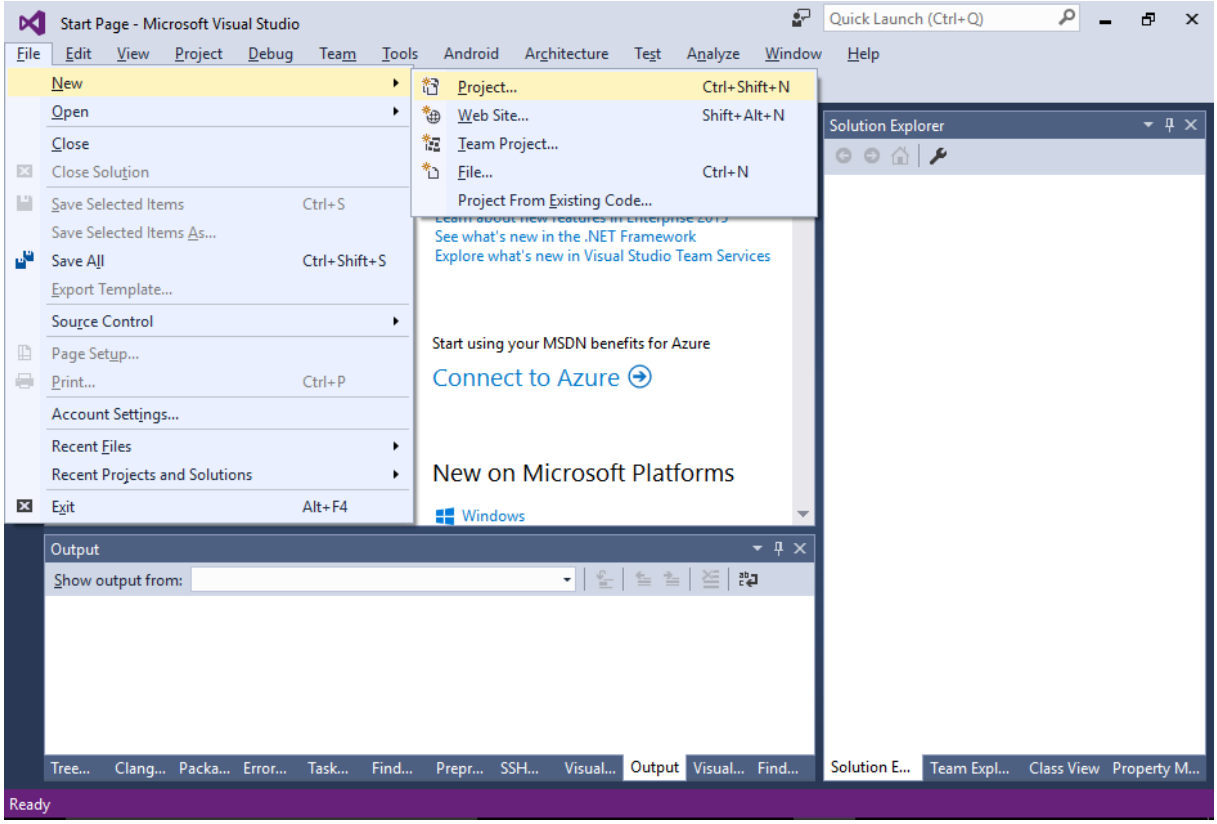
Úvod do STM32F407 dosky

- STM32F407VGT6 microcontroller featuring 32-bit ARM® Cortex® -M4 with FPU core, 1-Mbyte Flash memory, 192-Kbyte RAM in an LQFP100 package
- On-board ST-LINK/V2 on STM32F4DISCOVERY (old reference) or ST-LINK/V2-A on STM32F407G-DISC1 (new order code)
- USB ST-LINK with re-enumeration capability and three different interfaces:
  - Debug port
  - Virtual Com port (with new order code only)
  - Mass storage (with new order code only)
- Board power supply: through USB bus or from an external 5 V supply voltage
- External application power supply: 3 V and 5 V
- LIS302DL or LIS3DSH ST MEMS 3-axis accelerometer
- MP45DT02 ST-MEMS audio sensor omni-directional digital microphone
- CS43L22 audio DAC with integrated class D speaker driver
- Eight LEDs:
  - LD1 (red/green) for USB communication
  - LD2 (red) for 3.3 V power on
  - Four user LEDs, LD3 (orange), LD4 (green), LD5 (red) and LD6 (blue)
  - 2 USB OTG LEDs LD7 (green) VBUS and LD8 (red) over-current
- Two push-buttons (user and reset)
- USB OTG FS with micro-AB connector
- Extension header for all LQFP100 I/Os for quick connection to prototyping board and easy probing
- Comprehensive free software including a variety of examples, part of STM32CubeF4 package or STSW-STM32068 to use legacy standard libraries

