Gymnázium, Brno, Vídeňská 47
Předmět: programování

Maturitní práce

Měření teploty na RaspberryPi

Technická specifikace; dokumentace

Jan Horáček; jan.horacek@seznam.cz

15. dubna 2015

Obsah

1 Úvod

1.1 Zadaní maturitní práce

Cílem této maturitní práce je navrhnout řešení inteligentního teplotního senzoru. Tedy takového senzoru, který bude průběžně zaznamenávat teploty a na požádání je zobrazí v přehledném grafu.

1.2 Řešení

K řešení problému jsem využil hardware a software postavený na univerzální desce RaspberryPi, a to především díky její malé spotřebě a univerzálnosti. Popis navrhnutého řešení je předmětem této práce a je rozebírán níže. Cílem této práce je detailně informovat o principech tohoto řešení.

2 Hardware

Hardwrové řešení využívá dvou klíčových komponent:

1. RaspberryPi B+ první generace a
2. digitální teplotní senzor DS18B20.

2.1 RaspberryPi

RaspberryPi je počítač velikosti kreditní karty vybavený procesorem architektury ARM a mnoha běžnými rozhraními (např. USB, ethernet), na kterém lze spustit plnohodnotný operační systém. Pro své řešení jsem zvolil pokročilejší model B+.

Klíčovými výhodami RaspberryPi pro tento projekt jsou:

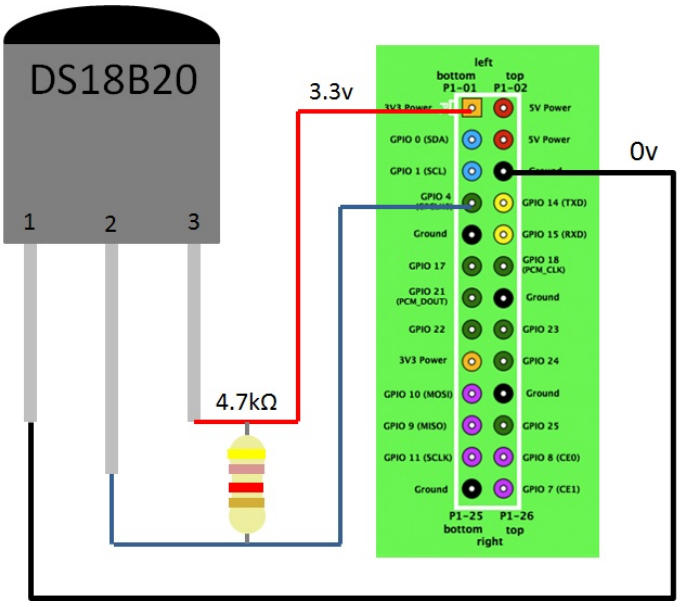
- nízká spotřeba elektrické energie: $1 - 2\ W$ a
- rozhraní *GPIO*.

RaspberryPi je napájeno přes *micro usb*, kterému je v našem případě předřazen běžný síťový napájecí adaptér. RaspberryPi je dále připojeno do sběrnice ethernet a konečně je k němu připojen teplotní senzor.

2.2 Teplotní senzor DS18B20

Teplotní senzor *DS18B20* nabízí přesné měření teploty (řádově 2 platná místa °C) a digitální rozhraní k datům reprezentujícím aktuální teplotu. Nejedná se tak o pouhý termistor, který je nutné připojit k AD převodníku a složitě vyhodnocovat teplotu na základě vstupního napětí. Tyto operace řeší už samotné teplotní čidlo, ve kterém je vestavěná mimo zmiňovaného termistoru i obslužná elektronika. Tato elektronika poskytuje informace pomocí sériové sběrnice podobné *RS232-TTL*. Tato sběrnice je detailně popsána datasheetem teplotního senzoru, viz [Max08].

Výstupem teplotního senzoru jsou tedy digitální data, která čteme RasberryPi skrze GPIO porty. Propojení obou zařízení přehledně zobrazuje obrázek 1.



