

České
vysoké učení
technické
v Praze

Fakulta Elektrotechnická
Katedra radioelektroniky

Bakalářská práce

Interaktivní hra využívající IoT prostředky

Jan Závorka

Praha, Květen 2019

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Závorka** Jméno: **Jan** Osobní číslo: **466133**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra radioelektroniky**
Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Interaktivní hra využívající IoT prostředky

Název bakalářské práce anglicky:

Interactive Game Based on IoT Technology

Pokyny pro vypracování:

Cílem práce je návrh a implementace platformy, která bude demonstrovat možnosti využití levných a energeticky nenáročných zařízení v IoT sítích. Aplikace bude mít podobu interaktivní hry pro více hráčů. Koncová zařízení budou založena na kitu Arduino (nebo podobném) s potřebnými periferiemi (dotykový displej) a budou pomocí protokolu navrženém studentem komunikovat s centrální jednotkou.

Seznam doporučené literatury:

- [1] WAHER, Peter. Learning internet of things. Packt Publishing Ltd, 2015.
[2] IEEE INTERNET INITIATIVE, et al. Towards a definition of the Internet of Things (IoT). Revision-1, on-line: http://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Stanislav Vitek, Ph.D., katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **14.02.2019**

Termín odevzdání bakalářské práce: **24.05.2019**

Platnost zadání bakalářské práce: **20.09.2020**

Ing. Stanislav Vitek, Ph.D.
podpis vedoucí(ho) práce

prof. Mgr. Petr Páta, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Obsah

1	Úvod	1
2	Hardware	3
2.1	Zařízení typu server	3
2.2	Zařízení typu klient	5
3	Popis softwaru	7
3.1	Knihovny	7
3.2	Konfigurační hodnoty	7
3.2.1	Konfigurace serveru	8
3.2.2	Konfigurace klienta	9
3.3	Komunikace server - klient	12
3.4	Komunikace klient - server	12
3.5	Sestavení spojení	14
3.6	Průběh hry	14
4	Oževení a užívání	15
5	Conclusions and future research	17
	Reference	19
6	Seznam použitého softwaru	20
A	Appendix A	21

1 Úvod

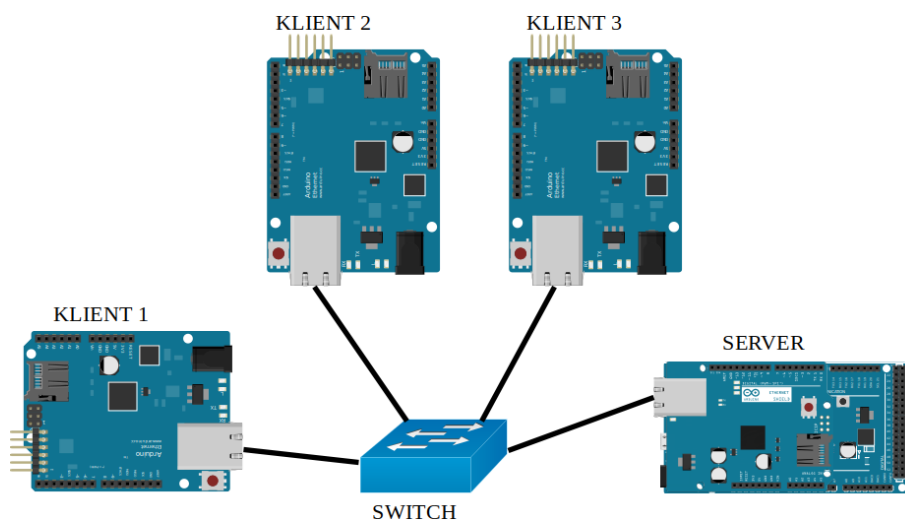
Cílem práce bylo vytvořit zařízení, které by demonstrovalo využití jednodeskových počítačů v IoT sítích. Pro realizaci byla zvolena platforma Arduino, především kvůli své rozšířenosti a dostupnosti doplňujících periférií. Samotné zařízení má funkci hry pro více hráčů, kde každý hráč má mikropočítač s dotykovým displejem a celá hra je řízena serverem (také založen na platformě Arduino), komunikace pak probíhá po Ethernetové síti. Pro toto zařízení byla implementována hra piškvorky z důvodu nízké náročnosti na výkon mikropočítače a dobré názornosti co se týče samotné komunikace.

Nedokončeno, doplnit !!!

