s teplotním čidlem

Příklad 2: Na sběrnici je připojeno pouze jedno **DS1822** zařízení a je napájeno parazitním způsobem. **Master** provede zápis do registru T_H , T_L a konfiguračního registru v paměti **Scratchpad** v teplotním senzoru **DS1822,** a poté přečte opětovně paměť **Scratchpad**, dále přepočítá **CRC** k ověření platnosti dat. **Master** poté zkopíruje data do paměti **EEPROM**.

MASTER MÓD	DATA (LSB PRVNÍ)	KOMENTÁŘ
TX	Reset	Master vyšle Reset pulz
RX	Presence	DS1822 vrátí odpověď
TX	CCh	Master vyvolá příkaz Skip ROM
TX	4Eh	Master vyvolá příkaz Write Scratchpad
TX	3 datové bajty	Master pošle 3 bajty (T_H, T_L a konfiguraci) do Scratchpad
		Master vyšle Reset pulz
TX	Reset	DS1822 vrátí odpověď
RX	Presence	Master vyvolá příkaz Skip ROM
TX	CCh	Master vyvolá příkaz Read Scratchpad
TX	BEh	Master přečte CRC ze Scratchpad, přepočítá CRC z prvních 8 datových bajtů a porovná s přečtenými 9 bajty. Jestliže se CRC rovnají, poté se pokračuje, v jiném případě se provede opětovné čtení CRC
RX	9 bajtů dat	
		Master vyšle Reset pulz
		DS1822 vrátí odpověď
TX	Reset	Master vyvolá příkaz Skip ROM
RX	Presence	Master vyvolá příkaz Copy Scratchpad
TX	CCh	Master drží sběrnici pod napětím po dobu 10μs, dokud probíhá kopírování
TX	48h	
TX	DQ drží sběrnici pod napětím	

Tabulka 6. Příklad 2 - operace s teplotním čidlem

4. Aplikace pro zobrazení naměřených hodnot z DS1822

Hlavním cílem této webové aplikace je umožnit uživateli zobrazovat naměřené teploty v uživatelsky přívětivé formě. Naměřené teploty jsou zde znázorněny grafem nebo tabulkou. Aktuální teploty na čidlech se zobrazují pouze v grafické podobě. Aplikace dále umožňuje graficky znázornit maximální a minimální hranice dosažených teplot v časovém období. Aplikace umožňuje změnu časového období pro zobrazení dat, způsob zobrazení naměřených hodnot a také měnit typy grafů.

4.1. Popis použitých prostředků

Webová aplikace je tenko-klientní aplikace typu Klient-Server, která je naprogramovaná ve skriptovacím jazyce **PHP**. Ke svému běhu využívá webový prohlížeč, ve kterém se prostřednictvím webového serveru vykreslují jednotlivé stránky aplikace. Naměřené teploty jsou uloženy v databázi (**MySQL**), odkud si je aplikace vyčítá a následně zobrazuje. Javascriptový framework **jQuery** zprostředkovává asynchronní získávání dat pro zobrazení. Pro vykreslování grafů je použit framework pChart, který vytváří obrázky grafů dle zadaných vstupních dat.

4.2. Popis funkce aplikace

Po spuštění webového prohlížeče a zobrazení webové stránky aplikace se zobrazí v levé části stránky formulář (Filtr), kde se zadávají podmínky pro zobrazení naměřených teplot. Poté co jsou zadána kritéria pro zobrazení a je potvrzena volba, se asynchronním dotazem do databáze načtou potřebná data pro zobrazení naměřených hodnot dle zvolené volby (graf, tabulka, nebo aktuální teplota).

4.3. Použité prostředky

Při prvním načtení stránky se vykreslí pouze formulář (Filtr) a to tak, že klient (webový prohlížeč) požaduje po webovém serveru (**Apache**) zobrazení stránky a předá mu adresu (URL) požadované stránky. Webový server vykoná část programu, který obsahuje požadovaná stránka, a klientovi zašle odpověď (HTML). S filtrem dat lze volně pohybovat prostřednictvím metody „Drag and Drop“, kterou zprostředkovává javascriptový framework **jQuery**. Při potvrzení volby na formuláři se asynchronně prostřednictvím **jQuery** načítají data, případně s grafem na pozadí, aniž by došlo k překreslení celého obsahu webové stránky. Při této komunikaci se využívá technologie **Ajax**, která vyšle požadavek na webový server v jiném vláknu a očekává odpověď.

4.4. Uživatelský filtr

Uživatelský filtr slouží pro zadávání časového období, typu zobrazení a filtrování naměřených hodnot z teplotních čidel **DS1822**. Dále pak pomocí ukazatele stavu zprostředkuje uživateli vizuální informaci o tom, v jakém stavu je právě probíhající operace.

Obrázek 14. Uživatelský filtr

4.4.1 Časové období

Nastavení časového období slouží ke stanovení mezních podmínek pro zobrazení naměřených hodnot. Časové období se zadává prostřednictvím filtru, kde se vyplní datum a čas počátku a konce časového období. Časové období není omezeno. Pouze se doporučuje volit menší časové období z důvodu větší přehlednosti v grafu.

4.5. Zobrazení naměřených hodnot

Naměřené hodnoty se zobrazují buď ve formě grafu, tabulky, nebo popřípadě jako aktuální teploty z teplotních čidel. Tabulové zobrazení dat je v textové formě, graf je zobrazen v grafické formě jako obrázek. Aktuální hodnoty z teplotních čidel jsou pouze v grafické formě.

4.5.1. Zobrazení grafem

Při volbě zobrazení naměřených teplot pomocí grafu je graf zobrazen ve formě obrázku tak, že webový klient předá webovému serveru data z filtru, která byla zadána. Na straně serveru se poté zpracují vstupní parametry a vygeneruje se příslušný obrázek (pomocí pChar knihovny), který se vrátí na klienta, který ho zobrazí. S tímto obrázkem lze volně pohybovat po obrazovce, jelikož tento graf je dotčen metodou „Drag and Drop“.

4.5.2. Zobrazení tabulkou

Při volbě zobrazení naměřených dat tabulkou předá webový klient webovému serveru data z filtru, která byla zadána. Na straně serveru se poté zpracují vstupní parametry a na klienta se vrátí požadovaná data, která potom klient zobrazí ve formě tabulky.

4.5.3. Zobrazení aktuální teploty

Při volbě zobrazení aktuálních naměřených hodnot z teplotních čidel předá webový klient webovému serveru data z filtru, která byla zadána. Na straně serveru se poté zpracují vstupní parametry a na klienta se vrátí požadovaná data, která potom klient zobrazí ve formě obrázku.

4.6. Vytvoření grafu

Graf je vytvořen pomocí frameworku pChart. Na webovém serveru se začne zpracovávat programový kód, který vygeneruje graf. Nejprve jsou však zpracovávána data (z filtru), která byla zadána. Interpret jazyka PHP se poté dotáže databáze MySQL na data, a výslednou odpověď s daty uloží do lineární datové struktury pole (asociativní pole). Tímto způsobem se nadefinuje časové období s konkrétními naměřenými teplotami, které slouží jako podklad pro vytvoření grafu (osa x, jednotlivé body atd.). Dále se vyčítá nastavení grafu, a výsledkem je, že se na klienta vrátí odpověď. V hlavičce odpovědi se uvede poznámka, že se jedná o dokument typu obrázek.

4.7. Parametrizace grafu

Ve frameworku pChart lze snadno parametrizovat budoucí graf. Základem parametrizace jsou data, pomocí kterých se skrze **API** nastaví budoucí vzhled grafu. Parametrizace grafu se nastavuje ve filtru dat. Je velice jednoduché nakonfigurovat vlastní typ grafu.

4.8. Hlavní atributy grafu

Hlavní atributy grafu slouží k základnímu zobrazení grafu. Mezi tyto atributy patří:

- Pozadí grafu - jedná se o plátno, kde je vykreslena datová část grafu společně s pomocnými objekty (legenda, popisky os).
- Datová část - je místo, kde se vykresluje samotný graf (osy, křivky atd.).
- Nadpis - popisuje sémantiku grafu.
- Legenda - popisuje jednotlivé křivky.

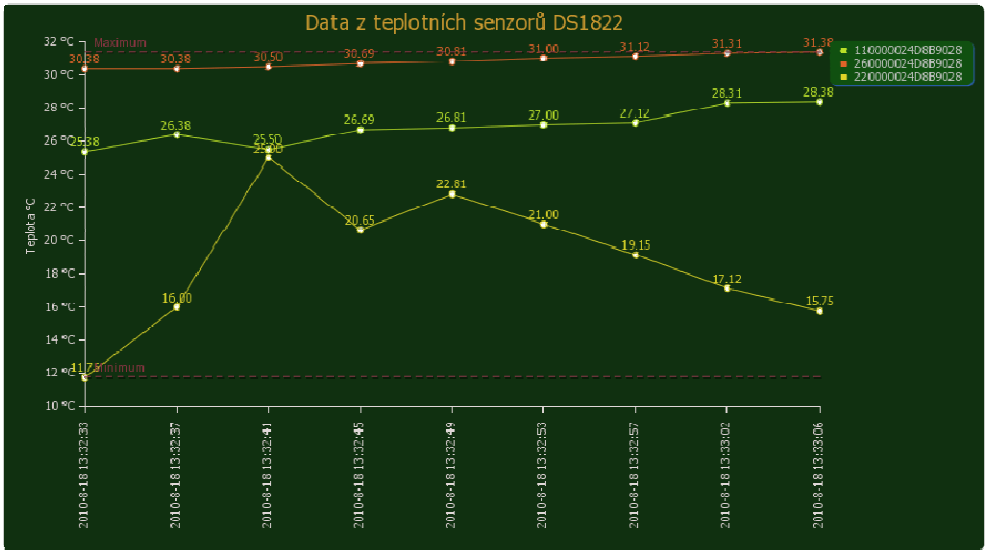
4.9. Typy grafu

Na příslušném formuláři (filtru) je možné vybrat určitý typ grafu, který má být vykreslen. Výčet typů grafů je následující:

- čárový graf
- graf s kubickou křivkou
- graf s vyplněnou kubickou křivkou
- graf maxima a minima

4.9.1. Čárový graf

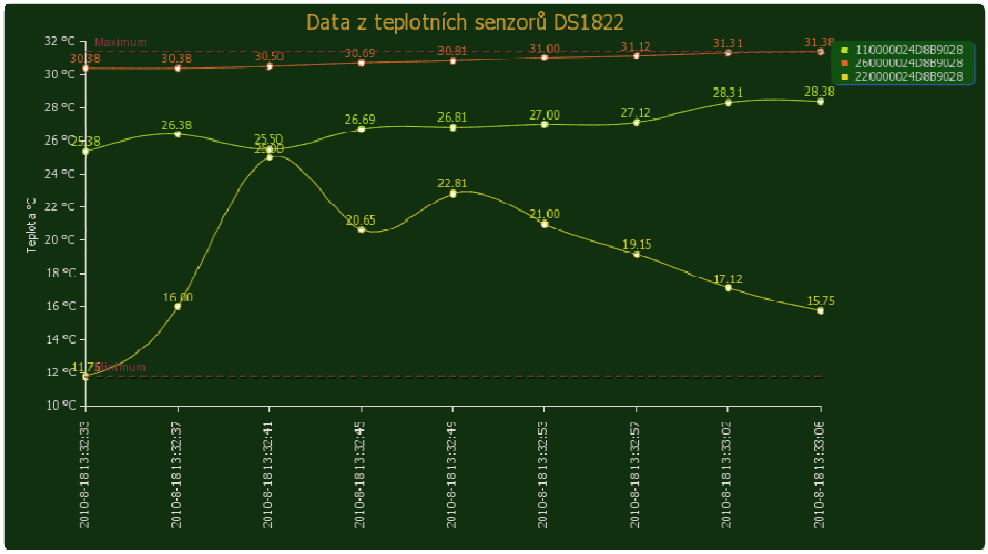
Čárový graf tvoří pouze přímé spojnice (úsečky) mezi definovanými body, v našem případě hodnotami teplot.



Obrázek 15. Čárový graf

4.9.2. Graf s kubickou křivkou

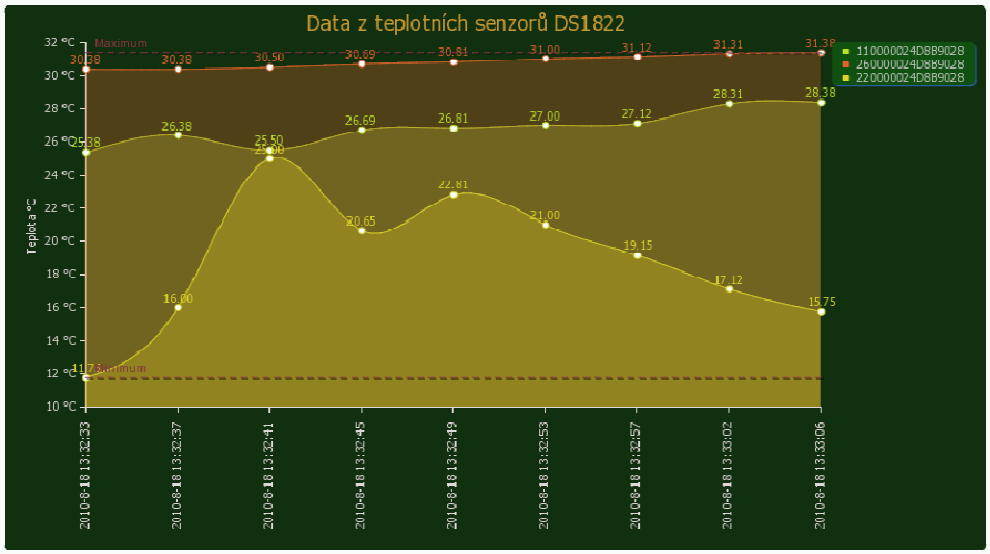
Graf s kubickou křivkou využívá interpolace křivek po obloucích.



Obrázek 16. Graf s kubickou křivkou

4.9.3. Graf s vyplněnou kubickou křivkou

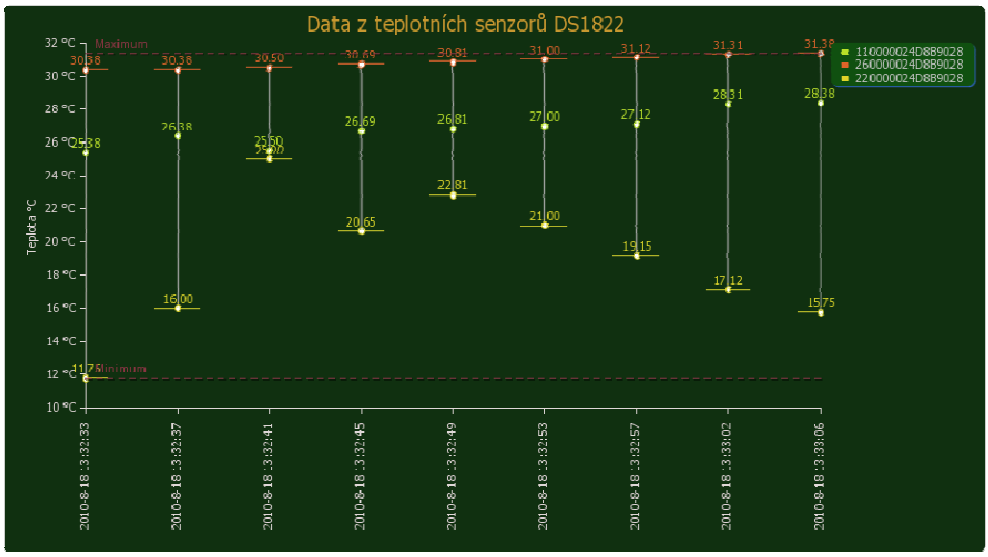
Tento graf využívá kubickou interpolační křivku, jako ohraničení oblasti pro výplň.



