ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

Μικροεπεξεργαστές - DSPs

Εργαστηριακή Άσκηση 5: ADC και USB

Εαρινό Εξάμηνο
2022 – 2023

Ενότητα 1: **Εισαγωγή**

1-1Σκοπός της άσκησης

Σκοπός της άσκησης είναι η εξοικείωση με τη μονάδα **adc** και τη μονάδα **usb**. Στην άσκηση θα χρησιμοποιήσετε τον αναλογικό αισθητήρα θερμοκρασίας της πλακέτας για να μετρήσετε την θερμοκρασία του μικροελεγκτή και θα μεταφέρετε την τιμή στον υπολογιστή μέσω usb.

1-2 Board setup και Δημιουργία Project.

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.

Στη συνέχεια εκτελέστε το **e² studio** και επιλέξτε τις ίδιες ρυθμίσεις με την προηγούμενη άσκηση.
(“**Create a new C/C++ project**”, “**Renesas Synergy C Executable Project**”, “**S7G2 SK**”, **BSP)**.

Ενότητα 2 : Μονάδα ADC

2-1: Εισαγωγικές έννοιες

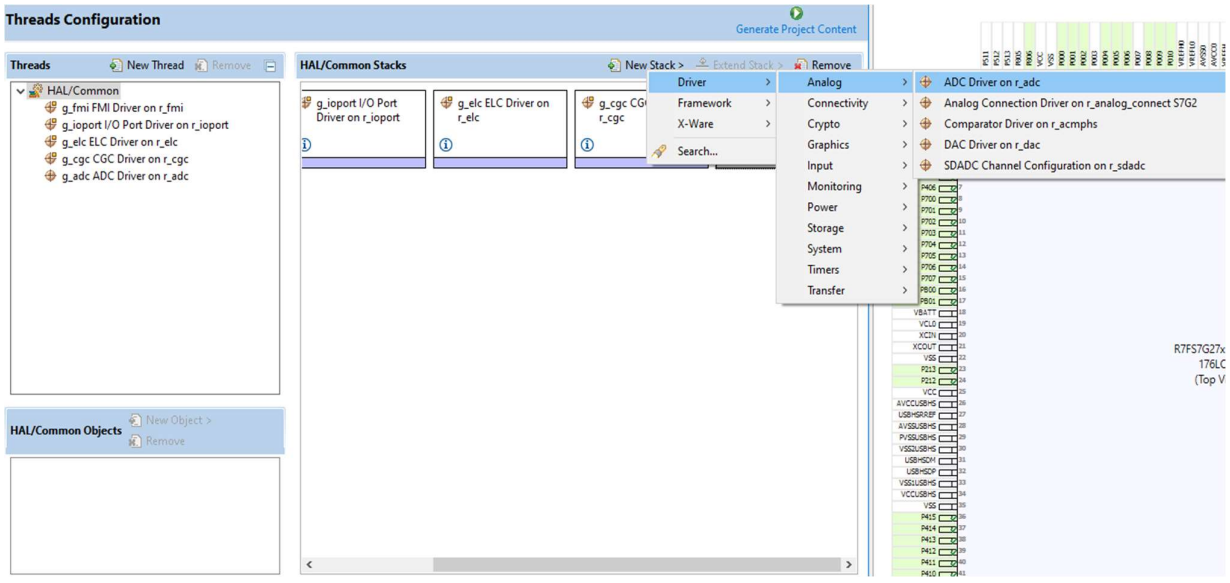
Πολλές φορές οι πληροφορίες που δέχεται ο μικροελεγκτής από τον "έξω κόσμο" βρίσκονται σε αναλογική μορφή. Για την επεξεργασία αναλογικών σημάτων από τον μικροελεγκτή είναι απαραίτητη η μετατροπή τους σε ψηφιακά. Η περιφερειακή συσκευή που αναλαμβάνει αυτή τη μετατροπή είναι ο **μετατροπέας αναλογικού σήματος σε ψηφιακό (ADC)**.

2-2 : Ενεργοποίηση και Ρύθμιση μονάδας ADC στο γραφικό περιβάλλον.

Για να χρησιμοποιήσετε τη μονάδα ADC σε κάποια εφαρμογή, πρέπει να ακολουθήσετε τα παρακάτω βήματα.

- Επιλέξτε την καρτέλα "**Threads**" από την μπάρα της καρτέλας "**Synergy Configuration**".

- Για την εισαγωγή του κατάλληλου stack επιλέξτε "**New Stack** ", "**Driver**", "**Analog**", "**ADC driver on r_adc**" όπως φαίνεται στην παρακάτω φωτογραφία.



- Από την καρτέλα "**Threads**" επιλέξτε το stack που ορίσατε στο προηγούμενο βήμα.

- Στη συνέχεια επιλέξτε την καρτέλα "**Properties**". Σε αυτή μπορείτε να αλλάξετε τις ρυθμίσεις για τη μονάδα adc.

- Πατήστε την επιλογή **Generate Project Content**. Με αυτό θα παραχθούν τα κατάλληλα αρχεία για τη χρήση του stack και θα εισαχθούν αυτόματα στο φάκελο του project σας.

Ακολουθεί μια σύντομη εξήγηση των πεδίων στη καρτέλα properties:

1. Το πεδίο **Parameter Checkings** αφορά το αν ο κώδικας για τον έλεγχο παραμέτρων περιλαμβάνεται στην εφαρμογή.