Obrázek 1: Propojení RaspberryPi a senzoru DS18B20 [Pri]

3 Software

Toto řešení využívá operační systém *Raspbian*, který je načítán z 16 GB vysokorychlostní paměťové karty.

Základními softwarovými stavebními elementy pak jsou:

1. webový server *Apache*,
2. automatický spouštěč procesů *Cron*,
3. databáze *SQLite*,
4. skript pro monitoring teploty a
5. skript generující webové rozhraní.

Základní princip celé plejády softwaru lez shrnout do dvou bodů:

- Každých 5 minut zapíšeme do databáze *SQLite* aktuální teplotu a aktuální čas.
- Na základě požadavku od klienta vygenerujeme webovou stránku s grafem, který zobrazuje závislost teploty na čase.

Nyní podrobně popíši zajímavé části softwaru.

3.1 Monitorovací skript

Monitorovací skript je spouštěn každých 5 minut *cron*em a zapisuje aktuální teplotu do databáze.

Po aktivací modulů `w1-gpio` a `w1-therm` je k dispozici soubor `w1_slave` v adresáři `/sys/bus/w1/devices/ID/`, kde *ID* je unikátní ID teplotního senzoru, v konkrétním případě `/sys/bus/w1/devices/28-0000065d0146/`. Tento soubor je klíčový pro získávání dat z teplotního senzoru.

Data se získávají velmi jednoduše:

```
1 user@dev:/sys/bus/w1/devices/28-0000065d0146 cat w1_slave
2 54 01 4b 46 7f ff 0c 10 fd : crc=fd YES
3 54 01 4b 46 7f ff 0c 10 fd t=21250
```

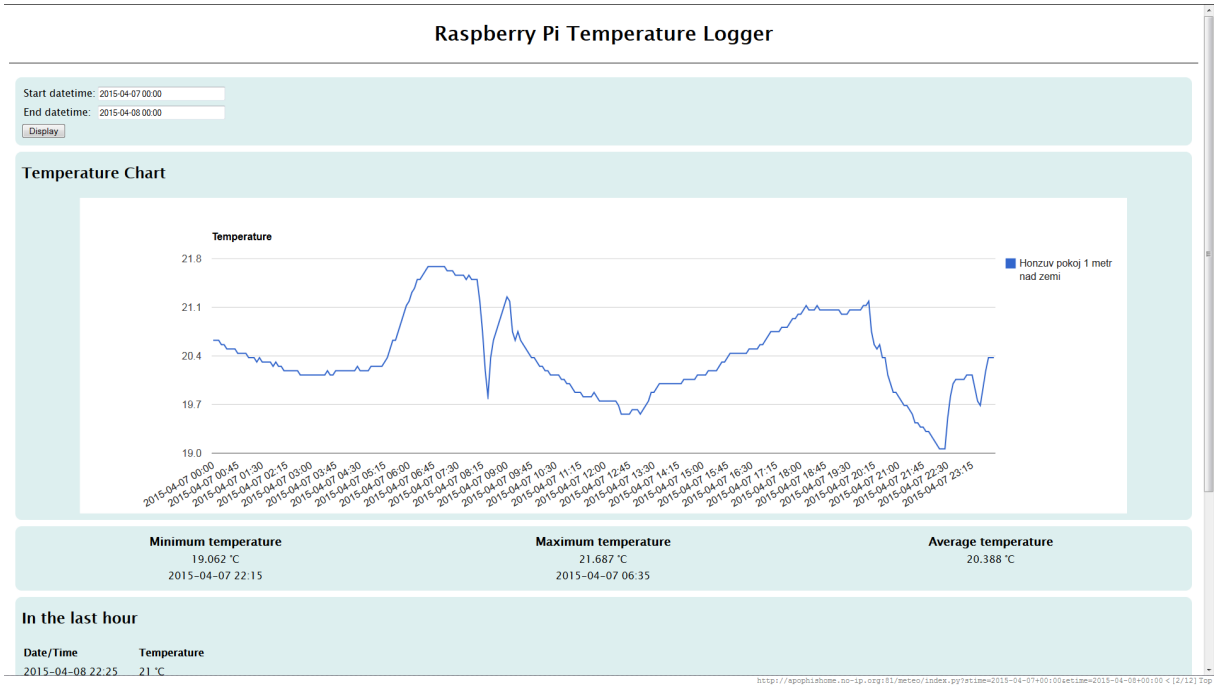
Aktuální teplota je číslo `21250` vydělené konstantou 1000, tedy $t = 21.250\ °C$.