Obrázek 17. Graf s vyplněnou kubickou křivkou

4.9.4. Graf maxima a minima

Tento graf zobrazuje maximální a minimální hodnoty v určitém časovém intervalu.



Obrázek 18. Graf maxima a minima

4.10. Použité prostředky

4.10.1. Apache HTTP Server

Apache HTTP Server je softwarový webový server s otevřeným kódem pro GNU/Linux, BSD, Solaris, Mac OS X, Microsoft Windows a další platformy. V současné době dodává prohlížečům na celém světě většinu internetových stránek.

4.10.2. PHP Hypertext Preprocessor

PHP je skriptovací programovací jazyk určený pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktur jazyka HTML, XHTML či WML, což lze využít při tvorbě webových aplikací. **PHP** lze použít i k tvorbě konzolových a desktopových aplikací.

Skripty vytvořené v **PHP** jsou většinou prováděny na straně serveru a k uživateli je přenášen jen výsledek jejich činnosti (interpret **PHP** skriptu je možné volat i pomocí příkazové řádky). Syntaxe jazyka je inspirována několika programovacími jazyky (Perl, C, Pascal a Java). **PHP** je nezávislý na platformě a skripty fungují bez větších úprav na mnoha operačních systémech. **PHP** podporuje možnost rozšířit tento jazyk o knihovny, které se poté volají přímo ve skriptech. Jsou to například knihovny pro zpracování textu, grafiky, práci se soubory, přístup k většině databázových systémů (mj. **MySQL**, ODBC, Oracle, PostgreSQL).

4.10.3. Databáze MySQL

MySQL je databázový systém, vytvořený švédskou firmou, nyní však vlastněný společností Sun Microsystems, tj. dceřinou společností Oracle Corporation. **MySQL** se distribuuje pod bezplatnou licencí GPL.

MySQL je multiplatformní databáze. Komunikace s ní probíhá po- mocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními.

4.10.4. PHP Framework pChart

pChart je **PHP** třídově orientovaný framework. Mnoho dnešních nástrojů, které generují grafy, nedosahují kvality tohoto frameworku. Tento framework **pChart** je zcela zdarma a je k dispozici pod bezplatnou licencí GNU.

4.10.5. Javascriptový framevork jQuery

jQuery je lehká, malá javascriptová knihovna, která klade důraz na interakci mezi javascriptem a **HTML**. Byla vydána Johnem Resigem v lednu 2006 na newyorském BarCampu.
jQuery je svobodný a otevřený software a je k dispozici pod duální licencí MIT a GPL.

4.11. Požadavky na systém a software

- Windows XP 3SP a vyšší (doporučený Windows 7)
- Internet Explorer 7, 8, Firefox 6.0, Opera 10
- Apache HTTP Server
- Databáze MySQL
- PHP 5.2 a vyšší
- Framework pChart
- Framework jQuery 1.5.0

5. Aplikace pro sběr dat z čidel DS1822

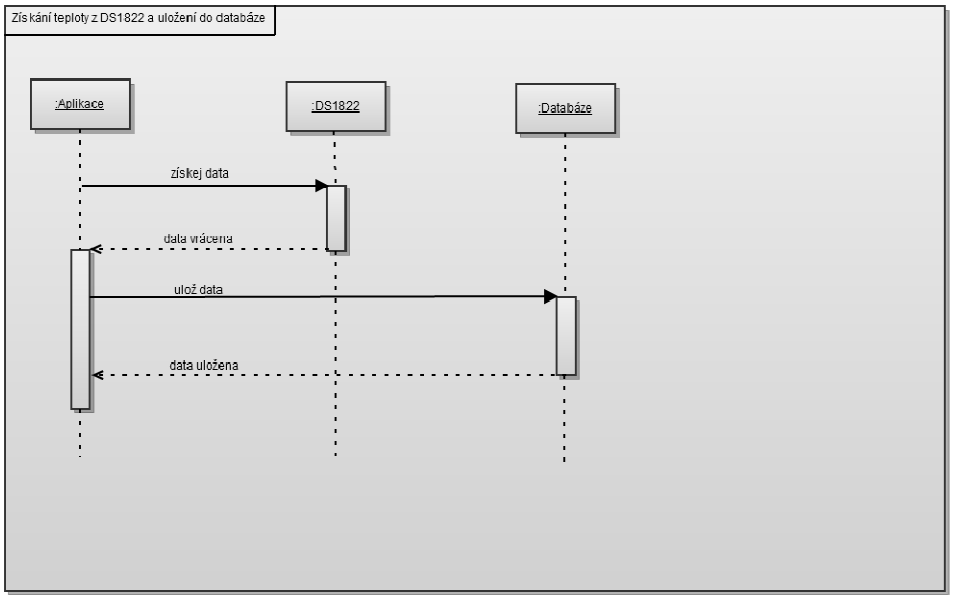
Hlavním cílem této aplikace je sběr dat z teplotních čidel **DS1822**, která jsou připojena k počítači prostřednictvím sériového portu (lze použít i převodník RS232-USB), a následně jsou tato data ukládána do databáze **MySQL**.

5.1. Popis použitých prostředků

Aplikace pro sběr dat z čidel **DS1822** je standalone aplikace, která je spuštěna v reálném čase. Tato aplikace je naprogramovaná v jazyku **C#** a využívá Microsoft Framework 2.0. Komunikace s teplotními čidly probíhá přes **1-Wire** sběrnici. Naměřené teploty z teplotních čidel se ukládají do databáze **MySQL**.

5.2. Princip sběru dat

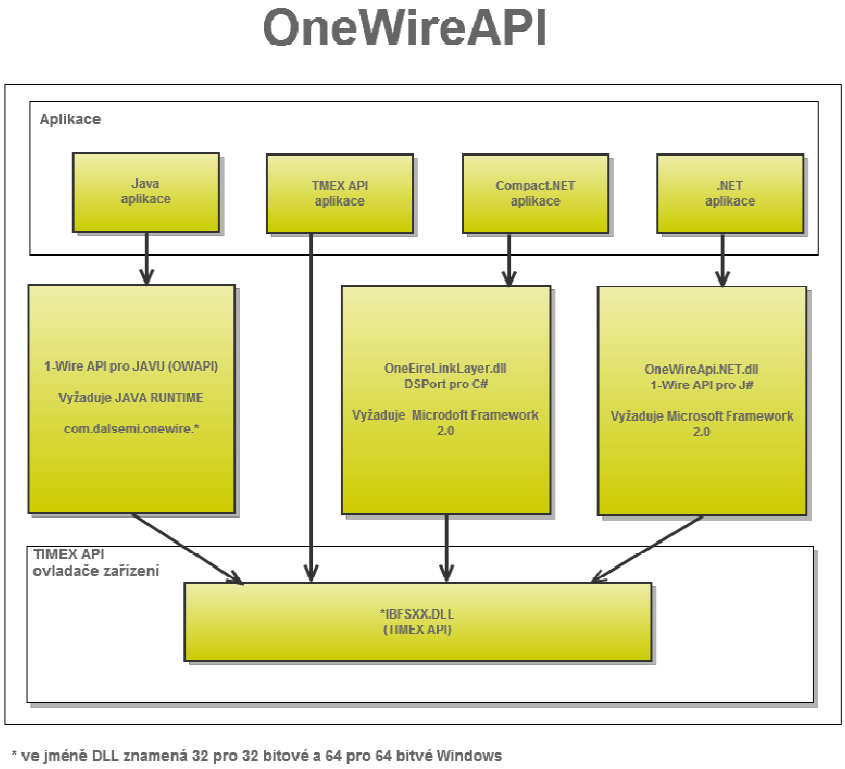
Po spuštění aplikace je v první řadě načten soubor, ve kterém jsou uloženy parametry s nastavením aplikace. Následně se tyto parametry zpracují a spustí se vlastní výkonný kód aplikace. V první řadě se naváže spojení s databází **MySQL**. Následně aplikace prověří, zda jsou ke sběrnici **1-Wire** připojena teplotní čidla **DS1822**. Jestliže je některé teplotní čidlo připojeno (odešle odpověď), provede aplikace odečet teploty z tohoto čidla a naměřenou hodnotu uloží do databáze.



Obrázek 19. Sekvenční diagram sběru a ukládání dat

5.3. Komunikace s teplotními čidly

Aplikace komunikuje s teplotními čidly prostřednictvím **1-Wire** sběrnice. Tato komunikace probíhá prostřednictvím rozhraní, které zprostředkovává komunikaci mezi zařízením **Master** (v našem případě PC) a zařízením **Slave** (v našem případě **DS1822**). Toto rozhraní je tvořeno ovladačem ”OneWireAPI.NET.dll”, který dodává výrobce teplotních čidel ve vývojovém prostředí. Princip je znázorněn na obrázku 20.



Obrázek 20. Rozhraní OneWireAPI

Komunikace je zahájena tzv. sezením (**SESSION**), které připojeným teplotním senzorům nastaví příznak tohoto sezení. Toto sezení je inicializováno prostřednictvím pulzu, který vyšle **Master** zařízením **Slave** po sběrnici **1-Wire**. Jestliže má teplotní čidlo nastaveno příznak sezení, poté komunikuje výhradně v tomto sezení. Následně probíhá komunikace se všemi teplotními čidly, přičemž se **Master** spojuje s jednotlivými zařízeními postupně. Při tomto sezení je prováděn sběr naměřených teplot na příslušném teplotním čidle, se kterým je **Master** v komunikaci. V případě, že je teplota odečtena z teplotního čidla, pokračuje se v odečtu u následujícího teplotního čidla.

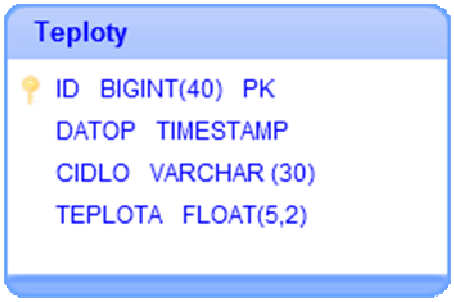
5.3.1. Sled komunikace s teplotními čidly

- sezení
- pulzu (RESET)
- pulzu (PRESENCE)
- dat
- sezení

5.4. Komunikace s databází **MySQL**

Naměřená data jsou ukládána do databáze **MySQL**. Komunikace probíhá prostřednictvím rozhraní, které je tvořeno ovladačem ”MySql.Data.dll”. V první řadě se provede navázání spojení s databází a toto spojení se otevře pro čtení a zápis. Po celou dobu běhu aplikace pro sběr dat se drží spojení s databází. V rámci tohoto spojení zde aplikace vykonává řadu SQL dotazů.

Naměřená data jsou ihned po přečtení z teplotního čidla uložena do databáze. Nežli se data uloží, musí se zahájit transakce pro případ, že během ukládání dat dojde k chybě. Jakmile je transakce zahájena, provede se vlastní uložení dat do tabulky prostřednictvím SQL příkazu INSERT. Tato transakce se potvrdí v případě, že vše proběhlo korektně, anebo se transakce odvolá a žádná změna nebude uložena v databázi. Potvrzením transakce se data stávají perzistentními.



Obrázek 21. ER diagram tabulky Teploty

5.5. Sled komunikace s databází **MySQL**

- zahájení transakce
- dat
- potvrzení či odvolání transakce

5.6. Požadavky na systém a software

- Windows XP 3SP a vyšší (doporučený Windows 7)
- ovladač teplotních čidel
- MySQL

6. Uživatelská dokumentace

6.1. Úvod

Tento dokument se zabývá detailním popisem webové aplikace pro zobrazení naměřených hodnot z DS1822 a aplikace pro sběr dat z čidel DS1822. Součástí této dokumentace je i instalace těchto aplikací. Tyto aplikace umožňují uživateli využívat teplotních čidel pro monitorování teplot v uživatelsky přívětivém prostředím.

6.2. Instalace webové aplikace pro zobrazení naměřených hodnot z DS1822

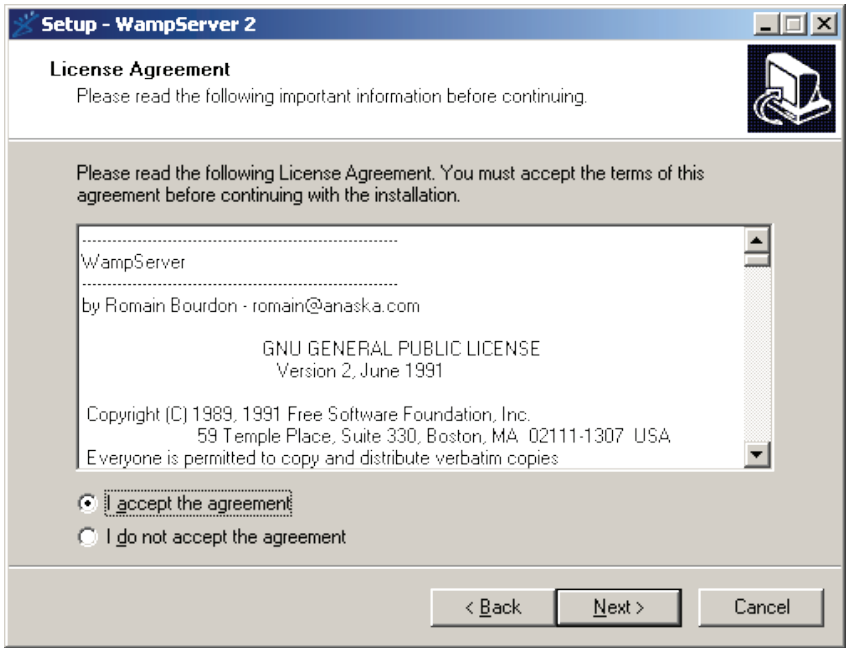
Instalace webové aplikace pro zobrazení naměřených hodnot z DS1822 vyžaduje pouze uživatelskou znalost PC. Instalace má několik kroků, které je zapotřebí dodrže a které nelze zaměňovat. Samotná instalace trvá přibližně 10 minut.