2 Hardware

Celá platforma se skládá z jednoho řídicího prvku (viz. kapitola 2.1) a třech (maximální počet klientů je hardwarově limitován na čtyři) klientských zařízení (viz. kapitola 2.2). Datové spojení je realizováno hvězdicovou topologií (schéma na obrázku 1). Jako centrální prvek byl použit switch D-Link DGS-105. Celá demonstrační sestava se pak skládá z:

- 1x switch D-Link DGS-105
- 1x server s Arduino DUE
- 3x klient s Arduino Ethernet a dotykovým displejem
- 4x propojovací UTP kabel
- 1x napájecí adaptér pro switch (5 V/1 A součástí balení)
- 4x napájecí adaptér 12 V/1500 mA



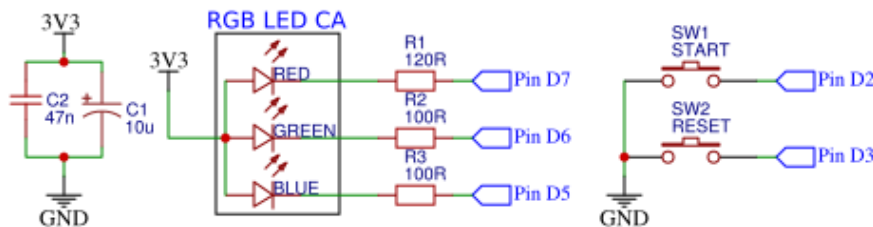
Obrázek 1: Schéma zapojení jednotlivých zařízení do sítě, zdroj [1, 2, 3, 4]

2.1 Zařízení typu server

Zařízení je založeno na desce *Arduino Due*, které obsahuje mikrokontrolér Atmel SAM3X8E ARM Cortex-M3 s 512 kB flash paměti a nabízí dostatečný výkon pro správné fungování serveru. Původní varianta totiž počítala s nasazením desky *Arduino Ethernet* i jako serveru. To se ovšem vzhledem k omezeným prostředkům ukázalo jako problematické, proto byla zvolena právě deska *Arduino Due*.

Pro připojení do sítě je použit Ethernetový shield s čipem Wiznet W5100, který je přímo napojen na Arduino. Tímto je dána ona limitace maximálně na 4 hráče, protože dle datasheetu výrobce čipu [6] je maximální počet spojení právě čtyři.

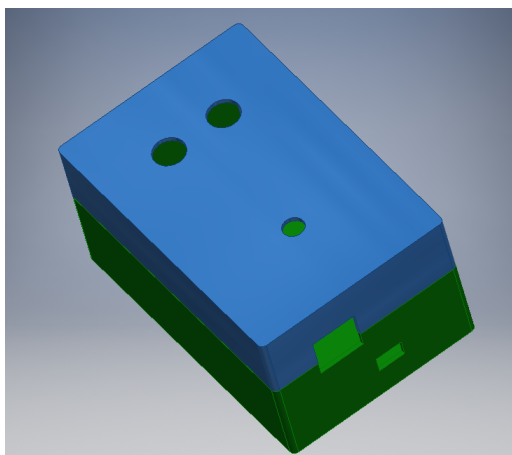
Pro pohodlné ovládání jsou k serveru připojena dvě tlačítka a jedna barevná svítivá dioda. Význam jednotlivých stavů svítivé diody a funkce tlačítek je popsána v kapitole ???. Propojení těchto periférií s Arduinem je realizováno pomocí jednostrané DPS, schéma zapojení je pak na obrázku 2.



Obrázek 2: Schéma zapojení periférií u serveru

Napájení je řešeno externím adaptérem, dle stránek výrobce [5] je možné pužit napětí 6 - 16 V (využívá se interní stabilizátor), přízemž odběr je kolem 140 mA při napájení 12 V. V případě, že je pro ovládání použita sériová linka (server je připojen USB kabelem k počítači, ovládání tímto způsobem je popsáno v kapitole ???), postačuje napájení dodané přes USB kabel a není potřeba připojovat externí napájecí zdroj.

Celé zařízení je pak umístěno v krabičce jejíž návrh je na obrázku 3 a realizace na obrázku 4. Krabička byla navrhnutá v programu Autodesk Inventor Professional 2019 Student Edition a realizovaná 3D tiskem na tiskárně Original Prusa i3 MK3S. Krabička je osazena červeným a zeleným tlačítkem, barevnou svítivou 5 mm diodou (se společnou anodou) a souosým napájecím konektorem 5,5x2,1 mm. Pro upevnění Arduina jsou použity šrouby M2,5x10 a závitové vložky M2,5x6 vtavené do připravených otvorů v krabičce. Arduino deska sice nabízí montážní otvory o průměru 3 mm, ale vzhledem k rozložení součástek zde není dostatek místa pro hlavu šroubu M3.



