

# Pretaktovanie ARM Cortex-M4 jednoducho a pohodlne

Publikované: 11.02.2015, Kategória: Mikroprocesory

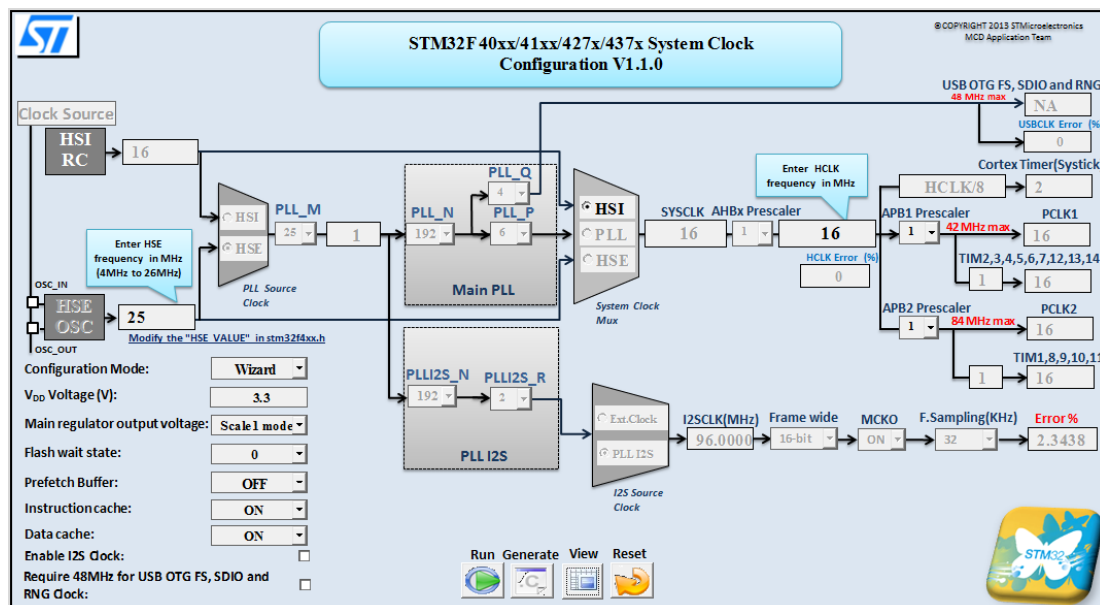
www.svetelektro.com

V tomto článku sa dozviete ako rýchlo a jednoducho pretaktovať MCU ARM Cortex-M4.

Na tento účel nám posluží platforma STM32F4DISCOVERY a taktovacia utilita od ST Microelectronics STM32F4\_Clock\_Configuration\_V1.1.0 voľne dostupná na stránkach výrobcu [1]. Utilita nám posluží na základne porozumenie taktovania MCUov celej rodiny Cortex-M a tiež na vygenerovanie systémového nastavenia system\_stm32f4xx.c, ktoré využíva knižnica CMSIS.

