

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

TERMINÁL PRO DOCHÁZKOVÝ SYSTÉM

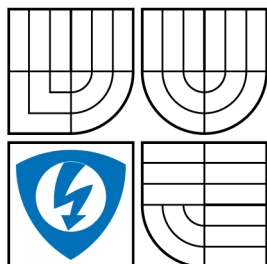
DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. JAKUB CHMELAR



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY  
A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ



FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

## TERMINÁL PRO DOCHÁZKOVÝ SYSTÉM ATTENDANCE SYSTEM TERMINAL

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. JAKUB CHMELAŘ

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. IVO STRAŠIL

BRNO 2014



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Diplomová práce

magisterský navazující studijní obor  
Telekomunikační a informační technika

**Student:** Bc. Jakub Chmelař

**ID:** 125228

**Ročník:** 2

**Akademický rok:** 2013/2014

**NÁZEV TÉMATU:**

## Terminál pro docházkový systém

### POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce je vyvinout, naprogramovat a popsat terminál pro docházkový systém, založený na OS Linux nebo Android, běžícím na vhodném minipočítači (Hackberry, Cubieboard atd.).

### DOPORUČENÁ LITERATURA:

- [1] PEARSON, Robert L. Electronic security systems a manager's guide to evaluating and selecting system solutions. Amsterdam: Butterworth-Heinemann. ISBN 00-804-9470-6.
- [2] STEVEN M. FURNELL ... [ET AL.], Steven M. Furnell ... [et al. editors. Securing information and communications systems principles, technologies, and applications. Boston: Artech House, 2007. ISBN 15-969-3229-5.

**Termín zadání:** 10.2.2014

**Termín odevzdání:** 28.5.2014

**Vedoucí práce:** Ing. Ivo Stražil

**Konzultanti diplomové práce:**

**doc. Ing. Jiří Mišurec, CSc.**  
*Předseda oborové rady*

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Tato práce se zabývá problematikou elektronických docházkových systémů (EDS), které jsou v současné době používány k evidenci docházky zaměstnanců.

V první části práce se autor věnuje evidenci docházky obecně a koncepčnímu návrhu docházkového terminálu, na který navazuje obvodový návrh základní desky pro minipočítač Cubieboard2, RFID čtečku a dotykový LCD displej, který tvoří uživatelské rozhraní. Po vyrobení a vyzkoušení prototypu navržené DPS byl terminál sestaven, díky čemuž autor mohl upřít pozornost na softwarovou část řešení.

Druhá polovina práce je věnována specifikům platformy Allwinner A20 a její podpoře v prostředí operačního systému GNU/Linux. Následně se autor věnuje nástroji Buildroot, který v průběhu práce využil k vytvoření binárního obrazu OS GNU/Linux pro uvedený minipočítač. Po ověření funkce operačního systému a podpory jednotlivých periférií bylo možné přistoupit k realizaci docházkové aplikace, jejíž návrh a řešení jsou rozebrány v poslední části práce.

## KLÍČOVÁ SLOVA

evidence, docházka, elektronický docházkový systém, píchačky, terminál, ARM, Allwinner A20, Cubieboard2, LCD, dotykový displej, RFID, Linux, sunxi, U-Boot, Buildroot, Qt

## ABSTRACT

This thesis deals with electronic attendance systems which are nowadays being widely used to record employee attendance.

General problem of attendance tracking is introduced in first part of the thesis along with a conceptual solution of an attendance system terminal. Author then describes a circuitry and a PCB design of a baseboard for a Cubieboard2 minicomputer, a RFID reader and a touchscreen LCD display. After successful testing of said PCB, the terminal has been completed and it was possible to focus on a software solution.

Latter part of the thesis is dedicated to a specifics of the Allwinner A20 platform and its support by a GNU/Linux operating system. In following chapter author describes use of a Buildroot software for creating a Linux based binary image for an aforementioned minicomputer. Since an operating system and a peripheral support were successfully tested author could focus on a design and implementation of an attendance software that is described in the final chapter.

## KEYWORDS

register, attendace, electronic time and attendance systems, punch clock, terminal, ARM, Allwinner A20, Cubieboard2, LCD, touchscreen, RFID, Linux, sunxi, U-Boot, Qt

CHMELÁŘ, Jakub *Terminál pro docházkový systém*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 62 s. Vedoucí práce byl Ing. Ivo Stražil

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Terminál pro docházkový systém“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Ivu Stražilovi za ochotu, metodickou pomoc a cenné rady při jejím vypracování.

Poděkování patří také mému kolegovi Viktoru Sekaninovi za pomoc při návrhu obvodového řešení a DPS.

Dále bych chtěl poděkovat celé komunitě kolem projektu *linux-sunxi*, bez kterých by práce nemohla v této podobě vzniknout. Velké díky patří také autorům nástroje *Buildroot*, jenž mi byl při řešení praktické části práce nenahraditelným pomocníkem.

Brno .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno .....

.....  
(podpis autora)

# OBSAH

<b>Úvod</b>	<b>11</b>
<b>1 Docházkový systém</b>	<b>12</b>
1.1 Historie . . . . .	12
1.2 Elektronický docházkový systém . . . . .	13
1.2.1 Možnosti . . . . .	13
1.3 Autentizace . . . . .	14
1.3.1 RFID . . . . .	15
1.3.2 Biometrika . . . . .	16
<b>2 Návrh terminálu EDS</b>	<b>17</b>
2.1 Požadavky . . . . .	17
2.2 Koncepce . . . . .	18
2.2.1 Uživatelské rozhraní . . . . .	18
2.2.2 Autentizace . . . . .	18
2.2.3 Komunikace . . . . .	19
2.2.4 Minipočítač . . . . .	19
2.3 Obvodové řešení základní desky . . . . .	21
2.3.1 Napájecí část . . . . .	22
2.3.2 Nízkofrekvenční zesilovač . . . . .	23
2.3.3 Konektory pro připojení periférií . . . . .	24
<b>3 Realizace DPS</b>	<b>25</b>
3.1 Nalezené chyby . . . . .	25
3.1.1 Napájecí konektor . . . . .	25
3.1.2 Připojení LCD panelu . . . . .	26
3.2 Ověření funkce . . . . .	26
<b>4 Linux na platformě Allwinner A20</b>	<b>27</b>
4.1 Projekt linux-sunxi . . . . .	27
4.1.1 Zavaděč u-boot-sunxi . . . . .	28
4.1.2 Jádro linux-sunxi . . . . .	28
4.2 Obraz operačního systému . . . . .	28
4.2.1 Sestavení vlastního obrazu . . . . .	29
<b>5 Buildroot</b>	<b>30</b>
5.1 Konfigurace . . . . .	30
5.1.1 Cílová architektura . . . . .	31



5.1.2	Sada nástrojů pro překlad (toolchain) . . . . .	32
5.1.3	Linuxové jádro . . . . .	33
5.1.4	Zavaděč u-boot-sunxi . . . . .	33
5.1.5	Kořenový souborový systém (rootfs) . . . . .	34
5.2	Překlad a sestavení . . . . .	35
5.2.1	Úpravy rootfs . . . . .	35
5.3	Vytvoření obrazu SD karty . . . . .	36
<b>6</b>	<b>Docházková aplikace</b>	<b>37</b>
6.1	Koncept . . . . .	37
6.2	Řešení . . . . .	37
6.2.1	Komunikace s RFID čtečkou . . . . .	37
6.2.2	Databáze . . . . .	39
6.3	Uživatelské rozhraní . . . . .	40
6.3.1	Hlavní obrazovka . . . . .	40
6.3.2	Uživatelská obrazovka . . . . .	42
6.4	Webové rozhraní . . . . .	45
<b>7</b>	<b>Závěr</b>	<b>46</b>
	<b>Literatura</b>	<b>47</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>51</b>
	<b>Seznam příloh</b>	<b>54</b>
	<b>A Výkresová dokumentace základní desky</b>	<b>55</b>
	<b>B Fotografie prototypu</b>	<b>59</b>
	<b>C Buildroot – konfigurace</b>	<b>60</b>
C.1	Konfigurace prostředí (defconfig) . . . . .	60
C.2	Post-build skript . . . . .	61
<b>D</b>	<b>Obsah přiloženého CD</b>	<b>62</b>

# SEZNAM OBRÁZKŮ

1.1	Mechanické docházkové hodiny firmy ITR. . . . .	12
2.1	Blokové schéma koncepce terminálu EDS. . . . .	18
2.2	Minipočítač Cubieboard2. . . . .	21
2.3	Schéma napájecí části terminálu. . . . .	22
2.4	Schéma nízkofrekvenčního zesilovače. . . . .	24
3.1	Prototyp základní desky. . . . .	25
5.1	Konfigurační menu nástroje Buildroot. . . . .	31
6.1	Relační schéma databáze. . . . .	39
6.2	Hlavní obrazovka docházkové aplikace. . . . .	41
6.3	Detail časové osy. . . . .	41
6.4	Uživatelská obrazovka docházkové aplikace. . . . .	43
6.5	Přehled historie docházky. . . . .	44
6.6	Nastavení předpokládané docházky. . . . .	44
A.1	Základní deska – strana součástek. . . . .	55
A.2	Základní deska – strana spojů. . . . .	55
A.3	Základní deska – osazovací plán. . . . .	56
A.4	Základní deska – schéma zapojení. . . . .	57
B.1	Prototyp docházkového terminálu. . . . .	59
B.2	Detail propojení komponent terminálu. . . . .	59

# SEZNAM TABULEK

2.1	Technické parametry použitého LCD displeje. . . . .	19
2.2	Technické parametry SoC Allwinner A20. . . . .	20
2.3	Technické parametry minipočítače Cubieboard2. . . . .	20
5.1	Konfigurace virtuálního stroje. . . . .	35
6.1	Konfigurace rozhraní UART. . . . .	38
6.2	Stavové ikony a jejich význam. . . . .	42
A.1	Seznam součástí základní desky. . . . .	58

# ÚVOD

Nutnost evidence pracovní docházky zaměstnanců je pravděpodobně stejně stará jako vztah zaměstnanec – zaměstnavatel. Zatímco dříve k tomuto účelu stačil (a na mnoha místech stačí i dnes) pouze papír, tužka a nástěnné hodiny, na jiných místech je použit docházkový systém.

Za první exemplář takového systému lze považovat docházkové hodiny sestavené klenotníkem Williardem Bundym již na konci 19. století. Tyto hodiny sice uměly pouze zaznamenat čas příchodu a odchodu jednotlivých pracovníků, nicméně proti tužce a papíru to byl jistý pokrok. Další desítky let byly docházkové systémy řešeny stále mechanicky, což sice stačilo pro záznam docházky, ale zpracování zaznamenaných dat zatím nebylo možné automatizovat.

Tato nevýhoda se změnila s rozmachem výpočetní techniky, jejíž nástup a rozšíření ovlivnil prakticky všechny oblasti našich životů, evidenci docházky nevyjímaje. S příchodem elektronických docházkových systémů (EDS) bylo možné data automaticky nejen zaznamenat, ale také následně zpracovat. Tvorba měsíčních výkazů, mzdových podkladů a dalších náležitostí tím byla výrazně zjednodušena, což značně usnadnilo práci především administrativě a mzdovým oddělením.

Autor se v práci věnuje návrhu docházkového terminálu, který by umožnil evidenci docházky na pracovišti se zhruba deseti zaměstnanci. V kapitole druhé představuje koncepci docházkového terminálu sestávajícího se z minipočítače Cubieboard2, dotykového LCD displeje a RFID čtečky. K propojení těchto komponent byla navržena základní deska, jejíž obvodové řešení a návrh DPS je uveden v téže kapitole. Následně je věnován prostor praktickému ověření návrhu základní desky na vyrobeném prototypu.

Jelikož se při řešení hardwarové části terminálu nevyskytly větší problémy, mohl autor následně upřít pozornost k problematice softwarové podpory minipočítače Cubieboard2, respektive platformy Allwinner obecně, v prostředí operačního systému (OS) GNU/Linux, na němž je popisovaný terminál postaven. Kapitola čtvrtá je proto věnována uvedení specifik, se kterými se lze při práci s touto platformou setkat. Pro vytvoření binárního obrazu OS pro minipočítač byl využit nástroj Buildroot, který je stručně představen v kapitole páté. Autor zde uvádí, jaké výhody tento nástroj přináší a jak jej při zpracování využil.

Šestá kapitola je věnována popisu docházkové aplikace, která byla v rámci práce vyvinuta specificky pro popsany terminál. Autor uvádí koncepční návrh aplikace a jeho následnou implementaci s využitím programovacího jazyka C++, knihovny Qt a relační databáze SQLite. Popis je doplněn ukázkovými snímky aplikace, na kterých je popsán význam jednotlivých grafických prvků a možnosti interakce ze strany uživatele.

# 1 DOCHÁZKOVÝ SYSTÉM

## 1.1 Historie

Za první docházkový systém lze považovat docházkové hodiny, schopné zaznamenat okamžik příchodu a odchodu jednotlivých zaměstnanců. První exemplář takovýchto hodin byl sestaven již roku 1888 klenotníkem Willardem Bundy. Společnost Bundy Manufacturing Company, založená jeho bratrem o rok později, byla později sloučena do společnosti Computing Tabulating Recording Company, dnes známé jako International Business Machines – IBM. [4, 8, 32]



Obr. 1.1: Mechanické docházkové hodiny původní koncepce, vyrobené firmou International Time Recording Company (část IBM) kolem roku 1910. [Převzato z flickr.com pod licencí Creative Commons, autorem fotografie je Don Harder.]

Tyto docházkové hodiny byly samozřejmě realizovány čistě mechanicky a tak nedokázaly provést více než záznam okamžiku příchodu a odchodu jednotlivých zaměstnanců – tiskem na papírovou pásku. Nicméně na svou dobu to byl převratný vynález, který určil směr dalšího vývoje v oblasti záznamu docházky.

Zatímco vynález docházkových hodin výrazně zjednodušil život zaměstnancům (stačilo si pouze „píchnout“ příchod či odchod), účetním nijak zásadně nepomohl, jelikož byli i nadále nuceni tato data zpracovávat manuálně. To se změnilo až s příchodem elektronických systémů, které umožňují strojové zpracování dat bez nutnosti je nejdříve manuálně přepsat z kartiček.

Přechod od mechanických docházkových systémů k jejich elektronické variantě je docela přirozený a očekávatelný, jelikož pouze sleduje trend který se v technologiích uplatňuje prakticky všude. Elektronickým docházkovým systémům (EDS), jejich principu, možnostem a v neposlední řadě také realizaci se bude věnovat tato práce.

## 1.2 Elektronický docházkový systém

Elektronický docházkový systém (EDS) je, jak již název napovídá, realizován na rozdíl od svého předchůdce elektronicky. Hlavní rozdíl je v principu ukládání dat. Zatímco mechanické docházkové systémy využívaly nejčastěji papírové kartičky na které byl jednoduše tisknut čas, u elektronických systémů je záznam realizován uložením dat na paměťové médium, nejčastěji do polovodičové paměti (krátkodobě) či na pevný disk (dlouhodobě).

Toto řešení má mnoho výhodných vlastností, pro příklad lze uvést:

- strojové zpracování dat – data jsou v digitální podobě přímo uložena, není nutné je manuálně přepisovat např. z docházkových karet;
- jednoduché zálohování – stačí obsah paměti zkopírovat na jiné médium;
- snadná rozšiřitelnost – lze snadno zaznamenat doplňující data, není třeba modifikovat formát fyzického média;
- autentizace zaměstnanců – je možné bezpečně ověřit identitu uživatelů, čímž lze zabránit neoprávněné manipulaci se záznamy;
- kooperace s jinými systémy – EDS lze snadno propojit s dalšími systémy, například elektronickým přístupovým systémem.

### 1.2.1 Možnosti

Základní požadavky na EDS jsou prakticky stejné jako u mechanickým řešení, nicméně elektronické řešení nabízí podstatně více možností a vyšší flexibilitu, které by bylo škoda nevyužít.

První z možností je samozřejmě automatizované zpracování dat. Tato možnost existovala již u mechanických systémů používajících děrné štítky, nicméně tato řešení nebyla zdaleka tak rozšířená jako varianty u kterých bylo nutné data zpracovat manuálně. Automatizované zpracování výrazně ulehčuje práci administrativním pracovníkům, především pak mzdovým účetním. Tvorbu mzdových podkladů lze

přenechat stroji, který určí pracovní docházku jednotlivých zaměstnanců a na jejím základě vypočte mzdy dle nastavených podmínek a kritérií.

Další výhodou je možnost zvýšené ochrany proti neoprávněné manipulaci s daty. V podnicích využívajících mechanické *píchačky*<sup>(1)</sup> bylo poměrně běžnou praxí, že si zaměstnanci „vypomohli“ mezi sebou například tak, že *píchnuli* dřívější příchod nebo naopak pozdější odchod i kolegovi, který se na pracovišti tou dobou nenacházel. Elektronické systémy mohou tento nedostatek řešit použitím pokročilých autentizačních metod, především biometrickou autentizací. Metody autentizace aplikovatelné v EDS budou rozebrány v následující části práce.

V neposlední řadě přechod k elektronickému řešení umožňuje také použít systém na více místech s tím, že zaměstnanec nemusí nutně použít stejný terminál při příchodu i odchodu, což u mechanického řešení zpravidla nebylo možné – i pokud mělo pracoviště více docházkových hodin, bylo nutné použít ty, u kterých měl zaměstnanec uloženu docházkovou kartu. Také je možné realizovat čistě softwarový terminál (například jako webovou aplikaci), který zaměstnanec použije když pracuje mimo své pracoviště, například z domova.