Obrázek 3: Návrh krabičky pro server



Obrázek 4: Zkompletovaná krabička pro server

2.2 Zařízení typu klient

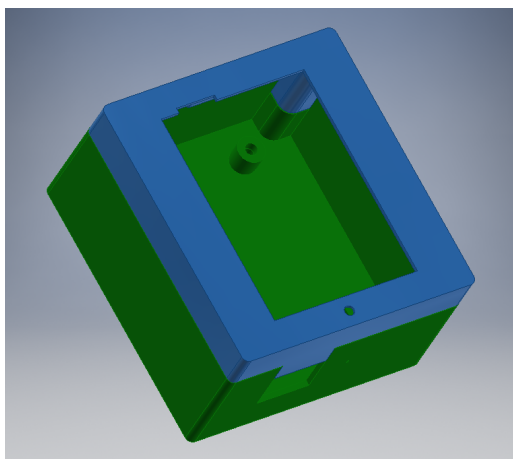
Zařízení je založeno na desce Arduino Ethernet, která je vybavena mikrokontrolérem ATmega328 s 32 kB flash paměti. Tato deska byla zvolena především kvůli tomu, že má vestavěný ethernetový kontrolér, který tak nezabírá piny pro připojení shieldu s displejem. Malá flash paměť se však během vývoje ukázala jako značně limitující, protože při nahrání všech potřebných knihoven (popsáno v kapitole ??) zůstalo k dispozici 30 % programové paměti. I z tohoto důvodu byla zvolena jako hra piškvorky, která není programově příliš složitá a také bylo nutné vynechat složitější menu například s nastavením barvy nebo změny IP adresy serveru (to se nyní musí provádět změnou v kódu a přeprogramováním Arduina, více v kapitole 3.2.2).

Jak už bylo zmíněno výše, tato deska má vestavěný ethernet kontrolér WIZnet W5100. U klienta není maximální počet spojení limitující (klient drží pouze jedno spojení se serverem), jedinou nevýhodou kontroléru W5100 tak zůstává, že neobsahuje registr, ve kterém je uložena informace o fyzickém připojení ethernetového kabelu k desce [6]. Tím je zkomplikována detekce připojení a odpojení kabelu.

Pro interakci s uživatelem je klient vybaven 2,4" barevným TFT LCD displej s rozlišením 320x240 pixelů s rezistivní dotykovou plochou. Displej je vybaven řadičem SUM74HC245T. Vzhledem k rozměrům (výšce) RJ-45 konektoru, který je umístěn na desce, je nutné pro správné připojení použít lištu oboustrannými kolíky o délce kolíku minimálně 15 mm. Protože u displejů použitých v tomto projektu byly kolíky připájeny už od výrobce, byla dodatečně vyrobená patice s dutinkové lišty a lišty s oboustrannými kolíky.

Pro napájení byl zvolen externí napájecí adaptér, avšak připojení na integrovaný stabilizátor není možné, protože displej při 5 V odebírá přibližně 400 mA, což integrovaný stabilizátor nedokáže poskytnout (vlivem velého ztrátového výkonu dochází k jeho značnému zahřívání). Napájení přímo napětím 5 V není také příliš vhodné, protože vlivem například ztrát přírodních vodičů může dojít ke kolísání napětí, tím dojde i k pohybu reference AD převodníku připojenému k dotykové vrstvě displeje a tak může docházet k nesprávnému vyhodnocení stisku (může se lišit reálné místo stisku od toho, které vyhodnotil mikrokontrolér). Jako nejlepší varianta se ukázalo použití modulu se snižujícím DC-DC měničem. Modul obsahuje spínací regulátor MP1584 a dle dodavatele je schopen pracovat s napětím 6 - 25 V (při výstupním napětí 5 V) a dodat proud až 1,5 A, což je pro tuto aplikaci dostačující.

Stejně jako v případě serveru je celé zařízení umístěno ve vytištěné krabici, návrh a realizovaná krabice jsou na obrázcích 5, 6. Principi uchycení Arduina je stejný jako v případě serveru, pro napájení je opět osazen souosý napájecí konektor 5,5x2,1 mm. Dále je z boku výřez pro konektor RJ-45 pro připojení do ethernetové sítě a na vrchu se nachází výřez pro displej vedle kterého se je umístěn otvor pro přístup k resetovacímu tlačítku.