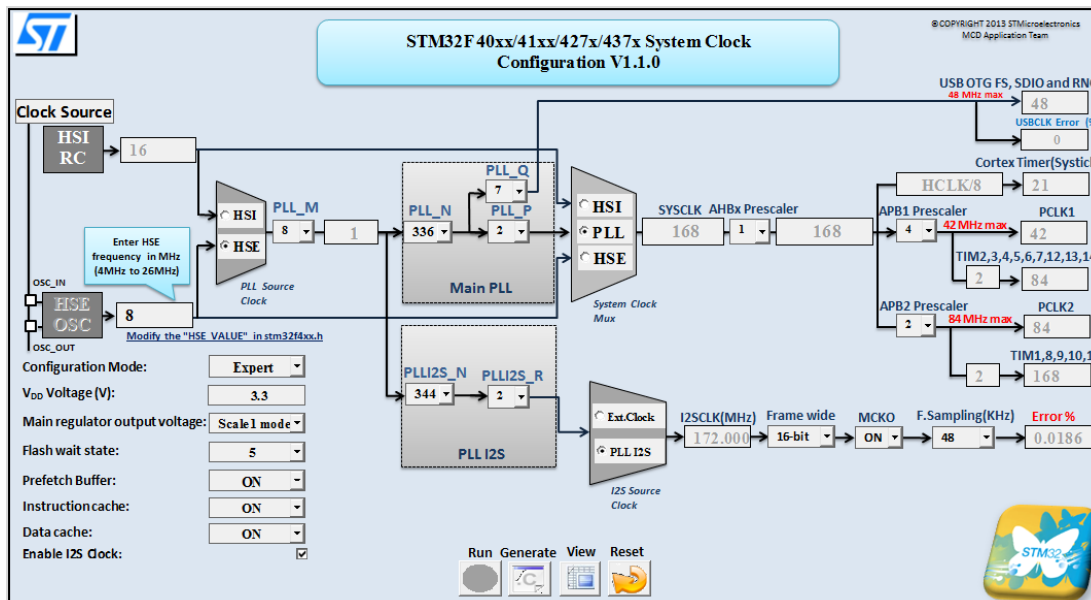
Utilita od STM je naprogramovaná ako súbor makier pre tabuľkový editor MS Excel. Keďže STM si dali pozor a utilitu ošetrili (google poradí ako hacknúť .xls), aby nedovolila nastaviť hodiny na vyššiu ako maximálnu rýchlosť svojich Cortex-M4xx MCUov, ktorá je 168MHz a tiež zamedzili prístup k editácii makier, tak si najskôr popíšeme ako utilitu použiť a ako sú jednotlivé hodinové signály celého systému odvodené. Po základnom pochopení fungovania hodinového systému MCU bude samotná editácia vygenerovaného súboru jednoduchá až triviálna.

Rozhranie utility po spustení priamo poskytuje viditeľnosť celého systému generátora hodín vo forme blokového diagramu. Keďže utilita je interaktívna, priamo podporuje zmenu jednotlivých parametrov a v reálnom čase prepočítava frekvencie všetkých signálov. Užitočnou funkciou utility je aj prepočet odchýlky frekvencie hodín pre protokoly I2S a USB.



Z rozhrania utility môžeme pozorovať, že na taktovanie celého MCU môžeme v základnom nastavení použiť interný HSI RC (High Speed Internal RC oscillator) alebo externý HSE (High Speed External) zdroj hodinového signálu. Zdroj hodín je vybraný pomocou multiplexeru a deličom je ďalej vedený na vstup PLL (Phase Locked Loop), ktorý umožňuje procesoru bežať na vyšších frekvenciách ako HSI alebo HSE oscilátory. Nasledujúci multiplexer umožňuje vybrať zdroj hodín pre samotný MCU z HSI, HSE alebo PLL. Výstupným signálom multiplexeru je taktované samotné jadro MCU a AHB matica. Hodinový signál je ďalej delený deličom pre zbernicu periférii APB a jednotlivé periférne zariadenia MCU. I2S rozhranie je časované dedikovanou PLL, ktorá umožňuje generovanie hodín pre externý DAC s minimálnou odchýlkou pri rôznych rýchlostiach jadra.

Keďže v našom príklade použijeme stm32f4discovery platformu, najskôr si ukážeme ako pomocou utility vygenerovať systémové nastavenie MCU a všetkých sub-systémov bežiacich na maximálnych povolených frekvenciách.



