

STM32 – Debugging

Petr Novák / novakpe@cvut.cz / 2022-04-22

Obsah

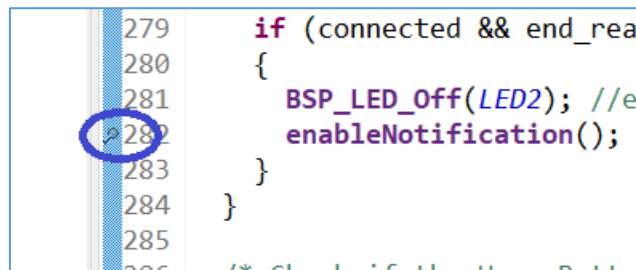
1	Úvod	1
2	Breakpoint	1
3	Zobrazení proměnných.....	1
4	OnLine zobrazení proměnných při běhu	4
5	Zobrazení FLASH a SRAM.....	6

1 Úvod

Zde jsou popsány některé možnosti, a techniky jak ladit program v procesoru STM32 za pomoci STM32CubeIDE.

2 Breakpoint

Breakpointy lze vkládat (skoro) na libovolné řádky kódu. Je vhodné je vkládat na výkonné řádky kódu a nikoli na příklad na definici nějaké proměnné (občas jsou s tímto problémy). Breakpoint je zobrazen vlevo na začátku programového řádku jako celkem malý puntík.



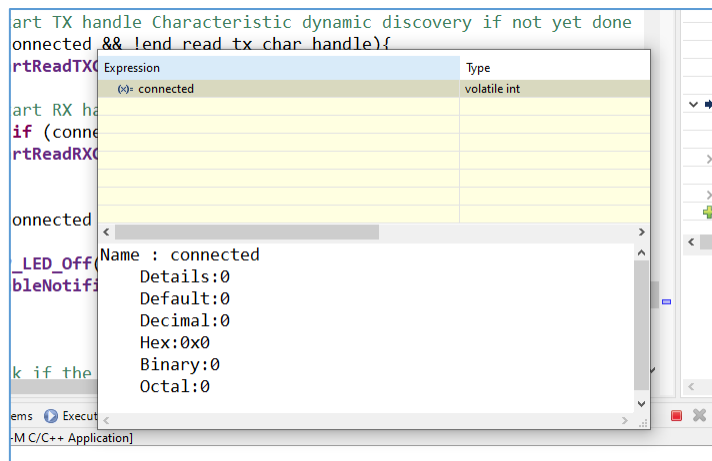
Vloží se dvojklikem na tuto modrou čáru. Opětovným dvojklikem se odstraní.

Poznámky:

- Před odstraněním programového řádku / kódu s breakpointem, doporučuji nejprve odstranit tento breakpoint a teprve potom programový kód. Občas se breakpoint po odstranění kódu někde ztratí / přesune a dělá to problémy. Často pomůže celý projekt zavřít a zase jej otevřít.

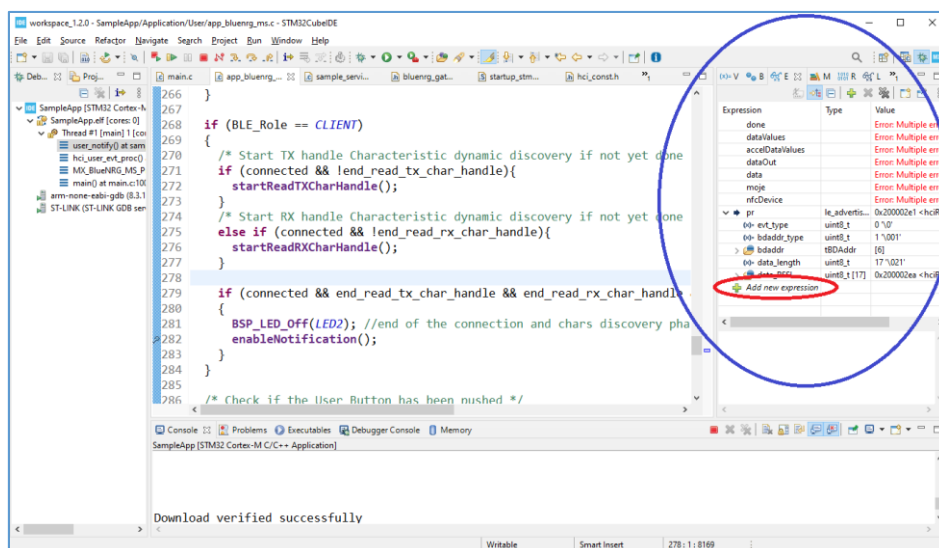
3 Zobrazení proměnných

Po zastavení programu (například dosažením breakpointu) si lze prohlížet definované proměnné. V naprosté většině případů stačí podržet ukazatel myši nad požadovanou proměnnou a zobrazí se dialog obsahující její hodnotu.



