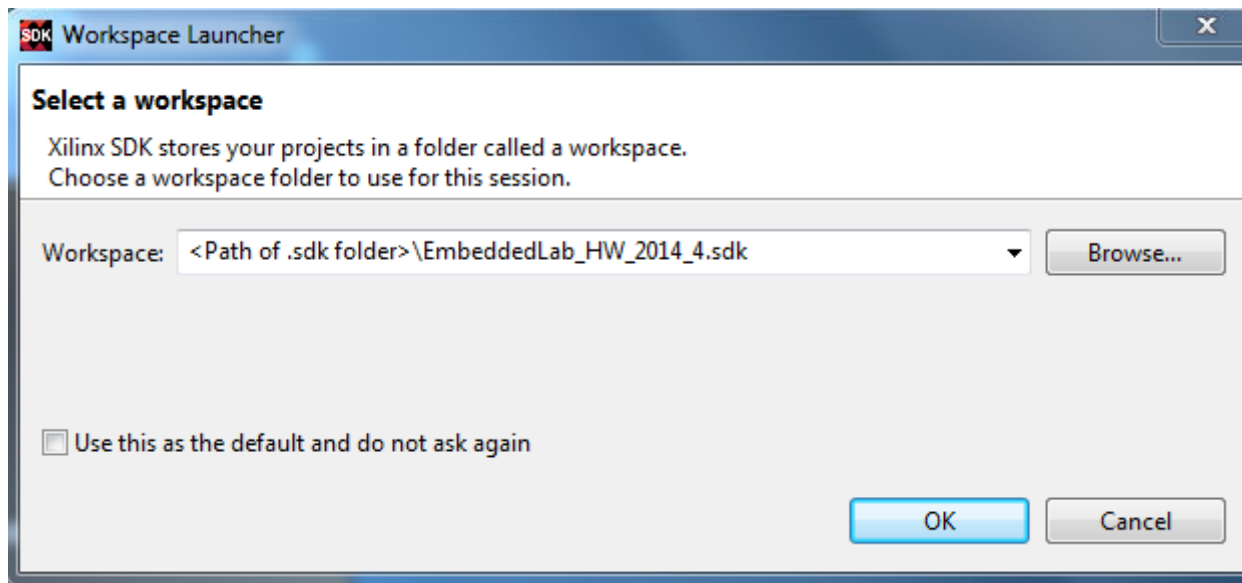


Xilinx SDK (Software Development Kit)