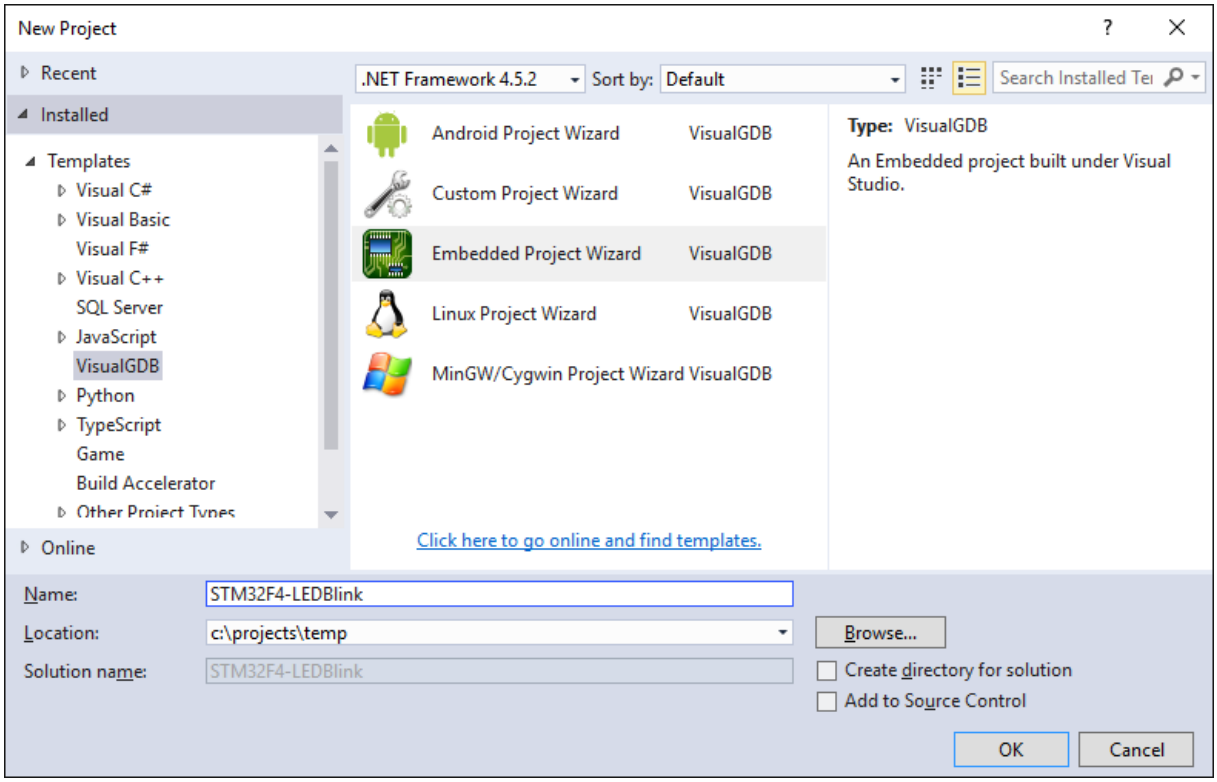
STM32 F4

Visual Studio a rozšírenie VisualGDB

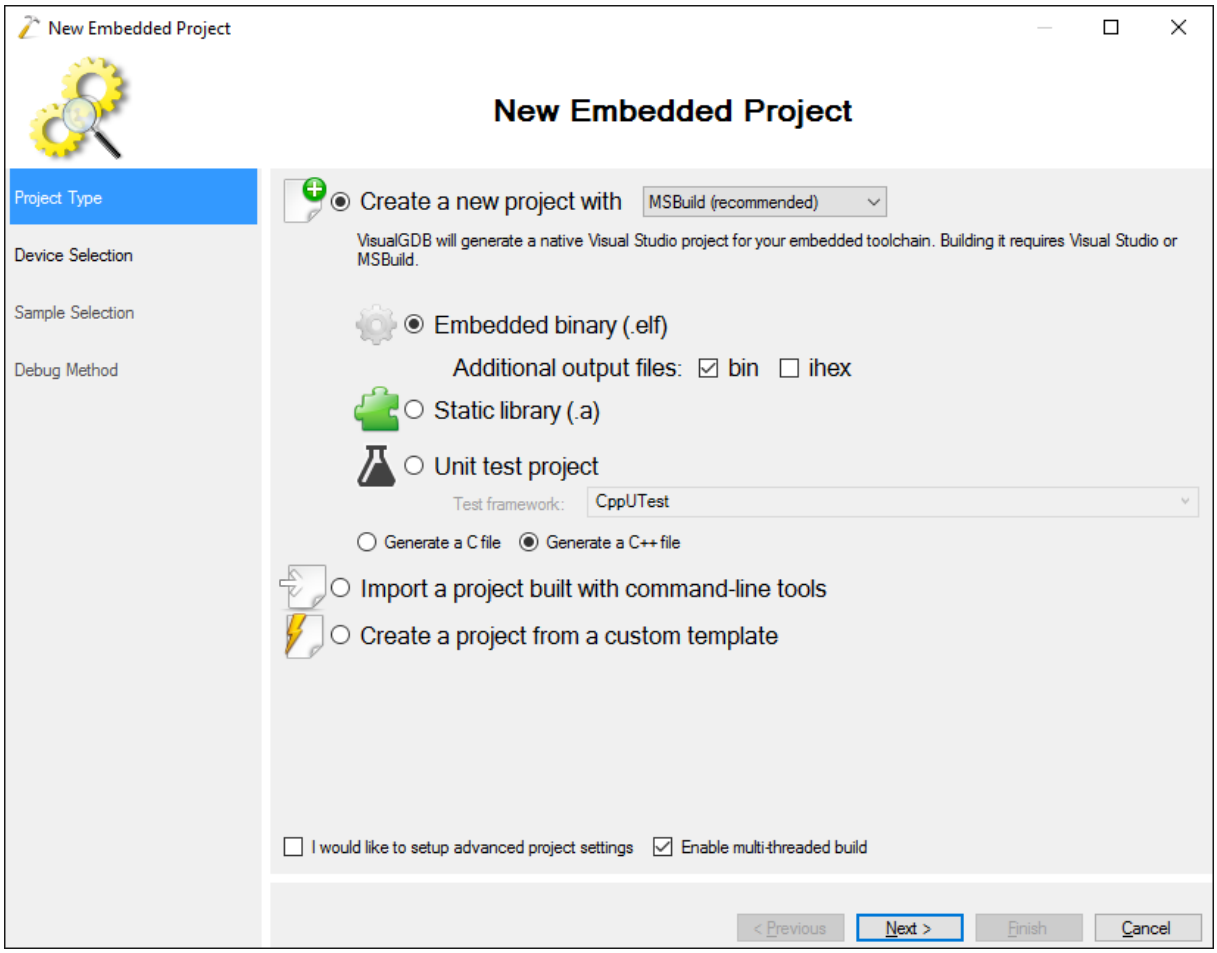
1. Spustíme Visual Studio, file, new, project



