

# Kompenzace vlivu samoohřevu elektronického zařízení měřicího prostorovou teplotu

## Diplomová práce

*Studijní program:*

N2612 Elektrotechnika a informatika

*Studijní obor:*

Informační technologie

*Autor práce:*

**Bc. Daniel Varnuška**

*Vedoucí práce:*

Mgr. Čeněk Jirsák

Katedra aplikované matematiky





## Zadání diplomové práce

# Kompenzace vlivu samoohřevu elektronického zařízení měřícího prostorovou teplotu

*Jméno a příjmení:* **Bc. Daniel Varnuška**  
*Osobní číslo:* M20000175  
*Studijní program:* N2612 Elektrotechnika a informatika  
*Studijní obor:* Informační technologie  
*Zadávací katedra:* Ústav mechatroniky a technické informatiky  
*Akademický rok:* 2021/2022

### Zásady pro vypracování:

1. Na pokusné DPS a v okolním prostoru rozmístěte teplotní čidla a naměřte dostatečné množství dat v termokomoře pro jejich následné zpracování.
2. Pokuste se nalézt optimální matematický vztah mezi získanými daty z teplotních senzorů a nepřímo měřenou (vypočtenou) okolní prostorovou teplotou tak, aby byl odstraněn vliv vlastního ohřevu součástkami na DPS.
3. Navržené řešení zrealizujte a aplikujte.
4. Provedte ověřovací měření v termokomoře a zhodnoťte výsledné chování z hlediska přesnosti měření.

*Rozsah grafických prací:*  
*Rozsah pracovní zprávy:*  
*Forma zpracování práce:*  
*Jazyk práce:*

dle potřeby dokumentace  
40–50 stran  
tištěná/elektronická  
Čeština



### **Seznam odborné literatury:**

- [1] ANDĚL, Jiří. Statistické metody. 2. přeprac. vyd. Praha: Matfyzpress, 1998. 274 s. ISBN 80-85863-27-8.
- [2] ZVÁRA, Karel: Regrese. MATFYZPRESS, Praha, 2008.

*Vedoucí práce:*

Mgr. Čeněk Jirsák  
Katedra aplikované matematiky

*Datum zadání práce:*

12. října 2021

*Předpokládaný termín odevzdání:*

16. května 2022

prof. Ing. Zdeněk Plíva, Ph.D.  
děkan

L.S.

doc. Ing. Josef Černožorský, Ph.D.  
vedoucí ústavu

V Liberci dne 12. října 2021

## Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědom toho, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

16. května 2022

Bc. Daniel Varnuška

# Kompenzace vlivu samoohřevu elektronického zařízení měřícího prostorovou teplotu

## Abstrakt

Práce se zabývá odhadem pokojové teploty z teplot naměřených elektronickým zařízením v reálném čase. Naměřené teploty podléhají chybám v důsledku samozahřívání zařízení. Parametry kompenzační rovnice jsou odhadnuty z lineárních a exponenciálních regresních modelů, které kompenzují sledované parametry teplotních změn zařízení při konstantní pokojové teplotě.

**Klíčová slova:** Samoohřev, lineární regresní model, exponenciální regresní model, odhad pokojové teploty

# Compensation of self-heating of the electronic device measuring the room temperature

## Abstract

The work deals with the estimation of room temperature from temperatures measured by electronic equipment. The measured temperatures are subject to error due to the self-heating of the device. The parameters of the compensation equation are estimated from linear and exponential regression models, which compensate the monitored parameters of temperature changes of the device at a constant room temperature.

**Keywords:** Self-heating, linear regression model, exponential regression model, room temperature estimation

## Poděkování

Rád bych poděkoval Mgr. Čěňku Jirsákovi za pomoc a cenné rady při vzniku této práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>10</b>
<b>2</b>	<b>Zařízení a jeho běžné použití</b>	<b>11</b>
<b>3</b>	<b>Teoretické základy</b>	<b>13</b>
3.1	Pracovní prostředí . . . . .	13
3.2	Oteplovací a ochlazovací děje . . . . .	13
3.2.1	Oteplení při přerušovaném chodu . . . . .	15
3.3	Šíření tepla materiály . . . . .	16
3.3.1	Kondukce . . . . .	16
3.3.2	Konvekce . . . . .	17
3.3.3	Sálání . . . . .	17
3.4	Regrese . . . . .	18
<b>4</b>	<b>Technická realizace</b>	<b>19</b>
4.1	Návrh testů . . . . .	19
4.2	Měřicí soustava . . . . .	23
4.3	Software měřicí soustavy . . . . .	25
4.4	Příprava dat . . . . .	26
<b>5</b>	<b>Statistické řešení</b>	<b>33</b>
5.1	Teplotní posun zařízení . . . . .	36
5.2	Teplotní posun displeje . . . . .	37
5.3	Náběh zařízení . . . . .	40
5.4	Kompenzace prostředí . . . . .	40
5.5	Aplikace řešení . . . . .	41
<b>6</b>	<b>Závěr</b>	<b>44</b>
	<b>Použitá literatura</b>	<b>45</b>
	<b>Přílohy</b>	<b>46</b>
A	Oteplovací tabulka	46
B	Tabulka náběhu zařízení	47

C Aplikace kompenzačního algoritmu	48
D Kompenzační algoritmus	54
E Logovací program pro Raspberry Pi	60
F Logovací program pro DPS	61



## Seznam zkratk

<b>DPS</b>	Deska plošných spojů
<b>RPi</b>	Raspberry Pi
<b>GPIO</b>	Univerzální vstupní/výstupní pin
<b>GND</b>	Uzemnění v elektronice
<b>IP</b>	Protokol používaný v počítačových sítích

# 1 Úvod

Tato práce se zabývá problematikou měření prostorové teploty v místnosti prostřednictvím víceúčelového elektronického zařízení, které disponuje třemi teplotními čidly. Z důvodu osazení teplotních čidel, které jsou součástí zařízení, dochází k nežádoucímu ovlivňování měřených hodnot teplem, které produkuje periférie zařízení jako vedlejší produkt své činnosti. Ohřevem zařízení vzniká rozdíl mezi měřenou teplotou na zařízení a prostorovou teplotou. Cílem této práce je navrhnout řešení, které by umožňovalo víceúčelovému zařízení měřit reálnou prostorovou teplotu s využitím teplotních čidel umístěných na desce plošných spojů. Posléze toto řešení zrealizovat, aplikovat a otestovat v reálném prostředí.

Ohřev zařízení rozdělíme na teplo z displeje a ze zbytku zařízení. Oteplení rozčleníme na ustálené oteplení a neustálené (v přechodu). Za ustáleného oteplení se nezvyšuje teplota zařízení. Za tímto účelem navrhne vhodný měřicí plán, abychom získali potřebná data k odhadům. Plán bude sestaven z jednotlivých měření, které pozorují zvyšování teploty zařízení po jeho zapnutí a zvýšení teplot po zapnutí displeje a posléze pokles teplot po vypnutí displeje. Sestavíme měřicí aparaturu pro realizaci tohoto měření. Navrhne parametrický model pro rozčleněný ohřev a parametry odhadneme pomocí regrese. Nalezené parametry dosadíme do parametrické funkce. Teplota, čas a podsvícení displeje jsou vstupními parametry funkce. Výstupní hodnota funkce reprezentuje vypočtenou odhadovanou prostorovou teplotu. Parametrickou funkci implementujeme do algoritmu napsaného v programovacím jazyce C, který bude v reálném čase provádět výpočet odhadu. V závěru shrneme výsledky práce.

## 2 Zařízení a jeho běžné použití

Nejprve si zařízení představíme. Zařízení je složeno ze tří hlavních dílů: deska plošných spojů, displej a plastová krabička. Na desce plošných spojů jsou osazeny tři teplotní čidla, výkonný procesor, bezdrátová komunikační jednotka a další periférie. Flexi páskem je k ní připojen 4,3 palcový LCD TFT displej s rozlišením 856 × 480 pixelů, který je zasazený v plastovém rámečku o rozměrech 124 × 81 × 16 milimetrů.

Zařízení je určeno k montáži na stěnu vnitřních prostorů budov. Instaluje se do elektroinstalační krabice nacházející se většinou na vnitřních stěnách budov. Plastová krabička zařízení má v rozích magnety jimiž se zařízení přichytí k plechové destičce uvnitř elektroinstalační krabice.

Stěny, které oddělují vnitřní prostor od venkovního prostoru, nejsou vhodné pro instalace podobných zařízení. Přestože stěny izolují od venkovního prostředí, tak nestálé venkovní prostředí ovlivňuje vlastnosti stěny. Ty mohou negativně ovlivnit zařízení umístěná na těchto stěnách.



Obrázek 2.1: Ukázka elektronického zařízení

Zařízení nedisponuje vlastním zdrojem napájení a vyžaduje připojení napájení o napětí 12 V. Zařízení je navrženo k neustálému provozu, kdy se jednou připojí a od té doby není zapotřebí jej odpojit od elektrického proudu. Mohou ovšem nastat výjimečné situace, při kterých se zařízení vypne nebo je zapotřebí vypnout či restartovat. Tomu tak může být před instalací zařízení, při výpadku elektrické energie, instalací aktualizací softwaru a podobně. Mimo těchto speciálních případů je zařízení vždy v provozu.

Deska plošných spojů neobsahuje žádné aktivní ani pasivní chlazení, které by pomáhalo odvádět vygenerovanou tepelnou energii perifériemi. Zařízení je přirozeně ochlazováno svým prostředím, ve kterém se nachází.

Jeho účelem je poskytnout uživateli své prostředky jako ovládacího prvku domácnosti. Zařízení má mnoho možností nasazení, kde může sloužit k regulaci teploty v místnosti, ovládání osvětlení, žaluzií, zobrazování výstupu z kamerových systémů v lokální síti a dalším způsobům použití.

Většinu svého operačního času má zařízení vypnutý displej z důvodu šetření energie a možnému světelnému rušení obyvatel domácnosti. Většinou v noci, ale i ve dne, může být světlo vyzařující zařízením nepříjemné pro uživatele. Z těchto důvodů je obrazovka zařízení vždy zhasnutá a rozsvící se v případech, kdy systém dostane signál z dotykové vrstvy obrazovky, že se jí někdo dotknul, nebo po signálu ze senzoru přiblížení (proximity), který ohlásí, že je něco v blízkosti zařízení, nebo vnitřním stavem zařízení, kdy zařízení ohlásí, že je nutné něco zobrazit uživateli, třeba nějakou výstrahu. Displej je v základu nastaven tak, že se zhasíná po pěti minutách od poslední interakce s uživatelem.

V zařízení je distribuce operačního systému Linux.

Zařízení také disponuje bezdrátovým komunikačním modulem, kterým je možné se připojit do lokální sítě domácnosti. Poté je možné ovládat zařízení vzdáleně z kteréhokoliv počítače v lokální síti. Mimo možnosti připojení se k bezdrátovému přístupovému bodu samotné zařízení vytváří bezdrátový přístupový bod se svou sítí. To je dobré, když nechceme mít zařízení v domácí síti, ale přesto potřebujeme v zařízení něco nastavit. Především se jedná o bezpečnostní opatření.

## 3 Teoretické základy

V této kapitole se podíváme na teoretické základy při měření ohřevu elektrických objektů. Elektronické zařízení si lze představit jako soubor těles, z nichž jen v některých vzniká teplo. Toto teplo pak proudí z místa vzniku přes sousední tělesa postupně až k rozhraní mezi souborem těles a vnějším ovzduším. Na rozhraní se uvolňuje tepelná energie do okolí. Akumulace tepla a také tepelná ztráta závisí na pracovním prostředí, ve kterém se zařízení nachází.

### 3.1 Pracovní prostředí

Pojmem pracovní prostředí rozumíme vlastnosti prostoru, v němž je přístroj umístěn. Tyto vlastnosti závisí na složení ovzduší, na jiných předmětech tam umístěných a na dalších jiných podmínkách, které ovlivňují činnost elektronického zařízení. Prostředí dělíme na:

- prostředí základní - bez zvláštního vlivu na elektrické zařízení,
- prostředí jednoduchá - jen s jedním druhem zvláštního vlivu,
- prostředí složitá - s několika vlivy jednoduchých prostředí současně.

Základní nebo také obyčejné prostředí je takové, v němž se teplota okolí pohybuje v rozmezí  $-10\text{ }^{\circ}\text{C}$  až  $35\text{ }^{\circ}\text{C}$ , vzduch neobsahuje více než 15 g vody na  $1\text{ m}^3$ , tj. relativní vlhkost nepřestoupí 80 %, a jeho nadmořská výška je nejvýše 1000 m. Příkladem základního prostředí jsou prostory uvnitř budov v mírném klimatu. Prostředími jednoduchými jsou například: prostředí studené, horké, vlhké, mokré, prašné, výbušné. Složitá prostředí poté vznikají různými kombinacemi prostředí jednoduchých. Zvláštní případ je prostředí venkovní.

### 3.2 Oteplovací a ochlazovací děje

Je-li elektronické zařízení zapnuté a provádí-li jakoukoliv činnost, tak generuje jako vedlejší produkt tepelnou energii. Následující text je převzatý z Elektrických přístrojů od Otty Havelky [2], kde je vysvětleno, jak se elektrické přístroje oteplují a ochlazují v závislosti na procházejícím elektrickém proudu.

Prochází-li úsekem dlouhého vodiče s odporem  $R$  ( $\Omega$ ) elektrický proud  $I$  (A), pak v každém časovém okamžiku  $dt$  v něm vznikají tepelné ztráty  $dQ = RI^2dt$ .

Toto teplo se dělí na dvě části. Jedna část se odvede ochlazením ve stejné době do okolí. Je to množství  $\alpha_0 A dt$ . Veličinu  $\alpha_0$  ( $\text{W}/\text{m}^2 \cdot \text{K}$ ) nazýváme součinitel přenosu tepla,  $A$  ( $\text{m}^2$ ) je ochlazovací povrch úseku,  $\Delta\vartheta$  ( $\text{K}$ ) je okamžité oteplení tělesa proti okolí. Druhá část tepla v tělese zůstává a zvyšuje jeho teplotu o  $d\vartheta$ . Místo  $d(\Delta\vartheta)$  budeme psát  $d\vartheta$ , protože první derivace oteplení podle času je stejná jako první derivace teploty podle času. Teplo k tomu spotřebované určuje vztah  $cV d\vartheta$ . Přitom  $c$  ( $\text{J}/\text{m}^3 \cdot \text{K}$ ) je objemová tepelná kapacita uvažovaného úseku vodiče s objemem  $V$  ( $\text{m}^3$ ). Platí tedy rovnice:

$$RI^2 dt = \alpha_0 A \Delta\vartheta dt + cV d\vartheta \quad (3.1)$$

Rovnici řešíme separací proměnných

$$dt = \frac{cV}{RI^2 - \alpha_0 A \Delta\vartheta} d\vartheta \implies t = -\frac{cV}{\alpha_0 A} \ln \left[ \frac{RI^2}{cV} - \frac{\alpha_0 A \Delta\vartheta}{cV} \right] + K$$

Integrační konstantu  $K$  stanovíme z počáteční podmínky: v okamžiku  $t = 0$  je také oteplení  $\Delta\vartheta = 0$

$$K = \frac{cV}{\alpha_0 A} \ln \frac{RI^2}{cV}$$

Výsledné řešení rovnice 3.1 je po úpravě

$$t = -\frac{cV}{\alpha_0 A} \ln \left( 1 - \frac{\alpha_0 A}{RI^2} \Delta\vartheta \right)$$

Rovnici antilogaritmujeme a vyjádříme explicitně okamžité oteplení

$$\Delta\vartheta = \frac{RI^2}{\alpha_0 A} \left[ 1 - \exp \left( -\frac{\alpha_0 A t}{cV} \right) \right] = \Delta\vartheta_\infty \left[ 1 - \exp \left( -\frac{t}{\tau} \right) \right] \quad (3.2)$$

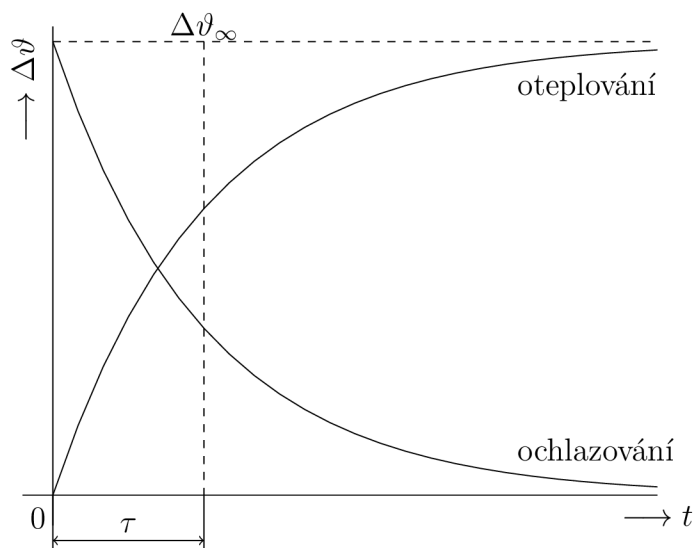
Přitom jsme zavedli označení

$$\Delta\vartheta_\infty = \frac{RI^2}{\alpha_0 A} \quad (3.3)$$

$$\tau = \frac{cV}{\alpha_0 A} \quad (3.4)$$

Křivka časového průběhu oteplení má exponenciální charakter. Z počáteční nulové hodnoty narůstá stále pomaleji, až za dobu  $t \rightarrow \infty$  se ustálí na hodnotě maximálního oteplení  $\Delta\vartheta_\infty$ . Časovou konstantu  $\tau$  (s) lze přitom definovat jako dobu, za kterou by těleso dosáhlo ustáleného oteplení  $\Delta\vartheta_\infty$ , kdyby se všechno uvolněné teplo spotřebovalo jen na zvyšování jeho teploty, tj. kdyby bylo  $\alpha_0 = 0$ . V grafickém záznamu rovnice (3.2) na obrázku 3.1 vytíná velikost této takzvané časové konstanty  $\tau$  na pořadnici  $\Delta\vartheta_\infty$  tečna k oteplovací křivce v jejím počátku ( $t = 0$ ).

Při odvozování rovnice (3.2) jsme poměry idealizovali a předpokládali, že časová konstanta se během oteplování nemění. To úplně není pravda, protože součinitel přestupu tepla  $\alpha_0$  je mírně závislý na oteplení  $\Delta\vartheta$  a během oteplování se mění. Měnit



Obrázek 3.1: Oteplovací a ochlazovací charakteristika

se může poněkud i měrná objemová tepelná kapacita  $c$ . Proto zpravidla měřená oteplovací křivka nesouhlasí zcela přesně s křivkou vypočítanou.

Prochází-li stále stejný proud vodičem po velmi dlouhou dobu, těleso dosáhne teplotně ustáleného stavu. V ustáleném stavu je  $\Delta\vartheta/dt = 0$ ,  $\Delta\vartheta = \Delta\vartheta_\infty$ , takže rovnice (3.1) nabude tvaru

$$RI^2 = \alpha_0 A \Delta\vartheta_\infty$$

tedy tvaru totožného s (3.3). Dosažením tohoto nejvyššího možného oteplení povrchu  $\Delta\vartheta_\infty$  za daných podmínek nastává tepelná rovnováha: veškeré teplo vznikající ve vodiči se jeho povrchem předává do okolí, zpravidla do ovzduší.

### 3.2.1 Oteplení při přerušovaném chodu

Přerušíme-li obvod, přestane vodiči procházet proud. Protože předchozím průchodem proudu ohřátý přístroj předává akumulované teplo dále do okolí, přístroj se ochlazuje a jeho teplota klesá. Základní rovnice (3.1) má nyní tvar

$$0 = \alpha_0 A \Delta\vartheta dt + cV d\vartheta \quad (3.5)$$

Provedeme separaci proměnných a rovnici integrujeme

$$t = -\frac{cV}{\alpha_0 A} \ln \frac{\Delta\vartheta}{\Delta\vartheta_1} + B, \quad \text{kde } (\Delta\vartheta_1 = 1\text{K})$$

Integrační konstantu  $B$  určuje opět počáteční podmínka  $t = 0$ ;  $\Delta\vartheta = \Delta\vartheta_\infty$

$$B = \frac{cV}{\alpha_0 A} \ln \frac{\Delta\vartheta}{\Delta\vartheta_1}$$

po dosazení a úpravě je řešení rovnice 3.5

$$t = \frac{cV}{\alpha_0 A} \ln \frac{\Delta\vartheta}{\Delta\vartheta_1}$$

Odtud oteplení

$$\Delta\vartheta = \Delta\vartheta_\infty \exp\left(-\frac{\alpha At}{cV}\right) = \Delta\vartheta_\infty \exp\left(-\frac{t}{\tau}\right) \quad (3.6)$$

Protože časová konstanta  $\tau$  je opět určena vztahem (3.4), je ochlazovací exponenciála v obrázku 3.1 zrcadlovou křivkou ke křivce oteplovací.