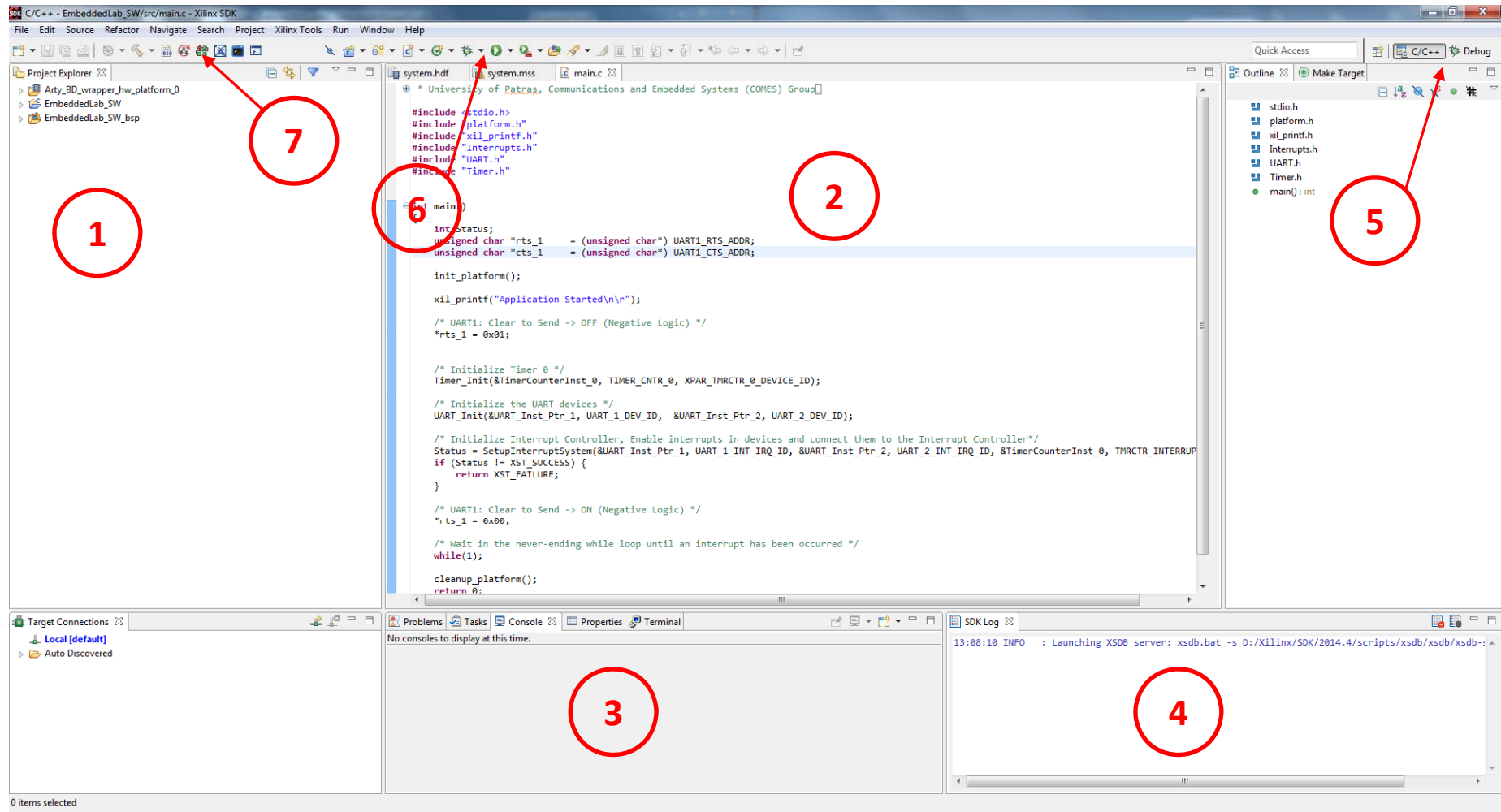
- Το Xilinx SDK (Software Development Kit) είναι ένα περιβάλλον ανάπτυξης εφαρμογών λογισμικού για ενσωματωμένους επεξεργαστές της Xilinx
- Τα projects του SDK βασίζονται σε designs hardware που έχουν δημιουργηθεί από το λογισμικό ανάπτυξης υλικού Vivado.
- Το SDK περιλαμβάνει τα εξής χαρακτηριστικά:
 - C/C++ editor και περιβάλλον για compilation κώδικα
 - Διαχείριση project λογισμικού
 - Περιβάλλον για debugging και profiling
 - Διαχείριση λαθών
 - Διαχείριση μνήμης flash και προγραμματισμό FPGA μέσω JTAG
 - Βιβλιοθήκες λογισμικού και drivers για τα περιφερειακά του επεξεργαστή

Επιλογή Workspace

- Ανάλογα με τις ρυθμίσεις του υπολογιστή κάποια εκτελέσιμα αρχεία που χρησιμοποιεί το SDK ενδεχομένως να χρειάζονται δικαιώματα Administrator.
- Κατά την εκκίνηση του “Xilinx SDK 2014.4” επιλέγουμε το Workspace στο οποίο υπάρχουν όλα τα απαραίτητα αρχεία για την αναγνώριση του hardware, των software project καθώς και τα αρχεία για αποθήκευση meta-information
- Τα αρχεία αυτά βρίσκονται στον φάκελο <HardwareDesignName>.sdk






Παράθυρα του SDK



Παράθυρα του SDK

1. Ο Project Explorer περιλαμβάνει τα εξής:

