



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών
— ΙΔΡΥΘΕΝ ΤΟ 1837 —

Μικροεπεξεργαστές - DSPs

Εργαστηριακή Άσκηση 3:
Interrupts, Timers/Counters, Reaction Game

Ενότητα 1: Εισαγωγή

1-1 Σκοπός της άσκησης

Σκοπός της άσκησης είναι η εξοικείωση με τις μονάδες **interrupt** και **timer/counter** καθώς και η εξοικείωση με την ανάπτυξη εφαρμογών, με χρήση πολλαπλών περιφερειακών μονάδων σε μικροελεγκτές.

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 : Interrupts

2-1: Εισαγωγικές έννοιες

Τα interrupt αναλαμβάνουν τη διακοπή της εκτέλεσης ενός προγράμματος ή μιας διεργασίας και την αίτηση για άμεση εκτέλεση κάποιας άλλης από τον επεξεργαστή. Με την χρήση των interrupt είναι δυνατό να ειδοποιηθεί ο επεξεργαστής, ακόμα και όταν εκτελεί κάποια διεργασία, καθώς και να τεθεί να εκτελέσει κάποια διεργασία μεγαλύτερης προτεραιότητας.

Τα interrupt μπορούν να πυροδοτηθούν, είτε από εσωτερικές περιφερειακές συσκευές του μικροελεγκτή, είτε από εξωτερικές, οι οποίες θα συνδέονται στους ακροδέκτες που γεννούν αιτήσεις για διακοπή (IRQ pin). Τη διαχείριση των αιτήσεων αυτών αναλαμβάνει η μονάδα διαχείρισης διακοπών, η οποία είναι σχεδιασμένη να ειδοποιεί τον επεξεργαστή για διακοπές, λαμβάνοντας υπόψιν τη προτεραιότητα τους. Την εξυπηρέτηση των αιτήσεων διακοπής αναλαμβάνει η ρουτίνα **ISR (Interrupt Service Routine)**, η οποία καλείται από την κεντρική μονάδα επεξεργασίας (CPU). Μια τυπική διαδικασία εξυπηρέτησης μιας αίτησης για διακοπή είναι η ακόλουθη :

Η CPU δέχεται την αίτηση από την μονάδα διακοπών, διακόπτει την εκτέλεση του τρέχοντος προγράμματος και αποθηκεύει όλες τις απαραίτητες πληροφορίες για αυτό στη μνήμη (program counter, δεδομένα κλπ). Στη συνέχεια, εκτελεί το πρόγραμμα που σχετίζεται με την αίτηση διακοπής. Με την ολοκλήρωση του προγράμματος διακοπής γίνεται επαναφορά των πληροφοριών του αρχικού προγράμματος και συνεχίζεται η εκτέλεση του.

Η διαχείριση των interrupt στο μικροελεγκτή S7G2 γίνεται με την συνεργασία δύο μονάδων. Η πρώτη είναι η περιφερειακή συσκευή διαχείρισης αιτήσεων διακοπής **ICU** (**interrupt controller unit**) του μικροελεγκτή και η δεύτερη είναι η μονάδα διαχείρισης διακοπών του επεξεργαστή Cortex-M4, **NVIC** (**nested vectored interrupt controller**). Όταν κάποια συσκευή γεννάει μια αίτηση για διακοπή, τότε πρέπει πρώτα να εγκριθεί από τις μονάδες ICU και NVIC πριν διακοπεί η εκτέλεση του τρέχοντος προγράμματος.