U zařízení, která mají provoz charakterizovaný dobami zatížení a klidu, nemusí proudová dráha nikdy dosáhnout ustáleného oteplení. Průběh oteplení se určuje nejlépe z oteplovací a ochlazovací křivky a časového programu chodu.

### 3.3 Šíření tepla materiály

Přenos tepla je fyzikální jev výměny tepelné energie mezi dvěma systémy širícím se teplem. Základními principy přenosu tepla jsou teplota a tepelný tok. Teplotou je určeno dostupné množství tepelné energie a tepelný tok představuje přesun tepelné energie.

V mikroskopickém měřítku existuje přímá souvislost mezi kinetickou energií molekul a tepelnou energií. Jak teplota stoupá, tak se v molekulách navyšuje tepelný neklid projevující se lineárním pohybem a kmitáním. Oblasti, které obsahují vyšší kinetickou energii, předávají energii oblastem s nižší kinetickou energií. Jednoduše řečeno, přenos tepla lze seskupit do tří kategorií: vedení (konduktce), proudění (konvekce) a sálání.

#### 3.3.1 Konduktce

Vedením se teplo přenáší přímou srážkou molekul. Z oblasti s větší kinetickou energií se tepelná energie přenáší do oblasti s nižší kinetickou energií. Částice s vyšší rychlostí se srážejí s částicemi s nižší rychlostí. V důsledku toho se zvýší energie pomalejších částic. Vedení je nejběžnější formou přenosu tepla a dochází k němu prostřednictvím fyzického kontaktu.

Proces vedení tepla závisí na následujících faktorech: teplotní spád, průřez materiálu, délka dráhy přenosu a fyzikální vlastnosti materiálu. Teplotní spád je fyzikální veličina, která popisuje směr a rychlost šíření tepla. Teplotní tok bude vždy probíhat od nejteplejšího k nejchladnějšímu nebo, jak bylo uvedeno výše, od vyšší k nižší kinetické energii. Jakmile se teplotní rozdíl vyrovná dojde k tepelné rovnováze, přenos tepla se zastaví.

Průřez a dráha pohybu hrají důležitou roli ve vedení. Čím větší je velikost a délka předmětu, tím více energie je potřeba k jeho zahřátí. A čím větší je povrchová plocha, která je vystavena, tím více tepla se ztrácí. Menší objekty s malými průřezy mají minimální tepelné ztráty.



Fyzikální vlastnosti určují, které materiály přenášejí teplo lépe než jiné. Konkrétně koeficient tepelné vodivosti určuje, že kovový materiál bude vést teplo lépe než látka, pokud jde o vodivost. Následující rovnice vypočítá rychlost vedení:

$$Q = \frac{k \cdot A \cdot (T_{hot} - T_{cold})}{d}$$

kde  $Q$  je teplo přenesené za jednotku času,  $k$  je tepelná vodivost bariéry,  $A$  je teplosměnná plocha,  $T_{hot}$  je teplota horké oblasti,  $T_{cold}$  je teplota chladné oblasti a  $d$  je tloušťka bariéry.

### 3.3.2 Konvekce

Když se tekutina, jako je vzduch nebo kapalina, zahřeje a poté proudí od zdroje, nese tepelnou energii. Tento typ přenosu tepla se nazývá konvekce. Tekutina nad horkým povrchem rozpíná, stává se méně hustou a stoupá.

Na molekulární úrovni se molekuly rozpínají po zavedení tepelné energie. Se zvyšující se teplotou dané hmoty tekutiny se objem tekutiny musí zvětšovat stejným faktorem. Tento jev způsobuje vytlačování tekutiny. Jak bezprostřední horký vzduch stoupá, tlačí dolů hustší, chladnější vzduch. Tato řada událostí představuje, jak se tvoří konvekční proudy. Rovnice pro rychlost proudění se vypočítá takto:

$$Q = hc \cdot A \cdot (T_s - T_f)$$

kde  $Q$  je teplo přenesené za jednotku času,  $hc$  je součinitel prostupu tepla konvekcí,  $A$  je teplosměnná plocha povrchu,  $T_s$  je teplota povrchu,  $T_f$  je teplota tekutiny.

### 3.3.3 Sálání

Tepelné sálání vzniká vyzařováním elektromagnetických vln. Tyto vlny odnášejí energii pryč od emitujícího objektu. Záření probíhá prostřednictvím vakua nebo jakéhokoli průhledného média (buď pevného nebo tekutého). Tepelné sálání je přímým výsledkem náhodných pohybů atomů a molekul v hmotě. Pohyb nabitých protonů a elektronů vede k emisi elektromagnetického záření.

Všechny materiály vyzařují tepelnou energii na základě své teploty. Čím je předmět teplejší, tím více bude vyzařuje. Slunce je jasným příkladem tepelného záření, které přenáší teplo napříč sluneční soustavou. Při normální pokojové teplotě vyzařují předměty infračervené vlny. Teplota předmětu ovlivňuje vlnovou délku a frekvenci vyzařovaných vln. S rostoucí teplotou se vlnové délky ve spektrech emitovaného záření zmenšují a vyzařují kratší vlnové délky se zářením o vyšší frekvenci.

## 3.4 Regrese

V práci budeme používat regresní modely s tím, že informace jsou převzaty z knih [1], [4] a [7]. Vyšetřujeme-li závislosti nějaké veličiny na teplotě měření v pracovním prostředí, nejlépe v laboratorních podmínkách, předpokládáme obvykle, že je popsána parametrickou funkcí určitého typu. Výsledky měření můžeme zobrazit jako množinu bodů  $[x_1, y_1], [x_2, y_2], \dots, [x_n, y_n]$  v rovině, kde na vodorovnou osu  $x$  vynášíme ve vhodném měřítku teploty měření jako hodnoty nezávislé proměnné veličiny a na svislou osu  $y$  vynášíme v jiném vhodném měřítku naměřené hodnoty veličiny, jejíž teplotní závislost studujeme. Z výsledků měření chceme vypočítat číselné hodnoty koeficientů a určit průběh grafu, který funkci znázorňuje, ve funkčním předpisu funkce  $Y = f(X, \beta) + \epsilon$ , kde  $X$  je vektor nezávislé proměnné veličiny,  $Y$  je vektor závislých veličin na  $X$ ,  $\beta$  jsou odhadované parametry funkce  $f$ ,  $\epsilon$  je vektor nezávislých náhodných veličin s rozdělením  $N(0, \sigma^2)$ .

Kdyby měření bylo naprosto přesné, ležely by všechny body zobrazující výsledky měření na grafu funkce. Pro každý by platilo  $y_i = f(x_i)$ . Měření je však zatíženo nestálým prostředím a vnesené body jsou podle jeho přesnosti více nebo méně rozptýleny okolo grafu, který skutečnou teplotní závislost popisuje. Jeho přesný průběh ale neznáme. Ze všech možných funkcí daného typu tedy hledáme tu, jejíž graf bude probíhat co nejlépe k vneseným bodům.

Pro tento úkol použijeme statistickou metodu nejmenších čtverců (3.7). Hledáme takové parametry  $\beta$  průběhu grafu, při kterých součet druhých mocnin svislých odchylek všech vnesených bodů do grafu je minimální. Metoda nejmenších čtverců je jedna z regresních metod.

$$S = \sum_{i=0}^n (f(x_i, \beta) - y_i)^2 \quad (3.7)$$

Nejběžnější regresní modely se zaměřují zejména na následující funkce jedné nezávislé proměnné  $x$ :

- $y = \beta_1 + \beta_2 x$  – lineární regrese,
- $y = \beta_1 + \beta_2 \cdot \ln(x)$  – logaritmická regrese,
- $y = \beta_1 e^{\beta_2 x}$  – exponenciální regrese,
- $y = \beta_1 x^{\beta_2}$  – mocninná regrese,
- $y = \beta_0 + \beta_1 x + \dots + \beta_n x_n$  – polynomičká regrese.

V práci odhadujeme parametry lineárních regresních modelů  $Y = \beta_1 X + \beta_2$  a exponenciálních regresních modelů vycházející z teorie oteplování elektrických přístrojů v závislosti na času a charakteristických vlastnostech přístroje (3.2), (3.6).

## 4 Technická realizace

Výpočet prostorové teploty budeme dělat rovnicí, jejíž parametry budou odhadnuty z regresních modelů. Abychom odhadli tyto parametry potřebujeme mít vektor naměřených bodů obsahující informace o vývoji teplot a podmínek, za kterých tyto teploty nastaly. Navrhujeme několik testů, při kterých budou tyto informace nasnímané. Pro návrhy testů použijeme dosavadních znalostí z teorie a ročníkového projektu[6], kde byl řešen náběh elektronického zařízení parametrickou rovnicí.

### 4.1 Návrh testů

Z běžného používání zařízení známe, že většinu času je zapnuté s vypnutým displejem a podle interakce s uživatelem se zapíná displej. Zařízení má na desce plošných spojů osazené periférie, které svojí aktivitou generují tepelnou energii. Ze zařízení není jednoduše rozpoznatelné, jaká periférie generuje kolik tepla, proto je zvoleno jednodušší rozdělení zařízení na dva objekty generující teplo – desku plošných spojů a displej.

V první řadě nejdůležitějším pozorováním je závislost teploty zařízení na teplotě vzduchu pracovního prostředí. Chceme pozorovat teplotu plně zahřátého zařízení s vypnutým a zapnutým displejem při konstantní prostorové teplotě. Mimo jiné chceme pozorovat vývoj teplot mezi těmito stavy. V neposlední řadě chceme sledovat vývoj teplot po zapnutí zařízení. Všechny tyto pozorování budeme provádět při různých prostorových teplotách a to několikrát, abychom získali dostatečné množství dat pro odhad parametrů regresních modelů.

Z rozdělení uvedeného v předchozím odstavci si zavedeme několik pojmů přesně definujících energetické hladiny při ohřevu a ochlazování zařízení. Tyto hladiny budeme nazývat *stavem* zařízení. Podle toho jestli je zařízení zapnuté, má rozsvícený či zhasnutý displej, tak zařízení přechází mezi těmito stavy.

Uvažujeme při definici těchto stavů, že se zařízení nachází v základním pracovním prostředí, to znamená, že prostředí nepůsobí žádným zvláštním vlivem na zařízení. Uvažujeme také, že teplota v tomto prostředí je neměnná. Vypnuté zařízení dlouhodobě umístěné v tomto prostředí mající totožnou teplotu jako je prostorová teplota prostředí budeme nazývat, že je ve *výchozím stavu*. Zapnutím zařízení se okamžitě mění jeho stav v přechodový stav mezi dvěma jinými stavy. Zapnutému zařízení s vypnutým displejem se zvyšuje teplota, jak se v něm akumuluje tepelná energie. Část vygenerovaného tepla se odevzdává do okolí. V moment kdy zařízení s vypnutým displejem veškerou aktuálně vygenerovanou energii vyzařuje do okolí

a nenavyšuje svou tepelnou energii, tak setrvává v rovnováze mezi generovaným teplem a tepelnou ztrátou do okolí. V této situaci zařízení dospělo do *konečného stavu bez displeje*. Zapnutím displeje se poruší rovnováha a vygenerované teplo je znovu vyšší než tepelná ztráta. V přechodovém stavu pozorujeme podobný jev jako jsme popsali v předešlých řádcích. Jakmile dojde k opětovné rovnováze, tak zařízení je v *konečném stavu s displejem*.

Ve stejnojmenném ročníkovém projektu[6] bylo řešeno hledání ideální funkce, která popisuje náběhovou křivku elektronického zařízení. Metodou nejmenších čtverců se porovnávaly různé modely a hledal se ten, který měl nejmenší chybu. U regresních modelů bylo zvoleno omezení na maximální počet pěti parametrů, které se odhadovaly.

Nalezená rovnice má předpis  $f(x) = b_1 + b_2 \cdot \arctan(b_3x) + b_4 \cdot \arctan(b_5x)$ . Kde  $x$  je čas od počátku ohřevu a  $f(x)$  je hodnota oteplení zařízení. Konkrétní parametry rovnice nejsou důležité, ale uvedeme tu, že parametry jednoho arctan několika násobně vyšší než druhého. Jeden arctan definoval hlavní nárůst teplot a druhý mírně hodnoty upravoval pro lepší přesnost. Tato rovnice sice velmi dobře reprezentuje náběh řešení v ročníkovém projektu, avšak není vhodná pro aktuální použití.

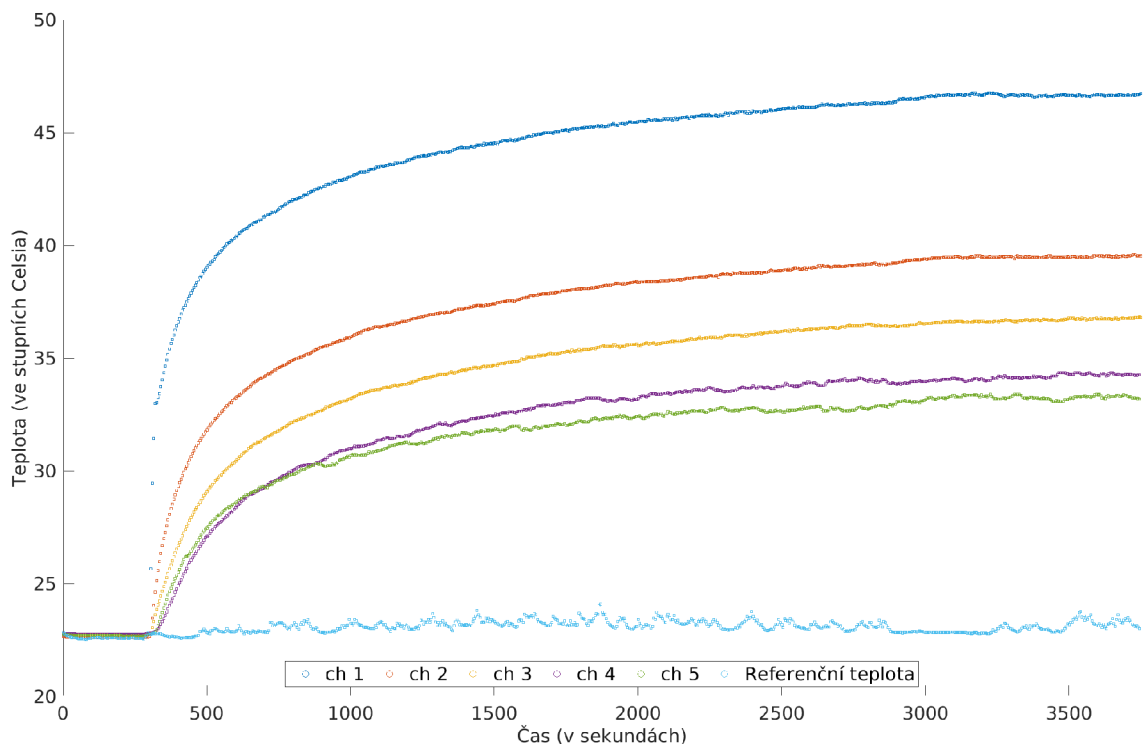
V projektu byla na desku plošných spojů přilepena externí teplotní čidla, kdežto nyní jsou čidla osazena přímo na desce. V projektu byla hledaná rovnice s nejmenší chybou vůči naměřeným hodnotám, ale nebral se ohled na tepelnou charakteristiku zařízení  $\tau$  z rovnic (3.2), (3.6). Tím že nebyl brán ohled na tepelnou charakteristiku, tak se veškeré parametry  $b_i$  mění v závislosti na tom, v jaké teplotě byl náběh měřen. Dále rovnice (3.2), (3.6) mají menší počet parametrů. Při náběhu hledáme pouze  $\tau$ . Parametr  $\Delta\vartheta_\infty$  získáme z rozdílu maximálního oteplení zařízení a prostorové teploty.

Z projektu nás zajímá náběhová křivka z obrázku 4.1. Teplotní křivka byla měřena z *výchozího do konečného stavu s displejem*. Data oteplovací křivky byla měřena po dobu 57 minut.

Z naměřených dat můžeme prohlásit, že nebylo dosaženo *konečného stavu s displejem*, jelikož se v posledních minutách měření teplota stále zvyšovala. Dále víme, že za posledních 30 minut záznamu se zařízení ohřálo o  $1,05^\circ\text{C}$  na celkové oteplení o  $10,6^\circ\text{C}$  od počátku. Kdyby měření pokračovalo další hodinu, tak by se zařízení ohřálo maximálně o  $2,1^\circ\text{C}$ . Počítáme-li s tím, že se jedná o exponenciálu, které se s navyšujícím časem snižuje teplotní nárůst, tak oteplení zařízení bude výrazně nižší než  $2,1^\circ\text{C}$ . Z grafu je taktéž patrné, že oteplení se v posledních minutách výrazně snížilo. Proto budeme předpokládat, že po dvouhodinovém měření se celkové oteplení zařízení  $\Delta\vartheta$  bude blížit 95. percentilu maximálního oteplení  $\Delta\vartheta_\infty$  v řádu desetin stupňů Celsia. Této informace využijeme při návrhu testů pro měření vstupních dat pro regresní model.

Podle tohoto předpokladu bude jedno měření ohřevu respektive ochlazení probíhat 2 hodiny. Testy chceme pokrýt většinu běžných prostorových teplot a standardní chování zařízení. Testy zvolíme následovně: displej zařízení bude periodicky po dvou hodinách zapínán a vypínán a při testech bude zařízení několikrát ručně vypnuto.

Během těchto testů bude měněna teplota v místnosti přibližně o  $1^\circ\text{C}$  v rozsahu od  $16^\circ\text{C}$  do  $27^\circ\text{C}$ . Tato volba teplot bude vysvětlena později. Teplota v místnosti bude změněna maximálně jednou denně. Těmito testy získáme minimálně 6 oteplo-



Obrázek 4.1: Graf naměřených dat z ročníkového projektu, ch – čidla rozmístěná různě po desce plošných spojů

vacích a 6 ochlazovacích křivek pro jednu konkrétní prostorovou teplotu a jedno zařízení. Měřením chceme získat dostatek dat pro nalezení parametrů regresních modelů popisujících teplotní charakteristiku. Měření bude probíhat na třech zařízeních ve třech různých místnostech.

Dalším aspektem měření je volba vzorkovací frekvence. Při relativně nízké vzorkovací frekvenci získáme řídké informace o průběhu měření. Pojmeme řídká data zde myslíme, že mezi dvěma po sobě jdoucími vzorky je dlouhá prodleva a proto může být relativně velký teplotní rozdíl mezi vzorky. Velký rozdíl pro nás znamená nepřesnost při hledání regresního modelu. Největší nepřesnost by se vyskytovala u náhlých změn například u náběhu ohřevu zařízení z *výchozího* stavu. Tuto oblast chceme mít relativně dobře zmapovanou. Naopak při relativně vysoké frekvenci zápisu dat se nám znatelně zvětšují datové soubory, data bude obtížnější zpracovat kvůli jejich hustotě.

Zvolili jsme, že jeden bude trvat 2 hodiny. Měřením po každé sekundě bychom získali velmi přesné informace o náběhu, ale měli bychom zbytečně moc dat. Proto zvolíme vzorkování po 10 sekundách, kde 720 vzorků pro jeden náběh je dostatečné množství.

Měřením chceme získat následující parametry:

- časovou značku,
- teploty zařízení,

- prostorovou teplotu,
- stav displeje.

Zařízení bude při měření namontované v elektroinstalační krabici, aby data co nejlépe reprezentovaly reálné nasazení. Volba umístění zařízení v prostoru, přesněji v elektroinstalační krabici, vychází z rozdílných akumulovaných energií závislých na volbě prostředí. Pokud zařízení není v elektroinstalační krabici, tak můžeme očekávat rozdílné maximální oteplení  $\Delta\vartheta_\infty$ . Změnou prostředí se mění velikost tepelné ztráty do okolí. Rozdílná tepelná ztráta potom znamená, že se mění velikost akumulované energie v zařízení a tím i jeho maximální oteplení  $\Delta\vartheta_\infty$ .

V našem použití v elektroinstalační krabici do soustavy vstupuje role stěny jako chladičím prvku. Funkcí stěny jako chladiče ztrácí zařízení více tepelné energie než volně položenému zařízení v prostoru. V opačném případě zařízení poklesne velikost energie předané do okolí a větší část tepelné energie bude akumulovat v sobě. To bude mít za následek vyšší teploty zařízení a tím pádem i vyšší teploty měřené čidly.

Veškeré tyto informace vychází z rovnice (3.4). Změnou polohy měníme veličinu součinitele přenosu tepla  $\alpha_0$  a ochlazovací povrch tělesa  $A$ . Změnou polohy se mění veličina  $\tau$ , kterou chceme odhadovat.