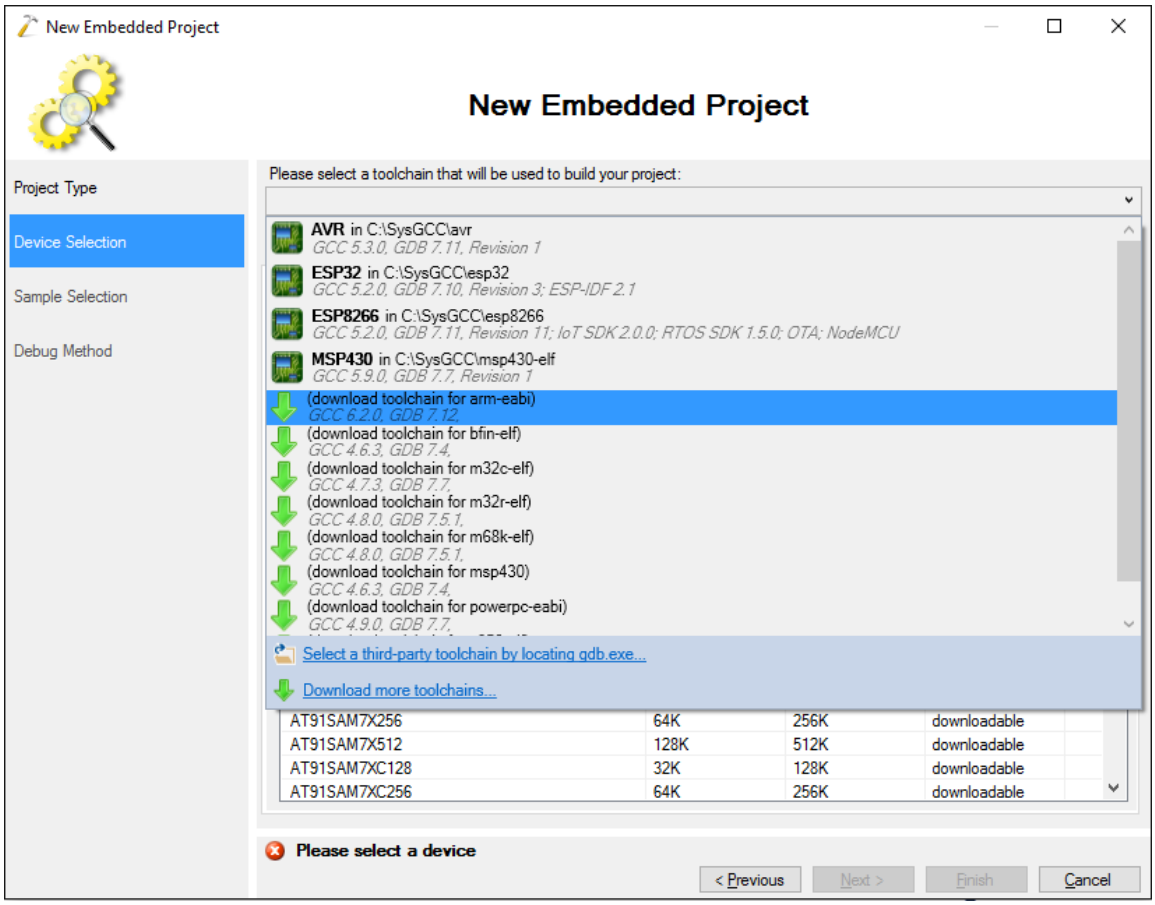
2. VisualGDB, embedded project wizard

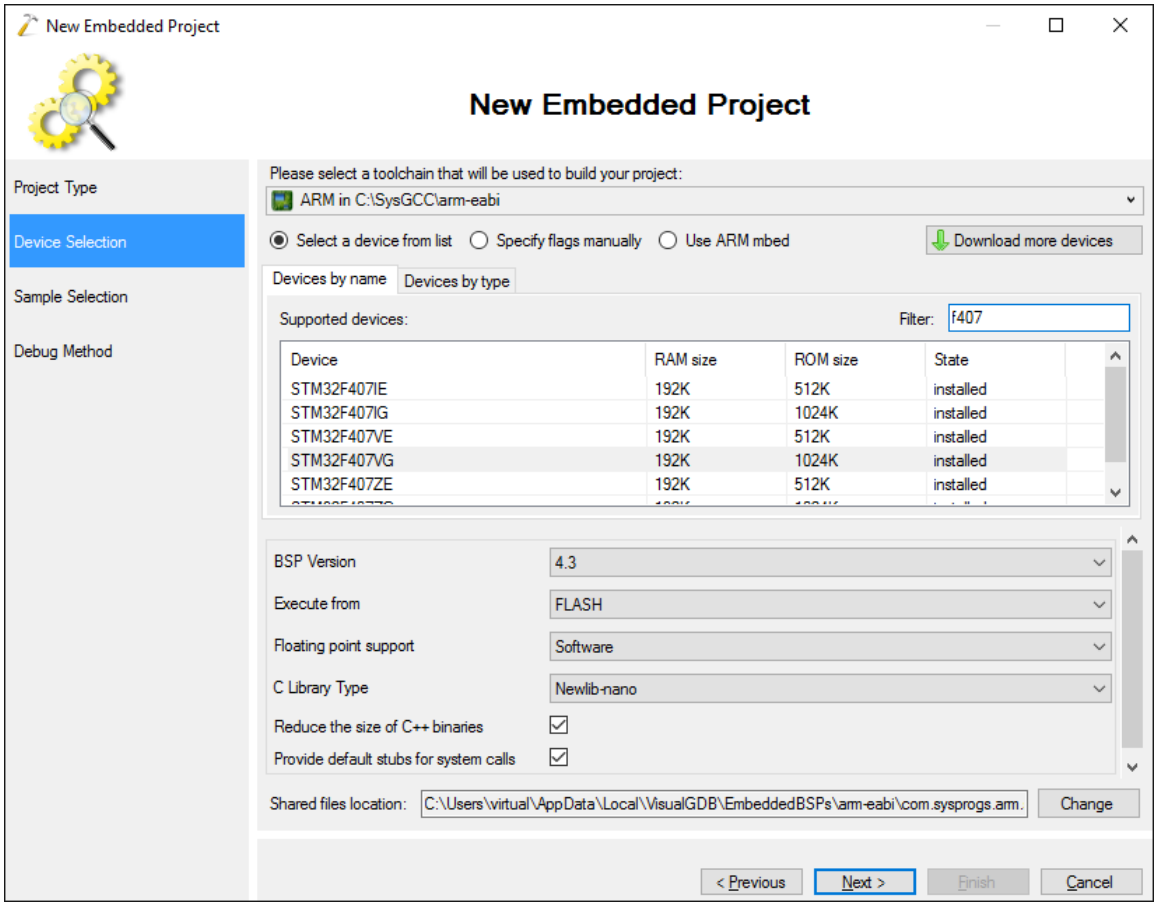


3. Vytvárame nový firmvérový súbor

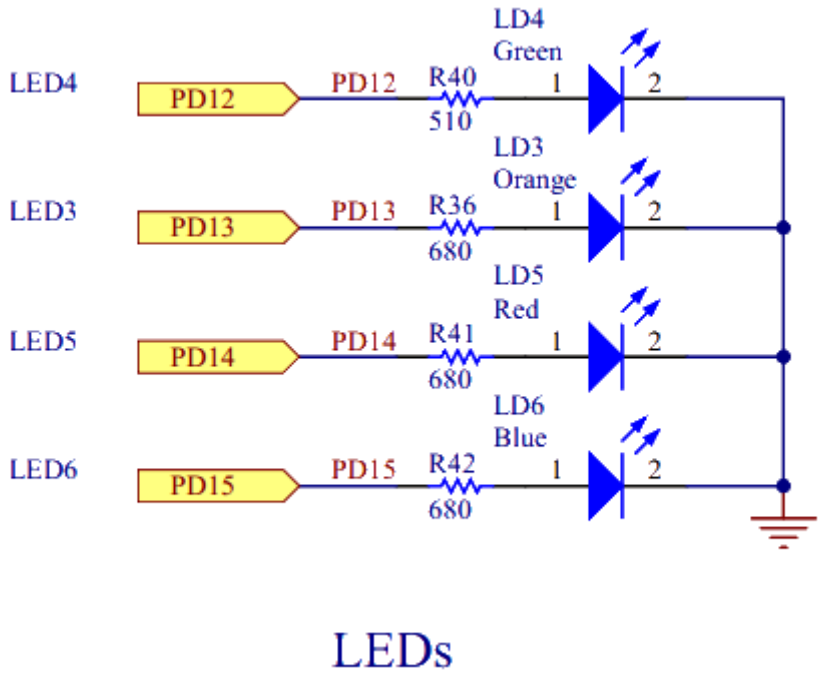


4. Ako robíme firmvér pre STM32F4 dosku, ktorá obsahuje STM32F407VG mikroregulátor, označíme ARM a nainštalujeme ovládače, ak ešte nie sú

