

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Bc. Kristián Košťál

PiMiner

semestrálny projekt

Predmet: Vnorené systémy

Študijný program: SI

Študijný odbor: Softvérové inžinierstvo

Termín cvičenia: štvrtok 10:00

máj 2016



SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
Fakulta informatiky a informačných technológií STU
Ústav počítačových systémov a sietí

ZADANIE SEMESTRÁLNEHO PROJEKTU

Predmet: **VNORENÉ SYSTÉMY**

Riešitelia: **Bc. Kristián Košťál**

Školský rok: **2015/2016**

Názov projektu: PiBitMiner - dolovací systém na báze raspberry na bitcoin menu

Zadanie:

Navrhnite a zrealizujte Bitcoin Miner za pomoci vývojového kitu Raspberry Pi. Navrhnite projekt tak, aby vám bolo umožnené používať Raspberry Pi ako správcu a sledovač stavu pre váš USB Bitcoin ASIC miner. Projekt zahŕňa LCD pre zobrazenie hashrate, chybovosti, zdieľania dát, sieťových problémov, trvania ťažby, a aktuálneho výmenného kurzu. Zariadenie musí byť možné používať za izbovej teploty a tak vyriešte aj problém s chladením a napájaním ASIC dolovacieho zariadenia.

Projekt musí obsahovať:

1. Analýzu problematiky
2. Opis postupu riešenia
3. Výsledky riešenia a ich zhodnotenie
4. Zoznam použitej literatúry
5. Technickú dokumentáciu
6. Pamäťové médium obsahujúce softvérové časti riešenia spolu so všetkou dokumentáciou

Termín odovzdania: Posledné cvičenie z predmetu v letnom semestri 2015/2016

V Bratislave dňa 16.5.2016

Obsah

1. Úvod.....	4
2. Analýza	5
2.1 Softvér	5
2.2 Hardvér	6
3. Návrh.....	9
3.1 Potrebný hardvér a softvér.....	9
4. Implementácia.....	13
4.1 LCD	13
4.2 WiFi	14
4.3 ASIC	14
4.4 Využitie API - obslužný program pre LCD.....	15
5. Zhodnotenie	16
Bibliografia.....	17
Príloha A - používateľská príručka	18
Príloha B - zdrojové kódy	19

1. Úvod

Táto práca pojednáva o dolovacom zariadení na virtuálnu menu Bitcoin. Toto zariadenie používa vývojový kit Raspberry Pi Model A (jeden z prvých Raspberry Pi), Asic bitcoin miner vo forme USB zariadenia a ďalšie periférie. Mena Bitcoin sa už pomaly stáva minulosťou a jej dolovanie je veľmi obtiažne, pretože stále narastá náročnosť jednotlivých blokov a aj počet jednotlivých "dolujúcich" ľudí a tiež výkon jednotlivých zariadení na dolovanie.

Ale čo je to vlastne Bitcoin? Mena Bitcoin je digitálnou menou, teda menou existujúcou len v podobe dát respektíve čísel a nie vo forme fyzických platidiel. Rovnako ako iné digitálne meny sa pre zabezpečenie svojej funkčnosti spolieha na kryptografiu. Bitcoin je plne decentralizovaná P2P mena nezávislá na akejkoľvek autorite, čo sa týka technickej prevádzky, vydávania novej meny aj dohľadu nad transakciami. Na dolovanie Bitcoinov sa v minulosti používali CPU, neskôr GPU a v dnešnej dobe ASIC alebo FPGA. S Bitcoin menou je možné nakladať vo forme tzv. elektronických Bitcoin mincí, pričom minca je štandardne v danom čase vlastnená jednou tzv. Bitcoin adresou. S Bitcoin adresou je spätý tajný kľúč, ktorým vlastník adresy preukazuje svoje vlastníctvo tejto adresy realizovaním kryptografického podpisu a môže tak mince previesť na nového vlastníka. Bitcoin používa 256-bitové kľúče asymetrického algoritmu ECDSA, Elliptic Curve Digital Signature Algorithm. Bitcoin adresa je odvodená z verejného kľúča prislúchajúceho tomuto tajnému kľúču pomocou hash funkcií SHA-256 a RIPEMD-160, pričom obsahuje aj kontrolnú sumu pre zabránenie omylom v adrese [3].

Práca je rozdelená na kapitoly, prv je uvedená analýza, kde sa venujem spomenutým Asic bitcoin miner zariadeniam a existujúcemu softvéru na ťažbu bitcoinov. V ďalšej kapitole sa venujem návrhu, kde je spomenutý všetok potrebný hardvér a softvér na poskladanie kitu. Potom nasleduje časť s implementáciou a napokon zhodnotenie celého projektu.

2. Analýza

V nasledujúcej kapitole si rozoberieme problematiku Asic dolovacích zariadení a tiež dolovacieho softvéru.