6.2.1. Instalace Apache,PHP, MySQL

1. spustíme instalaci souborem "WampServer2.0i.exe"

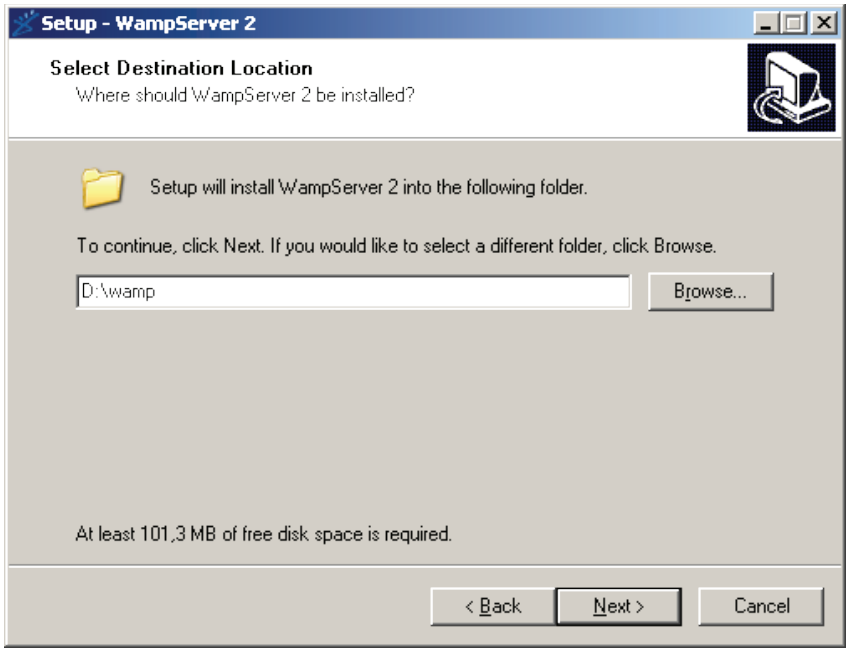
Obrázek 22. Start instalace

2. Potvrdíme licenční ujednání



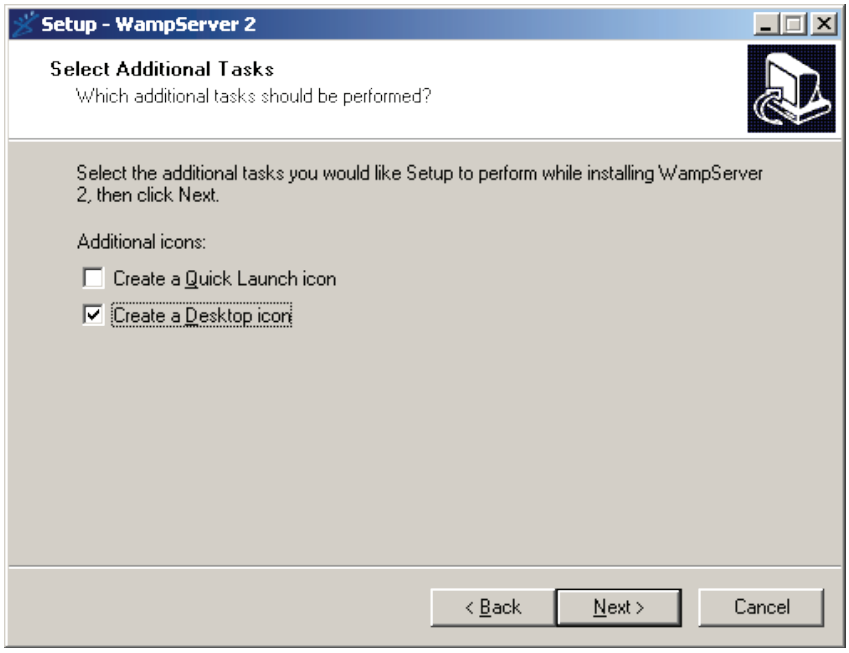
Obrázek 23. Licenční ujednání

3. Zadáme instalační adresář



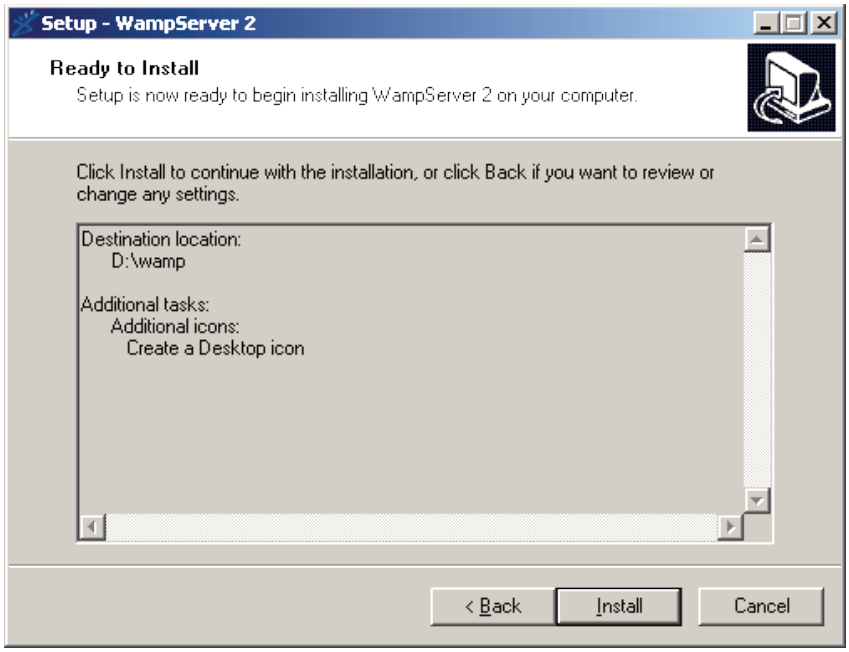
Obrázek 24. Zadání instalačního adresáře

4. Zvolíme možnost umístění zástupce



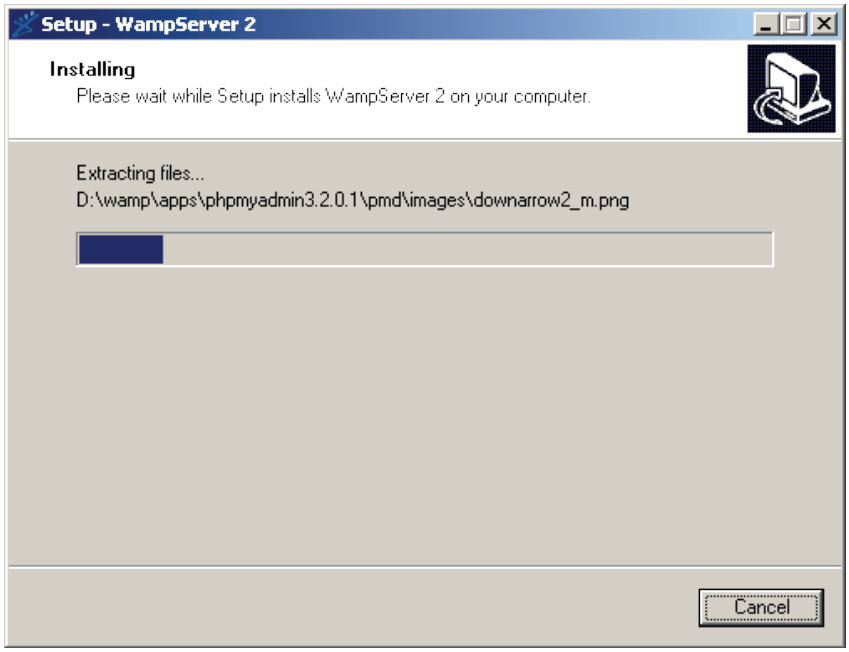
Obrázek 25. Umístění zástupce

5. Spuštění vlastní instalace



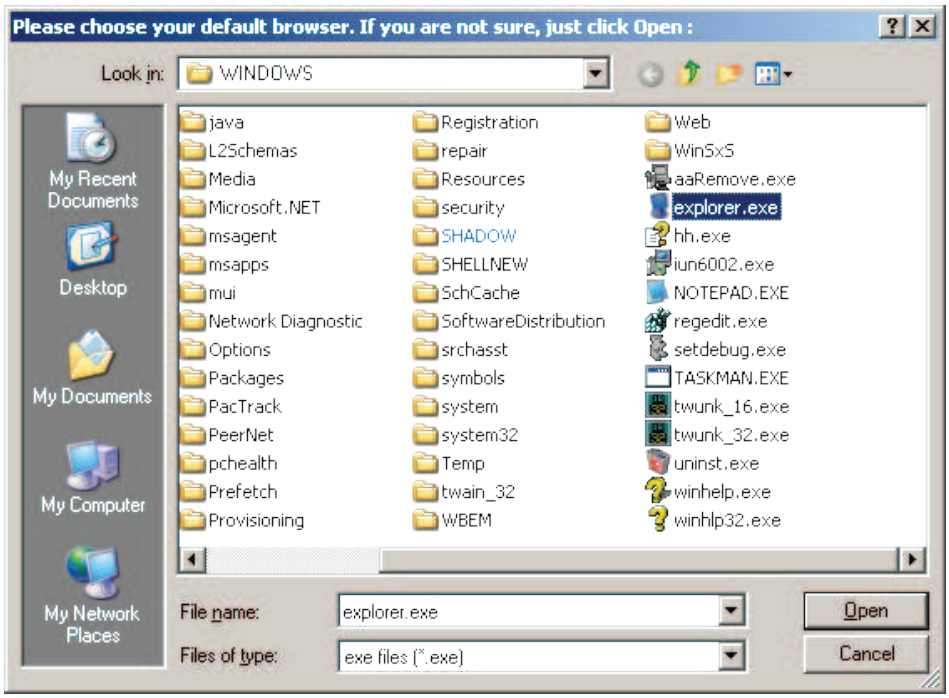
Obrázek 26. Vlastní instalace

6. Průběh instalace



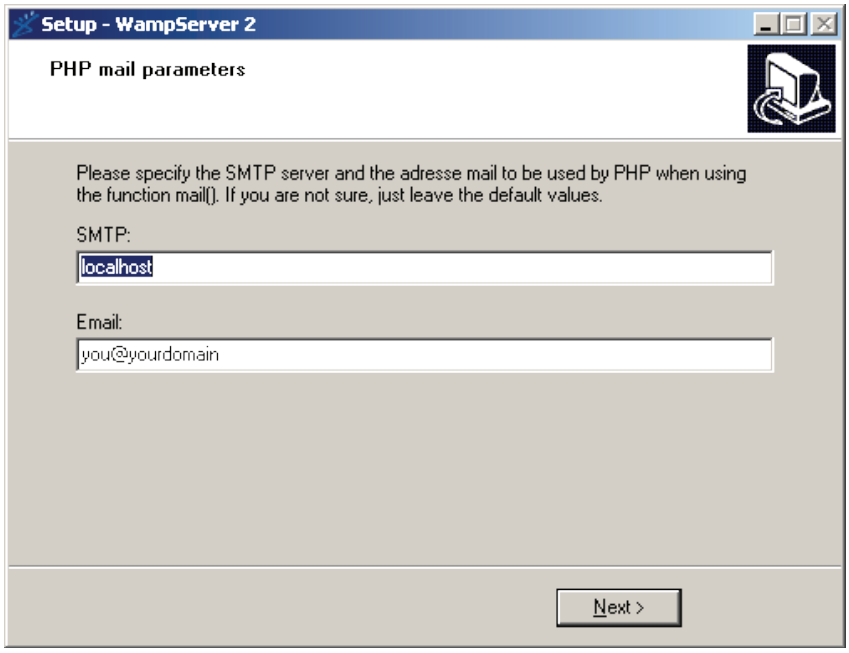
Obrázek 27. Průběh instalace

7. Volba výchozího prohlížeče



Obrázek 28. Volba výchozího prohlížeče

8. Nastavení protokolu SMTP a emailu

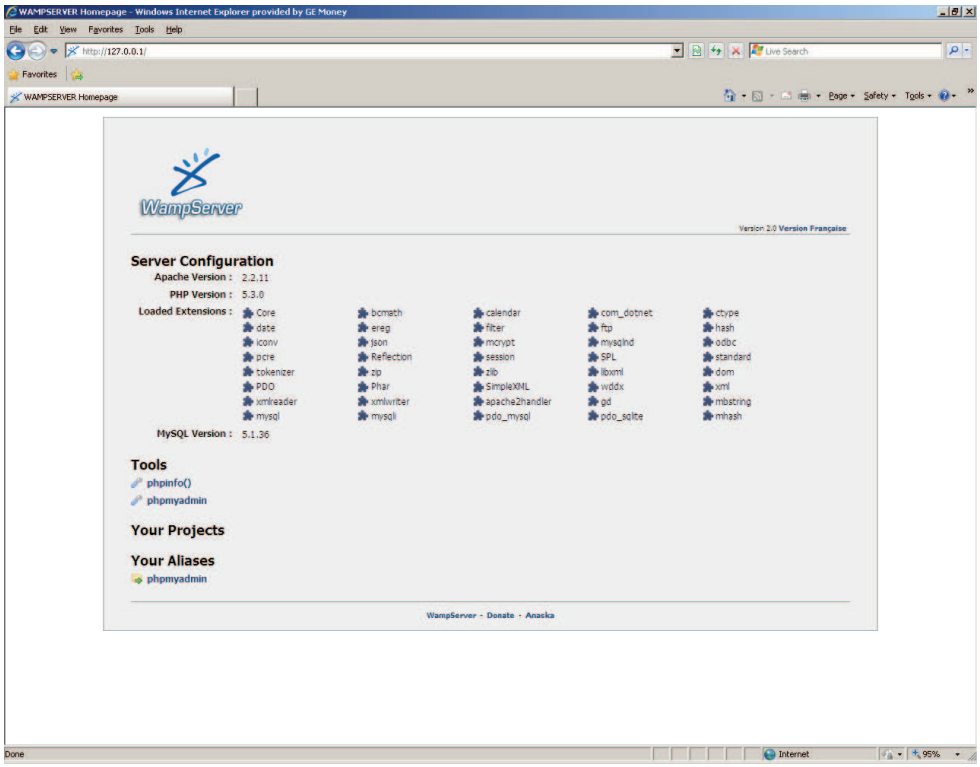


Obrázek 29. Nastavení protokolu SMTP a emailu

9. Dokončení instalace

Obrázek 30. Dokončení instalace

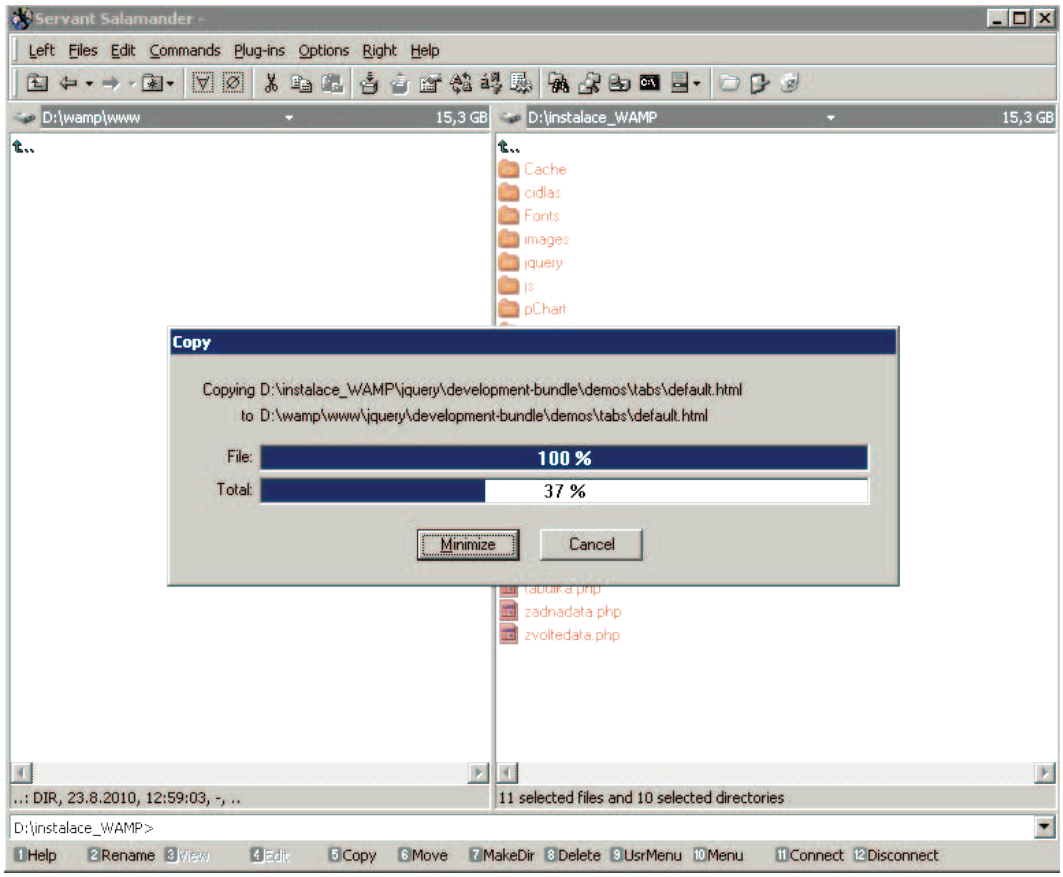
10. Ověření instalace



Obrázek 31. Ověření instalace

6.2.2. Kopírování souborů webové aplikace pro zobrazení naměřených hodnot z DS1822

Rozbalte soubor cidlas.zip a překopírujte do adresáře "D:/wamp/www".



