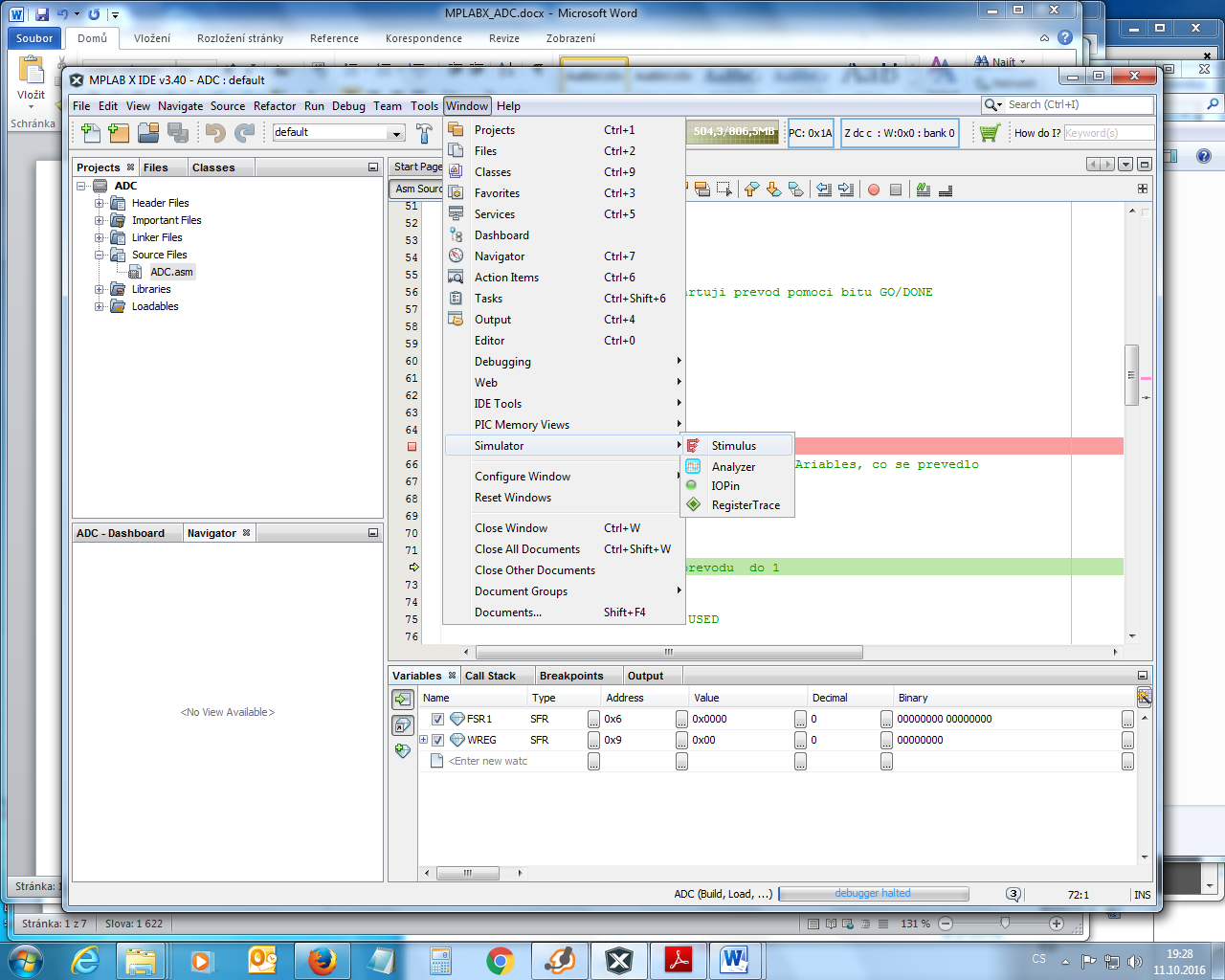
**Simulace převodníku pomocí MPLABX**

Uděláme si program, který převádí. Předpokládejme, že vstupní pin je na AN10 . Můžete použít soubor ADC.asm

