

Príloha A

Plán práce na letný semester

Do vianočných sviatkov očakávame doručenie našich zariadení. Cez sviatky je plán zoznámiť sa so zariadeniami, vyskúšať si ich komunikáciu. Do začiatku budúceho semestra je predpoklad mať prácu so zariadeniami zvládnutú na takej úrovni, že nebude problém začať na nich vyvíjať a testovať naše riešenie. Taktiež je v pláne do začiatku semestra popremýšľať a vybrať vhodný konsenzuálny algoritmus pre blockchain, pouvažovať nad implementáciou blockchainu a vybrať vhodné komunikačné a manažovacie protokoly.

Týždeň	Plán práce
1. týždeň	Konzultácia vybraných riešení a ich schválenie vedúcim. Prípadne vybratie iného riešenia. Preriešenie dôležitých otázok, ktoré vyvstanú počas sviatkov a skúškového obdobia.
2. týždeň	Prípadná dokonzultácia vybraných riešení a ich finálne potvrdenie. Dozodpovedanie dôležitých otázok. Stretnutie s kolegami spolupracujúcimi na rovnakej téme a dohodnutie formy komunikácie a formátu súborov s dátami. Rozchodenie komunikácie na zariadeniach.
3. týždeň	Analýza a začiatok implementácie blockchainu. Rozchodenie komunikácie na zariadeniach.
4. týždeň	Dokončenie implementácie blockchainu a analýza implementácie konsenzuálneho algoritmu. Testovanie blockchainu.
5. týždeň	Implementácia konsenzuálneho algoritmu. Testovanie blockchainu na zariadeniach.
6. týždeň	Doimplementovanie konsenzuálneho algoritmu.
7. týždeň	Testovanie konsenzuálneho algoritmu s blockchainom
8. týždeň	Testovanie komunikácie na zariadeniach s použitím blockchainu a konsenzuálneho algoritmu.
9. týždeň	Spojenie časti práce s kolegami.
10. týždeň	Spojenie časti práce s kolegami a začiatok testovania celkového riešenia.
11. týždeň	Zapracovanie poznatkov z testovania a opravenie chýb. Začiatok písania dokumentácie.
12. týždeň	Finalizácia drobných úprav na riešení. Dopísanie dokumentácie.
13. týždeň	Zapracovanie dokumentácie do finálneho dokumentu. Formálna úprava dokumentu. Finálne odovzdanie.

A.1 Zhodnotenie plánu práce

Plán práce nebol navrhnutý úplne správne, vzhľadom na to, že pri plánovaní sa nepočítalo s tým, že sa v tejto práci budú implementovať viaceré algoritmy, ktoré sa budú následne porovnávať. Zo začiatku plán práce nebolo problém dodržať, no postupne, s pribúdajúcimi akademickými povinnosťami a ubúdajúcim časom bolo problematické plán práce skĺbiť s realitou. Napriek tomu, že približne od polovice semestra nebolo toľko času sa venovať tejto práci, koľko by sme chceli, a napriek tomu, že bol implementovaný jeden algoritmus navyše, od toho, čo sme si navrhli, myslím, že bola práca včas úspešne dokončená, a to najmä kvôli zvýšenému úsiliu ku koncu semestra.

Príloha B

Používateľská príručka

B.1 Príprava prostredia

Na spustenie vyvinutého riešenia je potrebné sketche nahrať na Arduino zariadenia. K tomu je potrebné mať nainštalované Arduino integrované vývojové prostredie (IDE) vo verzii 1.8.8.