Obrázek 5: Návrh krabičky pro klienta



Obrázek 6: Zkompletovaná krabička pro klienta

3 Popis softwaru

3.1 Knihovny

Pro realizaci byly použity následující knihovny:

- *Ethernet library*: knihovna slouží pro připojení Arduina k internetu. Jak je uvedeno na webových stránkách [7] jsou podporovány desky a shieldy založené na kontrolérech W5100, W5200, W5500. Tato knihovna je použita jak u serveru tak i u klientů. U serveru připojený ethernet shield obsadí piny 10, 11, 12, 13. Komunikace Arduina se shieldem (ethernet kontrolérem) probíhá po SPI sběrnici.
- *MCUFRIEND_kbv library*: knihovna slouží k ovládání displeje u klienta. Pro správnou funkci je nutné, jak se zmiňuje autor na domovské stránce knihovny [8], je nutné mít k dispozici také knihovnu *Adafruit_GFX* [9]. Pro použití displeje se nejdříve vytvoří instance třídy *UTFTGLUE*, kde je nutné správně definovat piny pro daný shield, v tomto případě `UTFTGLUE LCD(0x0154,A2,A1,A3,A4,A0);`. Pro řízení displeje jsou pak volány příslušné metody formou `LCD.metoda(parametry);`, například pro vyplnění celého displeje černou `LCD.clrScr();`. Pomocí ukázkového zdrojového kódu z této knihovny (*examples/TouchScreen_calibr_kbv*) byly získány kalibrační hodnoty pro dotykovou plochu displeje.
- *Adafruit_TouchScreen*: slouží k zaznamenání hodnot ze čtyřvodičové dotykové plochy. Stejně jako u knihovny pro displej je nejdříve nutné vytvořit instanci dané třídy *TouchScreen* v tomto případě tak `TouchScreen Touch(XP, YP, XM, YM, 300);`, kde *XP*, *YP*, *XM*, *YM* jsou piny, na které je vyvedená dotyková plocha (pro použitý shield jsou to: *XP* = 6, *YP* = A1, *XM* = A2, *YM* = 7). Hodnota 300 by pak dle popisu knihovny [10] označuje elektrický odpor dotykové plochy *X* (měřeno multimetrem mezi piny *XP* a *XM*). Pro použité shiledy je hodnota přibližně 300 Ω. Dále je třeba určit minimální a maximální tlak pro vyhodnocení dotyku pomocí `#define MINPRESSURE` a `#define MAXPRESSURE`. Vyhovující jsou hodnoty uvedené v ukázkových kódech pro knihovnu a to konkrétně 10 pro `MINPRESSURE` a 1000 pro `MAXPRESSURE`.
- *SimpleTimer Library for Arduino*: jednoduchá knihovna, která slouží k řízení určitých časových událostí (kde není třeba velká přesnost), například pro zobrazení a skrytí chybových hlášek, blikání indikační svítivé diody u serveru. Dle webových stránek [11] je knihovna založená na funkci `millis();` (vrací počet milisekund od začátku běhu programu) a nevyužívá přerušení ani hardwareový timer.

3.2 Konfigurační hodnoty

V této kapitole jsou popsány jednotlivé konfigurační hodnoty, která by mohlo být třeba upravit pro správnou funkci celého zařízení. Změny se vždy provádí přímo ve zdrojovém kódu, vždy pouze v hlavní souboru: *piskvorky_MP_client.ino* pro klienta a *piskvorky_MP_server.ino* pro server. Jak už bylo zmíněno v kapitole 2.2, nebylo možné, kvůli malé paměti pro program, implementovat menu, proto je nutné změny

provádět přímo ve zdrojovém kódu a dané zařízení pak přeprogramovat. V případě serveru je k dispozici microUSB konektor, který je dostupný skrze obdélníkový otvor na boku zařízení. V případě klienta je nutné demontovat víko, odpojit napájecí modul (v modré bužírce) a připojit Arduino k počítači pomocí převodníku USB - UART.

Jednotlivé konfigurační bloky jsou vždy ohraničeny komentáři mezi nimiž se nacházejí jednotlivé proměnné a popis jejich funkce a omezení hodnot. Níže je uveden příklad ohraničení bloku pro nastavení sítě (IP adresa a port):

```
/* ----- KONFIGURACE - nastavení sítě -----*/
```

```
/* ----- KONEC - nastavení sítě -----*/
```

3.2.1 Konfigurace serveru

V prvním případě je nutné vybrat síťový režim, v případě *ETHMODE_STATIC* je použita IP adresa, která je uložena v proměnné *serverAddress*. V případě volby *ETHMODE_DHCP* je adresa přiřazena DHCP serverem a je nutné dodržet aby se přiřazená adresa neměnila a byla zároveň nastavená u jednotlivých klientů.