Aktuální hodnota proměnné je zobrazena v několika číselných soustavách. Takto lze však zobrazit obsah pouze jedné proměnné a to ještě těch základních číselných.

Pokud je potřeba zobrazit více proměnných současně, nebo těch co mají poněkud složitější obsah (pole, struktury, ...) tak je vhodný seznam proměnných / hodnot na pravé straně (zobrazen samozřejmě pouze při po-zastavení běhu dříve spuštěného programu).



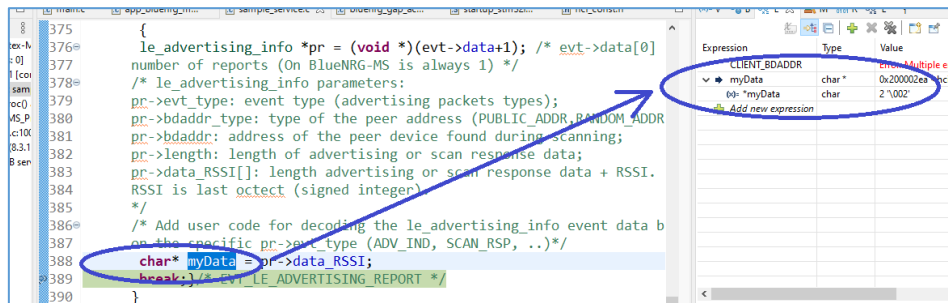
Přidat další proměnnou do seznamu zobrazených lze klikem na zelené „plus“, zobrazí se editační pole, do něhož lze název proměnné vepsat (ideálně zkopírovat ze zdrojového kódu / textu). V seznamu / tabulce jsou tři sloupce:

- Název – Název proměnné zapsaný / zkopírovaný do tabulky.
- Typ – Jakého typu je zobrazená proměnná.
- Hodnota – Jakou má proměnná aktuálně hodnotu.

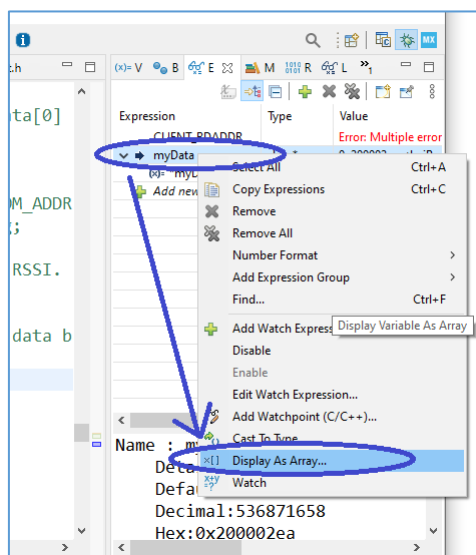
Pokud se jedná o proměnnou běžného typu (int, char, ...), tak je zobrazení její hodnoty vcelku jednoduché a přehledné. Dokonce i u statického pole lze jeho prvky zobrazit jeho snadným rozbalením (značka pro rozbalení před názvem proměnné). Jsou však proměnné, jejichž zobrazení je potřeba vhodně upravit aby se v jejich datech člověk vyznal:

Odkaz do paměti na pole hodnot

Pokud je potřeba zobrazit podle hodnot, na který je pouze odkaz (například vstupující do funkce) zle jej rovněž zapsat do seznamu proměnných. Bohužel po rozbalení je zobrazena pouze první hodnota, protože zobrazovač neví jak je pole dlouhé.