-  Στο hardware platform βρίσκονται όλα τα απαραίτητα αρχεία που περιγράφουν το hardware (soft processors, περιφερειακά, διευθύνσεις περιφερειακών, bitstream κλπ)
-  Το C application project περιέχει όλους τους κώδικες που αναπτύσσει ο χρήστης για την εφαρμογή του, τις βιβλιοθήκες που κάνει include, το αρχείο του linker και το makefile. Στον ίδιο φάκελο παράγεται αυτόματα το εκτελέσιμο αρχείο (.elf) κάθε φορά που αποθηκεύεται ο κώδικας, αρκεί να μην περιέχονται errors σε αυτόν
-  Το BSP (Board support package) περιλαμβάνει μία συλλογή από βιβλιοθήκες και drivers η οποία αποτελεί το κατώτερο στρώμα για να μπορέσει να εκτελεστεί μία εφαρμογή software. Το λογισμικό που γίνεται compile στο C application project θα πρέπει να γίνει link με ένα ήδη υπάρχον BSP. Σε απλές εφαρμογές το BSP παράγεται αυτόματα από το ίδιο το SDK όταν παράγεται ένα νέο Application project και ο χρήστης δε χρειάζεται να αλλάξει κάτι σε αυτό.

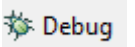
2. Στον C/C++ editor φαίνονται οι κώδικες που παράγει ο χρήστης καθώς και οποιαδήποτε άλλα αναγνώσιμα αρχεία ανοιχτούν μέσα από το περιβάλλον του SDK. Στα αρχεία .c και .h, τα λάθη υπογραμμίζονται αυτόματα από τον compiler με κόκκινη γραμμή, ενώ οι προειδοποιήσεις με πορτοκαλί γραμμή

Παράθυρα του SDK


3. Στο κάτω μέρος του editor εμφανίζεται το Console Window στο οποίο μπορούν να κατευθυνθούν τα μηνύματα του STDIO της εφαρμογής, μέσω των ρυθμίσεων εκτέλεσης του project. Υπάρχουν περιπτώσεις όπου μπορεί να εμφανιστούν προβλήματα με τη χρήση του, γι' αυτό μπορεί να απαιτείται η εγκατάσταση εξωτερικού προγράμματος (πχ. Tera Term).
4. Στο SDK Log, εμφανίζονται όλα τα μηνύματα του SDK καθώς και οι προηγούμενες ενέργειες που εκτελέστηκαν από τον χρήστη κατά τη διάρκεια ενός session
5. Από τα δύο αυτά κουμπιά ο χρήστης επιλέγει δύο διαφορετικές όψεις:



Είναι ένα View το οποίο παρέχει όλα τα απαραίτητα εργαλεία και παράθυρα για ευκολότερη ανάπτυξη του λογισμικού

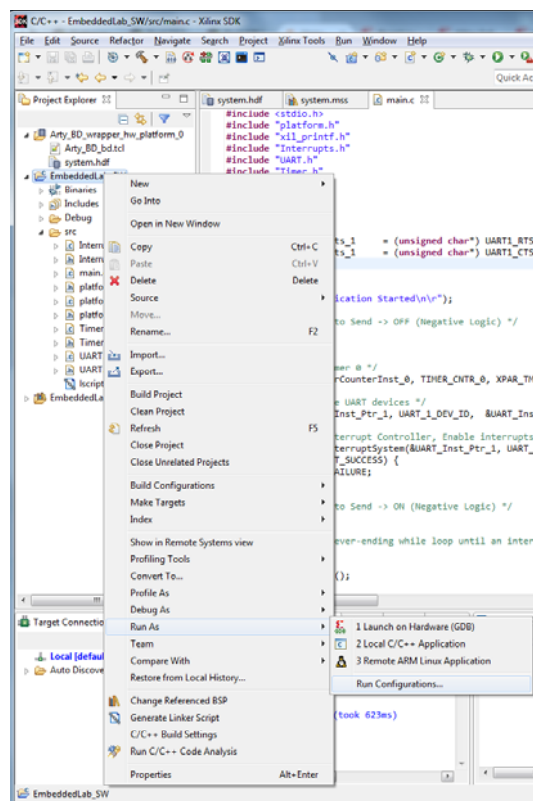


Το Debug View χρησιμοποιείται όταν ο χρήστης κάνει debugging μίας εφαρμογής. Στο View αυτό γίνεται ευκολότερη διαχείριση των breakpoints και φαίνονται οι τιμές των μεταβλητών και των καταχωρητών κάθε φορά που σταματάει η εκτέλεση ενός προγράμματος σε ένα από τα breakpoints

