

## ***Instrukcja uruchomienia projektu***

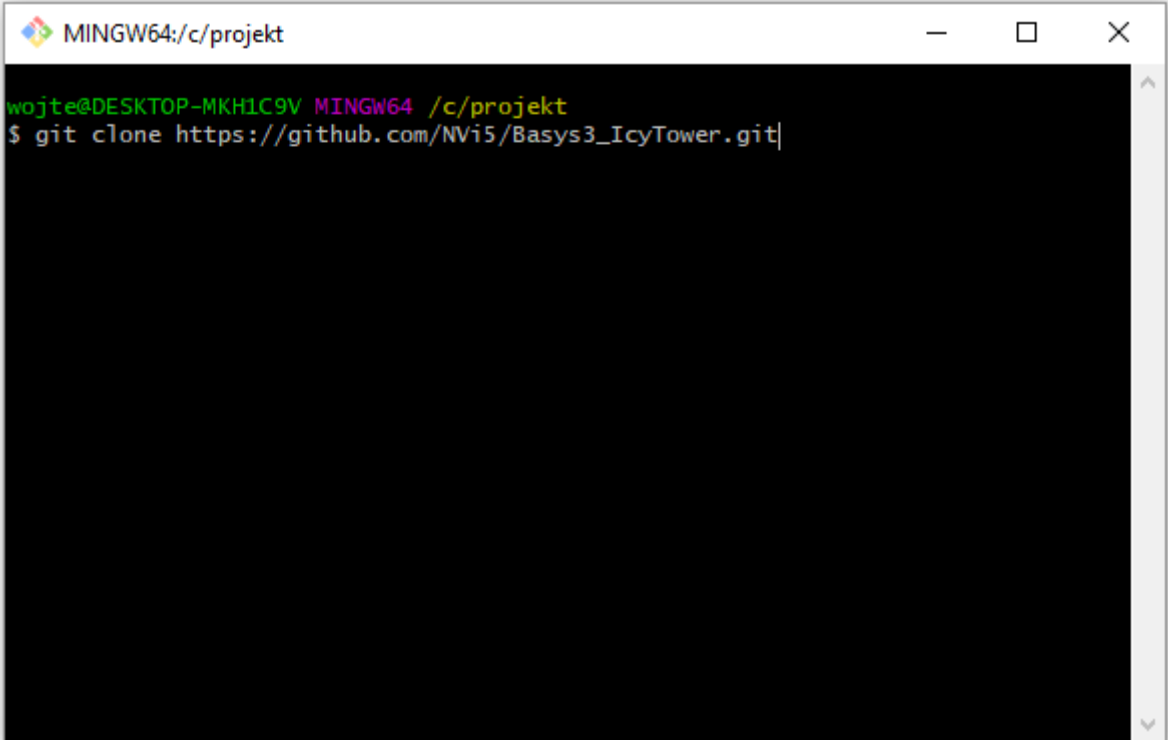
1. Tworzymy folder, do którego klonowane będzie repozytorium projektu.  
np. C:/projekt.

**Uwaga!** – ścieżka dostępu nie może zawierać spacji ani żadnych niestandardowych znaków, spowoduje to nieprawidłowe działanie Eclipse SDK.

2. Klonujemy repozytorium do utworzonego folderu.

Adres repozytorium:

[https://github.com/NVi5/Basys3\\_IcyTower.git](https://github.com/NVi5/Basys3_IcyTower.git)

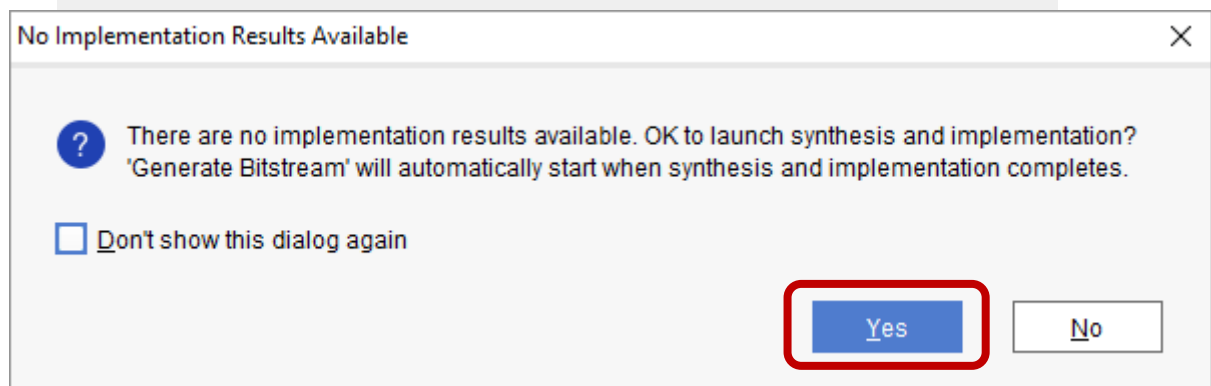
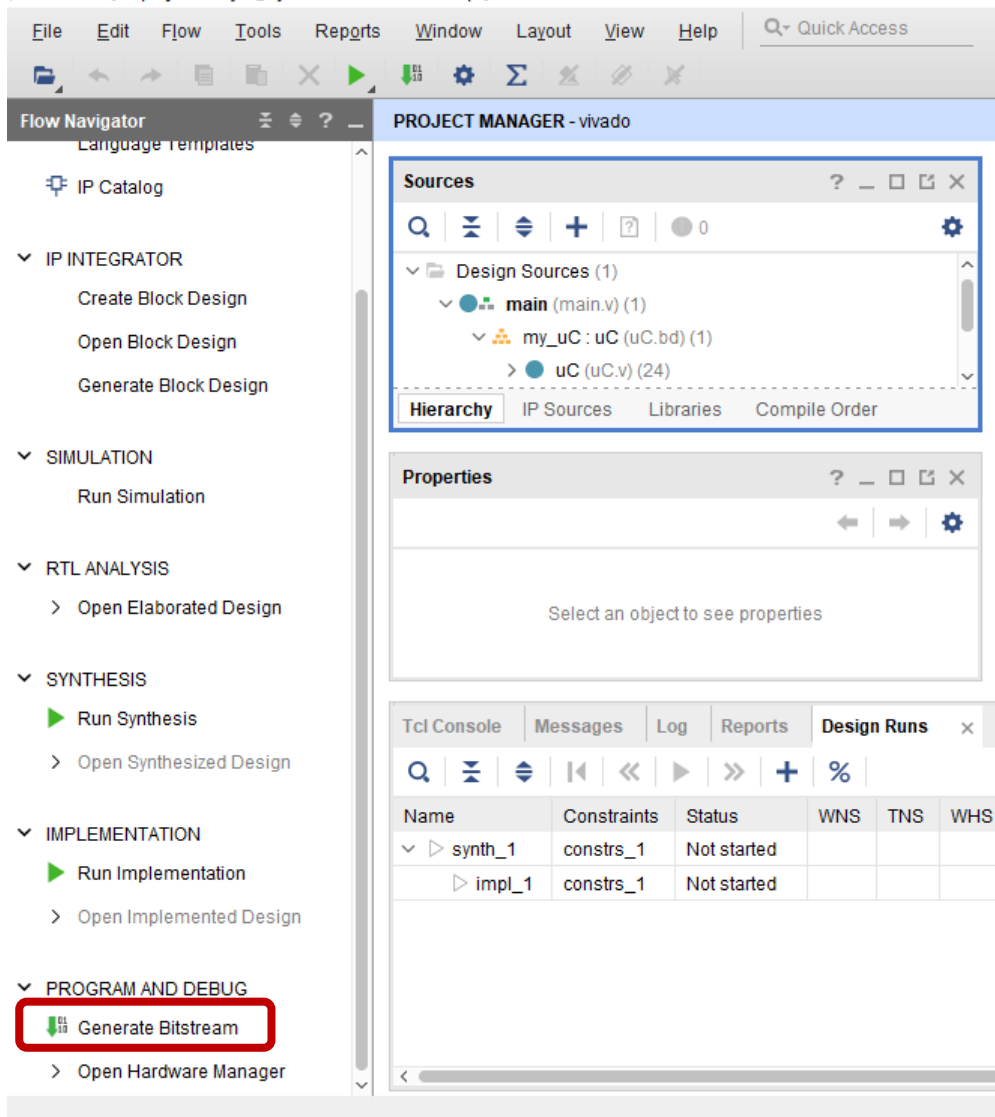


