Semestrální práce PPC

Grafické rozhraní pro ovládání laboratorního zdroje

Milan Kopper

Cíle práce

Cílem této práce bylo vytvořit program pro ovládání laboratorního zdroje skrz sériovou komunikaci. K tomu bylo nutné upravit kód v řídícím MCU zdroje a vytvořit aplikaci ve frameworku QT.

Práce vychází z mé předchozí práce v rámci studia SŠ. Pro zpřehlednění tento dokument a soubory či funkce na které odkazuje obsahují jen změny a vylepšení realizované v rámci semestrální práce. Práce jako celek je obsahem GitHubu, kde se nachází i kompletní dokumentace k původní verzi projektu: https://github.com/Milan2mecha/zdroj

Komunikace po sériové lince

Pro komunikaci byly stanoveny rámce, tak aby bylo možné rozpoznat správnost přijatých údajů. U napětí a proudů byla zvolena přesnost na dvě desetinná místa, čili s rozlišením 0,01V a 0,01A. U teploty pak s přesností na jedno desetinné místo.

PC => MCU

Rámec má délku 11 bajtů. Jako startovací a dělící znaky jsou použity symboly $ a |.

Start bajt	Napětí				Dělící znak	Proud				Ukončovací znak
$	U	U	U	U	|	I	I	I	I	|

MCU => PC

Rámec má délku 22 bajtů. Jsou zde obdobně jako u komunikace opačným směrem použity znaky $ a |. Bajt s označením mód přenáší informaci, zda zdroj pracuje v jako napěťový (V) či proudový (C). Bajt error pak přenáší informaci o případném selhání zdroje, pokud zdroj funguje správně je přeneseno E pokud ne je přenášeno číslo chyby.

	Napětí					Proud					Mód cv/cc		teplota				Výkon ventilátoru			error
$	V	V	V	V	|	I	I	I	I	|	V/C	|	T	T	T	|	%	%	|	E

Implementace

V QT aplikaci je použita knihovna QtSerialPort, jako hrubý obrys fungovaní jsem vycházel s příkladu Serial terminal. Pro připojení a odpojení od portu jsou použity funkce openSerialPort() a closeSerialPort(). Pro listování portů je pak použita funkce QSerialPortInfo::availablePorts().

Pro přijímání: Při signálu readReady je zavolána funkce readData, ta následně přijme data a předá je pro kontrolu funkci checkmess(). Pokud je zpráva uznána jako kompletní je zobrazena, v opačném případě se vyvolá chybová hláška

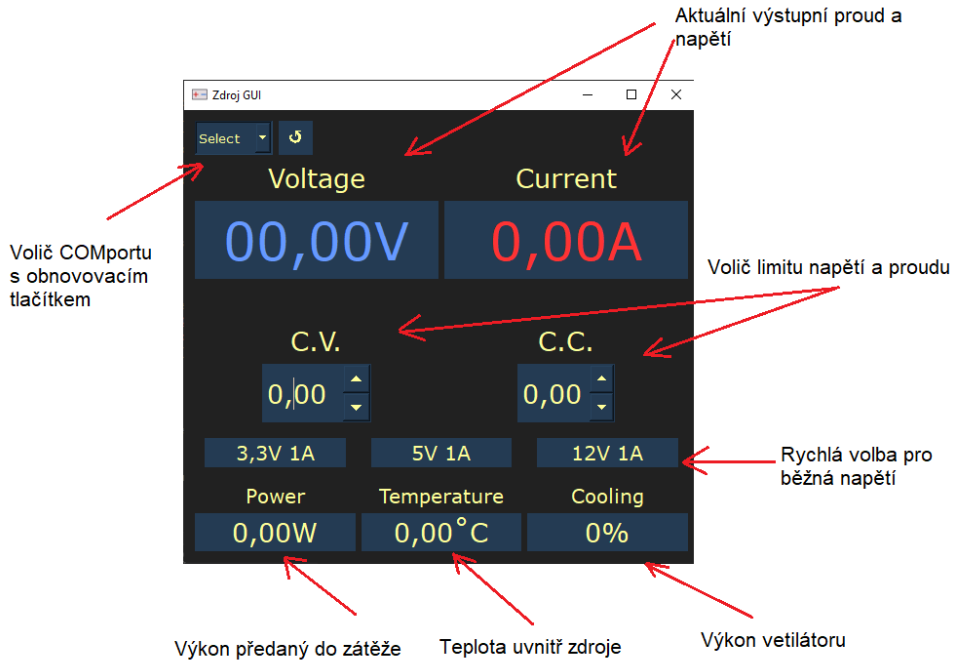
Pro odesílání: Je zavolána funkce writeData(), kterápoužívá funkci create_packet() pro sestavení rámce.

V MCU je pak použit middleware USB_DEVICE. U něj byl mírně modifikován soubor usbd_cdc_if.c který obsahuje funkci pro příjímání a odesílání dat.

Přijímání: Ve funkci CDC_Receive_FS() která je vyvolána při přijetí dat je zapsána proměnná značící nová data v bufferu. Při nejbližším průběhu hlavního loopu je vyvolána funkce read_ser() která přenastaví pomocí existujících funkcí výstupní napětí a proud zdroje.

Odesílání: Při překleslení displeje nebo v každém 256 průběhu hlavního cyklu je vytvořen pomocí funkce create_mess() vytvořen rámec pro odeslání. Následně se ho zařízení pokusí odeslat, max 100 pokusů.

Uživatelské rozhraní



Ovládací prvky tvoří dropbox pro volbu COMportu s refresh tlačítkem. A dvou spinboxů pro nastavení výstupního proudu a napětí doplněných o tři tlačítka rychlé volby běžných napětí, která

nastaví spinboxy na danou hodnotu. Při změně hodnoty spinboxu nebo rychlé volbě je odeslán nový rámec do seriového zařízení.

Výstupní periferie tvoří qLabely pro zobrazení napětí, proudu, výkonu, teploty a stavu ventilátoru. Dále je k indikaci módu ve kterém se zdroj nachází použita změna stylu spinboxů. Zeleně podbarvený spinbox značí, že se jedná o aktuální řídící hodnotu.

Chyby

V případě chyby je vyvolána funkce setError() která zobrazí QMessageBox s příslušným textem.

Číslo chyby	Název	Text
0	COM open error	Couldn’t open COM port
1	Read error	Check connection and try again.
2	Device error	Temperature reading error.
3	Device error	Device is overheatted
4	Device error	Error occured while reading U0
5	Device error	Error occured while reading U1
6	Device error	Error occured while reading U2