2. Το πεδίο **Name** αφορά το όνομα του module, που θα χρησιμοποιηθεί.

3. Το πεδίο **Unit** αφορά τη μονάδα adc που θα χρησιμοποιηθεί. Η πλακέτα S7G2 διαθέτει 2 μονάδες, τη 0 και την 1.

4. Το πεδίο **Resolution** αφορά τη διακριτικότητα του αποτελέσματος μέτρησης της μονάδας, δηλαδή το πόσα bits θα είναι το μέγεθος του αποτελέσματος.

5. Το πεδίο **Alignment** αφορά την ευθυγράμμιση του αποτελέσματος.

6. Το πεδίο **Clear after read** αφορά το αν ο καταχωρητής του αποτελέσματος θα αδειάζει, μετά την ανάγνωση του αποτελέσματος. Αν η επιλογή είναι ενεργοποιημένη, η παρακολούθηση της τιμής του καταχωρητή με τον debugger, θα έχει πάντα ως αποτέλεσμα το 0.

7. Το πεδίο **Mode** αφορά τον τρόπο λειτουργίας της μονάδας adc που έχει οριστεί παραπάνω.

   -- **Single scan:** Μετατρέπει την αναλογική είσοδο **του επιλεγμένου καναλιού**, μια φορά σε κάθε εκτέλεση. Η μετατροπή γίνεται διαδοχικά κατά αύξουσα σειρά, με βάση τον αριθμό του κάθε καναλιού.

   -- **Continuous scan:** Μετατρέπει την αναλογική είσοδο **των επιλεγμένων καναλιών,** μια φορά σε κάθε εκτέλεση. Η μετατροπή γίνεται διαδοχικά κατά αύξουσα σειρά, με βάση τον αριθμό του κάθε καναλιού.

   -- **Group Scan:** Επιτρέπει στο χρήστη, να εισάγει κανάλια σε ένα από τα δύο groups (A ή B) . Η μετατροπή της αναλογικής εισόδου των καναλιών του κάθε group γίνεται διαδοχικά κατά αύξουσα σειρά, με βάση τον αριθμό του κάθε καναλιού, όταν ληφθεί εντολή για ενεργοποίηση του συγκεκριμένου group.

8. Το πεδίο **Channels 0-27** αφορά τον αριθμό των καναλιών που θα χρησιμοποιηθούν.

   -- **Αν έχει επιλεχτεί το Normal mode:** Αφορά τα κανάλια που θα χρησιμοποιηθούν στη μονάδα adc.
   -- **Αν έχει επιλεχτεί το Group mode:** Αφορά και το σε ποιο group ανήκει το αντίστοιχο κανάλι.

9. Τα πεδία **Temperature/Voltage Sensor** αφορουν το αν και σε ποιο group θα χρησιμοποιηθουν ο ανιχνευτές θερμοκρασίας και τάσης αντίστοιχα.

10. Το πεδίο **Scan Mask Group B** αφορά το ποια κανάλια ανοίκουν στον Group B. Προσοχή! Χρησιμοποιείται μόνο στο group mode και τα κανάλια που οριστούν εδώ, δεν πρέπει να ανήκουν και στο group A.

11. Το πεδίο **Normal/Group A Trigger** αφορά τον τύπο της ενεργοποίησης της μονάδας. Αν το group mode είναι ενεργοποιημένο, τότε η επιλογή αυτή, χρησιμοποιείται για τη ρύθμιση της ενεργοποίησης του Group A.

12. Το πεδίο **Group B Trigger** αφορά τον τύπο της ενεργοποίησης του group B.

13. Το πεδίο **Group priority** ρυθμίζει τον τρόπο αλληλεπίδρασης των 2 group. Για παράδειγμα το αν μια μέτρηση του group B σταματήσει, όταν ενεργοποιηθεί μέτρηση του group A. Το πεδίο αυτό έχει νόημα, μόνο όταν το group mode είναι ενεργοποιημένο.

14. Το πεδίο **Add/Average count** δηλώνει το αν θα γίνει κάποια πράξη πρόσθεσης ή μέσου όρου, μεταξύ των τιμών ενός αριθμού καναλιών της μονάδας.

15. Το πεδίο **Channels 0-27** αφορά το ποια κανάλια θα χρησιμοποιηθούν για την παραπάνω λειτουργία. Το πεδίο έχει νόημα, μόνο όταν είναι ενεργοποιημένη η παραπάνω λειτουργία.

16. Τα πεδία **Temperature/Voltage Sensor** αφορουν το αν οι ανιχνευτές θερμοκρασίας/Τάσης χρησιμοποιηθούν για τις πράξεις πρόσθεσης ή μέσου όρου.

17. Το πεδίο **Channels 0-2** αφορά το ποια κανάλια χρησιμοποιούν τις ενημερωμένες τιμές του sample and hold.

18. Το πεδίο **Sample hold states** αφορά τον αριθμό των καταστάσεων για την παραπάνω λειτουργία.

19. Το πεδίο **Callback** αφορά το αν θα χρησιμοποιηθεί κάποια συνάρτηση callback του χρήστη, κατά τη λήξη των μετρήσεων. Αν χρησιμοποιηθεί, τότε καλείται μέσω των Interrupt, κάθε φορά που ολοκληρώνεται μια μέτρηση της μονάδας adc.