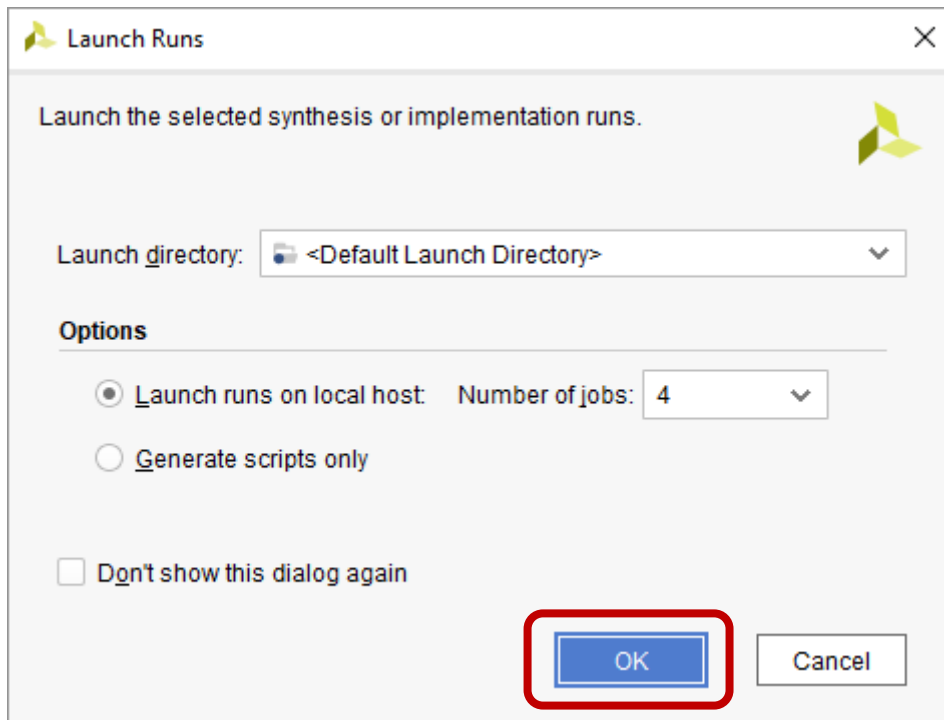
A screenshot of a MINGW64 terminal window. The title bar shows 'MINGW64:/c/projekt'. The terminal content shows the prompt 'wojte@DESKTOP-MKH1C9V MINGW64 /c/projekt' followed by the command '\$ git clone https://github.com/NVi5/Basys3\_IcyTower.git' which is currently being entered.

3. Po sklonowaniu repozytorium przechodzimy do katalogu **./Basys3\_IcyTower/vivado** i otwieramy plik **vivado.xpr**, następnie czekamy na uruchomienie środowiska Vivado.

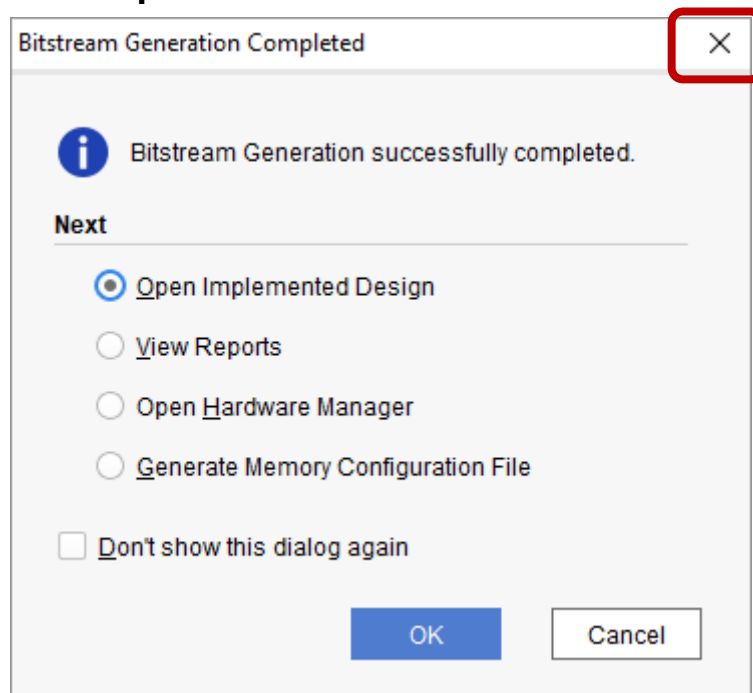
4. Po uruchomieniu Vivado wybieramy opcję Generate Bitstream, z domyślnymi ustawieniami (może to zająć ponad 40 minut).