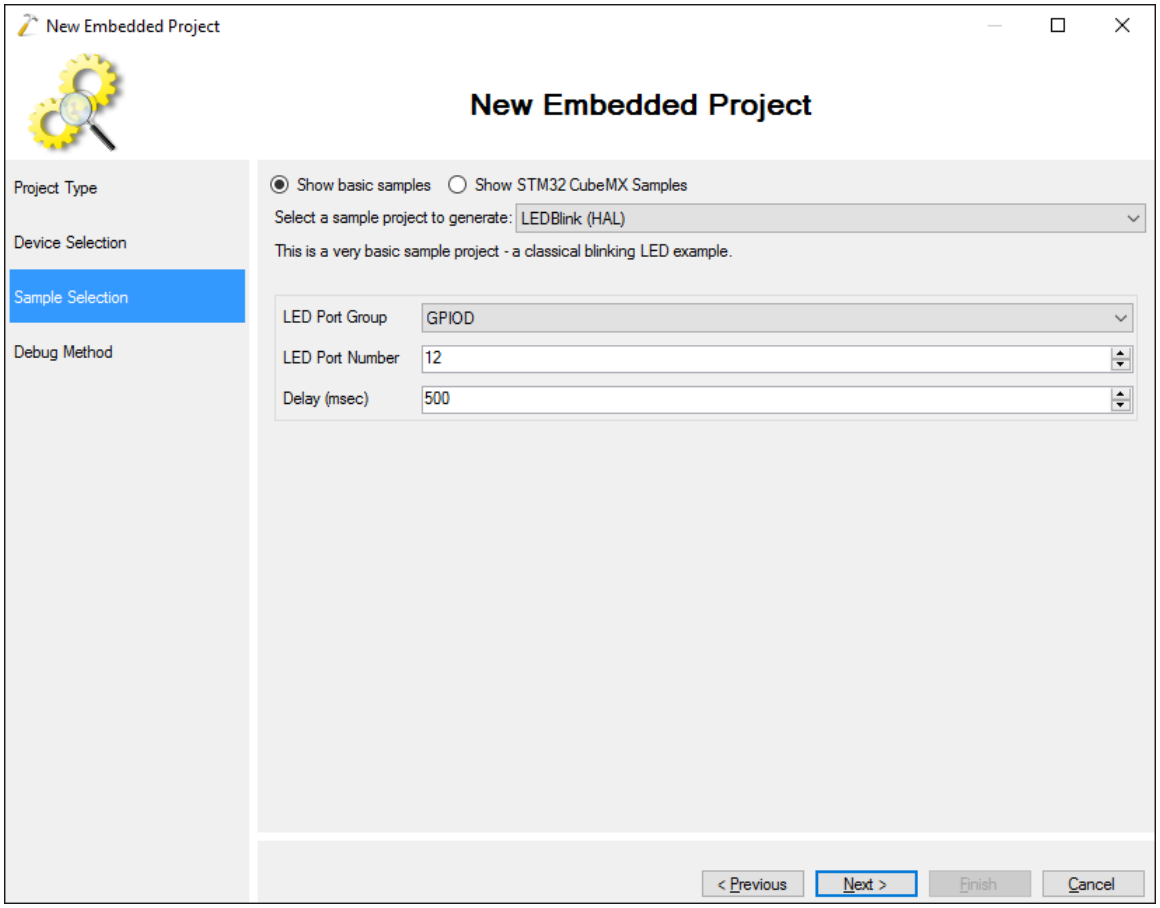

New Embedded Project
New Embedded Project
Project Type
Device Selection
Sample Selection
Debug Method
Please select a toolchain that will be used to build your project:
AVR in C:\SysGCC\avr
GCC 5.3.0, GDB 7.11, Revision 1
ESP32 in C:\SysGCC\esp32
GCC 5.2.0, GDB 7.10, Revision 3; ESP-IDF 2.1
ESP8266 in C:\SysGCC\esp8266
GCC 5.2.0, GDB 7.11, Revision 11; IoT SDK 2.0.0; RTOS SDK 1.5.0; OTA; NodeMCU
MSP430 in C:\SysGCC\msp430-elf
GCC 5.9.0, GDB 7.7, Revision 1
(download toolchain for arm-eabi)
GCC 6.2.0, GDB 7.12,
(download toolchain for bfin-elf)
GCC 4.6.3, GDB 7.4,
(download toolchain for m32c-elf)
GCC 4.7.3, GDB 7.7,
(download toolchain for m32r-elf)
GCC 4.8.0, GDB 7.5.1,
(download toolchain for m68k-elf)
GCC 4.8.0, GDB 7.5.1,
(download toolchain for msp430)
GCC 4.6.3, GDB 7.4,
(download toolchain for powerpc-eabi)
GCC 4.9.0, GDB 7.7,
Select a third-party toolchain by locating gdb.exe...
Download more toolchains...
AT91SAM7X256 64K 256K downloadable
AT91SAM7X512 128K 512K downloadable
AT91SAM7XC128 32K 128K downloadable
AT91SAM7XC256 64K 256K downloadable
Please select a device
< Previous
Next >
Finish
Cancel


New Embedded Project
New Embedded Project
Project Type
Device Selection
Sample Selection
Debug Method
Please select a toolchain that will be used to build your project:
ARM in C:\SysGCC\arm-eabi
Select a device from list
Specify flags manually
Use ARM mbed
Download more devices
Devices by name
Devices by type
Supported devices:
Filter: f407
Device RAM size ROM size State
STM32F407IE 192K 512K installed
STM32F407IG 192K 1024K installed
STM32F407VE 192K 512K installed
STM32F407VG 192K 1024K installed
STM32F407ZE 192K 512K installed
BSP Version 4.3
Execute from FLASH
Floating point support Software
C Library Type Newlib-nano
Reduce the size of C++ binaries
Provide default stubs for system calls
Shared files location: C:\Users\virtual\AppData\Local\VisualGDB\EmbeddedBSPs\arm-eabi\com.sysprogs.arm.
Change
< Previous
Next >
Finish
Cancel