20. Το πεδίο **Scan end interrupt priority,** αφορά την προτεραιότητα του interrupt, που θα εκτελεστεί κατά τη λήξη της μέτρησης.

21. Το πεδίο **Scan end Group B interrupt priority,** αφορά την προτεραιότητα του interrupt, που θα εκτελεστεί κατά τη λήξη της μέτρησης του group B.

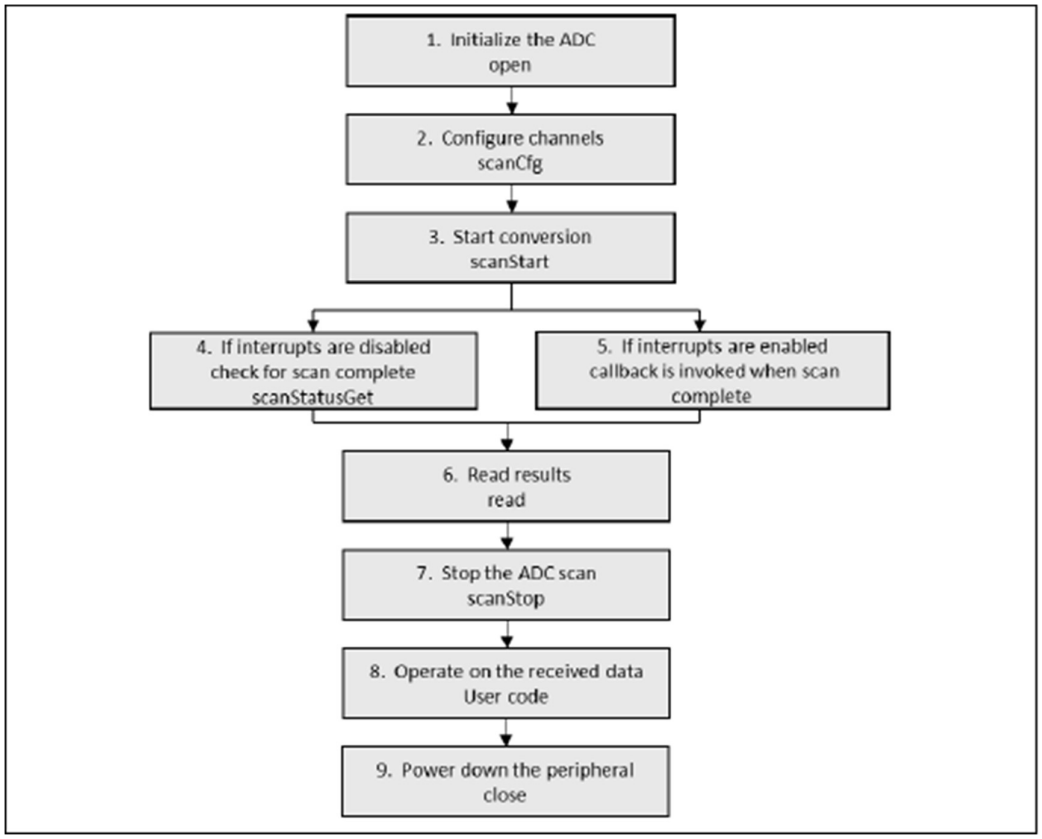
2-3 : Χρήση της μονάδας ADC σε εφαρμογές.

Η χρήση της μονάδας σε μια εφαρμογή μπορεί να γίνει μέσω των συναρτήσεων του HAL. Ακολουθεί μια λίστα με τις συναρτήσεις αυτές.

Όνομα	Κλήση και περιγραφή λειτουργίας
.**open**	**g_adc.p_api->open(g_adc.p_ctrl, g_adc.p_cfg);** Αρχικοποιεί τη μονάδα adc. Ορίζει τον τρόπο λειτουργίας, τις πηγές ενεργοποίησης, την προτεραιότητα των interrupt και τις ρυθμίσεις των καναλιών και των ανιχνευτών.
.**scanCfg**	**g_adc.p_api->scanCfg(g_adc.p_ctrl, g_adc.p_channel_cfg);** Ρυθμίζει τα κανάλια, τα group και τον τρόπο ενεργοποίησης της μέτρησης, που θα χρησιμοποιήσει η μονάδα, που ορίστηκε με την εντολή open.
.**scanStart**	**g_adc.p_api->scanStart(g_adc.p_ctrl);** Ξεκινάει τις μετρήσεις.
.**scanStop**	**g_adc.p_api->scanStop(g_adc.p_ctrl);** Σταματάει τις μετρήσεις
.**read**	**g_adc.p_api->read(g_adc.p_ctrl, ADC_REG_CHANNEL_13, &adc_data);** Χρησιμοποιείται για την ανάγνωση της μετατροπής του αποτελέσματος της μέτρησης της μονάδας adc.
.**sampleStateCountSet**	**g_adc.p_api-> sampleStateCountSet(g_adc.p_ctrl,&adc_sample);** Ρυθμίζει τις καταστάσεις δειγματοληψίας του συγκεκριμένου καναλιού.
.**close**	**g_adc.p_api->close(g_adc.p_ctrl);** Σταματάει τη λειτουργία της συγκεκριμένης μονάδας adc.