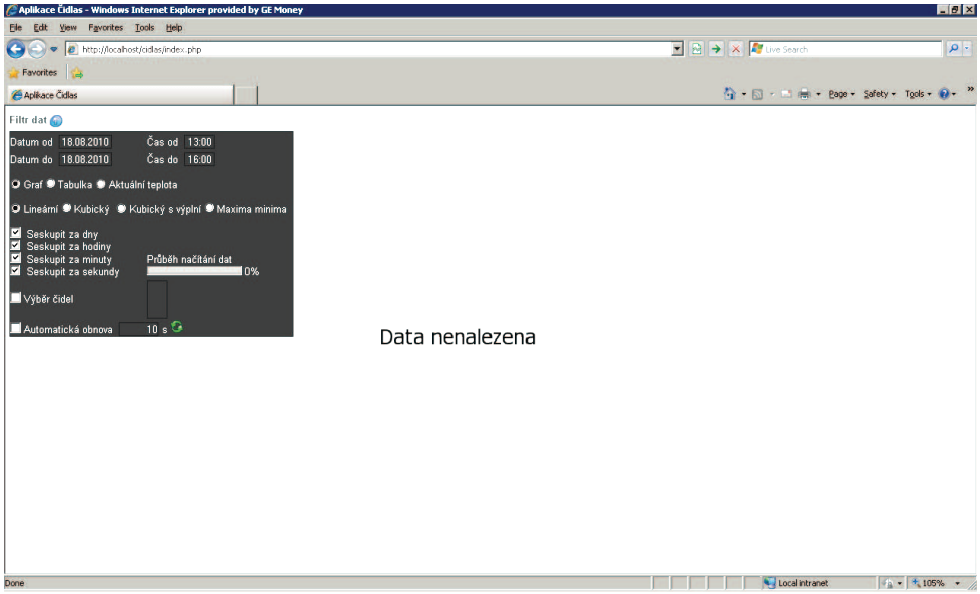
Obrázek 32. Kopírování souboru webové aplikace pro zobrazení naměřených hodnot z DS1822

6.2.3. Nastavení přístupu do databáze MySQL

Otevřete soubor z tohoto umístění "D:/wamp/www/cidlas/hlavicka.php". V případě že jste při instalaci databáze MySQL změnili umístění databáze, nebo chcete použít jinou databázi, změňte položku "$db_host" na adresu Vaší databáze. Jestliže-li jste změnili při instalaci MySQL jméno případně heslo, změňte položky "$db_user" a "$db_pass".

6.3. Spuštění webové aplikace pro zobrazení naměřených hodnot z DS1822

Spuštění aplikace se provede zadáním URL adresy do prohlížeče ve tvaru "http://localhost/cidlas/index.php"



Obrázek 33. Spuštění webové aplikace pro zobrazení naměřených hodnot z DS1822

6.3.1. Filtr dat

Filtr dat slouží ke sběru požadavků, kterými si uživatel volí způsob zobrazení naměřených hodnot. Uživatel má možnost zadat:

- časové období
- způsob zobrazení (graf, tabulka, nebo graficky(aktuální teplota))
- výběr typu grafu (lineární, kubický, kubický s výplní a maxima minima)
- seskupení naměřených hodnot (přesnější filtrace dat)
- zobrazení pouze vybraných čidel
- automatické obnovy

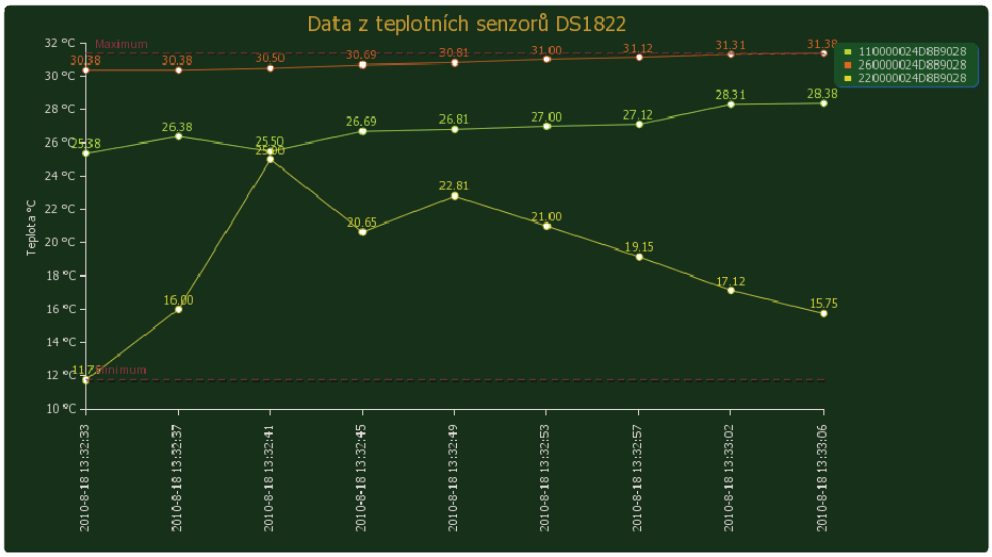
Zadání časového období - časové období je možné zadat dle předem definované masky pro datum a čas.

Maska pro datum má tvar DD-MM-YYYY.
- DD značí den v měsíci (01 - 28/29/30/31)
- MM značí měsíc v roce (01 - leden, 02 únor … 12 prosinec)
- YYYY značí rok (např. 2010)

Maska pro čas má tvar HH:MM.
- HH značí hodiny (00 - 24).
- MM značí minuty (00 - 59).

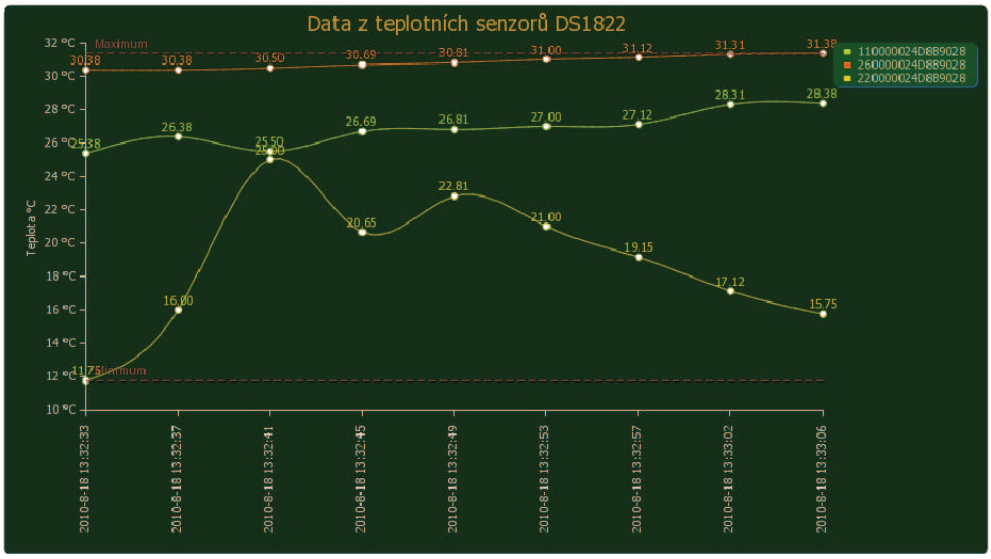
Způsob zobrazení naměřených hodnot - naměřené hodnoty je možné zobrazit:
- grafem(lineární, kubický, kubický s výplní, maxima minima)
- tabulkou
- graficky(aktuální teplota)

Čárový graf - tvoří pouze přímé spojnice(čáry) mezi definovanými body v našem případě hodnotami teplot.



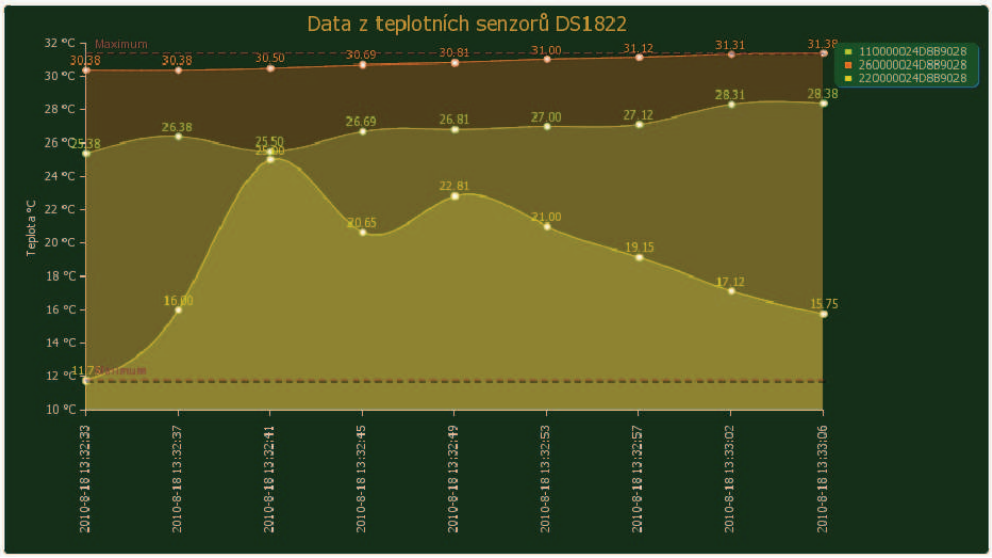
Obrázek 34. Čárový graf

Graf s kubickou křivkou - využívá interpolace křivek po obloucích.



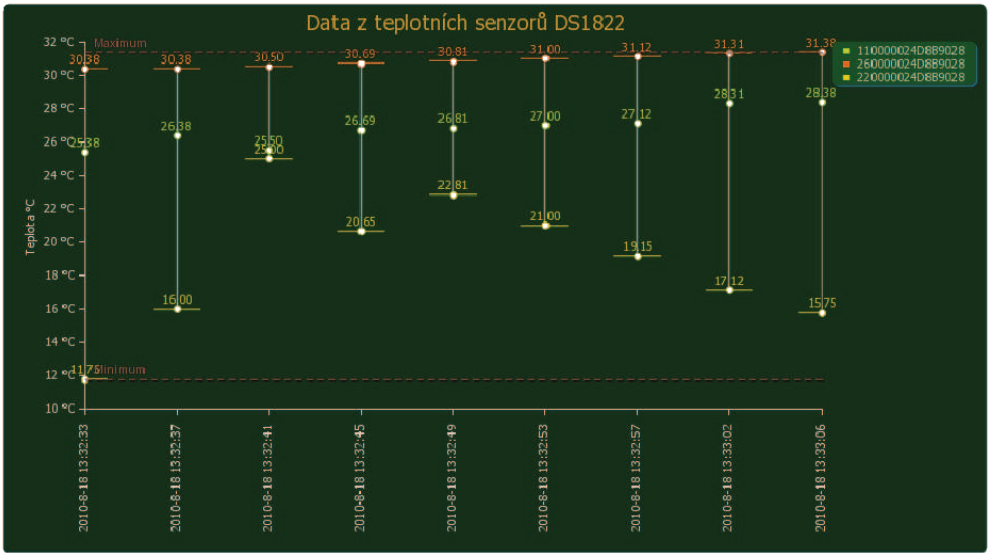
Obrázek 35. Graf s kubickou křivkou

Graf s vyplněnou kubickou křivkou - využívá kubickou interpolační křivku jako ohraničení oblasti pro vyplň.



Obrázek 36. Graf s vyplněnou kubickou křivkou

Graf maxima a minima - zobrazuje maximální a minimální hodnoty v určitém intervalu.



Obrázek 37. Graf maxima a minima

Zobrazení tabulkou - při volbě zobrazení tabulkou se zobrazí tabulka s požadovanými daty (datum, čidlo, teplota).

Zobrazení aktuální teploty - při volbě zobrazení aktuální teploty se graficky zobrazí aktuální teploty na vybraných čidlech.

Seskupení naměřených hodnot - se využívá především pro bližší filtraci dat tj. jestliže-li chceme zvýšit či snížit abstrakci nad naměřenými hodnotami. Např. z čidla 001 se zaznamenala teplota dne 10.10.2010 v 10:00:05 a z čidla 002 se zaznamenala teplota dne 10.10.2010 v 10:00:25.

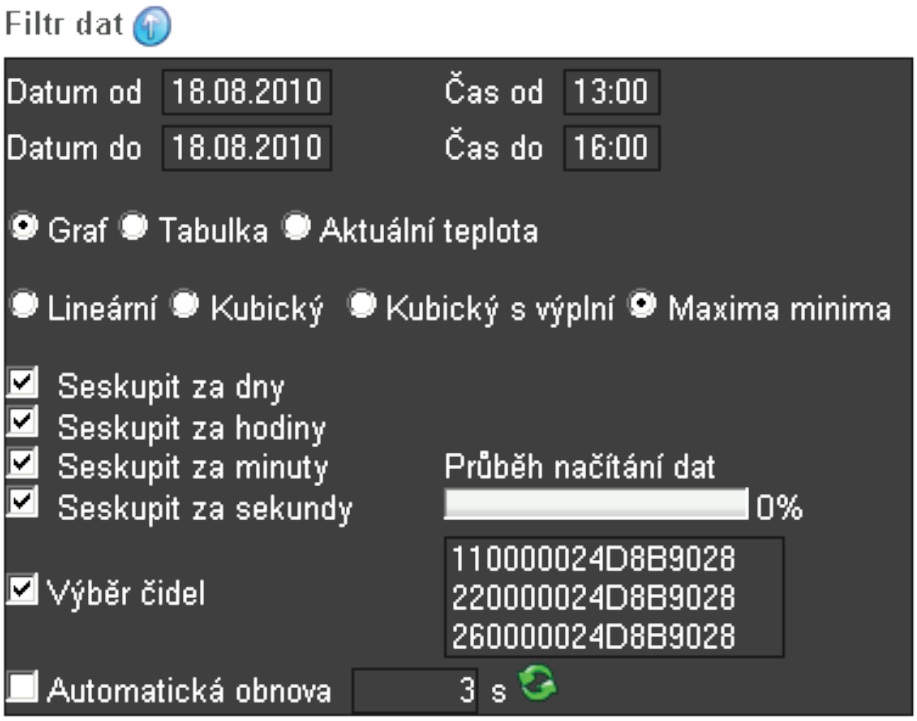
Chceme-li v grafu, nebo tabulce vidět tyto naměřené hodnoty ve stejný datum a čas bez sekund, provedeme zaškrtnutí za dny, hodiny a minuty. Poté se nám na časové ose zobrazí pouze 1x 10.10.2010 10:00 a v průsečíku osy x a osy y se zobrazí naměřené hodnoty z obou teplotních čidel. Analogicky je možné pokračovat pro dny, hodiny, minuty a sekundy a to v libovolné kombinaci.

Zobrazení pouze vybraných čidel - jestliže-li jsme si zobrazili graf, nebo tabulku, potom se nám naplnil seznam s čidly pro které jsme vyfiltrovali data. Následně můžeme dodatečně zpřísnit výběrové kriterium pro konkrétní čidla. Zaškrtnutím volby "Výběr čidel" a následným vybráním konkrétních čidel se při příštím zobrazení grafem, nebo tabulkou použije toto výběrové kriterium.

Automatická obnova - zaškrtnutím volby "Automatická obnova" se aktivuje nekonečný cyklus, který po určitém časovém období který je uživatelem

zadám v sekundách vyvolá překreslení naměřených hodnot. Vypnutí se provádí zrušením zaškrtnutí u této volby

Potvrzení volby - se provede po kliknutí myši na "zelené šipky". Poté následuje provedení požadované akce tj. v našem případě vykreslení dat požadovaným způsobem.