Na tomto jednoduchém principu staví monitorovací skript, který jednoduše provede parsing teploty a výslednou hodnotu uloží do databáze.

Monitorovací skript `monitor.py` psaný v *Pythonu* přikládám. Tento skript je založen na projektu [Ras].

3.2 Skript generující webovou stránku

Webové rozhraní tohoto řešení přehledně zobrazuje obrázek 2.



Obrázek 2: Webové rozhraní

Webová stránka je opět generována skriptem v Pythonu, který přikládám v souboru `index.py`. Tento skript je založen na projektu [Ras].

Program je jednoduše volán Apachem jako *cgi skript*, na standardní výstup tedy vytiskne http hlavičku a html reprezentující webovou stránku. K této webové stránce je připojen css soubor `style.css`, který také přikládám.

Samotný skript obsahuje mnoho neatraktivních dotazů do databáze, vypisování html elementů a parsing dat z formuláře. Přesto bych se rád zastavil u jednoho zajímavého elementu: u grafu. Pro vykreslování grafu jsem použil poměrně zajímavý nástroj od *Google*, konkrétně *Google Charts*, založený na *Javascriptu*. Celé vykreslování grafu tedy funguje velmi jednoduše:

1. Definujeme data v grafu, typ grafu, název grafu.
2. Zavoláme funkci *Google Charts* pro vykreslení grafu.

Do příslušného elementu `<div>`, který specifikujeme, se graf následně vykreslí (samozřejmě vektorově a interkativně).

4 Zhodnocení, výhody a nevýhody

Svou maturitní práci považuji za úspěšnou, projekt je plně funkční. Závěrem bych rád diskutoval některé parametry navrhnutého řešení.

1. **Spotřeba elektrické energie**
   Celý projekt byl navrhován s tím, že bude neustále v provozu a tudíž byl a je požadavek na nízkou spotřebu navrhnutého zařízení více než klíčový. Při navrhování řešení jsem zvažoval měřit teplotu na desce typu *Arduino*, či na vlastní navrhnuté DPS s jednoduchým mikrokontrolérem (např. *ATmega8a*). Výhodou takového řešení by byla o několik řádů nižší spotřeba energie (řádově mW), zařízení by bylo možné napájet např. solárním článkem (v případě umístění venku) a bylo by tak energeticky soběstačné.

   I přes energetické výhody jsem se rozhodl pro RaspberryPi a to především pro to, že *2 W* jsou pro mě dostatečně malá spotřeba a protože RaspberryPi je jednoduše konfigurovatelné a univerzální.

2. **Snadná opakovatelnost**
   Za velkou přednost svého projektu považuji to, že jsem schopen vytvořit další inteligentní teplotní senzory bez nároku na složitou výrobu vlastního HW, čí velké úpravě SW. Projekt lze jednoduše zopakovat na jakémkoliv RaspberryPi.

3. **Snadná konfigurovatelnost, dynamičnost**
   Za další přednost svého řešení považuji to, že jsem použil plnohodnotný počítač a tudíž jsou pro mě změny SW víceméně otázkou několika kliknutí a především snadného debugování (porovnáme-li např. s použitím mikrokontroléru a vlastního FW).

Reference

[Max08] MaximIntegrated: *DS18B20 Programmable Resolution 1-Wire Digital Thermometer Datasheet*. http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf. Version: 2008

[Pri] PrivateEyePi: *Monitor your home temperature using your Raspberry Pi*. http://www.projects.privateeyepi.com/home/temperature-gauge

[Ras] RaspberryWebServer: *Building an SQLite temperature logger*. http://raspberrywebserver.com/cgiscripting/rpi-temperature-logger/