Po nainštalovaní musíme do IDE prostredia doplniť náš konkrétny typ Arduino zariadenia pridaním Arduino SAMD Boards prostredníctvom Arduino Boards Manager-a (Nástroje\Doska\Manažér dosiek). Následne medzi ponúkanými typmi dosiek zvolíme nami použitý typ MKR WiFi 1010. Do IDE prostredia je taktiež potrebné doinštalovať knižnice WiFiNINA.h, CRC32.h a ArduinoSTL.h cez Správcu knižníc (Nástroje\Spravovať knižnice). IDE je podporované na Windows XP a vyšších, MacOS X a Linux operačných systémoch. Arduino zariadenie sa pripája ku počítaču s nainštalovaným IDE USB káblom. Po jeho pripojení si OS Windows doinštaluje potrebné ovládače. Následne je potrebné vybrať sériové rozhranie, cez ktoré bude pripojené Arduino zariadenie komunikovať, čo sa vykoná v Serial Port

Menu (Nástroje\Port) zvolením určitého portu označeného ako COMx (kde x je číslo portu).



Obr. B.1: Výber sériového rozhrania COM9, na ktorom komunikuje pripojené Arduino zariadenie

Ako alternatívu je možné použiť webové IDE rozhranie Arduino Web Editor nachádzajúce sa na stránkach Arduino projektu (arduino.cc). Pri využití tejto alternatívy nie je potrebné na počítači nič inštalovať. Arduino Web Editor umožňuje nahrať vytvorené sketchy do cloudu, odiaľ sú dostupné aj z iných zariadení. Na jeho využitie je potrebná registrácia.

B.2 Spustenie sketchu

Pre spustenie Sketchu je potrebné sketch najskôr otvoriť v IDE prostredí z jeho umiestnenia na filesystéme počítača (Súbor\Otvoriť). IDE očakáva umiest-

nenie všetkých sketchov prináležiacich ku jednému projektu v jednom adresári. V našom prípade máme vytvorené dva adresáre, čiže dva projekty, zvlášť pre každý implementovaný algoritmus. V každom adresári sa nachádza sketch s algoritmom a sketch wifi, potrebný na inicializáciu WiFi modulu. V IDE je možné otvoriť celý projekt alebo určitý sketch z daného projektu. Pri otvorení konkrétneho sketchu IDE automaticky otvorí aj ostatné sketche z daného projektu. Otvorený projekt a všetky jeho sketche je potom možné nahráť.

Pri nahratí projektu na zariadenie nastáva automaticky kompilácia všetkých sketchov, na samotné zariadenie sa nahrá až skompilovaný kód.

B.3 Monitor sériového portu

Pri nasadení riešenia na Arduino zariadenia je možné využiť Monitor sériového portu (Nástroje\Monitor sériového portu), kde sa zobrazujú kontrolné alebo informatívne výpisy posielané z programu na sériový port využívajúc funkciu `serial.print`. Z nich je možné v prípade potreby identifikovať problém v programe, príp. miesto, v ktorom sa vykonávanie programu práve nachádza.

B.4 Nastavenie premenných

Na začiatku každého sketchu sa nachádza inicializácia premenných. Tu je možná, v prípade potreby, ich zmena editáciou sketchu v IDE prostredím s následným nahratím projektu na zariadenie.

Ďalej uvádzame premenné, ktorých zmena môže byť potrebná, a ich popis pre jednotlivé sketche a algoritmy.

B.4.1 Súbor pBFT/PoS

- unsigned int listeningPort = 2390; - číslo UDP portu použitého na komunikáciu zariadení v rámci našej práce
- IPAddress Ip1(192, 168, 200, 4); - IP adresa Arduino zariadenia č.1
- IPAddress Ip2(192, 168, 200, 6); - IP adresa Arduino zariadenia č.1
- IPAddress Ip3(192, 168, 200, 7); - IP adresa Arduino zariadenia č.1
- IPAddress Ip4(192, 168, 200, 8); - IP adresa Arduino zariadenia č.1
- IPAddress pc(192, 168, 200, 3); - IP adresa počítača, na ktorom bolo nainštalované IDE prostredie a ďalšie nástroje, ktoré sme pri práci používali.

B.4.2 Súbor wifi

- char ssid[] - SSID (meno) použitej WiFi siete
- char pass[] - heslo na pripojenie do WiFi siete

Príloha C

Technická dokumentácia

V tejto technickej dokumentácii sú opísané kľúčové časti zdrojového kódu výsledného programu, ktorý je výstupom našej práce. Postupne sú opísané jednotlivé súbory a v nich použité knižnice, premenné a funkcie.

