



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών

— ΙΔΡΥΘΕΝ ΤΟ 1837 —

Μικροεπεξεργαστές - DSPs

Εργαστηριακή Άσκηση 1: Blink LED

Ενότητα 1: Εισαγωγή

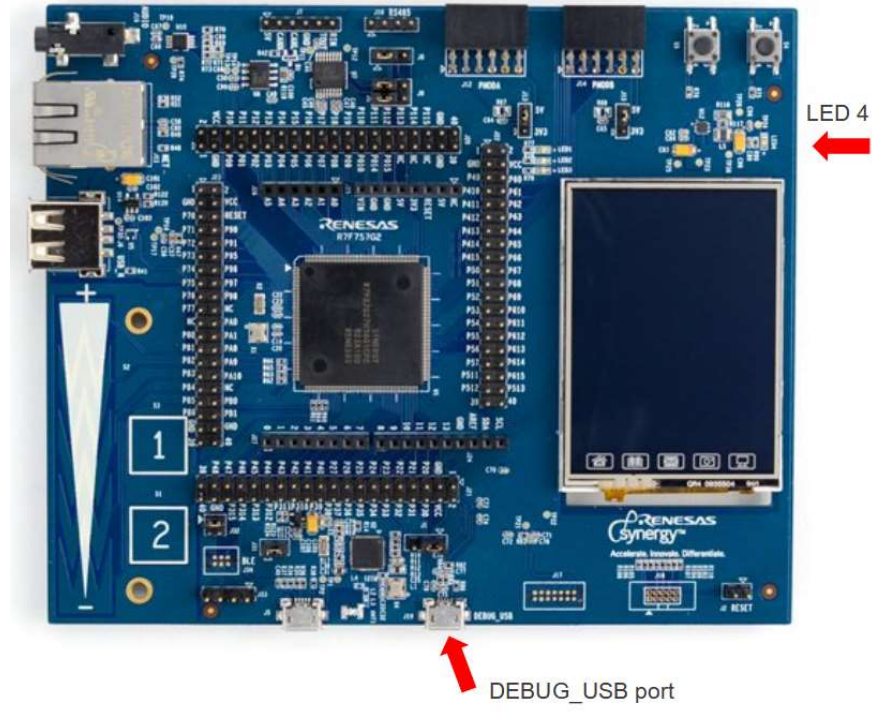
1-1 Σκοπός της άσκησης

Σκοπός της άσκησης αυτής είναι η ανάπτυξη προγράμματος σε μικροεπεξεργαστή. Το πρόγραμμα θα ανάβει και θα σβήνει μια λυχνία LED, με κάποια περιοδικότητα. Πιο συγκεκριμένα θα δημιουργήσετε πρόγραμμα, το οποίο θα ανάβει μια λυχνία LED και θα τη κρατάει σε αυτή τη κατάσταση για ένα δευτερόλεπτο. Στη συνέχεια θα τη σβήνει, επίσης για ένα δευτερόλεπτο. Η διαδικασία αυτή θα πρέπει να επαναλαμβάνεται για όσο βρίσκεται σε λειτουργία ο μικροεπεξεργαστής.

1-2 Board setup

Στο εργαστήριο θα χρησιμοποιήσετε το development board **SK-S7G2**, που περιέχει τον μικροελεγκτή **S7G2**. Οι μικροελεγκτές είναι μικροεπεξεργαστές, οι οποίοι διαθέτουν ενσωματωμένα περιφερειακά εξαρτήματα. Μερικά τέτοια περιφερειακά εξαρτήματα, που συναντώνται στους μικροελεγκτές, είναι οι μετατροπείς αναλογικού σήματος σε ψηφιακό, χρονιστές-απαριθμητές, μονάδες επικοινωνίας, και πολλά άλλα.

Για να συνδέσετε το board με τον υπολογιστή, συνδέστε το καλώδιο **micro-USB** στη θύρα **J19-Debug USB** της πλακέτας. Η λυχνία LED 4 υποδηλώνει ότι το board βρίσκεται σε λειτουργία.



Ενότητα 2: Δημιουργία project

Για να δημιουργήσετε ένα καινούργιο project ακολουθείστε τα παρακάτω βήματα:

2-1 Επιλέξτε το εικονίδιο του **e² studio**.