Obrázek 38. Uživatelský filtr

6.4. Požadavky na systém a software

- Windows XP a vyšší
- Apache HTTP Server
- PHP 5.0 a vyšší
- MySQL
- framework pChart, jQuery 1.5.0
- Internet Explorer 7, 8, Firefox 6.0, Opera 10

6.5. Instalace aplikace pro sběr dat z čidel DS1822

Instalace aplikace pro sběr dat z čidel DS1822 vyžaduje pouze uživatelskou znalost PC. Instalace má několik kroků, které je zapotřebí dodržet a které nelze zaměňovat. Samotná instalace trvá pouze 5 minut.

6.5.1. Instalace ovladače čidla DS1822

Spustíme instalaci ovladače čidla prostřednictvím souboru "install_1_wire_drivers_x86_v403beta.msi". Dále postupujte dle instrukcí na obrazovce.

6.5.2. Kopírování souborů

Rozbalte soubor "meric.zip" a obsah souboru zkopírujte na disk (zvolte si umístění).

6.5.3. Instalace Apache, PHP, MySQL

Nejprve je zapotřebí nainstalovat balíček "WAMP". Podrobný popis instalace je uveden v sekci 6.2.1.

6.5.4. Nastavení přístupu do databáze MySQL

Otevřete nyní soubor "meric.ini" a změnte v případě potřeby přihlašovací údaje k databázi MySQL (server, port, uživatelské jméno, heslo, databáze).

6.5.5. Nastavení přístupu pro čtení z čidla DS1822

V konfiguračním souboru "meric.ini" v sekci "CIDLO" nastavte tyto atributy:

- "PORT" - hodnota portu ke kterému je čidlo připojeno.
- "ROZLISENI" - rozlišení přesnosti měření (12,10,11,9)
- "FREKVENCE_CTENI" - frekvence čtení dat v sekundách
- "ALARM_MAX" - vyhrazeno pro budoucí rozšíření
- "ALARM_MIN" - vyhrazeno pro budoucí rozšíření

6.5.6. Spuštění aplikace pro sběr dat z čidel DS1822

Aplikace se spouští prostřednictvím souboru "meric.exe" z umístění, které jste si zvolili v sekci 6.4.2.

6.5.7. Ukončení aplikace

ukončení aplikace - zavřením okna s aplikací

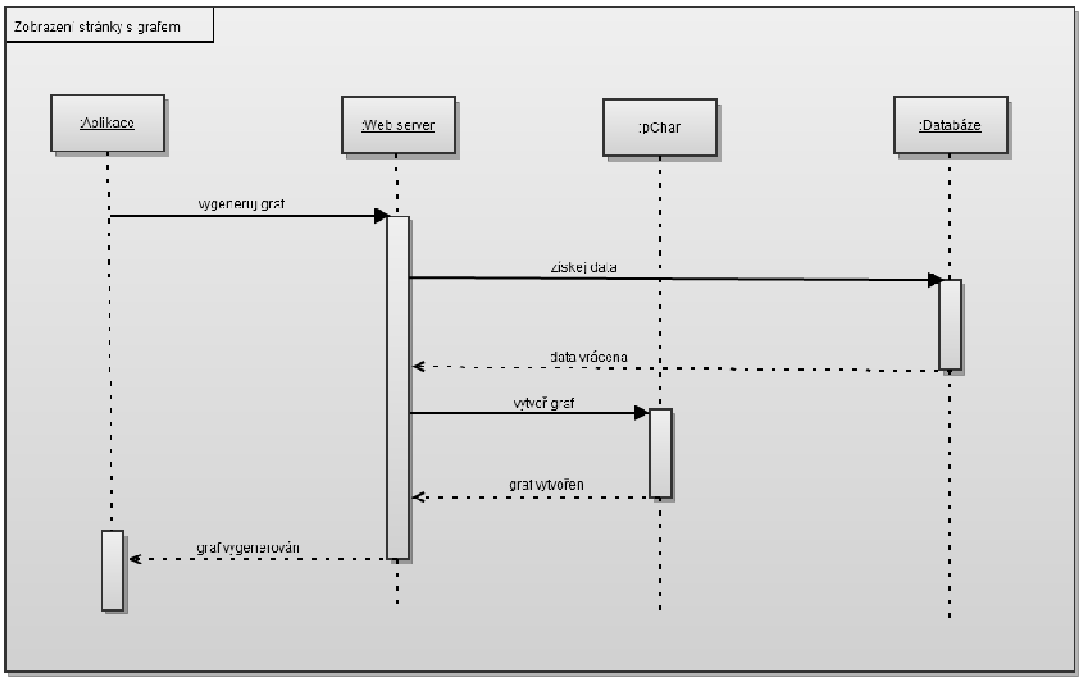
6.6. Požadavky na systém a software

- Windows XP a vyšší
- Microsoft Framework 2.0
- ovladač teplotního čidla

7. Programátorská dokumentace

7.1 Webová aplikace pro zobrazení naměřených hodnot z DS1822

Hlavním cílem této webové aplikace je umožnit uživateli zobrazovat naměřené teploty v uživatelsky přívětivé formě.



Obrázek 39. Sekvenční diagram webové aplikace

7.1.1. Použité technologie

Tato aplikace využívá technologii založenou na principu klient-server. Aplikace je vytvořena v jazyku **PHP** a jsou zde použité tyto technologie: **HTML**, **jQuery**, **pChart**, **GD** knihovna PHP, **Apache** a **MySQL**.

7.1.2. Skriptovací jazyk PHP

Jazyk **PHP** (Hypertext Preprocessor) se řadí mezi skriptovací jazyky obdobně jako Python, Perl, Ruby atd. Je často používán k vývoji webových aplikací. Vyznačuje se především přívětivou formou syntaxe jazyka. Program zapsaný ve skriptovacím jazyce se označuje jak **skript**.

7.1.3. HTML

Jazyk **HTML** (HyperText Markup Language) je značkovací jazyk pro hypertext. Patří mezi jazyky, pomocí kterých se vytvářejí stránky v systému World Wide Web.

7.1.4. jQuery

jQuery je malá javascriptová knihovna, která klade důraz na interakci mezi Javascriptem a **HTML**. Byla vydána Johnem Resigem.

7.1.5. GD knihovna a jazyk PHP

GD knihovna vyváří obrázky složené z úseček, oblouků a textů za použití různých barev. Knihovna je často obsažena v různých skriptovacích jazycích. **PHP** umožnuje zobrazovat výsledek přímo v prohlížeči.

7.1.6. Apache

Apache HTTP Server je softwarový webový server.

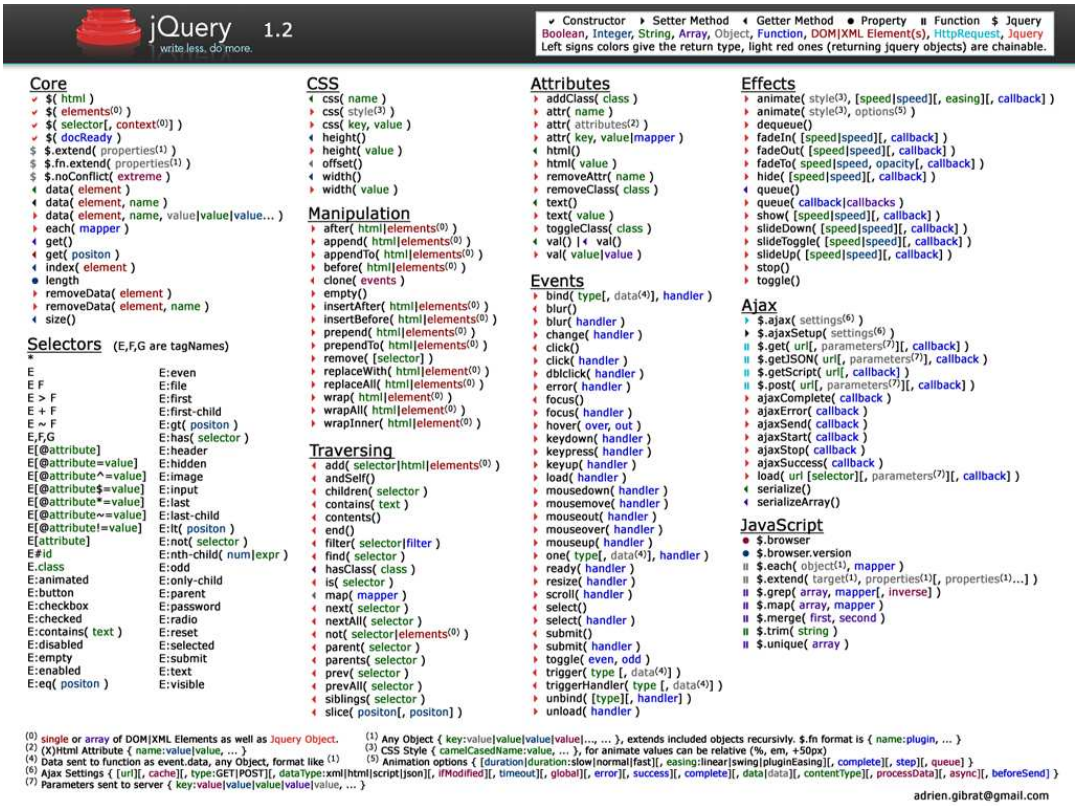
7.2.0. Rozdělení programu

Program je tvořen čtyřmi moduly:

- modul pro sběr požadavků (jQuery)
- modul pro tvorbu grafu (pChart)
- modul pro kreslení obrázků (GD knihovna PHP)
- modul pro komunikaci s databází (MySQL)

7.2.1. Modul pro sběr požadavků jQuery

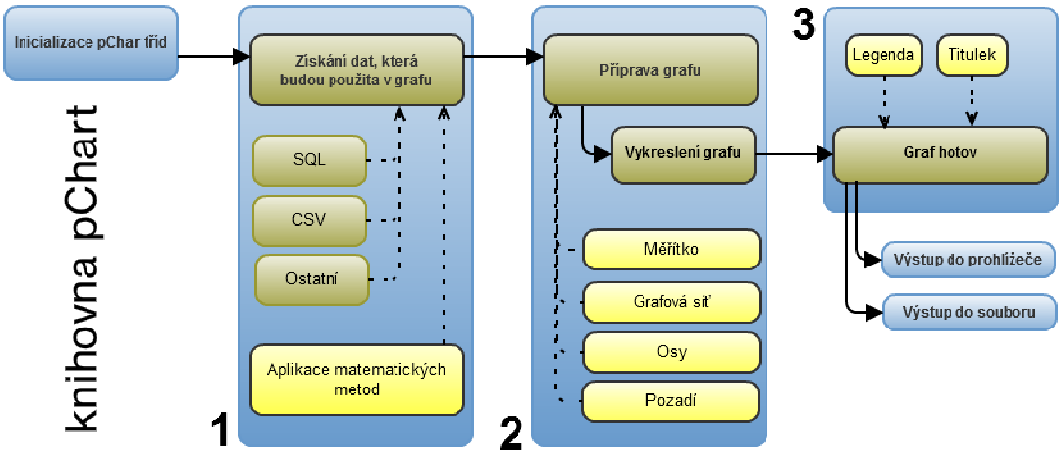
Modul (Filtr) je tvořen souborem (index.php). V tomto modulu se sbírají data od uživatele a zároveň slouží pro zobrazení požadovaného výstupu tj. grafu nebo tabulky. Filtr je tvořen **HTML** formulářem a prostřednictvím něho se získávají dat od uživatele. Je zde využit framework **jQuery**, který efektivně přistupuje k jednotlivým DOM HTML objektům na formuláři a zároveň umožňuje používat kaskádové styly (CSS). jQuery je zde rozšířen pomocí UI pluginů (User interface). Tyto pluginy složí pro práci s událostmi. Je zde implementován plugin pro události myši, výběru a přetahování (mouse, selectable, draggable). Dále je zde implementován plugin pro zobrazení stavu načítání tzv. progressbar.



Obrázek 40. jQuery přehled

7.2.2. Modul pChart

Modul **pChart** je tvořen jedinou **PHP** třídou. Pomocí tohoto modulu se efektivním způsobem vytváří graf. Jedna se o třídu, která je volně použitelná a distribuuje se pod licenci GNU. Tato třída je velice rozsáhlá a vyžaduje detailní prostudování dokumentace. Tato webová aplikace využívá pouze základní grafy.

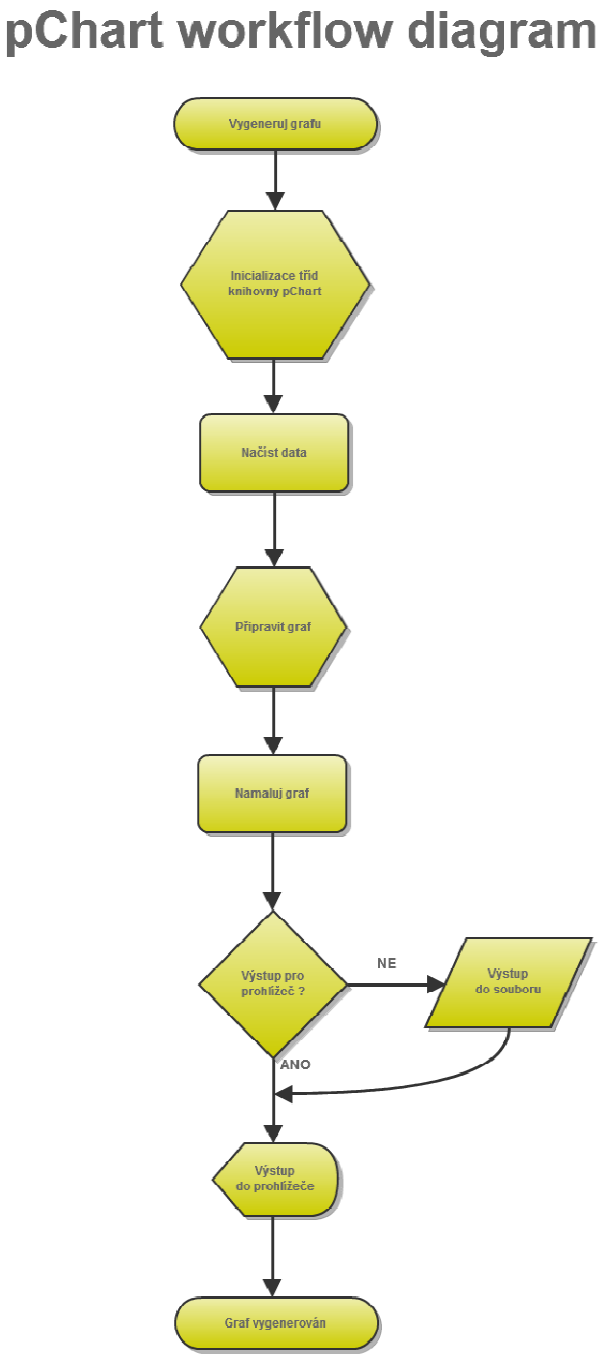


Obrázek 41. Knihovna pChart – postup vykreslení grafu

Knihovna **pChart** obsahuje **API** (Application Programming Interface), pomocí kterých se přistupuje k parametrizaci budoucího grafu.

```php
<?php
 // Standard inclusions
 include("pChart/pData.class");
 include("pChart/pChart.class");

 // Dataset definition
 $DataSet = new pData;
 $DataSet->ImportFromCSV("Sample/bulkdata.csv",",",array(1,2,3),FALSE,0);
 $DataSet->AddAllSeries();
 $DataSet->SetAbsciseLabelSerie();
 $DataSet->SetSerieName("January","Serie1");
 $DataSet->SetSerieName("February","Serie2");
 $DataSet->SetSerieName("March","Serie3");
 $DataSet->SetYAxisName("Average age");
 $DataSet->SetYAxisUnit("µs");

 // Initialise the graph
 $Test = new pChart(700,230);
 $Test->setFontProperties("Fonts/tahoma.ttf",8);
 $Test->setGraphArea(70,30,680,200);
 $Test->drawFilledRoundedRectangle(7,7,693,223,5,240,240,240);
 $Test->drawRoundedRectangle(5,5,695,225,5,230,230,230);
 $Test->drawGraphArea(255,255,255,TRUE);
 $Test->drawScale($DataSet->GetData(),$DataSet->GetDataDescription
(),SCALE_NORMAL,150,150,150,TRUE,0,2);
 $Test->drawGrid(4,TRUE,230,230,230,50);

 // Draw the 0 line
 $Test->setFontProperties("Fonts/tahoma.ttf",6);
 $Test->drawTreshold(0,143,55,72,TRUE,TRUE);

 // Draw the line graph
 $Test->drawLineGraph($DataSet->GetData(),$DataSet->GetDataDescription());
 $Test->drawPlotGraph($DataSet->GetData(),$DataSet->GetDataDescription(),3,2,255,255,255);

 // Finish the graph
 $Test->setFontProperties("Fonts/tahoma.ttf",8);
 $Test->drawLegend(75,35,$DataSet->GetDataDescription(),255,255,255);
 $Test->setFontProperties("Fonts/tahoma.ttf",10);
 $Test->drawTitle(60,22,"example 1",50,50,50,585);
 $Test->Render("example1.png");
?>
```