## 1.3 Autentizace

Pojem autentizace označuje proces ověření identity uživatele. K provedení autentizace je zapotřebí tzv. nosič dokazovacího faktoru, který může být reprezentován autentizačním předmětem, nebo jím může být sám uživatel. Dokazovací faktor jsou data, na základě kterých lze určit, zda je uživatel tím, za koho se vydává. Tato data mohou obsahovat tajnou informaci (heslo) nebo unikátní charakteristické rysy uživatele (biometrika). [10, volně převzato]

Nejjednodušší metodou autentizace je autentizace znalostí, při které je dokazovacím faktorem heslo a jeho nosičem sám uživatel, respektive jeho paměť. Hlavní výhodou tohoto druhu autentizace jsou především nízké pořizovací náklady, jelikož není zapotřebí investovat do autentizačních předmětů. Nevýhodou je pak především fakt, že uživatel může heslo zapomenout, což znemožní jeho autentizaci.

Hojně rozšířeným způsobem je autentizace předmětem, při které se uživatel prokazuje nosičem dokazovacího faktoru. Tímto nosičem může být například magnetická karta nebo RFID čip. Právě autentizace pomocí RFID se v poslední době velice rozšířila a proto jí bude v této práci věnováno více prostoru.

Velkou nevýhodou obou zmíněných autentizačních metod je fakt, že nejsou odolné proti úmyslnému zneužití, což pro účely EDS hraje významnou roli. Uživatelé

---

<sup>(1)</sup>Slangový pojem označující mechanické docházkové hodiny, známé především z větších podniků za dob ČSFR. Na některých místech se dochovaly až do současnosti a jsou stále funkční.

(zaměstnanci) si tak mohou vzájemně sdělit své heslo, případně předat autentizační předmět, díky čemuž se mohou vydávat za někoho jiného. V praxi tak může dojít, jak již bylo uvedeno v části 1.2.1, k tomu, že si zaměstnanci vzájemně vypomáhají v tom smyslu, že jsou evidováni jako přítomní i v době, kdy na pracovišti nejsou. V angličtině existuje pro tento způsob podvádění termín *buddy-punching*, který lze do češtiny volně přeložit jako „píchnutí za kamaráda“. Těmto praktikám lze naštěstí zabránit užitím biometrické autentizace (viz část 1.3.2).

### 1.3.1 RFID

Jak již bylo uvedeno, jedním z možných způsobů ověření identity uživatele (autentizace) je použití fyzického nosiče dokazovacího faktoru – autentizačního předmětu. Autentizačních předmětů může být celá řada, nicméně mezi ty rozšířené patří především tzv. RFID tagy, kterým bude věnováno několik následujících odstavců. Nebude-li uvedeno jinak, informace byly čerpány především z [17, 30, 39].

RFID (z anglického *Radio Frequency Identification*) označuje technologii, která umožňuje identifikovat předmět označený tzv. tagem pomocí rádiové komunikace. Využití je v zásadě obdobné jako u čárových kódů, u kterých je ale potřeba čtečku přesně zamířit na čárový kód. RFID tag lze naproti tomu přechíst (detekovat) bez přímé viditelnosti, stačí aby se dostal do dosahu čtecího zařízení. Tagy samotné jsou složeny z integrovaného obvodu řídicího komunikaci a cívký, která funguje jako anténa pro jeho napájení a komunikaci se čtečkou.

Čtečka periodicky do svého okolí vysílá pulzy na nosné frekvenci dané konkrétní specifikací (nejčastěji 125 kHz nebo 13,56 MHz). Dostane-li se do jejího dosahu RFID tag, dojde k indukci této energie v jeho anténě (cívce), která je využita k nabití kondenzátoru a napájení integrovaného obvodu. Tento IO následně vyšle své identifikační číslo, k čemuž většinou využívá zátěžovou modulaci při které dochází ke změně amplitudy nosné vlny úpravou impedance tagu – do obvodu je střídavě připojován a odpojován zatěžovací rezistor, což čtečka detekuje jako změnu zátěže. Tímto způsobem dojde k přenesení informace (většinou numerického identifikátoru), uložené v integrovaném obvodu, do čtečky, která může data dále zpracovat nebo je předat nadřazenému systému.

Na obdobném principu fungují i bezkontaktní karty definované standardem ISO/IEC 14443 [20], které ale na rozdíl od RFID tagů obsahují větší množství paměti, ze které lze data nejen číst, ale také do ní zapisovat. Jako příklad tohoto typu karet lze uvést řadu MIFARE vyráběnou firmou NXP Semiconductors (dříve divize společnosti Philips), které jsou ve světě hojně používány pro nejrůznější aplikace, velmi často pak v sektoru hromadné dopravy.<sup>(2)</sup>

---

<sup>(2)</sup>Příkladem budiž pražská multifunkční karta *opencard*.



V poslední době se bezkontaktní karty dostaly do kurzu také v souvislosti s bezkontaktními platebními kartami, které umožňují provádět platby rychleji a jednodušeji než při použití klasické – kontaktní – platební karty. Tento druh karet většinou nevyžaduje zadání PIN pro platby do hranice cca 500 Kč, díky čemuž se celý proces platby významně urychluje. Na druhou stranu se již objevují obavy z možných zneužití, u kterých teprve čas ukáže jak moc jsou oprávněné.

### 1.3.2 Biometrika

Následující odstavce jsou věnovány ověření identity uživatele pomocí biometrických rysů. Informace byly čerpány především z [10, 17, 30], pokud není uvedeno jinak.

Jak již bylo zmíněno, ani použití autentizace předmětem nezajišťuje ochranu proti úmyslnému zneužití držitelem autentizačního předmětu. Tento problém řeší až autentizace pomocí biometrických údajů, což jsou specifické rysy samotného uživatele. Ty mohou být buď fyziologické (otisk prstu, tvar ruky, zbarvení duhovky, ...) nebo behaviorální (spektrum hlasu, způsob psaní, ...).

V obou případech jde o znaky, které jsou unikátní pro každého jedince, což je pro autentizaci velmi výhodné. Na druhou stranu je nelze získat vždy zcela identicky, takže je nutné při kontrole shody autentizačního faktoru akceptovat jistou odchylku, která do systému vnáší nejistotu, jež může představovat bezpečnostní riziko.

Velkou výhodou je ale nepřenositelnost autentizačního faktoru, jelikož je vázán na konkrétní osobu a nelze ho jednoduše napodobit někým jiným. Tento způsob autentizace je tedy vhodný pro docházkové systémy, jelikož prakticky eliminuje podvádění v podobě *buddy-punchingu*.

V případě EDS se patrně nejvíce používá biometrická autentizace pomocí otisku prstu [27]. Důvodem je jednak jednoduchost použití, rychlost autentizace a také nízká cena takového systému – u nejlevnějších modelů čteček otisků prstů se dnes pohybuje kolem 200 Kč.

## 2 NÁVRH TERMINÁLU EDS

### 2.1 Požadavky

Před vypracováním návrhu terminálu bylo nutné definovat požadavky, které má systém měl ve finální podobě splňovat. Tyto požadavky vychází z předpokládaného nasazení terminálu v malé firmě – pro řádově jednotky až desítky zaměstnanců. Očekává se využití jednoho terminálu na centrálním pracovišti, není tedy nutné řešit provázání více jednotek do kooperujícího systému.

Není nutno použít autentizaci pomocí biometrických údajů – nepředpokládá se zneužívání systému ze strany zaměstnanců v podobě *buddy-punchingu* apod. Je tedy možné použít autentizaci předmětem, např. kartou (s čárovým kódem, kontaktní, bezkontaktní, ...) či RFID čipem v podobě klíčenky, apod.

Mezi základní funkcionalitu poskytovanou systémem (kromě zjevného záznamu odpracovaných hodin jednotlivých zaměstnanců) by mělo patřit generování periodických výkazů pro tvorbu mzdových podkladů.

Systém by měl také umožňovat uživatelům zadat doplňující informace. Ty lze očekávat jak v předem známém formátu, vhodném pro strojové zpracování (např. předpokládaný čas příchodu na pracoviště), tak ve formátu zcela obecném, umožňujícím vložení textových poznámek (důvod předčasného odchodu, vzkaz pro kolegu, apod.).

Veškerá data získaná systémem by měla být uložena ve vhodném formátu umožňujícím (i zpětně) nejen snadné strojové zpracování, ale také zobrazení ve formátu snadno čitelném člověkem.

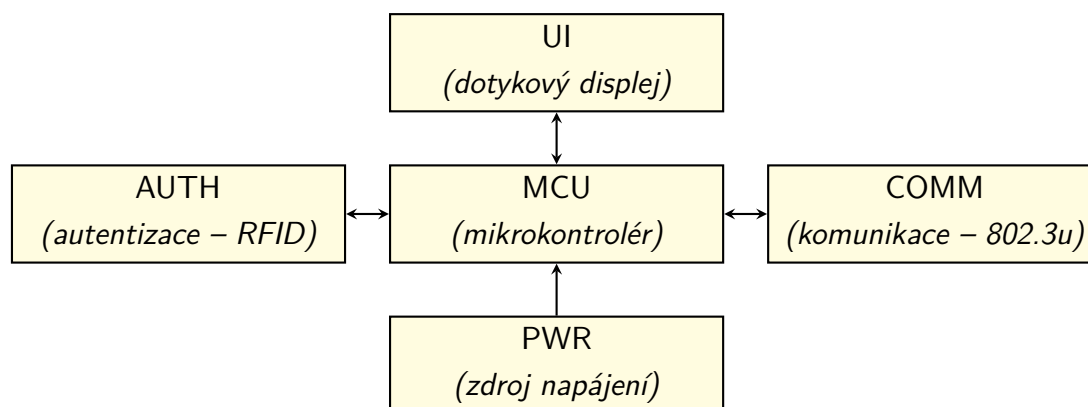
Vzhledem k důležitosti zpracovávaných dat (jsou na nich přímo závislé personální a finanční operace podléhající legislativě) by měl systém data průběžně zálohovat na vhodné médium, např. síťový disk, server připojený do internetu, apod.

Jelikož předchozí bod předpokládá připojení terminálu do počítačové sítě, jeví se jako vhodné doplnit jej také možností vzdáleného přístupu (např. přes webový prohlížeč) přinejmenším pro čtení aktuálního stavu – přítomnost zaměstnanců na pracovišti, čas jejich příchodu a další informace.

Terminál by měl disponovat přehledným, uživatelsky přívětivým grafickým rozhraním. Po mechanické stránce by mělo jít o kompaktní zařízení vhodné pro instalaci na stěnu, případně do panelu.

## 2.2 Koncepce

V předchozí sekci jsou uvedeny požadavky, které výrazně ovlivnily volbu hardwarové koncepce zařízení. Obecnou koncepci zařízení lze vidět v blokovém schématu na obrázku 2.1. Jednotlivé bloky reprezentují fundamentální funkce poskytované terminálem, popsané v následujících podkapitolách.



Obr. 2.1: Blokové schéma koncepce terminálu EDS.

### 2.2.1 Uživatelské rozhraní

Z pohledu uživatele je patrně nejdůležitější provedení uživatelského rozhraní (UI, z anglického *User Interface*), se kterým bude denně pracovat. Mělo by být jednoduché, intuitivní a přehledné, ale zároveň dostatečně flexibilní pro případné rozšíření funkcionality celého systému.

Z těchto důvodů padla volba na dostatečně velký barevný LCD displej, doplněný dotykovou vrstvou (*touchscreen*), který prakticky všichni znají z mobilních telefonů, respektive tabletů. Při vhodně provedeném návrhu uživatelského rozhraní v docházkovém softwaru lze očekávat, že s ovládáním terminálu nebudou mít uživatelé sebemenší problém právě díky použití jim známého konceptu.

Tabulka 2.1 obsahuje konkrétní technické parametry displeje použitého v popisovaném návrhu.

### 2.2.2 Autentizace

Jak již bylo uvedeno v části věnované požadavkům, předpokládá se ověření identity uživatelů za pomoci autentizačního předmětu, např. čipové karty. Pro jednoduchost použití (z pohledu uživatelů) byly zvoleny bezkontaktní RFID tagy dle standardu ISO/IEC 14443 A [20], které jsou v podobných aplikacích hojně rozšířené. Princip funkce těchto tagů je stručně popsán v části 1.3.1.

výrobce	LG Display
typové označení	LP101WX1-SLB1
úhlopříčka	25,65 cm (10,1 palce)
rozlišení	1280×800 px
podsvětlení	LED, 340 cd/m <sup>2</sup>
připojení panelu	LVDS (1 kanál)
dotyková vrstva	kapacitní, USB výstup

Tab. 2.1: Technické parametry použitého LCD displeje [21].

Implementace RFID technologie bude v docházkovém terminálu realizována pomocí modulu SL031 [6] od firmy StrongLink. Tento modul umožňuje komunikaci s RFID tagy dle standardu ISO/IEC 14443 A. Pro komunikaci modul disponuje rozhraním UART pracujícím v TTL úrovních, díky čemuž ho lze použít prakticky do jakékoliv aplikace, jelikož jsou tímto rozhraním vybaveny téměř všechny běžně používané mikrokontroléry.

### 2.2.3 Komunikace

Díky požadavku na připojení terminálu do počítačové sítě, respektive jeho schopnosti ukládat data na síťový disk, je nutné jej vybavit síťovým rozhraním. Vzhledem k tomu, že terminál bude umístěn stacionárně, není důvod k použití bezdrátového spojení technologií Wi-Fi (IEEE 802.11). Pro připojení do počítačové sítě tak bude použit standard IEEE 802.3u [19], známý spíše jako Fast Ethernet.

S použitím ethernetu pro přístup do počítačové sítě se nabízí také možnost využít napájení po ethernetu (anglicky *Power over Ethernet*, zkráceně PoE) definované standardem IEEE 802.3af [18]. Tato možnost ale nebude v práci dále rozebrána, jelikož se nepředpokládá nasazení terminálu v prostředí kde se již technologie PoE využívá a její zavedení pro jediný prvek – docházkový terminál – nemá význam.

### 2.2.4 Minipočítač

V předchozí sekci byly stručně popsány jednotlivé komponenty, které je potřeba v systému použít. V blokovém schématu 2.1 je uveden také blok MCU, který reprezentuje mikrokontrolér implementující řídicí logiku terminálu.

Volba konkrétního mikrokontroléru se samozřejmě odvíjí od použitých periférií. Zatímco uvedená čtečka bezkontaktních karet používá pro svou komunikaci rozhraní UART, kterým disponují prakticky všechny běžně dostupné MCU, připojení LCD displeje pomocí LVDS rozhraní omezuje výběr podstatně více.

Jelikož klasické mikrokontroléry většinou LVDS rozhraním nedisponují, je nutné volit řešení v podobě *systemu na čipu*, zkráceně SoC (z anglického *System on Chip*). Bohužel použití SoC řešení výrazně zvyšuje požadavky na návrh obvodového řešení, především v oblasti signálové integrity.

Protože takový návrh není hlavním tématem této práce, bylo rozhodnuto o použití běžně dostupného jednodeskového minipočítače, který vhodný SoC využívá. Pravděpodobně nejznámějším představitelem této kategorie je Raspberry Pi, které ale LVDS rozhraním nedisponuje a proto není pro tuto aplikaci vhodné [15].

Volba padla na platformu Allwinner A20, která je v současné době velice rozšířenou mimo jiné v tabletech s OS Android. Základní technické parametry tohoto SoC řešení jsou uvedeny v tabulce 2.2.

CPU	dvě jádra Cortex-A7
GPU	Mali400MP2
paměť RAM	až 2 GB DDR2/DDR3
video výstup	HDMI/VGA/LVDS
I/O rozhraní	USB 2.0, UART, I <sup>2</sup> C, I <sup>2</sup> S, SPI

Tab. 2.2: Technické parametry SoC Allwinner A20 [2].

Zařízení postavených na této platformě se objevuje stále více, především z dílen čínských výrobců. Kromě již zmíněných tabletů jsou to také nejrůznější HTPC pro přehrávání multimédií, ale také jednodeskové počítače určené spíše pro vývojáře k zástavbě do komplexnějších aplikací.

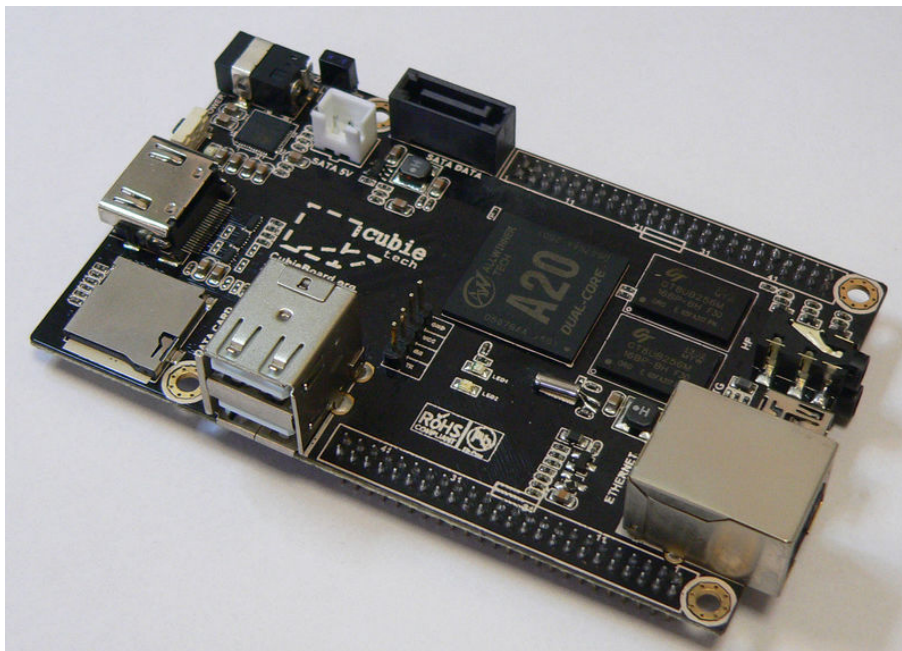
Vzhledem k poměrně specifickým požadavkům (rozhraní LVDS, UART a USB dostupná pro propojení na jinou desku) byl zvolen jednodeskový počítač Cubieboard2, jehož technická specifikace je uvedena v tabulce 2.3.

platforma	Allwinner A20
paměť RAM	1 GB DDR3 SDRAM
úložiště dat	4 GB flash paměti
rozměry	100×60 mm
I/O konektory	HDMI, USB, LAN, SATA, audio, ...