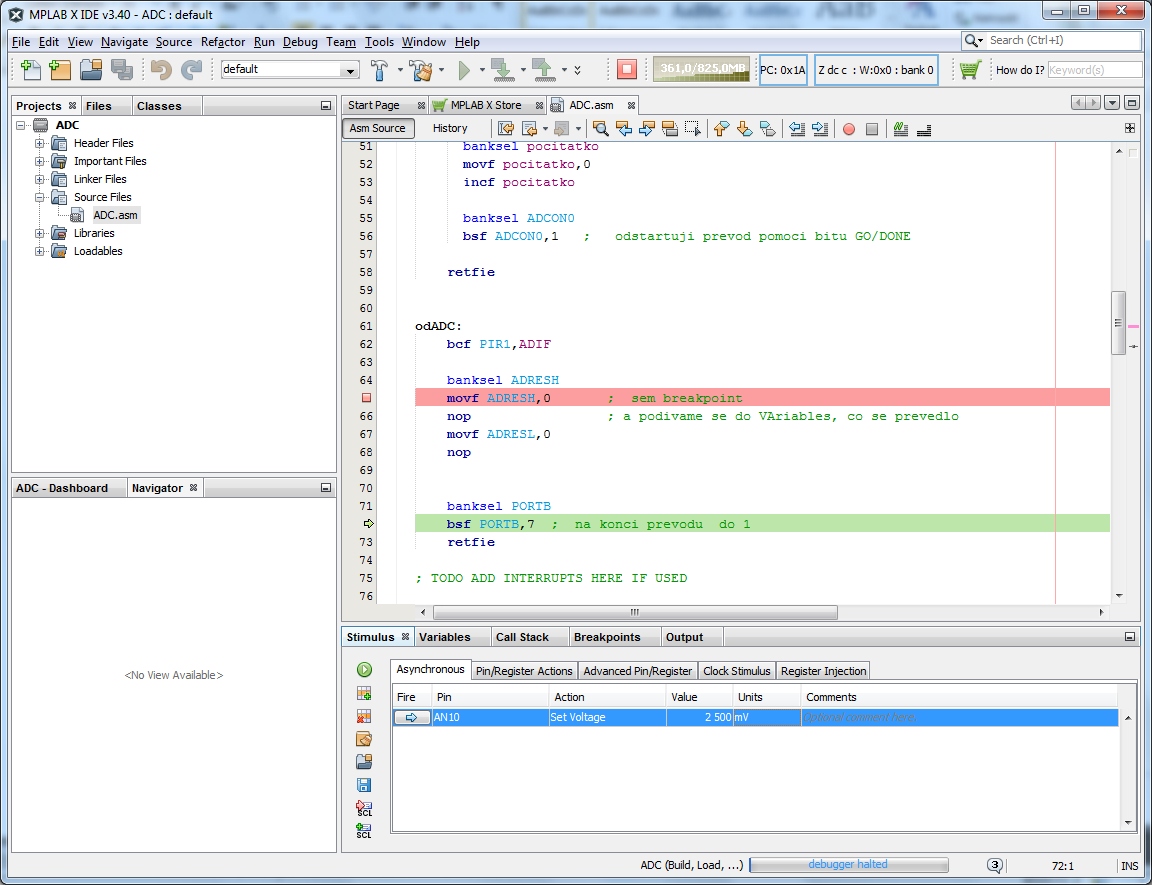
Nyní potřebujeme nasimulovat napětí, které zvenku přivádíme na pin AN10. Referenční napětí máme nastaveno na 0V a +5V.

Simulaci provedeme pomocí stimulus (Window – Simulator -Stimulus).