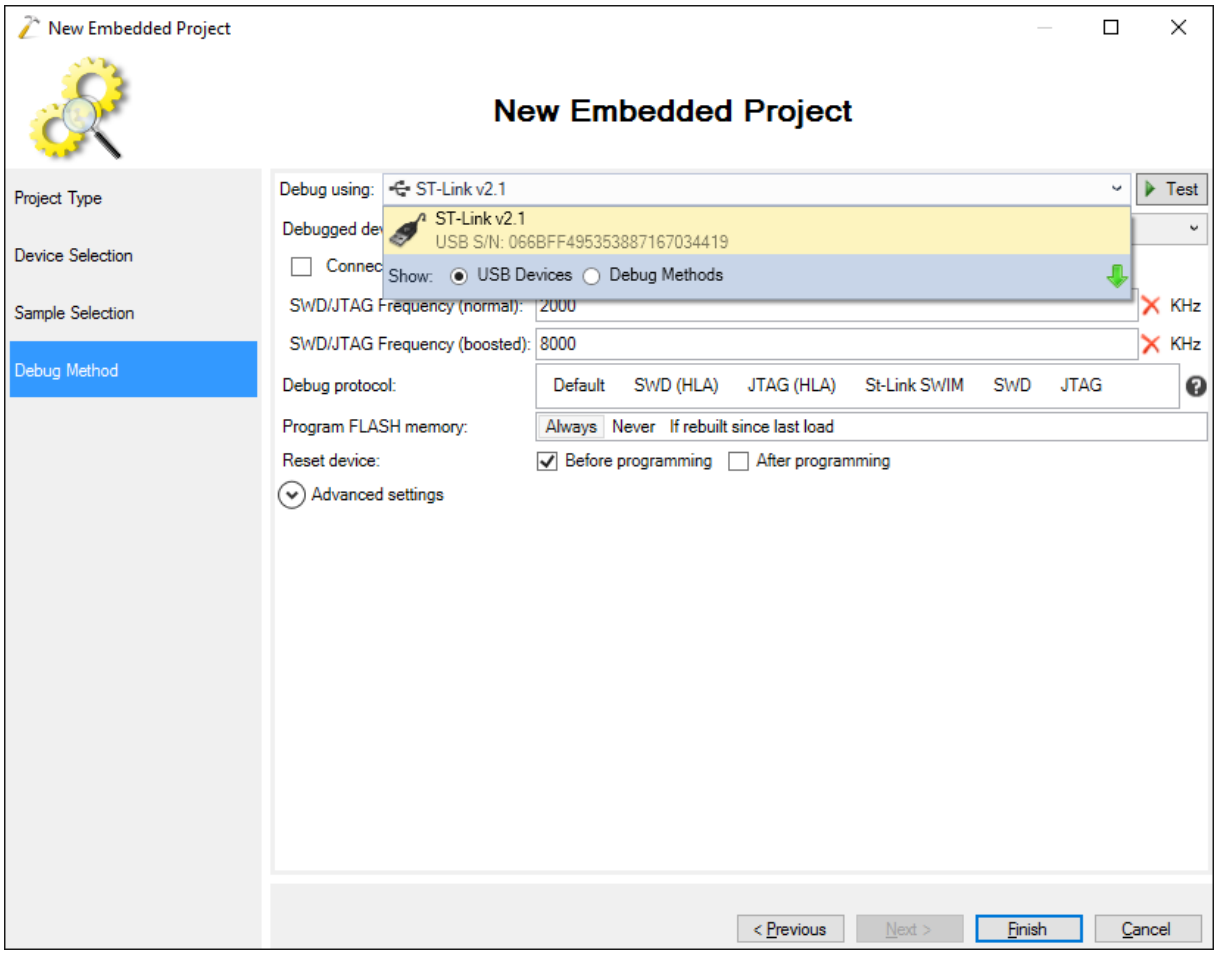
5. STM32F4 doska obsahuje 4 LED pripojené na piny PD12 až PD15, pozri schéma



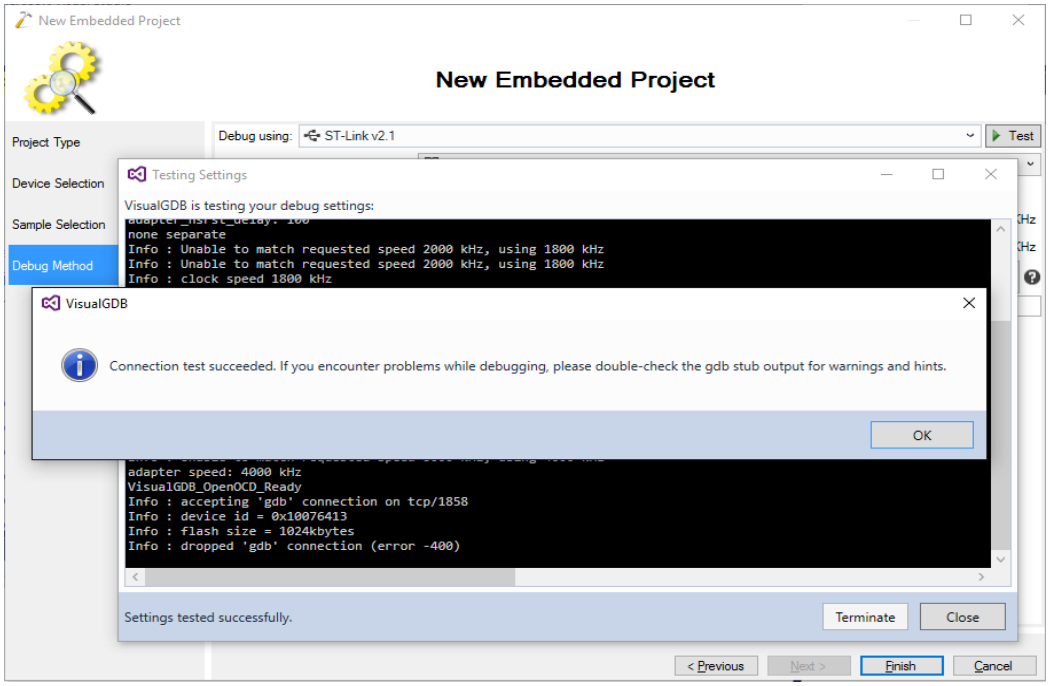
V tomto projekte rozblikáme zelenú LED, pripojenú na PD12. Nastavíme GPIO a 12 ako port a číslo.