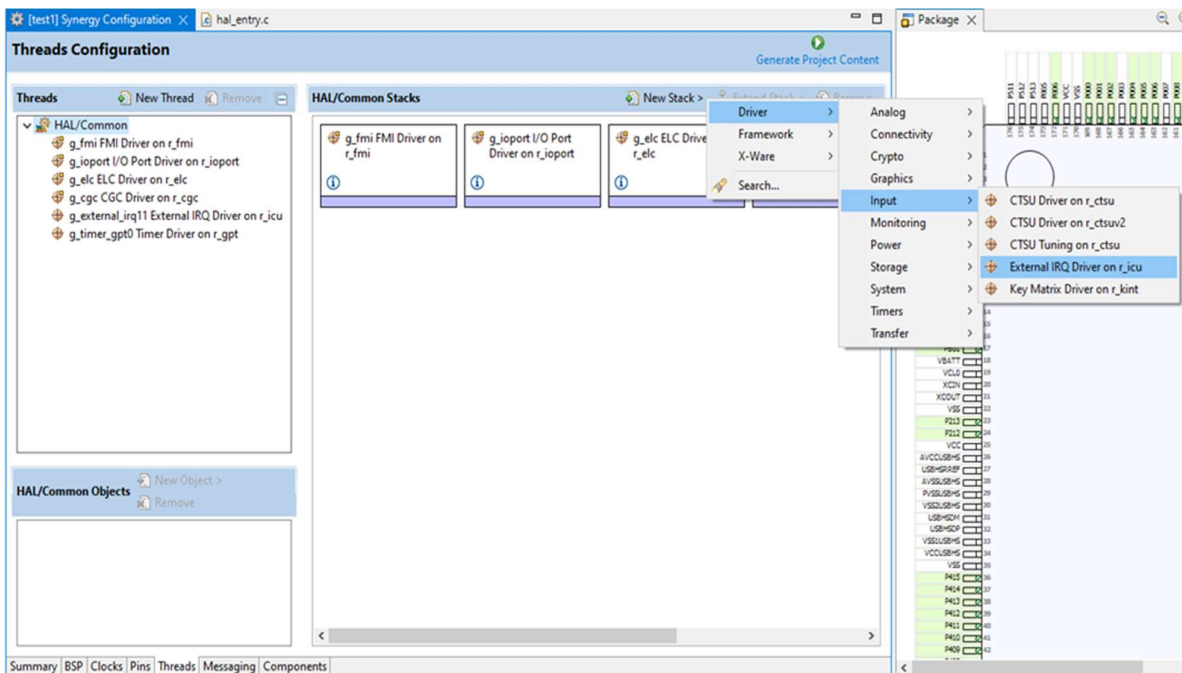
Η μονάδα ICU αναλαμβάνει την ανίχνευση των interrupt και την ενεργοποίηση και απενεργοποίηση τους. Η ανίχνευση των interrupt από εξωτερικές συσκευές γίνεται, είτε από το επίπεδο της τάσης στους ακροδέκτες IRQ, είτε από την αλλαγή στην τάση (rising edge, falling edge, rising and falling edge).

Όπως όλες οι περιφερειακές συσκευές στον μικροελεγκτή, η λειτουργία της ICU μπορεί να ρυθμιστεί με τους καταχωρητές ελέγχου της μονάδας. Στα πλαίσια αυτής της άσκησης, θα γίνει χρήση του BSP (board support package) που δίνει τη δυνατότητα στον χρήστη να ρυθμίσει τη λειτουργία των μονάδων μέσω των διάφορων συναρτήσεων και του γραφικού περιβάλλοντος.

2-2 : Ενεργοποίηση και Ρύθμιση της μονάδας ICU στο γραφικό περιβάλλον.

Για να χρησιμοποιήσετε τη μονάδα ICU 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**”, “**Input**”, “**External irq driver on r_icu**” 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 ICU.
- 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 **Name** αφορά το όνομα του module που χρησιμοποιούμε.
2. Το πεδίο **Channel** αφορά το κανάλι με το οποίο θα συνδεθεί το αντίστοιχο module. Τα κανάλια 10,11 αντιστοιχούν στους διακόπτες **S5** και **S4** της πλακέτας, αντίστοιχα.
3. Το πεδίο **Triger** αφορά το επίπεδο τάσης, στο οποίο θα ανιχνεύεται η αίτηση για τα interrupts.
 - Αλλαγή από ψηλή σε χαμηλή τάση (falling edge).
 - Αλλαγή από χαμηλή σε ψηλή τάση (rising edge).
 - Αλλαγή τάσης (Rising and falling edges).
4. Το πεδίο **Digital Filtering** αφορά στην ενεργοποίηση ψηφιακού φίλτρου.
5. Το πεδίο **Callback** αφορά το όνομα της συνάρτησης, που έχει ορίσει ο χρήστης και θα εκτελείται κάθε φορά που υπάρχει κάποια αίτηση για διακοπή.
6. Το πεδίο **Pin Interrupt Priority** αφορά το ποιο interrupt θα εξυπηρετηθεί πρώτο.