vivado - [C:/projekt/Basys3\_IcyTower/vivado/vivado.xpr] - Vivado 2018.2

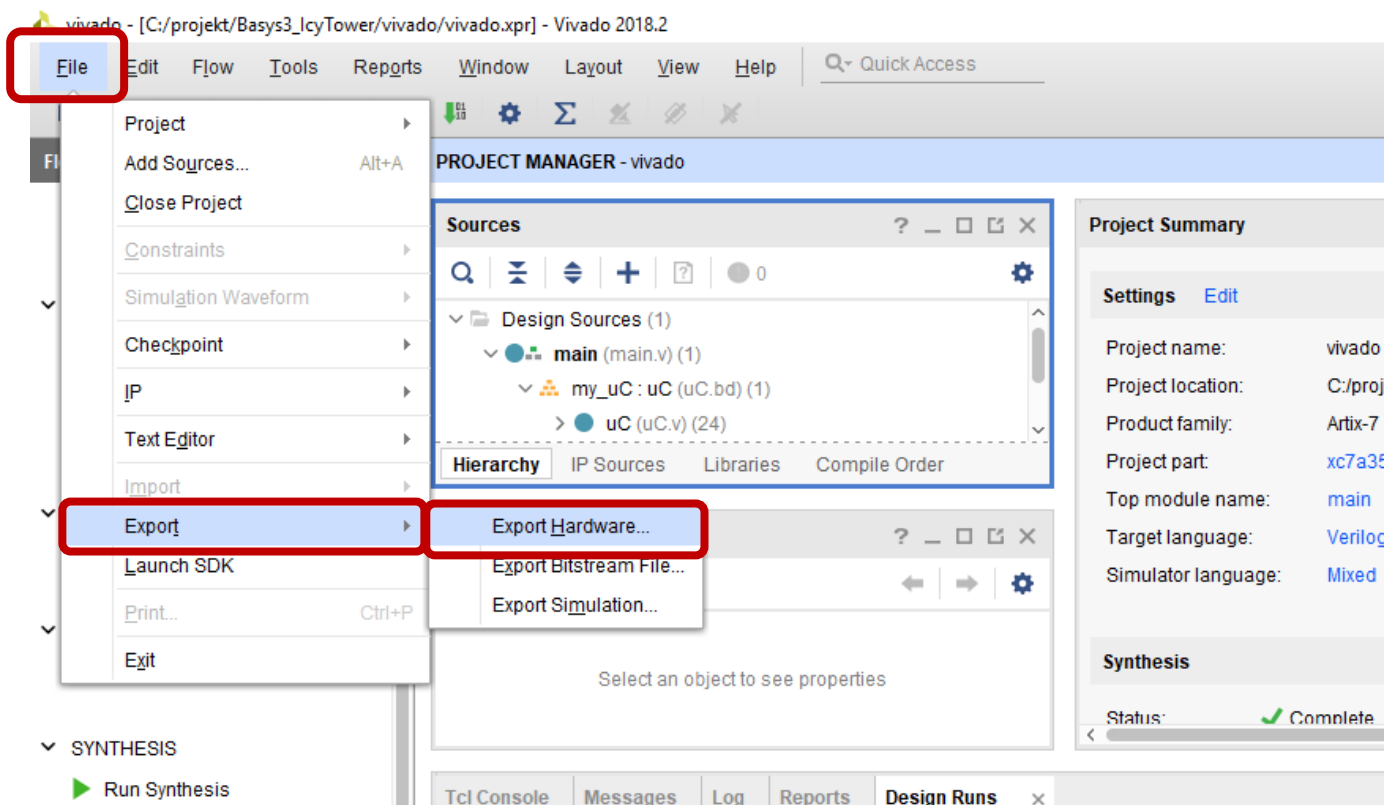




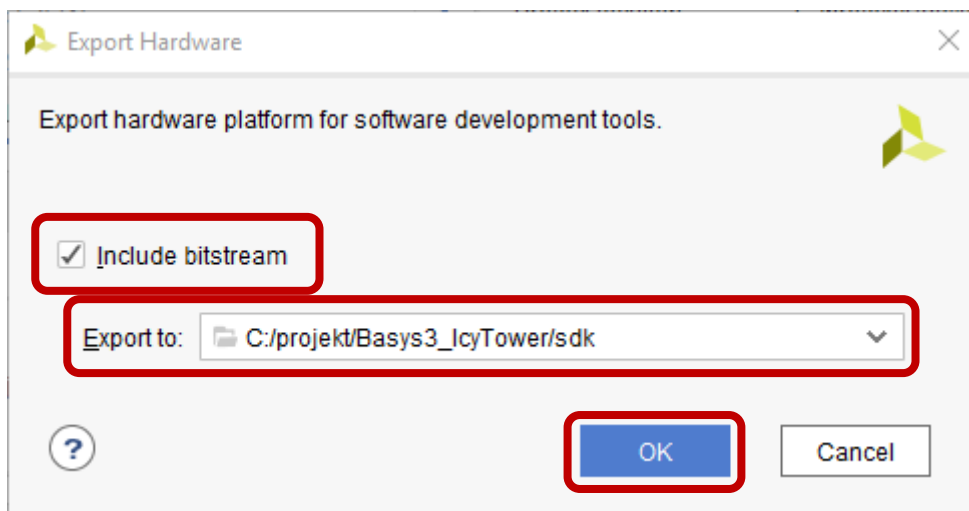
5. Po zakończeniu generowanie bitstreamu **zamykamy okno Bitstream Generation Completed.**



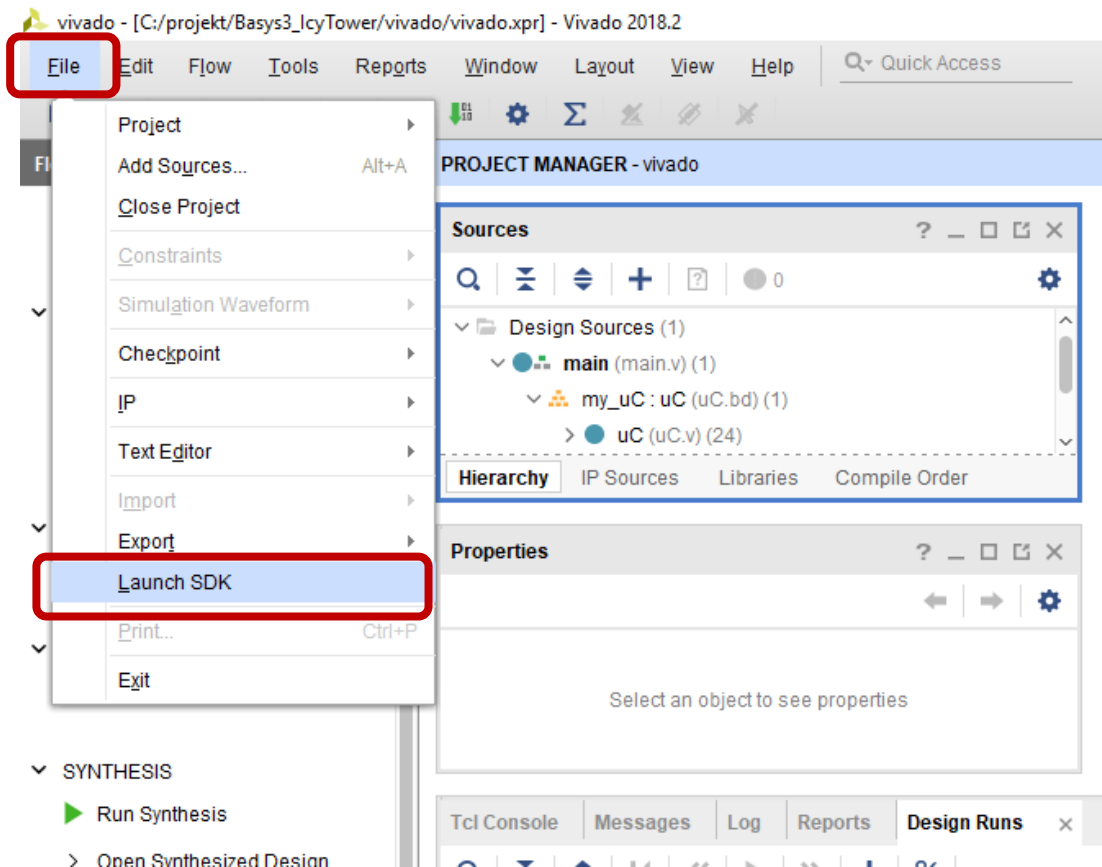
6. W Vivado wybieramy opcję **File > Export > Export hardware.**



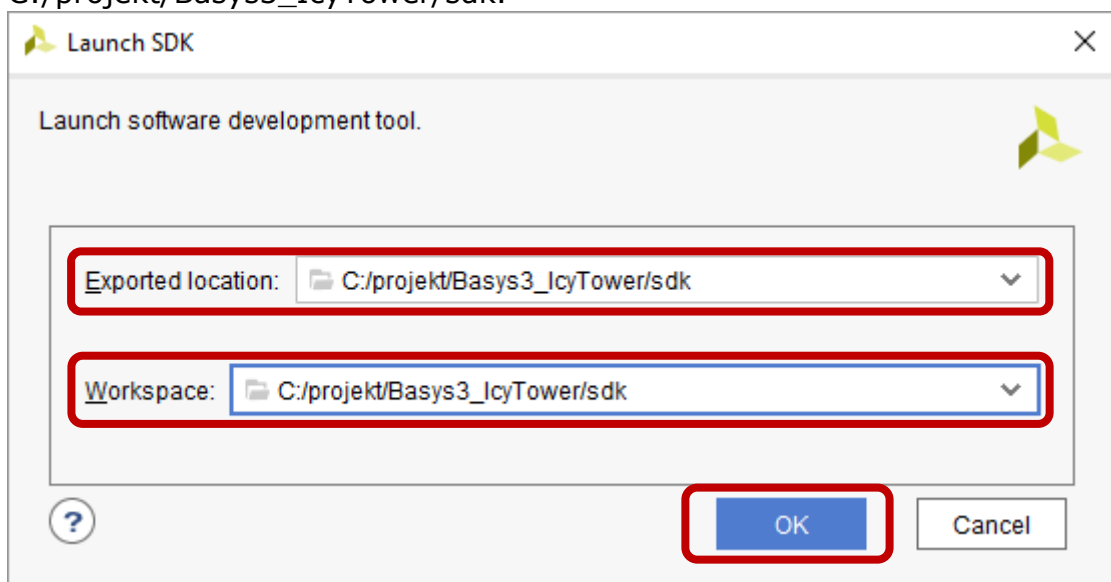
7. W oknie, które się pojawi wybieramy opcję **Include bitstream** i w polu **Export to:** wybieramy Choose Location i wskazujemy folder `${Lokalizacja_projektu}/sdk`, np. `C:/projekt/Basys3_IcyTower/sdk`.



8. W vivado wybieramy opcję **File > Launch SDK**.

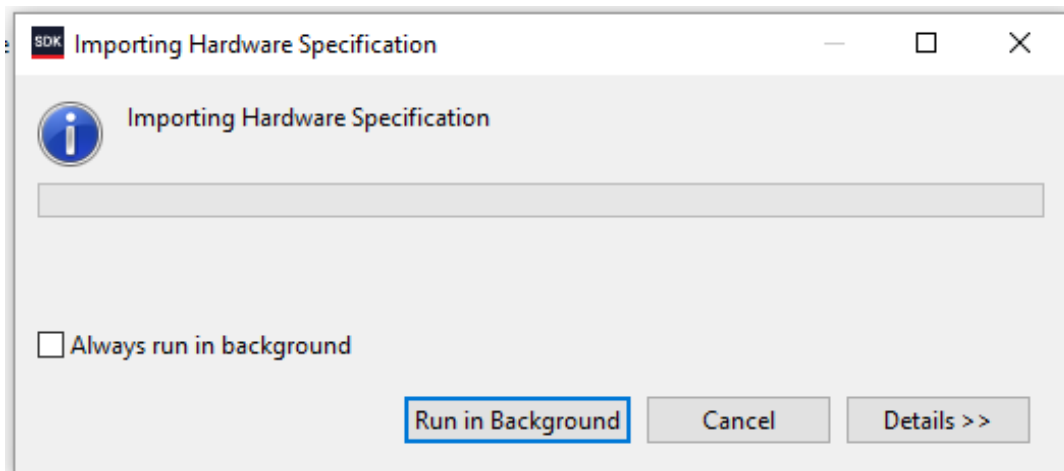


9. W oknie, które się pojawiło w obu polach wybieramy Choose Location i wybieramy lokalizację `${Lokalizacja_projektu}/sdk`, np. `C:/projekt/Basys3_IcyTower/sdk`.