Tab. 2.3: Technické parametry minipočítače Cubieboard2 [13].

Velkou výhodou tohoto modelu je vyvedení většiny rozhraní (tedy těch, u kterých to má smysl) na hřebínkové lišty v rastru 2 mm. Kromě GPIO a rozhraní vyvedených na daných pinech (tj. UART, SPI, I<sup>2</sup>C, I<sup>2</sup>S, ...) je na tento způsob připojení nachystáno i USB a také napájení celého minipočítače.

Díky tomu je velmi snadné vytvořit DPS s dalšími komponentami, případně propojovacími konektory. Návrhem takové základní desky pro potřeby docházkového terminálu se zabývá následující podkapitola.



Obr. 2.2: Minipočítač Cubieboard2.

## 2.3 Obvodové řešení základní desky

V předchozí kapitole byla uvedena bloková koncepce řešení docházkového terminálu a stručně popsány jednotlivé komponenty, ze kterých se terminál bude skládat. Jelikož je minipočítač Cubieboard2 připraven pro přímé spojení s jinou DPS, byl zvolen koncept základní desky zajišťující elektrické propojení všech komponent. Při jejím návrhu bylo potřeba vyřešit následující:

- napájení minipočítače Cubieboard2 (5 V/1 A),
- napájení LCD displeje (3,3 V kontrolér, 4,5 až 21 V podsvětlení),
- propojení LCD displeje s minipočítačem (LVDS pro obrazová data, USB pro dotykovou vrstvu),
- propojení RFID čtečky s minipočítačem (napájení, rozhraní UART),
- napájení hodin reálného času (RTC) napětím cca 1,3 V.

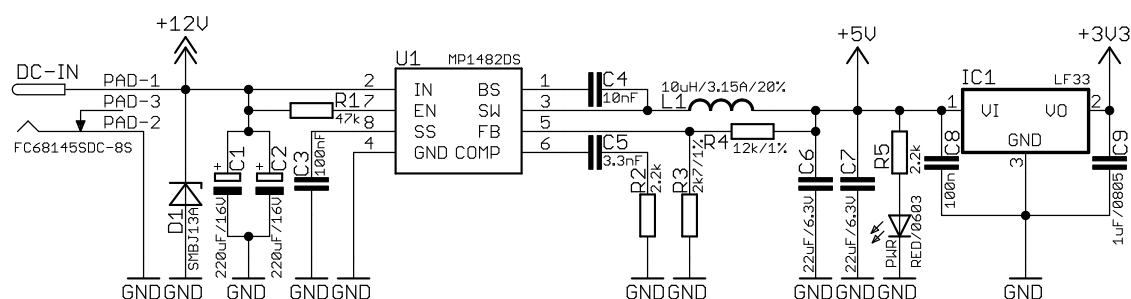
Po mechanické stránce byla zvolena velikost 100×60 mm, tedy stejný rozměr jako má minipočítač Cubieboard2. Navržená základní deska se bude k minipočítači připojovat zespodu, pomocí hřebínkových lišt s rastrem 2 mm, které jsou již na desce minipočítače osazeny, viz fotografii 2.2. Tyto konektorové lišty mají při spojení

s protikusem výšky cca 6,5 mm, od které se odvíjel výběr jednotlivých součástek, respektive jejich pouzder tak, aby je bylo možné mezi obě DPS umístit.

Jelikož bylo již v rané fázi návrhu patrné, že návrh nebude nijak striktně rozměrově omezen, bylo též rozhodnuto o osazení nízkofrekvenčního zesilovače. Díky tomu se otevírá možnost doplnění uživatelského rozhraní o zvuková upozornění, například potvrzení načtení karty apod.

### 2.3.1 Napájecí část

Napájecí část je koncipována tak, aby bylo možné použít k napájení celého terminálu externí (zásuvkový) spínaný zdroj 12 V/1 A. Toto napětí je využito k napájení podsvětlení LCD displeje a přivedeno na vstup měniče MP1482DS [26], který na svém výstupu vytváří napětí 5 V.



Obr. 2.3: Schéma napájecí části terminálu.

Napětí 5 V je použito k napájení minipočítače Cubieboard2 a vyvedeno na konektor pro připojení modulu RFID čtečky. Dále je přivedeno na vstup regulátoru LF33 [36], který vytváří napětí 3,3 V pro napájení logiky LCD displeje a také nízkofrekvenčního zesilovače.

Pro připojení externího zdroje byl vybrán souosý konektor. Jelikož byl návrh rozměrově omezen maximální výškou, nešlo použít standardní typ 5,5/2,1.<sup>(1)</sup> Byl proto zvolen konektor C68145S [12] od firmy Cliff, který používá vnější průměr 3,5 mm, vnitřní pak 1,3 mm. Výška tohoto konektoru je pouze 4,7 mm, takže jej není problém umístit mezi obě DPS.

### Napájení hodin reálného času

Pro správnou činnost docházkového terminálu je nutné, aby disponoval zdrojem reálného času. SoC Allwinner A20 jako takový v sobě blok hodin reálného času (RTC, z anglického *Real-Time Clock*) obsahuje, takže není nutné je doplnit externě.

<sup>(1)</sup>Označení konektoru značí odkazuje na jeho rozměry – průměr vnějšího kontaktu 5,5 mm, průměr vnitřního kontaktu 2,1 mm.

Bohužel s touto variantou patrně nepočítali tvůrci minipočítače Cubieboard2, který není připraven na osazení baterie pro zálohování napájení RTC obvodu. Bohužel není tato napájecí větev ani vyvedena na některý z propojovacích konektorů. Pro napájení RTC obvodu tak bylo nutné desku Cubieboard2 částečně modifikovat.

Napětí  $VDD_{RTC}$  použité pro napájení hodin reálného času je získáváno z výstupu LDO1 napěťového regulátoru AXP209. Tato napájecí větev je připojena pouze na SoC A20 a dva blokové kondenzátory (C65 a C123). Kondenzátor C123 je umístěn v blízkosti IR přijímače U9, který je osazen v THT pouzdře [1]. Jelikož v popisované aplikaci není zamýšleno jakékoliv využití IR přijímače, lze jej odstranit a na jeho místo osadit hřebínkovou lištu, která bude použita pro přivedení napětí ze základní desky.

Pro montáž lišty byl zvolen napájecí pin (VSS) IR přijímače, protože jej lze snadno elektricky odpojit od dalších částí minipočítače. Tento pin je spojen blokovacím kondenzátorem C156 se zemí a rezistorem R71 s napájecí větví 3,3 V [1]. Odstraněním těchto součástek byl získán prokov vhodný pro osazení lišty. Pro smysluplné využití tohoto spoje k přivedení napětí do obvodu RTC je nutné jej propojit s napájecí větví  $VDD_{RTC}$ , což bylo realizováno drátovou propojkou na kondenzátor C123, který se nachází v těsné blízkosti zmiňovaného prokovu.

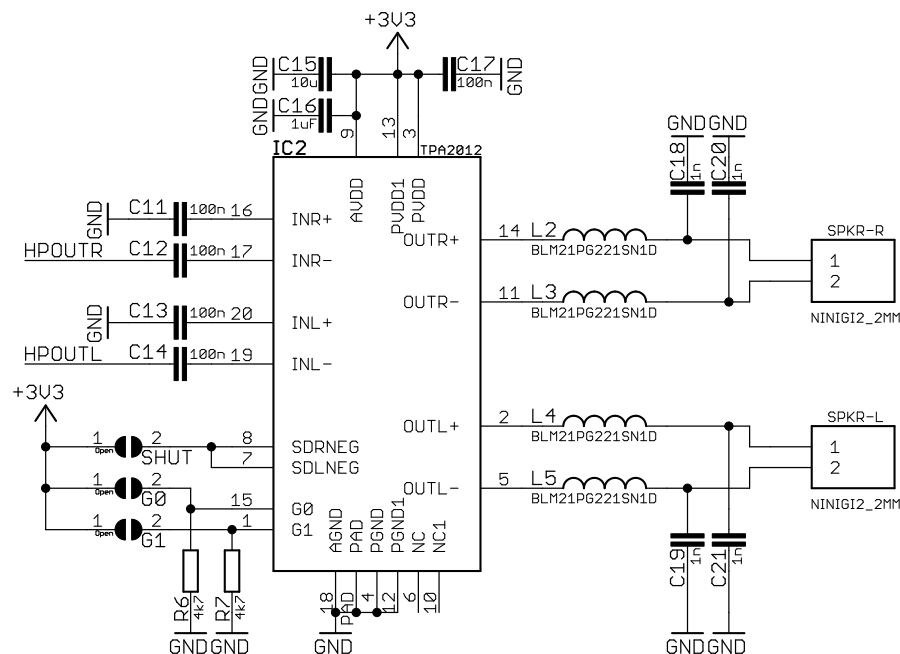
Na základní desce je jako zdroj napětí pro hodiny reálného času osazen držák, ve kterém je umístěna baterie na bázi oxidu stříbrného s nominálním napětím 1,55 V. Jelikož je při provozu na napájecí větví RTC obvodu napětí cca 1,3 V, je mezi baterií a tuto větev vložena schottkyho dioda typu BAT165, na které jednak vznikne úbytek cca 0,3 V a také zabrání dobíjení baterie z napájecí větve  $VDD_{RTC}$ .

Během stavby prototypu bylo provedeno měření odběru RTC obvodu pro odhad životnosti baterie. Měřením byla zjištěno, že velikost odebíraného proudu se pohybuje kolem 7  $\mu$ A. V katalogového listu použité baterie VARTA V391 [38] je uvedena typická kapacita baterie 42 mAh. Vyjdeme-li z těchto hodnot, vyjde nám životnost baterie cca 6000 hodin. Jelikož se očekává, že baterie bude RTC obvod napájet pouze výjimečně (při výpadku napájení), lze tuto hodnotu považovat za dostatečnou.

### 2.3.2 Nízkofrekvenční zesilovač

Protože nejsou na NF zesilovač kladeny nijak specifické nároky (bude použit pouze pro reprodukci jednoduchých zvukových upozornění), byl použit integrovaný stereo zesilovač TPA2012D pracující ve třídě D. Nabízí výkon až 2,1 W na kanál při impedanci reproduktorů 4  $\Omega$  a napájení 5 V, což pro dané použití více než dostačuje. Použité zapojení vychází z typického zapojení uvedeného v katalogovém listu [37].





Obr. 2.4: Schéma nízkofrekvenčního zesilovače.

### 2.3.3 Konektory pro připojení periférií

Navržená DPS je osazena konektory pro propojení s dalšími dílčími částmi terminálu, jmenovitě s:

- LCD displejem – datový konektor (obrazová data a dotyková vrstva) a napájení podsvětlení,
- RFID čtečkou – napájení (5 V) a rozhraní UART,
- stereo reproduktory – dva samostatné konektory (levý a pravý kanál).

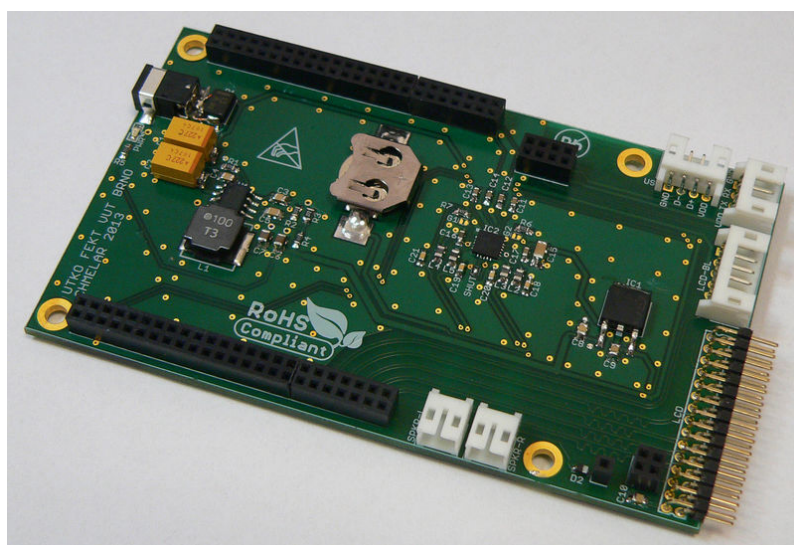
Sestava LCD displeje LP101WX1-SLB1 (displej, dotyková vrstva a kontrolér) je osazena 40pinovým miniaturním konektorem s roztečí 0,5 mm (označení UJU IS050-L40B-C10) [21]. Pro připojení sestavy k základní desce byl použit standardně vyráběný kabel, který na straně LCD displeje disponuje kompatibilním protikusem, na straně druhé pak dutinkovou lištou 2×15 pinů rozteče 2 mm. Aby nebyla porušena podmínka maximální výšky součástek, je na základní desce protikus (kolíková lišta 2×15 pinů) osazena v úhlové variantě. Pro napájení displeje je na kabelu ještě 6pinový konektor NINIGI NXG-06 [28], ke kterému je na desce umístěn úhlový protikus NXW-06K [29].

Pro připojení dalších komponent byly použity konektory stejné řady. Pro RFID čtečku je tak připraven 4pinový konektor NXW-04K, který obsahuje jak napájecí napětí 5 V, tak datové linky rozhraní UART. Další dva 2pinové konektory jsou pak určeny pro připojení stereo reproduktorů.

## 3 REALIZACE DPS

V předchozích částech práce byla teoreticky popsána koncepce docházkového terminálu a navrženo možné řešení s použitím konkrétních technologií. Následně bylo rozebráno navrhované obvodové řešení, zakončené návrhem DPS základní desky.

Na základě tohoto návrhu byl realizován prototyp základní desky, na kterém byly jednotlivé funkce ověřeny. Během osazování a oživování prototypu bylo objeveno několik drobných chyb, které byly opraveny a relevantní změny byly promítnuty do schématu, respektive návrhu DPS.



Obr. 3.1: Prototyp základní desky.

### 3.1 Nalezené chyby

Jak již bylo zmíněno, během oživování vyšlo najevo několik chyb v návrhu prototypu. Žádná z těchto chyb ale nezpůsobila vážnější problémy, jelikož je bylo možné snadno odstranit. Jednotlivé chyby jsou stručně popsány v následujících odstavcích.

#### 3.1.1 Napájecí konektor

Použitý napájecí konektor není standardní součástí a proto není obsažen v knihovnách navrhovaného systému EAGLE, ve kterém byl návrh realizován. Z tohoto důvodu bylo nutné do vlastní knihovny doplnit záznam reprezentující tento typ konektoru.

Katalogový list výrobce bohužel obsahuje chybu v popisu jednotlivých pinů, což způsobilo nesprávný návrh DPS (souhlasil s katalogovým listem, ale ne se skutečnou

součástí). Na prototypu byl tento problém vyřešen drátovými propojkami, v návrhu DPS (příloha A) je tento problém již opraven.

### 3.1.2 Připojení LCD panelu

Během návrhu DPS došlo omylem k přehození pořadí kontaktů v konektoru sloužícím pro napájení podsvětlení LCD panelu. Tuto chybu bylo možné vyřešit záměnou jednotlivých kontaktů, jelikož je lze z pouzdra konektoru vyjmout a vložit zpět.

Také datový konektor pro LCD nebyl zapojen zcela správně – zde došlo k přivedení napájení na kontakt, který není v kabelu zapojen. Vzhledem k tomu, že napájení vedou i další dva vodiče, nebyla zde modifikace kabelu zcela nutná, nicméně byla provedena. Obě popisované chyby jsou v přiloženém návrhu DPS opraveny tak, aby nebylo nutné kabel jakkoliv modifikovat.

## 3.2 Ověření funkce

Vyjma výše uvedených chyb nebyly na prototypu nalezeny žádné další nedostatky a prototyp plní očekávané funkce. V rámci testování byla úspěšně provedena zkouška funkce následujících částí:

- minipočítač Cubieboard2 (testováno načtení OS GNU/Linux z SD karty, připojení USB zařízení, práce v síti – rozhraní ethernet);
- LCD panel – po úpravě kabelu funguje bez problému;
- dotyková vrstva – po instalaci ovladače funguje dle očekávání;
- RFID čtečka – detekuje přítomnost tagu (další funkcionality jako zápis paměťových bloků nebyla ověřována);
- nízkofrekvenční zesilovač – pracuje dle očekávání, výkon se zdá být pro daný účel více než dostatečný;
- napájení obvodu hodin reálného času z baterie.

Vzhledem k tomu, že prototyp splnil očekávání a nebylo nutné dělat další úpravy v návrhu DPS, mohl autor upřít pozornost na softwarové řešení terminálu. Následující kapitola je proto věnována podpoře SoC řešení Allwinner A20 v prostředí operačního systému GNU/Linux, který je pro provoz docházkového terminálu použit.

## 4 LINUX NA PLATFORMĚ ALLWINNER A20

Zařízení postavených na čipech firmy Allwinner řady „A“-Series existuje poměrně velké množství.<sup>(1)</sup> Právě díky svému masivnímu rozšíření, způsobenému především velice příznivou cenou hardwarových komponent, se tato platforma těší značné podpoře především ze strany distribucí OS GNU/Linux.