6. Με τα κουμπιά αυτά εκτελείται μία εφαρμογή που έχει γίνει ήδη compile ή κάνουμε debug, στο οποίο η εκτέλεση ενός προγράμματος θα σταματήσει στα breakpoints που έχει θέσει ο χρήστης
7. Με το κουμπί  γίνεται ο προγραμματισμός του FPGA με ένα αρχείο bitstream, το οποίο θα παραγάγει στο FPGA το κύκλωμα που έχει προσδιοριστεί μέσω του Vivado

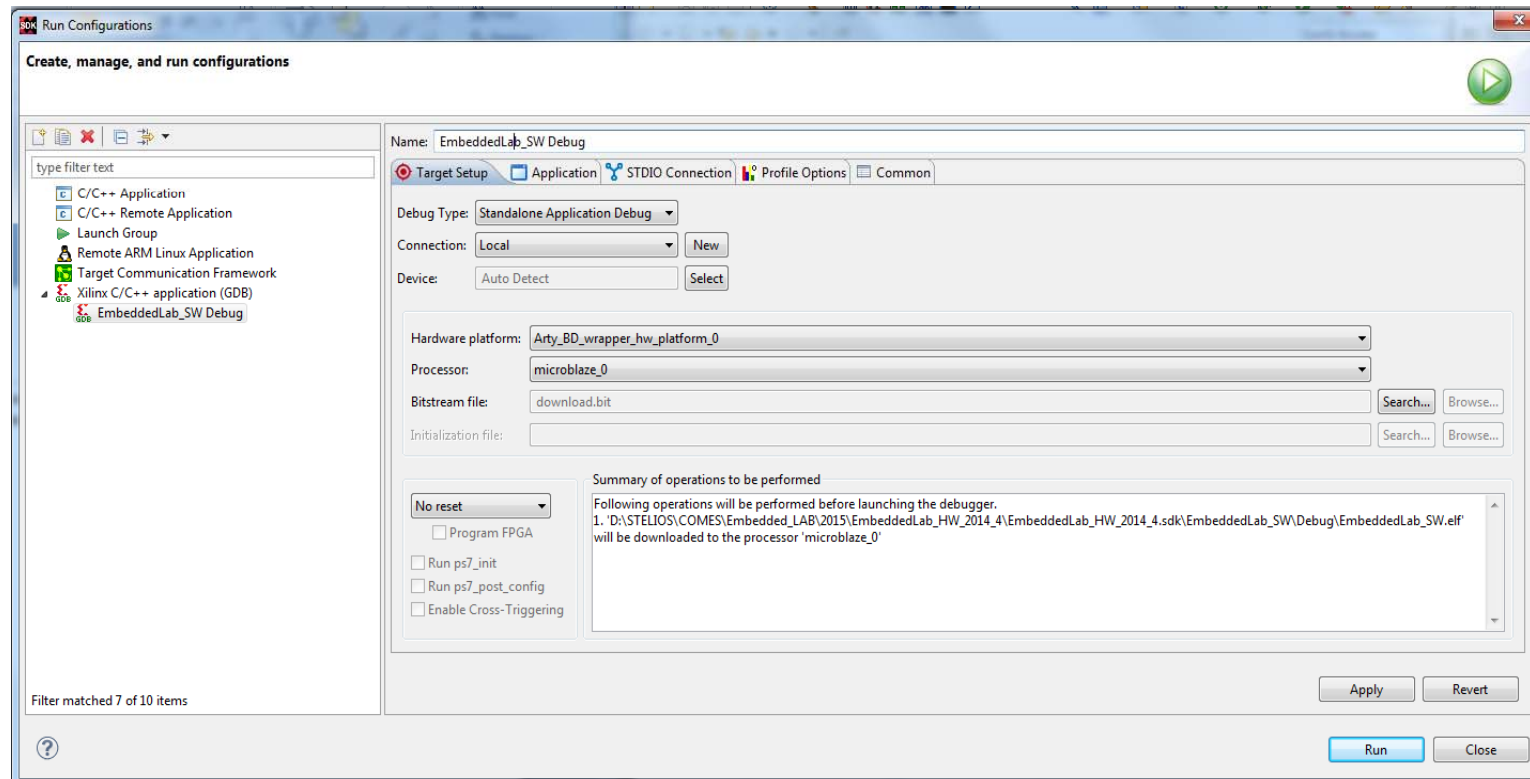
Προετοιμασία για εκτέλεση προγράμματος

- Μετά την παραγωγή ενός Application project είναι απαραίτητο να γίνουν κάποιες ρυθμίσεις σε αυτό προτού εκτελεστεί για πρώτη φορά μία εφαρμογή
- Για το λόγο αυτό κάνουμε δεξί κλικ πάνω στο Application project στον Project Explorer και επιλέγουμε “Run As” > “Run Configurations...”