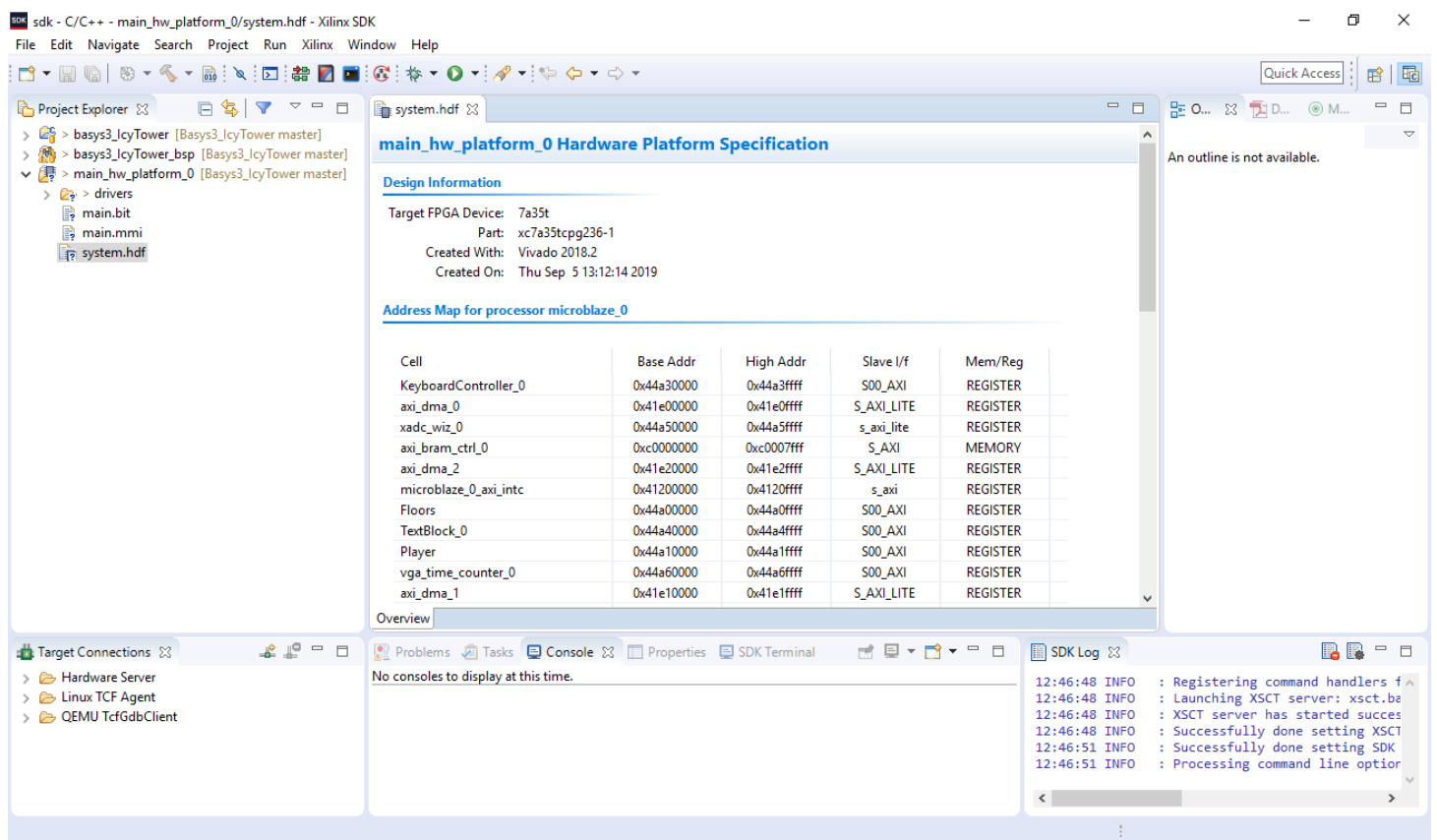


10. Czekamy na załadowanie się SDK.

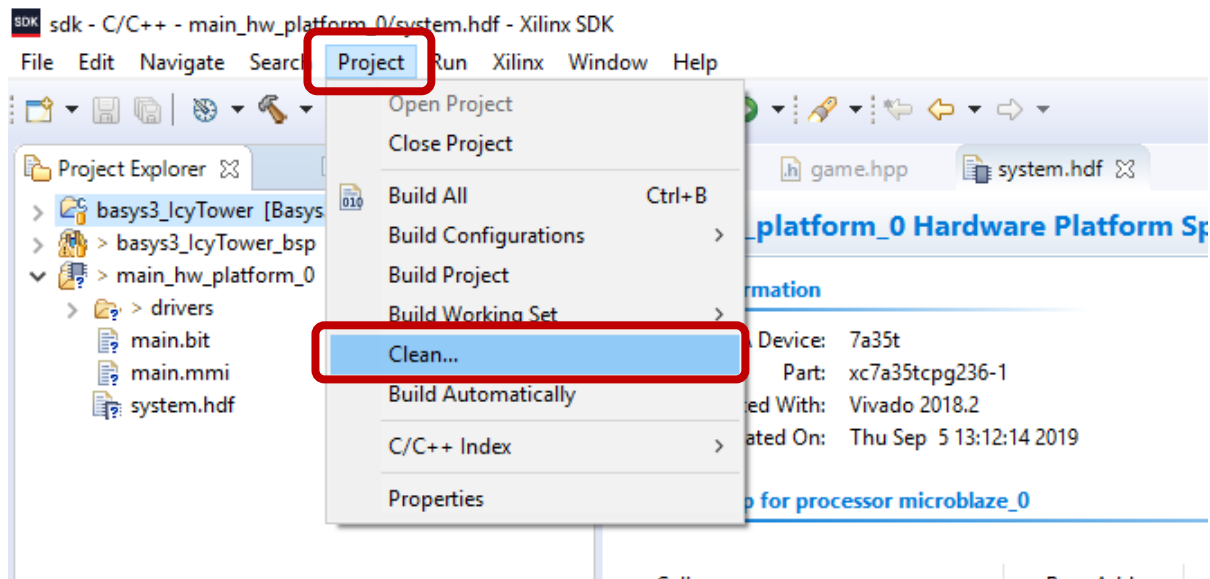
11. Czekamy na pojawienie się poniższego okna i zaimportowanie specyfikacji sprzętowej.



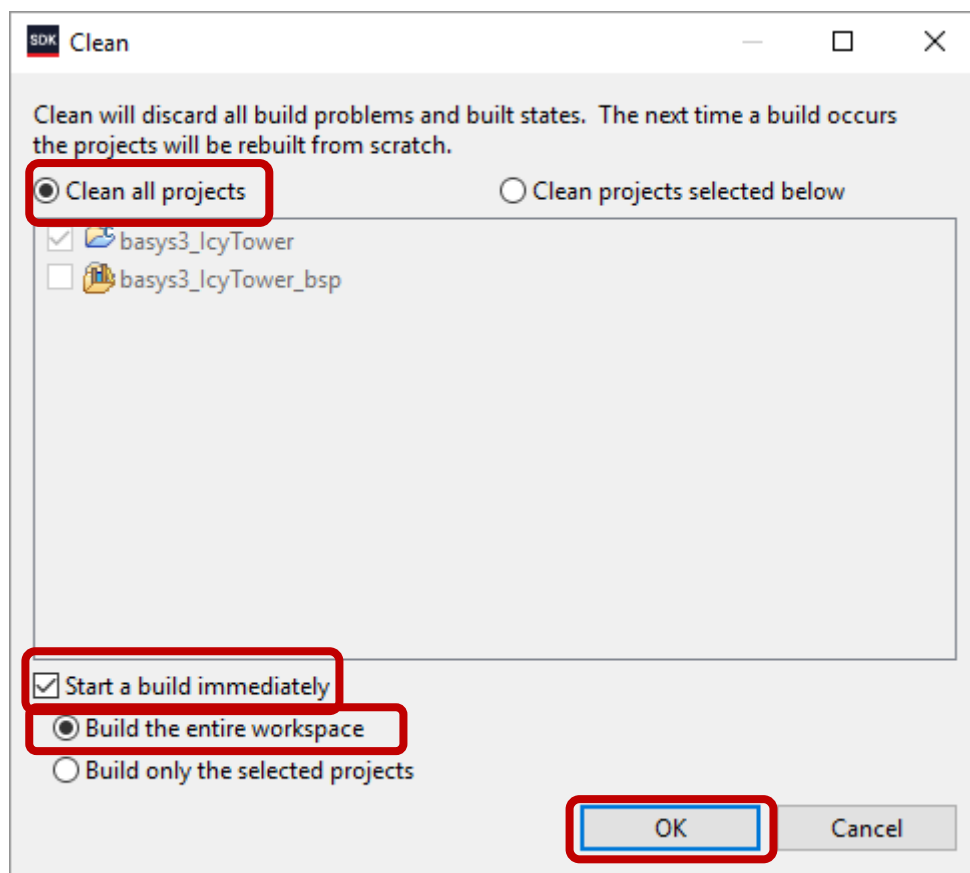
12. Po zakończeniu wszystkich operacji powinniśmy uzyskać taki widok w SDK