Následující stránky jsou věnovány stručnému představení odlišností, na které může uživatel, který by se chtěl se zmíněnou platformou blíže seznámit, ve srovnání s „klasickým“ nasazením Linuxu,<sup>(2)</sup> narazit.

Nebude-li uvedeno jinak, informace uvedené v této kapitole jsou převzaty především z [23, 35].

### 4.1 Projekt linux-sunxi

V současné době (květen 2014) je hlavní vývoj softwaru pro platformu Allwinner A20 soustředován do projektu *linux-sunxi*. Hlavním cílem tohoto projektu je vytvoření softwarové podpory pod svobodnou licencí pro SoC Allwinner řady „A“-Series,<sup>(3)</sup> mezi které patří i A20, jímž je osazen minipočítač Cubieboard2 použitý v rámci této práce (viz část 2.2.4).

Ačkoliv se počet aktivních vývojářů projektu linux-sunxi pohybuje pouze kolem dvou desítek, jsou jejich výsledky na velmi dobré úrovni a vzniklý software lze označit za stabilní a výborně použitelný.

Nejdůležitějším softwarem, který v rámci projektu vznikl, je odnož (anglicky *fork*) jádra projektu GNU/Linux, do něhož bylo nutné začlenit ovladače specifické pro danou platformu. Druhým, neméně podstatným celkem je pak zavaděč (*bootloader*), který slouží pro prvotní inicializaci hardwaru, načtení Linuxového jádra do paměti a jeho spuštění.

Opomenout nelze ani rozsáhlou sbírku konfiguračních souborů specifických pro jednotlivá zařízení, ať již jde o tablety, jednodeskové počítače (SBC, z anglického *Single Board Computer*) nebo vývojové kity. Díky zmíněným konfiguračním souborům, které jsou pro platformu Allwinner specifické, je možné po prvotním seznámení se s některým ze zařízení velice snadno přejít na zařízení odlišné.

---

<sup>(1)</sup>Poměrně obsáhlý seznam zařízení lze najít na <http://linux-sunxi.org/Category:Devices>.

<sup>(2)</sup>V tomto případě je „klasickým“ nasazením myšleno především použití Linuxové distribuce na platformě x86, respektive x86\_64, se kterými se zatím lze setkat nejčastěji.

<sup>(3)</sup>Název projektu *linux-sunxi* odkazuje na interní označení těchto čipů jejich výrobcem, které je sun4i (A10), sun5i (A10s, A13), sun7i (A20), atd.

### 4.1.1 Zavaděč u-boot-sunxi

Zavaděč vzniklý v rámci projektu linux-sunxi je odnož zavaděče *U-Boot*, který je velice populární a často používaný na mnoha embedded platformách především pro svou univerzálnost a široké možnosti konfigurace. [14]

Zavaděč je schopen provést inicializaci hardwaru a načtení Linuxového jádra z SD/MMC karty. Na rozdíl od zavaděče dodávaného výrobcem SoC, firmou Allwinner, není zavaděč z projektu linux-sunxi schopen provést zavedení systému z NAND paměti integrované na desce minipočítače Cubieboard2. To ovšem pro potřeby této práce nevadí, jelikož možnost použít jako zdroj systému SD kartu je z hlediska vývoje vhodnější – především proto, že se s ní snadněji pracuje. [24]

### 4.1.2 Jádro linux-sunxi

Jelikož není platforma Allwinner A20 v současné době plně podporována přímo v hlavní větvi linuxového jádra,<sup>(4)</sup> je nutné použít specifickou vývojovou větev, spravovanou vývojáři projektu linux-sunxi. Zdrojové kódy této větve jsou v současnosti dostupné pomocí služby GitHub na adrese <https://github.com/linux-sunxi/linux-sunxi>.

Uvedená větev obsahuje ovladače specifické pro tuto platformu, které je nutné pro provoz operačního systému na bázi Linuxu na minipočítači Cubieboard2 použít. Pro potřeby této práce je podstatná především podpora těchto komponent:

- řadič SD karty,
- sériové rozhraní UART,
- rozhraní LVDS pro LCD displej,
- ethernetový řadič,
- zvukový kodek.

## 4.2 Obraz operačního systému

Vzhledem k tomu, že je minipočítač Cubieboard2 vybaven poměrně výkonným hardwarem a nabízí velmi bohatou množinu vstupně/výstupních rozhraní (viz část 2.2.4), lze na něj pohlížet jako na plnohodnotný osobní počítač (PC). Díky tomu je podporován několika Linuxovými distribucemi, které umožňují použití jednodeskového počítače v roli PC.

---

<sup>(4)</sup>Hlavní větev *kernelu* (linuxového jádra) je označována *mainline*. Vývojáři projektu linux-sunxi průběžně zasílají patche (sady změn), které jsou do této větve postupně začleňovány, ale v současnosti (květen 2014) chybí podpora mnoha rozhraní, např. LVDS pro displej.

V současné době patří mezi distribuce s nejlepší podporou hardwaru na platformě Allwinner především *Fedora* [16]. Alternativou je pak odnož distribuce Arch Linux zvaná *Arch Linux ARM*, která se na systémy postavené na procesorech architektury ARM přímo specializuje [3]. V případě použití těchto distribucí stačí pro instalaci systému prakticky pouze stažení binárního obrazu a její zápis na SD kartu.

#### 4.2.1 Sestavení vlastního obrazu

Další z možností získání vhodného binárního obrazu systému, která ovšem již není tak jednoduchá jako instalace některé z výše zmíněných distribucí, je jeho sestavení přímo dle potřeb uživatele. Zatímco z hlediska jednoduchosti jde o řešení řádově složitější, z hlediska možnosti konfigurace a úprav výsledného systému je zde patrná jasná výhoda.

Pro sestavení funkčního obrazu je zapotřebí zkombinovat tři části – zavaděč (bootloader), jádro systému (kernel) a *rootfs*, tedy kořenový souborový systém,<sup>(5)</sup> obsahující nejdůležitější data pro běh systému (knihovny, systémové aplikace, konfigurační soubory apod.). Zavaděč a jádro systému lze získat z již představeného projektu linux-sunxi ve formě zdrojových kódů, které je možné po správné konfiguraci sestavit překladačem do vhodného formátu.

Poslední část, tedy *rootfs*, je možné získat vícero způsoby. Vývojáři projektu linux-sunxi je doporučováno použití existujícího *rootfs* z projektu *Linaro*, který momentálně nabízí čtyři různě zaměřené *rootfs* archivy, ze kterých je možné vybrat si ten nejvhodnější pro danou aplikaci [22]. Druhou možností je pak vytvoření vlastního *rootfs*, k čemuž lze použít automatizované nástroje, které tuto operaci výrazně zjednoduší. Příkladem takového nástroje je *Buildroot*, který byl při realizaci této práce využit a proto mu bude věnována následující kapitola.

---

<sup>(5)</sup>Z důvodu délky přeloženého pojmu bude nadále používán originální tvar.

## 5 BUILDROOT

V předchozí kapitole bylo stručně uvedeno z čeho se skládá obraz systému pro takové zařízení, jakým je minipočítač Cubieboard2. Tato kapitola bude věnována popisu systému Buildroot, který byl úspěšně použit pro tvorbu binárního obrazu systému pro toto zařízení.

Buildroot je nástroj, který dokáže významně ulehčit přípravu obrazu systému pro prakticky libovolný embedded systém. Umožňuje z větší části automatizovat tvorbu obrazu od začátku až do konce, jelikož je schopen realizovat následující:

- sestavení sady nástrojů pro křížový překlad (*cross-compilation toolchain*),<sup>(1)</sup>
- nastavení a překlad zavaděče (bootloader),
- konfiguraci a překlad linuxového jádra (kernel),
- tvorbu rootfs archivu s vybraným softwarem,
- přípravu vytvořeného systému pro nasazení na cílové zařízení.

Díky tomu, že po technické stránce je Buildroot realizován jako sada *Makefile*<sup>(2)</sup> souborů, které jsou logicky uspořádány a dobře zdokumentovány, není problém systém využít i pro platformy, které zatím nejsou tímto nástrojem oficiálně podporovány. Pro mnoho platform stačí Buildroot pouze jinak nastavit, v některých případech je nutné doplnit specifické balíky (např. zavaděč). Právě vysoká flexibilita a možnost velice detailně ovlivnit prakticky všechny kroky přípravy obrazu systému, dělají z tohoto nástroje ideální řešení pro úzce specializované embedded systémy.

Vzhledem k rozsahu možností nabízených nástrojem Buildroot není možné je v této práci podrobně popisovat. Proto budou následně uvedeny pouze možnosti, které byly při zpracování praktické části práce využity. Pro podrobnější seznámení se s tímto nástrojem je možné doporučit zdroje [9, 25].

### 5.1 Konfigurace

Jak již bylo napsáno výše, funkcionalita Buildrootu je postavena na Makefile souborech obsahujících instrukce pro sestavení jednotlivých částí, doplněných o konfigurační soubory systému *Kconfig*, známého především z konfigurace linuxového jádra. Systém *Kconfig* používá k uložení konfigurace jednoduché textové soubory obsahující data ve formátu **výraz = hodnota**, díky čemuž je možné je upravovat prakticky libovolným textovým editorem.

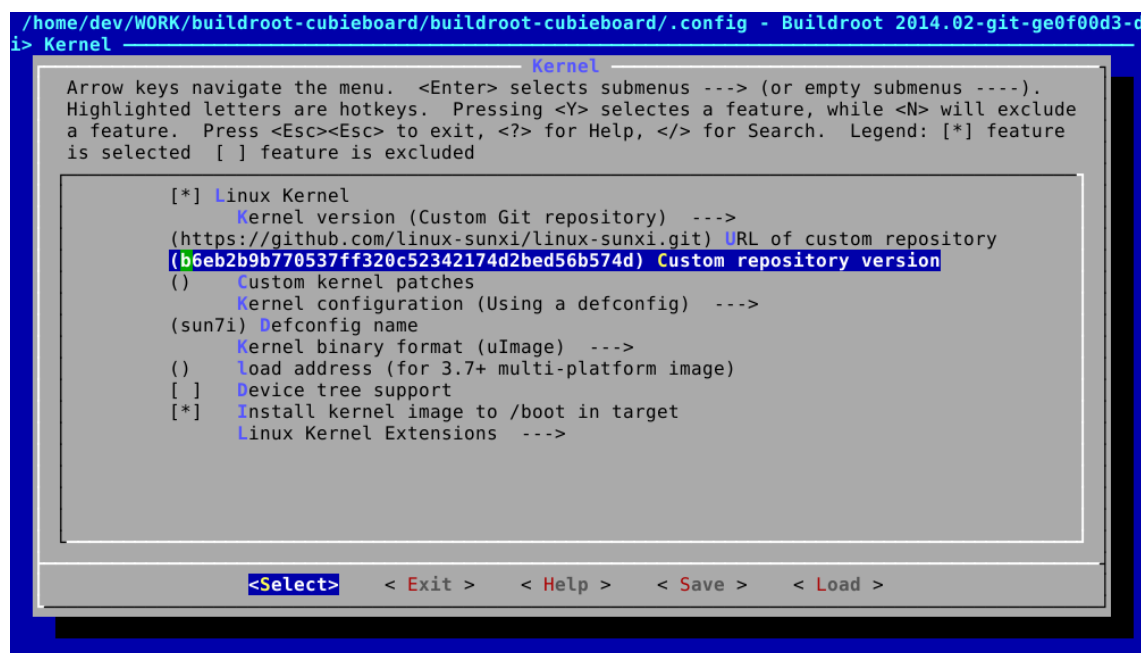
---

<sup>(1)</sup>Křížový překlad (*cross-compilation*) je sestavení binárních souborů pro jinou architekturu, než na jaké je překlad prováděn. Příkladem může být kompilace softwaru pro architekturu *ARM* na architekturu *x86\_64*.

<sup>(2)</sup>*Makefile* soubor obsahuje množinu pravidel, na základě kterých nástroj *make* dokáže určit co, jak a v jakém pořadí bude přeloženo.

Protože ale taková konfigurace není pohodlná, obsahuje Buildroot několik konfiguračních menu, ve kterých lze velmi snadno a přehledně nastavit veškeré parametry výsledného systému. Buildroot nabízí více podob těchto konfiguračních nabídek, pro jednoduchost ale bude věnována pozornost pouze tzv. *menuconfig*, který je nejpoužívanější.

Pro vstup do konfigurační nabídky je zapotřebí z adresáře, kde je Buildroot umístěn, spustit příkaz `make menuconfig`. Buildroot zpracuje vstupní konfigurační soubory pro jednotlivé balíčky a nabídne hierarchické menu, ve kterém lze jednotlivé balíčky a jejich parametry nastavit.



Obr. 5.1: Konfigurační menu nástroje Buildroot.

Jak taková konfigurační nabídka vypadá je možné vidět na obr.5.1, kde je zobrazena konfigurace specifické větve a verze linuxového jádra, které Buildroot pro sestavení obrazu použije.

Jelikož postrádá smysl uvádět použité nastavení přímo v textu, budou stručně uvedeny pouze nejdůležitější části. Kompletní konfiguraci je možné nalézt v příloze C.1.

### 5.1.1 Cílová architektura

Jedním z nejdůležitějších parametrů, které je potřeba v systému Buildroot nastavit, je architektura cílové platformy (zařízení, na němž výsledný systém poběží).

Nastavení cílové architektury ovlivňuje formát výstupních binárních souborů a proto je nutné, aby bylo provedeno v souladu se skutečným hardwarem. Pokud



nebudou parametry nastaveny správně, nebude možné vzniklý systém na daném zařízení pravděpodobně vůbec spustit.

Konfigurace cílové platformy byla přizpůsobena SoC řešení Allwinner A20, které obsahuje dvě výpočetní jádra *ARM Cortex-A7*. Ta jsou postavena na architektuře ARMv7 a používají rozšířenou instrukční sadu *NEON*.

### 5.1.2 Sada nástrojů pro překlad (toolchain)

K tomu, aby byl schopen nástroj Buildroot sestavit obraz systému pro cílové zařízení, je zapotřebí disponovat sadou nástrojů pro překlad, nazývanou *toolchain*.<sup>(3)</sup> Ta obecně obsahuje překladač, knihovny, hlavičkové soubory a další podpůrné nástroje, bez kterých se sestavení obrazu neobejde.

Jak již bylo uvedeno, Buildroot je velmi flexibilní nástroj nabízející mnoho možností. Nejinak je tomu i v případě toolchainu, u něhož si lze vybrat z toolchainu sestaveného přímo pomocí Buildrootu, nebo toolchainu externího, který je možné buď stáhnout hotový (např. z projektu Linaro) nebo jej vytvořit (třeba pomocí nástroje *Crosstool-NG*).

Použití toolchainu sestaveného Buildrootem (na základě zvolené konfigurace) je pro začátek jednodušší, což je ale vyváženo jistými nevýhodami (viz dokumentaci nástroje Buildroot [9]). V rámci této práce ovšem byla tato možnost úspěšně použita bez jakýchkoliv problémů. Samotný výběr toolchainu se provádí v hlavní konfigurační nabídce (lze spustit příkazem `make menuconfig`).

### Knihovna jazyka C

Při použití interního toolchainu lze změnit použitou knihovnu jazyka C (*C library*), která bude v cílovém systému použita. V embedded systémech se často využívá minimalistická knihovna *μClibc*, která je zjednodušenou alternativou ke standardně používané *glibc*. Výhodou knihovny *μClibc* je zejména její malá velikost (ve srovnání s *glibc*). Jistou nevýhodou může být naopak pomalejší implementace některých funkcí.

Alternativou ke knihovně *μClibc* je knihovna *eglibc* (*embedded glibc*), která sice není tak úsporná, nicméně poskytuje vyšší kompatibilitu s knihovnou *glibc*. Právě knihovna *eglibc* byla použita pro zařízení Cubieboard2 v rámci této práce. Důvodem byla nutnost použití ovladačů pro grafické výpočetní jádro (GPU) Mali400, které jsou poskytnuty ve formě binárních souborů sestavených vůči knihovně *glibc* a nejsou tak s knihovnou *μClibc* kompatibilní.

---

<sup>(3)</sup>Stejně jako v případě rootfs, i zde bude pro jednoznačnost a kratší zápis použit nadále originální pojem, místo přeloženého tvaru.

### 5.1.3 Linuxové jádro

Vzhledem k tomu, že v sektoru embedded zařízení není často možné použít hlavní vývojovou větev linuxového jádra (tzv. mainline kernel), je Buildroot na tuto možnost velmi dobře připraven. Jako alternativní zdroj pro získání zdrojových kódů jádra je možné použít jak balík ve formátu tar (tzv. *tarball*), tak repozitář systému pro správu verzí Subversion, případně dnes velmi populární Git.

Protože jsou zdrojové kódy projektu linux-sunxi uloženy v repozitáři systému Git, byla zvolena právě tato možnost. Pokud není vhodné, aby byla při překladu jádra použita poslední dostupná verze, je možné specifikovat konkrétní revizi, která má být stažena. Taktéž lze určit výchozí konfiguraci, kterou Buildroot při konfiguraci jádra použije.

Jelikož je často nutné konfiguraci dostupnou z distribučního kanálu dané větve jádra specificky doladit pro konkrétní nasazení, je možné vyvolat standardní konfigurační nabídku linuxového jádra příkazem `make linux-menuconfig`.

### 5.1.4 Zavaděč u-boot-sunxi

V případě zavaděče byla situace trochu složitější, než v případě linuxového jádra. Buildroot sice podporuje zavaděč U-Boot, ale vzhledem k tomu, že jeho odnože (forky) pro specifické platformy se téměř nevyskytují, není Buildroot připraven na možnost alternativní verze tak dobře, jako v případě jádra.