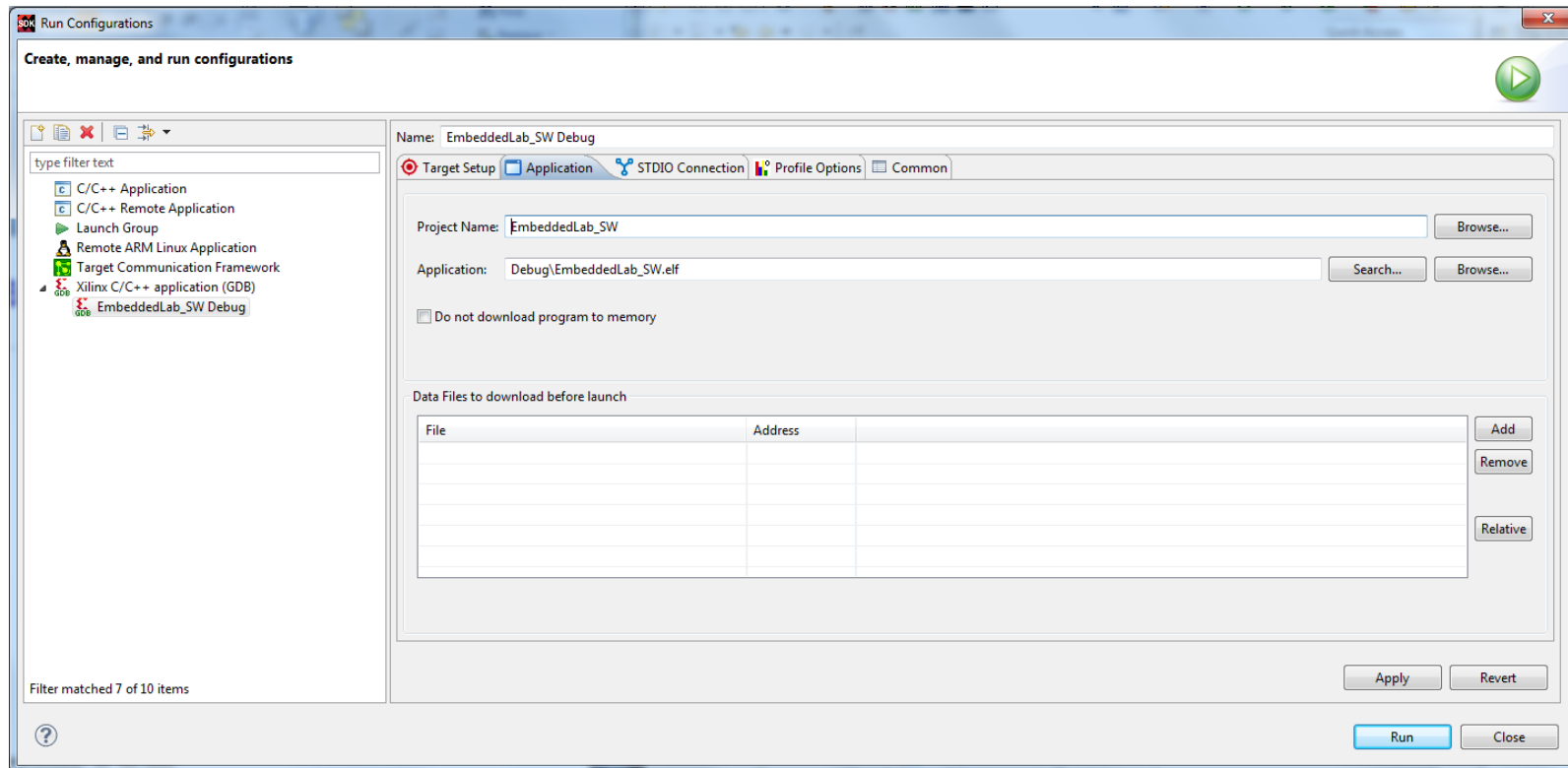
Προετοιμασία για εκτέλεση προγράμματος

- Με διπλό κλικ πάνω στο “Xilinx C/C++ application (GDB)” παράγεται ένα Run/Debug configuration στο οποίο επιλέγουμε τις παρακάτω ρυθμίσεις