Για να χρησιμοποιήσετε τη μονάδα ICU σε κάποια εφαρμογή, πρέπει να προσθέσετε στο κώδικα σας την ακόλουθη εντολή:

```
g_external_irq11.p_api→open(g_external_irq11.p_ctrl,  
g_external_irq11.p_cfg);
```

Η εντολή αυτή αρχικοποιεί τη μονάδα ICU με τις ρυθμίσεις που έχετε επιλέξει.

2-3: Εργαστηριακή Άσκηση

Αναπτύξτε εφαρμογή, στην οποία θα χρησιμοποιείτε τη μονάδα ICU. Η εφαρμογή θα πρέπει να αναβοσβήνει περιοδικά ένα από τα LEDs της πλακέτας. Το ποιο LED θα αναβοσβήνει, θα εξαρτάται από το αν έχει γίνει κάποια αίτηση για διακοπή (interrupt).

Χρησιμοποιείστε το διακόπτη S4 της πλακέτας για τη γέννηση των interrupt. Ορίστε μια μεταβλητή flag (τιμή 0 ή 1), η οποία θα είναι καθολική (global) και θα είναι τύπου 'volatile int'. Το προσδιοριστικό 'volatile' ενημερώνει τον compiler ότι η τιμή της μεταβλητής αλλάζει εκτός της 'hal_entry()'.

Δημιουργείστε μια συνάρτηση με όνομα 'switch_callback()', η οποία θα καλείται με κάθε αίτηση για διακοπή (κάθε φορά που πατάτε το διακόπτη S4) και θα αλλάζει τη τιμή της μεταβλητής flag.

Στη hal_entry(), απενεργοποιείστε τα LEDs και δημιουργείστε μια super-loop. Σε αυτή εισάγετε έναν έλεγχο για τη τιμή της μεταβλητής flag ώστε να υπάρχει επιλογή στο LED που θα αναβοσβήνει.

Η συνάρτηση **switch_callback** θα πρέπει να έχει την ακόλουθη δομή και θα πρέπει να βρίσκεται εκτός της hal_entry():

```
void switch_callback(external_irq_callback_args_t *p_args)
{
    //your code
}
```

Ρυθμίστε την μονάδα σύμφωνα με τον παρακάτω πίνακα:

g_external_irq0 External IRQ Driver on r_licu		
Settings	Property	Value
API Info	▼ Common	
	Parameter Checking	Default (BSP)
	▼ Module g_external_irq0 External IRQ Driver on r_licu	
	Name	g_external_irq0
	Channel	11
	Trigger	Rising
	Digital Filtering	Disabled
	Digital Filtering Sample Clock (Only valid when Digital Fil	PCLK / 64
	Interrupt enabled after initialization	True
	Callback	switch_callback
	Pin Interrupt Priority	Priority 5

Ενότητα 3: Timers/counters.

3-1: Εισαγωγικές έννοιες

Η μονάδα timer-counter είναι ακόμη μια ενσωματωμένη περιφερειακή συσκευή, σκοπός της οποίας είναι να μετράει χρονική διάρκεια, να μετράει γεγονότα και να παράγει σήματα PWM.

Οι timers/counters ακόμη, μπορούν να πυροδοτούν αιτήσεις για διακοπή (interrupt) καθώς και ADC μετατροπές, με τη λήξη της χρονικής διάρκειας που μετράνε.