2.1 Softvér

Na dolovanie virtuálnych "peňazí" existuje veľké množstvo programov, ktoré funguje ako sprostredkovateľ výpočtového výkonu do poolu a tiež informácií z poolu na rôzne zariadenia. Nasledovný zoznam uvádza všetky dolovacie softvéry aj s krátkym popisom:

- [EASYMINER](#) – pre Windows, Linux, Android, s podporou GUI. Funguje ako obalovač pre cgminer alebo bfgminer.
- [BFGMiner](#) – Modulárny FPGA/GPU miner v jazyku C. Má podporu aj pre ASIC.
- [CGMINER](#) – multi-vláknový multi-pool GPU, FPGA a ASIC Bitcoin miner s podporou ATI GPU monitoringu, pretaktovania a ovládania ventilátorov. Má podporu pre bitcoiny ale aj iné intermé coins(over)clocking and fanspeed support for bitcoin and derivative coins.
- [50Miner](#) – GUI frontend pre Windows(Poclbm, Phoenix, DiabloMiner).
- [BTCMiner](#) – Bitcoin Miner pre dosky typu ZTEX FPGA.
- [Bit Moose](#) – Softvér, ktorý umožňuje spúšťať dolovače ako služby systému Windows.
- [Poclbm](#) – Python/OpenCL GPU miner (GUI(Windows & MacOS X)).
- [Poclbm-mod](#) – upravená, vylepšená verzia [Poclbm](#) (GUI).
- [DiabloMiner](#) – Java/OpenCL GPU miner (len pre MAC OS X GUI).
- [RPC Miner](#) –RPC miner, ktorý je možný ovládať aj z remote prístupu (MAC OS X GUI).
- [Phoenix miner](#) – Dolovač, ktorý vie pracovať aj s pamäťami podtaktovanými na 300MHz a preskúmvava celý 32bitový priestor.
- [Cpu Miner](#) – jednoduchý klientský program, ktorý dokáže ťažiť aj z poolu, ale aj ako sólo dolovanie.

- [Ufasoft miner](#) – CPU/GPU dolovač pre Windows, ktorý ťaží z poolov.
- [Pyminer](#) – napísaný komplet v jazyku Python a preto je dosť pomalý.
- [Remote miner](#) – softvér na dolovanie z poolu. prostredníctvom JSON protokolu dobre funguje s RPC miner.
- [Open Source FPGA Bitcoin Miner](#) – softvér špeciálne určený pre FPGA dosky.
- [Flash Player Bitcoin Miner](#) – Len nejaký pokusný kúsok od Adobe Flash Player [2].

Keďže používame na dolovanie špeciálny USB ASIC a z tejto sumarizácie vidieť, že jedine dva programy majú podporu pre ASIC a to konkrétne BFGMiner alebo CGMiner, tak sme sa museli rozhodnúť pre jeden z nich. Oba sa správajú pomerne rovnako a majú takmer totožnú funkcionality, takže výber kandidáta nie je ničím ovplyvnený a je to len o sympatiách k jednému z týchto programov. My sme si vybrali CGMiner.

2.2 Hardvér

Na trhu existuje mnoho hardvérových zariadení na dolovanie Bitcoinov. Ja sa už špeciálne v tejto práci venujem len ASIC dolovacím zariadeniam, pretože aj tých je už veľa a tak uvádzam prehľadovú tabuľku 1 s cenou a výkonom.

Tabuľka 1 Prehľad jednotlivých ASIC dolovacích zariadení vrátane výkonu a ceny [1]

Zariadenie	Výkon	Efektivita	Cena
AntMiner S1	180 Gh/s	2.0 W/Gh	\$299.0
AntMiner S2	1000 Gh/s	1.1 W/Gh	\$2259.0
AntMiner S3	441 Gh/s	0.77 W/Gh	\$382.0
AntMiner S4	2000 Gh/s	0.7 W/Gh	\$1400.0
AntMiner S5	1155 Gh/s	0.51 W/Gh	\$370.0
AntMiner S5+	7722 Gh/s	0.44 W/Gh	\$2307.0
AntMiner S7	4.73 Th/s	0.25 W/Gh	\$595.99
AntMiner U1	2 Gh/s	1.25 W/Gh	\$29.0
AntMiner U2	2 Gh/s	1.0 W/Gh	\$32.99
AntMiner U3	63 Gh/s	1.0 W/Gh	\$38.0
ASICMiner BE Blade	11 Gh/s	7.72 W/Gh	\$350.0
ASICMiner BE Cube	30 Gh/s	6.67 W/Gh	\$550.0
ASICMiner BE Sapphire	0,33 Gh/s	7.59 W/Gh	\$20.0