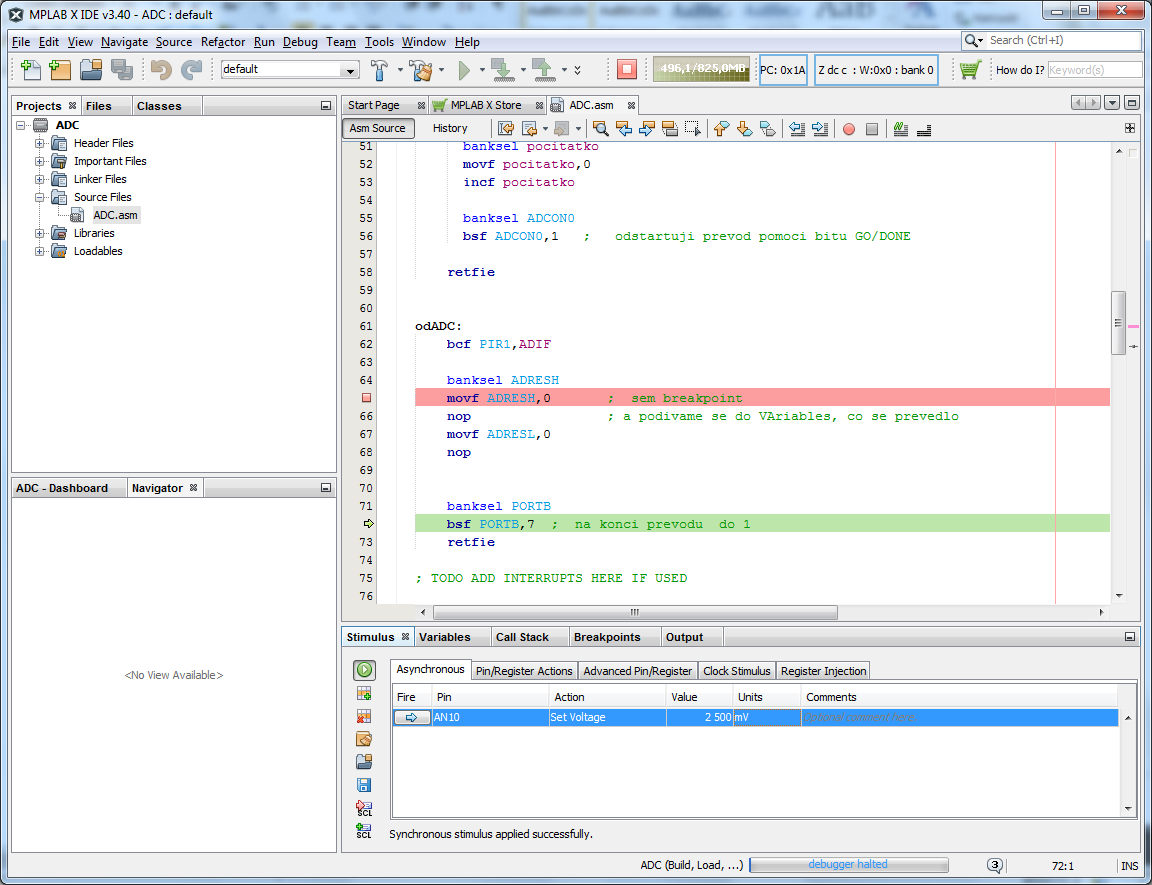


Dole se objeví známé okénko Stimulus.