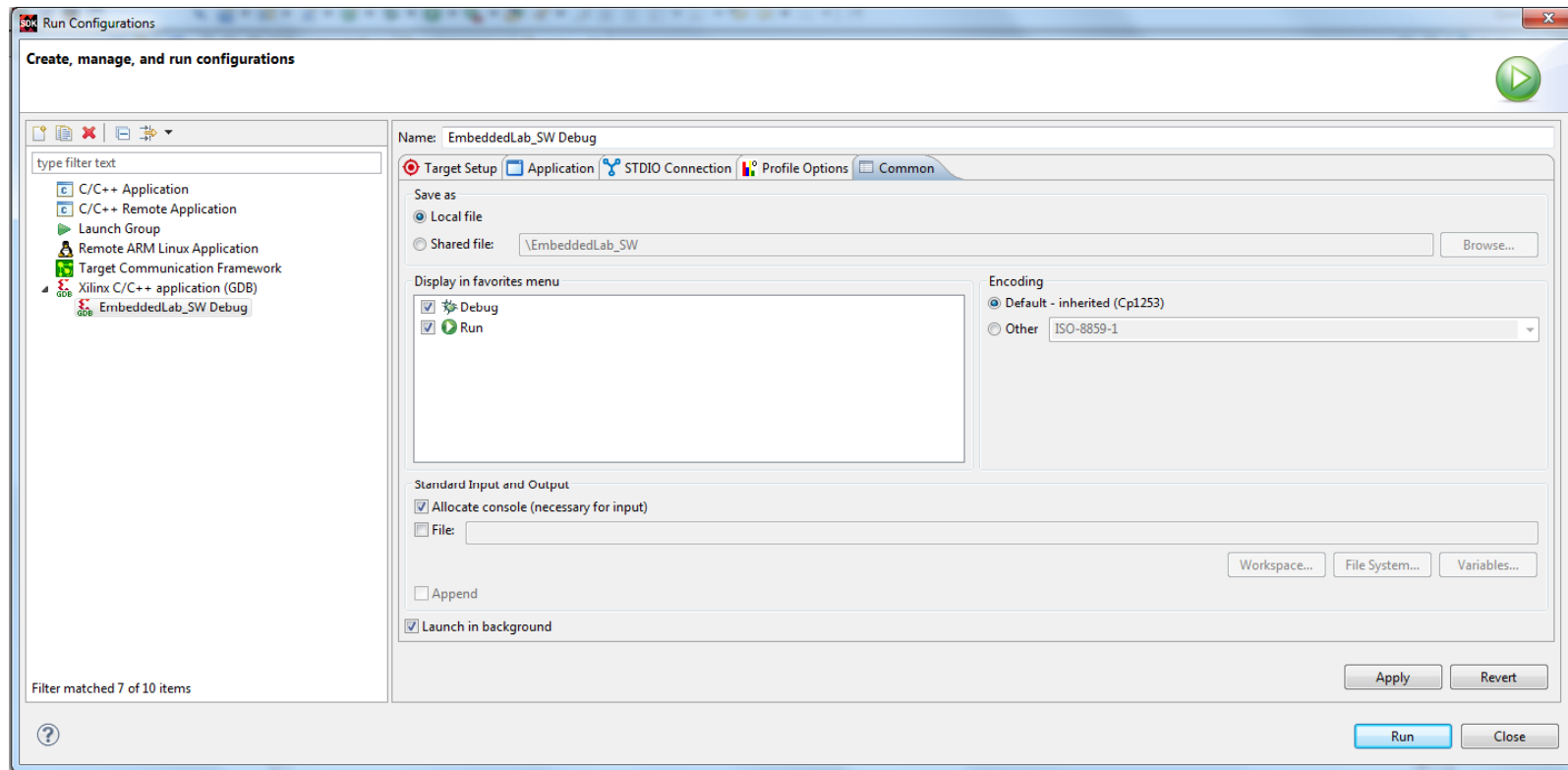
Target Setup Tab: Αλλάζουμε το “Reset processor” σε “No reset”