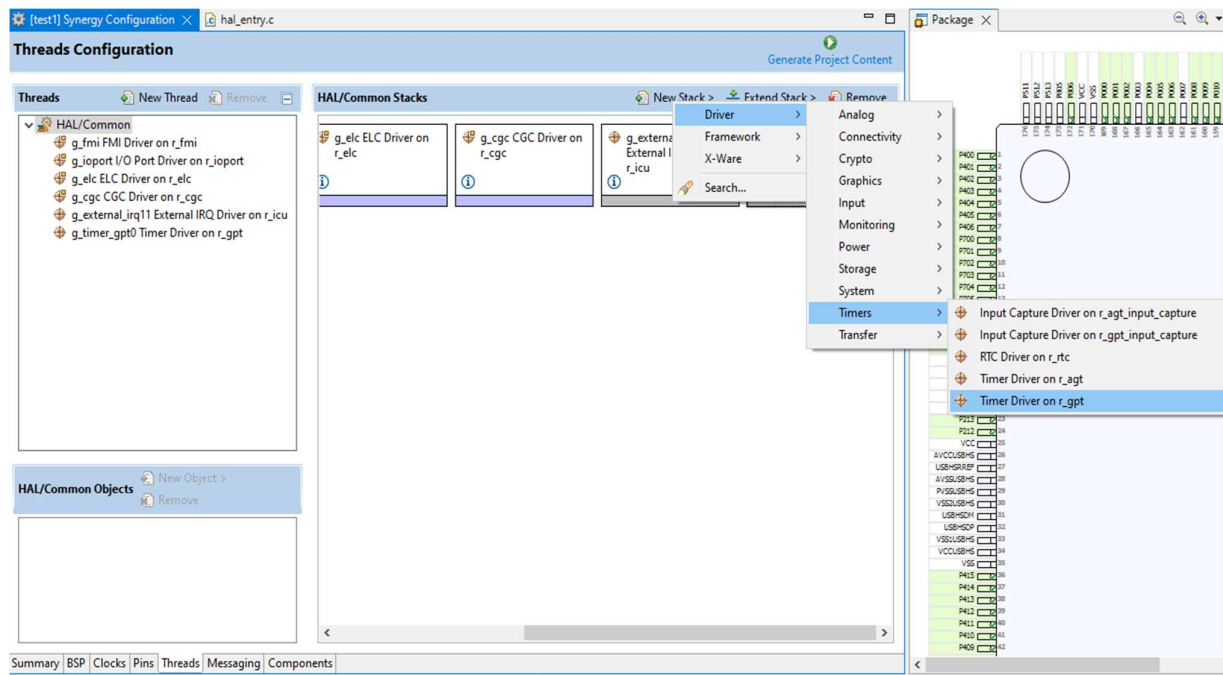
Βασικό στοιχείο ενός timer/counter είναι ο ψηφιακός μετρητής ο οποίος είτε θα μετράει γεγονότα (counter), είτε παλμούς του ρολογιού αν έχει τεθεί σε λειτουργία μέτρησης χρονικής διάρκειας (timer). Σημειώνεται πως η μέγιστη χρονική διάρκεια που μπορεί να μετρηθεί εξαρτάται από τη περίοδο του ρολογιού και τον αριθμό των bit που διαθέτει ο μετρητής.

Στο μικροελεγκτή S7G2 υπεύθυνη για όλες τις παραπάνω λειτουργίες είναι η μονάδα GPT (General PWM Timer), η οποία διαθέτει 14 κανάλια με το καθένα να έχει έναν μετρητή των 32-bit. Η μονάδα λειτουργεί με το ρολόι PCLKD (120 MHz).

3-2: Ενεργοποίηση και Ρύθμιση της μονάδας GPT.

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

- Επιλέξτε στην καρτέλα “**Threads**” από την μπάρα της καρτέλας “**Synergy Configuration**”.
- Για την εισαγωγή του κατάλληλου stack επιλέξτε “**New Stack**”, “**Driver**”, “**Timers**”, “**Timers driver on r_gpt**” 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 ICU.
- Στην καρτέλα “Synergy Configuration” 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 **Name** αφορά το όνομα του module που χρησιμοποιούμε.
- 2) Το πεδίο **Channel** προσδιορίζει το κανάλι που θα χρησιμοποιηθεί (0-13).
- 3) Το πεδίο **Mode** αφορά την επιλογή του τρόπου λειτουργίας της μονάδας.
 - **Periodic**: Περιοδική λειτουργία. Όταν ολοκληρωθεί ο προσδιορισμένος χρόνος, γίνεται επανεκκίνηση του timer. Η callback καλείται κάθε φορά που τερματίζει ο timer.
 - **One-shot**: Ο timer μετράει μια φορά.
 - **PWM**: Παράγονται σήματα PWM.
- 4) Τα πεδία **Period Value** και **Period Unit** αφορούν το μέτρο και τη μονάδα μέτρησης της περιόδου του timer, δηλαδή το χρόνο στον οποίο θα λήγει.