Proto bylo nutné doplnit do Buildrootu zavaděč *u-boot-sunxi* jako další ze zvolitelných zavaděčů. Díky tomu, že jsou zařízení s SoC Allwinner široce rozšířená, existují uživatelé, kteří již Buildroot pro tuto platformu používají. Jedním z nich je slovenský vývojář Miroslav Bendík, který na webovém portále *LinuxOS.sk* publikoval článek o použití Buildrootu pro vývojový kit *OLinuXino A13 WIFI*,<sup>(4)</sup> osazený čipem Allwinner A13. [7]

Díky tomu stačilo převzít jím vytvořenou konfiguraci pro podporu uvedeného zavaděče, fakticky realizovanou dvěma konfiguračními soubory, které Buildroot zpracuje a následně v konfiguračním menu nabídne možnost vybrat jako použitý zavaděč právě u-boot-sunxi. V nastavení je nutné specifikovat, pro jaké zařízení má být zavaděč nakonfigurován. Poté již nic nebrání překladu zdrojových kódů, jehož výsledkem je obraz zavaděče určený pro zapsání na SD kartu.

---

<sup>(4)</sup>Podrobnosti o tomto kitu je možné nalézt na webových stránkách <https://www.olimex.com/Products/OLinuXino/A13/A13-OLinuXino-WIFI/>.

### 5.1.5 Kořenový souborový systém (rootfs)

Jak již bylo uvedeno, rootfs (kořenový souborový systém) označuje množinu adresářů a souborů potřebných pro běh systému. Jde především o systémové utility, knihovny, konfigurační soubory apod. Obsah a velikost rootfs je výrazně ovlivněn tím, k jakému účelu má hotové zařízení sloužit. Na základě požadavků je možné vytvořit minimalistický systém obsahující jen to, co bude opravdu zapotřebí.

#### Busybox

V případě embedded systémů je často jako základ rootfs používán *Busybox*. Jde o program, který dokáže nahradit většinu standardních systémových utilit potřebných pro funkcionality OS GNU/Linux. Pokud je Busybox použit, dojde k nahrazení binárních souborů zmíněných utilit symbolickými odkazy na spustitelný soubor Busyboxu (nejčastěji `/bin/busybox`).

Díky tomu, že je funkcionality sdružena do jednoho souboru, je možné ušetřit nemalé množství paměti, které by jinak samostatné utility zabraly. Nevýhodou tohoto přístupu může být chybějící funkcionality některých utilit, případně také jejich nižší výkon.

Program Busybox je v nástroji Buildroot použit jako výchozí volba pro cílový systém. Pro potřeby této práce nebylo nutné do výchozího nastavení (tj. zahrnutí jednotlivých utilit, apod.) zasahovat. Pokud by ale byla potřeba nastavení změnit, je možné vyvolat konfigurační nabídku příkazem `make busybox-menuconfig`.

#### Knihovna Qt

Autor práce se rozhodl docházkovou aplikaci vytvořit pomocí jazyka *C++* a knihovny *Qt*. Důvodem byla především možnost využití *Qt* knihovny v její embedded verzi, která pro běh grafického uživatelského rozhraní (GUI) nepotřebuje podporu v podobě *X Windows* serveru, se kterým se lze setkat při použití distribucí GNU/Linux na běžném PC.

Knihovna *Qt* totiž umožňuje pro vykreslování GUI využít nízkoúrovňový framebuffer. Nabízí také použití knihovny *tslib* (*touchscreen library*) pro spolupráci s dotykovými displeji. Právě díky těmto vlastnostem je možné aplikace napsané s využitím knihovny *Qt* spustit na embedded systémech bez *X Windows*.

Nástroj Buildroot prostředí *Qt* podporuje přímo v základu, není nutné cokoli doplňovat. Pro jeho využití stačí v hlavní konfigurační nabídce, části *Target packages / Graphic libraries and applications (graphic/text)*, povolit instalaci balíku *Qt5* a nakonfigurovat, které prvky knihovny *Qt* je žádoucí do systému zahrnout. Konfigurace knihovny *Qt* použitá v této práci je patrná z přílohy C.1.

## 5.2 Překlad a sestavení

Dosavadní část práce kapitoly se věnovala doplnění systému Buildroot o zavaděč u-boot-sunxi, který není oficiálně podporovaný, a konfiguraci jednotlivých částí systému pro potřeby vytvoření obrazu vhodného pro minipočítač Cubieboard2. Po konfiguraci je nutné pustit překlad příkazem `make`, který postupně provede – zcela automatizovaně – následující:

1. stažení zdrojových kódů všech potřebných balíků,
2. sestavení toolchainu (sady nástrojů pro překlad) pro nastavenou architekturu,
3. kompilaci všech vybraných balíků (v pořadí daném závislostmi),
4. sestavení jádra a zavaděče ve specifikovaných formátech,
5. vytvoření obrazu rootfs (kořenového souborového systému),
6. spuštění tzv. *post-build* skriptu.

Doba potřebná pro celý proces se odvíjí od počtu a typu vybraných balíků, jež mají být do výsledného systému zahrnuty. Pro sestavení systému v konfiguraci, která je uvedena v příloze C.1, bylo zapotřebí stáhnout celkem 472 MB dat. Samotný překlad (kompilace) pak trval cca 109 minut. Uvedené hodnoty byly naměřeny ve virtuálním stroji, jehož parametry jsou uvedeny v tabulce 5.1.

CPU	Intel Core i5-3340M
RAM	2 GB
úložiště dat	SSD Intel 525
virtualizační prostředí	Oracle VM VirtualBox 4.3.10
virtualizovaný OS	Debian 8 „jessie“ alpha1

Tab. 5.1: Konfigurace virtuálního stroje.

Je vidět, že prvotní sestavení systému je dosti náročný úkol, který i na poměrně výkonném hardwaru není otázkou pár minut. Je ovšem nutné si uvědomit, že výraznou část z této náročnosti tvoří toolchain, který většinou není nutné měnit ani při následných změnách v konfiguraci výsledného systému.

Jednou ze silných stránek nástroje Buildroot je právě fakt, že následné úpravy (např. doplnění další knihovny, změna konfigurace jádra, ...) je možné realizovat poměrně snadno a rychle. V případě změn je totiž nutné kompilovat jen nové či změněné balíky.

### 5.2.1 Úpravy rootfs

I přesto, že Buildroot umožňuje opravdu podrobné nastavení prakticky všech komponent systému, je nutné částečně modifikovat výsledný rootfs. Jde především o doplnění konfiguračních souborů či jednoduchých skriptů.

Postup, kterým byly do rootfs doplněny potřebné změny, vychází z doporučení vývojářů Buildrootu, uvedeném v prezentaci [31]. To navrhuje vytvoření specifické složky obsahující přidané či změněné soubory, které jsou následně do rootfs zkopírovány tzv. post-build skriptem.

Tento skript (je-li povolen) je zavolán po dokončení sestavení všech balíků a jejich instalace do rootfs a současně před tím, než dojde k vytvoření finálního rootfs archivu. Díky tomu je zaručeno, že uvedené změny se do výsledného obrazu skutečně promítnou.

Pro funkčnost EDS terminálu bylo nutné do rootfs zahrnout:

- konfiguraci sítě (IP adresa, maska, ...),
- nastavení knihovny tslib (obsluha dotykového displeje),
- nastavení prostředí Qt.

Uvedené změny jsou (v rámci prostředí Buildroot) obsaženy v adresáři s konfigurací pro minipočítač Cubieboard2 (`board/a20_cubieboard2/rootfs-additions`) a hierarchicky uspořádány dle požadovaného umístění ve výsledném rootfs.

Do post-build skriptu bylo zapotřebí doplnit část kódu, která provede zkopírování změn do cílového adresáře. Výsledná podoba post-build skriptu je uvedena v příloze C.2.

## 5.3 Vytvoření obrazu SD karty

Výsledkem překladu a sestavení systému je struktura rootfs zabalená v archivu formátu tar (tarball), obraz linuxového jádra ve formátu *uImage* a obraz zavaděče u-boot-sunxi. Pro reálné použití je potřeba tato data vhodně zkombinovat, o což se stará skript *make-sdimg.sh*, který byl převzat od Miroslava Bendíka a částečně upraven. Po spuštění tento skript provede tyto kroky:

- vytvoření prázdného souboru dané velikosti,
- připojení oddílu pro zavaděč (*/boot*) a vytvoření souborového systému typu *FAT32*,
- připojení kořenového oddílu (*/*) a vytvoření souborového systému *ext3*,
- zkopírování obrazu jádra, obrazu zavaděče a konfiguračních dat pro zavaděč,
- zkopírování obsahu rootfs do kořenového oddílu.

Výsledkem zmíněných operací je binární obraz, který stačí zapsat na SD kartu dostatečné velikosti. Z takto připravené SD karty je následně minipočítač schopen načíst zavaděč, jádro a celý operační systém GNU/Linux.

## 6 DOCHÁZKOVÁ APLIKACE

Doposud se praktická část práce věnovala pouze řešení terminálu ze stránky hardwaru a operačního systému GNU/Linux, který je schopen na použitém hardwaru běžet a obsluhovat použité periferie.

Obsahem této kapitoly je popis docházkové aplikace, která byla v průběhu práce vyvinuta. Funkce, které by měla realizovaná aplikace, potažmo terminál jako celek, nabízet, byly uvedeny již v kapitole 2.

### 6.1 Koncept

Pracoviště, které bylo pro autora práce impulzem k vývoji EDS terminálu v této podobě, je v jistých ohledech poměrně specifické. Tomu je přizpůsoben i návrh celého systému tak, aby splňoval tyto atypické požadavky.

Výrazná část zaměstnanců jsou studenti prezenčního studia VŠ, kteří se z časových důvodů na pracovišti vyskytují značně nepravidelně. Současně s tím je ovšem zapotřebí mít ze strany vedení možnost plánování práce z důvodu dodržení smluvních termínů zakázek. Z toho důvodu je zapotřebí, aby terminál uživatelům (tj. zaměstnancům) umožňoval uložení předpokládané docházky – informace o tom, kdy s největší pravděpodobností na pracovišti budou či nebudou.

### 6.2 Řešení

Jak již bylo zmíněno v části 5.1.5, pro vytvoření aplikace byla využita knihovna Qt, která umožňuje vývoj grafických aplikací pro poměrně velké množství platform [34]. Při vývoji byla použita nejnovější dostupná verze knihovny Qt, tedy 5.2.1.

Z implementačního hlediska autor rozdělil programový kód do tří oddělených částí. První je věnována čtečce RFID tagů použitých pro autentizaci uživatelů, druhá část má na starost uložení dat do relační databáze typu *SQLite*. Poslední částí je pak implementace uživatelského rozhraní, která obsahuje většinu funkční logiky.

#### 6.2.1 Komunikace s RFID čtečkou

Pro komunikaci s RFID čtečkou je použito sériové rozhraní UART, pro nějž existuje v Qt podpora v podobě třídy *QSerialPort*. Pro implementaci komunikace byla vytvořena třída *RfidReader*, která řeší veškerou komunikaci s RFID čtečkou.

Použití této třídy je velmi jednoduché. Po zavolání konstruktoru, kterému je zapotřebí předat název sériového portu, k němuž je čtečka připojena (v tomto případě `ttyS1`), dojde k vytvoření instance třídy *QSerialPort* s názvem `serialPort`.

Na vytvořeném objektu je následně zavolána metoda `setPortName()`, kterou je nastaveno použité zařízení. Pokud se podaří toto zařízení úspěšně otevřít pro čtení i zápis, dojde k nastavení parametrů sériového přenosu, nastavené hodnoty jsou uvedeny v tabulce 6.1.

modulační rychlost	115200 baudů
počet datových bitů	8
počet stop bitů	1
parita	žádná
řízení toku	nepoužito

Tab. 6.1: Konfigurace rozhraní UART.

Následně je vytvořena instance třídy *QTimer*, která slouží k periodickému spouštění funkce `pollTag()`, která se dotazuje čtečky na přítomnost RFID tagu. Perioda, s jakou bude dotaz prováděn, je dána konstantou `POLLING_MSEC`, která je ve výchozím hodnotu nastavena na 100, což odpovídá periodě 100 ms.

Poslední funkcí konstruktoru je propojení signálu<sup>(1)</sup> `readyRead()` generovaného objektem `serialPort` se slotem `readData()`. Tím je zajištěno, že pokaždé, když rozhraní UART přijme jakákoliv data a budou připravena k vyčtení z vyrovnávací paměti (bufferu), dojde k zavolání metody `readData()`, která přijatá data z bufferu přečte a zpracuje.

### Metoda `readData()`

Metoda `readData()` implementuje část komunikačního protokolu RFID čtečky StrongLink SL031 tak, jak je definován v uživatelském návodu [5]. Jelikož není pro potřeby autentizace uživatele předmětem nutné více, než přečtení ID z přiloženého RFID tagu, nebyl komunikační protokol implementován v celém rozsahu.

Je-li tato metoda zavolána, provede přečtení dat z UART rozhraní pomocí metody `readAll()`, která vrací objekt typu *QByteArray* reprezentující pole bajtů. Poté následuje zpracování dat, které lze stručně popsat takto:

1. kontrola preamble (hodnota prvního bajtu musí být 189),
2. přečtení délky zprávy z 2. bajtu,
3. výpočet kontrolního součtu funkcí XOR,
4. vyčtení ID dle typu RFID tagu.

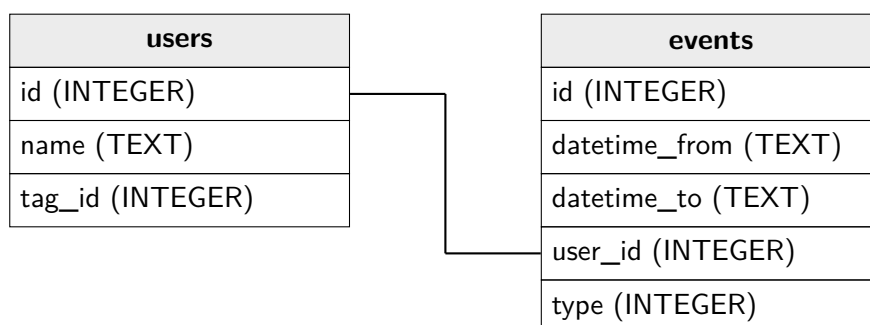
<sup>(1)</sup>Signály a sloty jsou specifickým prvkem knihovny Qt, sloužícím k realizaci vnitřní komunikace, který umožňuje velmi snadno specifikovat, jak má program reagovat na nějakou událost. Pro podrobnější informace viz dokumentaci knihovny Qt [33].

Pokud vše proběhne v pořádku (tj. data nebyla při přenosu jinak poškozena), dojde k vyslání signálu `tagDetected()`, ve kterém je předán typ detekovaného RFID tagu a jeho unikátní ID.

Metoda umožňuje, a ve výchozím nastavení také provádí, filtraci opakovaných zpráv. V takovém případě je signál vyslán pouze tehdy, je-li detekován nový tag. Pokud je tag ponechán v dosahu RFID čtečky delší dobu, bude s ním čtečka sice komunikovat opakovaně, ale nedojde k opakování zmíněného signálu. Další signál bude vyslán až poté, co se tag dostane z dosahu čtečky a zpět, případně pokud se v dosahu objeví jiný tag.

## 6.2.2 Databáze

Problematika uložení dat o uživateli a jejich docházce byla vyřešena použitím relační databáze typu SQLite, která je nenáročná na systémové prostředky.



Obr. 6.1: Relační schéma databáze.

Databáze, jejíž relační schéma je zobrazeno na obr. 6.1, je složena ze dvou tabulek. Jak napovídá její název, tabulka *users* obsahuje informace o uživateli systému (zaměstnancích), zatímco tabulka *events* ukládá data o veškerých docházkových událostech (příchod, odchod, pauza, ...).

Z programového hlediska je podstatná třída *DataStorage*, která zajišťuje veškeré operace prováděné nad databází.

### Třída *DataStorage*

Třída *DataStorage* implementuje veškeré databázové operace potřebné pro účely docházkové aplikace. Nemá význam zde popisovat způsob řešení jednotlivých metod, ten je možné nalézt ve zdrojových kódech obsažených na příloženém CD.



Uvedme alespoň seznam dostupných metod a stručný popis toho, co realizují:

- `getUserName()` – vrací jméno uživatele pro zadané ID,
- `getId()` – vrací ID uživatele pro zadané ID RFID tagu,
- `getUserList()` – vrací pole obsahující ID všech uživatelů,
- `openEvent()` – uloží začátek události (např. příchod, začátek pauzy),
- `closeEvent()` – ukončí otevřenou událost a uloží informace o tom kdy skončila,
- `getUserState()` – vrací aktuální stav uživatele (nepřítomen, přítomen, ...),
- `getEvents()` – vrací seznam událostí pro zadané datum,
- `saveExpected()` – uloží předpoklad docházky,
- `deleteExpected()` – smaže záznam o předpokladu docházky.

Jednotlivé funkce jsou volány z uživatelského rozhraní, které implementuje většinu funkční logiky terminálu. Popisu jeho řešení se věnuje následující část textu.

## 6.3 Uživatelské rozhraní

Návrh grafického uživatelského rozhraní (GUI) byl ovlivněn především použitím dotykového displeje jako jediného vstupního zařízení. Z toho důvodu jsou všechny prvky uzpůsobeny tak, aby nebyl problém s nimi tímto způsobem pracovat.