Obrázek 42. Příklad použiti technologie pChart



Obrázek 43. Přiklad použiti technologie pChart

7.2.3. Modul GD

Modul **GD** slouží pro vykreslování obrázků za pomocí čar a křivek. V aplikaci se využívají tyto funkce:

- imagecreatetruecolor (Vytvoření prostoru pro obrázek)
- imagecolorallocate (Namíchání nové barvy obrázku)
- imagefilledrectangle (Vykreslí vyplněný obdélník)

- imagettftext (Vloží text do obrázku)

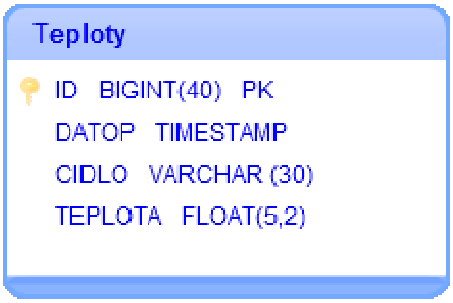
Detailnější popis funkcí je popsán v dokumentaci této knihovny, která je volně ke stažení a je distribuovaná pod licenci GNU.

7.2.4. Modul MySQL

Modul **MySQL**, zajišťuje komunikaci mezi databází a **PHP** skriptem. V konfiguračním souboru (hlavicka.php) je uložena konfigurace připojení databáze z jazyka **PHP**. Databázové schéma „cidla“ obsahuje tabulku „Teploty“, ve které jsou uloženy naměřené teploty. Prostřednictvím jazyka SQL, jsou prováděny všechny operace s daty v této databázi.

Příklad použití SQL dotazu v aplikaci:

```
 $sql = "SELECT cidlo,max(datop),teplota AS teplota FROM `teploty` WHERE datop >= DATE_ADD( now( ) , INTERVAL -1 MINUTE ) AND datop < now( ) ".$maskacidel." group by cidlo";

$sql= "SELECT cidlo FROM `teploty` where datop >= DATE_ADD(now(), INTERVAL -1 MINUTE) and datop < now() ";

$sql= "SELECT cidlo, round( avg( teplota ) , 2 ) as teplota, date_format( datop, '".$var."' ) AS datum
 FROM teploty WHERE datop >= '".$datumod." ".$casod."' AND datop <= '".$datumdo." ".$casdo."' ".$maskacidel." group by cidlo,datum ORDER BY date_format( datop, '".$var."' )";
```



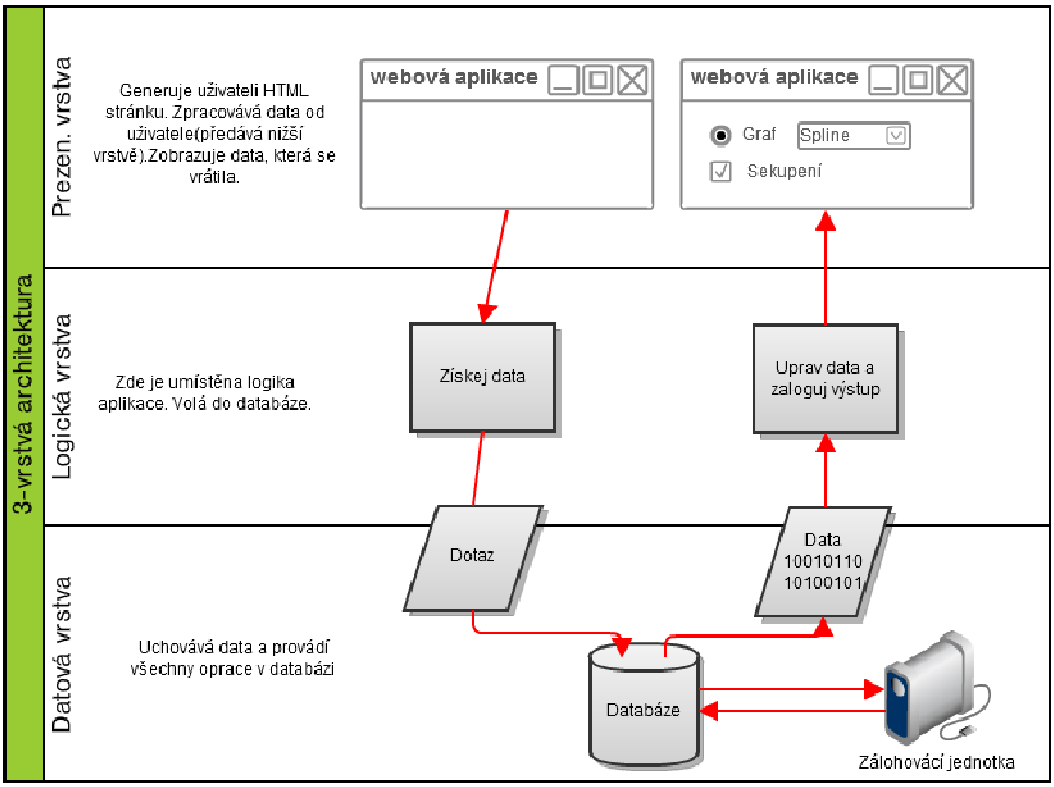
Obrázek 44. ER diagram tabulky Teploty

7.3.0 Třívrstvá architektura

Webová aplikace pro zobrazení naměřených hodnot z DS1822, využívá třívrstvou architekturu. Vrstvy jsou takto členěny:

- prezenční vrstva - tvoří ji webový prohlížeč
- logická vrstva - **PHP** skripty (logika aplikace)

- datová vrstva - databáze **MySQL**



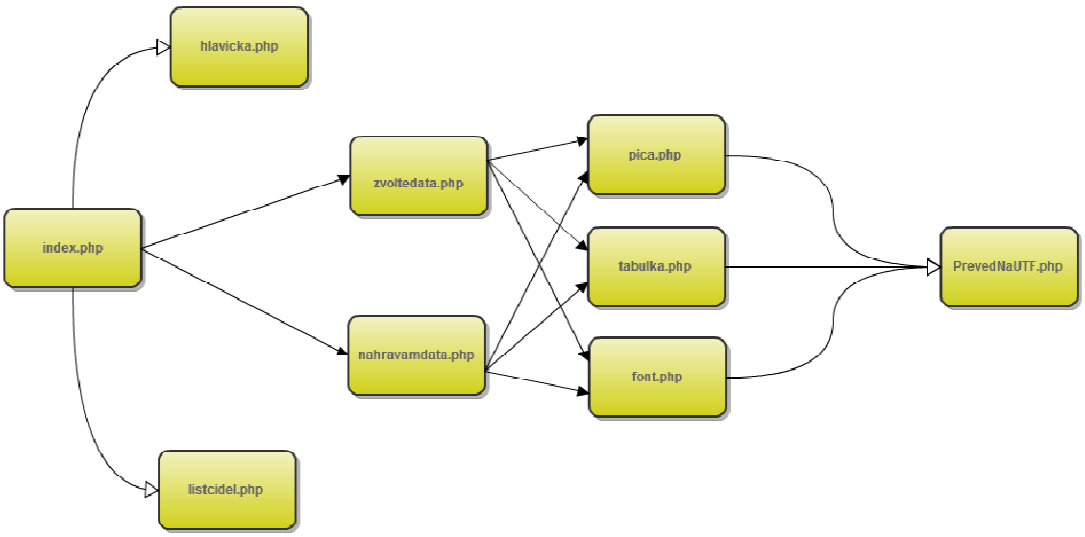
Obrázek 45. Třívrstvá architektura

7.3.1 PHP skripty

Webová aplikace je tvořena **PHP** skripty.

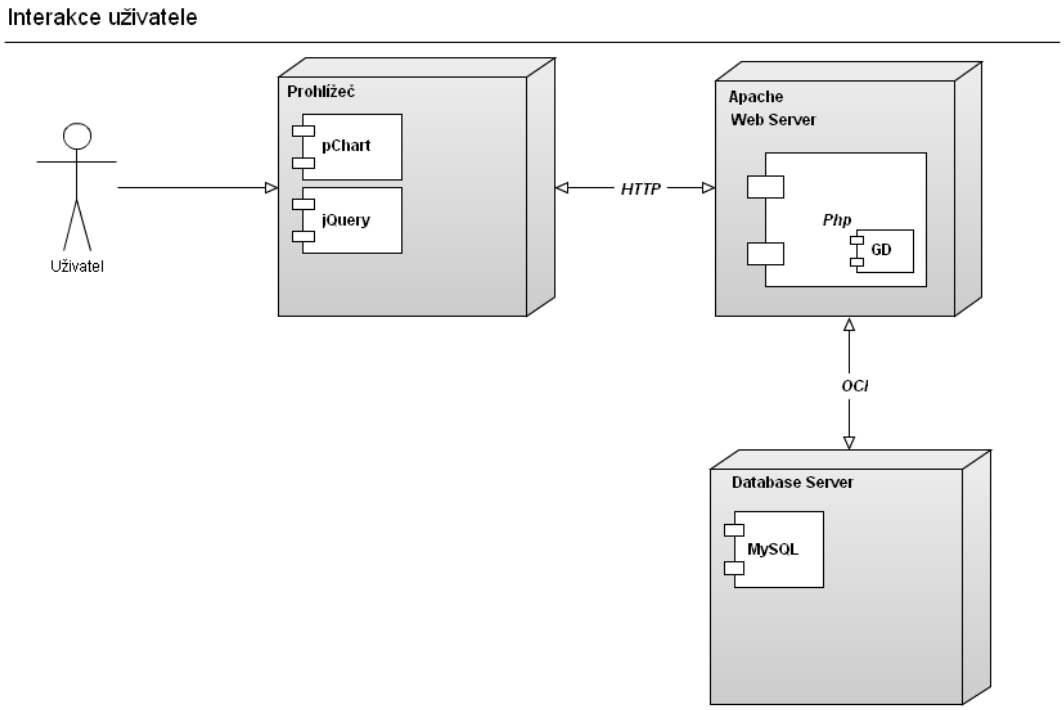
- index.php hlavní skript (sběr požadavků a vykreslení dat)
- zvoltedata.php vykreslí obrázek s textem „Zadejte data“
- nahravamdata.php vykreslí obrázek s textem „Nahrávám data“
- tabulka.php zobrazí data v tabulce
- pica.php vykreslí graf
- font.php vykreslí obrázek s názvy teplotních čidel
- listcidel.php vrátí seznam obsažených čidel v dotazu
- PrevedNaUTF.php převádí znaky do UTF kódování
- hlavicka.php parametrizuje připojení do databáze **MySQL**

7.4.0. Mapa Webu



Obrázek 46. Mapa Webu

7.5.0 Interakce uživatele



Obrázek 47. Interakce uživatele

7.6.0. Hlavní vlákno programu

Skript index.php tvoří hlavní vlákno celého programu. Tento skript vyvolává řadu činností, na základě vstupních podmětů, které zadává uživatel prostřednictvím webového formuláře. Data jsou získávána prostřednictvím javscriptových funkcí, které asynchronně získávají požadovaná data prostřednictvím **jQuery**.

7.6.1. Popis funkcí

- randomString() generuje náhodný řetězec, který složí pro případnou identifikaci objektu.
- centerpos() vystřeďuje graf na střed obrazovky
- centertab() vystřeďuje tabulku na střed obrazovky
- akce() načítá požadovaná data a následně je zobrazuje na obrazovce
- display() posouvá stav progresbaru (aktuální stavu)

7.7.0. Spuštění

Spuštění aplikace se provede zadáním URL adresy do prohlížeče ve tvaru http://nazev_serveru/adresar_kde_je_aplikace_nainstalovaná

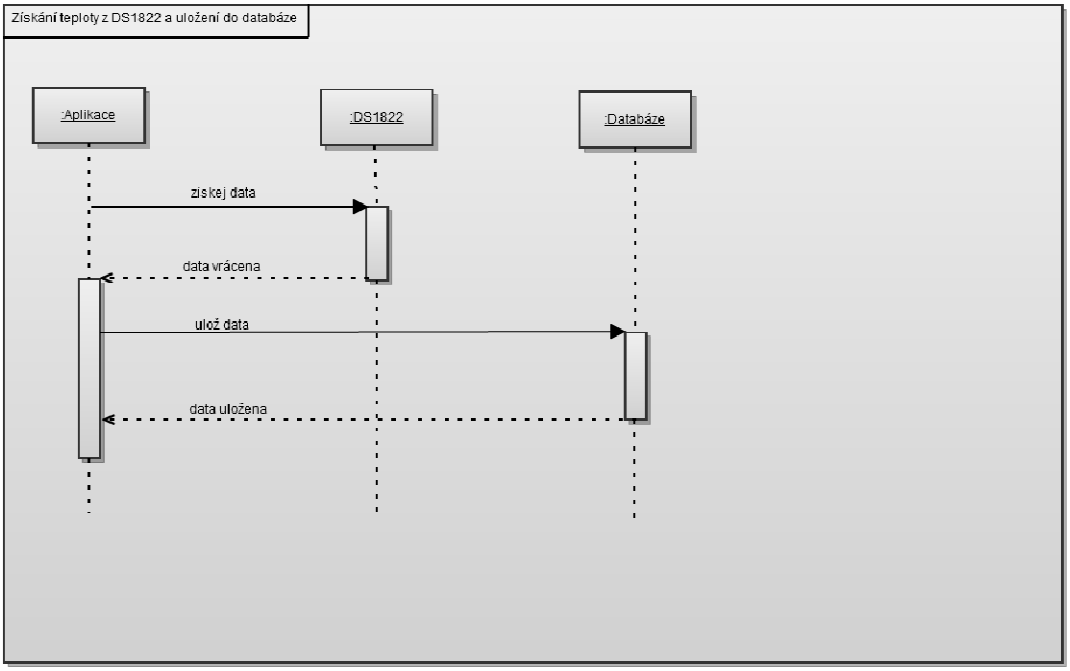
7.8.0. Použité aplikace

Nástroje, které byly použity pro vývoj aplikace .

- MySQL, databáze
- PHP, skriptovací jazyk
- jQuery, javascriptový framework **jQuery**
- pChart, **PHP** třída pro kreslení grafu
- Gliffy.com, vývojové diagramy

7.2. Aplikace pro sběr dat z čidel DS1822

Aplikace získává data z čidel **DS1822** prostřednictvím **1-Wire** sběrnice, která perzistentně ukládá do databáze **MySQL**.