Προετοιμασία για εκτέλεση προγράμματος




Application Tab: Συμπληρώνουμε τα πεδία “Project Name” και “Application” με τα αντίστοιχα ονόματα του Project name και του εκτελέσιμου αρχείου

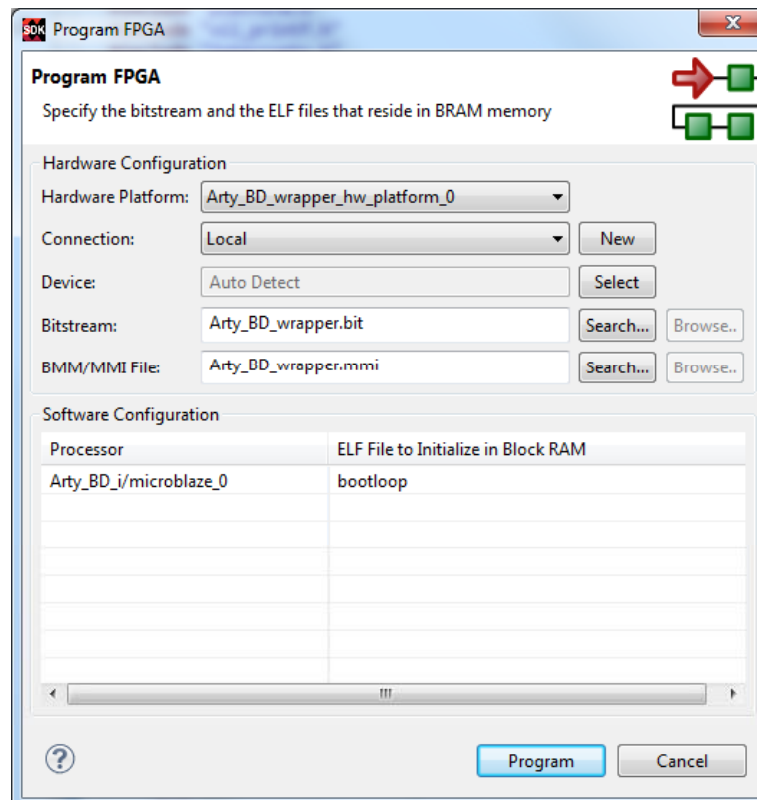
Προετοιμασία για εκτέλεση προγράμματος



Common Tab: Επιλέγουμε τα check boxes Debug και Run. Στη συνέχεια πατάμε “Apply” και “Close”.