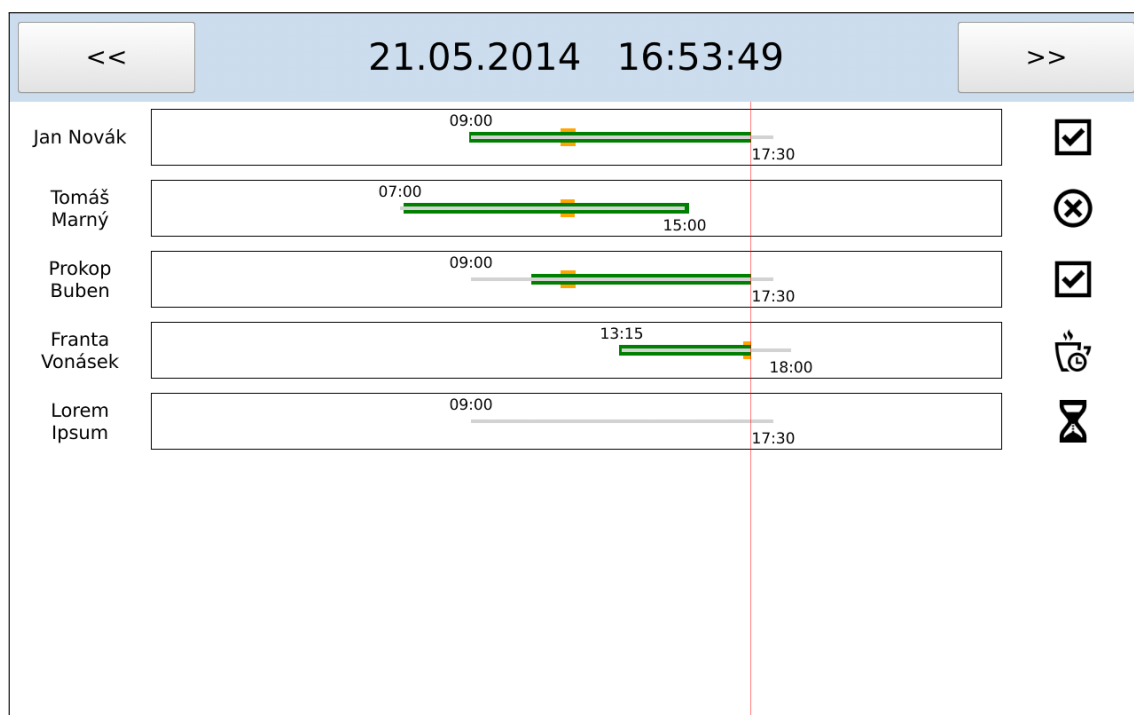
### 6.3.1 Hlavní obrazovka

V klidovém stavu aplikace zobrazuje hlavní obrazovku, na které jsou uvedeny aktuální informace o přítomnosti všech zaměstnanců na pracovišti tak, jak je patrné z obr. 6.2. Zobrazení je uskutečněno pomocí jednoduchého seznamu, ve kterém jsou pod sebou uvedena jména uživatelů, časová osa reprezentující jejich docházku za daný den a grafická ikona indikující jejich stav.

#### Časová osa

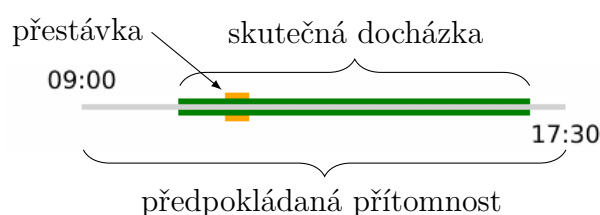
Ústředním prvkem zobrazení jsou časové osy náležící jednotlivým uživatelům, ze kterých je patrný jejich stav v průběhu celého dne. Provedení časové osy je sice poměrně dobře patrné ze snímku hlavní obrazovky, pro vysvětlení se ale bude hodit její detail (obr. 6.3).

Časová osa vždy reprezentuje úsek 24 hodin za konkrétní den. Ačkoliv se může zdát zobrazení celého dne jako plýtvání místem, které by šlo využít i jinak, není tomu tak. Díky značnému rozptylu docházky zaměstnanců se poměrně často stává, že zatímco někdo je na pracovišti již před šestou hodinou ránní, někdo jiný se spíše zdrží dlouho do noci. Autor během vývoje usoudil, že je vhodnější nechat na časové ose celý den, než odstranit úsek například pouze 6 hodin.



Obr. 6.2: Hlavní obrazovka docházkové aplikace.

Jak je z obrázků vidět, doba odpovídající předpokládané docházce je zobrazena jako čára šedé barvy. Zelenou barvou je zvýrazněn interval odpovídající přítomnosti zaměstnance na pracovišti, oranžová barva náleží přestávkám. Předpoklad docházky je navíc doplněn textovou informací o pravděpodobném příchodu a odchodu zaměstnance – terminál tak poskytuje okamžitou odpověď na otázku „kdy přijde zaměstnanec  $x$ “.








Obr. 6.3: Detail časové osy.

Různá tloušťka čar reprezentujících jednotlivé události není bezvýznamná. Umožňuje použití aplikace i lidem s poruchami barvocitu, kteří by mohli mít problémy s rozlišením jednotlivých barev.

## Stavové ikony

Vedle časové osy jsou umístěny stavové ikony, dle kterých lze na první pohled určit, zda je zaměstnanec na pracovišti přítomen apod. Seznam všech ikon a jejich význam je uveden v tabulce 6.2.

	nepřítomen
	přítomen
	opožděn (měl by být přítomen, ale není)
	zaměstnanec má přestávku
	předpokládaná docházka v daný den

Tab. 6.2: Stavové ikony a jejich význam.

## Ovládací panel

Hlavní obrazovka umožňuje zobrazit nejen současný stav (dnešní den), ale také předchozí dny s historií docházky, respektive předpoklad docházky na následující dny. K tomu slouží nabídka umístěná na horní straně obrazovky.

Nabídka se skládá ze tří prvků, konkrétně levého tlačítka, pravého tlačítka a středové části určené pro text. Levé tlačítko slouží k přepnutí na předcházející den, pravé umožňuje přepnout zobrazení na následující den. Pro rychlý návrat na zobrazení dnešního dne je možné stisknout středový popisek nabídky, který obsahuje informace o zobrazeném dni (datum, případně aktuální čas).

### 6.3.2 Uživatelská obrazovka

Pokud RFID čtečka detekuje přiložení identifikačního tagu, dojde k jeho ověření vůči databázi. Pokud je v databázi nalezen uživatel, který má tag s tímto ID přiřazen, dojde k přepnutí z hlavní obrazovky na obrazovku uživatelskou.

Uživatelská nabídka je realizována v podobném stylu jako hlavní obrazovka. Horní ovládací panel je v tomto režimu doplněn spodním ovládacím panelem, ve kterém jsou umístěna tlačítka reprezentující jednotlivé uživatelské akce.

Jak je patrné z obr. 6.4, jde o tlačítka umožňující signalizovat příchod na pracoviště, odchod z pracoviště a začátek, respektive konec pauzy. Pokud uživatel dané tlačítko stiskne, provede se zapsání informace o této události do databáze a následně návrat

historie

Jan Novák

předpoklad

denní statistika

☑

08:57 — 16:56

7,99 h

🕒

11:32 — 11:55

(-0,38 h)

celkem odpracováno 7,61 h

příchod

pauza

odchod

Obr. 6.4: Uživatelská obrazovka docházkové aplikace.

na hlavní obrazovku. Díky tomu ze strany uživatele stačí „pípnout čipem“ a zvolit požadovanou akci, takže není nutné se u terminálu dlouho zdržovat.

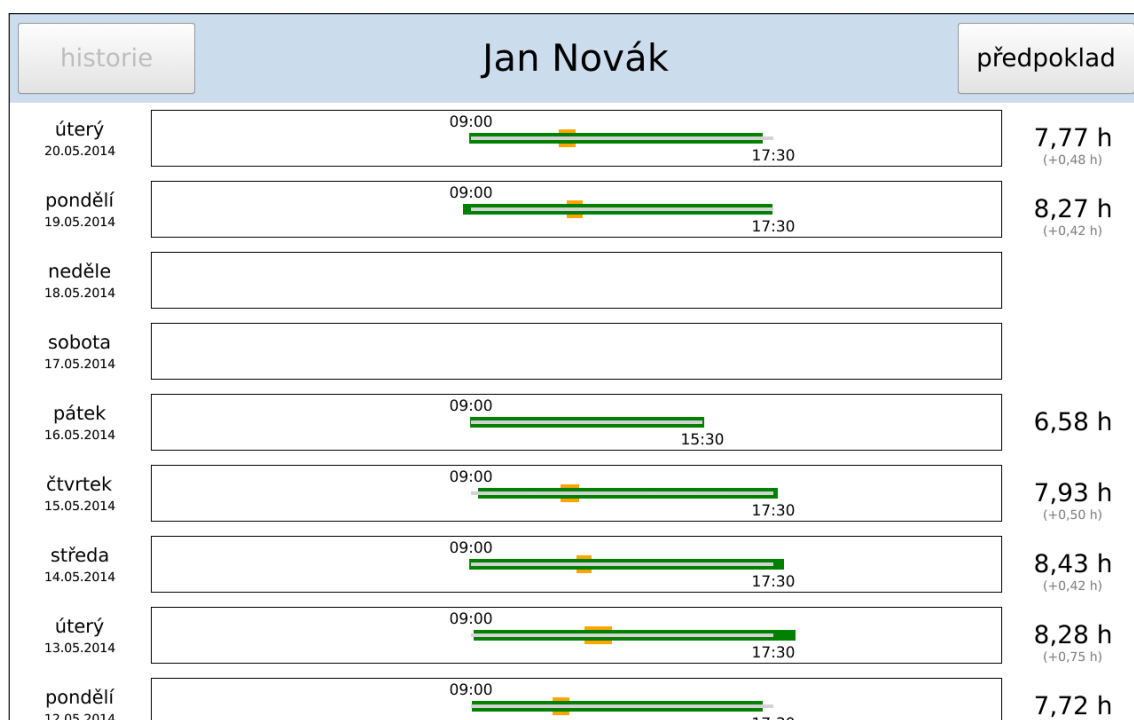
Středová část této obrazovky je věnována souhrnu informací za aktuální den, obsahuje tedy intervaly všech událostí daného dne. Také je možné zde zjistit, kolik hodin daný zaměstnanec za den již odpracoval.

### Historie docházky

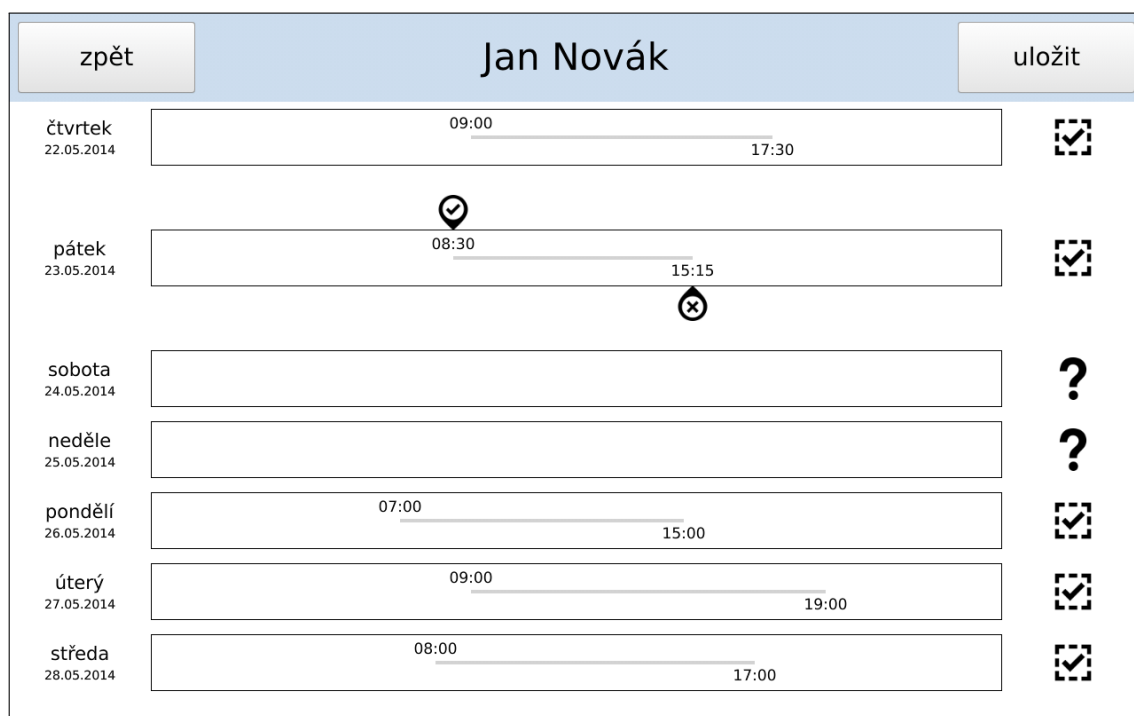
Z uživatelské obrazovky je možné přejít na zobrazení historie docházky za posledních 30 dní. K přepnutí do tohoto režimu slouží tlačítko „historie“ vlevo na horním panelu. Historie je zobrazena velmi podobně jako přehled na hlavní obrazovce. Pro každý den je zobrazena časová osa, na které je vyznačena docházka. Protože stavové ikony nemají v tomto režimu žádný smysl, je na jejich místě uvedena balance odpracovaných hodin.

### Nastavení předpokládané docházky

K nastavení předpokládané docházky je možné přejít tlačítkem „předpoklad“ umístěným v pravém horním rohu obrazovky. Jak je vidět na obr. 6.6, je možné nastavit předpoklad docházky na následujících 7 dní. Pro nastavení daného dne stačí kliknout na ikonu umístěnou vpravo od časové osy. Tím dojde k přepnutí do editačního



Obr. 6.5: Přehled historie docházky.



Obr. 6.6: Nastavení předpokládané docházky.

režimu, ve kterém je možné pomocí posuvných značek nastavit (s rozlišením 15 minut) předpokládaný okamžik příchodu a odchodu.

Pomocí tlačítek v horním panelu je možné opustit tuto nabídku bez uložení změn, nebo jejich uložení potvrdit. Časové údaje lze kdykoliv opětovně změnit, stejně tak lze předpoklad na daný den zcela odstranit.

## 6.4 Webové rozhraní

Jelikož navržený terminál disponuje rozhraním ethernet pro připojení do počítačové sítě, rozhodl se autor administraci docházkového systému realizovat prostřednictvím webového rozhraní, jenž umožňuje následující funkce:

- správu uživatelů (přidání/odebrání, změnu ID RFID tagu),
- zobrazení aktuálního stavu (kdo je přítomen, čas příchodu, ...),
- zpětnou úpravu existujících záznamů,
- zobrazení historie docházky za konkrétní období (rok, měsíc či den).

Pro provoz webového rozhraní je na terminálu spuštěn webový server *lighttpd* (verze 1.4.33), doplněný o interpret jazyka *PHP* ve verzi 5.3.27. Samotné webové rozhraní je vytvořeno v programovacím jazyce PHP, doplněném na straně klienta o skripty v jazyce JavaScript vytvořené s využitím knihovny *jQuery*.

Pro zobrazení tohoto rozhraní stačí libovolným webovým prohlížečem přistoupit na adresu ve formátu `http://<IP_adresa_terminálu>/`.

## 7 ZÁVĚR

Cílem této diplomové práce byl popis problematiky elektronických docházkových systémů (EDS) a návrh možného provedení docházkového terminálu.

V úvodní části práce se autor stručně věnuje historii docházkových systémů, které lidstvu pomáhají s evidencí docházky pracovníků již více než 120 let. Za dobu své existence prošly výrazným vývojem, v rámci kterého je jednoznačně nejdůležitější změnou přechod od mechanického řešení k elektronickému, jenž byl umožněn rozvojem výpočetní techniky.

Po uvedení možností EDS je představen koncept terminálu postaveného kolem minipočítače Cubieboard2, doplněného dotykovým LCD displejem a RFID čtečkou. K tomu, aby bylo možné popsaný terminál sestavit, bylo nutné navrhnout základní desku řešící elektrické propojení jednotlivých komponent. Obvodové řešení této desky je popsáno v kapitole druhé společně s návrhem desky plošných spojů (DPS).

Představený návrh DPS byl prakticky ověřen vyrobením prototypu, s pomocí něhož bylo v návrhu nalezeno několik drobných chyb. Výsledky testování a popis zmíněných chyb jsou popsány v kapitole třetí. Uvedené chyby se na prototypu podařilo snadno odstranit a navržená základní deska je tak plně funkční.

Kapitola čtvrtá je věnována problematice softwarové podpory platformy Allwinner A20, na níž je minipočítač Cubieboard2 postaven, v prostředí operačního systému GNU/Linux. Pozornost je upřena především na vhodnou větev linuxového jádra (kernel) a zavaděč (bootloader), které jsou vyvíjeny v rámci projektu linux-sunxi.

Následující kapitola je věnována nástroji Buildroot, pomocí jehož byl vytvořen binární obraz operačního systému pro uvedený minipočítač. Jsou představeny výhody použití tohoto nástroje a především úpravy, které je zapotřebí provést k praktickému nasazení na uvedené platformě. Výstupem uvedeného popisu je minimalistický obraz OS GNU/Linux nakonfigurovaný specificky pro použití v popsaném docházkovém terminálu.

V závěrečné kapitole je představena docházková aplikace, která byla pro popsaný terminál vytvořena. Je nastíněn navržený koncept, na který navazují detaily o jeho implementaci v prostředí jazyka C++ za použití knihovny Qt a relační databáze SQLite. Text doplňují ukázkové snímky hotové aplikace, na nichž je popsán význam jednotlivých prvků grafického uživatelského rozhraní (GUI) a možnosti uživatelské interakce.

Na příloženém CD jsou obsaženy všechny výstupy práce, tedy návrh DPS základní desky, prostředí nástroje Buildroot pro minipočítač Cubieboard2 a zdrojové kódy vyvinuté docházkové aplikace. Příložena je také fotodokumentace vyrobeného prototypu terminálu.