.scanStatusGet	**g_adc.p_api->scanStatusGet(g_adc.p_ctrl);** Ελέγχει την κατάσταση της μέτρησης.

Η διαδικασία για τη χρήση της μονάδας σε μια εφαρμογή είναι η ακόλουθη:



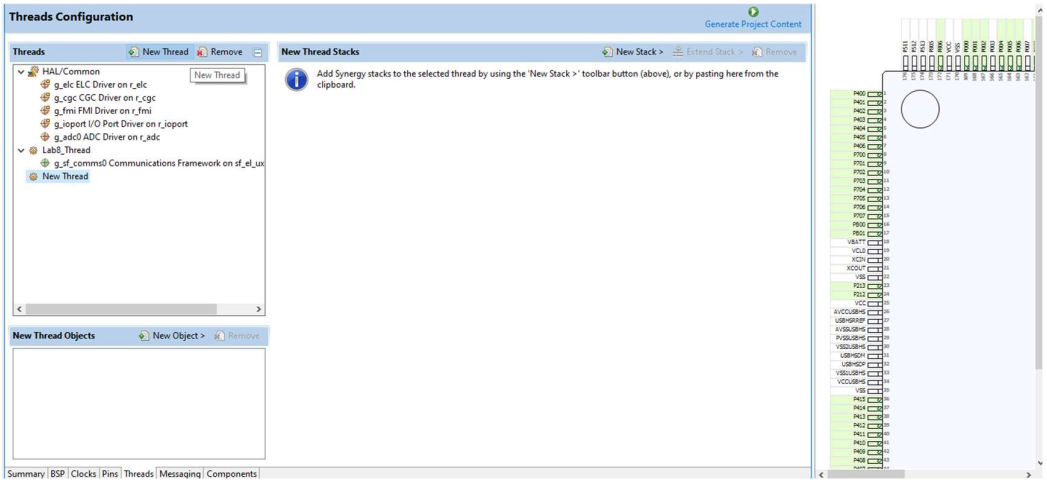
Ενότητα 3 : Μονάδα USB

3-1 : Ενεργοποίηση και Ρύθμιση της μονάδας USB στο γραφικό περιβάλλον.

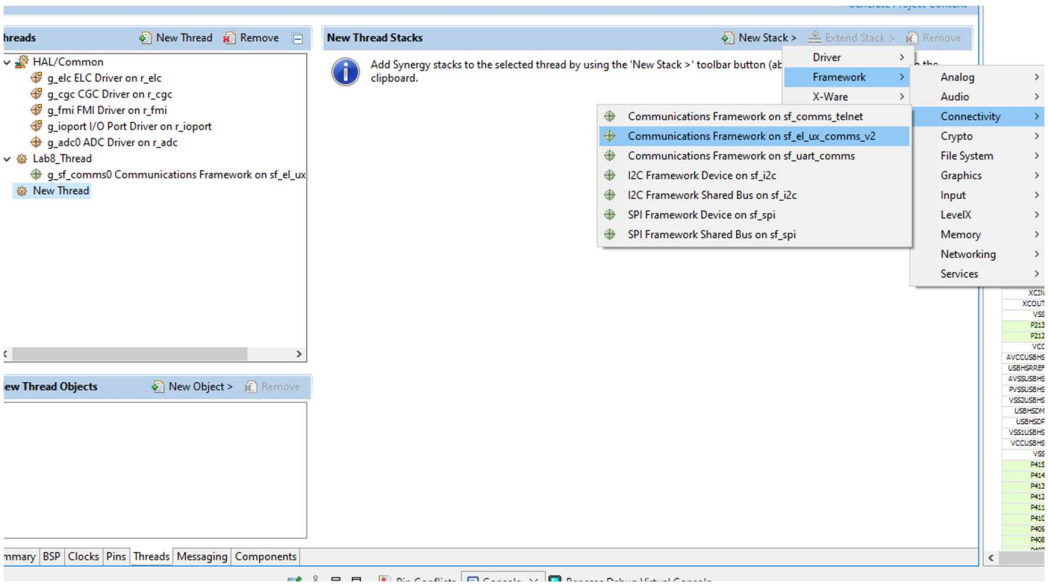
Για να χρησιμοποιήσετε τη μονάδα USB σε κάποια εφαρμογή, πρέπει να ακολουθήσετε τα παρακάτω βήματα.

- Επιλέξτε την καρτέλα "**Threads**" από την μπάρα της καρτέλας "**Synergy Configuration**".

- Για την εισαγωγή του κατάλληλου **THREAD** επιλέξτε "**New thread** ".