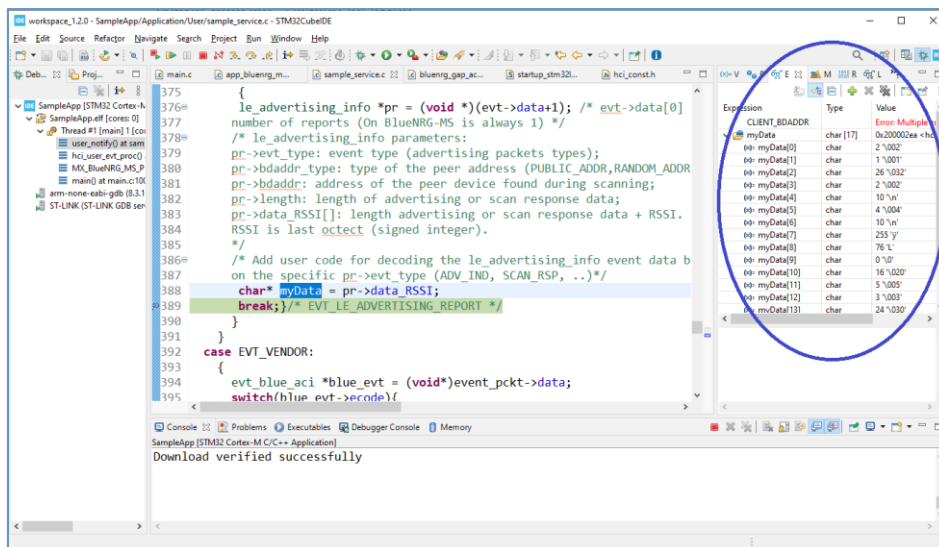


To lze však velmi snadno napravit. Stačí na název pole kliknout pravým tlačítkem a z lokálního menu vybrat položku „Display As Array ...“.



Zobrazí se (malý) dialog pro zadání délky pole. Po zadání čísla a potvrzení dialogu lze již pole rozbalit a zobrazit všechny jeho hodnoty.

Zobrazení struktury pomocí odkazu

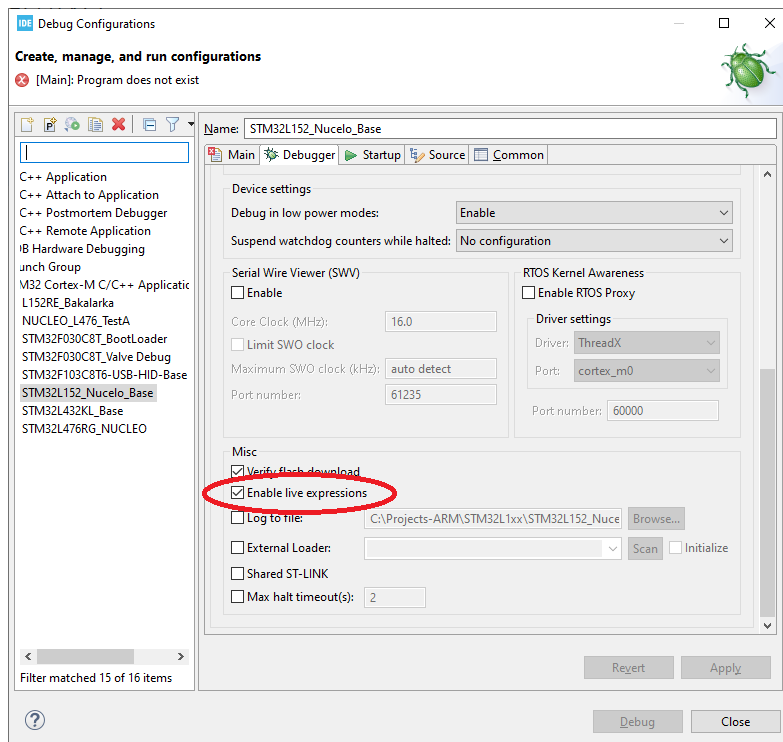


...