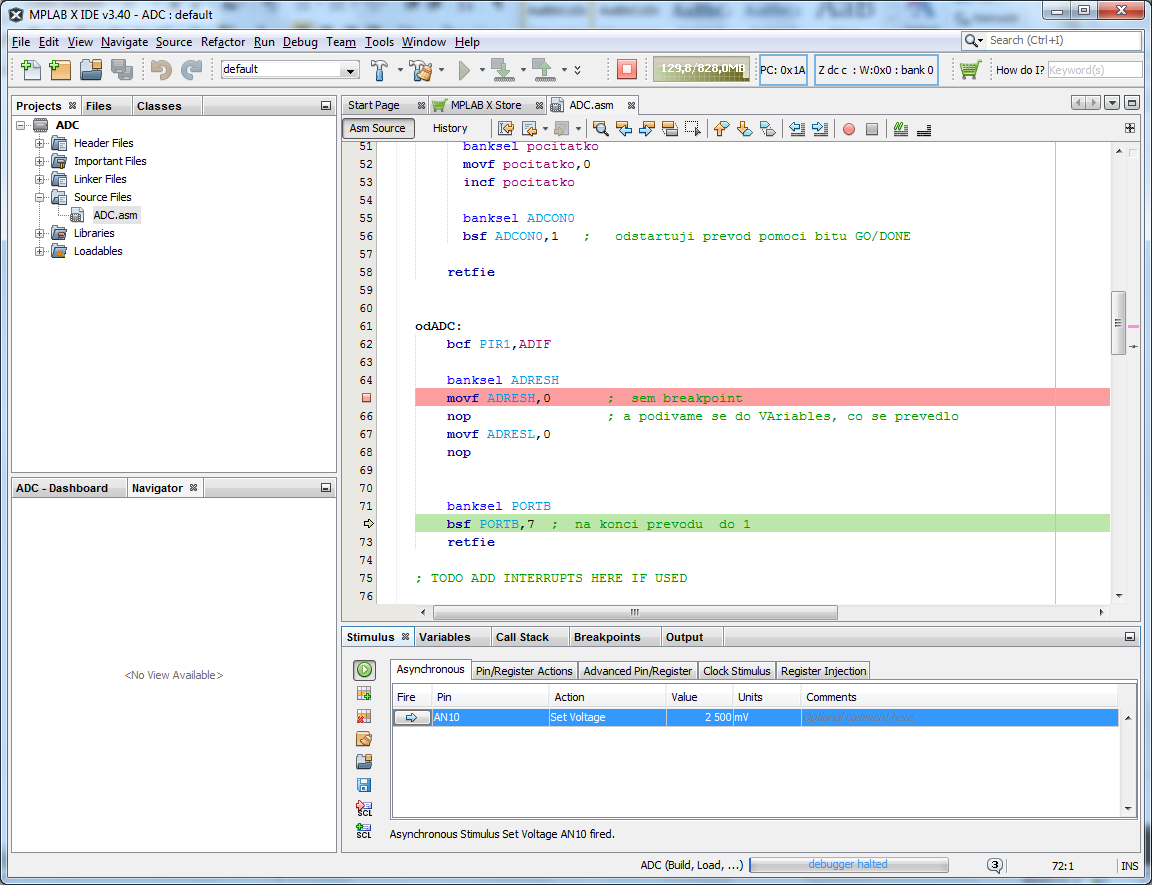
Volíme záložku Asynchronous, do sloupečku Pin dáme AN10 – pin, na kterém chceme mít vstupní napětí. Do Action dáme Set Voltage, do Value hodnotu, kterou potřebujeme, do Units jednotky. Na obrázku je AN10 , napětí 2.5V , tedy 2500mV . Netuším, zda je to schopno sežrat desetinnou tečku.



Cvakáme na zelenou ikonku vlevo nahoře – Synchronous stimulus applied successfully (??? to se možná nemusí, zkuste to)



Dále je postup naprosto stejný jako u nastavování vnějších pinů – v okamžiku, kdy cvakneme na „Fire“ , se zvenku na pin přivede nastavené napětí. No a můžeme simulovat chování převodníku.



Stejně jako u pinů si můžete zadat několik řádků s různými hodnotami, a pomocí „Fire“ přivádět na vstup různá napětí.

Při přivedení napětí 2.5V je výsledek ADRES == 0x200 – to je krásná polovina rozsahu, right justified.

Při přivedení napětí 5V dostaneme ADRES == 0x3ff – maximální číslo. Ovšem v okně „Output - Simulator“ se objeví hláška: „W0222-ADC: ADC input voltage high. ADC output overflow.“ Je to jasné – 5V už je mimo rozsah.

Maximální možné napětí na pinu AN10 je 4,995V. Tady už je simulátor spokojený

Krásně to funguje !

Simulátor dále simuluje referenční napětí Uref+ na pinu AN1

Simulátor NESIMULUJE napětí z FVR (vnitřní reference ) – nebo se mi to prostě nepodařilo správně nastavit

**Hodiny převodníku**

Simulátoru je úplně jedno, jaké hodiny jste nastavili. Neupozorní vás, když hodnota TAD vyleze z povolených mezí 1 – 9 mikrosekund. Ale hodiny simuluje, když změníte nastavení děliček pro hodiny, změní se doba převodu.