# LITERATURA

- 1 ALLWINNER TECHNOLOGY. *A10-cubieboard-2012-08-08* [online]. 2012 [cit. 2013-12-28]. Elektrotechnické schéma. Dostupný z WWW: [http://dl.cubieboard.org/hardware/cubieboard\\_schematic\\_2012-08-08.pdf](http://dl.cubieboard.org/hardware/cubieboard_schematic_2012-08-08.pdf).
- 2 ALLWINNER TECHNOLOGY. *A20* [online]. 2013 [cit. 2013-12-27]. Katalogový list. Dostupný z WWW: [https://github.com/OLIMEX/OLINUXINO/raw/de2a3019c7f4ee959553f8d718309132dd978c32/HARDWARE/A20 - PDFs / A20 %20Datasheet%20v1.0%2020130227.pdf](https://github.com/OLIMEX/OLINUXINO/raw/de2a3019c7f4ee959553f8d718309132dd978c32/HARDWARE/A20-PDFs/A20%20Datasheet%20v1.0%2020130227.pdf).
- 3 ARCH LINUX ARM. *Cubieboard 2* [online]. [cit. 2014-05-13]. Dostupný z WWW: <http://archlinuxarm.org/platforms/armv7/allwinner/cubieboard-2>.
- 4 ASWAD, Ed; MEREDITH, Suzanne. *IBM in Endicott*. Mount Pleasant (South Carolina, USA) : Arcadia Publishing, 2005. 128 s. Dostupný také z WWW: <http://books.google.cz/books?id=Yz1DdhWK3IsC>. ISBN 9780738537009.
- 5 BEIJING STRONGLINK TECHNOLOGY. *Mifare Reader / Writer SL031 User Manual* [online]. [cit. 2014-05-17]. Dostupný z WWW: <http://www.stronglink-rfid.com/download/SL031-User-Manual.pdf>.
- 6 BEIJING STRONGLINK TECHNOLOGY. *Small RFID Reader/Writer SL031* [online]. 2014 [cit. 2013-12-27]. Dostupný z WWW: <http://www.stronglink-rfid.com/en/rfid-modules/sl031.html>.
- 7 BENDÍK, Miroslav. *OLinuxino A13 WIFI - výkonnejší konkurent Raspberry Pi* [online]. [cit. 2014-05-13]. Dostupný z WWW: <http://linuxos.sk/clanok/olinuxino-a13-wifi-vykonnejsi-konkurent/>.
- 8 BERNER, Jeff. *The Joy of Working from Home: Making a Life While Making a Living*. San Francisco (California, USA) : Berrett-Koehler Publishers, 1994. 163 s. Berrett-Koehler Series. Dostupný také z WWW: [http://books.google.cz/books?id=Xu\\_\\_e7mtM8sC](http://books.google.cz/books?id=Xu__e7mtM8sC). ISBN 9781881052463.
- 9 BUILDROOT. *The Buildroot user manual* [online]. [cit. 2014-05-13]. Dostupný z WWW: <http://buildroot.uclibc.org/downloads/manual/manual.html>.
- 10 BURDA, Karel; STRAŠIL, Ivo. *Zabezpečovací systémy*. Brno : Vysoké učení technické v Brně, 2011. 187 s. Skriptum. ISBN 9788021444416.
- 11 CADSOFT. *Download - Get the latest Version of EAGLE PCB Design Software* [online]. 2011 [cit. 2013-12-29]. Dostupný z WWW: <http://www.cadsoftusa.com/download-eagle/>.



- 12 CLIFF ELECTRONIC COMPONENTS. *SMD DC Sockets* [online]. 2011 [cit. 2013-12-27]. Katalogový list. Dostupný z WWW: <http://www.cliffuk.co.uk/products/dcconnectors/SmdDcSockets.pdf>.
- 13 CUBIEBOARD.ORG. *products:start [Cubieboard Docs]* [online]. 2013 [cit. 2013-12-27]. Dostupný z WWW: <http://docs.cubieboard.org/products/start#cubieboard2>.
- 14 DENX SOFTWARE ENGINEERING. *Das U-Boot – the Universal Boot Loader* [online]. [cit. 2014-05-13]. Dostupný z WWW: <http://www.denx.de/wiki/U-Boot>.
- 15 ELINUX.ORG. *RPi Hardware* [online]. 2013 [cit. 2013-12-27]. Dostupný z WWW: [http://elinux.org/RPi\\_Hardware](http://elinux.org/RPi_Hardware).
- 16 FEDORA PROJECT. *Cubie Board* [online]. [cit. 2014-05-13]. Dostupný z WWW: [https://fedoraproject.org/wiki/Cubie\\_Board](https://fedoraproject.org/wiki/Cubie_Board).
- 17 FURNELL, Steven. *Securing Information and Communications Systems: Principles, Technologies, and Applications*. Boston (Massachusetts, USA) : Artech House, 2007. 362 s. Artech House computer security series. Dostupný také z WWW: <http://books.google.cz/books?id=VAUtQ9MjcLUC>. ISBN 9781596932296.
- 18 INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. *Amendment: Data Terminal Equipment (DTE) Power via Media Dependent Interface (MDI)*. New York (New York, USA) : Institute of Electrical and Electronics Engineers, 1995. IEEE Std 802.3af-2003. Norma. ISBN 0738136964.
- 19 INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. *Media Access Control (MAC) Parameters, Physical Layer, Medium Attachment Units, and Repeater for 100 Mb/s Operation, Type 100BASE-T (Clauses 21–30)*. New York (New York, USA) : Institute of Electrical and Electronics Engineers, 1995. IEEE Std 802.3u-1995. Norma. ISBN 1559375426.
- 20 INTERNATIONAL STANDARDIZATION ORGANIZATION. *Identification cards – Contactless integrated circuit(s) cards – Proximity cards*. Geneva (CH) : International Standardization Organization, 2000. ISO 14443 A. Norma.
- 21 LG DISPLAY. *LP101WX1 – Liquid Crystal Display* [online]. 2011 [cit. 2013-12-27]. Katalogový list. Dostupný z WWW: <http://english.lookpanel.com/pdf/LP101WX1-SLB1.pdf>.
- 22 LINARO PROJECT. *Ubuntu based Rootfs provided by Linaro* [online]. 2014 [cit. 2014-05-13]. Dostupný z WWW: <https://wiki.linaro.org/Platform/DevPlatform/Rootfs>.

- 23 LINUX-SUNXI. *linux-sunxi.org* [online]. 2014 [cit. 2014-05-12]. Dostupný z WWW: [⟨http://linux-sunxi.org/Main\\_Page⟩](http://linux-sunxi.org/Main_Page).
- 24 LINUX-SUNXI. *u-boot-sunxi Wiki* [online]. [cit. 2014-05-13]. Dostupný z WWW: [⟨https://github.com/linux-sunxi/u-boot-sunxi/wiki⟩](https://github.com/linux-sunxi/u-boot-sunxi/wiki).
- 25 MATERNA, Zdeněk. *Polohovatelný stojan pro přehledovou kameru*. 2011. 87 s. Diplomová práce. FEKT VUT. Vedoucí práce Pavel Kučera. Dostupný také z WWW: [⟨http://hdl.handle.net/11012/8190⟩](http://hdl.handle.net/11012/8190).
- 26 MONOLITHIC POWER SYSTEMS. *MP1482 – 2A, 18V Synchronous Rectified Step-Down Converter* [online]. 2012 [cit. 2013-12-27]. Katalogový list. Dostupný z WWW: [⟨http://www.monolithicpower.com/Page/Download.aspx?ListID=1a4312a6-2709-4b7d-af52-9f1cedbffd415&&ItemID=97⟩](http://www.monolithicpower.com/Page/Download.aspx?ListID=1a4312a6-2709-4b7d-af52-9f1cedbffd415&&ItemID=97).
- 27 NEWMAN, Robert. *Biometrics: Application, Technology, and Management*. Stamford (Connecticut, USA) : Course Technology, 2009. 512 s. Dostupný také z WWW: [⟨http://books.google.cz/books?id=4VbhZcTAccoC⟩](http://books.google.cz/books?id=4VbhZcTAccoC). ISBN 9781435441057.
- 28 NINIGI. *NX Plugs, Cable Mount, 2.0 mm Pitch* [online]. 2005 [ cit. 2013-12-27]. Katalogový list. Dostupný z WWW: [⟨http://www.ninigi.com/usrFiles/docs/nxg-02.pdf⟩](http://www.ninigi.com/usrFiles/docs/nxg-02.pdf).
- 29 NINIGI. *NX Sockets, PCB Mount, 2.0 mm Pitch* [online]. 2005 [ cit. 2013-12-27]. Katalogový list. Dostupný z WWW: [⟨http://www.ninigi.com/usrFiles/docs/nxw-02.pdf⟩](http://www.ninigi.com/usrFiles/docs/nxw-02.pdf).
- 30 PEARSON, Robert. *Electronic Security Systems: A Manager's Guide to Evaluating and Selecting System Solutions*. Amsterdam (NL) : Butterworth-Heinemann, 2006. 384 s. Dostupný také z WWW: [⟨http://books.google.cz/books?id=pnR-1R0nikgC⟩](http://books.google.cz/books?id=pnR-1R0nikgC). ISBN 9780080494708.
- 31 PETAZZONI, Thomas. *Using Buildroot for A Real Project* [online]. 2011 [cit. 2014-05-20]. Videozáznam z konference. Dostupný z WWW: [⟨https://www.youtube.com/watch?v=tkK3oqdi45Q⟩](https://www.youtube.com/watch?v=tkK3oqdi45Q).
- 32 PUGH, Emerson L. *Building IBM: Shaping an Industry and Its Technology*. Cambridge (Massachusetts, USA) : MIT Press, 1995. 405 s. Dostupný také z WWW: [⟨http://books.google.cz/books?id=Bc8BGhS0awgC⟩](http://books.google.cz/books?id=Bc8BGhS0awgC). ISBN 9780262161473.
- 33 QT PROJECT. *Signals & Slots* [online]. 2013 [cit. 2014-05-17]. Dostupný z WWW: [⟨http://qt-project.org/doc/qt-5/signalsandslots.html⟩](http://qt-project.org/doc/qt-5/signalsandslots.html).
- 34 QT PROJECT. *Supported Platforms* [online]. 2013 [cit. 2014-05-13]. Dostupný z WWW: [⟨http://qt-project.org/doc/qt-5/supported-platforms.html⟩](http://qt-project.org/doc/qt-5/supported-platforms.html).

- 35 SCHINAGL, Olliver M. *ARM Allwinner sunxi SoCs and the community behind it* [online]. 2014 [cit. 2014-05-12]. Videozáznam z konference. Dostupný z WWW: [⟨https://www.youtube.com/watch?v=un-BJTISCdk⟩](https://www.youtube.com/watch?v=un-BJTISCdk).
- 36 STMICROELECTRONICS. *LFxxAB/LFxxC – Very low drop voltage regulators with inhibit* [online]. 2012 [cit. 2013-12-27]. Katalogový list. Dostupný z WWW: [⟨http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00000546.pdf⟩](http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00000546.pdf).
- 37 TEXAS INSTRUMENTS. *TPA2012D2 – 2.1 W/Ch Stereo Filter-Free Class-D Audio Power Amplifier* [online]. 2008 [cit. 2013-12-27]. Katalogový list. Dostupný z WWW: [⟨http://www.ti.com/lit/ds/symlink/tpa2012d2.pdf⟩](http://www.ti.com/lit/ds/symlink/tpa2012d2.pdf).
- 38 VARTA. *V 391 MF – Mercury-free Primary Silver* [online]. 2009 [cit. 2013-12-28]. Katalogový list. Dostupný z WWW: [⟨http://www.tme.eu/cz/Document/9162599a67e341484fe8b14f772b250a/V391.pdf⟩](http://www.tme.eu/cz/Document/9162599a67e341484fe8b14f772b250a/V391.pdf).
- 39 WANT, Roy. *RFID Explained: A Primer on Radio Frequency Identification Technologies*. San Rafael (California, USA) : Morgan & Claypool Publishers, 2006. 83 s. Synthesis lectures on mobile and pervasive computing. Dostupný také z WWW: [⟨http://books.google.cz/books?id=sNnABdDhs\\_8C⟩](http://books.google.cz/books?id=sNnABdDhs_8C). ISBN 9781598291087.

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

CPU	Central Processing Unit – výpočetní jádro
DDR	Double Data Rate – typ SDRAM paměti přenášející data při obou hranách (náběžné i sestupné)
DPS	deska plošných spojů
EDS	elektronický docházkový systém
GPIO	General Purpose Input/Output – vstupně/výstupní rozhraní pro obecné použití
GPU	Graphical Processing Unit – grafické výpočetní jádro
HTPC	Home Theater Personal Computer – počítač určený k přehrávání multimediálního obsahu
I <sup>2</sup> C	Inter-Integrated Circuit – sériová sběrnice pro propojení integrovaných obvodů
I <sup>2</sup> S	varianta I <sup>2</sup> C sběrnice pro přenos zvuku
IEC	International Electrotechnical Commission – Mezinárodní elektrotechnická komise
IEEE	Institute of Electrical and Electronics Engineers – Institut pro elektrotechnické a elektronické inženýrství
HDMI	High-Definition Multimedia Interface – rozhraní určené pro přenos obrazového a zvukového signálu ve vysokém rozlišení
I/O	Input/Output – vstupně/výstupní
IO	intergovaný obvod
IP	Internet Protocol – protokol síťové vrstvy, jenž je zcela zásadní pro provoz sítě Internet
IR	infrared – infračervený
ISO	International Organization for Standardization – Mezinárodní organizace pro normalizaci
LAN	Local Area Network – místní počítačová síť (dnes nejčastěji na bázi technologie Fast Ethernet)

LCD	Liquid Crystal Display – displej z tekutých krystalů
LED	Light-Emitting Diode – dioda emitující světlo
LVDS	Low Voltage Differential Signaling
MCU	Microcontroller Unit – mikrokontrolér (jednočipový počítač)
MMC	MultiMediaCard – standard paměťových karet, předchůdce standardu Secure Digital
NAND	hradlo provádějící negovaný logický součin, bývá použito pro tvorbu paměťových buněk ve flash pamětech
OS	operační systém
PC	Personal Computer – osobní počítač
PIN	Personal Identification Number – číselný kód používaný pro ověření identity uživatele platebních karet
PoE	Power over Ethernet – napájení přes ethernet
RAM	Random Access Memory – paměť s přímým přístupem – druh paměti u které je libovolná oblast dostupná za stejnou dobu
RFID	Radio Frequency Identification – identifikace na rádiové frekvenci
RTC	Real-Time Clock – hodiny reálného času
SATA	Serial ATA – vysokorychlostní sběrnice pro připojení datových úložišť (především pevných disků)
SBC	Single Board Computer – jednodeskový počítač
SD	Secure Digital – standard paměťových karet, v dnešní době nejčastěji používaný
SDRAM	Synchronous Dynamic Random Access Memory – typ RAM paměti používaný v dnešních počítačích
SoC	System on Chip – systém na čipu
SPI	Serial Peripheral Interface – sériové periferní rozhraní
UI	User Interface – uživatelské rozhraní

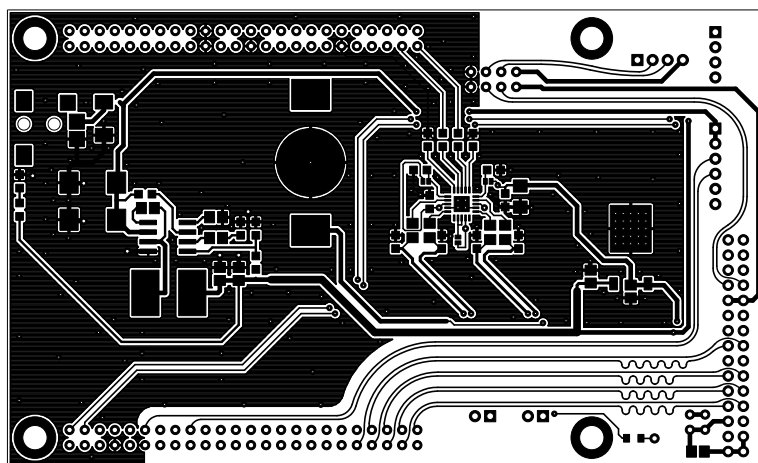
THT	Through-Hole Technology – technologie součástek s drátovými vývody, které procházejí skrz DPS
TTL	Transistor-Transistor Logic – tranzistorově-tranzistorová logika
UART	Universal Asynchronous Receiver/Transmitter – rozhraní pro asynchronní sériovou komunikaci (např. pro RS232 apod.), implementováno ve většině mikrokontrolérů
USB	Universal Serial Bus – univerzální sériová sběrnice
VGA	Video Graphics Array – analogový standard pro připojení zobrazovacích zařízení
Wi-Fi	obecné označení bezdrátových sítí dle standardu 802.11

# SEZNAM PŘÍLOH

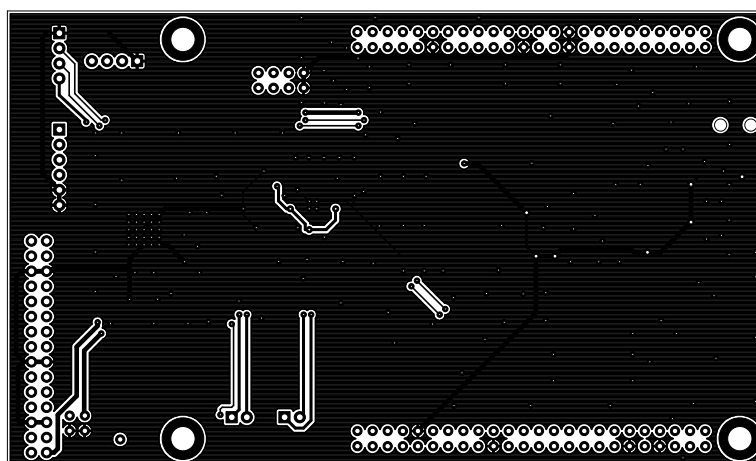
<b>A</b>	<b>Výkresová dokumentace základní desky</b>	<b>55</b>
<b>B</b>	<b>Fotografie prototypu</b>	<b>59</b>
<b>C</b>	<b>Buildroot – konfigurace</b>	<b>60</b>
C.1	Konfigurace prostředí (defconfig) . . . . .	60
C.2	Post-build skript . . . . .	61
<b>D</b>	<b>Obsah přiloženého CD</b>	<b>62</b>

## A VÝKRESOVÁ DOKUMENTACE ZÁKLADNÍ DESKY

Návrh DPS byl proveden v návrhovém systému EAGLE 6.5.0 od firmy CadSoft jehož variantu *Light Edition* je možné získat zdarma [11]. Uvedený návrh DPS nepřekračuje maximální rozměr DPS daný licenčním omezením ( $100 \times 80$  mm), díky čemuž jej lze plně upravovat i v této volně dostupné verzi.