C.1 Súbor pBFT

C.1.1 Popis knižníc použitých v súbore pBFT

SPI.h - knižnica na komunikáciu so zariadeniami pomocou Serial Peripheral Interface (SPI) Bus

WiFiUdp.h – knižnica s funkciami na prácu s UDP protokolom

SD.h – knižnica s funkciami na čítanie a zápis na SD kartu

WiFiNINA.h - knižnica na ovládanie WiFi modulu a pripájanie k WiFi

CRC32.h – knižnica na výpočet CRC32 checksum kontrolného súčtu

C.1.2 Popis premenných použitých v súbore pBFT

listeningPort – premenná typu unsigned integer, v ktorej je uložené číslo portu, ktorý používa náš program na komunikáciu do siete

packetBuffer[255] – jednorozmerné pole typu char o veľkosti 256 používané ako buffer na uloženie prijatého paketu zo siete

d[] – jednorozmerné pole typu char obsahujúce testovací reťazec používaný ako dáta pri posielaní správ do siete našim komunikačným protokolom

ULONG_MAX – premenná typu unsigned long integer, ktorá má hodnotu maximálneho možného čísla uloženého v dátovom type unsigned long, používaná na nastavenie premennej start, aby nikdy nebola splnená podmienka, v ktorej sa kontroluje, či nečakáme na odpoveď dlhšie ako 5 sekúnd

start – premenná typu unsigned long integer používaná na zistenie času, ako dlho čakáme na odpoveď

chipSelect – premenná typu integer obsahujúca číslo PINu na rozhraní použitých Arduino zariadeniach, ktorým komunikuje Arduino s SD modulom

numberOfDevices – premenná typu integer obsahujúca (počet - 1) Arduino zariadení používaných pri behu programu; v zmysle terminológie pBFT algoritmu je to (počet - 1) uzlov v sieti

numberOfResponses – premenná typu integer obsahujúca počet odpovedí, ktoré danému zariadeniu boli doručené ako odpovede na dopyt na predprípravu

waitingResponse – premenná typu integer, v ktorej je uložená informácia, či zariadenie očakáva odpoveď (1) alebo nie (0)

prepAttempt – premenná typu integer, v ktorej je uložená informácia, koľký v

poradí pokus o zaslanie dopytu predprípravy sa vykonáva

nOfResponses – premenná typu integer, v ktorej je uložený počet odpovedí, ktorý sa vrátil na požiadavku na zápis do blockchainu

IPAddress Ip1 – premenná typu IPAddress, v ktorej je uložená IP adresa prvého Arduino zariadenia

IPAddress Ip2 – premenná typu IPAddress, v ktorej je uložená IP adresa druhého Arduino zariadenia

IPAddress Ip3 – premenná typu IPAddress, v ktorej je uložená IP adresa tretieho Arduino zariadenia

IPAddress Ip4 – premenná typu IPAddress, v ktorej je uložená IP adresa štvrtého Arduino zariadenia

IPAddress myIp – premenná typu IPAddress, v ktorej je uložená IP adresa zariadenia Arduino, na ktorom bol ladený program

C.1.3 Popis funkcií v súbore pBFT

setup()

Funkcia, ktorej prítomnosť je v Arduino projekte nutná, spúšťa sa jedenkrát a to na začiatku vykonávania programu. V tejto funkcii prebehne otvorenie sériového portu, nastavenie WiFi adaptéra, otvorenie UDP portu do WiFi siete, inicializácia SD karty a iníciaľne nastavenie premenných, ktoré neboli nastavené pri ich deklarácii

loop()

Funkcia, ktorej prítomnosť je v Arduino projekte taktiež nutná, je periodicky

spúšťaná, pokiaľ je mikrokontrolér pripojený ku zdroju elektrickej energie. V tejto funkcii je overované prijatie dát po sieti, ak sú dáta prijaté, sú poslané na spracovanie do funkcie `packetReceived()`. Taktiež v tejto funkcii prebieha opakovanie dvoch pokusov o zaslanie dopytu predprípravy, ak nebol počas piatich sekúnd doručený potrebný počet odpovedí na predprípravu.