Dále je třeba nastavit příslušnou MAC adresu, v případě použitého nebyla přiřazena výrobcem, takže je nutné nějakou zvolit. Vzhledem k tomu, že zařízení se provozuje na samostatné lokální síti, byla MAC adresa vybrána náhodně, v případě, že v síti jsou další zařízení, je možné vybrat MAC adresu například podle postupu popsáno v [13]. Vezme se MAC adresa nějaké zařízení v síti a poslední bajt se zvětší o jedna a takto vzniklá MAC adresa se přiřadí shieldu. Vybraná MAC adresa je v poli *mac*.

Jako poslední je potřeba přiřadit port, na kterém budou zařízení komunikovat. Dle normy [12] je možné zvolit jakoukoliv dynamický port (49152-65535), protože tyto porty nebudou nikdy přiřazeny žádné službě. Port je uložen v proměnné *localPort* a stejný port musí být nastaven i u klientů.

```
/* !!! ----- KONFIGURACE - nastaveni ethernetu ----- !!! */
#define ETHMODE_STATIC //Varianty: ETHMODE_DHCP; ETHMODE_STATIC
byte mac[] = {
    0xDE, 0xAD, 0xBE, 0xEE, 0xFE, 0x00
};
IPAddress serverAddress(10,0,0,8);
unsigned int localPort = 3333;
/* !!! ----- KONEC nastaveni ethernetu ----- !!! */
```

Blok konfigurace ethernet shieldu pro server

Jak je popsáno v kapitole 2.1, jsou k serveru připojena dvě tlačítka a jedna barevná svítivá dioda. V případě, že je nutné tyto periferie připojit jinak, je možné změnit čísla pinů v konfiguračním bloku *nastavení pinů a LED*. Tlačítka využívají interní pull-up rezistory a jejich zapojení je uvedeno na obrázku 2.

```
/* !!! ----- KONFIGURACE - nastaveni pinu a LED ----- !!! */
//Piny tlacitka start a reset
#define startPIN 2
#define resetPIN 3
```

```
//
//Piny na kterých jsou připojeny jednotlivé barvy LED, musí podporovat
//PWM
#define LED_red 7
#define LED_green 5
#define LED_blue 6
//
//Nastavení maximálního jasu všech LED, interval <0; 100>
const byte LED_br = 40;
/* !!! ----- KONEC nastavení pinů a LED ----- !!! */
```

Blok konfigurace pinů pro svítivou diodu a tlačítka

Jako poslední je možné nastavit jakou barvu budou mít jednotliví klienti. V ukázkové případě jsou barvy přiřazeny z výběru, který je uveden v kódu, ale je možné také definovat vlastní ve formátu RGB565 (16-bit barva).

```
/* !!! ----- KONFIGURACE - nastavení barev hráčů ----- !!!
*/
#define PLAYER1COLOR RED //0xF800
#define PLAYER2COLOR GREEN //0x07E0
#define PLAYER3COLOR PURPLE //0x780F
#define PLAYER4COLOR GREENYELLOW //0xAFE5
#define PLAYER5COLOR OLIVE //0x7BE0
/* !!! ----- KONEC nastavení barev hráčů ----- !!! */
```

Blok konfigurace jednotlivých barev pro klienty/hráče

3.2.2 Konfigurace klienta

Klientů se v celé sestavě nachází několik (v tomto případě tři) a liší se pouze určitým nastavením (IP adresa, MAC adresa, kalibrační hodnoty displeje) proto jsou vytvořeny profily pro jednotlivé klienty (softwarově omezeno na pět). Každému profilu se nastaví požadované parametry a při nahrávání programu do Arduina se pak profily mění pomocí `#define CLIENTx`, kde `x` je číslo jednotlivých klientů/profilů. Pokud není zvolen žádný profil, použítí je defaultní hodnoty a uživatel je o tom informován při překladu pomocí direktivy `#warning` (zobrazí se hlášení, ale překlad neukončí).

```
/* ----- KONFIGURACE - nastavení schemat ----- */
#define CLIENT1 //Vyber profilu pro klienta
/* ----- KONEC - nastavení schemat ----- */
```