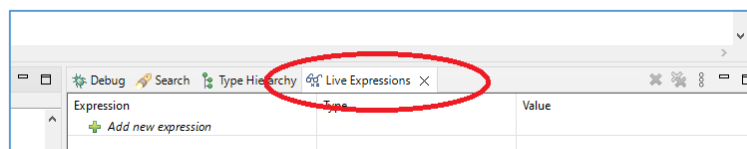
4 OnLine zobrazení proměnných při běhu

Pokud je potřeba se podívat na hodnoty nějakých proměnných tak v předešlé kapitole bylo nutno program zastavit (například pomocí breakpointu) a teprve poté bylo možno prohlížet hodnoty jednotlivých proměnných. Nová verze ST-Link umožňuje v omezené míře zobrazovat průběžně hodnoty některých proměnných v prostředí STM32CubeIDE i bez zastavení programu, tedy i přímo při jeho běhu. Postup je následující:

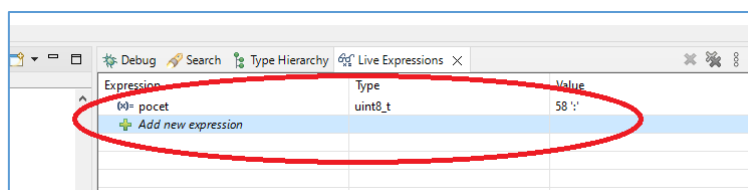
- Zkontrolovat, zda je povoleno / zatrženo **Enable Live Expressions**. Uvedeno v: **Run** (horní menu) – **Debug Configurations** (položka v rozbalovacím menu) – **Debugger** (záložka na nově otevřeném dialogu)



- Sledovanou proměnnou vytvořit jako globální, tedy mimo funkci (v podstatě existující po celou dobu běhu programu).
- Spustit program pomocí ladění a zastavit jej na jeho začátku.
- Na pravé straně (nahore / dole podle umístění) najít záložku „Live Expressions“.



- Pokud tato záložka není (nelze ji najít) tak: kliknout na hledací lupu v horním pravém rohu – začít psát **Live Expression** – v zobrazeném seznamu najít tuto možnost a vybrat (například kliknout). Zobrazí se požadovaná záložka / dialog.
- Do připraveného řádku vepsat název (dříve vytvořenou globální) proměnnou.
- Spustit běh programu.
- Zhruba jednou za vteřinu (mnohdy i častěji) je průběžně čtena hodnota uvedené proměnné a zobrazována v této záložce.



Poznámky:

- Pozor na skutečnost, že **Live Expression** je vhodné zejména pro pomaleji se měnící proměnné, například pro globální stav programu.

5 Zobrazení FLASH a SRAM

Někdy je potřeba zobrazit přímo adresy a jejich hodnoty ve FLASH nebo SRAM. Nejprve je potřeba vyvolat okno pro zobrazení obsahu paměti pomocí:

- **Window** (položka v horní menu)
- **Show View** (položka v rozbalovacím menu)
- **Memory Browser** (položka v rozbalovacím pod-menu)

V dolní části STM32CubeIDE se zobrazí okno (se záložkou) zobrazující obsah paměti. Do horního řádku v tomto okně lze zapsat adresu a stisknout vlevo od něho tlačítko **GO**. Bude zobrazen obsah části požadované paměti. Paměť lze listovat nahoru / dolů zcela bez omezení. Pozor však na následující skutečnosti:

- Tímto se zobrazuje část paměti z celkové paměti, tedy z celého adresního prostoru 4GB. Pozor tedy na tyto skutečnosti:
 - o **Programová FLASH** paměť (nejčastěji) začíná od adrese 0x08000000. Pokud je tedy požadavek na zobrazení například adresy 0x1000 ve FLASH paměti je potřeba zadat skutečnou / celkovou adresu:
 $0x08000000 + 0x1000 = 0x08001000$ (na toto pozor)
 - o **Datová SRAM** paměť (nejčastěji) začíná od adrese 0x02000000. Pokud je tedy požadavek na zobrazení například adresy 0x1000 ve SRAM paměti je potřeba zadat skutečnou / celkovou adresu:
 $0x02000000 + 0x1000 = 0x02001000$ (na toto pozor)
- Pomocí tlačítka **New Tab** lze zobrazit / přidat další záložku s částí zobrazené paměti. Takto lze vytvořit několik záložek, v každé zobrazit požadovanou část paměti a ty pouze přepínat (místo rolování obsahu nahoru / dolů).

Poznámky:

- Toto zobrazení je (velmi) vhodné například při kontrole ukládání hodnot do FLASH využívané jako emulace paměti typu EEPROM. Data jsou ukládána do stránky FLASH paměti od známé adresy a lze je tedy takto snadno kontrolovat.