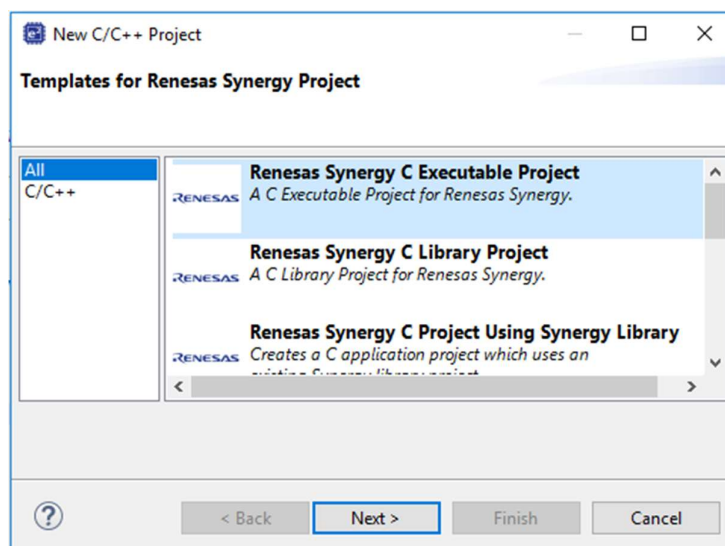


2-2 Ορίστε το workspace, ως εξής :

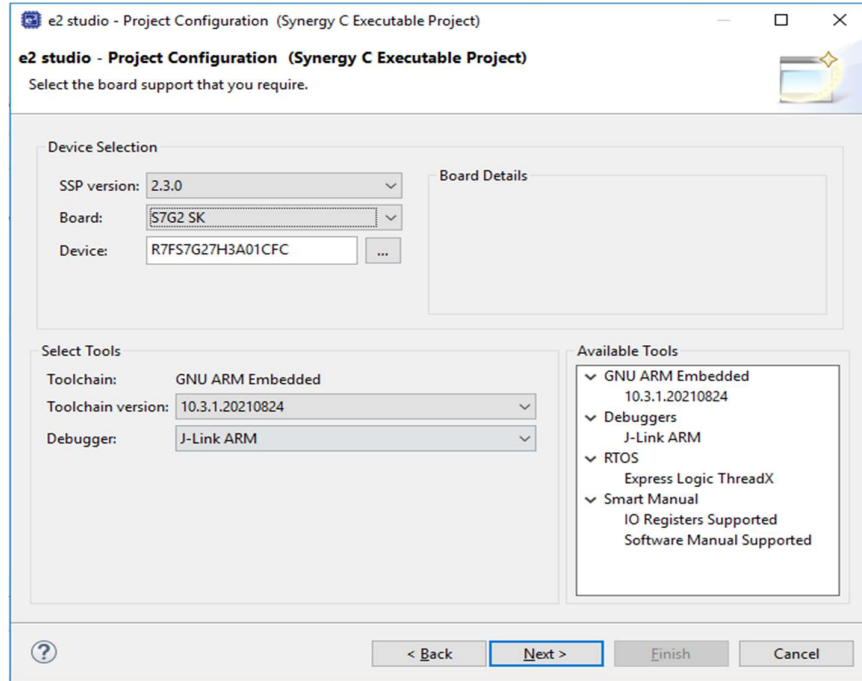
C:\Users\Student\Desktop\micro lab\lab

2-3 Πατήστε **“Create a new C/C++ project”** για να δημιουργήσετε ένα νέο project.

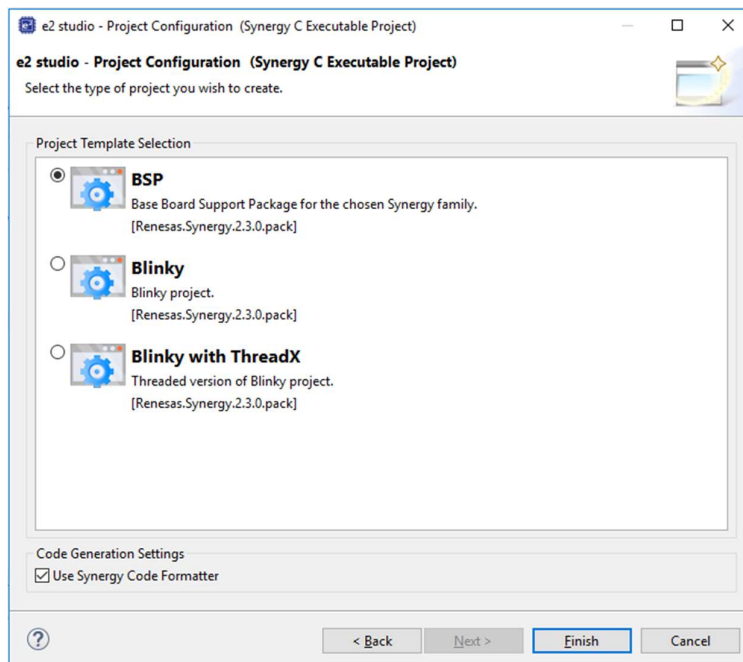
2-4 Στο παράθυρο που εμφανίζεται επιλέξτε **“Renesas Synergy C Executable Project”**



2-5 Στο παράθυρο “**Project configuration**” επιλέξτε όνομα για το project. Στη συνέχεια επιλέξτε “**S7G2 SK**” στο πεδίο Board. Τα υπόλοιπα πεδία επιλέγονται αυτόματα.