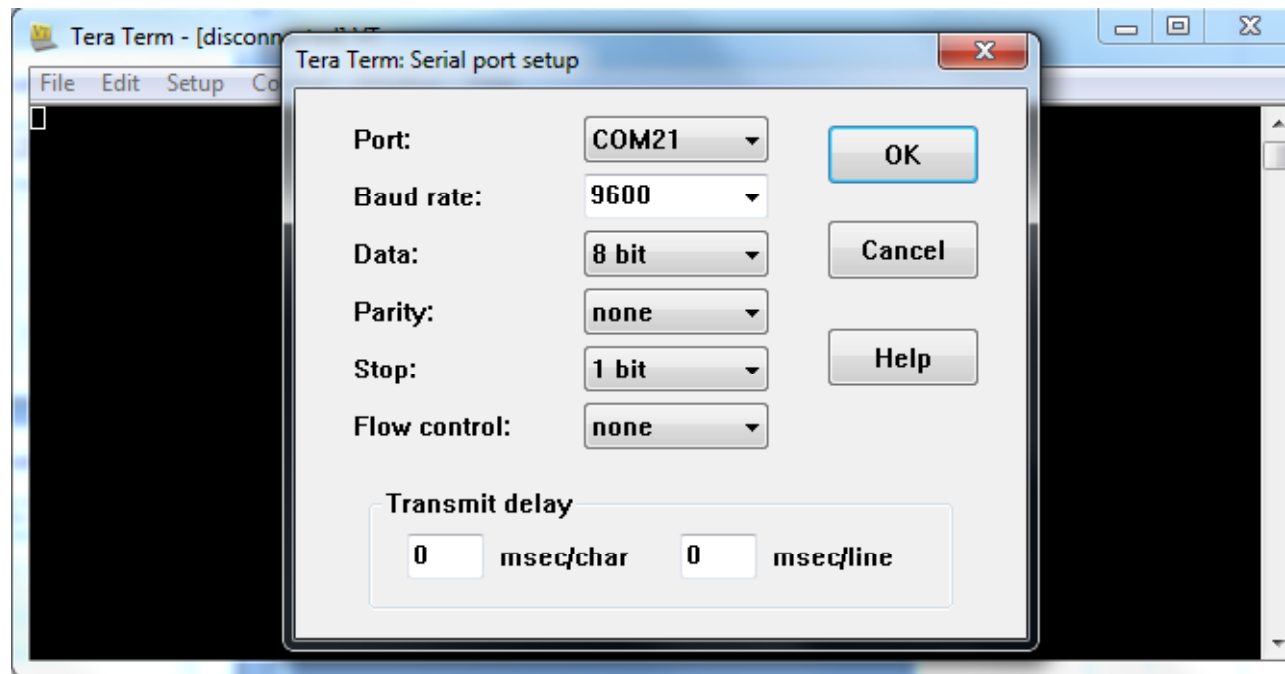
Προγραμματισμός FPGA

- Για να προγραμματιστεί ένας επεξεργαστής που έχει υλοποιηθεί στο FPGA (soft-processor) μαζί με τα περιφερειακά του (πχ. UART, GPIO, Timers, Interrupt Controllers κλπ) θα πρέπει να προγραμματίσουμε το FPGA με το αντίστοιχο bitstream αρχείο.
- Η διαδικασία αυτή γίνεται πατώντας το κουμπί  και στη συνέχεια πατώντας “Program”




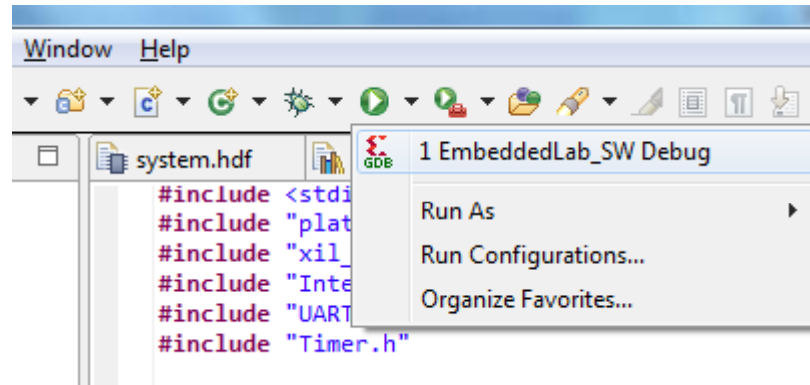
Console

- Το ενσωματωμένο console του SDK ενδέχεται να δημιουργήσει πρόβλημα στον προγραμματισμό του FPGA
- Συνίσταται η χρήση εξωτερικού terminal προγράμματος το οποίο θα δέχεται τα δεδομένα από τη UART και θα τα τυπώνει. Το Tera Term είναι ένα δωρεάν (opensource) terminal για Windows
- Για τη ρύθμισή του πάμε στο Setup > Serial Port..., επιλέγουμε το COM port στο οποίο είναι συνδεδεμένο το STDIO του ARTY board και επιλέγουμε τις παρακάτω ρυθμίσεις



Εκτέλεση/Debug προγράμματος

- Η εκτέλεση του προγράμματος γίνεται από το πλήκτρο  στο οποίο επιλέγουμε το configuration που ολοκληρώθηκε σε προηγούμενο βήμα όπως φαίνεται παρακάτω



- Εκτός από εκτέλεση ενός προγράμματος μπορούμε να κάνουμε και debugging, κατά το οποίο η εκτέλεση σταματάει σε προκαθορισμένα σημεία του κώδικα (breakpoints) στα οποία μπορούμε να δούμε τις τιμές μεταβλητών ή θέσεων μνήμης. Το debug γίνεται με το πλήκτρο 