ASICMiner BE Tube	800 Gh/s	1.13 W/Gh	\$320.0
ASICMiner BE Prisma	1400 Gh/s	0.79 W/Gh	\$600.0
Avalon Batch 1	66 Gh/s	9.35 W/Gh	\$1299.0
Avalon Batch 2	82 Gh/s	8.54 W/Gh	\$1499.0
Avalon Batch 3	82 Gh/s	8.54 W/Gh	\$1499.0
Avalon2	300 Gh/s	N/A	\$3075.0
Avalon3	800 Gh/s	N/A	N/A
Avalon6	3.5 Th/s	0.29 W/Gh	\$750.95
bi*fury	5 Gh/s	0.85 W/Gh	\$209.0
BFL SC 5Gh/s	5 Gh/s	6.0 W/Gh	\$274.0
BFL SC 10 Gh/s	10 Gh/s	N/A	\$50.0
BFL SC 25 Gh/s	25 Gh/s	6.0 W/Gh	\$1249.0
BFL Little Single	30 Gh/s	N/A	\$649.0
BFL SC 50 Gh/s	50 Gh/s	6.0 W/Gh	\$984.0
BFL Single 'SC'	60 Gh/s	4.0 W/Gh	\$1299.0
BFL 500 GH/s Mini Rig SC	500 Gh/s	5.4 W/Gh	\$22484.0
BFL Monarch 700GH/s	700 Gh/s	0.7 W/Gh	\$1379.0
Bitmine.ch Avalon Clone 85GH	85 Gh/s	7.65 W/Gh	\$6489.0
Black Arrow Prospero X-1	100 Gh/s	1.0 W/Gh	\$370.0
Black Arrow Prospero X-3	2000 Gh/s	1.0 W/Gh	\$6000.0
Blue Fury	3 Gh/s	1.0 W/Gh	\$140.0
BTC Garden AM-V1 310 GH/s	310 Gh/s	1.05 W/Gh	\$309.0
BTC Garden AM-V1 616 GH/s	616 Gh/s	1.05 W/Gh	\$350.0
CoinTerra TerraMiner IV	1600 Gh/s	1.31 W/Gh	\$1500.0
HashBuster Micro	20 Gh/s	1.15 W/Gh	\$688.0
HashCoins Apollo v3	1100 Gh/s	0.91 W/Gh	\$599.0
HashCoins Zeus v3	4500 Gh/s	0.67 W/Gh	\$2299.0
HashFast Baby Jet	400 Gh/s	1.1 W/Gh	\$5600.0
HashFast Sierra	1200 Gh/s	1.1 W/Gh	\$7080.0
HashFast Sierra Evo 3	2000 Gh/s	1.1 W/Gh	\$6800.0
Klondike	5 Gh/s	6.15 W/Gh	\$20.0
KnCMiner Mercury	100 Gh/s	2.5 W/Gh	\$1995.0
KnC Saturn	250 Gh/s	1.2 W/Gh	\$2995.0
KnC Jupiter	500 Gh/s	1.2 W/Gh	\$4995.0

KnC Neptune	3000 Gh/s	0.7 W/Gh	\$12995.0
LittleFury	3,4 Gh/s	1,32 W/Gh	\$50
Metabank	120 Gh/s	1.42 W/Gh	\$2160.0
NanoFury / IceFury	2 Gh/s	1.25 W/Gh	N/A
NanoFury NF2	4 Gh/s	1.35 W/Gh	\$50.0
BPMC Red Fury USB	2.5 Gh/s	0.96 W/Gh	N/A
ROCKMINER R3-BOX	450 Gh/s	1.0 W/Gh	\$200.0
ROCKMINER R4-BOX	470 Gh/s	1.0 W/Gh	\$210.0
ROCKMINER Rocket	450 Gh/s	1.07 W/Gh	\$599.0
ROCKMINER R-BOX	32 Gh/s	1.41 W/Gh	\$65.0
ROCKMINER R-BOX 110G	110 Gh/s	1.09 W/Gh	\$88.0
ROCKMINER T1 800G	800 Gh/s	1.25 W/Gh	\$325.0
Spondooliestech SP10 Dawson	1400 Gh/s	0.89 W/Gh	\$2845.0
SP20 Jackson	1.3-1.7 Th/s	0.65 W/Gh	\$248.99
Spondooliestech SP30 Yukon	4500 Gh/s	0.67 W/Gh	\$4121.0
Spondooliestech SP31 Yukon	4900 Gh/s	0.61 W/Gh	\$2075.0
Spondooliestech SP35 Yukon	5500 Gh/s	0.66 W/Gh	\$2235.0
TerraHash Klondike 16	5 Gh/s	7.11 W/Gh	\$250.0
TerraHash Klondike 64	18 Gh/s	7.06 W/Gh	\$900.0
TerraHash DX Mini (full)	90 Gh/s	7.11 W/Gh	\$6000.0
TerraHash DX Large (full)	180 Gh/s	7.11 W/Gh	\$10500.0
Twinfury	5 Gh/s	0.85 W/Gh	\$216.0
Avalon USB Nano3	3.6 Gh/s	0.85 W/Gh	\$55.0