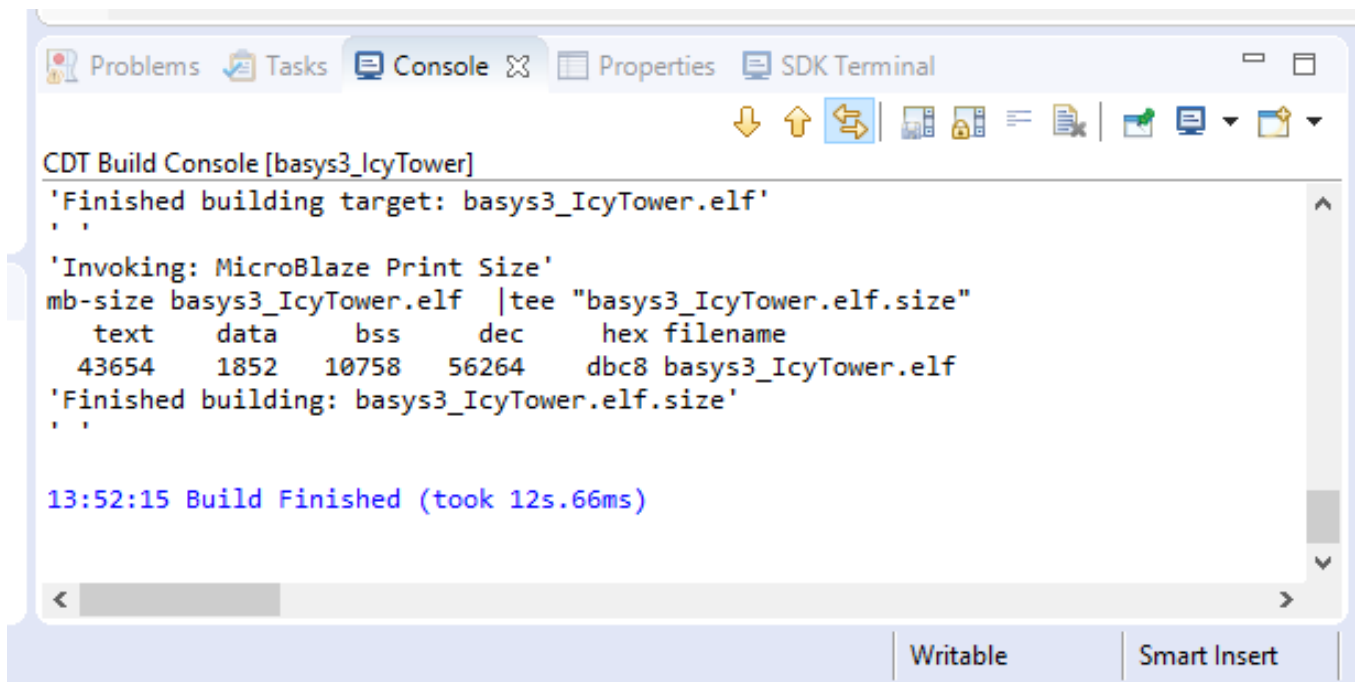
13. Wybieramy opcję **Project > Clean**.



14. W oknie, które się pojawi wybieramy opcje: **Clean all projects**, **Start a build immediately** i **Build the entire workspace**.



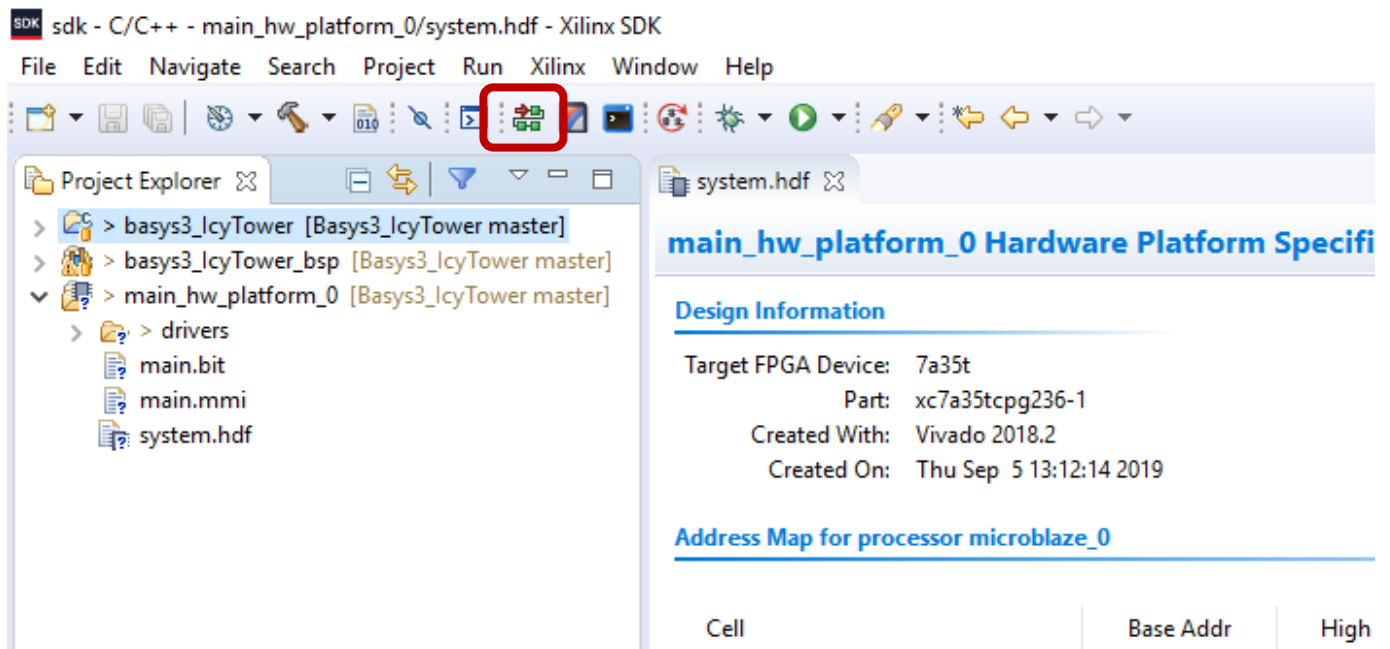
15. Po zakończeniu, w oknie Console powinniśmy zobaczyć następujący komunikat:



```
CDT Build Console [basys3_IcyTower]
'Finished building target: basys3_IcyTower.elf'
'Invoking: MicroBlaze Print Size'
mb-size basys3_IcyTower.elf |tee "basys3_IcyTower.elf.size"
  text    data    bss    dec    hex filename
  43654    1852   10758   56264   dbc8 basys3_IcyTower.elf
'Finished building: basys3_IcyTower.elf.size'

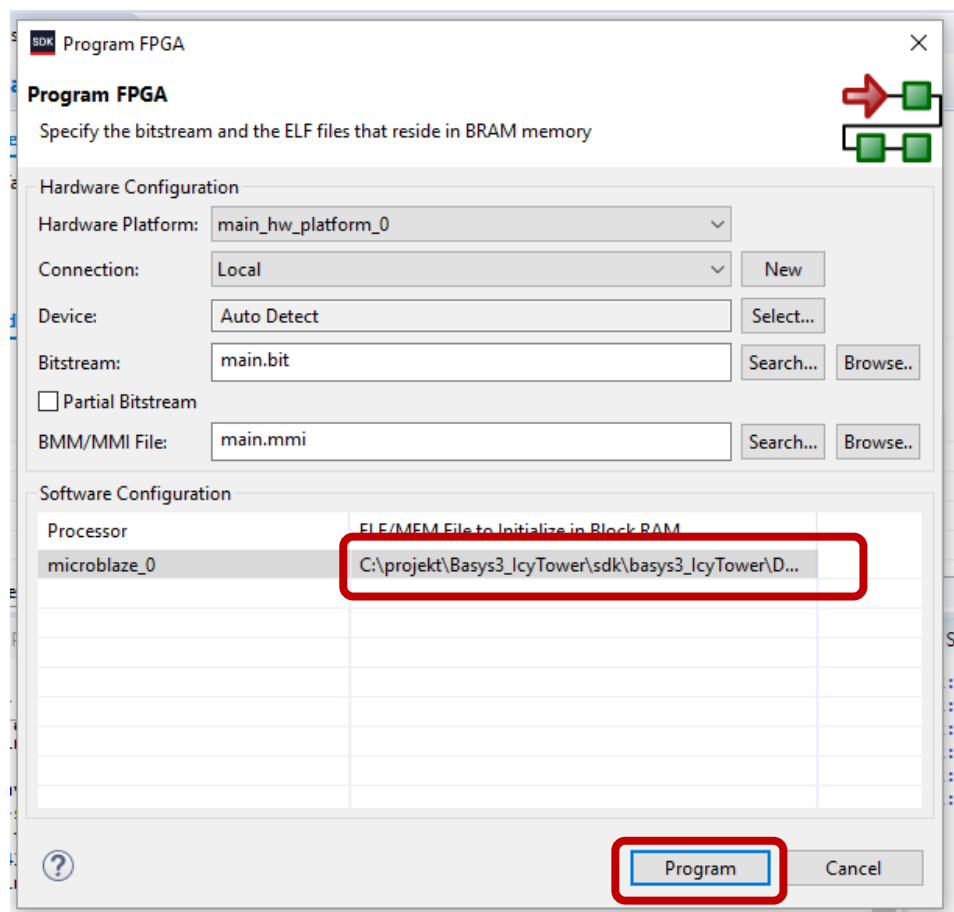
13:52:15 Build Finished (took 12s.66ms)
```