`sendPrepRequest()`

Funkcia zabezpečujúca odoslanie dopytu na predprípravu na všetky zariadenia okrem samého seba.

`sendPrepResponse(IPAddress remoteIp)`

Funkcia zabezpečujúca odoslanie odpovede na dopyt na predprípravu na zariadenie s `remoteIp` IP adresou, ktorá bola do funkcie poslaná ako parameter.

`sendData(String data)`

Funkcia zabezpečujúca odoslanie požiadavky na zápis do blockchainu na všetky zariadenia okrem samého seba. V požiadavke na zápis do blockchainu sa pošlú dáta, ktoré boli do funkcie poslané ako string parameter. Na začiatok dát je vo funkcii pridaná hlavička podľa popisu nášho komunikačného protokolu a taktiež je na koniec dát pridaný kontrolný súčet vypočítaný z dát.

`sendDataResponse(IPAddress remoteIp)`

Funkcia zabezpečujúca odoslanie odpovede na požiadavku na zápis do blockchainu na zariadenie s `remoteIp` IP adresou, ktorá bola do funkcie poslaná ako parameter.

`packetReceived(int packetSize)`

Funkcia typu integer zabezpečujúca spracovanie prijatej správy, ktorá je v jej úvode načítaná do premennej `packetBuffer`. Spracovanie prebieha v jednotlivých

častiach funkcie definovaných podmienkou na typ správy definovaný v použitom komunikačnom protokole na pozícii prvého bytu správy:

if(packetBuffer[0] == "), kde v úvodzovkách je vo funkcii uvedená hodnota typu správy.

Ak bola prijatá správa dopyt na predprípravu (*if packetBuffer[0] == '1'*), vo funkcii je volaná funkcia *sendPrepResponse* zabezpečujúca odoslanie odpovede na dopyt na predprípravu. Odpoveď je zasielaná na *remoteIp* IP adresu, ktorá bola získaná z hlavičky UDP paketu odosielateľa dopytu na predprípravu.

Ak bola prijatá správa odpoveď na dopyt na predprípravu (*if packetBuffer[0] == '2'*), inkrementuje sa počítadlo prijatých odpovedí *numberOfResponses*, následne ak má toto počítadlo potrebnú hodnotu ($2/3$ z počtu zariadení *numberOfDevices*), t.j. potrebný počet zariadení je pripravených, bude odoslaná požiadavka na zápis do blockchainu na všetky uzly zavolaním funkcie *sendData*.

Ak bola prijatá správa požiadavka na zápis do blockchainu (*if packetBuffer[0] == '3'*), prebehne kontrola prijatých dát porovnaním kontrolného súčtu CRC zo správy (*checksum*) s kontrolným súčtom, ktorý sa vypočíta vo funkcii (*checksumcalc*). Ak sa obidva kontrolné súčty rovnajú, sú dáta prijaté v správe očistené od hlavičky komunikačného protokolu poslané do funkcie *writeToLedger* na zapísanie do blockchainu a je zavolaná funkcia *sendDataResponse* na odoslanie potvrdenia o zápise do blockchainu odosielateľovi požiadavky na zápis. Ak sa kontrolné súčty nerovnajú, dáta sa do blockchainu nezapíšu a taktiež sa neodošle potvrdenie o zápise dát do blockchainu.