**Simulace AUTO-CONVERSION TRIGGER**

Jediná možnost, kterou simulátor podporuje, je odstartování převodu bitem GO/DONE. Pokud se pokusíte nastavit jiný zdroj pro začátek převodu (například Timer4 – T4\_match), tak převodník nepřevádí a v okně Output – Simulator se objeví lakonická hláška: „W0230-ADC: TRIGSEL values other than default not supported.“

**Doporučení:**

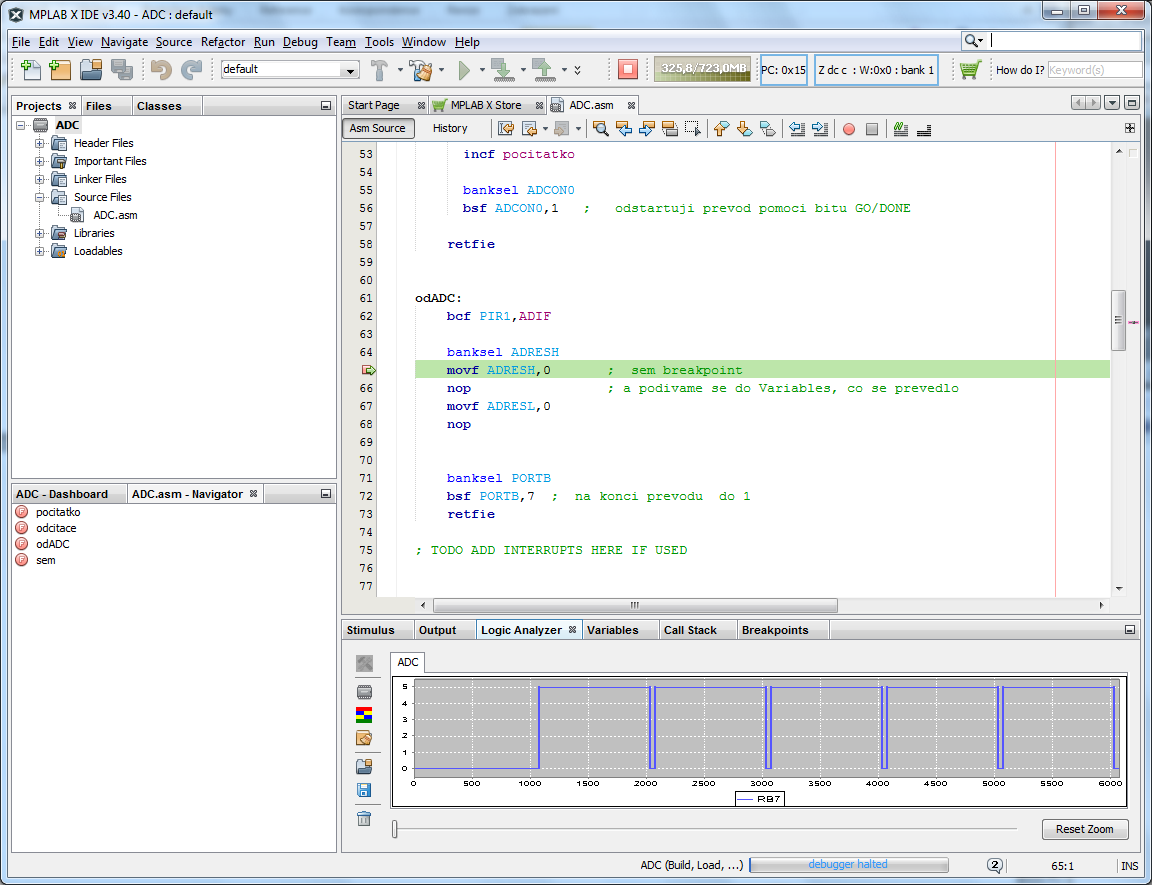
Doporučuji tedy v programu nastavit referenční napětí ze zdroje, tedy 0V a +5V . Převod odstartujeme pomocí nastavení bitu GO/DONE. podíváme se, jaká čísla nám převodník dodává, jak dlouho převod trvá atd. atd. Až vše odsimulujeme, upravíme program tak, aby pracoval podle našeho přání ( referenční napětí, Auto-conversion-Trigger atd. ) a budeme doufat, že to v doopravdickém PICovi bude fungovat správně.

