

Dokumentace k semestrální práci  
z předmětu KIV/OS  
Síťový monitoring

Jiří Winter

20.12.2024

# 1. Zadání

Cílem této semestrální práce je rozšířit operační systém pro zařízení Raspberry Pi Zero o podporu síťové komunikace v jádře systému a monitorovací program, který bude komunikovat s ostatními zařízeními. Síťová komunikace může být realizována na libovolném rozhraní dostupné na zařízení Raspberry Pi Zero kromě rozhraní UART. Na UART bude každé zařízení vypisovat svůj aktuální stav. Spolu s implementací podpory síťové komunikace je potřeba navrhnout síťový protokol, který bude efektivně využívat komunikační kanál.

Monitorovací program bude sloužit ke sběru dat ze senzorů glukózy a jejich monitoringu. Program bude pracovat buď v roli „master“, kdy data sbírá a upozorňuje na nebezpečné hodnoty, nebo v roli „slave“, kdy posílá naměřená data. Jednotlivá zařízení se tak nejprve musí dohodnout, kdo bude master a kdo slave, a následně vykonávat svou funkci.

## 2. Analýza

Pro implementaci podpory síťové komunikace v jádře operačního systému je potřeba vybrat komunikační rozhraní, implementovat jeho ovladač a zakomponovat jeho ovládání do jádra, aby jej uživatel mohl využívat. Vybrané rozhraní je I2C, které využívá k propojení 2 kabely – SDA (data) a SCL (clock). Ke komunikaci je potřeba jedno „master“ zařízení a alespoň jedno „slave“ zařízení. Master zařízení může být zapojeno buď na pinech 0 (SDA) a 1 (SCL) nebo na pinech 2 (SDA) a 3 (SCL) a slave zařízení musí být zapojeno na pinech 18 (SDA) a 19 (SCL). Realizace komunikace z pohledu master a slave se liší, a to je potřeba zohlednit v implementaci ovladače. Zařízení tak musí obsahovat ovladač pro I2C master i pro I2C slave.

Zakomponování I2C ovladače do jádra systému tak, aby jej mohl uživatel využívat i v uživatelském prostředí je pomocí filesystemu. Pro I2C je definován tzv. filesystem driver, který umožní uživateli otevřít a nastavit spojení, poslat/přijmout zprávu a uzavřít spojení. Uživatel zavolá funkce knihovny stdfile, která pomocí systémového volání provede požadovanou akci v jádře systému a vrátí uživateli výsledek.

Monitorovací program v uživatelském prostředí nejprve připraví spojení s ostatními zařízeními a domluví se na aplikační roli master/slave. Po domluvě začne provádět svoji příslušnou akci – slave posílá naměřená data a master je přijímá a monitoruje.

### 3. Implementace

V diagramu 1 je vidět návrh implementovaného systému a vzájemné komunikace. Monitorovací program je celý v uživatelském prostředí a pomocí systémového volání a filesystem driveru využívá I2C ovladače v jádře systému pro komunikaci s dalšími zařízeními.