Ak bolo prijaté potvrdenie o zápise do blockchainu (*if packetBuffer[0] == '4'*), inkrementuje sa počítadlo prijatých odpovedí *nOfResponses*, následne ak má

toto počítaadlo potrebnú hodnotu (2/3 z počtu zariadení numberOfDevices), t.j. potrebný počet zariadení potvrdilo požiadavku na zápis do blockchainu, budú dáta zapísané do blockchainu zavolaním funkcie writeToLedger.

void writeToLedger(char *data)

Funkcia zabezpečujúca zápis dát do blockchainu. Zápis sa realizuje ako zápis do súboru na SD karte SD modulu pripojeného ku Arduino zariadeniu.

***ipToHexa(int addr)**

Funkcia typu char zabezpečujúca zmenu IP adresy v desiatkovom tvare na hexadecimálny tvar.

reverse(char* str)

Funkcia na vytvorenie zrkadlového obrazu reťazca po pároch znakov.

swap(char *a, char *b)

Funkcia na vzájomnú výmenu obsahu dvoch premenných medzi sebou.

C.2 Súbor PoS

C.2.1 Popis knižníc použitých v sketchi PoS

SPI.h - for communicating with devices using the Serial Peripheral Interface (SPI) Bus

WiFiUdp.h – knižnica s funkciami na prácu s UDP protokolom

SD.h – knižnica s funkciami na čítanie a zápis na SD kartu

WiFiNINA.h - library to use the WiFi Nina module of the above boards.

CRC32.h – knižnica na výpočet CRC32 checksum kontrolného súčtu

ArduinoSTL.h – knižnica s dôležitými C++ funkciami ako printf, scanf, atď

C.2.2 Popis premenných použitých v súbore PoS

listeningPort – premenná typu unsigned integer, v ktorej je uložené číslo portu, ktorý používa náš program na komunikáciu do siete

packetBuffer[255] – jednorozmerné pole typu char o veľkosti 256 používané ako buffer na uloženie prijatého paketu zo siete

d[] – jednorozmerné pole typu char obsahujúce testovací reťazec používaný ako dáta pri posielaní správ do siete našim komunikačným protokolom

chipSelect – premenná typu integer obsahujúca číslo PINu na rozhraní použitých Arduino zariadeniach, ktorým komunikuje Arduino s SD modulom

requestSent - premenná typu integer, do ktorej sa uloží hodnota funkcie millis() pri odoslaní požiadavky na stávk

once - premenná typu integer, ktorá ak má hodnotu 1, tak sa z vybraného zariadenia pošle požiadavka na stávk

data - premenná typu integer, ktorá ak má hodnotu 1, ak sú k dispozícii dáta, ktoré majú byť poslané na validáciu

stakes – vektor typu integer, v ktorom sú ukladané hodnoty stávk, ktoré zaslali jednotlivé zariadenia

stakesIps – vektor typu IPAddress, v ktorom sú ukladané IP adresy zariadení, ktoré zaslali stávk

waitingResponse – premenná typu integer, v ktorej je uložená informácia, či zariadenie očakáva odpoveď (1) alebo nie (0)

IPAddress Ip1 – premenná typu IPAddress, v ktorej je uložená IP adresa prvého Arduino zariadenia

IPAddress Ip2 – premenná typu IPAddress, v ktorej je uložená IP adresa druhého Arduino zariadenia

IPAddress Ip3 – premenná typu IPAddress, v ktorej je uložená IP adresa tretieho Arduino zariadenia

IPAddress Ip4 – premenná typu IPAddress, v ktorej je uložená IP adresa štvrtého Arduino zariadenia

IPAddress myIp – premenná typu IPAddress, v ktorej je uložená IP adresa zariadenia Arduino, na ktorom bol ladený program

IPAddress validator - premenná typu IPAddress, v ktorej je uložená IP adresa zariadenia zvoleného validátora

C.2.3 Popis funkcií súboru PoS

setup()

Funkcia, ktorej prítomnosť je v Arduino projekte nutná, spúšťa sa jedenkrát a to na začiatku vykonávania programu. V tejto funkcii prebehne otvorenie sériového portu, nastavenie WiFi adaptéra, otvorenie UDP portu do WiFi siete, inicializácia SD karty a iniciálne nastavenie premenných, ktoré neboli nastavené pri ich deklarácii. Taktiež je odtiaľto volaná funkcia na odoslanie požiadavky na stávkku.