Pro ověření před sběrem dat pro regresní model byly provedeny testy, jak se mění maximální oteplení zařízení v závislosti na jeho poloze. Testy byly provedeny ve stejném pracovním prostředí s konstantní teplotou. Při testech byla měněna lokace zařízení. Testy byly provedeny se zařízením umístěným na stěně, zařízením položeným na stole displejem směrem nahoru (položeno o malou plochu napájecího konektoru), položeným na stole displejem dolů, opřným na stole o stěnu místnosti. Všechny testy probíhaly v podobné výšce od podlahy. Získané maximální teploty těchto testů se lišily i o jednotky stupňů Celsia a zařízení nemá žádný způsob pro rozpoznání rozdílu mezi polohami zařízení z měřených teplot. Proto měření probíhá v podmínkách reálného nasazení a nebyla zvolena ani možnost měření v termokomoře. Odhad teplot při odlišné instalaci v této práci neřešíme.

Rozsah zvolených teplot v místnosti vychází z maximální a minimální teploty, které bylo možné v místnosti dlouhodobě udržet. Zařízení byla rozmístěna v odlišných místnostech. Všechny prostory mají podlahové vytápění. Ve všech místnostech byla zavřena okna a dveře pro minimalizaci externích vlivů na změnu prostorové teploty.

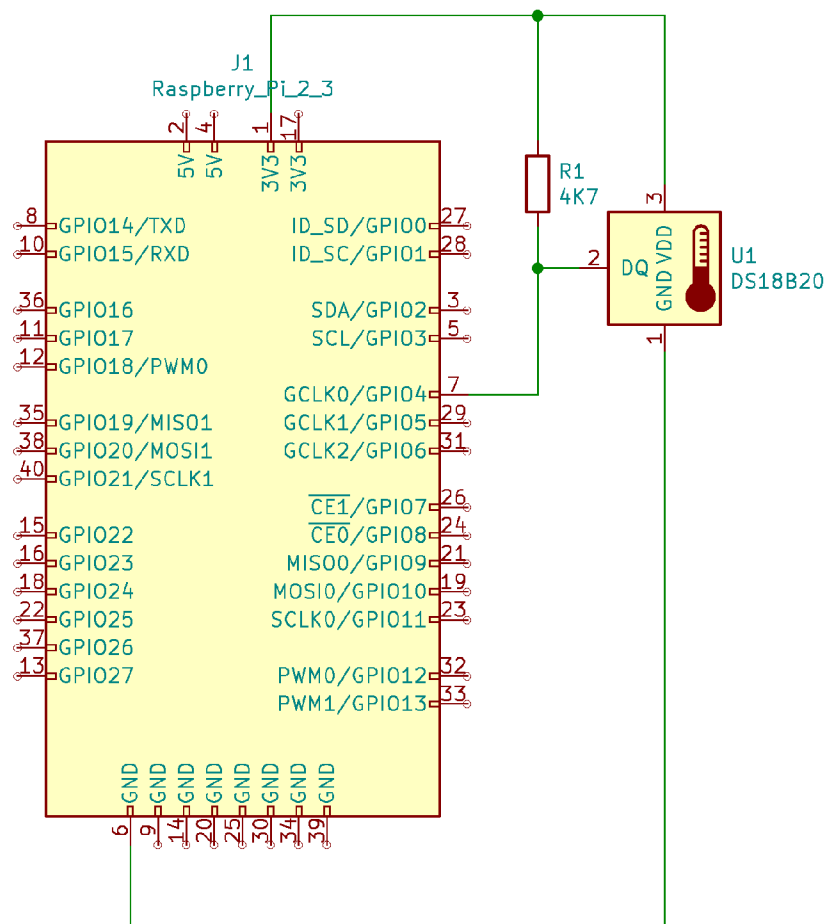
Každé vytápění má své omezení na dosažitelný ohřev. Měření probíhalo v zimní období a při těchto podmínkách byla minimální teplota v místnosti 16 °C. Přestože venkovní teplota byla kolem nuly, tak nebylo možné dosáhnout nižších teplot. Hlavním důvodem bylo vytápění okolních místností rezidenčních prostorů. V opačném případě plným vytápění byla dosažena teplota 27 °C, což byl limit generovaného tepla kotlem.

## 4.2 Měřicí soustava

Při sestavování aparatury pro měření teploty elektronickým zařízením použijeme dosavadní znalosti z předchozích kapitol. Časovou značku, teploty zařízení a stav displeje získáme ze zařízení. Jako poslední údaj nám zbývá měření prostorové teploty. Referenční prostorovou teplotu není možné měřit na zařízení v aktuálním provedení.

Prostorovou referenční teplotu získáme z externího přístroje. Do soustavy přidáme minipočítač Raspberry Pi 3B+.

Raspberry Pi je nízko-nákladový jednodeskový počítač malých rozměrů velikosti podobný mobilnímu telefonu. Raspberry Pi jsou vybaveny systémem Broadcom na čipu s integrovanou ARM kompatibilní centrální procesorovou jednotkou a grafickou procesorovou jednotkou na čipu. Raspberry Pi je vhodný pro takzvané udělej si sám projekty. Předností zařízení je dobře zdokumentovaná softwarová podpora, jednoduchá instalace operačního systému vlastní volby a univerzální vstupní/výstupní pinový hřeben, na který je možné připojit velké množství různých periférií. Je vhodné poznamenat, že univerzální vstupní/výstupní piny jsou pouze digitální, minipočítač nemá žádné analogově digitální převodníky. Technickou dokumentaci nalezneme na



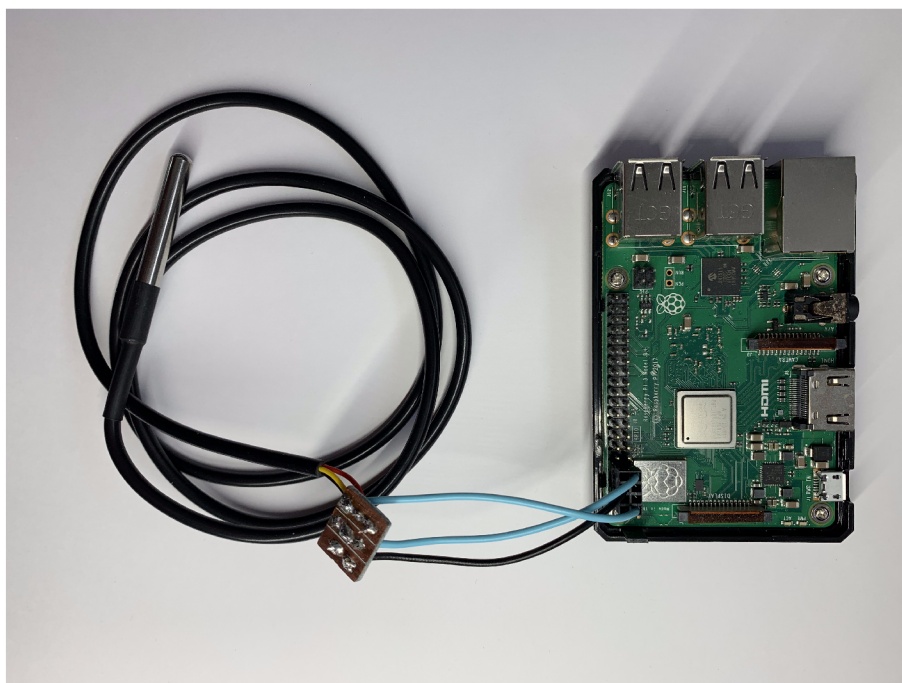
Obrázek 4.2: Schéma zapojení DS18B20 k Raspberry Pi

oficiálních stránkách výrobce [5].

Do univerzálních vstupních/výstupních pinů Raspberry Pi zapojíme digitální teploměr DS18B20. DS18B20 je jednovodičový programovatelný teplotní senzor, který odolává i v drsném prostředí. Dokáže měřit široký rozsah teplot od  $-55\text{ }^{\circ}\text{C}$  do  $125\text{ }^{\circ}\text{C}$  s přesností  $0,125\text{ }^{\circ}\text{C}$ . Každý senzor má jedinečnou adresu a vyžaduje pouze jeden pin mikrokontroléru pro přenos dat. Senzor má celkem tři vodiče, zmíněný datový s pull-up rezistorem, zbylé jsou použity pro napájení. Provozní napětí je od 3 V do 5 V. Rozlišení výstupu lze nastavit s přesností na 9 až 12 bitů. Doba konverze pro přesnost nastavenou na 12 bitů trvá 750 ms.

Řekněme, že máme pro datový vodič senzoru nastavený jeden pin z Raspberry Pi jako vstupní. Když není na pin nic připojeno, je těžké určit zda na pinu je vysoký (VCC) nebo nízký (GND) stav. Jevu říkáme plovoucí stav. Aby se tomuto neznámému stavu zabránilo, pull-up nebo pull-down rezistor zajistí, že pin je buď ve vysokém nebo nízkém stavu, a zároveň použije nízké množství proudu. Proto na datový vodič přidáme  $4,7\text{ k}\Omega$  pull-up rezistor. Pro napájení teploměru použijeme 3,3 V VCC pin Raspberry Pi. Schéma zapojení viz obrázek 4.2.

Jelikož měření chystáme provozovat na dvou separátních zařízeních zařídíme k tomu komunikaci mezi jednotkami pro výměnu dat. Obě zařízení disponují bezdrátovou komunikací Wi-Fi. Pro rozlišení jednotek zavedeme pro zařízení, na kterém chceme odhadovat prostorovou teplotu z čidel na desce plošných spojů, jednoznačné pojmenování – DPS. V základních informacích o DPS v kapitole 2 jsme uvedli, že DPS po spuštění automaticky vytváří bezdrátový přístupový bod do vlastní sítě. Toho využijeme a nastavíme Raspberry Pi, aby se po spuštění připojilo k tomuto přístupovému bodu a žádnému jinému. Takto zajistíme komunikaci pro výměnu dat



Obrázek 4.3: Raspberry Pi s připojeným digitálním teploměrem DS18B20



mezi jednotkami.

Pro Raspberry Pi byl zvolen operační systém Raspbian verze 10 (buster).

Přes terminál upravíme soubor síťových rozhraní lokalizovaný v `/etc/network/interfaces`, aby se zařízení automaticky připojilo k dostupné síti a aby se použil konfigurační soubor `wpa_supplicant`. Následně upravíme konfigurační soubor pro síťové připojení, `/etc/wpa_supplicant/wpa_supplicant.conf`, tak že zde zapíšeme údaje o přístupovém bodu, ke kterému má být zařízení připojeno. IP adresu Raspberry Pi nastavíme jako statickou, aby nedošlo k její změně po odpojení ze sítě.

### 4.3 Software měřicí soustavy

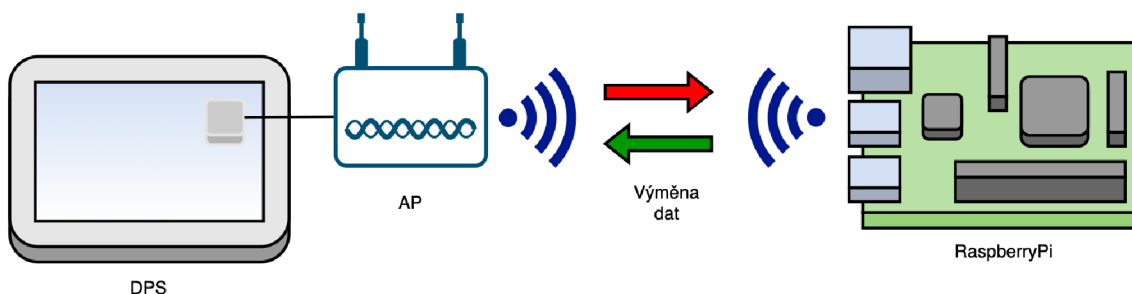
Z informací o teplotním senzoru DS18B20 víme, že konverze trvá 750 ms. Připočítáme-li k tomu zpoždění v komunikaci, tak můžeme vzorkovací frekvenci čidla zaokrouhlit na 1 Hz. Jelikož se jedná o takzvané úzké hrdlo v měření, protože konverze je jeho nejpomalejším článkem, tak pro Raspberry Pi napíšeme samostatný logovací program. Program se v cyklu dotazuje čidla na teplotu – přesněji vyčítá informace ze souboru vzniklého operačním systémem, když se čidlo připojilo, konkrétně na cestě `/sys/bus/w1/devices/<sensor id>/w1_slave`. Do souboru se zapisují data podobná:

```
buf: d9 00 4b 46 7f ff 0c 10 2e : crc=2e YES
d9 00 4b 46 7f ff 0c 10 2e t=23562
```

Z toho se vybere pouze hodnota teploty 23562 a převede se z milistupňů Celsia na stupně Celsia a vypíše se na standardní výstup. Teplotní čidlo je uchyceno ve stejné výšce jako DPS pár centimetrů od jednotky.

Hlavní logovací program byl napsán pro DPS, jakožto hlavního činitele měření. Program čte data ze tří souborů reprezentujících tři teplotní čidla DPS. Tyto soubory pracují na odlišném principu než soubor vzniklý z připojeného čidla DS18B20 do Raspberry Pi. U těchto souborů je prodleva při jejich čtení téměř nulová, protože jsou zakomponovány přímo do systému desky. Prodlevu lze zanedbat. Získání teploty z čidel je tedy okamžité oproti DS18B20.

Pro vzdálenou komunikaci s Raspberry Pi je aplikace napsána jako vícevláknová. Hlavní vlákno řeší zápis získaných dat do souborů. Vedlejší vlákno má na starosti



Obrázek 4.4: Diagram bezdrátové komunikace mezi zařízeními

komunikaci s Raspberry Pi.

Aplikace spouští v novém terminálu `ssh` příkaz s IP adresou Raspberry Pi pro navázání spojení. Spojení se kontroluje, přesněji je nastaven časový limit jedné sekundy, kdy pokud je výpadek v komunikaci delší jak tento limit, tak se `ssh` příkaz ukončí. Pokud je spojení přerušeno, aplikace se snaží navázat nové. Při úspěšném spojení se ke vzdálenému zařízení – Raspberry Pi – spustí výše zmíněný logovací program určený pro Raspberry Pi, ten který vypisuje referenční teplotu na standardní výstup. Při logování se zapisuje poslední známá referenční teplota.

Při měření zaznamenáváme parametry ve formátu

```
<cas>;<senzor 1>;<senzor 2>;<senzor 3>;<reference>;<stav  
displeje>;
```

K dispozici byly tři DPS, ale pouze dvě Raspberry Pi. Jedna deska plošných spojů byla upravena, aby měřila i prostorovou teplotu. Toho bylo docíleno tak, že čidlo, které je nejbližší procesoru a proto nabývalo nejvyšších teplot, bylo vyvedeno mimo desku plošných spojů volně do prostoru na drátkách dlouhých 8 cm. Tím je zamezen ohřev čidla perifériemi zařízení. Pro tuto desku byl logovací program příslušně upraven. Nemusí v něm být komunikace s Raspberry Pi.

## 4.4 Příprava dat

Celkem bylo nasnímano 413 163 bodů v 418 testech. Graf všech nasnímaných bodů je v obrázku 4.10. Než začneme řešit jednotlivé matematické úlohy zpracujeme si vstupní data do formátu, se kterým bude jednodušší pracovat. Naměřená data máme uloženy ve dvou různých formátech, protože k měření docházelo dvojím způsobem. Teploty měřené čidlem 3 odstraníme, protože u DPS bez propojení s Raspberry Pi je zapsaná pouze teplota 126 °C a u DPS bez propojení se jedná o nejteplejší čidlo, které neplánujeme při výpočtech použít. Hodnoty z nejteplejšího čidla si můžeme dovolit odstranit, protože je nejvíce ovlivněno samoohřevem zařízení. Ve výpočtech použijeme teploty z čidla s nižší měřenou teplotou, jelikož je nejméně ovlivněno samoohřevem a na teplotě bude nejvíce patrná změna teploty prostředí. Formát dat je v tabulce 4.1.

Po zapnutí zařízení mimo toho, že chybí historie, tak není ani synchronizovaný systémový čas. Po zapnutí je systémový čas nastaven na nulovou hodnotu a to na 00:00:00, 1. ledna 1970. Po synchronizaci zařízení dostane aktuální čas.

Podle toho zda je zařízení připojeno do lokální bezdrátové sítě s přístupem na internet, tak si zařízení může nebo nemusí získat aktuální čas. Nabytí nového systémového času nenastává hned po zapnutí a proto můžeme vidět v ukázce naměřených dat v tabulce 4.1 v prvních několika řádcích čas, který se výrazně liší posledních řádků. To se děje právě kvůli synchronizaci času, kde si zařízení získá reálný čas.

V prvním řádku tabulky 4.1 vidíme, že teploty jsou 126,00 °C. Jedná se o údaj, který značí, že nebylo možno z čidla vyčíst teplotu, nebo že je vadné nebo ještě není připraveno k vyčtení jeho teploty.

Řádky které obsahovaly zmíněnou teplotu 126,00 °C odstraníme. Čas vzorku

Čas	Čidlo 1	Čidlo 2	Čidlo 3 [Ref]	Displej
01-00:02:29	126,00	126,00	126,00	1
01-00:02:38	26,30	30,70	22,40	1
01-00:02:47	26,30	30,90	22,40	1
01-00:02:55	26,40	31,20	22,50	1
01-00:03:04	26,40	31,30	22,50	1
01-00:03:13	26,40	31,50	22,50	1
25-16:40:19	26,40	31,70	22,50	1
25-16:40:27	26,50	31,90	22,50	1
25-16:40:35	26,50	32,00	22,50	1

Tabulka 4.1: Ukázka nasnímaných dat

přepočteme z časového formátu `dd-hh:mm:ss` na dobu běhu od spuštění zařízení v sekundách. Problematická je synchronizace času a vzniklý časový skok v záznamu.

Provedeme rozdíl času  $t_{s_n} = t_{n+1} - t_n$ . Hodnoty  $t_{s_n}$  zobrazíme v histogramu. V histogramu na datech mají nejvíce vzorků sloupce 8 a 9 sekund, jedná se o vzorkovací frekvenci. Vzorkovací frekvence je nižší než byla zvolena kvůli prodlevám při získávání referenční teploty z Raspberry Pi. Ostatní hodnoty odstraníme, jelikož se jedná o časové skoky. Ze zbylých hodnot vypočteme průměrný čas mezi záznamy, to činí 8,69 sekundy. Touto hodnotou přepočteme hodnoty před časovým skokem.

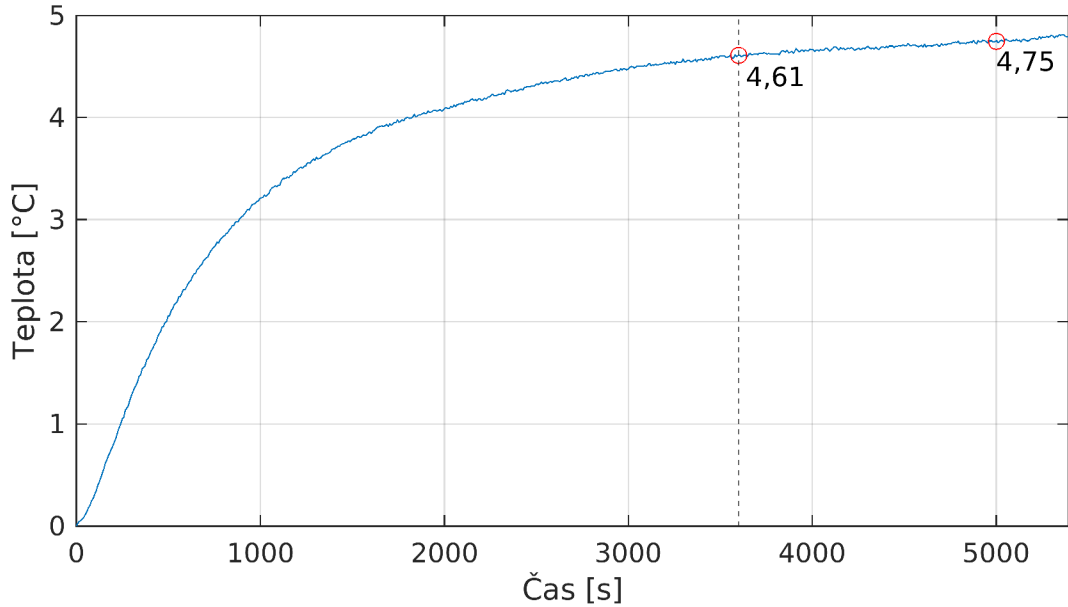
Zavedeme časovou konstantu  $t_{st}$ , kdy považujeme teplotu zařízení za ustálenou. Protože při měření byl zvolený časový interval dvou hodin pro změnu stavu displeje, máme k dispozici pouze tyto dvě hodiny dat po startu zařízení. Z dat vybereme pouze ty vzorky, které byly zaznamenány při náběhu zařízení dokud není v *konečném stavu bez displeje*.