16. W tym momencie podłączamy płytkę Basys 3 do komputera i czekamy na jej wykrycie przez system. Następnie w SDK wybieramy opcję **Program FPGA**.



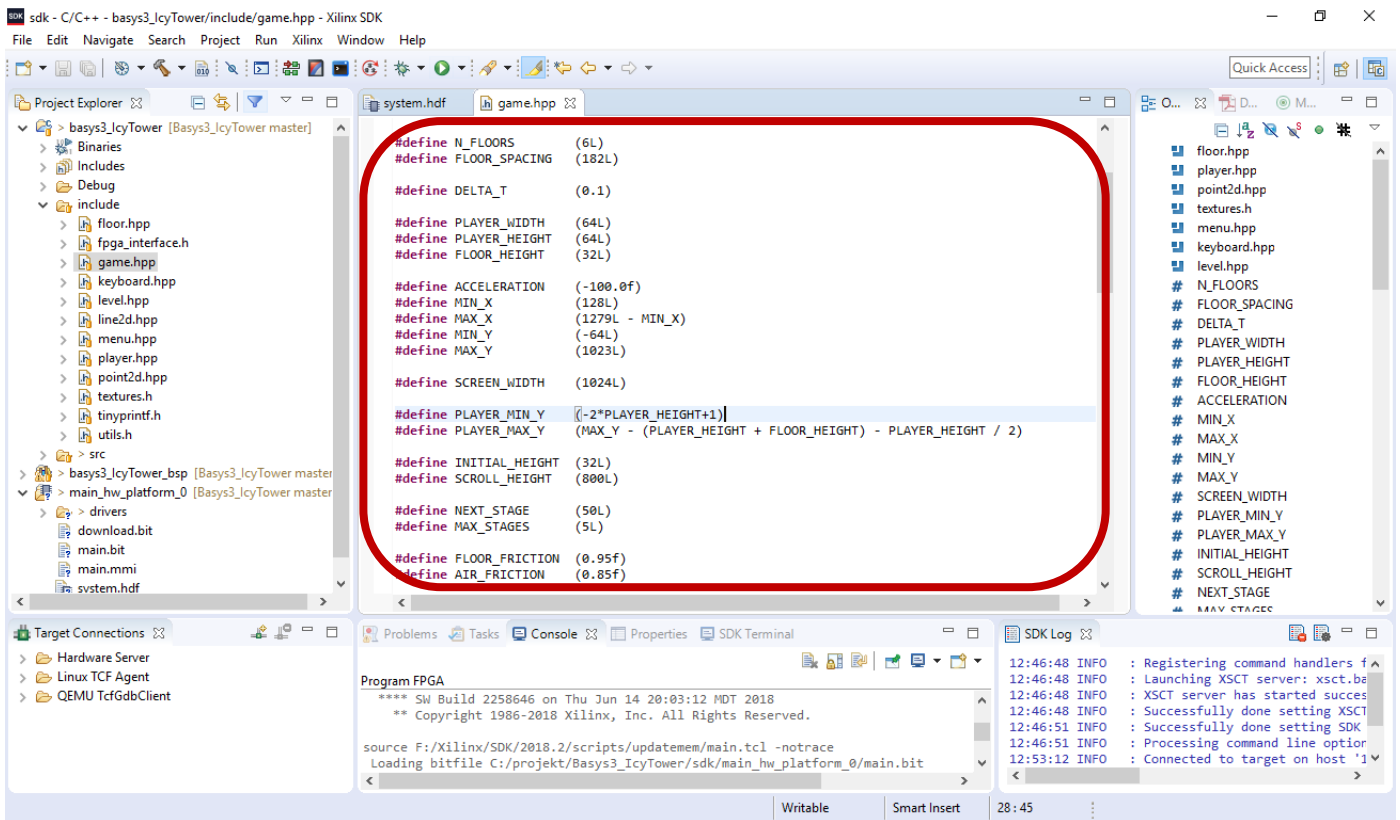


17. W oknie, które się pojawi konfigurujemy opcje tak jak na obrazku poniżej i w polu **EMI/ELF file to initialize in Block RAM** wybieramy wcześniej wygenerowany plik ELF (powinien pojawić się na rozwijanej liście).



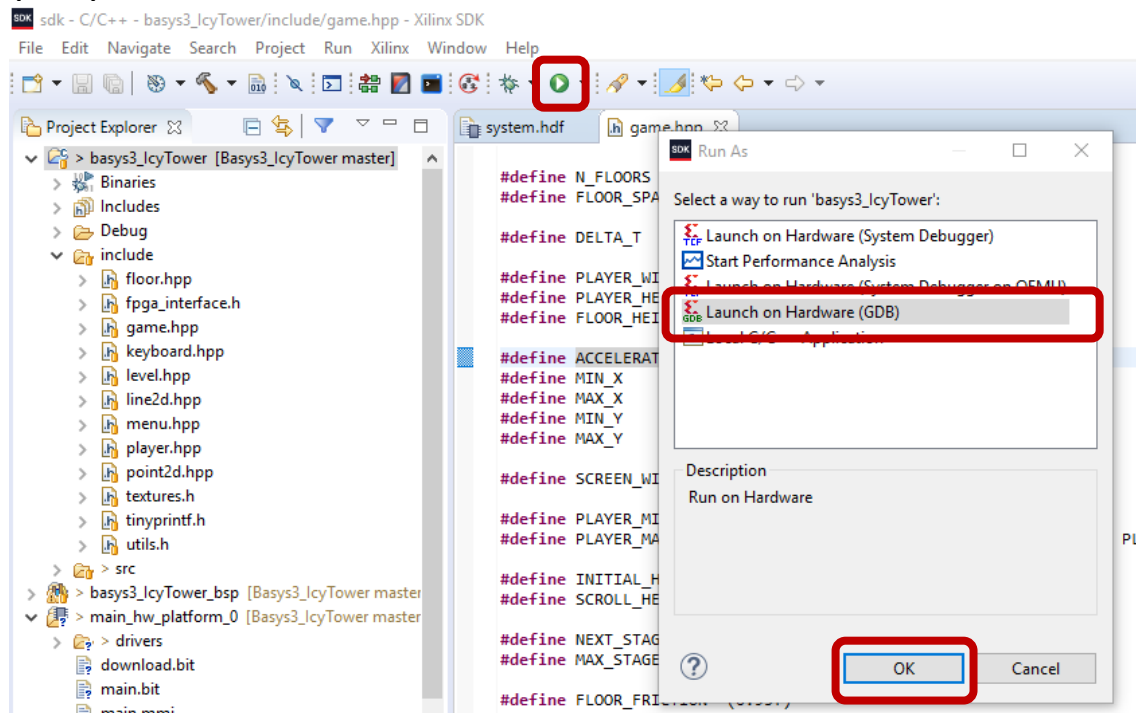
18. Po zakończeniu programowania, Płytką Basys 3 jest gotowa do podłączenia monitora, klawiatury i rozpoczęcia gry.