loop()

Funkcia, ktorej prítomnosť je v Arduino projekte taktiež nutná, je periodicky spúšťaná, pokiaľ je mikrokontrolér pripojený ku zdroju elektrickej energie. V tejto funkcii je overované prijatie dát po sieti, ak sú dáta prijaté, sú poslané na spracovanie do funkcie `packetReceived()`. Taktiež v tejto funkcii prebieha vyhodnotenie prijatých odpovedí na stávk, ktoré boli doručené počas dvoch sekúnd od odoslania požiadavky na stávk. V cykle sa prechádza vektor `stakes` a hľadá sa najvyššia hodnota v ňom uložená, tá sa uchováva v premennej `maxStake` a k nej prináležiaca IP adresa v premennej `maxStakeIp`. Tá sa následne pošle ako parameter do funkcie `sendValidatorInfo`, ktorou sa do siete oznámi, kto je validátorom. Takisto je z funkcie `loop()` volaná funkcia na odoslanie požiadavky na zápis dát do blockchainu `sendDataForValidation`.

sendStakeRequest()

Funkcia zabezpečujúca odoslanie požiadavky na stávk na všetky zariadenia okrem samého seba.

sendStakeResponse(IPAddress remoteIp)

Funkcia zabezpečujúca odoslanie odpovede na stávk na zariadenie s `remoteIp` IP adresou, ktorá bola do funkcie poslaná ako parameter. V správe sa odosiela náhodne vygenerované číslo z intervalu 1 až 100.

sendValidatorInfo(IPAddress remoteIp)

Funkcia zabezpečujúca rozoslanie informácie o tom, kto je validátorom, na všetky zariadenia okrem samého seba. V správe sa posiela IP adresa zvoleného validátora.

sendDataForValidation(String data)

Funkcia zabezpečujúca odoslanie požiadavky na zápis do blockchainu, ktorá sa zasiela validátorovi. V požiadavke na zápis do blockchainu sa pošlú dáta, ktoré boli do funkcie poslané ako string parameter. Na začiatok dát je vo funkcii pridaná hlavička podľa popisu nášho komunikačného protokolu a taktiež je na koniec dát pridaný kontrolný súčet vypočítaný z dát.

sendValidatedData (IPAddress remoteIp)

Funkcia zabezpečujúca rozoslanie zápisu do blockchainu na všetky zariadenia okrem samého seba. V správe sa posielajú dáta zvalidované validátorom, ktoré boli do funkcie poslané ako parameter typu pole znakov.

packetReceived(int packetSize)

Funkcia zabezpečujúca spracovanie prijatej správy, ktorá je v jej úvode načítaná do premennej packetBuffer. Spracovanie prebieha v jednotlivých častiach funkcie definovaných podmienkou na typ správy definovaný v použitom komunikačnom protokole na pozícii prvého bytu správy:

if(packetBuffer[0] == "), kde v úvodzovkách je vo funkcii uvedená hodnota typu správy.

Ak bola prijatá správa požiadavka na stávku (*if packetBuffer[0] == '1'*), vo funkcii je volaná funkcia sendStakeResponse zabezpečujúca odoslanie odpovede na stávku. Odpoveď je zasielaná na remoteIp IP adresu, ktorá bola získaná z hlavičky UDP paketu odosielateľa dopytu na predprípravu.

Ak bola prijatá správa odpoveď na stávku (*if packetBuffer[0] == '2'*) a táto odpoveď bola prijatá do 2 sekúnd od odoslania požiadavky na stávku, v správe prijatá výška stávky sa uloží do vektora stakes a IP adresa odosielateľa remoteIp, získaná z hlavičky UDP paketu, sa uloží do vektora stakesIps. Ďalšie spracovanie týchto vektorov prebieha vo funkcii loop().