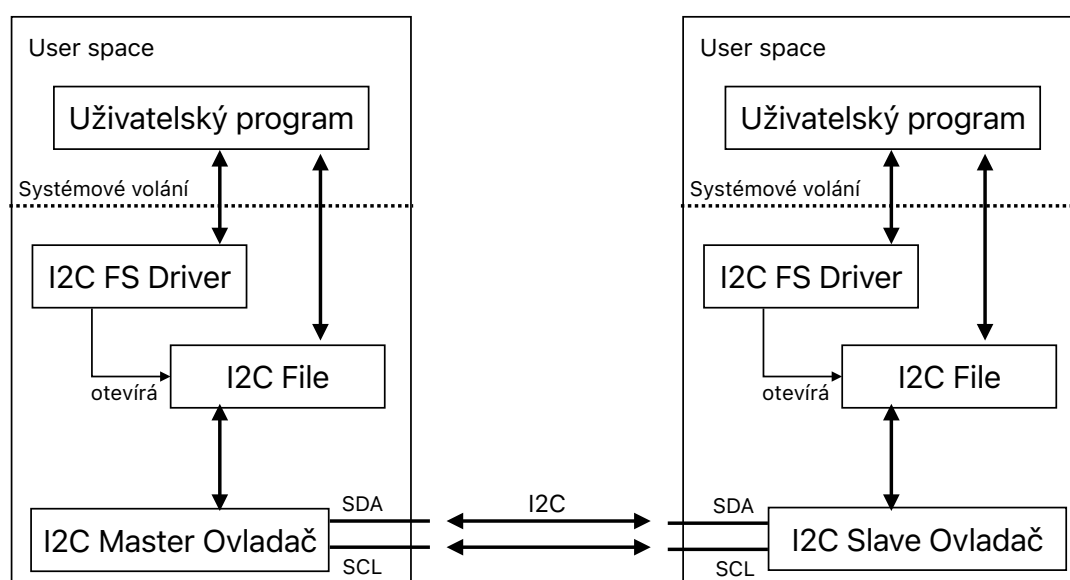


Diagram č.1 – návrh systému

#### 3.1. Protokol

Protokol je rozdělen do tří částí – navázání spojení, dohoda rolí a přenos hodnot. Každá část má fixní délku zprávy – navázání spojení a dohoda rolí pracuje se zprávou dlouhé 4 byty a přenos dat má zprávy dlouhé 6 bytů. Zároveň každá zpráva je ukončena hodnotou 0, protože při pokusu o čtení ze sběrnice slave stanice s prázdnou frontou pro odesílání zpráv (*TX FIFO*) je přenášena poslední hodnota, která zde byla uložena.

Pokud by zde tedy tato podmínka nebyla, nebyl by způsob, jak zjistit, zda jsou nějaká nová data od slave stanice k dispozici, či jsou zde pouze stále ty samá, již přečtená. To, zda jsou data k dispozici, je tedy rozeznáno na základě prvního bytu přijaté zprávy sběrnice masterem – pokud je první byte nulový, slave neposlal žádná nová data. Z tohoto dále vyplývá, že žádná přenášená zpráva nemůže hodnotou 0 začínat.

### Navázání spojení

- **syn** – navázání spojení od sběrnice mastera
- **ack** – potvrzení navázání spojení od slave

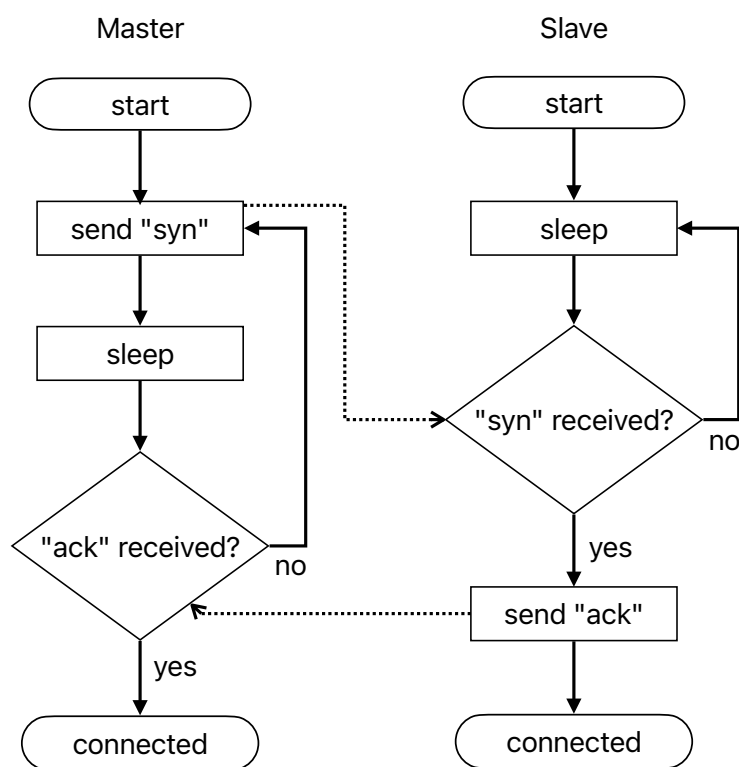


Diagram č.2 – navazování spojení

### Dohoda rolí

- **mst** – stanice si přeje stát se aplikačním masterem
- **slv** – stanice si přeje stát se aplikačním slave

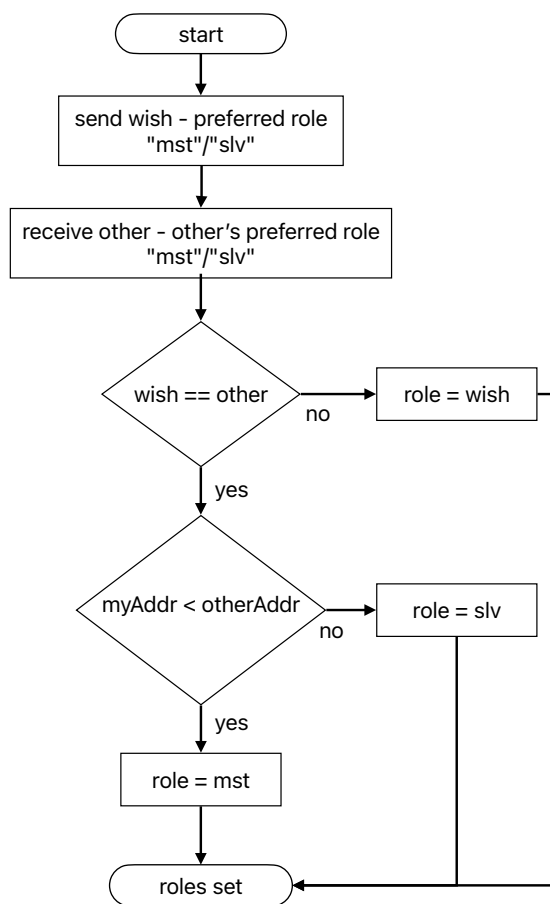


Diagram č.3 – dohoda rolí

## Přenos hodnot

Vždy jsou přenášeny 6bytové zprávy. První znak zprávy uvádí typ přenášené hodnoty následovně:

- **v** – hodnota je v pořádku
- **d** – hodnota je nebezpečná – je vyšší, než je horní, či nižší než dolní limit pro hodnoty
- **t** – nebezpečný trend – následující hodnota by mohla být nebezpečná

Dále následují 4 byty, ve kterých je zakódována hodnota jako datový typ float. Poslední znak zprávy je již zmíněný nulový byte.

## 3.2. Uživatelský program

Uživatelský program je jeden univerzální v obou zařízeních. Nejprve otevírá a nastavuje spojení, poté se s ostatními zařízeními dohodne na aplikačních rolích a následně začne vykonávat svoji práci.

K dohodnutí rolí si se zařízeními vymění zprávu s jejich preferovanými rolemi. V případě, že nedojde ke kolizi, role jsou nastaveny na jejich preferované. V případě, že ke kolizi dojde, zařízení s nižší adresou bude plnit roli aplikačního mastera.

## 3.3. I2C Master a Slave ovladač

Jak již bylo zmíněno, ovladač je implementovaný zvlášť pro komunikaci ze strany master zařízení a zvlášť ze strany slave zařízení. Pro přehlednost je definovaná základní třída *CI2C*, která slouží jako předek třídám *CI2C\_MASTER* a *CI2C\_SLAVE*, což jsou konkrétní implementace ovladačů.

Čtení na slave zařízení funguje pomocí IRQ přerušení. Jakmile je vstupní fronta z 1/8 zaplněná, je zavoláno IRQ přerušení a jeho obsluha ve funkci *CI2C\_SLAVE::IRQ\_Callback()*. Vše, co je dostupné k přečtení, přečte do bufferu a vyčká, až o to uživatel požádá.

## 3.4. I2C Filesystem driver

Uživatel může I2C rozhraní využívat pomocí souborového systému. Zavolá funkci *open* a zadá *DEV:i2c/#* (místo # bude číslo kanálu 0/1 pro master zapojení nebo 2 pro slave zapojení). Důležitou funkcí je *IOctl*, která nastavuje otevřené spojení a provádí úvodní pozdrav s ostatními zařízeními – tzv. *handshake* (viz. kapitola Protokol, část navazování spojení).

# 4. Testování

Testování této práce proběhlo na reálném zařízení Raspberry Pi Zero. Pro realizaci komunikace přes rozhraní I2C je potřeba spolu propojit SDA piny a SCL piny mezi master a slave zařízeními. Jednodušší způsob je spolu propojit dané piny na jednom zařízení. Jak

již bylo zmíněno výše, pro zapojení I2C master sběrnice je možné využít piny 2 a 3 nebo 0 a 1. A pro zapojení I2C slave sběrnice piny 18 a 19.

Piny k propojení jsou:

- Jedním kabelem SDA piny – 2 (nebo 0) a 18
- Druhým kabelem SCL piny – 3 (nebo 1) a 19

Na zařízení musí běžet 2 univerzální programy, kde jeden bude využívat master sběrnici a druhý slave sběrnici. V případě, že jsou spolu propojeny 2 zařízení, stačí jeden univerzální program na každém zařízení.

Každý program musí mít nastavenou vlastní jedinečnou adresu a adresu cílového zařízení, se kterým bude komunikovat. Dále musí mít nastavenou preferovanou aplikační roli. Program nejprve zkusí otevřít master sběrnici a v případě, že se mu to nepovede, zkusí otevřít slave sběrnici. Pokud se jedná o 2 procesy na jednom zařízení, prvnímu procesu se otevření povede a sběrnici zamkne. Druhému se to pak nepovede a otevře slave sběrnici.

## 5. Závěr

Dle zadání byla navržena a implementována podpora síťové komunikace přes rozhraní I2C. Součástí toho byla implementace ovladače zvlášť pro master sběrnici a zvlášť pro slave sběrnici a ty byli uživateli zpřístupněny pomocí filesystem driveru. Monitorovací program pro sdílení a monitorování hladiny glukózy v krvi byl implementován v uživatelském prostředí a úspěšně plní svou roli. Nejprve se s ostatními zařízeními dohodne na své roli a následně posílá/monitoruje naměřená data. Zadání tak bylo splněno a otestováno na jednom zařízení, které plnilo roli master i slave. Při zapojení více zařízení by bylo potřeba poladit dohodnutí rolí.