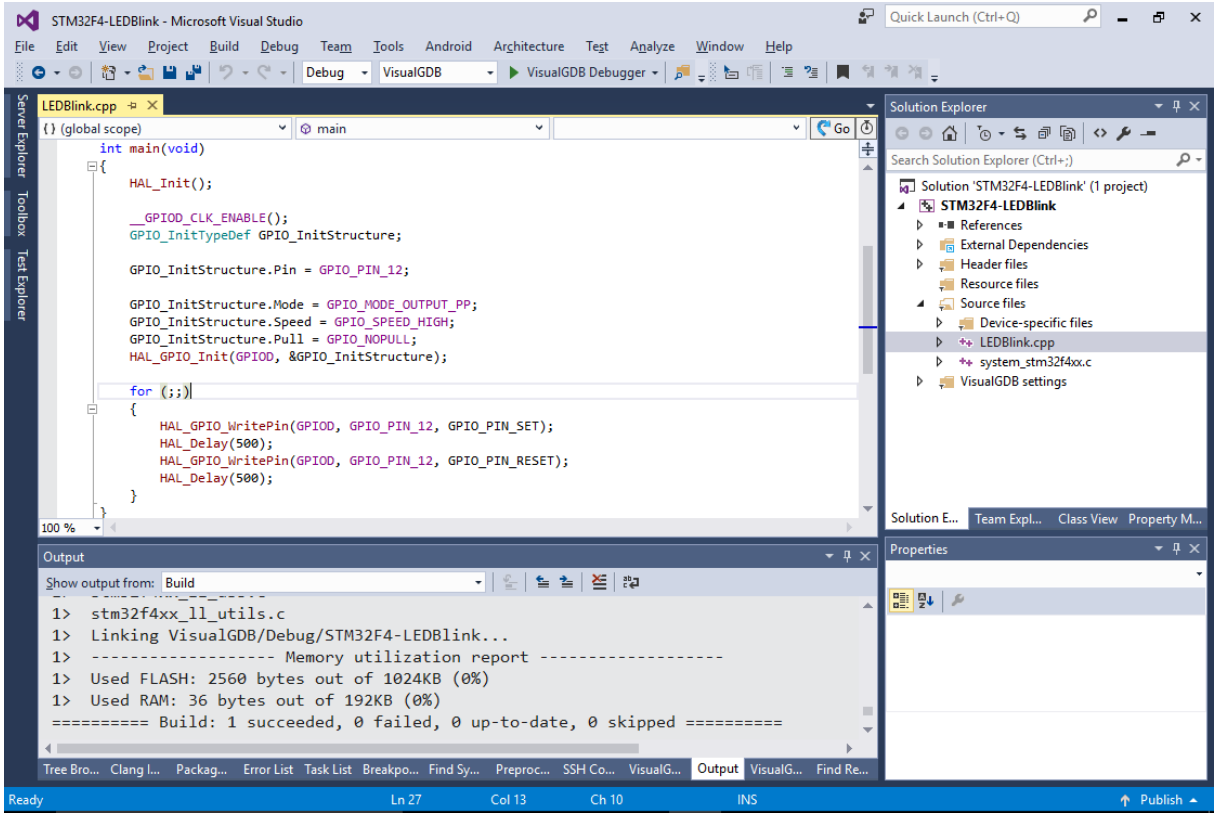
6. STM32F4 doska obsahuje vstavený ST-Link programátor. Ak pripojíme dosku, VisualGDB ju automaticky rozpozná. Vyberieme programátor a spustíme test.



VisualGDB sa pokúsi spustiť OpenOCD a pripojí sa, kde potvrdí, že debugovanie bude možné.