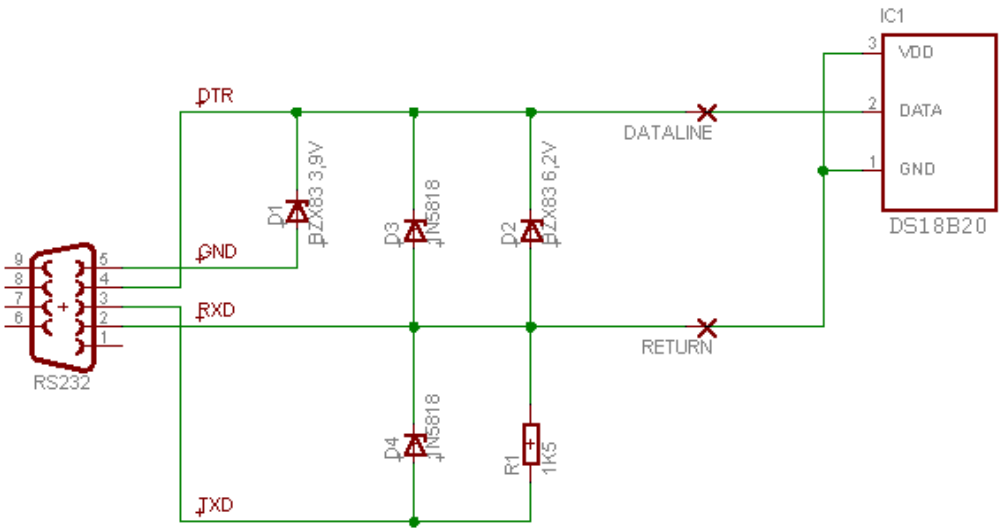
Obrázek 48. Sekvenční diagram aplikace pro sběr dat

7.2.1 Použité technologie

Aplikace je vytvořena v prostředí **Microsoft Visual studio 8.0 Express** v jazyku **C#**. Využívají se zde funkce **Microsoft Framework 2.0**. Komunikaci s čidly **DS1822** zprostředkovává interface **OneWireAPI**. Data jsou uložena v databázi **MySQL**.

7.2.2 Připojení DS1822 k PC

DS1822 se připojují k PC prostřednictvím sériového portu (RS-232). Pro připojení je možné použít adaptér RS-232 -> USB. Popis připojení **DS1822** k PC je znázorněn na obrázku 49. Toto připojení se shodné s připojení DS18B20, které je na tomto obrázku 49 znázorněno. Tyto dvě čidla patří do stejné rodiny (Family code), a proto je možné využít tohoto zapojení i pro **DS1822**.



Obrázek 49. Připojení DS1822 k PC

7.2.3 Popis inteface OneWireAPI

Prostřednictvím interface **OneWireAPI** komunikuje PC po **1-Wire** sběrnici s digitálními čidly **DS1822**. Ke komunikaci se používají příkazy tohoto rozhraní. Jednotlivé příkazy se volají v určitém sledu. Nedodržení tohoto pořadí má za následek špatnou komunikaci s čidly tj. nevalidní data.

7.2.4 Příkazy rozhraní OneWireAPI

- TMAccess(int,sbyte[])
  návěstí pro přístup do **Scratchpadu** paměti

- TMEndSession(int)
  ukončení spojení se sběrnicí

- TMExtendedStartSession(int, int, int[])
  navázání spojení se sběrnicí

- TMFisrt(int, sbyte[])
  spojení s 1. čidlem (vrací číslo sezení)

- TMNext(int, sbyte[])
  spojení s následujícím čidlem (vrací číslo sezení)

- TMOneWireLevel(int, short, short, short)
  změna stavu na sběrnici

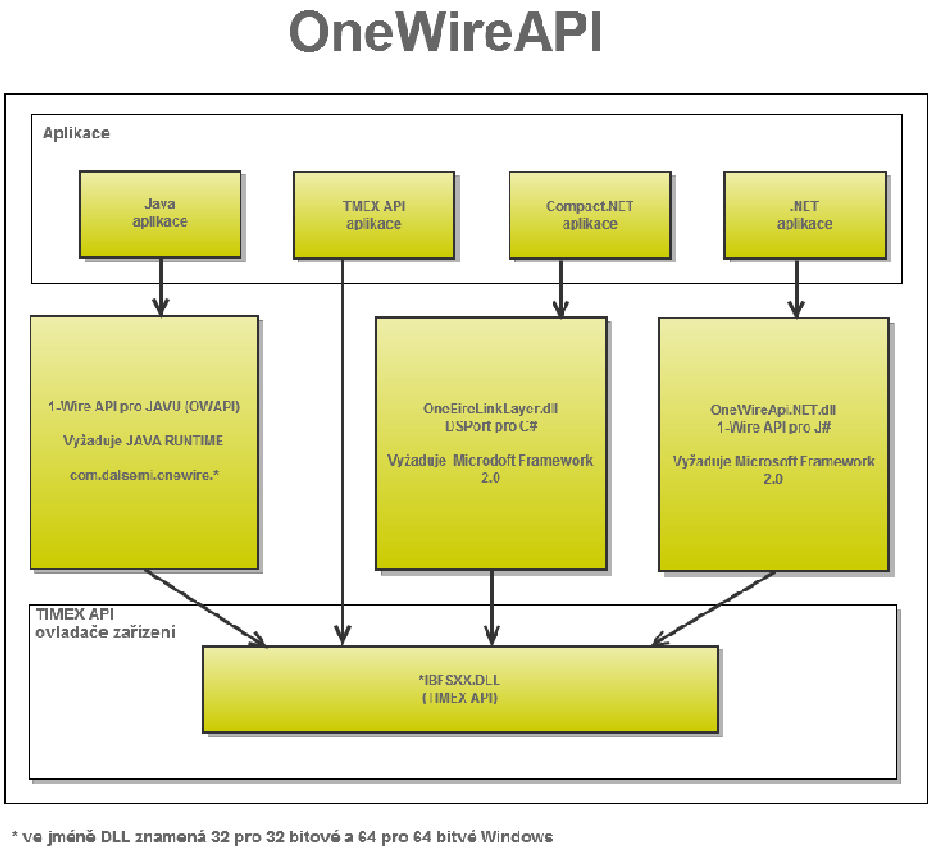
- TMRom(int, sbyte[], short[])
  získání 64. bitového čísla čidla

- TMSetup(int)
  nastavení příznnaku pro spojení se sběrnicí

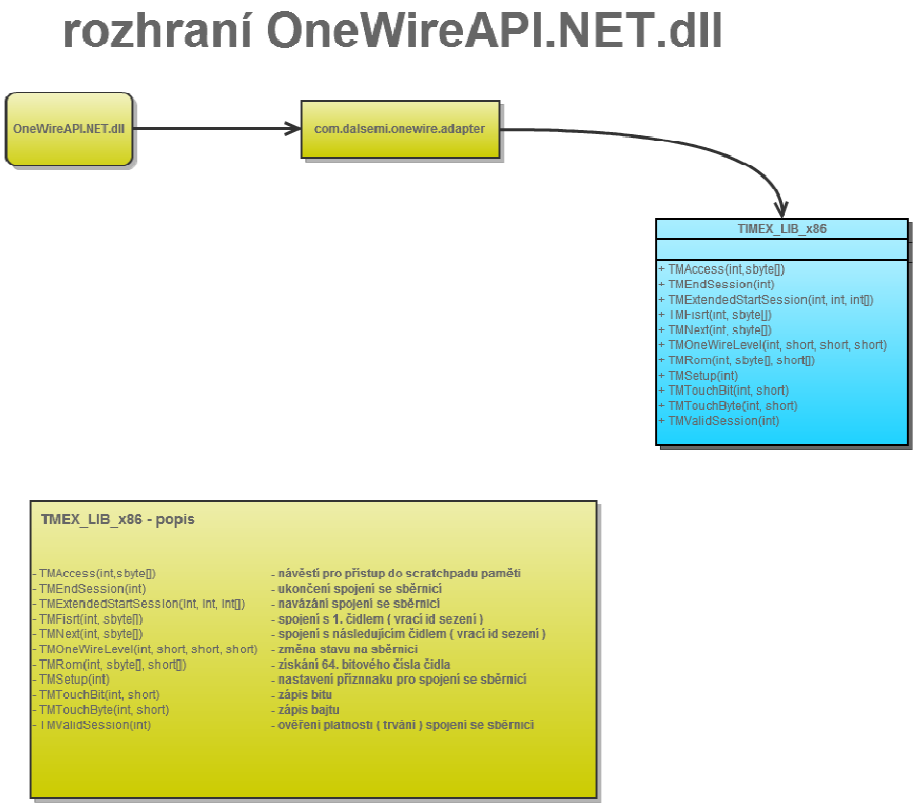
- TMTouchBit(int, short)
  zápis bitu

- TMTouchByte(int, short)
  zápis bajtu

- TMValidSession(int)
  ověření platnosti (trvání) spojení se sběrnicí



Obrázek 50. Abstrakce rozhraní OneWireAPI



Obrázek 60. Rozhraní OneWireAPI

7.3.0. Rozdělení programu

Program je tvořen moduly:

- modul pro čtení dat z DS1822
- modul pro zápis dat do databáze

7.3.1. Modul pro čtení dat z DS1822

Tento modul tvoři hlavní část programu. Jedná se o posloupnost kroků, které je zapotřebí dodržet, abychom byli schopni získat data z teplotních čidel **DS1822**.

Sled komunikace s teplotními čidly

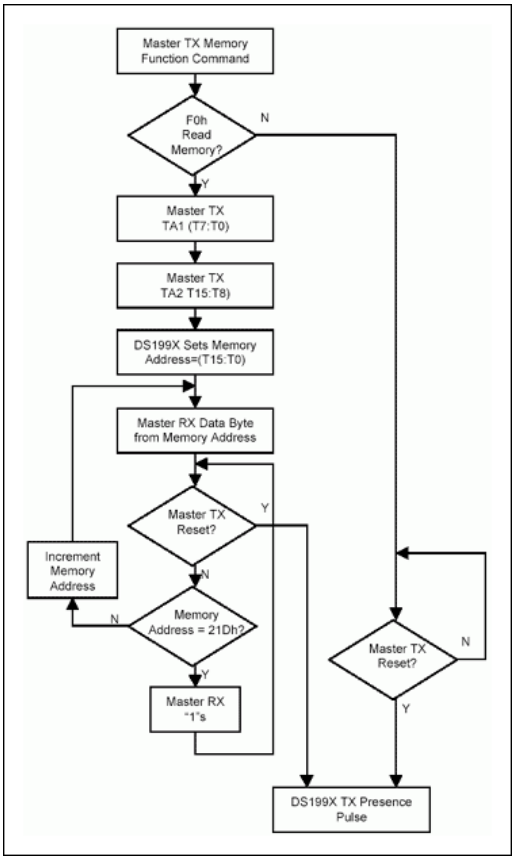
- sezení

- pulzu (RESET)

- pulzu (PRESENCE)

- dat

- sezení



Obrázek 61. Postup pro čtení dat z DS1822

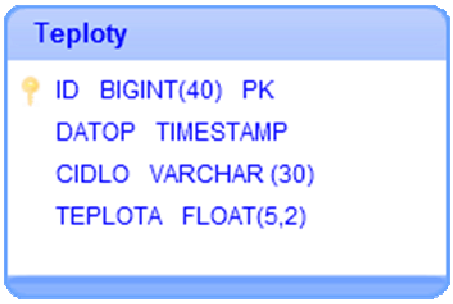
Modul pro čtení dat je reprezentován příkazem **ReadTemperature**().

```
private static bool ReadTemperature()
        {
            TMEX_LIB_x86.TMAccess(Meric.SHandle, Meric.StateBuffer)
            TMEX_LIB_x86.TMAccess(Meric.SHandle, Meric.StateBuffer)
            TMEX_LIB_x86.TMOneWireLevel(Meric.SHandle, 0, 1, 2)
            TMEX_LIB_x86.TMTouchByte(Meric.SHandle, 0x44);
            while (GetTickCount() < tick)
            TMEX_LIB_x86.TMValidSession(Meric.SHandle)
            TMEX_LIB_x86.TMOneWireLevel(Meric.SHandle, 0, 0, 0);
            TMEX_LIB_x86.TMTouchBit(Meric.SHandle, 0x01)
            TMEX_LIB_x86.TMAccess(Meric.SHandle, Meric.StateBuffer)
            TMEX_LIB_x86.TMTouchByte(Meric.SHandle, 0xBE);
            TMEX_LIB_x86.TMTouchByte(Meric.SHandle, (short)0xFF);
            Meric.DoCrc(DATA[i]);
        }
```

7.3.2. Modul pro zápis dat do databáze

Naměřená data jsou ukládána do databáze **MySQL**. Komunikace probíhá prostřednictvím rozhraní, které je tvořeno ovladačem ”MySql.Data.dll”. V první řadě se provede navázání spojení s databází a toto spojení se otevře pro čtení a zápis. Po celou dobu běhu aplikace pro sběr dat se drží spojení s databází. V rámci tohoto spojení zde aplikace vykonává řadu SQL dotazů.

Naměřená data jsou ihned po přečtení z teplotního čidla uložena do databáze. Nežli se data uloží, musí se zahájit transakce pro případ, že během ukládání dat dojde k chybě. Jakmile je transakce zahájena, provede se vlastní uložení dat do tabulky prostřednictvím SQL příkazu INSERT. Tato transakce se potvrdí v případě, že vše proběhlo korektně, anebo se transakce odvolá a žádná změna nebude uložena v databázi. Potvrzením transakce se data stávají perzistentními.



Obrázek 62. ER diagram tabulky Teploty

Zápis dat se provádí příkazem **setTemperatureToDB**(string sensor, double temperature).

```
private static void setTemperatureToDB(string sensor, double temperature)
 {
   …
  String sql = "INSERT INTO TEPLOTY (CIDLO,TEPLOTA) VALUES('" + sensor +
"','" + temperature.ToString().Replace(',', '.') + "')";
 …
}
```

7.3.3 Sled komunikace s databází MySQL

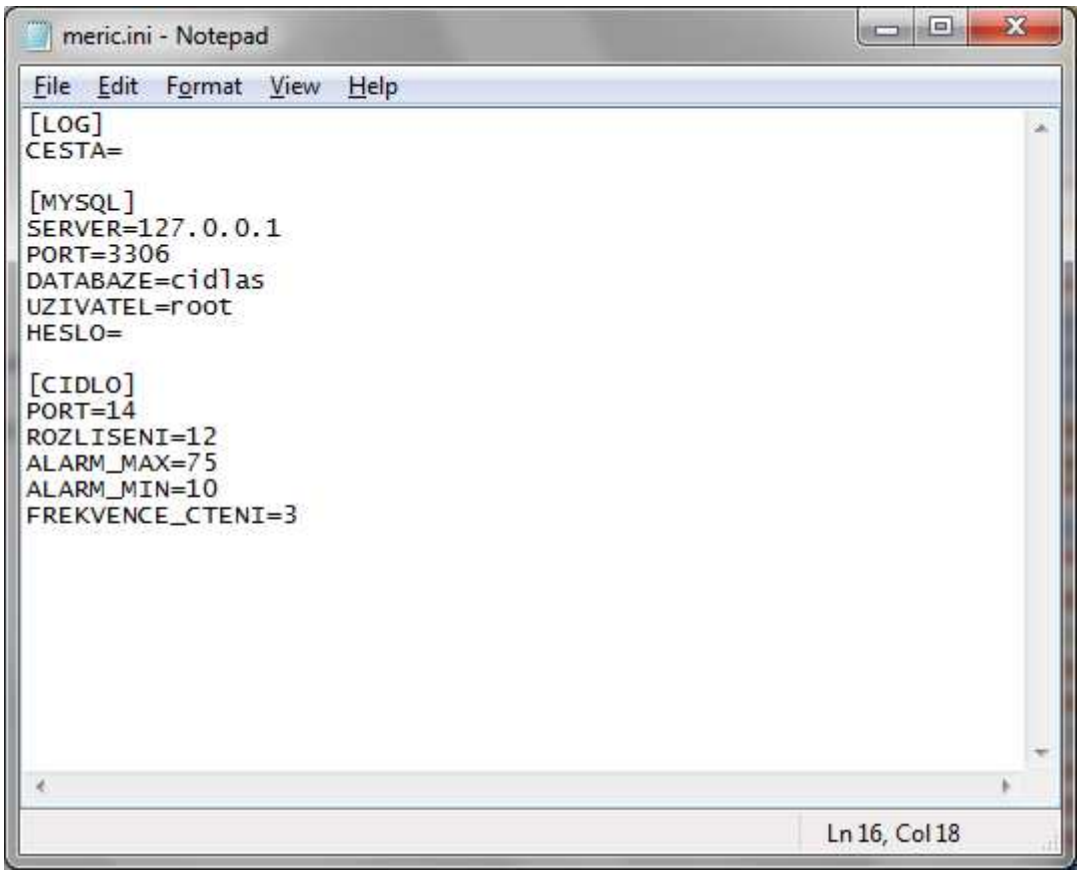
- zahájení transakce
- dat
- potvrzení či odvolání transakce

7.4.0. Spuštění

Spuštění aplikace se provádí prostřednictvím souboru ”meric.exe”.