Ak bola prijatá správa informácia o validátorovi (if packetBuffer[0] == '3'), zo správy je extrahovaná prijatá IP adresa validátora a táto je umiestnená do premennej validator na ďalšie použitie v programe.

Ak bola prijatá správa požiadavka na zápis do blockchainu (if packetBuffer[0] == '4'), prebehne kontrola prijatých dát porovnaním kontrolného súčtu CRC zo správy (checksum4) s kontrolným súčtom, ktorý sa vypočíta vo funkcii (checksumCalc4). Takisto sa skontroluje, či sa zhoduje IP adresa z UDP hlavičky prijatej správy s IP adresou uvedenou v dátach. Ak sa zhodujú IP adresy aj obidva kontrolné súčty, sú dáta prijaté v správe očistené od hlavičky komunikačného protokolu poslané do funkcie writeToLedger na zapísanie do blockchainu a je zavolaná funkcia sendValidatedData na rozoslanie zápisu do blockchainu na všetky uzly siete, okrem samého seba. Ak sa kontrolné súčty nerovnajú, dáta sa do blockchainu nezapíšu a taktiež sa nerozošle správa o zápise do blockchainu.

Ak bola prijatá správa o zápise do blockchainu (if packetBuffer[0] == '5'), prebehne kontrola prijatých dát porovnaním kontrolného súčtu CRC zo správy (checksum5) s kontrolným súčtom, ktorý sa vypočíta vo funkcii (checksumCalc5). Ak sa obidva kontrolné súčty zhodujú, sú dáta prijaté v správe očistené od hlavičky komunikačného protokolu poslané do funkcie writeToLedger na zapísanie do blockchainu.

writeToLedger(char *data)

Funkcia zabezpečujúca zápis dát do blockchainu. Zápis sa realizuje ako zápis do súboru na SD karte SD modulu pripojeného ku Arduino zariadeniu.

***ipToHexa(int addr)** Funkcia typu char na zabezpečujúca zmenu IP adresy v desiatkovom tvare na hexadecimálny tvar.

reverse(char* str)

Funkcia na vytvorenie zrkadlového obrazu reťazca po pároch znakov.

swap(char *a, char *b)

Funkcia na vzájomnú výmenu obsahu dvoch premenných medzi sebou.

C.3 Súbor wifi

Tento súbor je pri oboch algoritmoch rovnaký, preto jeho popis uvedieme len raz.

C.3.1 Popis premenných použitých v súbore wifi

ssid[] – pole typu char, obsahujúce SSID wifi siete, ku ktorej sa bude zariadenie pripájať

pass[] – pole typu char, obsahujúce heslo na pripojenie ku wifi sieti **ssid[]** **status** – premenná typu integer obsahujúca stav wifi adaptéra

C.3.2 Popis funkcií použitých v súbore wifi

setupWiFi()

Funkcia, v ktorej sa inicializuje WiFi modul daného Arduino zariadenia. Najskôr je overená prítomnosť WiFi modulu, následne je overená verzia jeho firmvéru. Po inicializácii je modul pripojený ku určenej WiFi sieti.

printWiFiData()

Funkcia na výpis sieťových nastavení, ktoré boli pripojenou WiFi sieťou pridelené WiFi sieťovému adaptéru daného Arduino zariadenia.

printCurrentNet()

Táto funkcia vypíše informácie o WiFi sieti, ku ktorej sa WiFi adaptér daného Arduino zariadenia pripojil (SSID, MAC adresu routra, atď).

printMacAddress(byte mac[])

Táto funkcia vypíše MAC adresu WiFi adaptéra daného Arduino zariadenia.

Príloha D

Opis digitálnej časti práce

Evidenčné číslo práce v informačnom systéme: FIIT-100241-79970

Obsah digitálnej časti práce (archív ZIP):

Priečinok

Opis

\BP_prilohy_digital_Zelenak

\pBFT

pBFT.ino

UpBFT algoritmus

wifi.ino

WiFi modul

\PoS

PoS.ino

PoS algortimus

wifi.ino

WiFi modul