Blok výběru profilu pro klienta

V této části se přiřazují jednotlivé MAC adresy daným profilům. Výhodou oficiálních Arduino Ethernet desek je, že mají MAC adresu přidělenou výrobcem a je možné jí nalézt na bílém štítku ze spodní strany.

```
#ifndef CLIENT1
//Klient 1
byte mac[] = {
```

```

    0x90, 0xA2, 0xDA, 0x11, 0x08, 0x18
};
#elif defined(CLIENT2)
//Klient 2
byte mac[] = {
    0x90, 0xA2, 0xDA, 0x11, 0x09, 0x78
};
#elif defined(CLIENT3)
//Klient 3
byte mac[] = {
    0x90, 0xA2, 0xDA, 0x11, 0x08, 0xA0
};
#elif defined(CLIENT4)
//Klient 4
byte mac[] = {
    0xDE, 0xAD, 0xBE, 0xEE, 0xFE, 0xBD
};
#elif defined(CLIENT5)
//Klient 5
byte mac[] = {
    0xDE, 0xAD, 0xBE, 0xEE, 0xFE, 0xAD
};
#else
//Defaultni hodnoty
#warning "Je pouzita defaultni MAC adresa"
byte mac [] = {
    0xDE, 0xAD, 0xBE, 0xEE, 0xFE, 0xAB
}
#endif
/* ----- KONEC - nastaveni mac adres ----- */

```

Blok přiřazení MAC adres jednotlivým klientským profilům

Stejně jako v případě serveru i zde je možné vybrat ze dvou režimů sítě a to *ETHMODE_STATIC* a *ETHMODE_DHCP*. V případě volby *ETHMODE_DHCP* není na adresu přiřazenou DHCP serverem kladen žádný zvláštní nárok. V případě použití *ETHMODE_STATIC* je ještě nutné dodefinovat IP adresy pro jednotlivé klienty. Opět lze s výhodou využít profilů, pokud se použije defaultní, je o tom uživatel opět při překladu informován. Kromě toho je ještě nutné doplnit IP adresu serveru, ke kterému se budou klienti připojovat. Ta byla nastavena při konfiguraci serveru (kapitola 3.2.1) stejně tak jako port, který je také nutné zvolit stejný.

```

/* ----- KONFIGURACE - nastaveni site ----- */
#define ETHMODE_STATIC //Varianty: ETHMODE_DHCP; ETHMODE_STATIC

#ifdef ETHMODE_STATIC
#ifdef CLIENT1
IPAddress clientAddress(10,0,0,138);

#elif defined(CLIENT2)
IPAddress clientAddress(10,0,0,139);

#elif defined(CLIENT3)
IPAddress clientAddress(10,0,0,140);

```



```

#elif defined(CLIENT4)
IPAddress clientAddress(10,0,0,141);

#elif defined(CLIENT5)
IPAddress clientAddress(10,0,0,142);
#else
//Defaultní nastavení
IPAddress clientAddress(10,0,0,100);
#warning "Pozor - je použito defaultní nastavení IP, pro každého
         klienta nutno změnit"
#endif
#endif
IPAddress serverAddress(10,0,0,8); //Nastavení IP adresy serveru
unsigned int localPort = 3333;      //Port
/* ----- KONEC - nastavení site ----- */

```

Blok nastavení síťového režimu a IP adres

Jako poslední je nutné nastavit kalibrační hodnoty dotykové plochy displeje. Nejjednodušší způsob jejich získání je použít ukázkový program v knihovně *MCUFRIEND_kbv library*, který lze nalézt v (*examples/TouchScreen_calibr_kbv*). V ukázkovém programu je třeba upravit podle použitého shledu nastavení pinů. Poté stačí postupovat podle pokynů na displeji a výsledek se vypíše na sériovou linku (lze použít integrovaný v Arduino IDE - *Tools->Serial Monitor*). Krom kalibračních hodnot je také nutné doplnit orientaci displeje (*#define TOUCH_LANDSCAPE* nebo *#define TOUCH_PORTRAIT*), v případě, že nebude zvolena ani jedna možná, překlad budou ukončen a vypsána chybová hláška (direktiva *#error*). I v tomto případě lze s výhodou využít profilů.

```

/* ----- KONFIGURACE - kalibrace displeje ----- */
#ifdef CLIENT1
//Klient 1
#define TOUCH_XMIN 221
#define TOUCH_XMAX 950
#define TOUCH_YMIN 200
#define TOUCH_YMAX 950
#define TOUCH_LANDSCAPE
#elif defined(CLIENT2)
//Klient 2
#define TOUCH_XMIN 233
#define TOUCH_XMAX 937
#define TOUCH_YMIN 210
#define TOUCH_YMAX 910
#define TOUCH_LANDSCAPE
#elif defined(CLIENT3)
//Klient 3
#define TOUCH_XMIN 230
#define TOUCH_XMAX 960
#define TOUCH_YMIN 220
#define TOUCH_YMAX 920
#define TOUCH_LANDSCAPE
#elif defined(CLIENT4)
//Klient 4