7.4.1. Parametry aplikace

Aplikace pracuje s parametry, které jsou uloženy v souboru ”meric.ini”.



Obrázek 63. Parametrizační soubor

Parametry aplikace jsou rozděleny do následujících sekcí:

- LOG
- MYSQL
- CIDLO

7.4.2. Parametrizační sekce LOG

V parametru CESTA se nastavuje cesta k souboru, do kterého se má zapisovat výstup z aplikace

7.4.3. Parametrizační sekce MYSQL

V této sekci se nastavuje přístup do databáze MySQL.

Parametry:

SERVER	IP adresa serveru kde běží databáze MySQL
PORT	Port databáze
DATABAZE	Databázové schéma
UZIVATEL	Uživatelské jméno
HESLO	Heslo uživatele

7.4.4. Parametrizační sekce CIDLO

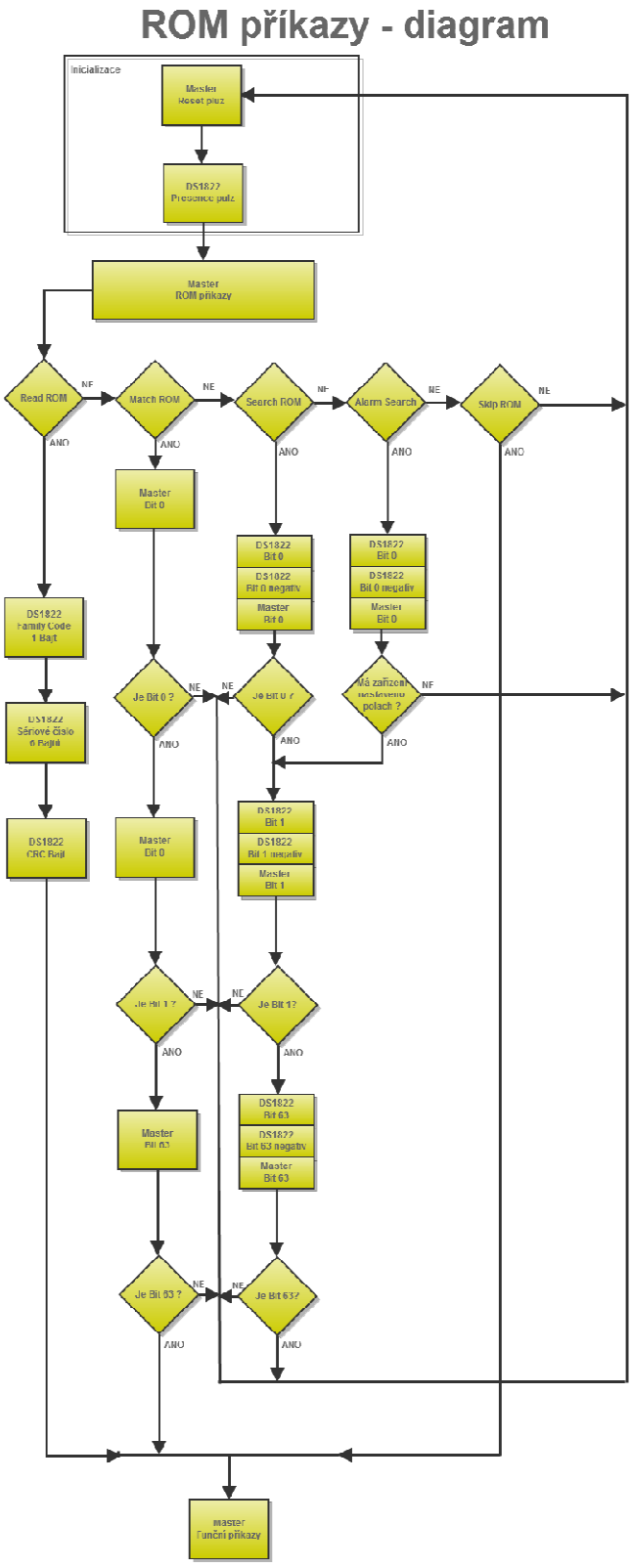
V této sekci se parametrizuje přístup k čidlu DS1822.

Parametry:

PORT	Port kde jsou připojena DS1822
ROZLISENI	Přesnost měření (9, 10, 11, 12)
ALARM_MAX	*Nastavení horní hranice poplachu
ALARM_MIN	*Nastavení spodní hranice poplachu
FREKVENCE_CTENIs	Udává periodu čtení dat z DS1822 v sekundách

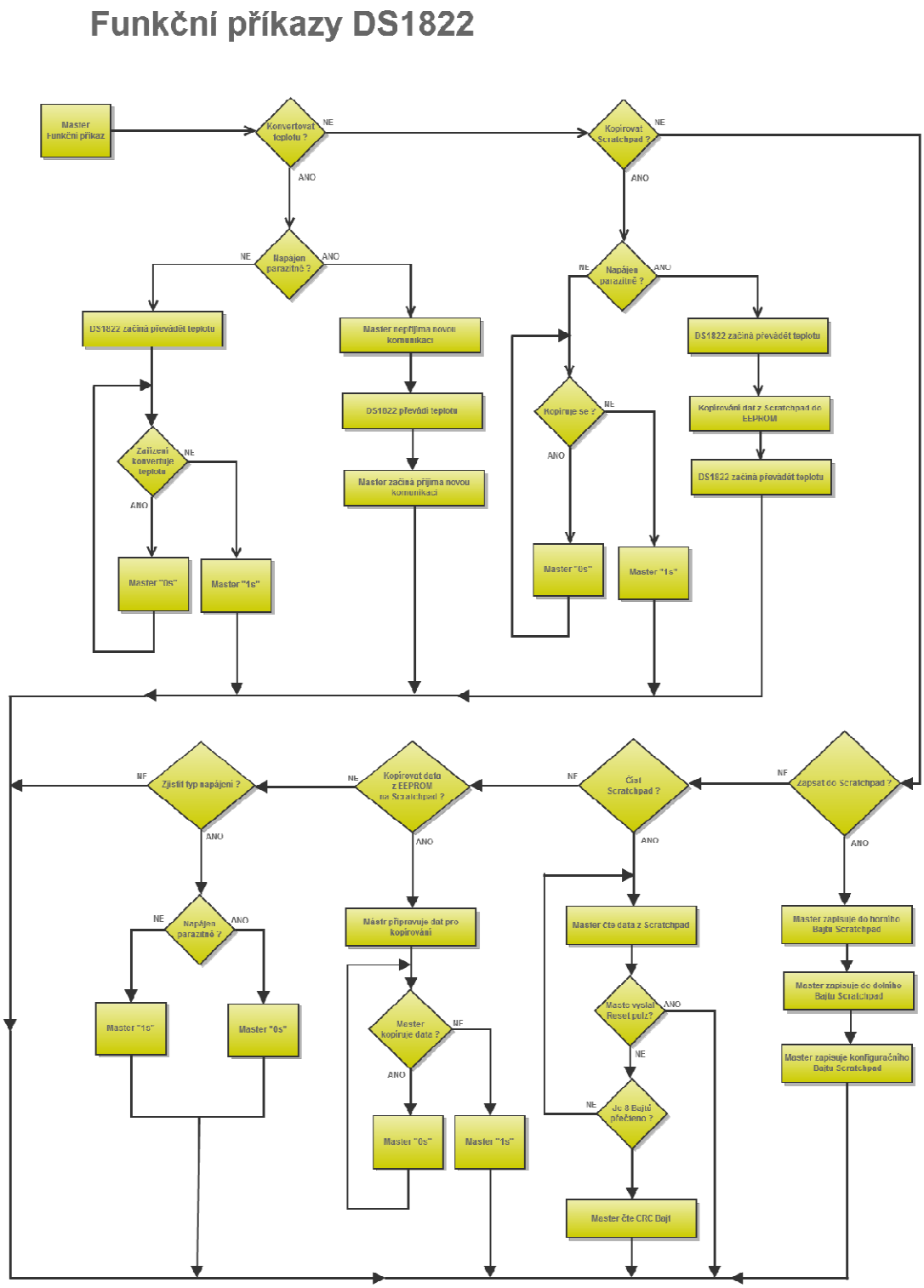
*- Vyhrazeno pro budoucí využití

7.5.0. ROM příkazy DS1822



Obrázek 64. ROM příkazy DS1822

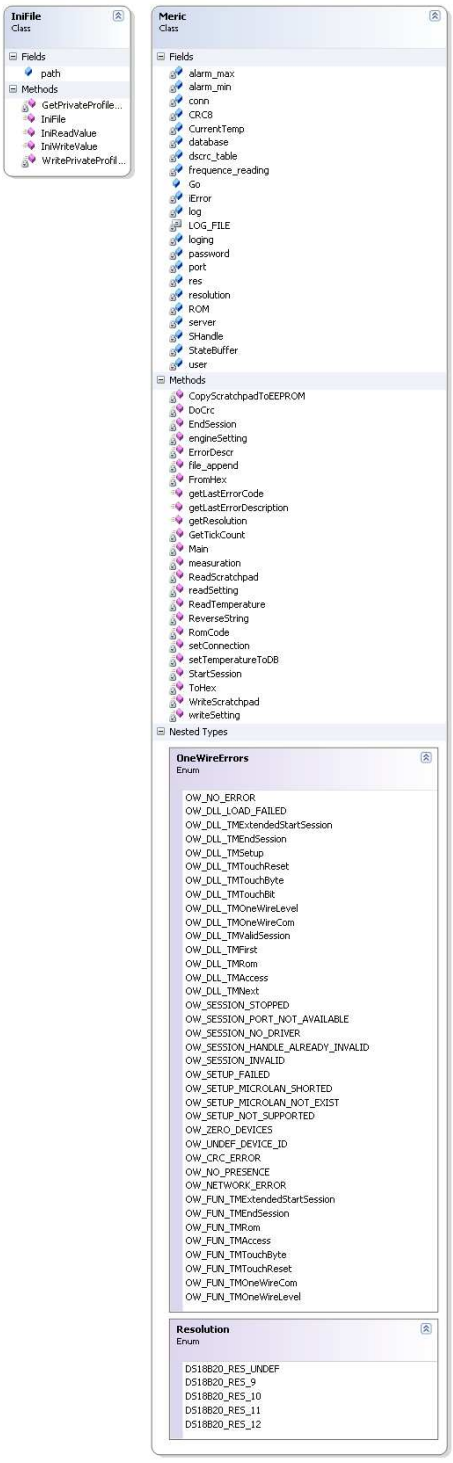
7.6.0. Funkční příkazy DS1822



Obrázek 65. Funkční příkazy DS1822

7.7.0. Třídy

V aplikaci pro sběr dat z čidel DS1822 jsou použity dvě třídy. Třída **IniFile**, která slouží pro zápis a čtení souboru „meric.ini“ ve kterém jsou uloženy parametry aplikace. Dále pak třída **Meric**, která tvoří hlavní vlákno aplikace. Na obrázku 25 je znázorněn ER diagram těchto tříd.



Obrázek 66. ER diagram tříd

7.8.0. Použité aplikace

Nástroje, které byly použity pro vývoj aplikace .

- MySQL, databáze
- Microsoft Visual Studio 8.0 Express, C#
- Gliffy.com, vývojové diagramy

Závěr

Mým cílem v bakalářské práci bylo v běžném programovacím jazyce naprogramovat aplikaci, která umožňuje monitorování teplot pomocí čidel DS1822. Vždy jsem chtěl naprogramovat aplikaci, která komunikuje s hardwarem.

V úvodu jsem se věnoval detailnímu popisu teplotního čidla DS1822. Toto hardwarové zařízení obsahuje řadu funkcí, avšak v rámci této bakalářské práce jsem naimplementoval pouze funkci čtení dat z teplotního registru.

Dále jsem se zabýval popisem 1-Wire sběrnice, pomocí které komunikuje teplotní čidlo DS1822 se svým okolím. Na této sběrnici je pozoruhodné, že ke komunikaci se zařízením využívá pouze jednu datovou linku (připojení jedním vodičem).

Při vývoji aplikace pro zobrazení naměřených dat jsem využil své zkušenosti v oblasti programování tenkých a tlustých aplikací. Využil jsem technologie, které usnadňují uživatelům práci s aplikací a nevyžadují řadu zbytečných kroků.

Prováděl jsem pokusy s třemi čidly a vše probíhalo, jak jsem předpokládal.

Věřím, že se najde využití této bakalářské práce a někomu poslouží jako základ pro další zkoumání v oblasti teplotních čidel.

Práce na tomto tématu mě velice bavila a dělal jsem ji s potěšením.

Conclusions

Main target was in common programming language make aplication, which measure temperature by force of senosors DS1822. I was which to programme a application, which communicate with hardware.

First, I was represent low level of describe sensor DS1822. This hard- ware machine, contain more functions. I was describ only measure temperature.

Next, I was represent low level of describe 1-Wire bus. This sensor DS1822 communicate during this bus with vicinity. It’s amazine, that this sensor communicate with vicinity throught one rib.

I did some test’s with 3 sensor’s and outcom is OK.

I believe that there will use this work for someone as a basis for further exploration.

Work on this topic, I was very entertained and I did it with pleasure.

Reference a zdroje

[1] Hrbáček, Jan. Komunikace mikrokontroléru s okolím - 2. díl Nakladatelství BEN, Praha, 2002.

[2] DS1822 Econo 1-Wire Digital Thermometer (datasheets.maxi-ic.com/en/ds/DS1822.pdf), použity obrázky pro kapitoly 2 a 3

[3] Mysql database (www.mysql.com)

[5] Apache HTTP Servere (www.apache.org)

[4] Apache HTTP Server

[5] Php (www.php.net)

[6] http://cs.wikipedia.org/wiki

[7] http://myego.cz/item/instalace-apache-mysql-a-php-na-windows

Z výše uvedených referencí a zdrojů byly převzaty obrázky a citace textů pro tuto bakalářskou práci. Především pro kapitoly 2 a 3.

H. První příloha

Obsah přiloženého CD

I. Obsah přiloženého CD

bin/
- cidlas.zip
- meric.zip

doc/
- programatorska dokumentace.zip
- uzivatelska dokumentace.zip

src/
- cidlas.zip
- meric.zip

readme.txt
- readme.txt

Navíc CD/DVD obsahuje:

data/
- sql.txt

install/
- wamp.zip
- install 1 wire drivers x86 _v403beta.msi

literature/
- readme.txt

Seznam obrázků

Seznam tabulek