V utilite najskôr prepneme konfiguračný mód z Wizard na Expert, ktorý nám umožní meniť jednotlivé parametre manuálne. Pri konfigurovaní celého generátora hodín musíme mať na pamäti možné odchýlky hodín najmä pre I2S a USB komunikačné rozhrania. Rozhranie pamäťovej karty SDIO je plne digitálne a tak aj neštandardné rýchlosti by nemali spôsobovať problémy, nemali by však prekročiť 50MHz v prípade použitia SD karty a 52MHz v prípade použitia MMC. Stm32f4discovey má vo svojej výbave externý 8MHz kryštál, ktorý použijeme ako zdroj hodinového signálu. Do riadku "Enter HSE frequency in MHz" teda vpišeme 8 a v multiplexeri vyberieme ako zdroj hodín HSE. Pre spoľahlivý beh PLL nám utilita oznámi, že vstupný signál môže mať hodnotu medzi 1 až 2MHz, delič PLL\_M teda nastavíme na 8, ktorý nám poskytne 1MHz vstupný hodinový signál do PLL. Použitím 1MHz znížime celkovú nepresnosť externého oscilátora 8 krát čo nám poskytne presnejší a stabilnejší beh PLL a jej výslednej frekvencie. Nastavením násobiča PLL\_N na 336 a deliča PLL\_P na 2 dosiahneme maximálnu povolenú frekvenciu MCU 168MHz, nastavením deliča PLL\_Q na hodnotu 7 umožníme bežať USB rozhraniu na presných 48MHz, teda plnej rýchlosti USBv2.0 Rozhranie I2S umožňuje generovanie hodín dedikovanou PLL. S nastavením PLL poexperimentujeme podľa požadovanej vzorkovacej frekvencie audia, odchýlky do 1% sa nemusíme obávať. Deliče subsystémov AHB, APB1 a APB2 nastavíme na hodnoty 4 a 2, ktoré umožnia bežať časovačom a ostatným periférnym zariadeniam na ich maximálnych frekvenciách 42 a 84MHz. Ak sme všetky hodnoty nastavili správne môžeme kliknúť na tlačidlo "Generate" v spodnej časti utility. Vygenerovaný súbor system\_stm32f4xx.c sa bude nachádzať v rovnakom adresári ako utilita. Systémový súbor je prehľadne vygenerovaný a patrične okomentovaný. Po jeho otvorení v textovom editore nájdeme definície hodnôt PLL a môžeme sa presvedčiť, že odpovedajú tým, ktoré sme nastavili v utilite.

```

158  /****** PLL Parameters *****/
159  /* PLL_VCO = (HSE_VALUE or HSI_VALUE / PLL_M) * PLL_N */
160  #define PLL_M      8
161  #define PLL_N      336
162
163  /* SYSCLK = PLL_VCO / PLL_P */
164  #define PLL_P      2
165
166  /* USB OTG FS, SDIO and RNG Clock = PLL_VCO / PLL_Q */
167  #define PLL_Q      7
168
169  /* PLLI2S_VCO = (HSE_VALUE or HSI_VALUE / PLL_M) * PLLI2S_N
170     I2SCLK = PLLI2S_VCO / PLLI2S_R */
171  #define PLLI2S_N   344
172  #define PLLI2S_R   2
173
174  /******

```

V prípade pretaktovania MCU musíme mať na pamäti zmenu celkových systémových frekvencií, ktorými sú časované všetky rozhrania od systémového "systick" časovača cez UART, až po SDIO a USB. Nesmieme teda zabudnúť na výpočet správnych deličov pre každé zariadenie, ktoré pre svoj časový základ využíva systémové hodiny, napríklad UART alebo ktorýkoľvek časovač a ostatné. Architektúra Cortex-M4 MCU od ST Microelectronics je pomerne dobre navrhnutá a umožňuje výrazné pretaktovanie. S použitou platformou sa podarilo dosiahnuť stabilných 250MHz. MCU sa pokúsime pretaktovať na stále úctyhodných 240MHz čo predstavuje 43% nárast rýchlosti jadra. Použitím 240MHz tiež umožníme USB rozhraniu bežať na presných 48MHz. Hodnoty PLL teda prepíšeme nasledovne: PLL\_M = 8, PLL\_N = 480, PLL\_P = 2, PLL\_Q = 10. V prípade použitia 48kHz audia môžeme použiť rovnaké hodnoty pre PLLI2S. Komentáre vo vygenerovanom súbore pomôžu sprehľadniť ako je výsledná hodnota vypočítaná. Na záver si môžeme ešte pripomenúť, že pretaktovaním MCU rastie jeho spotreba takmer exponenciálne a nie je teda vhodná pre zariadenia orientované na spotrebu. Prajem ešte raz veľa zdaru pri pretaktovaní ARM MCU a pamätajte : menej je niekedy viac.

Tz.