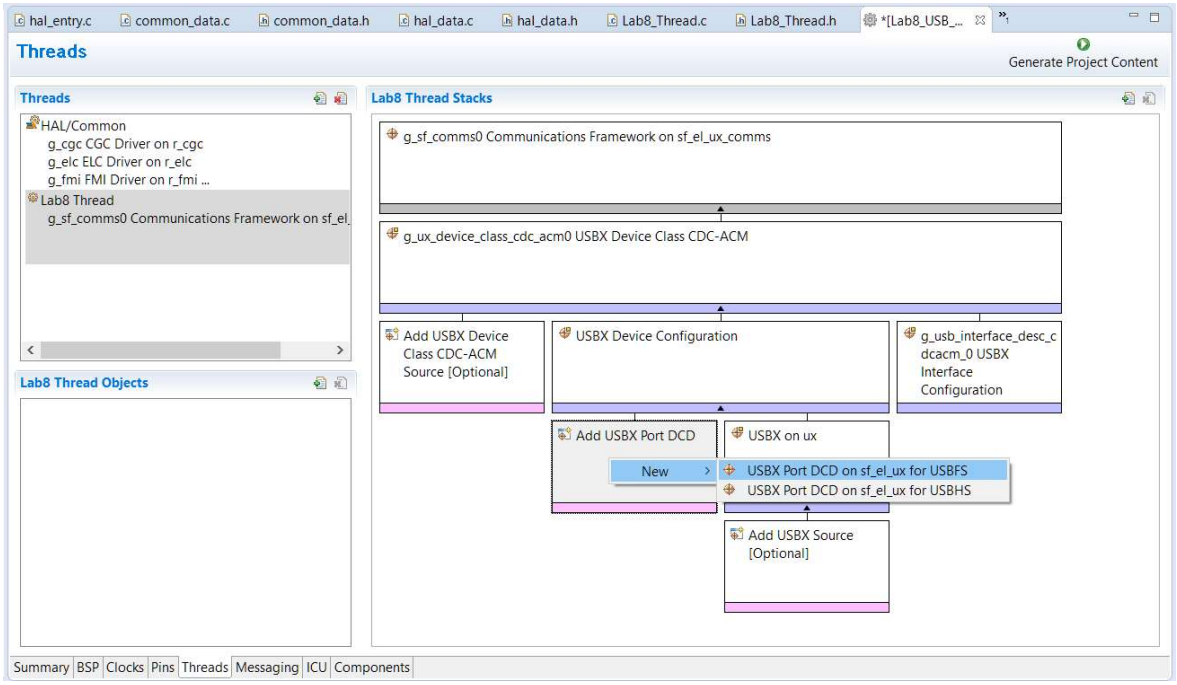


- Για την εισαγωγή του κατάλληλου stack, στο thread που ορίσατε παραπάνω, επιλέξτε "**New Stack** ", "**Framework**", "**Connectivity**", "**Communications Framework on sf_el_ux_comms_v2**" όπως φαίνεται στην παρακάτω φωτογραφία.



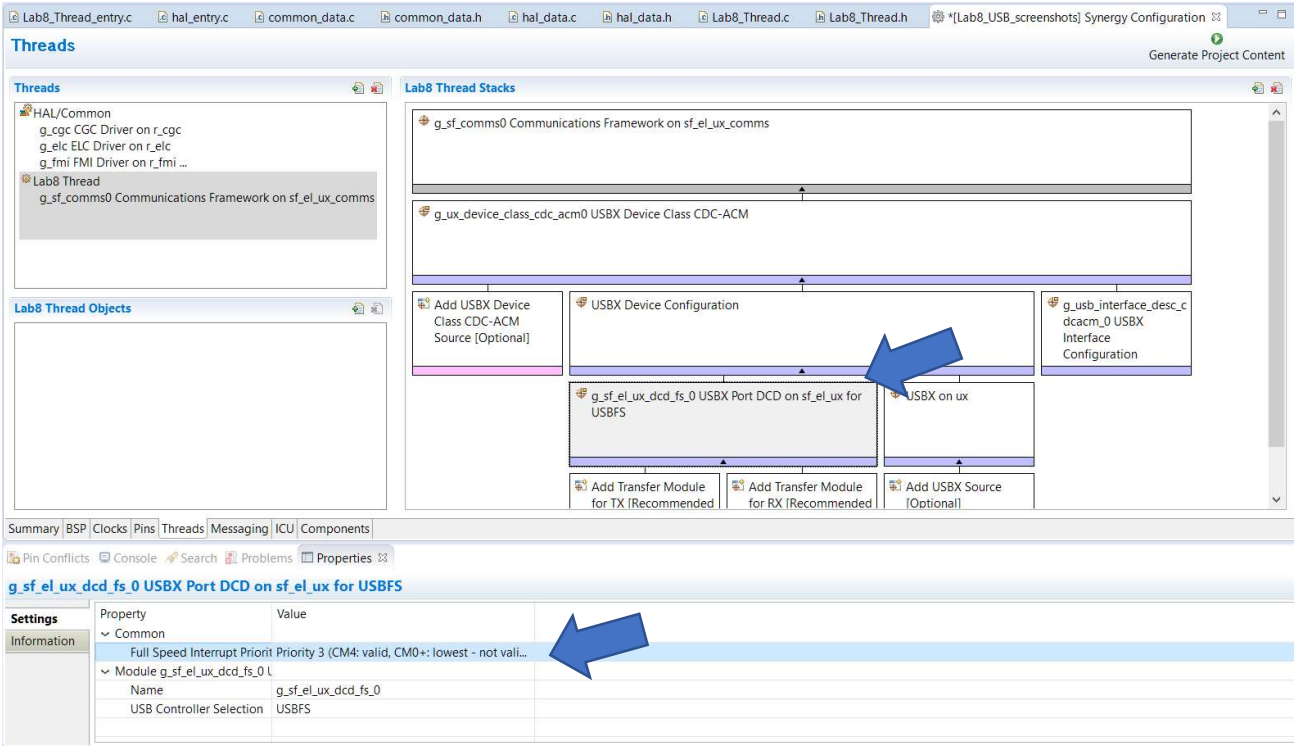
- Από την καρτέλα “**Threads**” επιλέξτε το stack που ορίσατε στο προηγούμενο βήμα.