```

```

#define TOUCH_XMIN 0
#define TOUCH_XMAX 100
#define TOUCH_YMIN 0
#define TOUCH_YMAX 100
#elif defined(CLIENT5)
//Klient 5
#define TOUCH_XMIN 0
#define TOUCH_XMAX 100
#define TOUCH_YMIN 0
#define TOUCH_YMAX 100
#else
//Defaultní nastavení, je nutné upravit
#define TOUCH_XMIN 0
#define TOUCH_XMAX 100
#define TOUCH_YMIN 0
#define TOUCH_YMAX 100
#endif
/* ----- KONEC - kalibrace displeje ----- */

```

Blok nastavení kalibračních hodnot dotykové plochy

3.3 Komunikace server - klient

Server komunikuje s klienty pomocí pole *board*. Jedná se o jednorozměrné pole typu *byte* o velikosti 136 (velikost musí být dělitelná osmi). V tomto poli jsou vyplněné všechny důležité informace o stavu hry i jednotlivých hráčích, význam jednotlivých hodnot v poli *board* je popsán v tabulce 1.

Při odesílání je toto pole rozděleno na části po osmi bajtech. Každá tato část je vybavena pořadovým číslem a dvoubajtovým kontrolním součtem (v něm je zahrnuto i pořadové číslo) a následně odeslána připojeným klientům.

Klienti postupně přijímají všechny části a v případě bezchybného příjmu data vyhodnotí. V případě, že klient na základě kontrolního součtu identifikoval chybnou část, odešle podle pravidel komunikace klient->server (popsáno v kapitole 3.4). Server pak v případě přijetí požadavku odešle danému klientovi vyžadovanou část pole.

3.4 Komunikace klient - server

Klient komunikuje se serverem, až na výjimku při sestavení spojení, prostřednictvím dvoubajtového pole. Popis jednotlivých částí pole je v tabulce 2. Pro kontrolu přenosu se každá zpráva odešle třikrát. Na straně serveru se pak porovnají dvě ze tří přijatých zpráv a ostatní přijatá data se zahodí. Klient může poslat data maximálně jednou za sekundu (posláním dat se rozumí odeslání tří stejných zpráv).

Pokud při porovnání server zjistí, že přijaté zprávy nejsou stejné, odešle danému klientovi znovu herní desku. V případě, že byl daný hráč na tahu a přenos požadavku na vyplnění pole se nezdařil, je mu prostřednictvím znovuodeslání pole znovu aktivován tah.

Tabulka 1: Rozložení pole *board* pro přenos dat a řízení hry mezi serverem a klientem

Index	Hodnoty	Popis
0-89		Každý index odpovídá jednomu čtverečku na herní desce piškvorek
	0	Pole je prázdné (neobsazené)
	1-5	Obsazeno některým hráčem/klientem
90		Přenos řídicí informace pomocí kódu
	0	nic nedělat
	1	Vše je OK, překreslit obrazovku
	3	Příprava nové hry, zobrazit úvodní obrazovku
	9	Informace pro hráč odpojen, že bude odpojen
	100	Hra skončila remízou
	10x	Hodnota podle hráče, který vyhrál: 101-105 ('x' je číslo hráče)
	20x	Problémy/odpojení s daného hráče: 201-205 ('x' je číslo hráče)
91	1-5	Číslo hráče, který je na tahu, pokud je 0, nikdo nehraje
93	0-89	Počet odehraných kol (vyplňuje server)
95-96	Kód barvy	Barva hráče 1
97-98		Barva hráče 2
99-100		Barva hráče 3
101-102		Barva hráče 4
103-104		Barva hráče 5
105-108	IPv4 adresa	IP adresa hráče 1
109-112		IP adresa hráče 2
113-116		IP adresa hráče 3
117-120		IP adresa hráče 4
121-124		IP adresa hráče 5

V případě, že klient přijal od serveru chybná data, odešle požadavek ve tvaru $\{20, x\}$, kde x je číslo chybně přijaté části pole.