19. Dodatkowo w oknie projektu możemy otworzyć plik Basys3\_IcyTower/include/game.hpp, i pozmieniać parametry gry.



20. Po zakończeniu zmian naciskamy Ctrl + s, aby zapisać i Ctrl + b, aby skompilować projekt.

21. Po skompilowaniu możemy zaktualizować program w pamięci procesora wybierając opcję **Run** i w oknie które się pojawiło **Launch on Hardware (GDB)**.

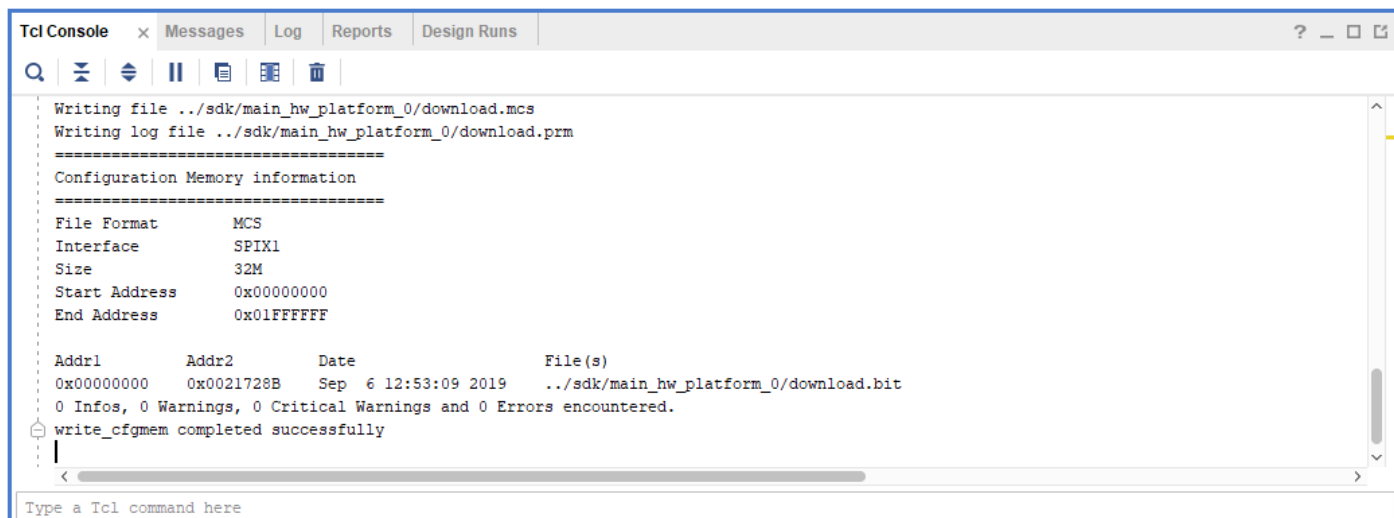


## ***Programowanie pamięci SPI.***

1. Aby zachować konfigurację FPGA i za każdym uruchomieniem płytki Basys 3 nasza gra została załadowana musimy skonfigurować pamięć SPI.  
**Uwaga! – wykonanie poprzedniej części instrukcji jest wymagane do zaprogramowania pamięci.**
2. Przełączamy się na okno Vivado.
3. W oknie Tcl Console wklejamy następującą komendę i wciskamy Enter:

```
write_cfgmem -force -format mcs -size 32 -interface SPIx1 -loadbit "up  
0x0 ../sdk/main_hw_platform_0/download.bit" -file  
../sdk/main_hw_platform_0/download.mcs
```

4. Po pozytywnym zakończeniu operacji zobaczymy komunikat:

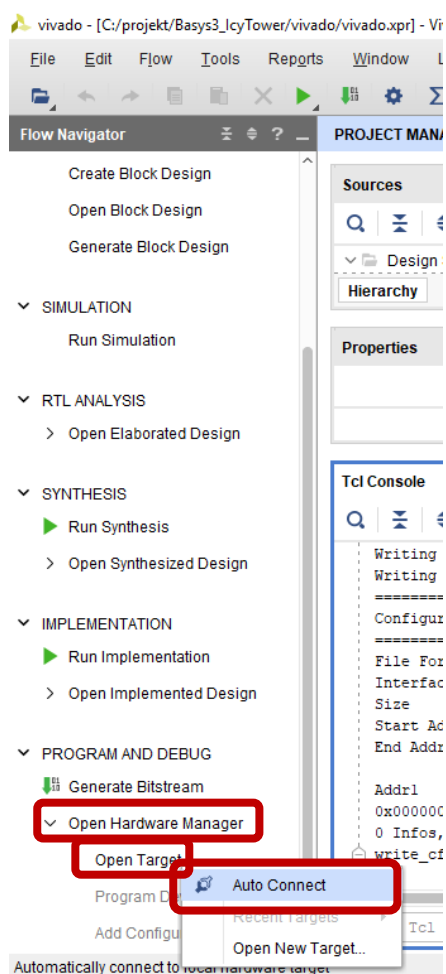


```
Tcl Console x Messages Log Reports Design Runs
Writing file ../sdk/main_hw_platform_0/download.mcs
Writing log file ../sdk/main_hw_platform_0/download.prm
=====
Configuration Memory information
=====
File Format      MCS
Interface       SPIX1
Size            32M
Start Address   0x00000000
End Address     0x01FFFFFF

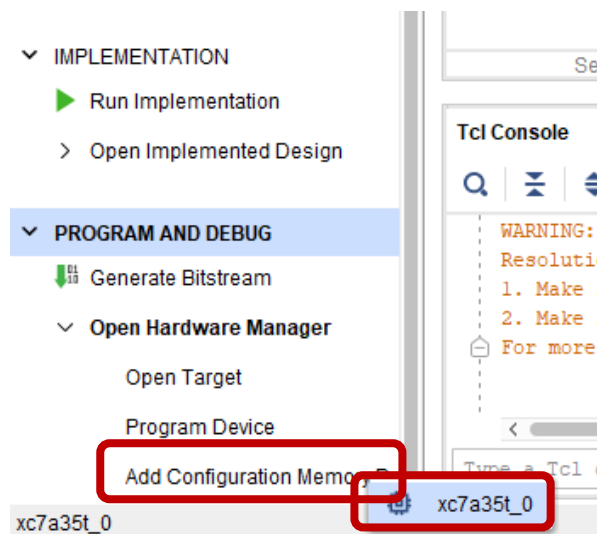
Addr1      Addr2      Date      File(s)
0x00000000 0x0021728B Sep  6 12:53:09 2019  ../sdk/main_hw_platform_0/download.bit
0 Infos, 0 Warnings, 0 Critical Warnings and 0 Errors encountered.
write_cfgmem completed successfully

Type a Tcl command here
```