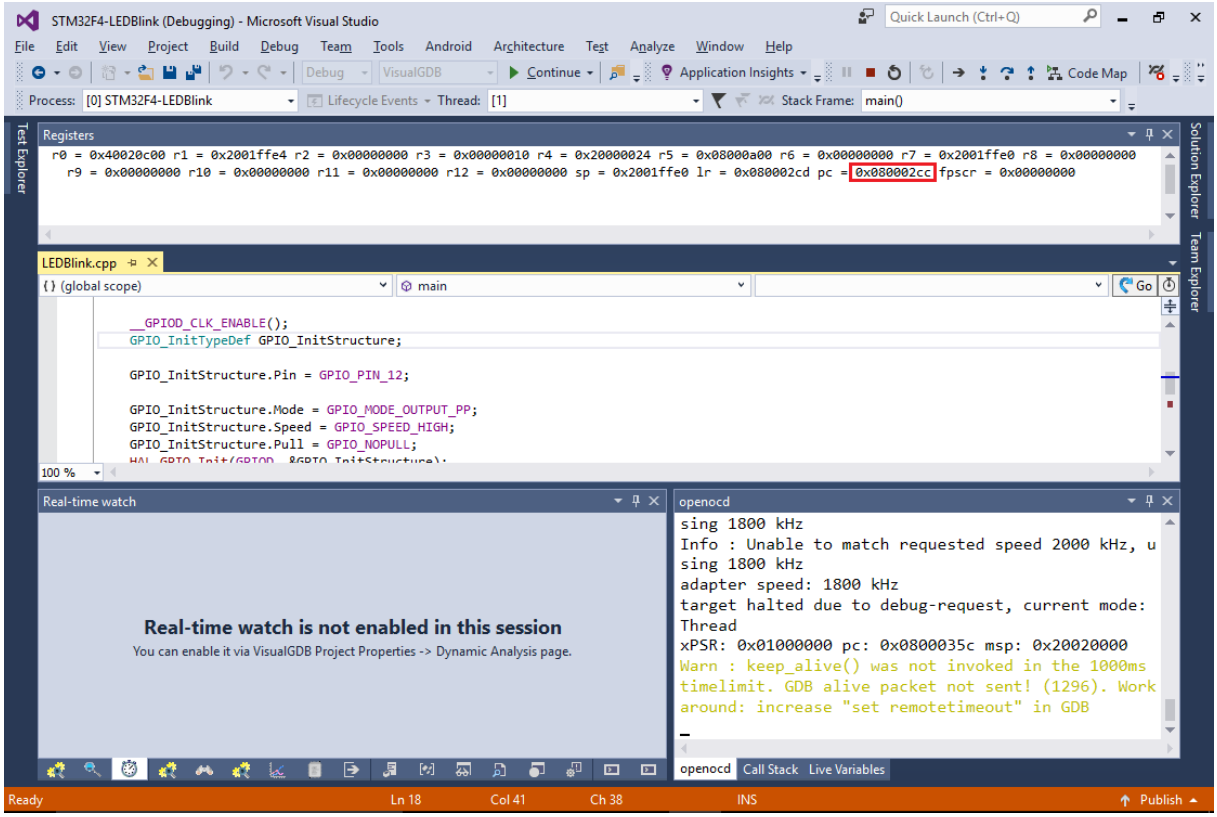


7. Stlačíme Finish. Vytvorí sa projekt. Ctrl-Shift-B a projekt sa „buildne“

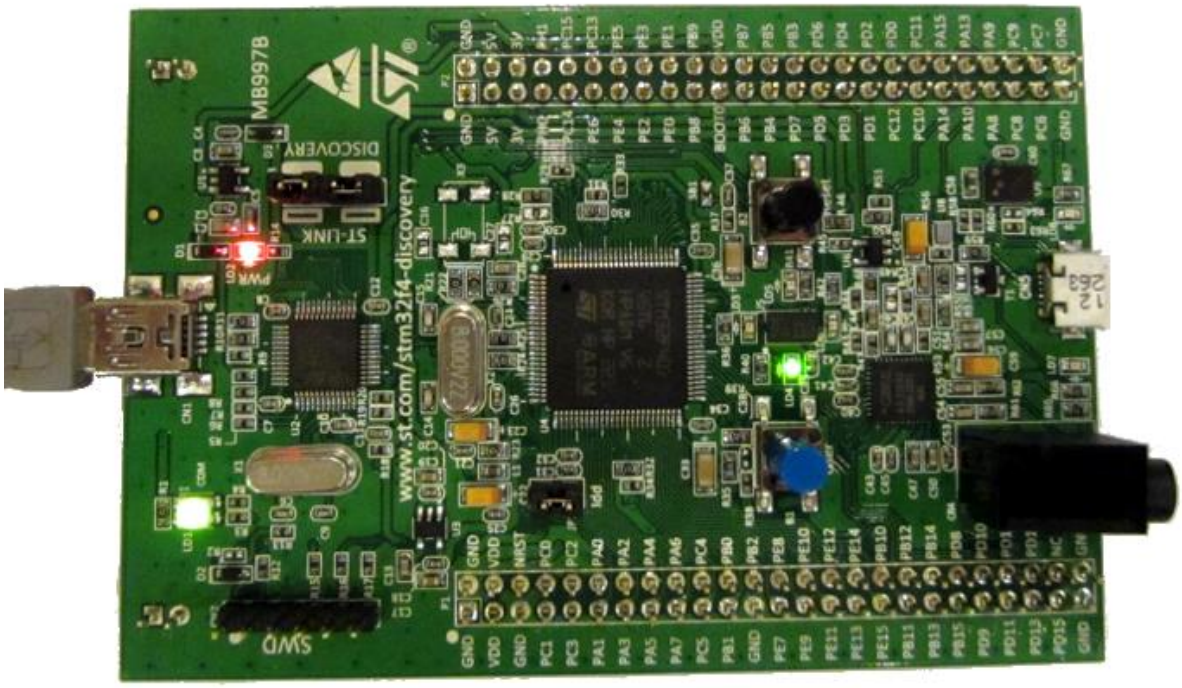


8. Stlačíme F5 a spustíme debuggovanie. Sledujme ako LED bliká. Nastavíme breakpoint na **GPIO_WriteBit() riadok.** Breakpoint sa spustí automaticky a ledka sa vypne.

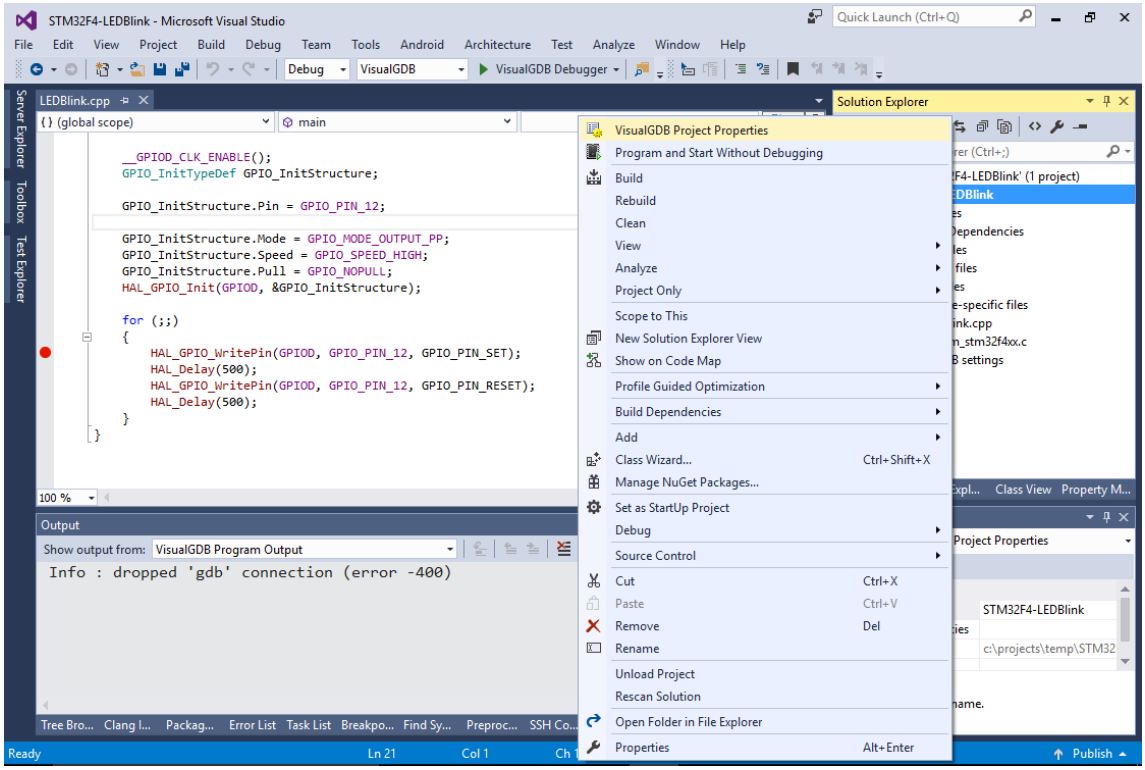
9. Ctrl-Shift-G otvoríme okno registrou.