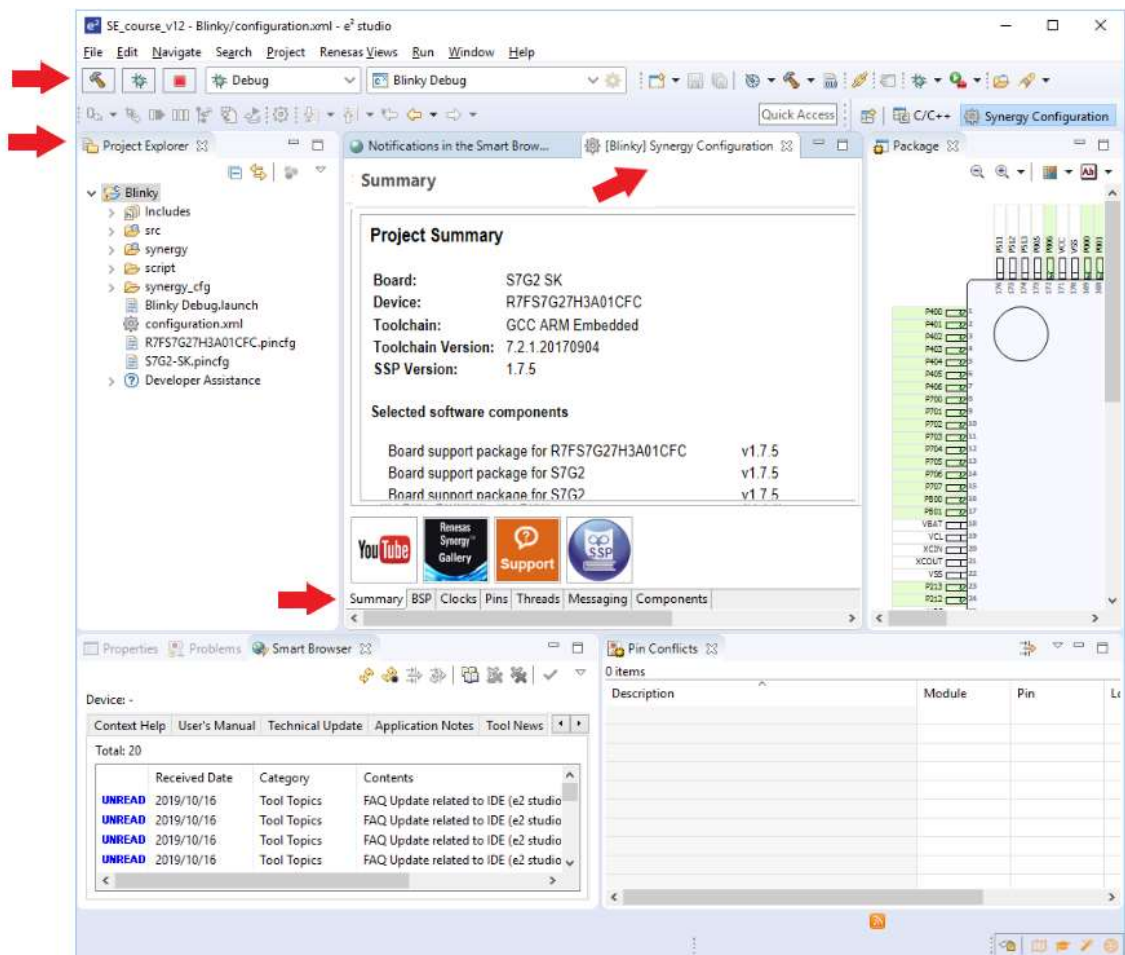


2-6 Στο παράθυρο Project Template Selection επιλέξτε το **BSP**.



2-7 Στην επόμενη σελίδα φαίνεται η επιφάνεια εργασίας του e² studio. Στην επιφάνεια εργασίας του e² studio, διακρίνουμε τις ακόλουθες μπάρες εργαλείων και παράθυρα :

Το παράθυρο **Project Explorer**, που περιέχει όλα τα project του workspace. Το παράθυρο **Synergy Configuration** περιέχει τα εργαλεία για τις ρυθμίσεις του project. Μέσα από αυτό επιλέγονται η συχνότητα λειτουργίας του μικροελεγκτή, η λειτουργία των pins, οι drivers για τις περιφερειακές μονάδες, το λειτουργικό **ThreadX**, το **toolbar** για μεταγλώττιση και εκτέλεση κλπ.



Ενότητα 3 : Εργαστηριακό κομμάτι

Για να ξεκινήσετε ανοίξτε το αρχείο **hal_entry.c** που βρίσκεται στο φάκελο **src** του project και ακολουθήστε τα παρακάτω βήματα για την εκτέλεση της άσκησης.

Βήμα 1 - LEDs

Για τις λυχνίες LED θα χρησιμοποιηήσετε αυτές που διαθέτει το board. Το board διαθέτει συνολικά τρεις λυχνίες LED, τις οποίες μπορείτε να χρησιμοποιήσετε, γράφοντας τις ακόλουθες εντολές στο πρόγραμμα σας:

- **bsp_leds_t leds;**
- **R_BSP_LedsGet(&leds);**

Η δομή **leds** περιέχει πληροφορίες για τα LED στη πλακέτα και η πρόσβαση σε αυτά γίνεται μέσω του πίνακα **leds.p_leds[i]**, όπου **i** από 0 έως 2.

Βήμα 2 - Super loop

Τα προγράμματα σε μικροελεγκτές συνήθως τρέχουν μέσα σε ένα super loop, δηλαδή μια while, η οποία δεν σταματάει ποτέ (π.χ while(1)). Τα κομμάτια του κώδικα, που δεν αποτελούν ορισμούς ή αναθέσεις και πρέπει να επαναλαμβάνονται διαρκώς, πρέπει να βρίσκονται μέσα στο super loop. Δημιουργείστε ένα super loop το οποίο θα περιέχει την “εγγραφή” στις λυχνίες LED και την περιοδικότητα.

Βήμα 3 - Pin Write

Για να “γράψετε” σε μια λυχνία, δηλαδή για να αλλάξετε την τάση σε αυτή (0 ή 1), χρησιμοποιείτε την ακόλουθη εντολή:

```
g_ioport.p_api→pinWrite(leds.p_leds[i], level);
```

Η συνάρτηση **pinwrite()** δέχεται ως ορίσματα τον ακροδέκτη προς “εγγραφή”, και την τιμή (0 ή 1). Στη περίπτωση των LED της πλακέτας περάστε στη συνάρτηση τον πίνακα **p_leds**, που είναι μέρος της δομής **leds** και περιέχει τις τρεις λυχνίες.

Προσοχή: Η έξοδος των LED είναι ανεστραμμένη, δηλαδή αν εισάγετε την εντολή **g_ioport.p_api→pinWrite(leds.p_leds[0], 1);**, το LED 0 θα σβήσει.

Βήμα 4 – Delay

Για να εισάγετε περιοδικότητα στο πρόγραμμα, χρησιμοποιείτε τη συνάρτηση **R_BSP_SoftwareDelay()**. Η συνάρτηση αυτή καθυστερεί την εκτέλεση του προγράμματος και δέχεται ως ορίσματα το χρόνο καθυστέρησης και τις μονάδες του χρόνου. Για παράδειγμα, για μια καθυστέρηση των 500ms θα κληθεί η

```
R_BSP_SoftwareDelay(500,BSP_DELAY_UNITS_MILLISECONDS)  
;
```

Βήμα 5 - Compile

Για να μεταγλωττίσετε το πρόγραμμά σας, επιλέξτε από τη λίστα το φάκελο του project και πατήστε το εικονίδιο **build**. Η πορεία της μεταγλώττισης αποτυπώνεται στο παράθυρο **Console**.



Βήμα 6 – Debug

Αν η μεταγλώττιση ήταν επιτυχής, πατήστε το εικονίδιο **debug**.



Θα μεταφερθείτε στο παράθυρο **debug**, όπου εκεί μπορείτε να ορίσετε **breakpoints** και να ελέγξετε τη ροή της εκτέλεσης του προγράμματος, χρησιμοποιώντας τα εργαλεία ροής **Resume-Pause-Terminate**.

Σημείωση: Το πρόγραμμα περιέχει ήδη κάποια breakpoints, πατήστε **Resume** για να εκτελεστεί.