Bit RB7

V programu ADC.asm máte manipulaci s bitem RB7. To je zárodek pro použití CS u obvodu MCP4822. V našem programu ( ADC.asm) slouží RB7 pro zobrazení začátku a konce převodu. Pustíme si na něj Logic Analyzer.

Perioda odstartování převodu – 1ms – viz TMR4

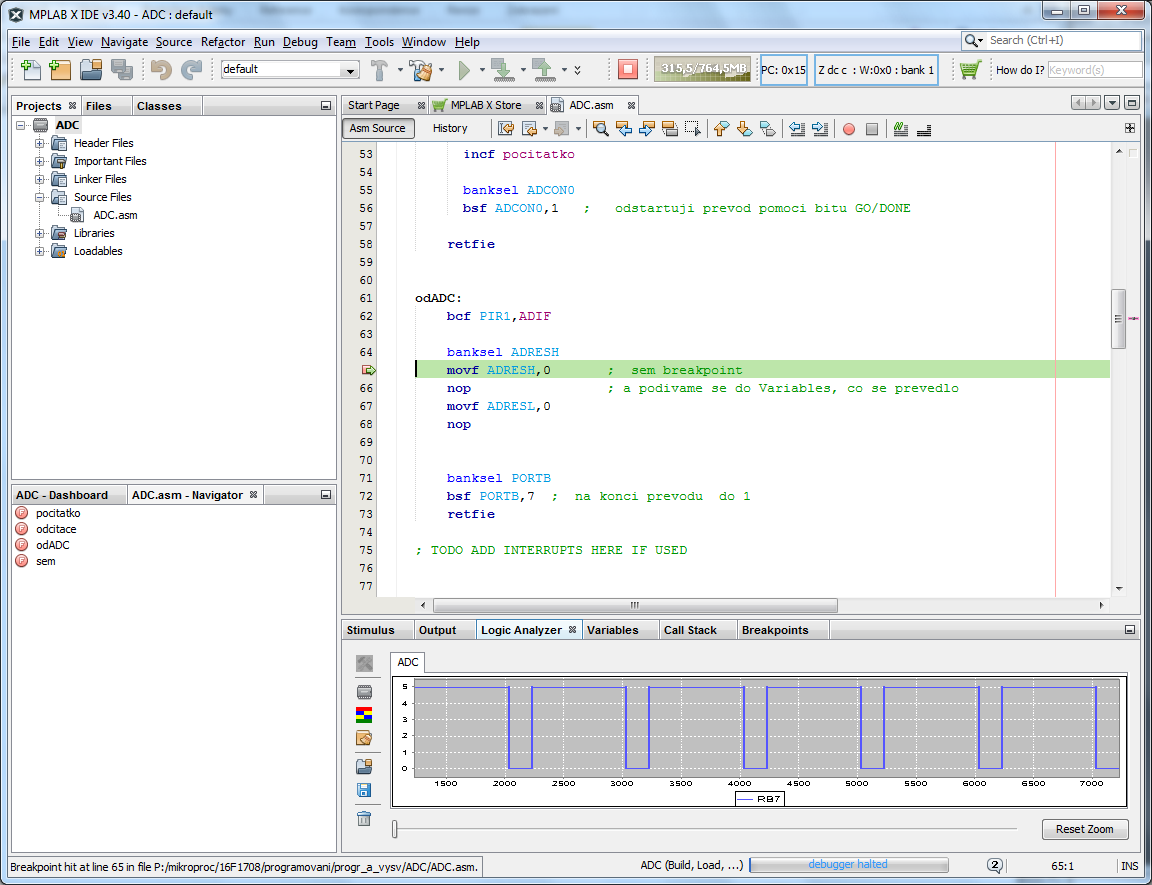
Hodiny převodníku – Fosc/8 , tedy TAD = 2 mikrosec, doba převodu 22 mikrosec



No, to by tak mohlo být

Perioda odstartování převodu – 1ms – viz TMR4

Hodiny převodníku – Fosc/64 , tedy TAD = ??? mikrosec, doba převodu ??? mikrosec



Prostě si s tím hrajte, ať vidíte, co ten převodník dělá