- Επιλέξτε την καρτέλα “ **Add USBX Port DCD**”, και μετά “ **"USBX Port DCD on sf_el_ux for USBFS**”, όπως φαίνεται στην παρακάτω φωτογραφία.



- Στη συνέχεια επιλέξτε την καρτέλα “**Properties**”. Σε αυτή μπορείτε να αλλάξετε τις ρυθμίσεις για τη μονάδα usb. Η μονάδα USB αποτελείται από πολλές καρτέλες. Θα χρειαστεί να ρυθμίσετε μόνο αυτή, που αναφέρεται στη συνέχεια.

- Μεταβείτε στην καρτέλα “**USBX Port DCD”** και ρυθμίστε το πεδίο “**Full Speed Interrupt Priority”**, ώστε να έχει μια τιμή συμβατή με την πλακέτα. Για παράδειγμα priority 5. Οι ρυθμίσεις φαίνονται στην ακόλουθη φωτογραφία. Το πεδίο **Full Speed Interrupt Priority** στην καρτέλα USBX Port DCD, αφορά το επίπεδο προτεραιότητας, που θα έχει το full speed interrupt.



- Πατήστε την επιλογή **Generate Project Content**. Με αυτό θα παραχθούν τα κατάλληλα αρχεία για τη χρήση του stack και θα εισαχθούν αυτόματα στο φάκελο του project σας.

Ακολουθεί μια λίστα με συναρτήσεις που θα χρησιμοποιηθούν για τη λειτουργία της μονάδας USB.

Όνομα	**Κλήση και περιγραφή λειτουργίας**
.write	**g_sf_comms.p_api->write(g_sf_comms.p_ctrl, Transfer_msg, (uint32_t) strlen((const CHAR*)Transfer_msg), TX_NO_WAIT);** Μεταφορά δεδομένων μέσω του USB
.read	**g_sf_comms.p_api->read(g_sf_comms.p_ctrl, &Received_msg, 1u, TX_WAIT_FOREVER);** Ανάγνωση δεδομένων από το USB και αποθήκευση τους στη μεταβλητή Received_msg
sprintf	**sprintf(msg_array, sizeof(msg_array), "Value of float variable is %f\r\n", float_variable);** Αποθήκευση ενός συνόλου από χαρακτήρες, σε έναν array τύπου char.

Επεξήγηση των ορισμάτων κάθε συνάρτησης.

1) **Sprintf**: Δέχεται 4 ορίσματα.
   -- **msg_array:** Ο array στον οποίο θα αποθηκευτεί το μήνυμα.
   -- **sizeof(msg_array):** Το μέγεθος του array στον οποίο θα αποθηκευτεί το μήνυμα.
   --**"Value of float variable is %f\r\n":** Το μήνυμα που θα αποθηκευτεί στον array. Ο χαρακτήρας %f σημαίνει ότι το μήνυμα θα συνεχιστεί από την τιμή μιας μεταβλητής τύπου float.
   -- **float_variable:** Η μεταβλητή που θα αποθηκευτεί και αυτή στο μήνυμα.

2) **write:** Δέχεται 4 ορίσματα.
   -- **g_sf_comms.p_ctrl:** Δείκτης στη μονάδα ελέγχου που είναι υπεύθηνη για την επικοινωνία.
   -- **Transfer_msg :** Δείκτης στον array που περιέχει το μήνυμα που θα μεταφερθεί.
   --**(uint32_t) strlen((const CHAR*)Transfer_msg):** Δηλώνει το μέγεθος του μηνύματος που θα μεταφερθεί. Το μέγεθος μετριέται σε bytes.
   -- **TX_NO_WAIT:** Σημαία κατάστασης, που δηλώνει τον τρόπο λειτουργίας της **write.** Στη συγκεκριμένη περίπτωση, η συνάρτηση write επιστρέφει αμέσως μετά τη μεταφορά του μηνύματος.

3) **read:** Δέχεται 4 ορίσματα.
   -- **g_sf_comms.p_ctrl:** Δείκτης στη μονάδα ελέγχου που είναι υπεύθηνη για την επικοινωνία.
   --**&Received_msg:** Pointer στη μεταβλητή, στην οποία θα αποθηκευτεί το μήνυμα.
   --**1u:** Δηλώνει το μέγιστο αριθμό των bytes, που πρόκειται να διαβαστούν. Στη συγκεκριμένη περίπτωση η read θα διαβάσει 1 byte.
   -- **TX_WAIT_FOREVER:** Σημαία κατάστασης, που δηλώνει τον τρόπο λειτουργίας της **read.** Στη συγκεκριμένη περίπτωση, η read περιμένει μέχρι να ληφθούν όλα τα δεδομένα.

3-3 : Ενεργοποίηση και Ρύθμιση της βιβλιοθήκης newlib-nano.

Η τιμή της θερμοκρασίας θα είναι τύπου float. Για τη χρήση μεταβλητών τύπου float στην sprintf, πρέπει να ενεργοποιηθεί η βιβλιοθήκη new-lib nano. Για να την ενεργοποιήσετε ακολουθήστε τα παρακάτω βήματα.