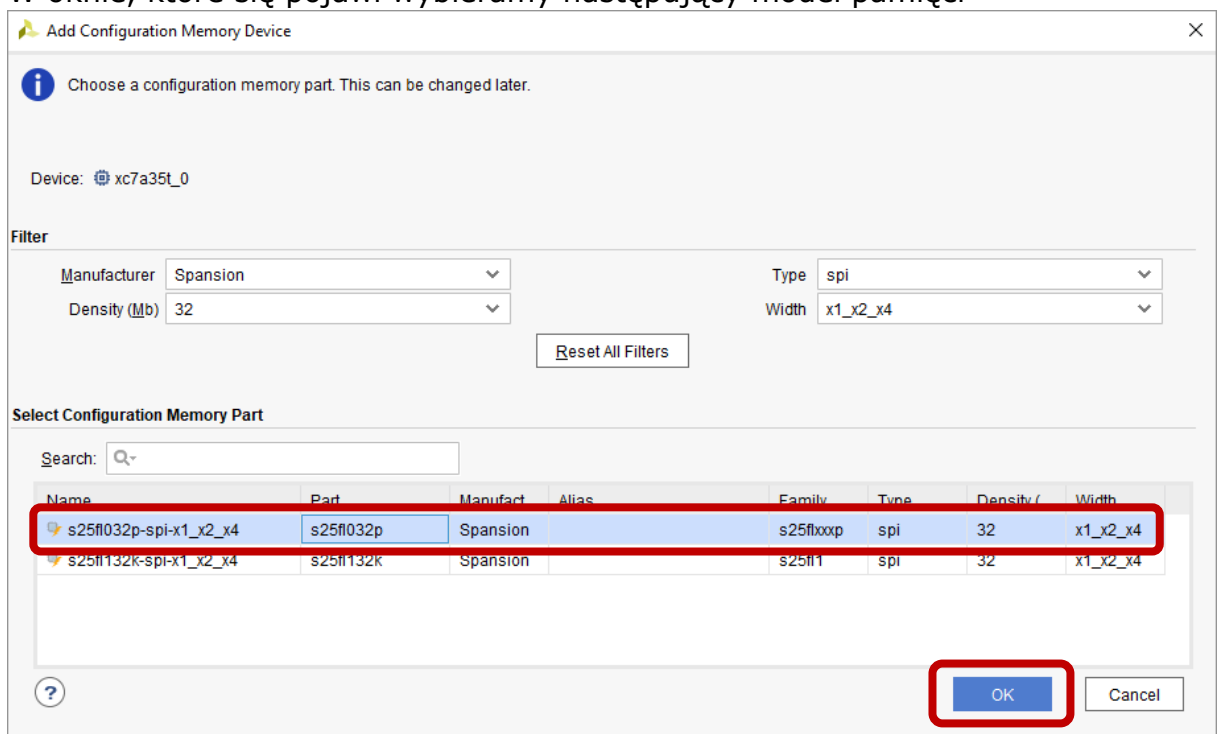
5. W vivado wybieramy opcję **Open Hardware Manager > Open Target > Auto Connect**



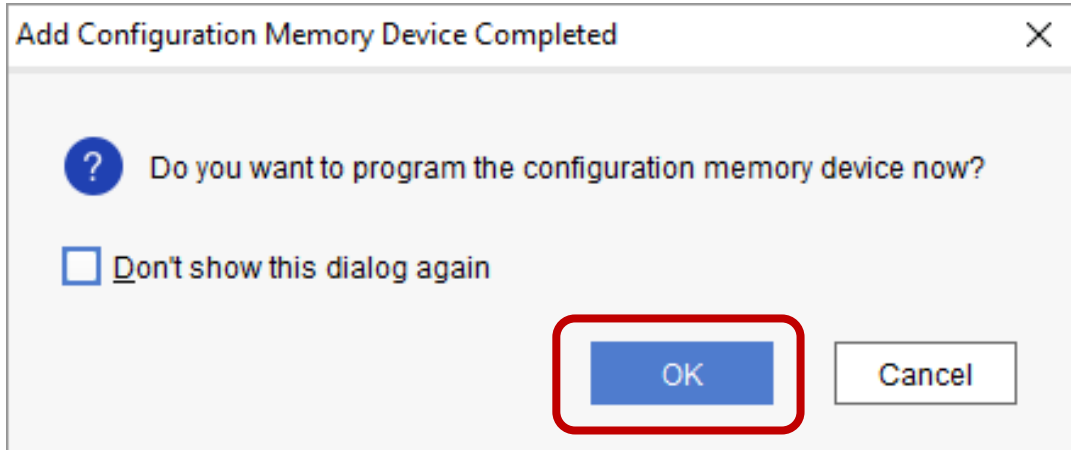
6. Po nawiązaniu połączenia z płytką Basys 3, wybieramy opcję **Add Configuration Memory Device**.



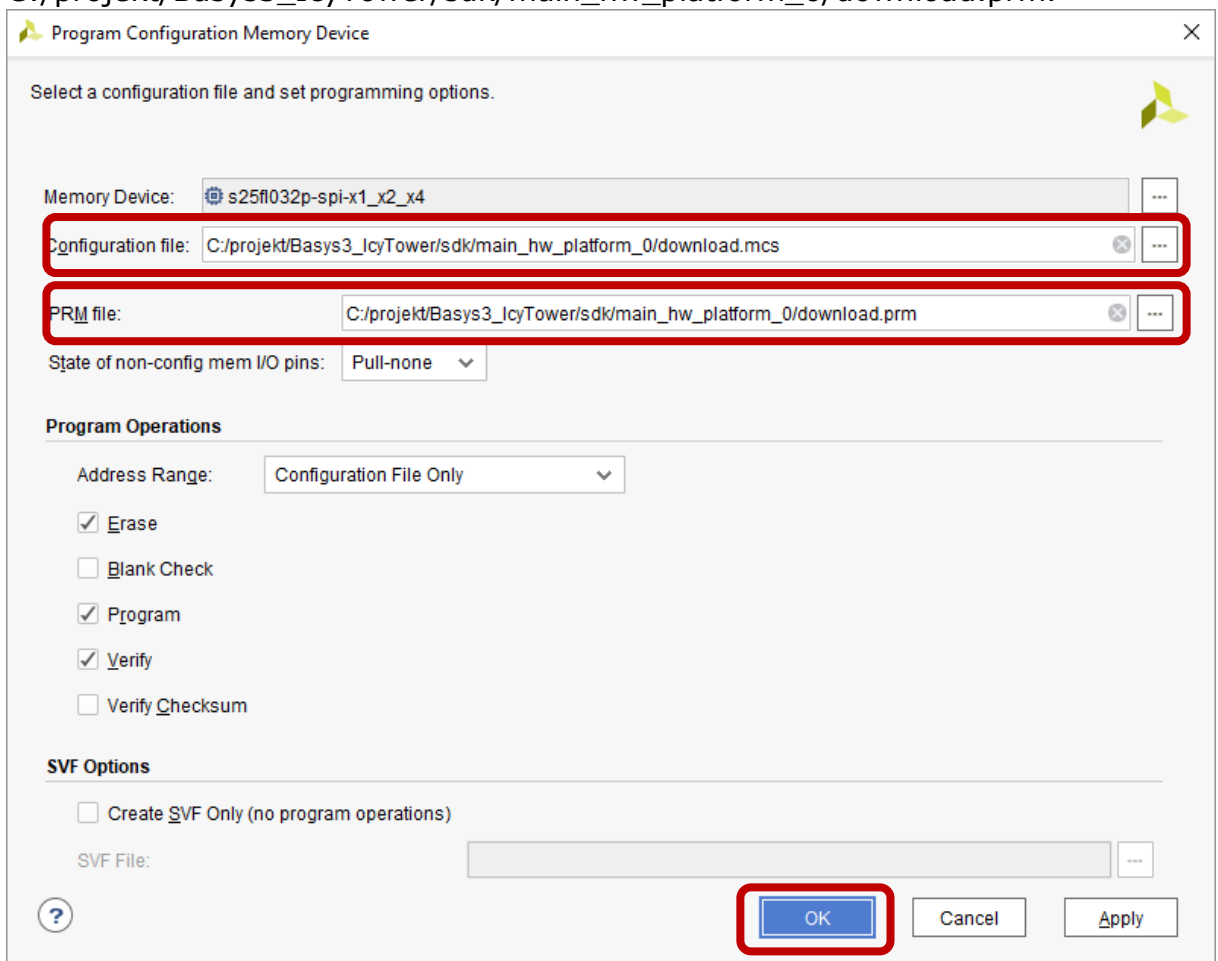
7. W oknie, które się pojawi wybieramy następujący model pamięci



8. Po naciśnięciu OK wybieramy opcję zaprogramowania pamięci



9. W oknie, które się pojawi wybieramy ścieżki dostępu do pliku download.mcs -  
\${Lokalizacja\_projektu}/sdk/main\_hw\_platform\_0/download.mcs, np.  
C:/projekt/Basys3\_IcyTower/sdk/main\_hw\_platform\_0/download.mcs,  
oraz pliku download.prm -  
\${Lokalizacja\_projektu}/sdk/main\_hw\_platform\_0/download.prm, np.  
C:/projekt/Basys3\_IcyTower/sdk/main\_hw\_platform\_0/download.prm.



10. Po zakończeniu programowania przełączamy zworkę **JP1** na płytce Basys 3 w pozycję **QSPI** i naciskamy przycisk **PROG** znajdujący się obok zworki.