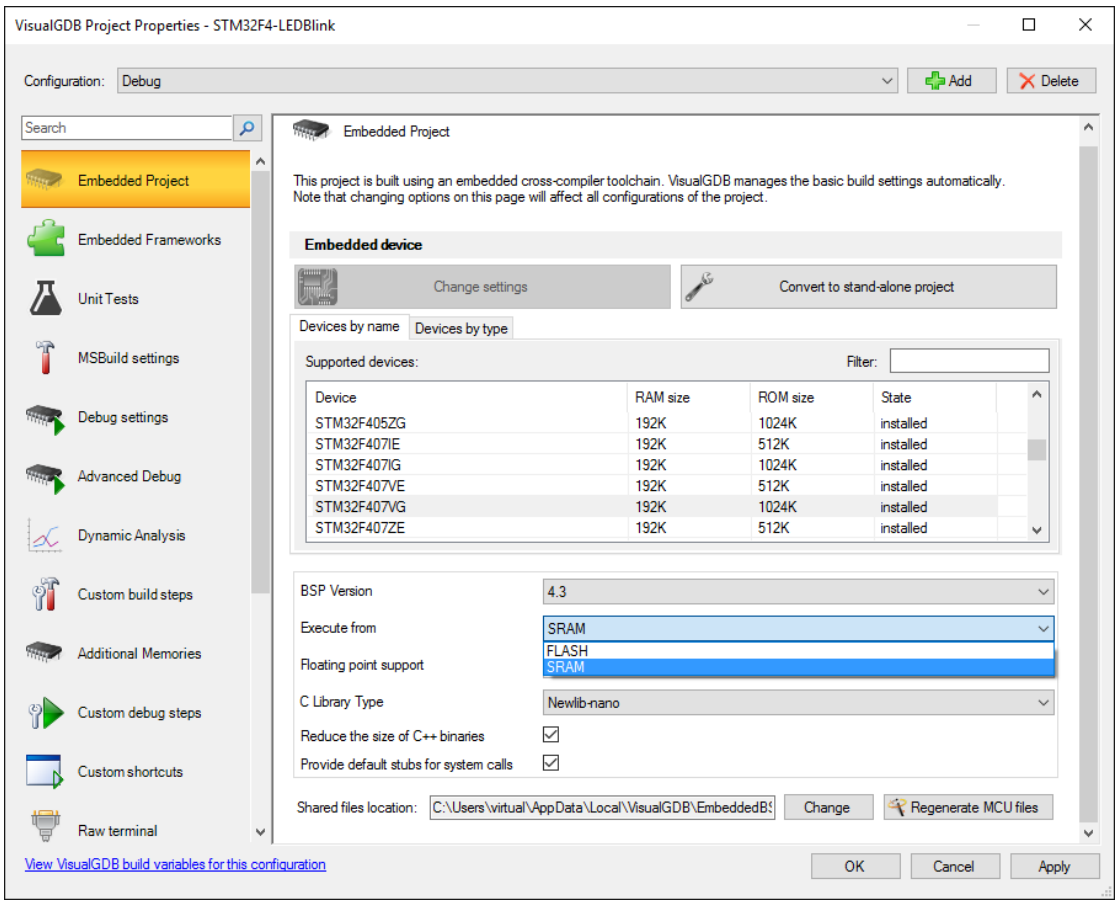
10. Hodnota PC registra 0x80xxxxxx indikuje, že program je spustený z flash pamäte. Stlačme F10 sa posunieme o jeden riadok v kóde. Teraz by LED mala svietiť.



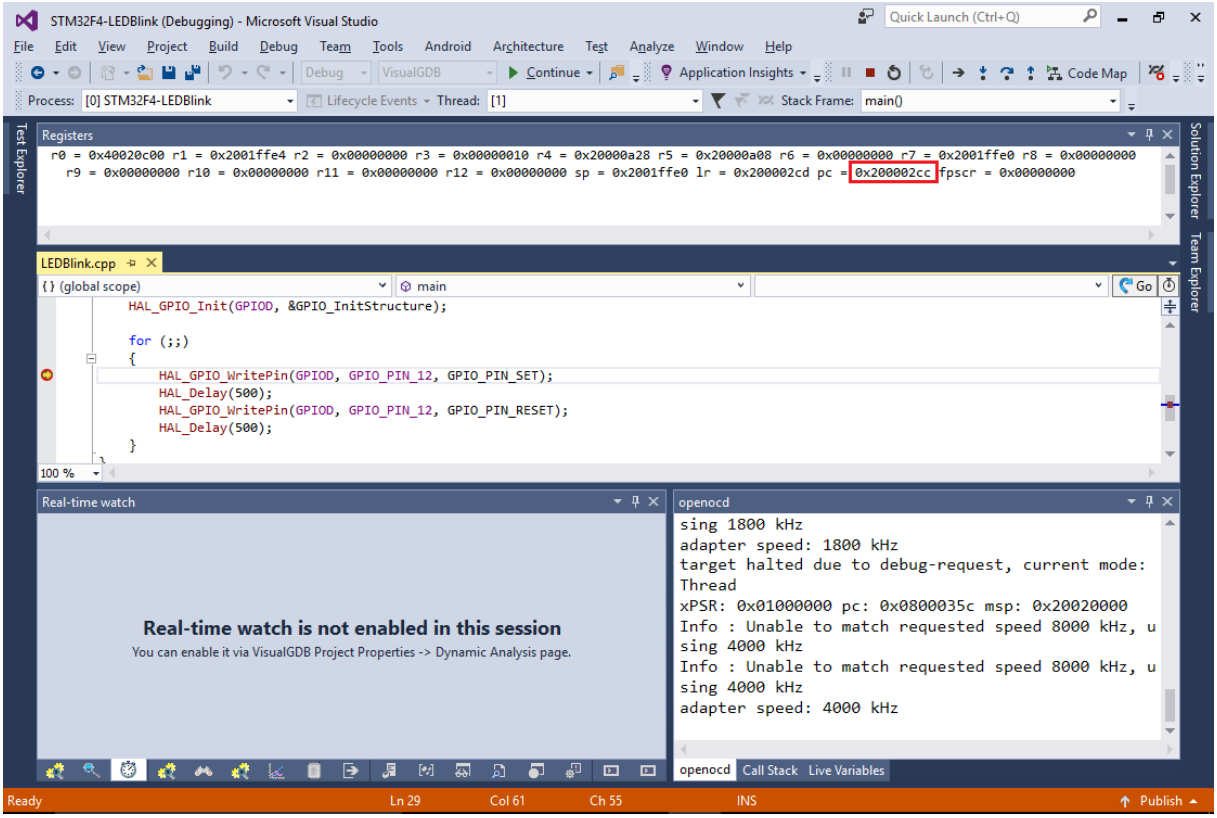
11. Debuggovanie ukončíme pomocou Shift-F5. Pravý click na projektový uzoľ v Solution Explorery a stlač „VisualGDB Project Properties“.



12. Clicknime “Change settings” and nastavme SRAM namiesto of FLASH



13. Nakoľko predošlí projekt bol nahraný vo Flash pamäti, budeme musieť znovu rebuildnúť projekt. Zopakujeme predošlé kroky, a pozrieme hodnotu v PC registry, ak je v rozsahu 0x20xxxxxx, tak vieme, že program bol nahraný do SRAM.



Chod programu z SRAM nám neskracuje FLASH prepisovacie cykly, ako keď nahrávame pamäť alebo používame breakpointy. Avšak používanie SRAM má svoje nevýhody. Väčšie programy sa nemusia zmestiť do SRAM, takže pre veľké programy s veľa premenami to nemusí byť najvhodnejšie riešenie.

Nakoniec si pozrieme View->Embedded Memory Explorer, aby sme videli, ktoré konkrétne funkcie sú zodpovedné za používanie pamätí.