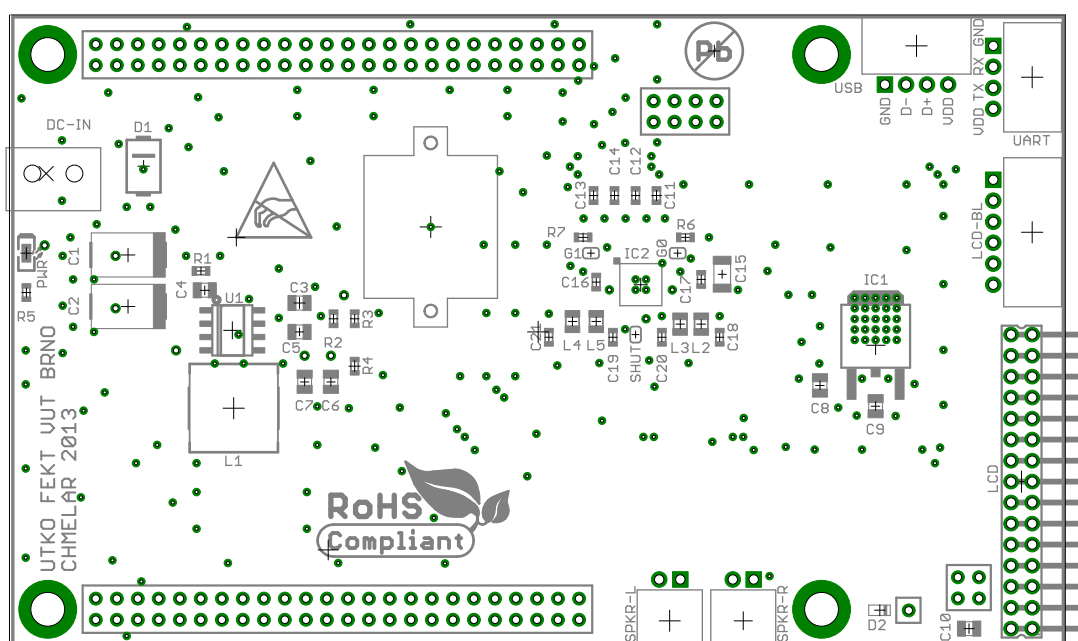


Obr. A.1: Základní deska – strana součástek (TOP), měřítko 1:1.

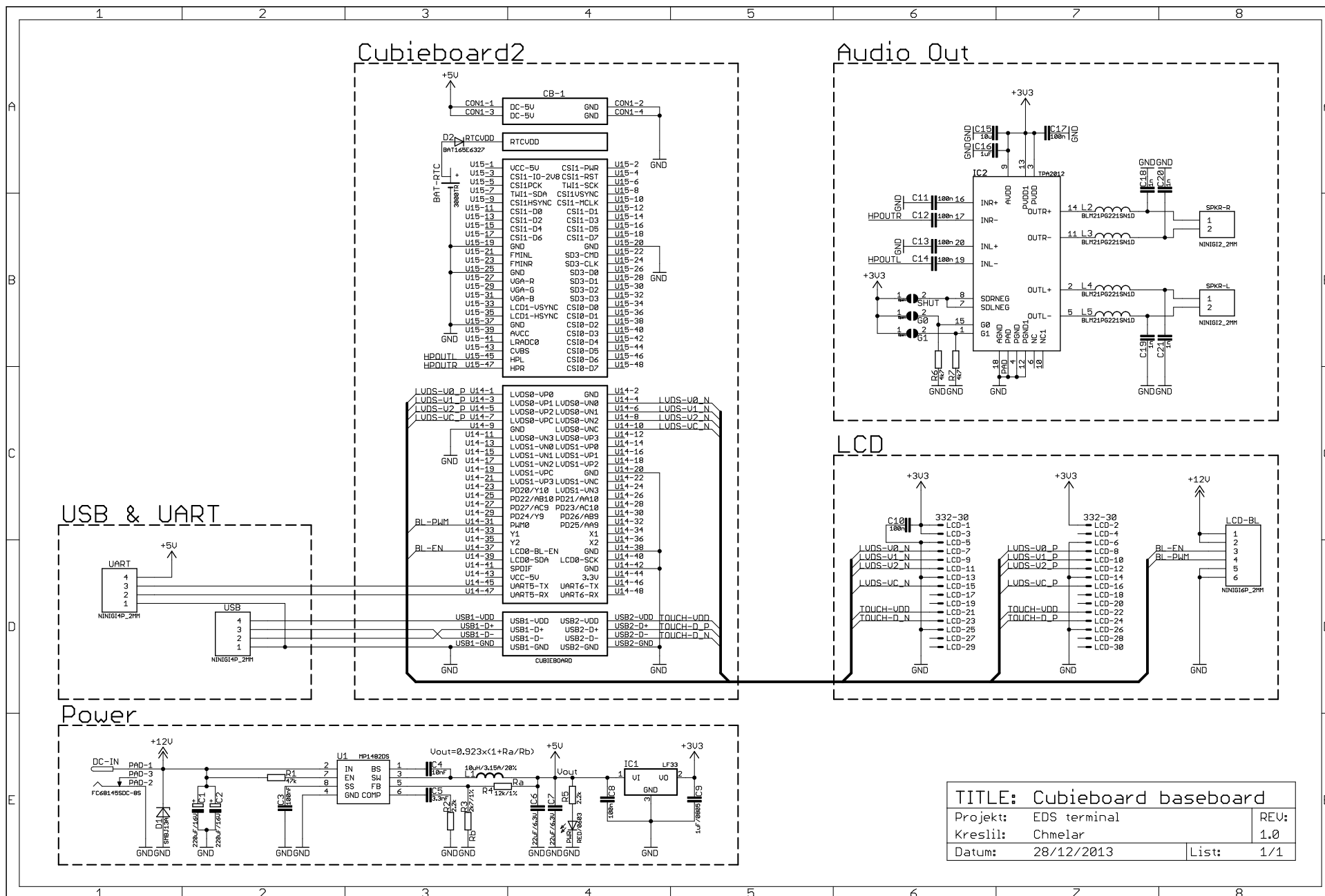


Obr. A.2: Základní deska – strana spojů (BOT), měřítko 1:1.





Obr. A.3: Základní deska – osazovací plán (všechny součástky na straně TOP).



Obr. A.4: Základní deska – schéma zapojení.

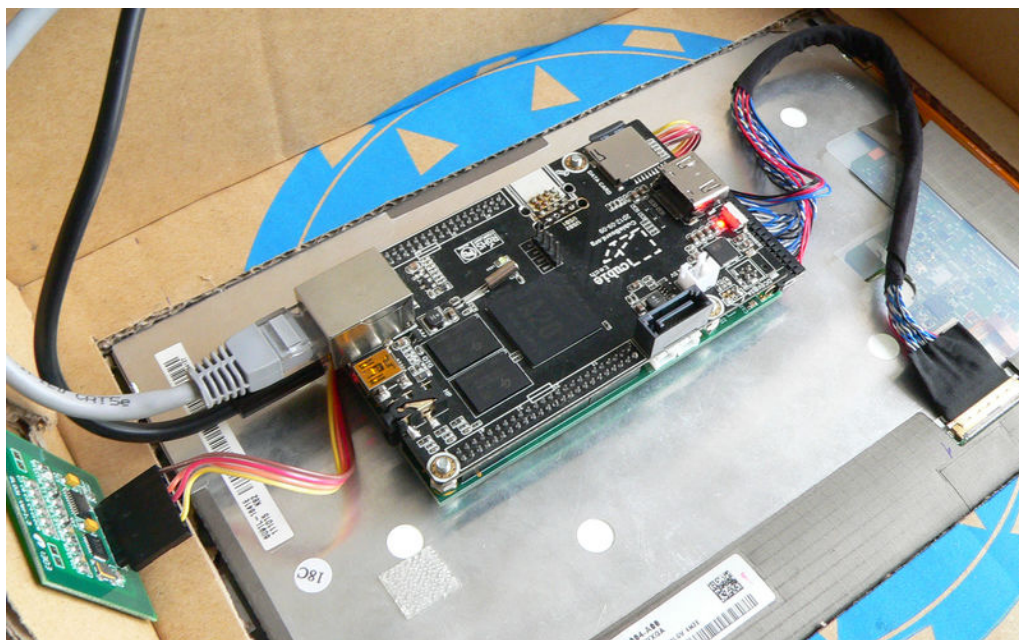
pouzdro	hodnota	počet	označení	poznámka
0603	2,7 k $\Omega$ /1 %	1	R3	
0603	1 nF	4	C18, C19, C20, C21	
0603	1 $\mu$ F	1	C16	
0805	1 $\mu$ F	1	C9	
0603	2,2 k $\Omega$	2	R2, R5	
0805	3,3 nF	1	C5	
0603	12 k $\Omega$ /1 %	1	R4	
0603	4,7 k $\Omega$	2	R6, R7	
0805	10 nF	1	C4	
1206	10 $\mu$ F	1	C15	
–	10 $\mu$ H/3,15 A/20 %	1	L1	pouzdro 8×8 mm
0805	22 $\mu$ F/6,3 V	2	C6, C7	
0603	47 k $\Omega$	1	R1	
0603	100 nF	5	C11, C12, C13, C14, C17	
0805	100 nF	2	C8, C10	
0805	100 nF	1	C3	
E	220 $\mu$ F/16 V	2	C1, C2	tantalový kondenzátor
SOD323	BAT165	1	D2	
0805	BLM21PG221SN1D	4	L2, L3, L4, L5	
0603	LED/RED	1	PWR	červená LED
D-PAK	LF33	1	IC1	
SOIC8	MP1482DS	1	U1	
DO-214AA	SMBJ13A	1	D1	
20QFN	TPA2012D	1	IC2	
–	NXW-02K	2	SPKR-L, SPKR-R	
–	NXW-04K	2	UART, USB	
–	NXW-06K	1	LCD-BL	
–	ZL310-2X20P	1	LCD	kolíková lišta 2 mm, 2×15 pinů, 90 °
–	C68145S	1	DC-IN	
–	3000TR	1	BAT-RTC	pouzdro na baterii
–	DS1002-02-2X24BT1F	2	CB-1 (GPIO)	dutinková lišta 2 mm, 2×24 pinů
–	DS1002-02-2X02BT1F	1	CB-1 (PWR)	dutinková lišta 2 mm, 2×2 piny
–	DS1002-02-2X04BT1F	1	CB-1 (USB)	dutinková lišta 2 mm, 2×4 piny
–	DS1002-02-1X01BT1F	1	CB-1 (RTC)	dutinková lišta 2 mm, 1×1 pin

Tab. A.1: Seznam součástek základní desky.

## B FOTOGRAFIE PROTOTYPU



Obr. B.1: Prototyp docházkového terminálu.



Obr. B.2: Detail propojení komponent terminálu.

# C BUILDROOT – KONFIGURACE

## C.1 Konfigurace prostředí (defconfig)

```
1 BR2_arm=y
2 BR2_cortex_a8=y
3 BR2_ARM_EABIHF=y
4 BR2_ARM_FPU_NEON=y
5 BR2_TOOLCHAIN_BUILDROOT_EGLIBC=y
6 BR2_TOOLCHAIN_BUILDROOT_CXX=y
7 BR2_TARGET_GENERIC_ROOT_PASSWD="root"
8 BR2_ROOTFS_POST_BUILD_SCRIPT="board/a20_cubieboard2/post-build.sh"
9 BR2_ROOTFS_POST_IMAGE_SCRIPT="board/a20_cubieboard2/post-image.sh"
10 BR2_LINUX_KERNEL=y
11 BR2_LINUX_KERNEL_CUSTOM_GIT=y
12 BR2_LINUX_KERNEL_CUSTOM_REPO_URL="https://github.com/linux-sunxi/linux-sunxi.git"
13 BR2_LINUX_KERNEL_CUSTOM_REPO_VERSION="b6eb2b9b770537ff320c52342174d2bed56b574d"
14 BR2_LINUX_KERNEL_DEFCONFIG="sun7i"
15 BR2_LINUX_KERNEL_INSTALL_TARGET=y
16 BR2_PACKAGE_BUSYBOX_SHOW_OTHERS=y
17 BR2_PACKAGE_QT5=y
18 BR2_PACKAGE_QT5BASE_LICENSE_APPROVED=y
19 BR2_PACKAGE_QT5BASE_NETWORK=y
20 BR2_PACKAGE_QT5BASE_CONCURRENT=y
21 BR2_PACKAGE_QT5BASE_SQL=y
22 BR2_PACKAGE_QT5BASE_SQLITE_QT=y
23 BR2_PACKAGE_QT5BASE_WIDGETS=y
24 BR2_PACKAGE_QT5BASE_TSLIB=y
25 BR2_PACKAGE_QT5BASE_GIF=y
26 BR2_PACKAGE_QT5BASE_JPEG=y
27 BR2_PACKAGE_QT5BASE_PNG=y
28 BR2_PACKAGE_QT5MULTIMEDIA=y
29 BR2_PACKAGE_QT5SERIALPORT=y
30 BR2_PACKAGE_SUNXI_MALI=y
31 BR2_PACKAGE_SUNXI_MALI_DBG=y
32 BR2_PACKAGE_PHP=y
33 BR2_PACKAGE_PHP_EXT_PDO=y
34 BR2_PACKAGE_PHP_EXT_PDO_SQLITE=y
35 BR2_PACKAGE_FONTCONFIG=y
36 BR2_PACKAGE_JQUERY_UI=y
37 BR2_PACKAGE_JQUERY_VALIDATION=y
38 BR2_PACKAGE_LIBFCGI=y
39 BR2_PACKAGE_LIGHTTPD=y
40 BR2_PACKAGE_LIGHTTPD_OPENSSL=y
41 BR2_PACKAGE_LIGHTTPD_ZLIB=y
42 BR2_PACKAGE_OPENSSSH=y
43 BR2_PACKAGE_KMOD=y
44 BR2_PACKAGE_KMOD_TOOLS=y
45 BR2_TARGET_UBOOT_SUNXI=y
46 BR2_TARGET_UBOOT_SUNXI_BOARDNAME="Cubieboard2"
47 BR2_TARGET_UBOOT_SUNXI_GITHUB_LOCATION= \
    "https://github.com/linux-sunxi/u-boot-sunxi/archive/"
48 BR2_TARGET_UBOOT_SUNXI_GITHUB_TAG="v2013.07-sunxi"
49 BR2_TARGET_UBOOT_SUNXI_SPL=y
50 BR2_PACKAGE_HOST_SUNXI_TOOLS=y
51 BR2_PACKAGE_HOST_UBOOT_TOOLS=y
```

## C.2 Post-build skript

```
1  #!/bin/sh
2
3  BOARD_DIR="$(dirname $0)"
4  BOOT=$TARGET_DIR/boot/
5
6  cp -v $BOARD_DIR/script.bin $BOOT
7  cp -v $BOARD_DIR/uEnv.txt $BOOT
8  $TARGET_DIR/../../host/usr/bin/mkimage -C none -A arm -T script -d \
    $BOARD_DIR/boot.cmd $BOARD_DIR/boot.scr
9  cp -v $BOARD_DIR/boot.scr $BOOT
10
11 if [ -e $BINARIES_DIR/u-boot.bin ] ; then
12     cp $BINARIES_DIR/u-boot.bin $BOOT
13 fi
14
15 if [ -e $BINARIES_DIR/sunxi-spl.bin ] ; then
16     cp $BINARIES_DIR/sunxi-spl.bin $BOOT
17 fi
18
19 ### Rootfs customization
20
21 # Enable rc.local support
22 grep -q ". /etc/rc.local" $TARGET_DIR/etc/init.d/rcS || cat >> \
    $TARGET_DIR/etc/init.d/rcS << EOF
23
24 # Run rc.local if it exists
25 if [ -x /etc/rc.local ]; then
26     . /etc/rc.local
27 fi
28
29 EOF
30
31 # Copy rootfs additions
32 echo "Rootfs additions:"
33 cp -av $BOARD_DIR/rootfs-additions/* $TARGET_DIR/
34
35 # Remove S80mali (breaks QT framebuffer)
36 if [ -x $TARGET_DIR/etc/init.d/S80mali ] ; then
37     rm $TARGET_DIR/etc/init.d/S80mali
38 fi
```

## D OBSAH PŘILOŽENÉHO CD

/	
—prace.....	elektronická verze této práce
—prilohy	
—DPS.....	návrh DPS v programu Eagle
—buildroot .....	prostředí Buildroot s konfigurací pro Cubieboard2
—aplikace .....	zdrojové kódy docházkové aplikace
—web .....	zdrojové kódy webového rozhraní
—foto .....	fotodokumentace hotového terminálu