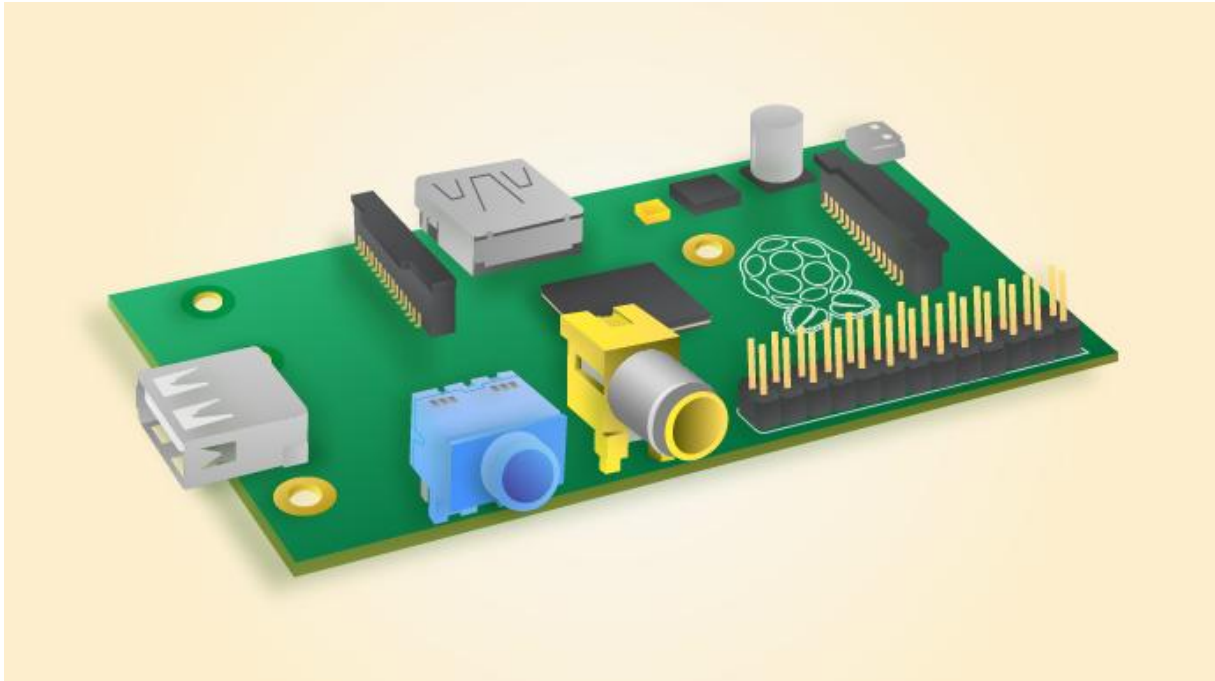
Týchto zariadení každým dňom pribúda a nedokážu robiť nič iné, iba dolovať kryptografickú menu na báze SHA-256, medzi ktoré patrí aj Bitcoin. Pre vyskúšanie sme zhľadali zariadenie, ktoré nebude veľmi drahé, bude mať kompaktné rozmery a na napájanie mu bude stačiť USB hub a tak sme dospeli ku zariadeniu Antminer U2.

3. Návrh

V tejto kapitole sa venujeme potrebnému vybaveniu na skonštruovanie dolovacieho zariadenia Bitcoinov.

3.1 Potrebný hardvér a softvér

1. Srdce celého systému - Raspberry Pi Model A

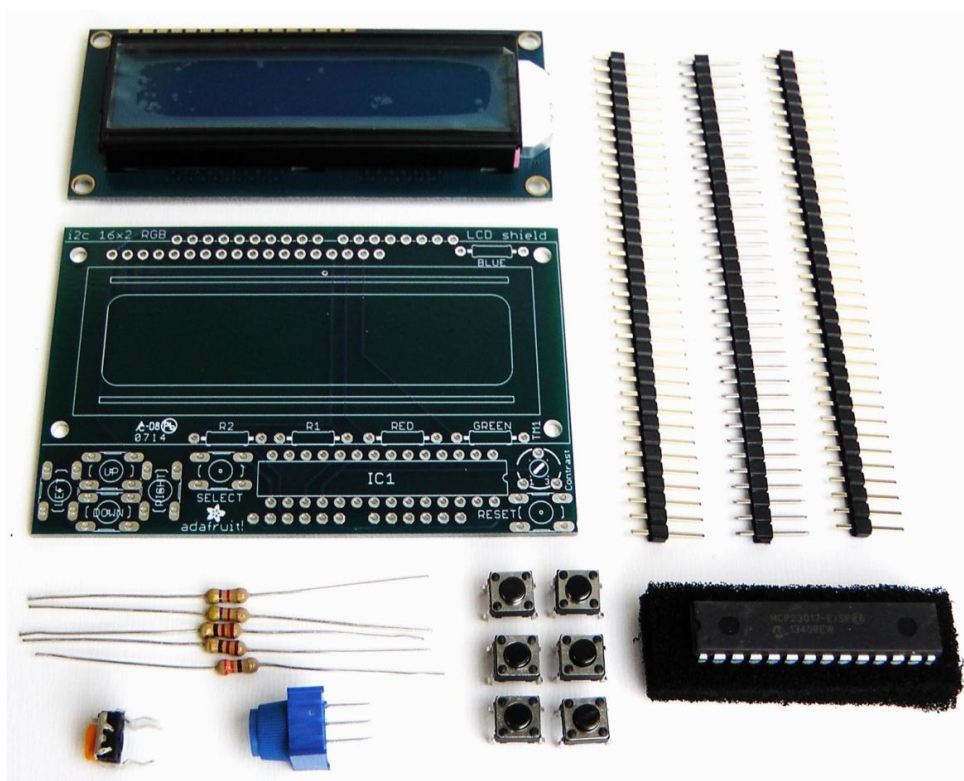


Obr. 1 Raspberry Pi Model A

Doska bude slúžiť ako srdce systému a bude spájať v sebe kontrolór a monitor stavu pre USB bitcoin zariadenia. Samozrejmosťou je potreba SD karty, na ktorú sa nahrá operačný systém Raspbian a príslušne nastaví pre fungovanie. My sme použili najnovší Raspbian Jessie. Ako ho nahráť na pamäťovú kartu sa možno dočítať na odkaze¹.