Data rozdělíme do samostatných náběhových křivek. Soubor křivek označme  $S$ , kde  $S_i$  pro  $i = 1, 2, \dots, n$  jsou jednotlivé křivky z  $S$  a  $S_{i,k}$  je konkrétní vzorek  $k$  křivky  $i$ . Ze všech vzorků každé křivky odečteme hodnotu prvního vzorku, abychom dostali velikost oteplení  $\Delta\vartheta$ . V posledním kroku křivky zprůměrujeme.

$$V = \frac{\sum_{i=1}^n [S_{i,1} - S_{i,1}, S_{i,2} - S_{i,1}, \dots, S_{i,k} - S_{i,1}]}{n}$$

Výsledná obecná křivka ohřevu zařízení  $V$  je znázorněna v obrázku 4.5. Tato křivka není závislá na teplotě zařízení, ani nevznikla z protříděných dat, a proto ji nelze použít k odhadu regresních parametrů. Avšak nám poslouží k identifikaci časového intervalu, po kterém prohlásíme teploty za ustálené.

V obrázku jsou označeny dva body. Jeden v čase 3 600 sekund – jedna hodina – s nárůstem od počátku o teplotu 4,61 °C. Druhý v čase 5 000 sekund s nárůstem o teplotu 4,75 °C. V grafu pozorujeme, že se křivka zplošťuje a rychlost nárůstu teploty se snižuje. Porovnáme-li křivku obecného náběhu zařízení z obrázku 4.5 s teplotní charakteristikou obecného zařízení v obrázku 3.1, tak víme, že nárůst teploty se ustálí po dlouhém časovém intervalu  $t \rightarrow \infty$ . Jelikož byla zvolena délka



Obrázek 4.5: Průměrná oteplovací křivka elektronického zařízení po jeho zapnutí

jednoho cyklu stavu displeje na dvě hodiny a vzhledem k postupně se snižující rychlosti nárůstu teploty, tak zvolíme, že po jedné hodině běhu zařízení, respektive jedné hodině po změně stavu displeje, považujeme teplotu za ustálenou. To znamená, že časová konstanta  $t_{st} = 3600$ . Teplotní charakteristiky po čase  $t \geq t_{st}$  interpretujeme jako kdyby  $t = t_{\infty}$ .

Aplikací dostaneme pro oteplovací křivku

$$\Delta\vartheta = \begin{cases} (1 - \vartheta_{\infty}) \cdot e^{-t/\tau}, & \text{pro } t < 3600 \\ \vartheta_{\infty}, & \text{pro } t \geq 3600 \end{cases} \quad (4.1)$$

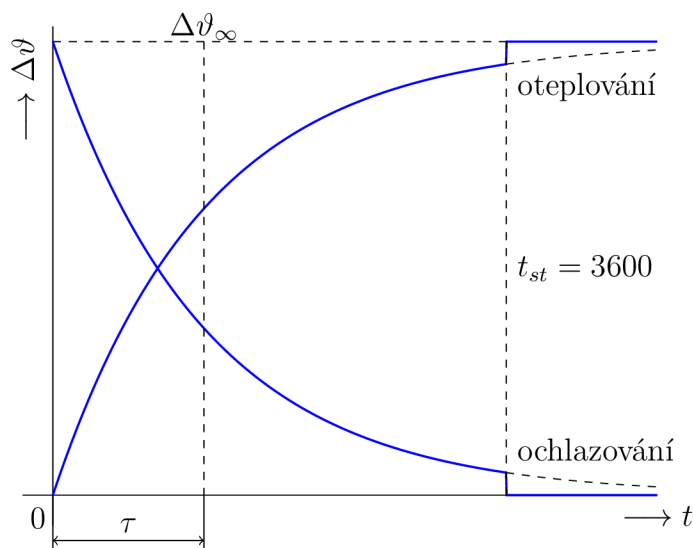
a pro ochlazovací křivku

$$\Delta\vartheta = \begin{cases} \vartheta_{\infty} \cdot e^{-t/\tau}, & \text{pro } t < 3600 \\ 0, & \text{pro } t \geq 3600 \end{cases} \quad (4.2)$$

Průběh je graficky znázorněn v obrázku 4.6. Použitím této metodiky docílíme zjednodušení výpočtu, protože se nám zmenší interval, na kterém by výpočet oteplování respektive ochlazování probíhal.

Poté co jsme definovali časovou konstantu  $t_{st}$ , tak můžeme přejít k separaci jednotlivých oteplovacích a ochlazovacích křivek z datasetu. Do tabulky záznamů 4.1 doplníme nový sloupec *Ustáleno*. Hodnota v něm značí zda proběhl jeden cyklus displeje delší jak  $t_{st}$  a stav displeje je minimálně po dobu  $t_{st}$  neměnný. Pro jednodušší interpretaci je znázorněno v obrázku 4.7. Pokud jsou podmínky splněny, pak v daném řádku je hodnota sloupce rovna 1. V opačném případě je ve sloupci hodnota 0.

Data rozdělíme do jednotlivých křivek v momentech, kdy je změněn stav displeje. Vybereme pouze ty křivky, které mají ve všech bodech hodnotu sloupce *Ustáleno*



Obrázek 4.6: Upravená oteplovací a ochlazovací charakteristika

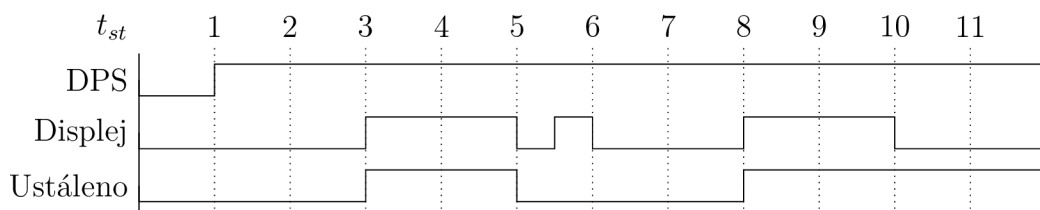
rovnou 1. U křivek provedeme úpravu teplot DPS, že odečteme od všech hodnot minimální teplotu. Tím získáme jak pro oteplovací, tak pro ochlazovací křivky velikost oteplení zařízení od předešlého stavu.

Při promítnutí vybraného intervalu záznamů do grafu znázorněných v obrázku 4.8 pozorujeme nestabilitu referenční teploty ve formě její oscilace. Tato oscilace se vyskytuje jen v některých částech nasnímaných dat. Jedná se o jev, který pravděpodobně nastal nedokonalým vytápěním místnosti. Především jev nastává u relativně vysokých teplot v místnosti.

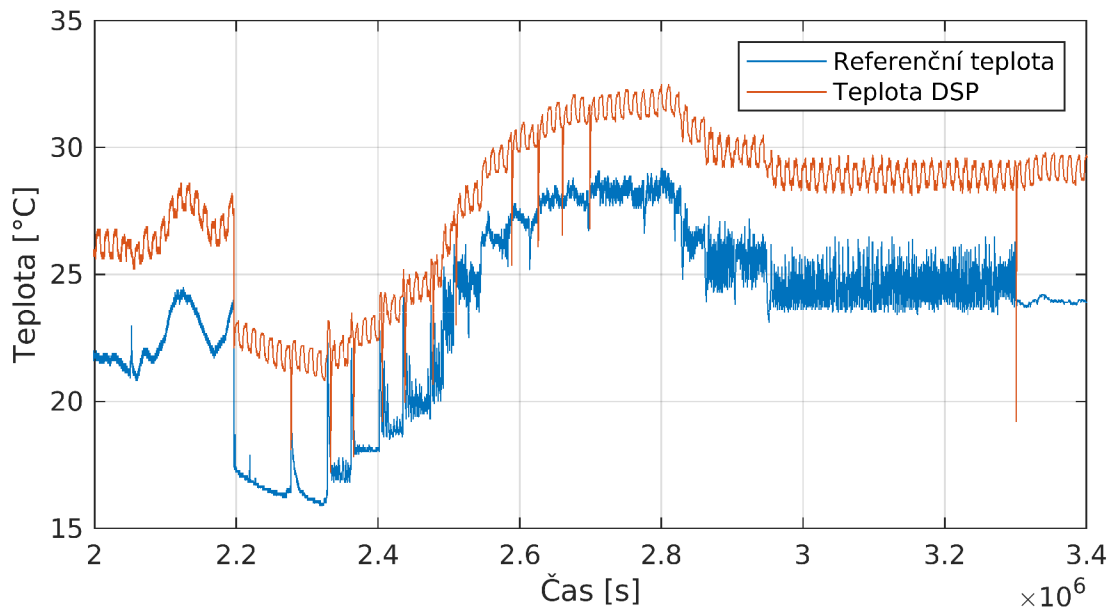
Logická jednotka kotle, která určuje jak se vytápí místnosti, nedokázala dlouhodobě udržet stálou teplotu v místnosti. Kotel je nejprve v režimu vytápění. Jakmile je vytopeno na požadovanou teplotu, tak kotel omezí přísun teplé vody do vytápěcí soustavy a teplota v místnosti začne klesat. Jakmile teplota klesne pod limit určeným kotlem, že je zapotřebí opět topit, tak do oběhu začne zpět dodávat teplou vodu. Tento proces se několikrát opakoval.

Naměřená data v momentech, kdy docházelo k oscilaci, nejsou vhodná pro použití při výpočtu regresních rovnic. Křivky je nutno systematicky odstranit. Podle rozptylu referenčních teplot v křivkách vyřadíme nežádoucí.

Teploty měřené čidly na DPS mají velikost nejmenšího kroku  $0,1\text{ }^{\circ}\text{C}$ . To je velmi hrubé rozlišení pro referenční teplotu, pokud ji chceme použít v matematickém mod-



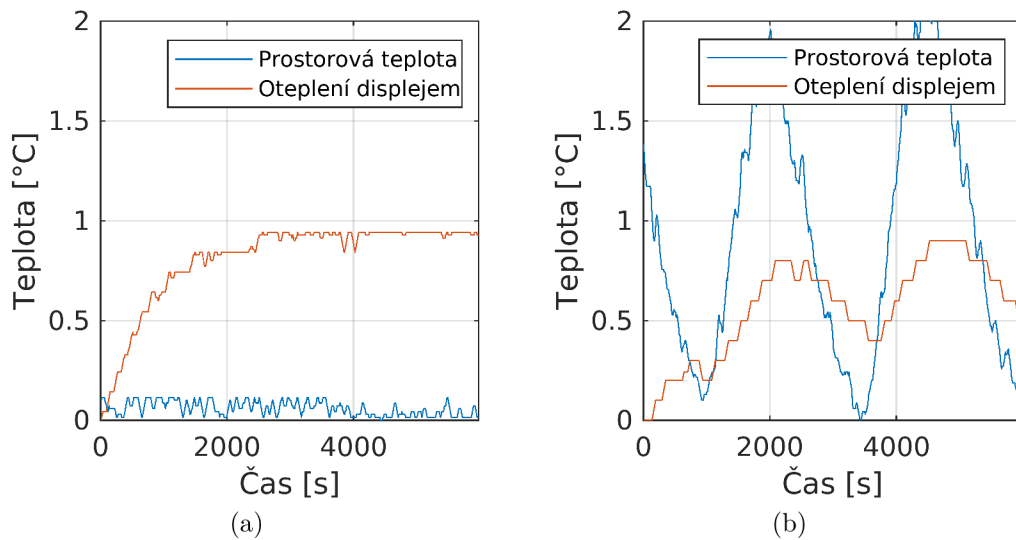
Obrázek 4.7: Graf znázorňující, kdy *Ustáleno* nabývá hodnoty 1



Obrázek 4.8: Výřez datasetu, kde nastává oscilace referenční teploty

elu. A to především u změn teplot displejem. Když se zaměříme na křivku DPS v grafu 4.8, tak vidíme jak se při změně stavu displeje mění o přibližně  $1\text{ }^{\circ}\text{C}$ . S nejmenším krokem  $0,1\text{ }^{\circ}\text{C}$  to dělá rozlišení 10 kroků na změnu displeje. Malý počet kroků pro nás znamená velkou nepřesnost při hledání regresní rovnice. Pro zlepšení rozlišení provedeme na datech aplikaci klouzavého průměru o velikosti tří vzorků.

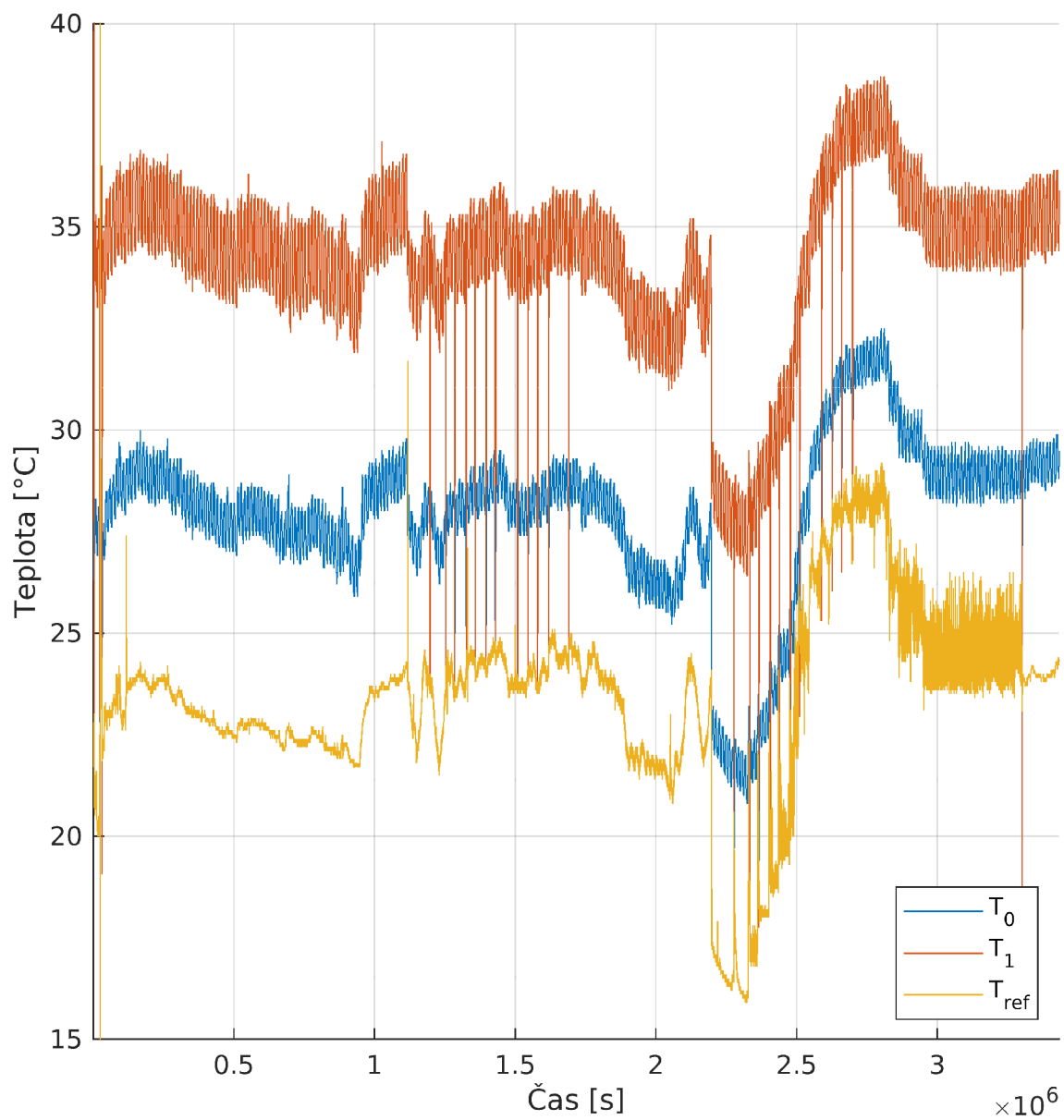
Data referenčních teplot si označme  $T_{ref}$  a nejchladnější čidlo  $T_0$ . V křivkách



Obrázek 4.9: Ukázka nasnímaných oteplovacích křivek displeje, a – naměřené oteplení při konstantní prostorové teplotě, b – naměřené oteplenění za oscilace prostorové teploty

nalezeneme minima a maxima pro  $T_{ref}$  a  $T_0$ . Od každého vzorku odečteme jejich minimální hodnotu, abychom pro  $T_0$  získali oteplení a ochlazení vzniklé displejem a z  $T_{ref}$  viděli rozptyl teploty v místnosti při pokusu.

Vybereme dvě specifické náběhové křivky displeje, jednu bez vlivu oscilace a druhou pod jejím vlivem. Vybrané křivky jsou v obrázku 4.9. V grafu 4.9a vidíme křivku při konstantní teplotě. Její tvar se shoduje s tepelnou charakteristikou z obrázku 3.1. Ovšem v grafu 4.9b je znatelně pozorovatelné působení rychlé změny teploty v místnosti na elektronické zařízení. Křivky, které se nepodobají tvarem charakteristickým křivkám, odstraníme. Pro každou křivku si stanovíme maximální možný rozptyl referenční teploty, tak aby  $T_{ref,max} - T_{ref,min} \leq 0,5$ . Tímto jsme eliminovali většinu negativních vlivů pro výpočet teplotní charakteristiky. Po aplikaci filtru zůstaly data z 193 testů. Celkem 158 839 bodů.



Obrázek 4.10: Veškeré naměřené body;  $T_0$  – teplota prvního čidla na zařízení;  $T_1$  – teplota druhého čidla na zařízení;  $T_{ref}$  – prostorová teplota



## 5 Statistické řešení

Cílem této práce je naprogramování algoritmu, který přepočítává měřenou teplotu zařízení v reálném čase na prostorovou teplotu. Algoritmus se zakomponuje do operačního systému v zařízení, to znamená že v souborovém stromě systémových periférií bude vytvořený soubor, jehož čtením získáme odhadovanou prostorovou teplotu. Jako vstupní parametry algoritmu máme teplotu zařízení  $T$ , stav displeje a čas běhu zařízení.

Funkce algoritmu budou napsané v programovacím jazyce C. Skupina funkcí algoritmu řeší odhad prostorové teploty  $T_e$  z ustálených teplot zařízení s vypnutým displejem  $\vartheta_{\infty}^{\text{OFF}}$ .

Ustálené teploty  $\vartheta_{\infty}$  jsou takové, které se při konstantní prostorové teplotě  $T^{\text{ref}}$  nemění v závislosti na čase. Jedná se o maximální naměřené teploty zařízení s vypnutým  $\vartheta_{\infty}^{\text{OFF}}$  a zapnutým displejem  $\vartheta_{\infty}^{\text{ON}}$  při konkrétní prostorové teplotě  $T^{\text{ref}}$ .

Změnou displeje (vypnutí/zapnutí) nastává přechodový stav, ve kterém se teploty zařízení odchyľují od ustálených teplot  $\vartheta_{\infty}$ , protože se zařízení otepluje respektive ochlazuje v závislosti na době  $t_1$  zapnutého respektive vypnutého displeje. Přechodový stav se ukončí v okamžiku, kdy dojde k ustálení teplot.

V přechodovém stavu počítáme korekci oteplení zařízení displejem  $\Delta\vartheta^{\text{BL}}$  a korekci teploty nedokončeným ohřevem po zapnutí zařízení  $\Delta\vartheta_i^{\text{S}}$  do ustálené teploty. Odečtením korekce oteplení displejem  $\Delta\vartheta_i^{\text{BL}}$  a přičtením korekce nedokončeného náběhu  $\Delta\vartheta_i^{\text{S}}$  k aktuálně měřené teplotě  $T_i$  v přechodovém stavu získáme ustálenou teplotu při vypnutém displeji  $\vartheta_{\infty}^{\text{OFF}} = T_i - \Delta\vartheta_i^{\text{BL}} + \Delta\vartheta_i^{\text{S}}$ .

Parametry rovnic charakterizující toto chování odhadujeme regresními modely. Korekci na prostorovou teplotu  $T_e$  při ustálených teplotách  $\vartheta_{\infty}^{\text{ON}}, \vartheta_{\infty}^{\text{OFF}}$  odhadujeme parametry  $\beta_1, \beta_2$  lineárním regresním modelem (5.1). Kompenzaci relativního oteplení zařízení displejem  $\Delta\vartheta_i^{\text{BL}}$  a korekci náběhu  $\Delta\vartheta_i^{\text{S}}$  v přechodovém stavu řešíme odhadem parametru  $\beta_3$  v nelineárním exponenciálním regresním modelu (5.2).

$$\Delta\vartheta_i = T_i - T_i^{\text{ref}} = \beta_1 T_i + \beta_2 \quad (5.1)$$

$$\Delta\vartheta_i^{\text{BL}} = 1 - \exp\left(-\frac{t_1^i}{\beta_3}\right) \quad (5.2)$$

Exponenciální regresní model je odvozen od oteplovací rovnice (3.2) a značení odhadovaného parametru  $\beta_3$  nahradíme parametrem  $\tau$ . Pro bodové odhady parametrů  $\beta_1, \beta_2, \tau$  použijeme metodu nejmenších čtverců. Parametr  $i$  značí jeden konkrétní naměřený bod.