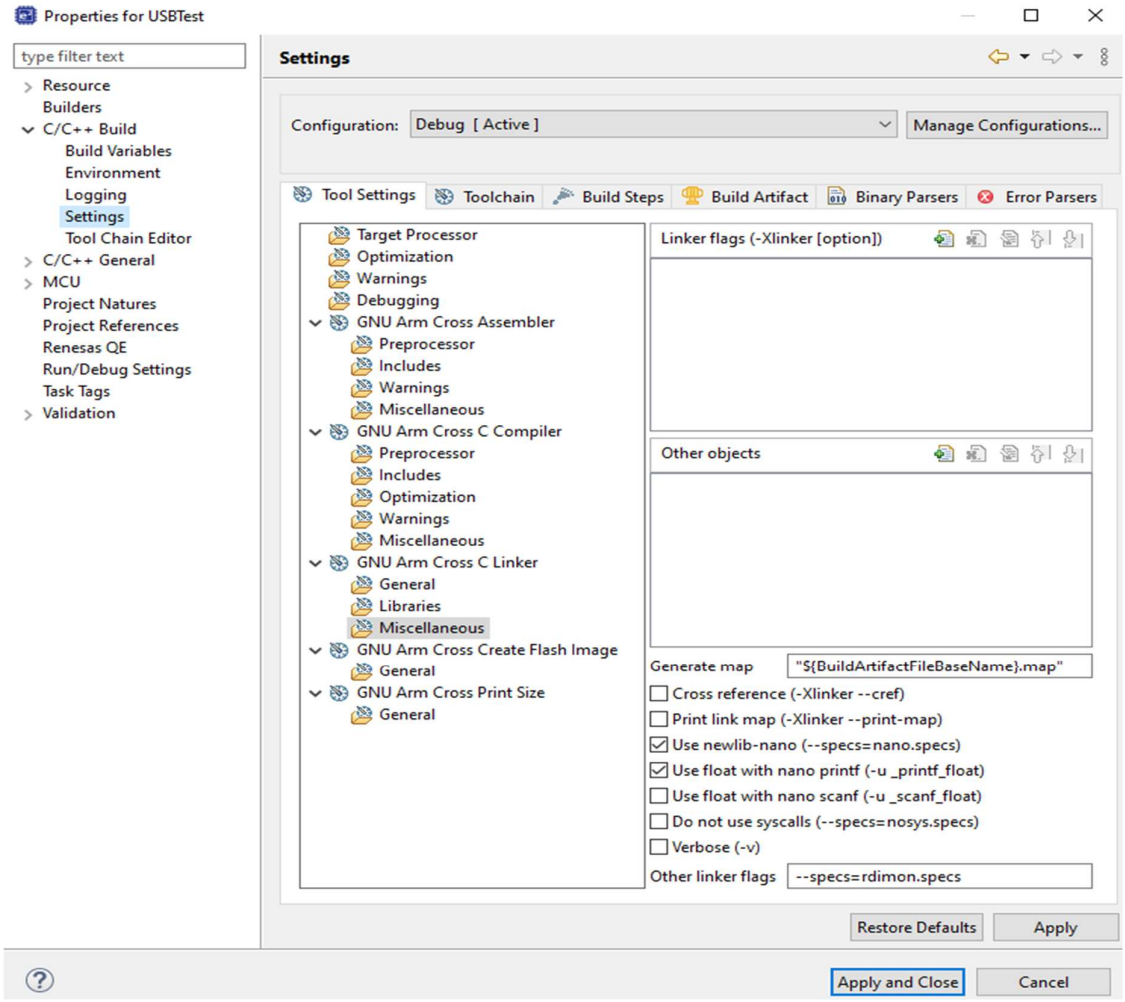
- Από την πάνω εργαλειοθήκη του e2 studio, επιλέξτε την ένδειξη "**project**" και στη συνέχεια επιλέξτε την ένδειξη "**properties**".

- Στο παράθυρο που θα ανοίξει, επιλέξτε **"C/C++ Build"**, **"Settings"**, **"Tool Settings"**, **"Cross ARM C Linker"**, **"Miscellaneous"**.

- Επιλέξτε τα κουτία **"use newlib-nano"** και **"use float with nano printf".**

- Στην μπάρα **"other linker flags",** εισάγετε την ακόλουθη εντολή εντολή : **--specs=rdimon.specs** .

- Βεβαιωθείται ότι οι ρυθμίσεις σας μοιάζουν με αυτής της παρακάτω φωτογραφίας και πατήστε την επιλογή **"apply and close".**


Properties for USBTest
type filter text
Resource
Builders
C/C++ Build
Build Variables
Environment
Logging
Settings
Tool Chain Editor
C/C++ General
MCU
Project Natures
Project References
Renesas QE
Run/Debug Settings
Task Tags
Validation
Settings
Configuration: Debug [Active]
Manage Configurations...
Tool Settings
Toolchain
Build Steps
Build Artifact
Binary Parsers
Error Parsers
Target Processor
Optimization
Warnings
Debugging
GNU Arm Cross Assembler
Preprocessor
Includes
Warnings
Miscellaneous
GNU Arm Cross C Compiler
Preprocessor
Includes
Optimization
Warnings
Miscellaneous
GNU Arm Cross C Linker
General
Libraries
Miscellaneous
GNU Arm Cross Create Flash Image
General
GNU Arm Cross Print Size
General
Linker flags (-Xlinker [option])
Other objects
Generate map
"${BuildArtifactFileBaseName}.map"
Cross reference (-Xlinker --cref)
Print link map (-Xlinker --print-map)
Use newlib-nano (--specs=nano.specs)
Use float with nano printf (-u _printf_float)
Use float with nano scanf (-u _scanf_float)
Do not use syscalls (--specs=nosys.specs)
Verbose (-v)
Other linker flags
--specs=rdimon.specs
Restore Defaults
Apply
Apply and Close
Cancel