¹ <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

2. LCD Adafruit 16x2 LCD + keypad kit

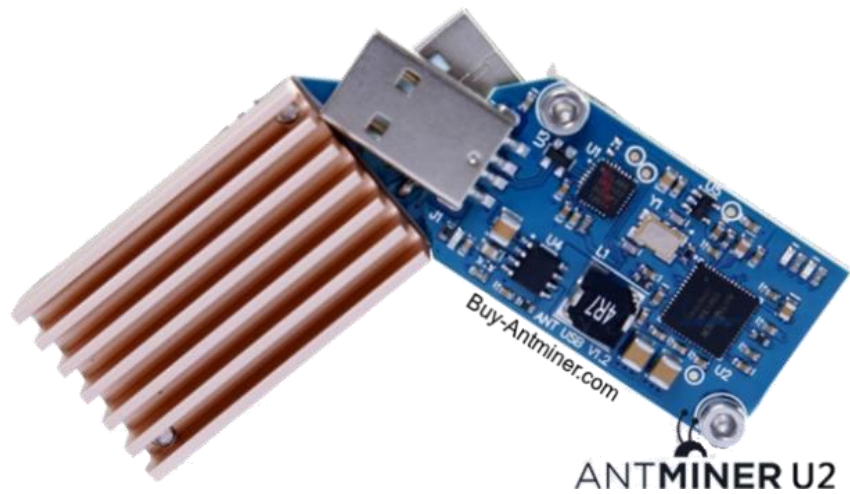


Obr. 2 LCD + keypad

Tento kit je potrebné si poskladať a prispájkovať jednotlivé piny². Po pripojení cez rozhranie I2C, ktoré šetrí GPIO piny, ktorých je už aj tak málo, sa na displeji budú zobrazovať štatistiky ako hashrate, miera chybovosti, podiely jednotlivých prijatých/odmietnutých dát, sieťová zložitosť, dĺžka dolovania a aktuálny kurz zmenárne. Pre správne fungovanie budeme potrebovať knižnice Adafruit I2C, ktorých nastaveniu sa venujeme v implementácii.

² <https://learn.adafruit.com/adafruit-16x2-character-lcd-plus-keypad-for-raspberry-pi/assembly>

3. USB bitcoin dolovacie zariadenie



Obr. 3 AntMiner U2

Mozog celého systému, ASIC, pomocou ktorého budeme dolovať bitcoiny. Toto zariadenie je určené iba na dolovanie bitcoinov a v pretaktovanom stave dosahuje výkon, cca 2,2 Gh/s, pre porovnanie, kebyže použijeme čisto Raspberry Pi model B, dosiahneme výkon v pretaktovanom stave 0,2 Mh/s. Jeho nastaveniu a taktiež pretaktovaniu sa venujeme v implementácii. V pretaktovanom stave budeme zariadenie chladit' USB ventilátorom.

4. Napájaný USB hub



Obr. 4 USB hub s napájaním

Keďže ASIC miner potrebuje vyššie napätie aj prúd, ako je Raspberry Pi schopné dodať, museli sme siahnuť po externe napájanom USB hube. Naše Raspberry Pi má aj tak len jeden usb port, takže aj kvôli rozšíreniu usb portov je toto správny krok, pretože sem vieme pripojiť mnoho ďalších zariadení ako napríklad aj USB wifi dongle.

5. USB WiFi Dongle TP-LINK TL-WN725N

Zariadenie, cez ktoré sa pripojíme do WiFi siete, aby bol celý finálny produkt prenosný kamkoľvek.



Obr. 5 TP-Link WiFi dongle

6. Softvér

Na základné fungovanie dolovania bitcoinov budeme potrebovať Mining pool account a Bitcoin peňaženku. Na internete sa nachádza mnoho možností. My sme zvolili ako pool, tzv. Slushpool, produkt od Čechov, ktorý sa teší veľkej obľube. Ďalej bude potrebný softvér, ktorý dokáže dolovať bitcoiny pomocou iného zariadenia ako je host pc, akým je napríklad cgminer, ktorý sme spomínali už v analýze. Bude ho potrebné ešte nastaviť tak, aby fungoval s ASIC USB zariadením spomenutým vyššie. No a napokon budeme musieť vytvoriť softvér, pomocou ktorého pôjdu výpisy na displej a budú môcť byť prepínané cez tlačidlá.

4. Implementácia

V tejto kapitole sa venujem nastaveniu všetkých knižníc a programov pre vzájomnú komunikáciu.

4.1 LCD

LCD od firmy Adafruit sme si poskladali podľa návodu už v predchádzajúcej kapitole, ale teraz ho treba ešte aj nastaviť. Keďže funguje cez I2C budeme potrebovať knižnice a následne upraviť základný konfig pre podporu I2C:

```
1. sudo apt-get install python-smbus
2. sudo apt-get install i2c-tools
3. sudo raspi-config
```

V konfigu nastavíme nasledovné: Advanced options -> I2C -> YES -> YES. Následne reštartneme. Po zapnutí musíme ešte upraviť ručne niektoré súbory:

```
1. sudo nano /etc/modules
```

Na koniec súboru pridáme dva riadky:

```
1. i2c-bcm2708
2. i2c-dev
```

Ďalší súbor:

```
1. sudo nano /boot/config.txt
```

Na koniec pridáme dva riadky:

```
1. dtparam=i2c1=on
2. dtparam=i2c_arm=on
```

Displej máme nastavený a môžeme ho overiť pomocou knižnice Adafruit_Python_CharLCD³.

³ https://github.com/adafruit/Adafruit_Python_CharLCD

4.2 WiFi

WiFi potrebujeme nastaviť manuálne, keďže sa nachádzame v konzolovom prostredí a to nasledovne:

```
1. sudo nano /etc/networking/interfaces
```

Tam pridáme riadky:

```
1. auto wlan0
2. inet ipv4 wlan0 dhcp
3. wpa-ssid "NázovSiete"
4. wpa-psk "HesloSiete"
```

4.3 ASIC

Po pripojení USB hubu s ASIC zariadením ešte stále raspberry nerozpoznáva toto zariadenie ako známe, preto budeme potrebovať program cgminer a správne ho nastaviť.

```
1. git clone https://github.com/ckolivas/cgminer
2. cd cgminer
3. sudo ./autogen.sh
4. sudo ./configure --enable-icarus #pre náš konkrétny ASIC Antminer U2
5. sudo make
```

Pre správne fungovanie si vytvoríme konfiguračný súbor "config.conf," ktorý bude spúšťať dolovanie na našom špecifickom ASIC a jeho obsah bude nasledovný:

```
1. {
2.   "pools" : [
3.     {#adresa dolovacieho mieste
4.       "url" : "stratum+tcp://stratum.bitcoin.cz:3333",
5.       "user" : "kikiakoki.worker1",
6.       "pass" : "x" #hocico, nezalezi na tom
7.     }
8.   ],
9.   "api-listen" : true, #aktivacia API
10.  "api-port" : "4028",
11.  "expiry" : "120",
12.  "failover-only" : true,
13.  "log" : "5",
14.  "no-pool-disable" : true,
```

```
15.     "queue" : "2",
16.     "scan-time" : "60",
17.     "worktime" : true,
18.     "shares" : "0",
19.     "kernel-path" : "/usr/local/bin",
20.     "api-allow" : "0/0",
21.     "icarus-options" : "115200:1:1",           #specifické nastavenia pre nás ASIC
22.     "icarus-timing" : "3.0=100"
23. }
```

Potom je možné spustiť samotný proces dolovania cez príkaz:

```
1. sudo nohup ./cgminer --config config.conf >/dev/null 2>&1&
```

Tým sa zapne program CGMiner, začne sa dolovať a aktivuje sa API na prístup k údajom.

4.4 Využitie API - obslužný program pre LCD

Pre správne výpisy na displej museli byť implementované dve triedy:

- PiBitMiner.py - tá ma na starosti výpisy na displej, obsluhu tlačidiel, údaje čerpá z api,
- CGMinerApi.py - trieda, ktorá spracúva volania na API programu cgminer.

Ďalšie knižnice, ktoré boli prebraté:

- Adafruit_CharLCDPlate.py - knižnica potrebná pre správne fungovanie displeja,
 - odtiaľ boli prebraté dve funkcie:
 - **message(text)** - vypíše string na displej. v stringu môže byť použitý znak pre nový riadok (\n),
 - **clear()** - vyčistí displej a dá kurzor pre výpisy na začiatočnú pozíciu.
- Adafruit_I2C.py - keďže displej aj tlačidlá fungujú cez I2C je potrebné použiť knižnicu aj pre toto rozhranie.

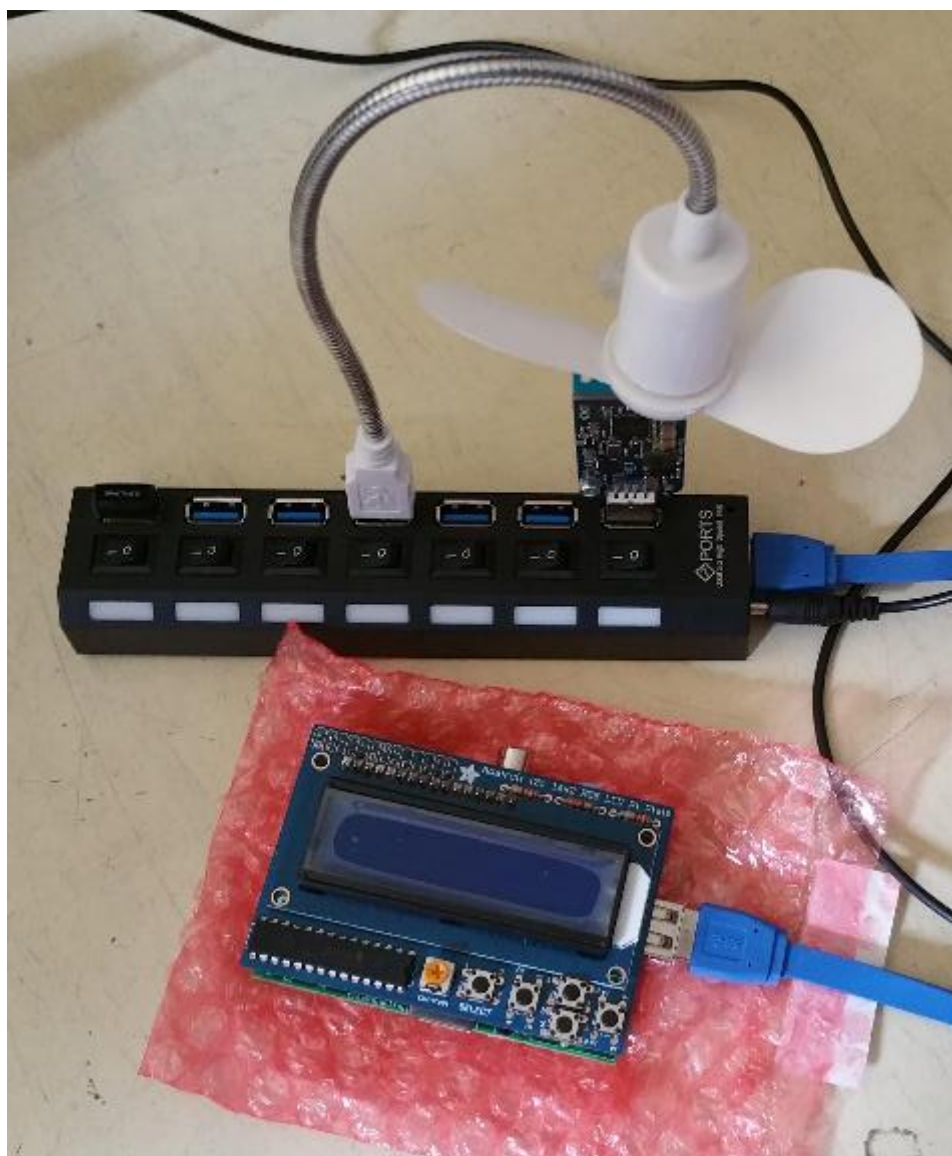
Tlačidlami hore/dole sa mení cez obrazovky:

- prvá - prijaté a zamietnuté transakcie, chybovosť hardvéru, priemerný dolovací výkon,
- druhá - adresa aktuálneho poolu a aktuálneho pripojeného workera,
- tretia - čas od zapnutia a aktuálna zložitosť dolovania,
- štvrtá - počet dolovacích zariadení a percentuálna chybovosť,
- piata - aktuálna cena bitcoinu v USD.

Tlačidlami doľava/doprava sa posúva výstup ak sa nezmestil.

5. Zhodnotenie

Projekt bol úspešne zvládnutý a dokončený. Bol poskladaný vnorený systém (obr. 6), ktorý pomocou ASIC dolovacieho zariadenia doluje virtuálnu menu Bitcoin. Keďže dochádzalo ku prehrievaniu tak ten ASIC je chladený USB ventilátorom. Na napájanie bol použitý externý USB hub, pretože samotné Raspberry by nezvládalo napájať toľko periférií. Kit bol testovaný v ostrej prevádzke 24/7 po dobu 30 dní a za tento čas sa podarilo vydolovať 0,00018984 BTC, čo je pri kurze 400€/1BTC cca 0,08€. Z toho vyplýva, že dolovanie Bitcoinov je už minulosť pri takýchto nízkych výkonoch, aké sa dajú bežne zohnať a stáva sa to devízou pár veľkých hráčov, ktorí investujú do výkonu každodenne. Projekt považujem za konečný a nevidím priestor pre ďalšie rozšírenie za súčasných podmienok a cien iných zariadení.



Obr. 6 Výsledok

Bibliografia

[1] BITCOINMINING Best Bitcoin Mining Hardware ASICs Comparison. Publikované: [online]. 2016. [cit. 2016-05-08]. Dostupné na internete: <<https://www.bitcoinmining.com/bitcoin-mining-hardware/>>.

[2] BITCOINX Bitcoin Mining Software - Bitcoinx. Publikované: [online]. 2016. [cit. 2016-05-08]. Dostupné na internete: <<http://www.bitcoinx.com/bitcoin-mining-software/>>.

[3] DSL.SK DSL.sk - Bitcoin I. - Čo je a ako funguje Bitcoin. Publikované: [online]. 2013. [cit. 2016-04-20]. Dostupné na internete: <<http://www.dsl.sk/article.php?article=13988>>.

Príloha A - používateľská príručka

Zapojíme celý vnorený systém, spustíme. Pripojíme sa cez ssh na raspberry s prihlasovacími údajmi pi:raspberry. Nainštalujeme všetky knižnice a závislosti:

```
sudo apt-get install libusb-1.0-0-dev libusb-1.0-0 libcurl4-openssl-dev  
libncurses5-dev libudev-dev screen libtool automake pkg-config libjansson-  
dev git
```

Nainštalujeme cgminer:

```
git clone https://github.com/AdvancedStyle/cgminer && cd cgminer  
sudo ./autogen.sh  
sudo ./configure --enable-icarus  
sudo make
```

Všetko spustíme:

```
sudo nohup ./cgminer --config config.conf >/dev/null 2>&1 &  
sudo python ../PiBitMiner.py &
```

Ak chceme spúšťať automatizovať pri štarte systému, tak pridáme do súboru "/etc/rc.local":

```
cd /home/pi/cgminer  
nohup ./cgminer --config config.conf >/dev/null 2>&1 &  
cd ..  
python PiBitMiner.py &
```

Príloha B - zdrojové kódy

PiBitMiner.py

```
import socket
import json

import sys, subprocess, time, urllib2, socket
from Adafruit_CharLCDPlate import Adafruit_CharLCDPlate
from CgminerAPI import CgminerAPI

#Vytvorene pre ucely projektu na predmet Vnorene systemy 2016
#Kristian Kostal

#trieda na spracovanie sprav od API a nasledne vypisy na displej, plus ovladanie displeja tlacidlami,
obsluzny program pre LCD

HOLD_TIME = 3.0 #Cas (sekundy) potrebny na drzanie tlacidla select pre vypnutie Raspberry
REFRESH_TIME= 3.0 #Cas (sekundy) medzi jednotlivymi refreshmi
HALT_ON_EXIT= False
lcd = Adafruit_CharLCDPlate()
prevCol = -1
prev = -1
lastTime= time.time()

def shutdown():
    lcd.clear()
    lcd.backlight(lcd.OFF)
    exit(0)

def abbrev(v):
    v = int(v)
    if v >= 1000:
        va = float(v) / 1000.0
        vs = '%.1f' % va
        vs = vs + 'k'
        return vs
    elif v >= 1000000:
        va = float(v) / 1000000.0
        vs = '%.1f' % va
        vs = vs + 'm'
        return vs
    #bilion
    else:
        return '%d' % v

def hashrate(h):
    u = 'Gh/s'
    if h >= 1000.0:
        u = 'Th/s'
        h = h / 1000.0
    elif h >= 1000000.0:
        u = 'Ph/s'
        h = h / 1000000.0
    s = '%s %s' % (h, u)
    return s

def displaysimplesummary():
    lcd.clear()
    try:
        cgminer = CgminerAPI()
        summarydata = cgminer.command('summary')
        summary = summarydata.get("SUMMARY",{})[0]
        acc = abbrev(summary.get("Accepted",{}))
        rej = abbrev(summary.get("Rejected",{}))
        hw = abbrev(summary.get("Hardware Errors",{}))
        s1 = 'OK:%s NIE:%s ZLE:%s' % (acc, rej, hw)
        s2 = 'avg:%s' % hashrate(float(summary.get("GHS av",{})))
        lcd.clear()
        lcd.message(s1 + '\n' + s2)
    except Exception as e:
```

```

        lcd.clear()
        print e
        lcd.message("Cakam na spravu:" + '\n' + "cgminer... ")

#skontroluj pripojenie do siete a vypis IP adresu
t = time.time()
while True:
    lcd.clear()
    lcd.message('testujem\npripojenie ...')
    if (time.time() - t) > 120:
        # ak sa nepodari pripojit do dvoch minut
        lcd.clear()
        lcd.message('nemam \ninternet')
        time.sleep(30)
        exit(0)

    try:
        s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        s.connect(('8.8.8.8', 0))
        lcd.backlight(lcd.ON)
        lcd.clear()
        lcd.message('IP adresa:\n' + s.getsockname()[0])
        time.sleep(5)
        break # uspesne pripojenie
    except:
        time.sleep(1) # skusaj kazdu sekundu

#kontroluj stlacenie klaves
while True:
    b = lcd.buttons()
    if b is not prev:
        if lcd.buttonPressed(lcd.SELECT):
            tt = time.time() #zaciatočný čas stlavenia tlačidla

select
            while lcd.buttonPressed(lcd.SELECT): #cakaj na uvolnenie tlačidla
                if (time.time() - tt) >= HOLD_TIME: #ak dlzim dlhsie ako 3 sekundy
                    lcd.message('vypinam...')
                    time.sleep(1)
                    shutdown() #vypni

raspberry
                    endif
                elif lcd.buttonPressed(lcd.LEFT):
                    display.scrollRight()
                elif lcd.buttonPressed(lcd.RIGHT):
                    display.scrollLeft()
                elif lcd.buttonPressed(lcd.UP):
                    display.modeUp()
                elif lcd.buttonPressed(lcd.DOWN):
                    display.modeDown()

            prev = b
            lastTime = time.time()
        else:
            now = time.time()
            since = now - lastTime
            if since > REFRESH_TIME or since < 0.0:
                #obnov vypisy na displeji
                displaysimplesummary()
                lastTime = now

```

CGMinerApi.py

```

import socket
import json

#Vytvorene pre ucely projektu na predmet Vnorene systemy 2016
#Kristian Kostal

#trieda na komunikacii s API, jedna sa o klasicke API, ku ktoremu dokumentaciu najdeme na
https://github.com/ckolivas/cgminer/blob/master/API-README

class CgminerAPI(object):
    """ Cgminer konstruktor """
    def __init__(self, host='localhost', port=4028):

```

```

self.data = {}
self.host = host
self.port = port

def command(self, command, arg=None):
    """ Inicializuje pripojenie na socket, posle prikaz vo forme json, prijme odpoved a
odkoduje ju """
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    try:
        sock.connect((self.host, self.port))
        payload = {"command": command}
        if arg is not None:
            # Paramter musime dat do stringu, nie int
            payload.update({'parameter': unicode(arg)})

        sock.send(json.dumps(payload))
        received = self._receive(sock)
    finally:
        sock.shutdown(socket.SHUT_RDWR)
        sock.close()

    return json.loads(received[:-1])

def _receive(self, sock, size=4096):
    msg = ''
    while 1:
        chunk = sock.recv(size)
        if chunk:
            msg += chunk
        else:
            break
    return msg

def __getattr__(self, attr):
    def out(arg=None):
        return self.command(attr, arg)
    return out

```