Při korekci teploty náběhu nemáme k dispozici čas  $t$  z rovnice (3.2), jelikož při vypnutí zařízení ztrácíme veškeré dosavadní informace kromě teploty zařízení  $T$ . Neznáme tedy o kolik stupňů Celsia je zařízení ohřáto oproti prostorové teplotě a nevíme jak dlouho tento ohřev probíhal. Při náběhu měříme, jak dlouho trvá nárůst teploty na zařízení o definovanou teplotu ( $0,5^\circ\text{C}$ ) a odhadujeme parametr  $\tau$  exponenciálního regresního modelu, který řeší kompenzaci teploty na základě této informace. Pokud trvá nárůst teploty o  $0,5^\circ\text{C}$  delší dobu jak 1 000 sekund, tak se žádná úprava neprovede. Úprava se používá pouze po zapnutí zařízení a po jedné hodině od zapnutí je korekce  $\Delta\vartheta_i^S$  vždy nulová.

Standardní praktikou při doplňování operačního systému Linux o nové funkce je, že máme k dispozici pouze datové typy celých čísel. Z toho vyplývá, že nejsou k dispozici datové typy float, double ani standardní matematické operace jako sin, cos, exp atp.

Exponenciální funkce pro korekce  $\Delta\vartheta_i^S$ ,  $\Delta\vartheta_i^{BL}$  jsou nahrazeny aproximačními tabulkami v přílohách A, B. V tabulkách se vybírají hodnoty podle časů  $t_1, t_2$ , kde  $t_1$  je čas pro náběhovou exponenciálu a  $t_2$  je čas od poslední změny stavu displeje. Pro výběr hodnot z tabulek zavedeme značení:  $S_{t_1}$  pro korekci teploty při náběhu a  $Q_{t_2}$  pro korekci oteplení displejem.

Prostorová teplota je z algoritmu získána rovnicí

$$T_e = (T + S_{t_1}) - \Delta\vartheta_{BL} - \Delta\vartheta_{DPS}.$$

Parametr  $\Delta\vartheta_{BL}$  značí úpravu teploty dle doby zapnutého displeje a aktuální teploty zařízení,  $\Delta\vartheta_{DPS}$  je úprava teploty zařízení z aktuálních teplot. Rovnice těchto úprav zní

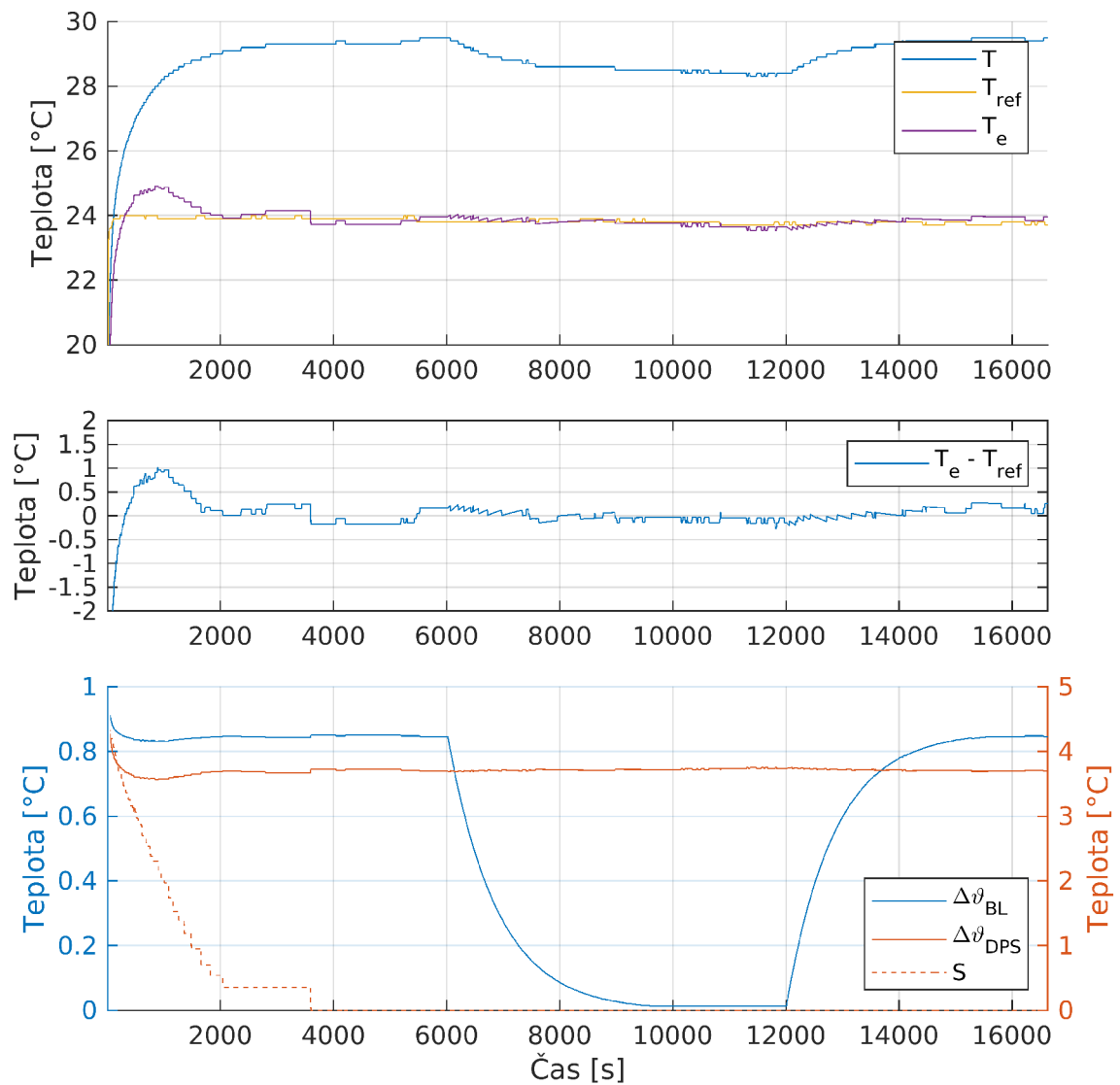
$$\Delta\vartheta_{BL} = (-0,019 \cdot (T + S_{t_1}) + 1,418) \cdot Q_{t_2},$$

$$\Delta\vartheta_{DPS} = -0,159 \cdot (T + S_{t_1} - \Delta\vartheta_{BL}) + 8,248.$$

Pro zkrácení zápisu zavedeme  $T_{t_1} = T + S_{t_1}$ . Výsledná rovnice potom

$$T_e = 1,159T_{t_1} - 1,643Q_{t_2} + 0,022T_{t_1}Q_{t_2} - 8,248.$$

Použití těchto rovnic, parametrů a tabulek bude vysvětleno v následujících sekcích. Výsledný algoritmus společně se simulační aplikací je v příloze D. Grafické znázornění algoritmu s křivkami teplotních posunů viz obrázek 5.1.



Obrázek 5.1: Grafické znázornění funkce algoritmu a jeho teplotních posunů;  $T$  – teplota zařízení,  $T_{ref}$  – prostorová teplota,  $T_e$  – odhadovaná prostorová teplota,  $\Delta\vartheta_{BL}$  – oteplení displejem,  $\Delta\vartheta_{DPS}$  – nejvyšší oteplení perifériemi na desce plošných spojů,  $S$  – kompenzace teploty po zapnutí zařízení

## 5.1 Teplotní posun zařízení

Kompensace maximálního oteplení  $\Delta\vartheta_{DPS}$  s vypnutým displejem činí největší podíl při korekci teploty zařízení. Měřením byla získána data maximálního oteplení při různých teplotách pracovního prostředí. Z dat vybereme pouze ta, která měla při měření vypnutý displej.

Data máme rozdělena po jednotlivých testech. A jsou vybrány pouze ty testy, kde bylo pozorováno ochlazování zařízení po vypnutí jeho displeje.

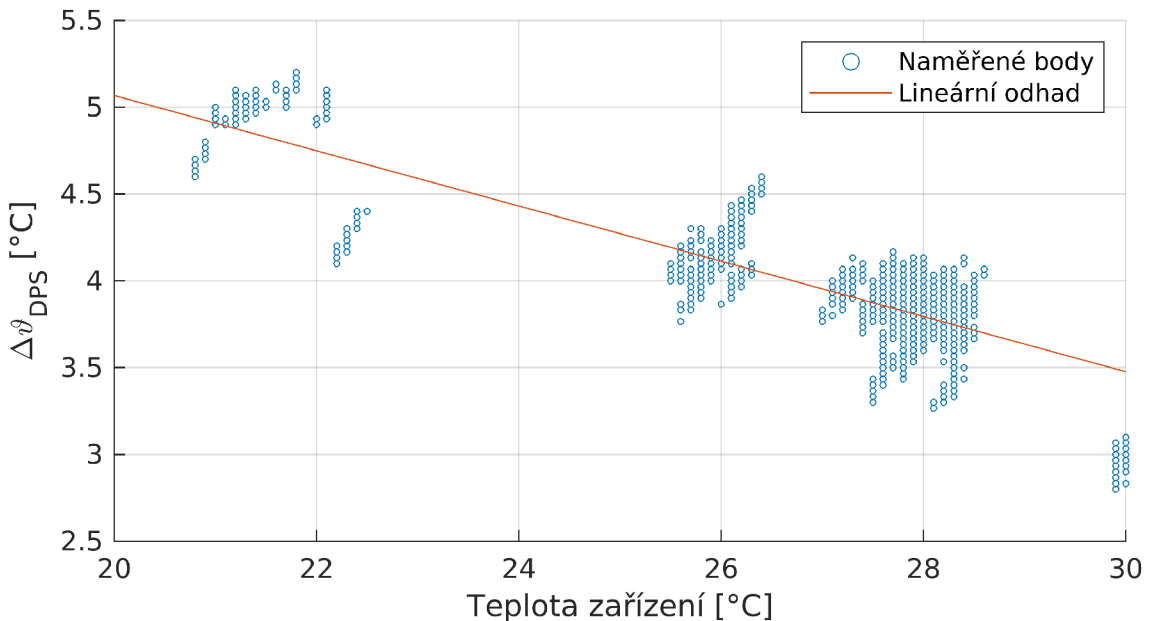
Výběrem bodů  $(T_i, T_i^{\text{ref}})$  při maximálně ochlazeném displeji  $\vartheta_{\infty}^{\text{OFF}}$  získáme ustálené teploty  $T_i$  zařízení při teplotě prostředí  $T_i^{\text{ref}}$ ,  $i$  je index bodu. Celkový počet naměřených bodů ve těchto testech je 3920. Tyto body jsou použity při bodových odhadech parametru  $\beta_1, \beta_2$  lineárního regresního modelu  $\Delta\vartheta_i^{\text{DPS}} = \beta_1 T_i + \beta_2$ , kde  $\Delta\vartheta_i^{\text{DPS}} = T_i - T_i^{\text{ref}}$ .

Aplikací těchto metod odhadneme parametry  $b_1 = -0,159, b_2 = 8,248$  a výsledná rovnice kompenzace má tvar

$$\Delta\vartheta_{DPS} = -0,159 \cdot T + 8,248. \quad (5.3)$$

Proložení křivky naměřenými body je v obrázku 5.2.

Lineární regresní model je zvolen, přestože teplota objektu v pracovním prostředí nemusí být lineárně závislá na teplotě pracovního prostředí. Avšak kvůli krátkému intervalu naměřených teplot zařízení  $[20, 30](^{\circ}\text{C})$ , tak chyba použitím lineárního modelu je minimální a můžeme ji zanedbat.



Obrázek 5.2: Regresní křivka teplotního posunu zařízení v *konečném stavu bez displeje*

## 5.2 Teplotní posun displeje

Kompenzace ohřevu displejem není přímočará jako u korekce maximálního oteplení s vypnutým displejem  $\Delta\vartheta_{\text{DPS}}$ . Kompenzace teplotního posunu vychází z rovnic (3.2) a (3.6). Maximální oteplení  $\Delta\vartheta_{\infty}$  je závislé na teplotě zařízení  $T$ . Relativní oteplení displejem je závislé na čase  $t$ .

Nejprve budou odhadovány parametry  $\beta_1, \beta_2$  lineární závislosti maximálního oteplení displeje  $\Delta\vartheta_{\text{BL}}^{\infty}$  na teplotě zařízení  $T$ . Druhým krokem je odhad parametru tepelné charakteristiky  $\tau$  z rovnic (3.2) a (3.6).

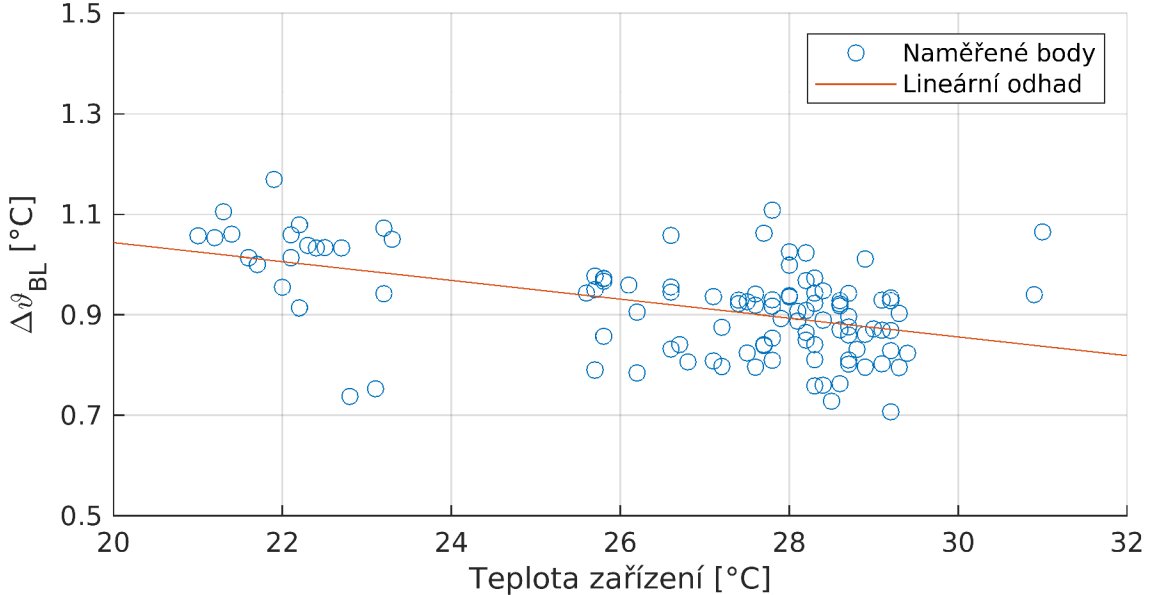
Maximální oteplení  $\Delta\vartheta_{\text{BL}}$  představuje rozdíl mezi ustálenou teplotou se zapnutým displejem a ustálenou teplotou s vypnutým displejem  $\Delta\vartheta_{\text{BL}}^{\infty} = \vartheta_{\infty}^{\text{ON}} - \vartheta_{\infty}^{\text{OFF}}$  za teploty prostředí  $T_{\text{ref}}$ . Výpočet je možné také provést z rozdílu oproti prostorové teplotě, kde  $\Delta\vartheta_{\text{BL}}^{\infty} = \vartheta_{\infty}^{\text{ON}} - \vartheta_{\infty}^{\text{OFF}} = \Delta\vartheta_{\infty}^{\text{ON}} - \Delta\vartheta_{\text{DPS}} = (\vartheta_{\infty}^{\text{ON}} - T_{\text{ref}}) - \Delta\vartheta_{\text{DPS}}$ . Teplota  $T_{\text{DPS}}$  je získána z rovnice pro výpočet teplotního posunu zařízení 5.3. Stejný postup při odhadech parametrů lineárního regresního modelu je proveden pro zařízení v *konečném stavu s displejem*. Z toho získáme rovnici maximálního oteplení s displejem

$$\Delta\vartheta_{\infty} = -0,178 \cdot T + 9,766. \quad (5.4)$$

Rozdílem rovnic 5.4 a 5.3 je rovnice maximálního oteplení displejem

$$\Delta\vartheta_{\text{BL}}^{\infty} = -0,019 \cdot T + 1,418. \quad (5.5)$$

Proložení křivky naměřenými body je v obrázku 5.3.



Obrázek 5.3: Regresní křivka maximálního teplotního posunu displeje

Na teplotní vývoje ohřevu a ochlazení je použita nelineární regrese

$$\Delta\vartheta = f(t, \tau) + \epsilon, \quad (5.6)$$

kde  $t$  je vstupní proměnná času a hledáme teplotní charakteristiku displeje  $\tau$ . Každý naměřený cyklus displeje má odlišné maximální oteplení, kvůli závislosti na teplotě prostředí. Parametry  $\beta_i$  v regresním modelu ((5.7), (5.8)) určují lineární závislost na teplotě zařízení  $T$  pro každou dimenzi  $i$ . Z rovnic odhadujeme parametr  $\tau$  pro všechny dimenze bodů. V závislosti zda je displej vypnutý nebo zapnutý se za funkci (5.6) dosadí rovnice (5.7) respektive (5.8).

$$\Delta\vartheta = \beta_i^1 \cdot \left(1 - \exp\left(-\frac{t}{\tau}\right)\right) + \beta_i^2 \quad (5.7)$$

$$\Delta\vartheta = \beta_i^1 \cdot \left(\exp\left(-\frac{t}{\tau}\right)\right) + \beta_i^2 \quad (5.8)$$

Z regresního modelu byl odhadnut parametr  $\tau = 860$ . Oteplovací charakteristika displeje má potom rovnici

$$\Delta\vartheta_{\text{BL}} = \Delta\vartheta_{\text{BL}}^{\infty} \cdot \left(1 - \exp\left(-\frac{t}{860}\right)\right) \quad (5.9)$$

a dosazením za  $\Delta\vartheta_{\text{BL}}^{\infty}$  dostaneme

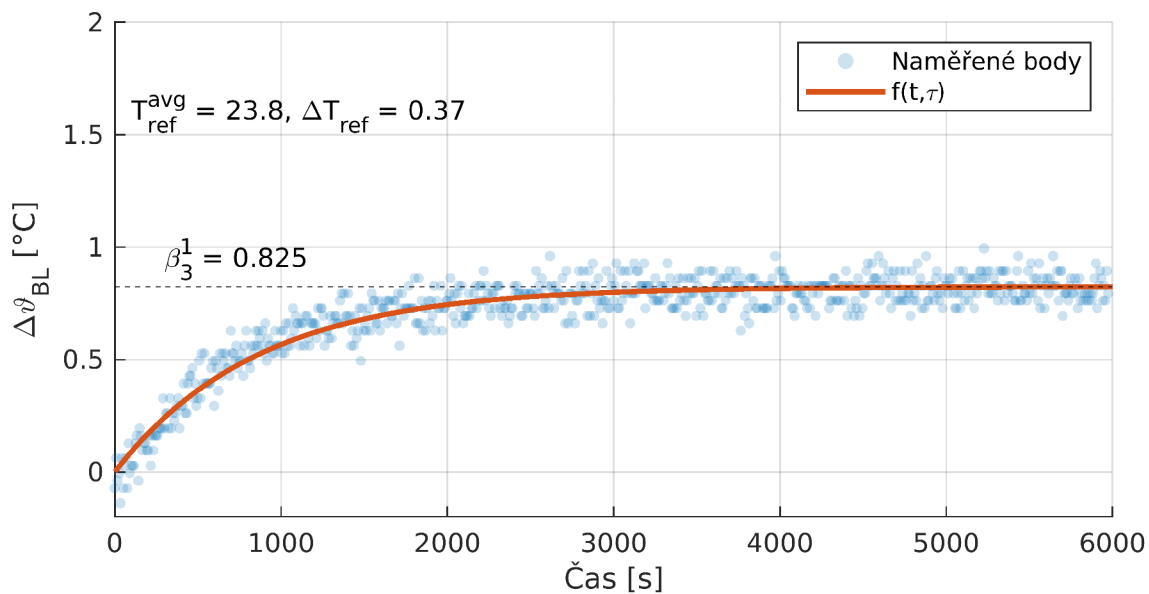
$$\Delta\vartheta_{\text{BL}} = (-0,019 \cdot T + 1,418) \cdot \left(1 - \exp\left(-\frac{t}{860}\right)\right) \quad (5.10)$$

Rovnice relativního oteplení má poté tvar

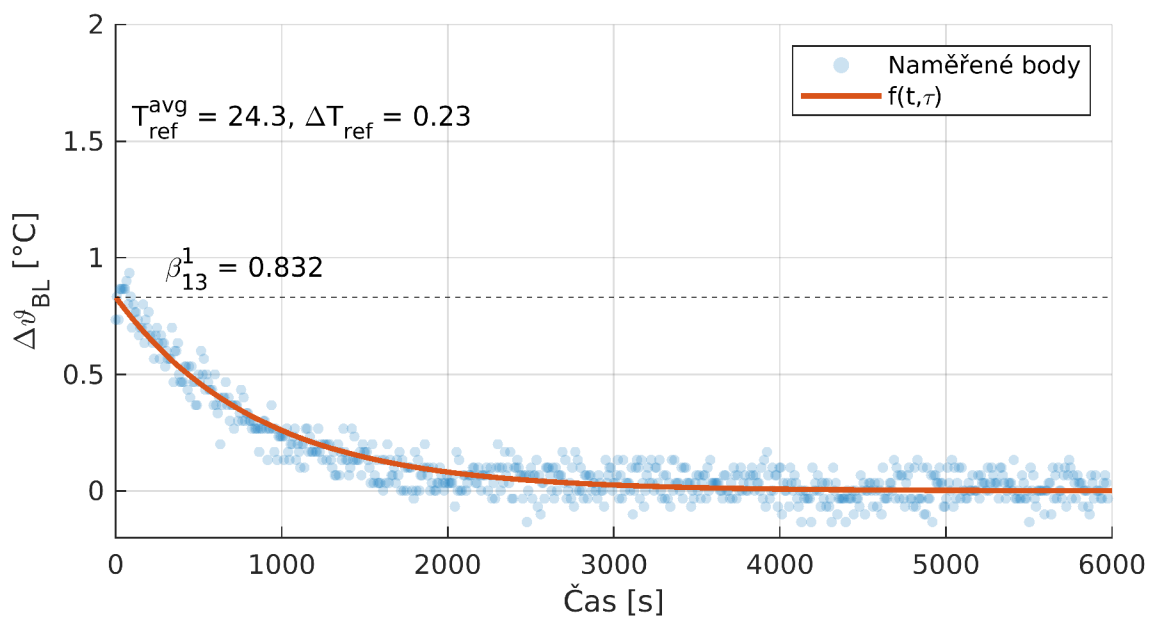
$$\Delta\vartheta_n = \left(1 - \exp\left(-\frac{t}{860}\right)\right). \quad (5.11)$$

Relativní oteplení dosahuje maximálního oteplení  $1^\circ\text{C}$ . Použijeme-li vstupní parametr  $t = 3600$  v rovnici (5.11) dostaneme  $\Delta\vartheta_{\text{BL}}^{t=3600} = 0,985$ . Použitím času  $t = 3600$  je dosaženo 98,5. percentilu náběhu. To znamená, že odhady z kapitoly 4.1 byly správné. Máme-li rozlišení teploty  $0,1^\circ\text{C}$ , tak pro zbytek náběhu můžeme použít  $t \rightarrow \infty$  a použít maximální oteplení  $\Delta\vartheta_{\infty}$  bez toho aniž by úprava byla patrná.

Z rovnicí (5.11) vytvoříme aproximační tabulku s oteplením krokovaným po 10 sekundách  $\{t_{10}, t_{20}, \dots, t_{3600}\}$  viz příloha A. Tabulka o 360 hodnotách popisující teplotní charakteristiku displeje (s maximálním oteplením v okolí  $1^\circ\text{C}$ ) s rozlišením měření  $0,1^\circ\text{C}$  je dostatečná. Některé regresní rovnice jsou vykresleny v obrázku 5.5.



Obrázek 5.4: Oteplovací křivka displeje;  $T_{\text{ref}}^{\text{avg}}$  – průměrná teplota prostředí,  $\Delta T_{\text{ref}}$  – rozptyl naměřených teplot prostředí,  $\beta_3^1$  – maximální oteplení 3. měření



Obrázek 5.5: Ochlazovací křivka displeje;  $T_{\text{ref}}^{\text{avg}}$  – průměrná teplota prostředí,  $\Delta T_{\text{ref}}$  – rozptyl naměřených teplot prostředí,  $\beta_{13}^1$  – maximální oteplení 13. měření

## 5.3 Náběh zařízení

Zařízení v běžném provozu se nevypíná. Občas však nastane situace, že je nutné ho restartovat. Po restartu zařízení nelze použít oteplovací rovnici (3.2), protože neznáme parametr času. Když je zařízení zapnuté v náhodném intervalu oteplovací charakteristiky, tak po zapnutí není známo v jakém bodu průběhu oteplování se zařízení nachází.

Řešením problému je využít informace, jak dlouho trvalo než se zařízení oteplilo o  $0,5^\circ\text{C}$ . Kompenzace se aplikuje pouze po startu zařízení a jakmile oteplení o  $0,5^\circ\text{C}$  trvá déle jak 1 000 sekund, tak provedení kompenzace již není žádoucí, jelikož se může jednat o oteplení prostorovou teplotou. Následkem oteplení prostorovou teplotou je pomalý růst teploty. Nejedná se o rychlý nárůst jako při náběhu zařízení.

Ze záznamů se ručně vybraly naměřené body při náběhu zařízení. Kompenzace náběhu neřeší při jaké prostorové teplotě kompenzujeme, protože informace není známa. Naměřené body  $p_i = [t_i, T_i]$  máme rozdělené do skupin bodů  $G_k$ , kde každá skupina reprezentuje jeden náběh  $G_k = \{p_i, p_{i+1}, \dots, p_{i+r-1}\}$ , kde  $i$  značí počáteční bod náběhu  $k$  a  $r$  počet bodů naměřeného náběhu  $k$ . Pro každý bod  $p_i$  ze skupiny  $G_k$  hledáme nejbližší následující bod  $p_{i+j}$  takový, který má  $T_{i+j} \geq T_i + 0,5$  a vytvoříme nový bod  $s_m = [t_m, \Delta T_m]$ , který přiřadíme do skupiny  $R$ . Parametr  $t_m$  bodu  $s_m$  je  $t_m = t_{i+j} - t_i$  a značí čas, jak dlouho trvalo oteplení o  $0,5^\circ\text{C}$ , a parametr  $\Delta T_m$  je kompenzace teploty získaná z  $\Delta T_m = T_{i+r-1} - T_{i+j}$ .

Následně odhadujeme parametry  $\beta_1, \beta_2$  nelineárního regresního modelu

$$\Delta T_m = \beta_1 \cdot \exp\left(-\frac{t_m}{\beta_2}\right) \quad (5.12)$$

bodovými odhady ze skupiny  $R$ .

Odhadnuté parametry jsou v následující rovnici

$$\Delta T = 4,343 \cdot \exp\left(-\frac{t}{317}\right). \quad (5.13)$$

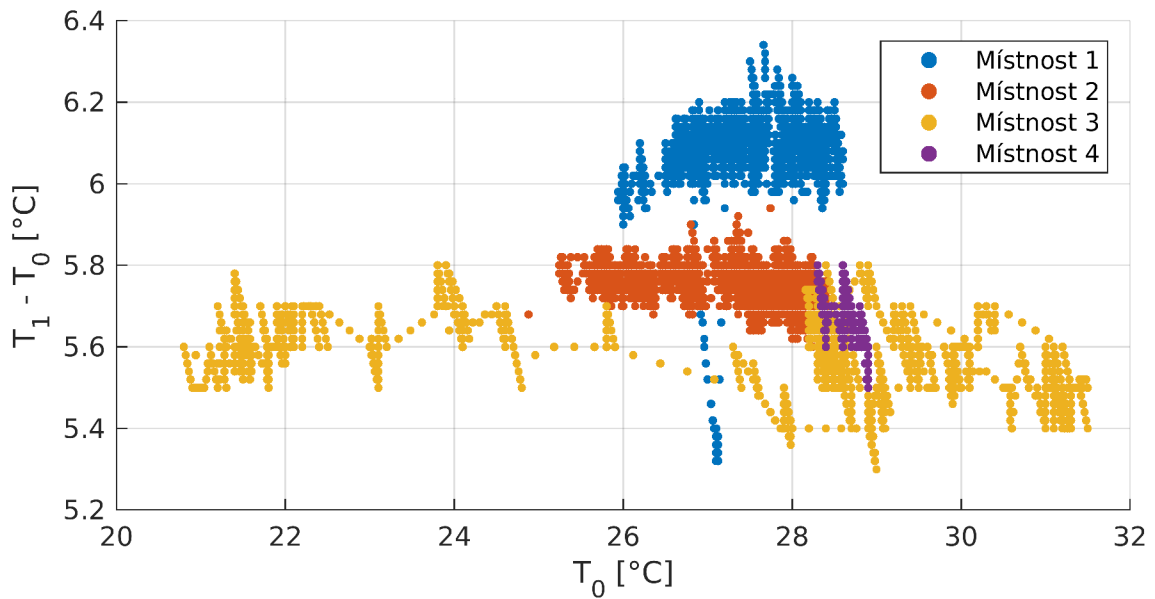
Z rovnice (5.13) vytvoříme aproximační tabulku s kroky po 10 sekundách pro prvních 1 000 sekund viz příloha B. Tabulka kompenzuje náběh zařízení tak, že k aktuální měřené teplotě přičte hodnotu  $\Delta T_t$  a tak se odhadne teplota zařízení v *konečném stavu bez displeje*  $\vartheta_\infty^{\text{OFF}}$ .

## 5.4 Kompenzace prostředí

Použitím kompenzačních rovnic pro zařízení v různých místnostech jsme naměřily body v grafech v příloze C. V obrázku C.2 je chyba výpočtu okolo  $1^\circ\text{C}$ . Jediné informace, které z měření jsou k dispozici, jsou teploty čidel. Proto bylo vyzkoušeno, zda není možné provést kompenzaci závislosti rozdílů teplot dvou čidel na pracovním prostředí. Tento rozdíl je znázorněn v grafu 5.6.

V počátcích se řešení zdálo jako platné, protože je pozorovatelný teplotní rozdíl  $0,4^\circ\text{C}$  mezi místnostmi 1 a 2. Vypracovala se kompenzace prostředí a ta se aplikovala.





Obrázek 5.6: Závislost rozdílu teplot čidel na prostředí;  $T_0$  – teplota prvního čidla,  $T_1$  – teplota druhého čidla

Při novém měření (data v grafu jako Místnost 4) v místnosti 1, byl rozdíl teplot odlišný a tato kompenzace se prokázala jako nepoužitelná.

Řešením chyby kompenzace bylo doplnění možnosti upravit si výslednou teplotu v řádu desetin stupňů Celsia. Kontrétně při použití v místnosti 1 byl nastaven tento parametr na  $1\text{ }^\circ\text{C}$  a tím je redukována vzniklá chyba výpočtu. Toto řešení je vhodné pouze v případech, kdy je pozorovaná chyba výpočtu stálá.

## 5.5 Aplikace řešení

Výsledná rovnice byla použita pro kompenzaci naměřených teplot na provedených testech ve třech místnostech. Celkem bylo provedeno 559 měření s celkovým počtem 413 163 bodů. Po rozdělení testů na jednotlivé místnosti jsme měli provedeno 188 měření v první místnosti, 184 v druhé a 187 ve třetí. Měření probíhalo dva týdny.

Průměrné směrodatné odchylky pozorovaných referenčních teplot  $T_{\text{ref}}$  ze směrodatných odchylek měření pro každou místnost jsou:  $s_1^{\text{ref}} = 0,0855$ ,  $s_2^{\text{ref}} = 0,1113$ ,  $s_3^{\text{ref}} = 0,4365$ .

Ze všech měření byla vybrána pouze ta, která měla rozptyl referenční teploty do  $0,5\text{ }^\circ\text{C}$ . Výsledné počty měření pro jednotlivé místnosti jsou: pro první 118, pro druhou 91 a pro třetí 24. V místnosti 3 byl výběr silně zredukován, kvůli oscilaci teploty v místnosti (viz příloha C). Tato měření byla použita k odhadům parametrů regresních modelů.

Rozdálem vypočtené prostorové teploty a reálnou prostorové teploty získáme jejich odchylku. Pro tento rozdíl pro každou místnost byly vypočteny střední hodnoty  $m$ , směrodatné odchylky  $s$  a rozptyly  $S^2$  zapsané v tabulce 5.1. Po zapnutí zařízení je tato odchylka nejvyšší, jelikož algoritmus sbírá několik sekund data pro

100% kvantil			
Místnost	$m$	$s$	$S^2$
1	0,8560	0,4909	0,2410
2	-0,0436	0,3000	0,0900
3	-0,0493	0,8470	0,7175

99% kvantil			
Místnost	$m$	$s$	$S^2$
1	0,8530	0,3382	0,1132
2	-0,0367	0,2288	0,0518
3	-0,0349	0,7707	0,5881

95% kvantil			
Místnost	$m$	$s$	$S^2$
1	0,8372	0,1996	0,0378
2	-0,0336	0,1987	0,0375
3	-0,0139	0,6843	0,4449

Tabulka 5.1: Porovnání středních hodnot  $m$ , směrodatné odchylky  $s$  a roptylu  $S^2$  z rozdílu odhadnuté prostorové teploty s naměřenou prostorovou teplotou

správný odhad prostorové teploty. Tento jev však nenastává v reálném použití příliš často, jelikož zařízení je nepřetržitě v provozu. Z tohoto důvodu byly vypočteny tyto proměnné i pro 99 a 95 % kvantily, které odstraní tyto odlehlá pozorování. Velké odchylky jsou v grafech v příloze C pozorovatelné jako dlouhé svislé přímky.

V místnosti 3 oscilovala relativně vysokou frekvencí referenční prostorová teplota zapříčiněná nedokonalým zdrojem vytápění místnosti. Předání tepla z místnosti do zařízení není ihned a normálně je pozorovatelné určité zpoždění v měřených teplotách. Tím, že je zde oscilace prostorové teploty a teplota zařízení se téměř neustálí, tak proto máme výsledek s velkým rozptylem.

Místnost 2 měla nejlepší výsledky, kde se její směrodatná odchylka také blíží nule. Odstraněním odlehlých pozorování dostaneme, že v 95 % případech je odchylka měření do 0,5 °C. To je výborný výsledek odhadu, který značí, že algoritmus funguje správně. V místnosti 1 byla střední hodnota odchylky měření poměrně vysoká. Každá místnost má jiné tepelné vlastnosti a proto není možné mít dobré výsledky ve všech pracovních prostředích. Řešením tohoto problému je možnost ručního zadání kompenzace prostředí v desetinách stupňů Celsia. V případě místnosti 1 by se jednalo o posun -0,8 °C.

Shrnutím předchozích odstavců vyplývá, že se naměřilo dostatečné množství dat pro natrénování parametrů regresních modelů. Z těchto parametrů se vytvořil funkční algoritmus, který odhaduje prostorovou teplotu s poměrně malou odchylkou.

## 6 Závěr

Cílem diplomové práce bylo odhadnout teplotu v místnosti z teplot měřených teplotními čidly osazenými na desce plošných spojů víceúčelového elektronického zařízení. Tato čidla jsou ovlivněna samoohřevem zařízení.

V práci byl rozdělen problém samoohřevu do dvou částí, a to na teplo dodávané z displeje a z desky plošných spojů. Za tímto účelem byla navržena měření, ve kterých se zaznamenávaly teplotní změny na zařízení po zapnutí zařízení a po změně stavu (zapnutí/vypnutí) podsvitu displeje. Naměřené hodnoty byly zpracovány a použity k odhadům parametrů regresních modelů.

Parametry závislosti maximálního ohřevu zařízení s vypnutým a zapnutým displejem byly odhadovány lineárními regresními modely. Jedná se hlavní část kompenzace samoohřevu. Další regresní model odhadoval parametr teplotní charakteristiky ( $\tau$ ) displeje z rovnice (3.2). Parametrická funkce, kam je tento parametr dosazen, počítá změnu teploty zařízení v závislosti na čase, jak dlouho je displej zapnutý respektive vypnutý. Jako poslední byla řešena kompenzace teploty zařízení po jeho zapnutí. Zde se taktéž odhadovala teplotní charakteristika ( $\tau$ ) avšak pro oteplení zařízení v závislosti na čase.

Princip výpočtu prostorové teploty spočívám v tom, že se po zapnutí zařízení nejprve spočte jeho maximální oteplení. Z této hodnoty je odečteno oteplení displejem. Nakonec z této teploty je lineární parametrickou funkcí získána odhadovaná prostorová teplota.

Teplotní charakteristiky ohřevu displeje a zařízení byly zjednodušeny na aproximační tabulky, kde podle nejbližšího času se vyhledá hodnota oteplení. Kompenzační funkce byla úspěšně implementována do algoritmu, který byl vložen do systému zařízení. Funkčnost algoritmu byla otestována v kapitole 5.5, kde jsme došli k závěru, že v místnosti 1 a 2 (viz příloha C) je v 95 % případů rozptyl odhadované teploty v místnosti v řádu desetin stupně Celsia od reálné teploty v místnosti. Rychlé změny prostorové teploty (viz místnost 3) by vyžadovaly další šetření. Algoritmus úspěšně odhaduje prostorovou teplotu z čidel na desce plošných spojů.

## Použitá literatura

- [1] Anděl, Jiří. *Statistické metody*. Praha: matfyzpress, 2007. ISBN: 80-7378-003-8.
- [2] Havelka, Otto. *Elektrické přístroje*. Praha: SNTL, 1985. ISBN: 04-529-85.
- [3] Hayes, Monson H. *Schaum's Outline of Theory and Problems of Digital Signal Processing*. New York: McGraw-Hill, 1998. ISBN: 978-0-07-027389-4.
- [4] Hušková, Marie. *Pravděpodobnost a matematická statistika*. Praha: Karolinum, 2005. ISBN: 80-246-0009-9.
- [5] *Raspberry Pi Documentation*. c2012-2022. URL: <https://www.raspberrypi.com/documentation/> (cit. 07.05.2022).
- [6] Varnuška, Daniel. „Kompenzace vlivu samoohřevu elektronického zařízení měřícího prostorovou teplotu“. Liberec: Technická univerzita v Liberci, 2021.
- [7] Zvára, Karel. *Regrese*. Praha: matfyzpress, 2008. ISBN: 978-80-7378-406-5.

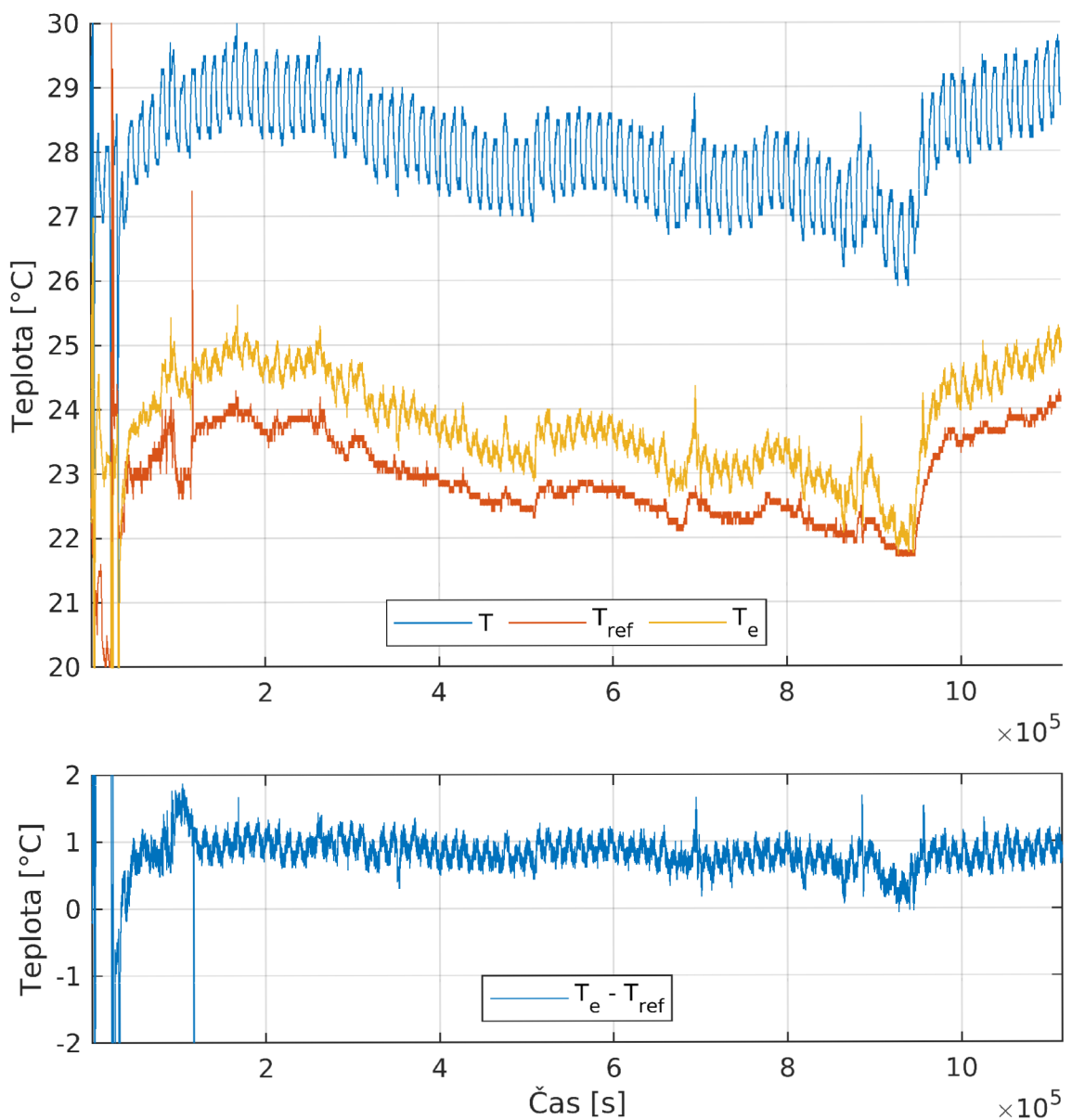
## A Oteplovací tabulka

Čas	10	20	30	40	50	60	70	80	90	100
0	12	23	34	45	56	67	78	89	99	110
100	120	130	140	150	160	170	179	189	198	207
200	217	226	235	244	252	261	269	278	286	294
300	303	311	319	327	334	342	350	357	365	372
400	379	386	393	400	407	414	421	428	434	441
500	447	454	460	466	472	479	485	491	496	502
600	508	514	519	525	530	536	541	546	552	557
700	562	567	572	577	582	587	592	596	601	606
800	610	615	619	623	628	632	636	641	645	649
900	653	657	661	665	669	673	676	680	684	687
1000	691	695	698	702	705	708	712	715	718	722
1100	725	728	731	734	737	740	743	746	749	752
1200	755	758	761	764	766	769	772	774	777	779
1300	782	785	787	789	792	794	797	799	801	804
1400	806	808	810	813	815	817	819	821	823	825
1500	827	829	831	833	835	837	839	841	843	844
1600	846	848	850	851	853	855	857	858	860	861
1700	863	865	866	868	869	871	872	874	875	877
1800	878	880	881	882	884	885	886	888	889	890
1900	891	893	894	895	896	898	899	900	901	902
2000	903	905	906	907	908	909	910	911	912	913
2100	914	915	916	917	918	919	920	921	922	923
2200	923	924	925	926	927	928	929	929	930	931
2300	932	933	933	934	935	936	936	937	938	939
2400	939	940	941	941	942	943	943	944	945	945
2500	946	947	947	948	948	949	950	950	951	951
2600	952	952	953	954	954	955	955	956	956	957
2700	957	958	958	959	959	960	960	961	961	961
2800	962	962	963	963	964	964	964	965	965	966
2900	966	966	967	967	968	968	968	969	969	969
3000	970	970	970	971	971	972	972	972	972	973
3100	973	973	974	974	974	975	975	975	976	976
3200	976	976	977	977	977	977	978	978	978	978
3300	979	979	979	979	980	980	980	980	981	981
3400	981	981	981	982	982	982	982	983	983	983
3500	983	983	984	984	984	984	984	984	985	985

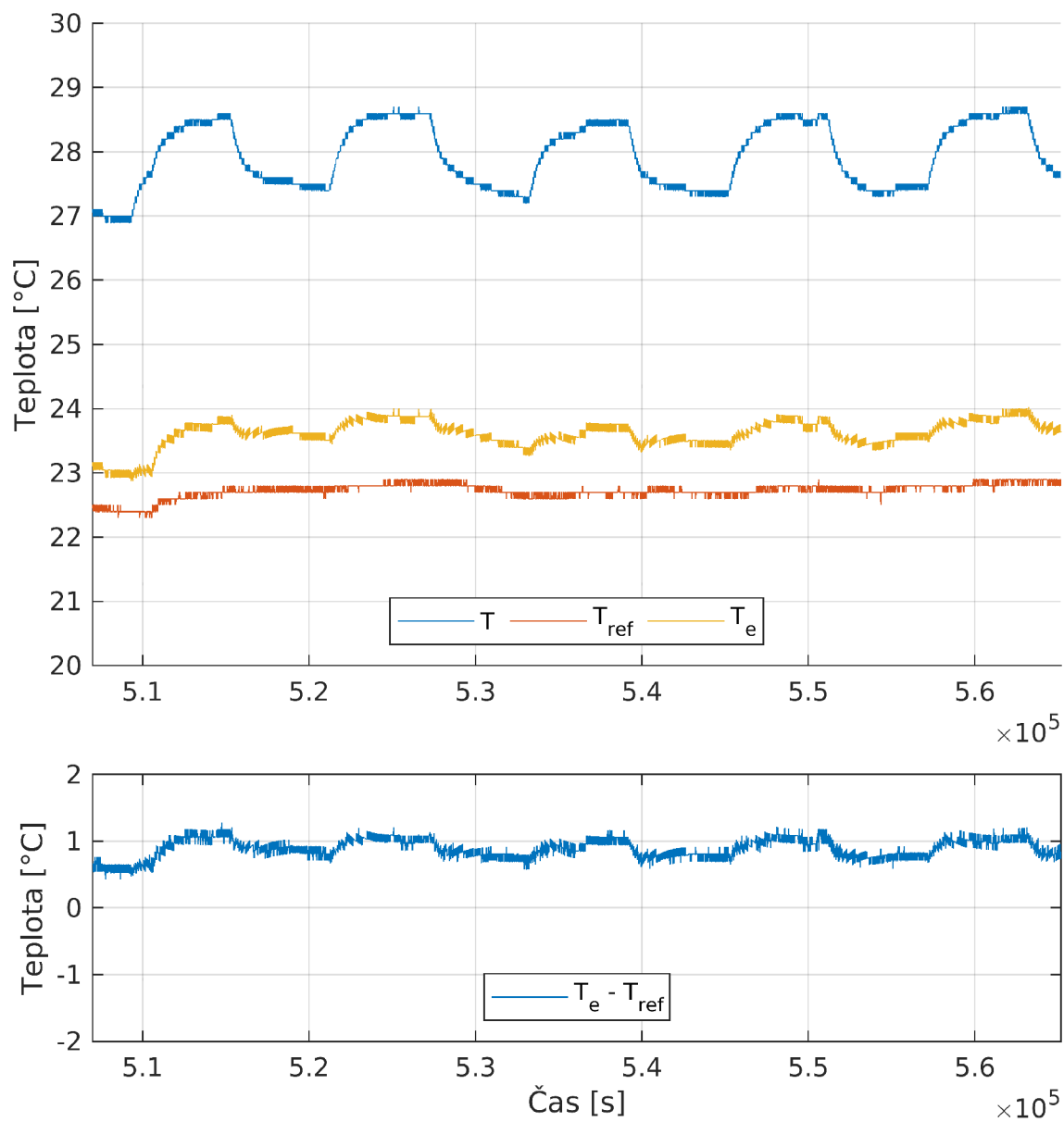
## B Tabulka náběhu zařízení

Čas	10	20	30	40	50	60	70	80	90	100
0	4330	4195	4065	3938	3816	3698	3583	3471	3363	3259
100	3158	3060	2964	2872	2783	2697	2613	2532	2453	2377
200	2303	2231	2162	2095	2030	1967	1906	1846	1789	1733
300	1680	1627	1577	1528	1480	1434	1390	1347	1305	1264
400	1225	1187	1150	1114	1080	1046	1014	982	952	922
500	893	866	839	813	787	763	739	716	694	672
600	652	631	612	593	574	556	539	522	506	490
700	475	460	446	432	419	406	393	381	369	358
800	347	336	325	315	305	296	287	278	269	261
900	253	245	237	230	223	216	209	203	196	190

## C Aplikace kompenzačního algoritmu

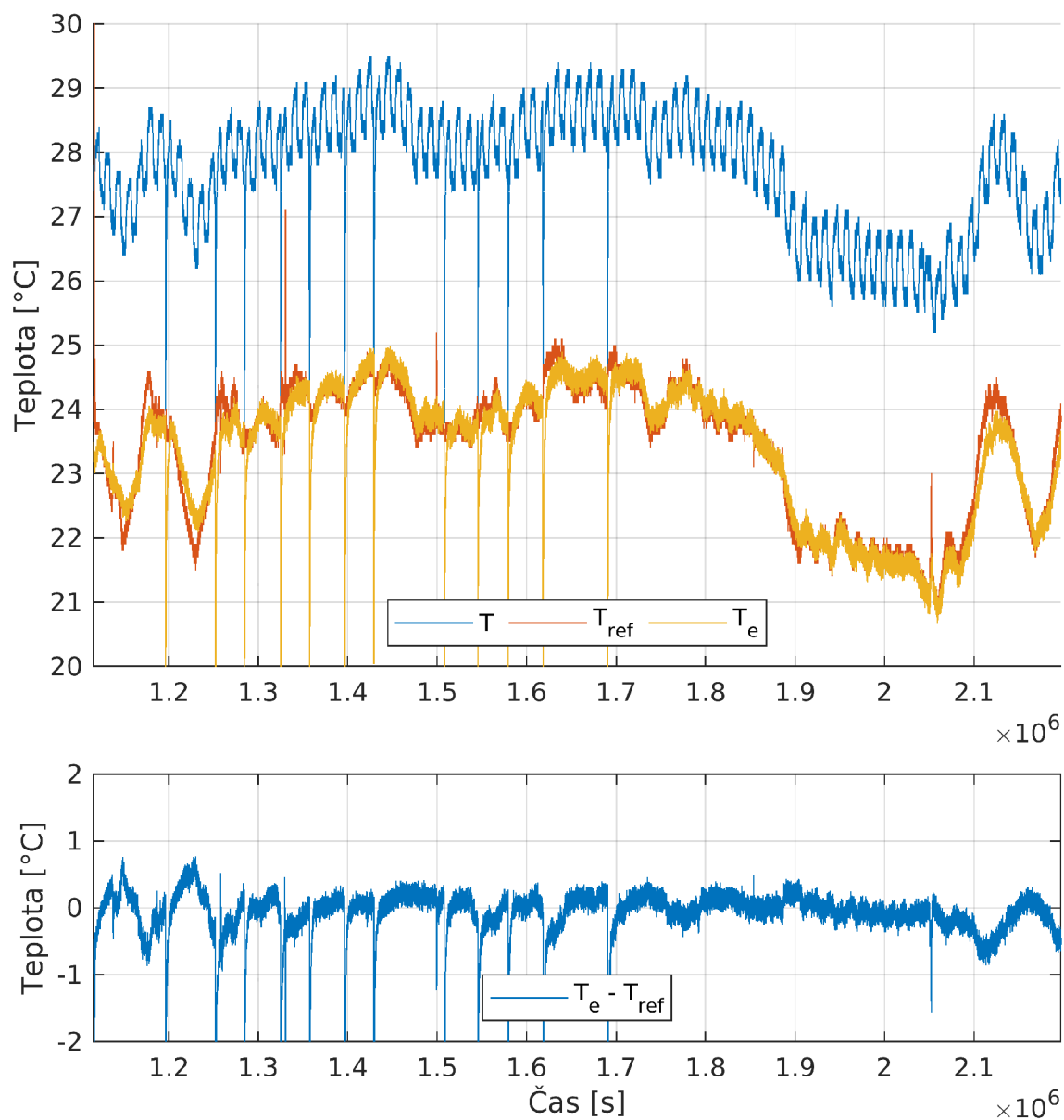


Obrázek C.1: Použití algoritmu v místnosti 1;  $T$  – teplota zařízení,  $T_{ref}$  – teplota v místnosti,  $T_e$  – odhadovaná teplota v místnosti

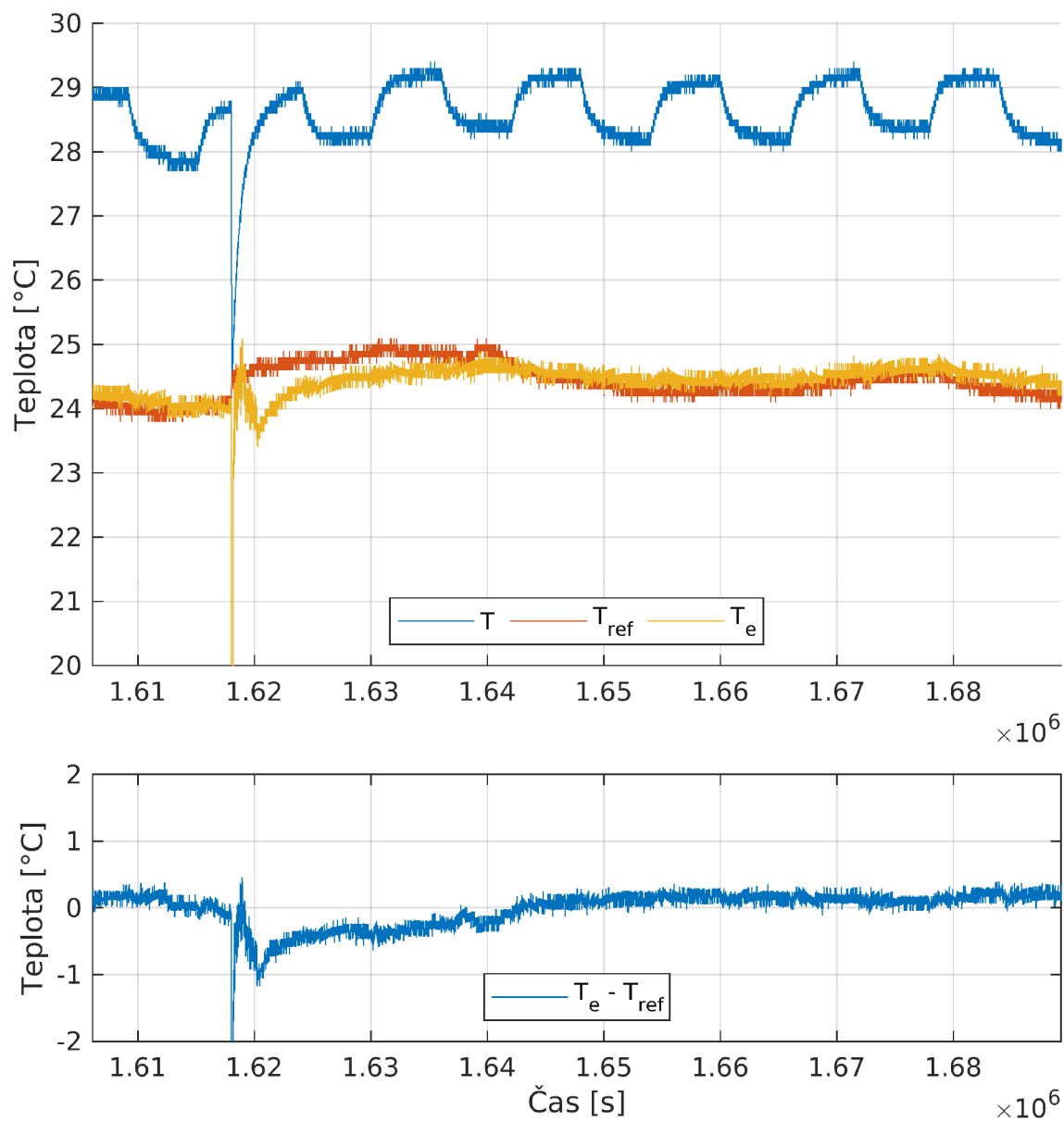


Obrázek C.2: Přibližný graf místnosti 1;  $T$  – teplota zařízení,  $T_{ref}$  – teplota v místnosti,  $T_e$  – odhadovaná teplota v místnosti

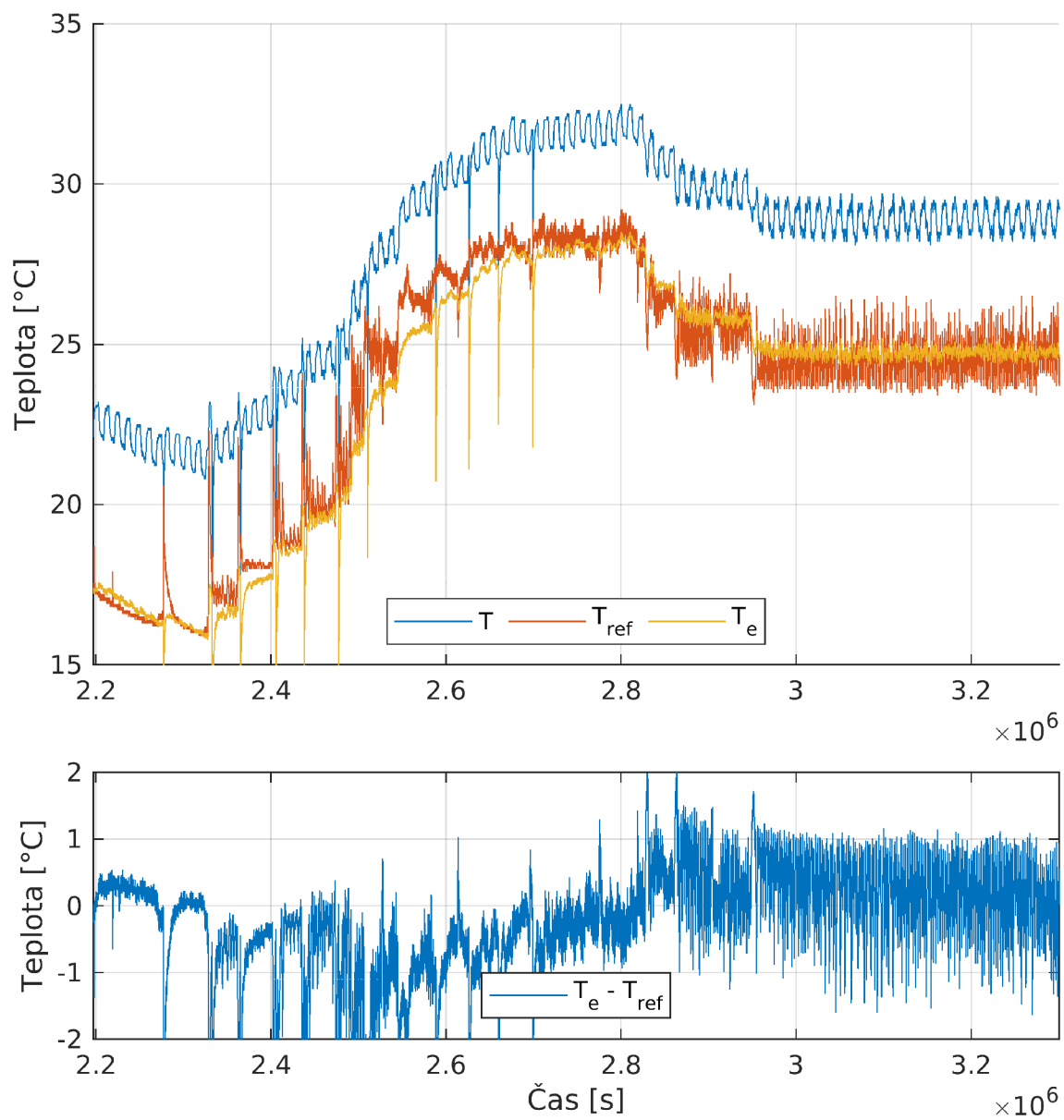




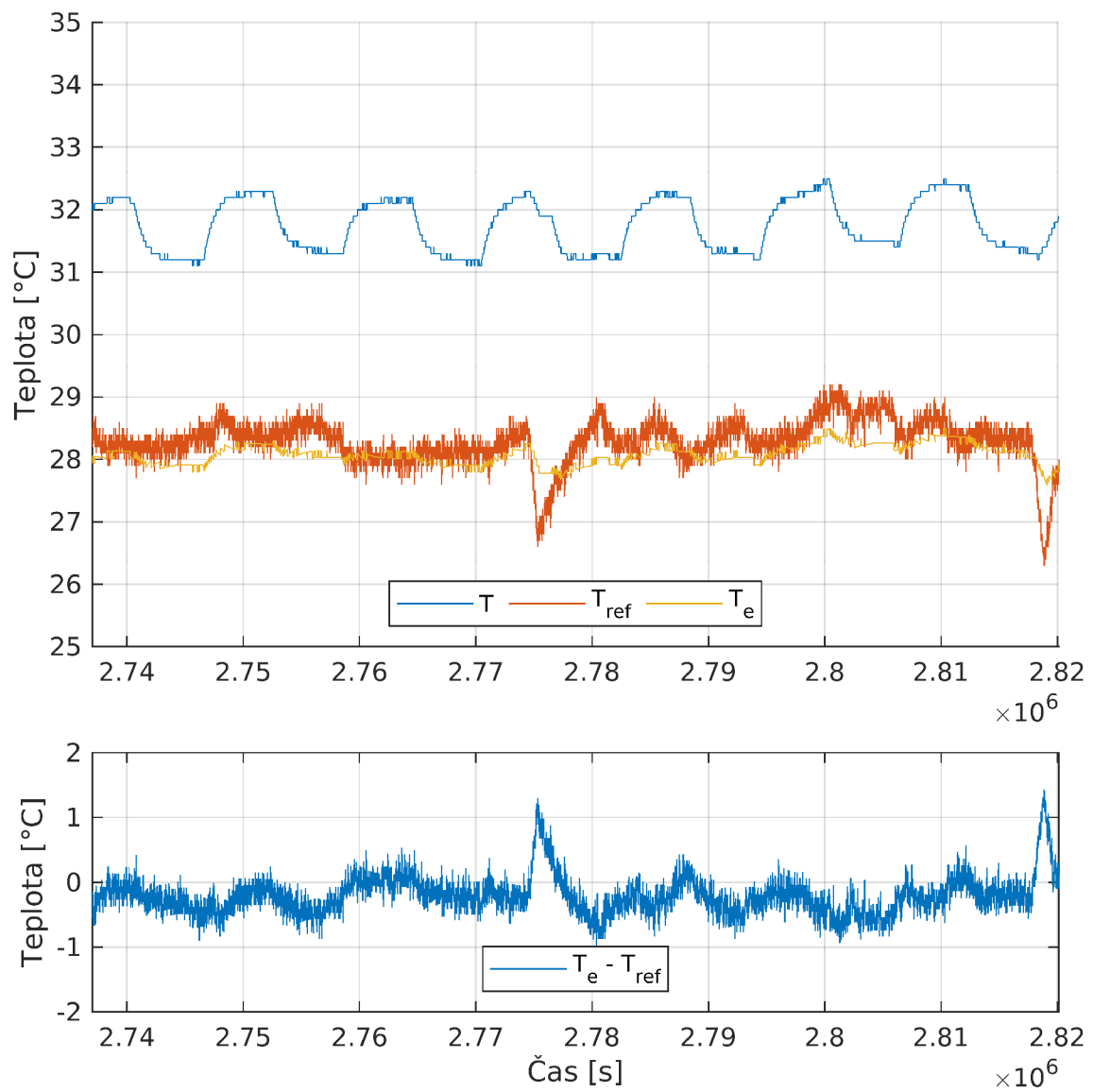
Obrázek C.3: Použití algoritmu v místnosti 2;  $T$  – teplota zařízení,  $T_{ref}$  – teplota v místnosti,  $T_e$  – odhadovaná teplota v místnosti