Ενότητα 4: **Εφαρμογή μέτρησης θερμοκρασίας.**

4-1: Σκοπός Άσκησης

Αναπτύξετε εφαρμογή, η οποία θα μετράει τη θερμοκρασία του μικροελεγκτή και θα μεταφέρει την τιμή στον υπολογιστή. Η εφαρμογή θα πρέπει να χρησιμοποιεί την μονάδα ADC και να μετατρέπει τη τιμή της σε θερμοκρασία και στη συνέχεια θα πρέπει να εμφανίζει το αποτέλεσμα στην εφαρμογή τερματικού tera term στον υπολογιστή.

4-2: Βαθμονόμηση της μονάδας ADC για την εφαρμογή.

Η μονάδα adc υποστηρίζει τη χρήση του ενσωματωμένου στην πλακέτα ανιχνευτή θερμοκρασίας. Η τιμή που επιστρέφει μία μέτρηση της μονάδας ADC όμως, αντιστοιχεί σε μονάδες τάσης και επομένως πρέπει να μετατραπεί σε θερμοκρασία. Για να μετατρέψετε την τιμή τάσης σε θερμοκρασία χρησιμοποιείστε τον ακόλουθο τύπο:

$$T = (Vs – V1) / slope + T1$$

- T: Η θερμοκρασία που προκύπτει από τη μετατροπή (°C).

- Vs: Η τιμή της τάσης από τη μέτρηση της θερμοκρασίας (Volt).

- Slope: Κλίση της ευθείας, προκύπτει από τον τύπο Slope = (V2 – V1)/(T2 – T1). Σας δίνετε ότι το **slope = 9.125**.

- T1: Τιμή που προκύπτει από πειραματική μέτρηση. Σας δίνεται ότι **T1 = 22**.

- V1: Τιμή που αντιστοιχεί στην T1. Σας δίνεται ότι **V1 = 1520**.

4-4: Οδηγίες Υλοποίησης.

- Δημιουργείστε ένα νέο thread σύμφωνα με τα βήματα της ενότητας 3-1.
- Στο thread που δημιουργήσατε προσθέστε το stack για τον adc driver σύμφωνα με τις οδηγίες της ενότητας 2-2.
- Στις ρυθμίσεις της μονάδας ADC επιλέξτε Normal Mode και ενεργοποιείστε τη μονάδα μέτρησης θερμοκρασίας.
- Ο κώδικας που θα αναπτύξετε για την εφαρμογή θα πρέπει να βρίσκεται στο αρχείο **new_thread0_entry.c** .
- Ορίστε πίνακα κατάλληλου μεγέθους, για την αποθήκευση του μηνύματος που θα μεταφέρετε μέσω του usb.
- Χρησιμοποιείστε την εντολή sprintf() για την αποθήκευση του μηνύματος στον πίνακα. Για παράδειγμα η ακόλουθη εντολή αποθηκεύει το μήνυμα “value of variable is “ και την τιμή της μεταβλητής value στον πίνακα message.

  ```
  sprintf(message, sizeof(message), " value of variable is %f\r\n", value);
  ```

4-3: Οδηγίες Χρήσης Τερματικού Tera Term.

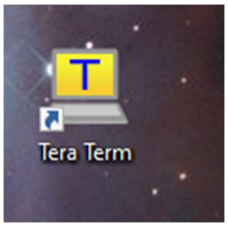
Η εφαρμογή tera term αποτελεί ένα τερματικό στον υπολογιστή και θα το χρησιμοποιήσετε για την εμφάνιση των αποτελεσμάτων μέτρησης της θερμοκρασίας στον υπολογιστή.

Για να χρησιμοποιήσετε την εφαρμογή tera term, ακολουθείστε τις παρακάτω οδηγίες.

- Συνδέστε το δεύτερο καλώδιο usb στη θύρα J5 της πλακέτας. Η θύρα j5 βρίσκεται δίπλα από τη θύρα J19.

- Κάντε build και debug το πρόγραμμά σας. Η εφαρμογή tera term μπορεί να ανιχνεύσει την πλακέτα μέσω usb, μόνο αν έχει ενεργή συνεδρία debug.

- Μεταβείτε στην επιφάνεια εργασίας και ανοίξτε την εφαρμογή tera term, το εικονίδιο της οποίας είναι το ακόλουθο:



- Με την εκτέλεση της εφαρμογής θα αναδυθεί το παράθυρο ρυθμίσεων, με την επιλογή του τύπου σύνδεσης. Επιλέξτε **serial** και το port, που αντιστοιχεί στην πλακέτα, για παράδειγμα com7. Το παράθυρο ρυθμίσεων είναι το ακόλουθο: