Dokumentace k webovému serveru

IMP 2019/20

xplsek03

Nastavení a spouštění

Před nahráním programu do zařízení je potřeba nastavit IP adresu, která se nastaví na rozhraní ethernet, masku sítě a výchozí bránu sítě (složka source/main.c, zadáváno po oktetech).

Webová stránka je dostupná na adrese+ "/" nebo "/index.html". Je podporováno ovládání prvků přes webovou stránku nebo přes klávesy na fitkitu (tlačítka SW2-5). Defualtně je zapnuté ovládání přes HW, ovládání přes přes web se aktivuje stisknutím tlačítka SW6. Stránka se neobnovuje (AJAX).

Obnovení stránky u klienta je možné pouze v módu ovládání prvků přes web! Pokud je snaha obnovit stránku v módu řízení přes HW, aplikace se znovu nenačte: je to proto, že v HW módu probíhá pollování a requesty na stránku jsou ignorovány.

Pro vývoj bylo použito MCUXpresso (Windows) a poslední verze MCUXpresso SDK, bez freertos a s lwip.

Načítání serveru chvíli trvá, v případě spouštění přes MCUXpresso je potřeba počkat, až se objeví ladicí zpráva.

Zdroje

Zdrojový kód vychází částečně z ukázkového serveru v MCUXpresso SDK nejnovější verze:

import SDK examples > examples/demo/lwip/bm_httpd,

částečně z oficiálních tutoriálů z http://git.savannah.gnu.org/cgit/lwip/lwip-contrib.git/tree/examples/httpd. Dokumentace k raw API funkcím se dají najít tady:

http://www.nongnu.org/lwip/2_1_x/group__httpd.html

Pro ovládání a inicializaci zařízení na fitkitu jsou použité funkce ze z demo fitkit zipu v souborech k předmětu IMP.

Aplikace

Aplikace nepoužívá operační systém, server používá lwIP a jeho raw callbacks API. V hlavním souboru Source/main.c se inicializují zařízení na fitkitu (tlačítka, diody), nastaví se lwIP stack a inicializuje se, nastaví se síťové rozhraní ethernet - přiřazení adresy, zapnutí, a na vytvořeném PCB rozhraní se na defaultním http portu sběr příchozích paketů. Taky se přiřadí určeným stránkám callback funkce, které zkoumají jejich parametry v URI (GET requesty). V httpd_opts.h byly změněny proměnné pro

zapnutí CGI a custom souborů, v include/lwip/opt.h byly změněny proměnné (`LWIP_TCP_KEEPALIVE`, `TCP_KEEPIDLE_DEFAULT`,`TCP_KEEPINTVL_DEFAULT`,`TCP_KEEPCNT_DEFAULT`), kvůli AJAX.

Samotné soubory, index.html a 404.html byly uloženy jako sekvence bajtů uvnitř struktur definujících jednotlivé soubory, v souboru lwip/src/apps/httpd/include/fsdata.c. Soubory by se daly generovat dynamicky, ale takhle je to rychlejší. Soubor Fsdata.h pak obsahuje pouze definici struktury jednoho souboru.

Soubor main.c zajistil inicializaci zařízení a stacku, samotné zpracovávání příchozích dat a používání funkcí, které posílají data zpátky do prohlížeče je umístěno za rozhraním webového serveru (inicializace pomocí funkce httpd_init). Soubor fsdata.c zajistil data, která se mají nahrát do stránky. Zajímavé části aplikace, které se fakticky starají o zpracování příchozího obsahu a jeho parsování jsou umístěny v souborech lwip/src/include/lwip/apps/fs.h a lwip/src/apps/httd/fs.c. Protože v podstatě všechny části souvisejí s parsováním a obsahem souborů ve vlastním souborovém systému (to jako ty bajty ve fsdata.c), dal jsem je kvůli úspoře místa do jednoho souboru.

Soubor fs.h obsahuje makra prvků na fitkitu, které se dají ovládat. Soubor fs.c definuje strukturu cgi_handlers, která přiřadí libovolné adrese callback funkci, která prozkoumá její argumenty a jejich hodnoty v URI, a podle nich posílá zpět vhodné odpovědi do prohlížeče. Protože se jedná o AJAX, funkce pro vytváření nových souborů definované ve fs.c vytváří pouze "soubor" s obsahem krátkého textového řetězce (např. "up" apod.), který poté klientská strana zapracuje do stránky. Soubor ve filesystému 404.html s epoužívá při zadání neplatné adresy, neplatné hodnoty GET argumentů se ignorují a je poslán znovu index.html.

Callback funkce **cgi_handler_basic** zajišťuje pouze znov odeslání dat uložených ve filesystému (index.html) a parsování argumentů GET requestů při zapnutém módu ovládání prvků přes web. Diodám a tlačítkům klávesnice (pouze SW2-5) se po stisknutí prvku na stránce změní jejich HW stav, requesty jsou jednoduché, např. index.html?led=l1. Pokud dojde na stránce k vypnutí HW módu ovládání (ve kterém je stránka defaultně), tlačítka SW2-5 jdou ovládat na fitkitu a na webu se zobrazuje aktuální stav stisku. Stav diod se při přepínání módu nemění. Odpověď serveru v případě ovládání přes webové rozhraní je request s obsahem "valid", aby odeslaný request u klienta dostal nějakou odpověď.

Callback funkce **cgi_server_update** se stará právě o ovládání prvků přes HW. Při spuštění módu klient pravidelně odesílá request na stránku update.html s vysokým timeoutem, který server zachytí a v pravidelných intervalech kontroluje v callback funkci, jeslti došlo k některé operace (stisknutí tlačítka, uvolnění apod.). Mód ovládání se dá vypnout tlačítkem SW6, v takovém případě se polling na serveru přeruší a do prohlížeče se pošle zpráva k nastavení módu ovládání přes web. Proměnné X_pressed definované v tomto souboru slouží k zaznamenání stisku knokrétních tlačítek. Generované odpovědi jsou jednoduché řetězce "pollkill" - přerušení HW módu nebo směr zmáčknutého tlačítka (řetězec: up,down,left,right).

Problém s pollingem: původní návrh byl takový, že lwIP raw API callbacky budou zpracovávat více požadavků zaráz, tj. před dokončením zpracování jednoho requestu přijmeme druhý request a ten zpracuje. Tím by se vyřešilo, že by server přijal request, držel ho na serveru a pokud by přišel stisk tlačítka z HW, poslal by odpověď (a mezitím by se přijalo a zpracovalo pomocí druhého handleru X dalších requestů). Ukázalo se, že lwIP raw API nepřijme jiný request, dokud není předchozí zpracován

v callback funkci (pravděpodobně by bylo nějak potřeba použít funkci tcp_poll nebo nějaké nastavení z vyšší vrstvy, ale tak daleko jsem se nedostal a ani na oficiálním fóru podpory o tom moc nevěděli). Proto je ovládání implementováno přes dva samostatné módy.