Obrázek C.4: Přibližný graf místnosti 2;  $T$  – teplota zařízení,  $T_{ref}$  – teplota v místnosti,  $T_e$  – odhadovaná teplota v místnosti



Obrázek C.5: Použití algoritmu v místnosti 3;  $T$  – teplota zařízení,  $T_{ref}$  – teplota v místnosti,  $T_e$  – odhadovaná teplota v místnosti



Obrázek C.6: Přibližný graf místnosti 3;  $T$  – teplota zařízení,  $T_{ref}$  – teplota v místnosti,  $T_e$  – odhadovaná teplota v místnosti

## D Kompenzační algoritmus

```
1 #include <string>
2 #include <iostream>
3 #include <fstream>
4 #include <sstream>
5 #include <vector>
6 #include <chrono>
7 #include <ctime>
8 #include <iomanip>
9 #include <string.h>
10
11 const int k = -159;
12 const int q = 8248;
13 const int bk = -19;
14 const int bq = 1419;
15
16 #define s_table_size 100
17 const uint16_t s_table[100] = {
18     0x10ea,0x1063,0xfe1,0xf62,0xee8,0xe72,0xdf,0xd8f,0xd23,0xcbb,0xc56
19     ,0xbf4,0xb94,0xb38,0xadf,0xa89,
20     0xa35,0x9e4,0x995,0x949,0x8ff,0x8b7,0x872,0x82f,0x7ee,0x7af,0x772,0
21     x736,0x6fd,0x6c5,0x690,0x65b,
22     0x629,0x5f8,0x5c8,0x59a,0x56e,0x543,0x519,0x4f0,0x4c9,0x4a3,0x47e,0
23     x45a,0x438,0x416,0x3f6,0x3d6,
24     0x3b8,0x39a,0x37d,0x362,0x347,0x32d,0x313,0x2fb,0x2e3,0x2cc,0x2b6,0
25     x2a0,0x28c,0x277,0x264,0x251,
26     0x23e,0x22c,0x21b,0x20a,0x1fa,0x1ea,0x1db,0x1cc,0x1be,0x1b0,0x1a3,0
27     x196,0x189,0x17d,0x171,0x166,
28     0x15b,0x150,0x145,0x13b,0x131,0x128,0x11f,0x116,0x10d,0x105,0xfd,0
29     xf5,0xed,0xe6,0xdf,0xd8,
30     0xd1,0xcb,0xc4,0xbe,
31 };
32
33 #define b_table_size 360
34 const uint16_t br_table[b_table_size] = {
35     0xc,0x17,0x22,0x2d,0x38,0x43,0x4e,0x59,0x63,0x6e,0x78,0x82,0x8c,0
36     x96,0xa0,0xaa,
37     0xb3,0xbd,0xc6,0xcf,0xd9,0xe2,0xeb,0xf4,0xfc,0x105,0x10d,0x116,0
38     x11e,0x126,0x12f,0x137,
39     0x13f,0x147,0x14e,0x156,0x15e,0x165,0x16d,0x174,0x17b,0x182,0x189,0
40     x190,0x197,0x19e,0x1a5,0x1ac,
41     0x1b2,0x1b9,0x1bf,0x1c6,0x1cc,0x1d2,0x1d8,0x1df,0x1e5,0x1eb,0x1f0,0
42     x1f6,0x1fc,0x202,0x207,0x20d,
43     0x212,0x218,0x21d,0x222,0x228,0x22d,0x232,0x237,0x23c,0x241,0x246,0
```

```

34     x24b,0x250,0x254,0x259,0x25e,
    0x262,0x267,0x26b,0x26f,0x274,0x278,0x27c,0x281,0x285,0x289,0x28d,0
35     x291,0x295,0x299,0x29d,0x2a1,
    0x2a4,0x2a8,0x2ac,0x2af,0x2b3,0x2b7,0x2ba,0x2be,0x2c1,0x2c4,0x2c8,0
36     x2cb,0x2ce,0x2d2,0x2d5,0x2d8,
    0x2db,0x2de,0x2e1,0x2e4,0x2e7,0x2ea,0x2ed,0x2f0,0x2f3,0x2f6,0x2f9,0
37     x2fc,0x2fe,0x301,0x304,0x306,
    0x309,0x30b,0x30e,0x311,0x313,0x315,0x318,0x31a,0x31d,0x31f,0x321,0
38     x324,0x326,0x328,0x32a,0x32d,
    0x32f,0x331,0x333,0x335,0x337,0x339,0x33b,0x33d,0x33f,0x341,0x343,0
39     x345,0x347,0x349,0x34b,0x34c,
    0x34e,0x350,0x352,0x353,0x355,0x357,0x359,0x35a,0x35c,0x35d,0x35f,0
40     x361,0x362,0x364,0x365,0x367,
    0x368,0x36a,0x36b,0x36d,0x36e,0x370,0x371,0x372,0x374,0x375,0x376,0
41     x378,0x379,0x37a,0x37b,0x37d,
    0x37e,0x37f,0x380,0x382,0x383,0x384,0x385,0x386,0x387,0x389,0x38a,0
42     x38b,0x38c,0x38d,0x38e,0x38f,
    0x390,0x391,0x392,0x393,0x394,0x395,0x396,0x397,0x398,0x399,0x39a,0
43     x39b,0x39b,0x39c,0x39d,0x39e,
    0x39f,0x3a0,0x3a1,0x3a1,0x3a2,0x3a3,0x3a4,0x3a5,0x3a5,0x3a6,0x3a7,0
44     x3a8,0x3a8,0x3a9,0x3aa,0x3ab,
    0x3ab,0x3ac,0x3ad,0x3ad,0x3ae,0x3af,0x3af,0x3b0,0x3b1,0x3b1,0x3b2,0
45     x3b3,0x3b3,0x3b4,0x3b4,0x3b5,
    0x3b6,0x3b6,0x3b7,0x3b7,0x3b8,0x3b8,0x3b9,0x3ba,0x3ba,0x3bb,0x3bb,0
46     x3bc,0x3bc,0x3bd,0x3bd,0x3be,
    0x3be,0x3bf,0x3bf,0x3c0,0x3c0,0x3c1,0x3c1,0x3c1,0x3c2,0x3c2,0x3c3,0
47     x3c3,0x3c4,0x3c4,0x3c4,0x3c5,
    0x3c5,0x3c6,0x3c6,0x3c6,0x3c7,0x3c7,0x3c8,0x3c8,0x3c8,0x3c9,0x3c9,0
48     x3c9,0x3ca,0x3ca,0x3ca,0x3cb,
    0x3cb,0x3cc,0x3cc,0x3cc,0x3cc,0x3cd,0x3cd,0x3cd,0x3ce,0x3ce,0x3ce,0
49     x3cf,0x3cf,0x3cf,0x3d0,0x3d0,
    0x3d0,0x3d0,0x3d1,0x3d1,0x3d1,0x3d1,0x3d2,0x3d2,0x3d2,0x3d2,0x3d3,0
50     x3d3,0x3d3,0x3d3,0x3d4,0x3d4,
    0x3d4,0x3d4,0x3d5,0x3d5,0x3d5,0x3d5,0x3d5,0x3d6,0x3d6,0x3d6,0x3d6,0
51     x3d7,0x3d7,0x3d7,0x3d7,0x3d7,
    0x3d8,0x3d8,0x3d8,0x3d8,0x3d8,0x3d8,0x3d9,0x3d9,
52 };
53
54 static int get_duration_b(unsigned int tick, bool reset)
55 {
56     static unsigned int start_tick = 0;
57     static int overflow = true;
58     int duration = 0;
59
60     if(reset)
61     {
62         start_tick = tick;
63         overflow = false;
64     }
65
66     duration = tick-start_tick;
67
68     if(duration >= b_table_size)
69         overflow = true;

```

```

70
71     if(overflow)
72     {
73         return b_table_size;
74     }
75
76     return duration;
77 }
78
79 static int get_b(unsigned int duration, int brightness)
80 {
81     if(brightness == 0)
82     {
83         return 1000-br_table[duration];
84     }
85     else
86     {
87         return br_table[duration];
88     }
89 }
90
91 static int b_to_duration(int b, int brightness)
92 {
93     unsigned int i;
94
95     if(brightness == 0)
96     {
97         for(i = 0; i < b_table_size-1; i++)
98         {
99             if(1000-br_table[i] <= b)
100                 break;
101         }
102     }
103     else
104     {
105         for(i = 0; i < b_table_size-1; i++)
106         {
107             if(br_table[i] >= b)
108                 break;
109         }
110     }
111
112     return i;
113 }
114
115 static int calc_i_offset(int ts, int tick)
116 {
117     static int offset = 0;
118     static int hist_ts[300];
119     static int hist_tick[300];
120     static int cnt = 0;
121
122     memmove(&hist_ts[1], &hist_ts[0], sizeof(hist_ts) - sizeof(int));
123     memmove(&hist_tick[1], &hist_tick[0], sizeof(hist_tick) - sizeof(

```

```

int));
124
125 hist_ts[0] = ts;
126 hist_tick[0] = tick;
127 cnt++;
128
129 if (cnt >= 300)
130 {
131     return 0;
132 }
133
134 while (cnt > 0 && hist_ts[0] - hist_ts[cnt-1] > 500)
135 {
136     int duration = hist_tick[0] - hist_tick[cnt-1];
137     int i = duration / 10;
138     if (i < s_table_size)
139     {
140         offset = s_table[i];
141     }
142     cnt--;
143 }
144 return offset;
145 }
146
147 static int calc_b_offset(int ts, int brightness, int tick)
148 {
149     static int last_brightness = 0;
150     static int b = 1000; //get_b(b_table_size, 0);
151     static int eq_duration = b_table_size;
152     int duration = 0;
153
154     if(last_brightness != brightness)
155     {
156         last_brightness = brightness;
157         get_duration_b(tick/10, true);
158         eq_duration = b_to_duration(b, brightness);
159     }
160
161     duration = eq_duration + get_duration_b(tick/10, false);
162
163     if(duration >= b_table_size)
164         duration = b_table_size-1;
165
166     b = get_b(duration/10, brightness);
167     return b * ( (bk * ts) / 1000 + bq ) / 1000;
168 }
169
170 static int calc_offset(int ts)
171 {
172     return (k * ts) / 1000 + q;
173 }
174
175 int get_temp(int t0, int t1, int brightness, unsigned int tick, std::
vector<int>& v)

```



```

176 {
177     int i_offset = 0;
178     if (tick < b_table_size)
179     {
180         i_offset = calc_i_offset(t0, tick);
181     }
182     t0 = t0 + i_offset;
183
184     int b_offset = calc_b_offset(t0, brightness, tick);
185     int offset = calc_offset(t0-b_offset);
186
187     v.push_back(b_offset);
188     v.push_back(offset);
189     v.push_back(i_offset);
190     return t0 - b_offset - offset;
191 }
192
193 using namespace std;
194 using namespace std::chrono;
195
196 vector<string> split (string s, string delimiter) {
197     size_t pos_start = 0, pos_end, delim_len = delimiter.length();
198     string token;
199     vector<string> res;
200
201     while ((pos_end = s.find (delimiter, pos_start)) != string::npos) {
202         token = s.substr (pos_start, pos_end - pos_start);
203         pos_start = pos_end + delim_len;
204         res.push_back (token);
205     }
206
207     res.push_back (s.substr (pos_start));
208     return res;
209 }
210
211 system_clock::time_point get_time(string s)
212 {
213     tm timeDate = {};
214     istringstream ss(s);
215     ss >> get_time(&timeDate, "%d-%H:%M:%S");
216     return std::chrono::system_clock::from_time_t(mktime(&timeDate));
217 }
218
219 int main(int argc, char** argv)
220 {
221     if (argc < 3)
222     {
223         cout << "Not enough arguments" << endl;
224         return -1;
225     }
226
227     ifstream log{argv[1]};
228     if (!log.is_open())
229     {

```

```

230     cout << "Cannot open input file" << endl;
231     return -1;
232 }
233
234 ofstream out{argv[2]};
235 if (!out.is_open())
236 {
237     cout << "Cannot open output file" << endl;
238     return -1;
239 }
240
241 string line;
242
243 out << "time;tick;ts0;ts1;ts2;bl;intemp;b;o;i" << endl;
244
245 getline(log,line);
246 double tickd = 0;
247 while (getline(log, line))
248 {
249     vector<string> values = split(line, ";");
250     // [0]   [1] [2] [3] [4]       [5]   [6]  [7]
251     // time, ts0, ts1, ts2, inttemp, b_off, off, bl
252
253     system_clock::time_point time = get_time(values[0]);
254     int t0 = stod(values[1]) * 1000;
255     int t1 = stod(values[2]) * 1000;
256     int ref = stod(values[3]) * 1000;
257     int bl = stoi(values[7]);
258
259     system_clock::time_point start = get_time("1-1-0:0:0");
260
261     std::vector<int> v;
262     tickd += 8.315;
263     int tick = tickd;
264     int temp = get_temp(t0, t1, bl, tick, v);
265
266     out << values[0] << ";"
267         << tick << ";"
268         << t0 << ";"
269         << t1 << ";"
270         << ref << ";"
271         << bl << ";"
272         << temp << ";"
273         << v[0] << ";"
274         << v[1] << ";"
275         << v[2] << endl;
276 }
277
278 out.close();
279 log.close();
280 return 0;
281 }

```

## E Logovací program pro Raspberry Pi

```
1 #include <stdio.h>
2 #include <fstream>
3 #include <vector>
4 #include <iostream>
5
6 using namespace std;
7
8 int main()
9 {
10     const char* path = "/sys/bus/w1/devices/28-011919fe8c21/w1_slave";
11
12     while(1)
13     {
14         fstream file;
15         file.open(path, ios::in);
16         vector<string> lines;
17         if (file.is_open())
18         {
19             string line;
20             while(getline(file, line))
21             {
22                 lines.push_back(line);
23             }
24             file.close();
25
26             line = lines.back();
27             size_t pos = line.find("t=");
28             string temp = line.substr(pos+2);
29             cout << temp << endl;
30         }
31     }
32     return 0;
33 }
```

## F Logovací program pro DPS

```
1 #include <logger_app.h>
2
3 int main()
4 {
5     logger::logger_app app;
6     app.run();
7
8     return 0;
9 }
```

Zdrojový kód F.1: main.cpp

```
1 #ifndef LOGGER_APP_H
2 #define LOGGER_APP_H
3
4 #include <mutex>
5 #include <string>
6 #include <thread>
7 #include <chrono>
8 #include <boost/process.hpp>
9
10 namespace logger {
11
12     using std::mutex;
13     using std::string;
14     using std::thread;
15     using std::chrono::time_point;
16     using std::chrono::steady_clock;
17     namespace bp = boost::process;
18
19     class logger_app
20     {
21     public:
22         logger_app();
23
24         void run();
25
26     private:
27         thread remote;
28         mutex remote_lock;
29         int remote_temp{0};
30         int id;
31         time_point<steady_clock> remote_watchdog;
```

```

32     bp::child c;
33
34     string local_read_temp();
35     void remote_read_temp();
36 };
37
38 }
39
40 #endif // LOGGER_APP_H

```

Zdrojový kód F.2: logger\_app.h

```

1 #include "logger_app.h"
2 #include <vector>
3 #include <iostream>
4 #include <boost/date_time.hpp>
5 #include <ctime>
6 #include <boost/asio.hpp>
7
8 namespace logger {
9
10 using std::vector;
11 namespace asio = boost::asio;
12 using namespace boost::posix_time;
13 using namespace boost::gregorian;
14
15 logger_app::logger_app()
16 {
17     std::ifstream in{"/root/.logger_config"};
18     if (in.good())
19     {
20         string line;
21         in >> line;
22         id = std::stoi(line);
23         in.close();
24     }
25     else
26     {
27         id = 0;
28     }
29
30     id++;
31
32     std::ofstream out{"/root/.logger_config"};
33     if (out.is_open())
34     {
35         out << id;
36         out.close();
37     }
38 }
39
40 void logger::logger_app::run()
41 {
42     string log = "";
43     log += "/root/logs/log";

```

```

44     log += std::to_string(id);
45
46     asio::io_service io;
47     asio::deadline_timer timer(io);
48
49     auto deadline = second_clock::local_time();
50     remote_watchdog = steady_clock::now();
51     //     std::cout << "creating thread" << std::endl;
52     remote = thread(&logger_app::remote_read_temp, this);
53
54     while (1)
55     {
56         auto now = steady_clock::now();
57         auto in_time_t = std::chrono::system_clock::to_time_t(std::
58 chrono::system_clock::now());
59     //     std::cout << std::chrono::duration_cast<std::chrono::
60 milliseconds>(now.time_since_epoch() - remote_tp.time_since_epoch()
61 ).count() << std::endl;
62         if (std::chrono::duration_cast<std::chrono::milliseconds>(now.
63 time_since_epoch() - remote_watchdog.time_since_epoch()).count() >
64 5000)
65         {
66             //timeout
67             remote_lock.lock();
68             remote_watchdog = steady_clock::now();
69             remote_temp = 0;
70             remote_lock.unlock();
71     //     std::cout << "timeout" << std::endl;
72             if (c.running())
73             {
74                 c.terminate();
75             }
76             if (remote.joinable())
77             {
78                 remote.join();
79             }
80             if (remote.joinable() == false)
81             {
82                 remote = thread(&logger_app::remote_read_temp, this);
83             }
84         }
85
86         std::stringstream ss;
87         ss << std::put_time(std::localtime(&in_time_t), "%d-%H%M:%S")
88 << ", ";
89
90         remote_lock.lock();
91         ss << remote_temp << ", ";
92         remote_lock.unlock();
93
94         string local_temp = local_read_temp();
95         ss << local_temp << std::endl;

```

```

92 //         std::cout << ss.str() << std::endl;
93
94         std::ofstream out{log, std::ios::app};
95         if (out.is_open())
96         {
97             out << ss.str();
98             out.close();
99         }
100
101         deadline += seconds(1);
102         timer.expires_at(deadline);
103         timer.wait();
104     }
105 }
106
107 string logger_app::local_read_temp()
108 {
109     string rtn = "";
110
111     std::ifstream in;
112
113     for (int i = 0; i <= 2; i++)
114     {
115         char path[100] = {0};
116         sprintf(path, "/sys/class/temp/ts%d/value", i);
117         in.open(path);
118         if (in.is_open())
119         {
120             string value;
121             in >> value;
122             rtn += value;
123             rtn += ",";
124             in.close();
125         }
126     }
127
128     return rtn;
129 }
130
131 void logger_app::remote_read_temp()
132 {
133     vector<string> args;
134     args.push_back("-o ConnectTimeout=1");
135     args.push_back("pi@11.1.1.11");
136     args.push_back("./rpi_therm");
137     bp::ipstream is;
138     c = bp::child(bp::search_path("ssh"), args, bp::std_out > is);
139
140     string line;
141
142     while (c.running() && std::getline(is, line))
143     {
144 //         std::cout << "remote temp: " << line << std::endl;
145         int temperature = std::stoi(line);

```

```
146     remote_lock.lock();
147     remote_watchdog = steady_clock::now();
148     remote_temp = temperature;
149     remote_lock.unlock();
150 }
151 //     std::cout << "remote down" << std::endl;
152 }
153
154 }
```

Zdrojový kód F.3: logger\_app.cpp