Tabulka 2: Rozložení packetu pro komunikaci klient->server

Index	Hodnoty	Popis
0	10	Jedná se o informaci, že další přenesený bajt bude číslo vyplněného pole (čtverečku) pole <i>boardu</i> ,
	20	Žádost klienta o posláni dané části herního pole (board), v další bajtu je číslo dané části <i>boardu</i>
1	0-89, 0-16	Podle hodnoty předchozího bajtu: index vyplněného pole nebo pořadí packetu, který má být poslán znovu

3.5 Sestavení spojení

Rozpracováno, nepublikováno !!!

3.6 Průběh hry

Rozpracováno, nepublikováno !!!

4 Oževení a užívání

Rozpracováno, nepublikováno !!!

5 Conclusions and future research

Reference

- [1] Fritzing Parts: Arduino_Ethernet. In: *Paulvollmer* [online]. c2018, 2013 [cit. 2019-05-05]. Dostupné z: https://paulvollmer.net/FritzingParts/parts/Arduino_Ethernet.html
- [2] Using the SD library to create and remove files on a SD card. In: *Arduino* [online]. c2019, 2015/08/18 [cit. 2019-05-05]. Dostupné z: <https://www.arduino.cc/en/tutorial/files>
- [3] Lan Switch Icon #83279. In: *Free Icons Library* [online]. c2018-2019 [cit. 2019-05-05]. Dostupné z: <http://chittagongit.com/icon/lan-switch-icon-25.html>
- [4] Fritzing Parts: Arduino_DUE_V02b. In: *Paulvollmer* [online]. c2018, 2013 [cit. 2019-05-05]. Dostupné z: https://paulvollmer.net/FritzingParts/parts/Arduino_DUE_V02b.html
- [5] Arduino store: ARDUINO DUE. *Arduino* [online]. c2019 [cit. 2019-05-05]. Dostupné z: <https://store.arduino.cc/due>
- [6] W5100 Datasheet. *WIZnet* [online]. c2009-2011, 8.1.2016 [cit. 2019-05-05]. Dostupné z: https://www.wiznet.io/wp-content/uploads/wiznethome/Chip/W5100/Document/W5100_Datasheet_v1.2.7.pdf
- [7] Arduino: Ethernet library. *Arduino* [online]. c2019 [cit. 2018-12-18]. Dostupné z: <https://www.arduino.cc/en/Reference/Ethernet>
- [8] PRENTICEDAVID. MCUFRIEND_kbv library. In: *Github* [online]. c2019 [cit. 2019-02-05]. Dostupné z: https://github.com/prenticedavid/MCUFRIEND_kbv
- [9] BURGESS, Phillip. Adafruit GFX Graphics Library: Overview. *Adafruit* [online]. 29.6.2012 [cit. 2019-05-08]. Dostupné z: <https://learn.adafruit.com/adafruit-gfx-graphics-library/overview>
- [10] ADAFRUIT. Adafruit_TouchScreen library. *Github* [online]. c2019 [cit. 2019-02-05]. Dostupné z: https://github.com/adafruit/Adafruit_TouchScreen
- [11] ROMANI, Marcello. SimpleTimer Library for Arduino. *Arduino* [online]. c2019 [cit. 2019-05-08]. Dostupné z: <https://playground.arduino.cc/Code/SimpleTimer/>
- [12] IETF [INTERNET ENGINEERING TASK FORCE], IANA [INTERNET ASSIGNED NUMBERS AUTHORITY]. *Service Name and Port Number Procedures: rfc6335, BCP165*. 2011, 33 s. ISSN: 2070-1721. Dostupné také z: <https://tools.ietf.org/html/rfc6335>
- [13] MALÝ, Martin. Arduino: webový server i klient do ruky. *Root.cz* [online]. 27. 7. 2010 [cit. 2019-02-06]. Dostupné z: <https://www.root.cz/clanky/arduino-webovy-server-i-klient-do-ruky/>

6 Seznam použitého softwaru

1. T_EXmaker, T_EXLive
2. [Tables Generator](#)
3. [Citace PRO](#)
4. [Photopea](#)
5. [Autodesk Inventor Professional 2019 Student Edition](#)
6. [Ardunino IDE](#)
7. [Atom IDE](#)
8. [EasyEDA](#)
9. Linux Mint 19.1 Cinnamon 64-bit
10. Windows 10 Home 64-bit

A Appendix A