- 5) Τα πεδία **Duty Cycle Value** και **Duty Cycle Unit** αφορούν στις ρυθμίσεις του duty cycle για σήματα PWM.
- 6) Το πεδίο **Auto-Start** ρυθμίζει την έναρξη του timer. Αν επιλεγθεί η τιμή False τότε ο timer θα αρχίσει όταν γίνει κλήση της συνάρτησης “start”.
- 7) Τα πεδία **GTIOCA/B Output Enabled** και **GTIOCA/B Stop Level** αφορούν την έξοδο και το επίπεδο αυτής, στους ακροδέκτες που συνδέονται με τη μονάδα GPT.
- 8) Το πεδίο **Callback** αφορά το όνομα της συνάρτησης, που έχει ορίσει ο χρήστης και θα εκτελεστεί με τη λήξη της χρονικής περιόδου.
- 9) Το πεδίο **Overflow Interrupt Priority** ορίζει την προτεραιότητα για το interrupt.

Ακολουθεί μια λίστα με συναρτήσεις για τη χρήση της μονάδας timer/counter:

Όνομα	Κλήση και περιγραφή λειτουργίας
.open	g_timer0.p_api->open(g_timer0.p_ctrl,g_timer0.p_cfg); Αρχικοποίηση του timer σύμφωνα με τις ρυθμίσεις που έχουν οριστεί.
.start	g_timer0.p_api->start(g_timer0.p_ctrl); Ξεκινάει τη μέτρηση του timer.
.stop	g_timer0.p_api->stop(g_timer0.p_ctrl); Σταματάει τη μέτρηση του timer.
.reset	g_timer0.p_api->reset(g_timer0.p_ctrl); Επαναφέρει τον timer στην αρχική του τιμή.
.counterGet	g_timer0.p_api->counterGet(g_timer0.p_ctrl, &value); Αποθηκεύει την τιμή του timer στην μεταβλητή value.
.periodSet	g_timer0.p_api->periodSet(g_timer0.p_ctrl, period, unit); Αλλάζει την περίοδο του timer στη δοθείσα.

3-3: Εργαστηριακή Άσκηση

Αναπτύξτε εφαρμογή που θα χρησιμοποιεί 2 κανάλια της μονάδας GPT για την ενεργοποίηση 2 LED της πλακέτας. Ορίστε 2 **διαφορετικά** stacks, ένα για το κάθε κανάλι. Το πρώτο κανάλι θα είναι το 0, θα έχει όνομα 'g_timer0' και θα είναι **one-shot** με περίοδο 1 δευτερόλεπτο. Όταν λήγει, θα εκτελεί τη συνάρτηση 'timer0_callback()' η οποία θα ανάβει ένα από τα LED.

Το δεύτερο κανάλι θα είναι το 1, θα έχει όνομα 'g_timer1' και θα είναι **periodic** με περίοδο 3 δευτερόλεπτα. Κάθε φορά που θα λήγει ο timer1, θα εκτελεί τη συνάρτηση 'timer1_callback()' και θα ανάβει ή θα σβήνει κάποιο άλλο LED, ανάλογα με την προηγούμενη του κατάσταση.

Οι ρυθμίσεις των timer παρατίθενται αναλυτικά στις παρακάτω φωτογραφίες.

Ρυθμίσεις για το κανάλι 0:

g_timer0 Timer Driver on r_gpt		
Settings	Property	Value
API Info	Common	
	Parameter Checking	Default (BSP)
	Module g_timer0 Timer Driver on r_gpt	
	Name	g_timer0
	Channel	0
	Mode	One Shot
	Duty Cycle Range (only applicable in PWM mode)	Shortest: 2 PCLK, Longest: (Period - 1) PCLK
	Period Value	1
	Period Unit	Seconds
	Duty Cycle Value	50
	Duty Cycle Unit	Unit Raw Counts
	Auto Start	True
	GTIOCA Output Enabled	False
	GTIOCA Stop Level	Pin Level Low
	GTIOCB Output Enabled	False
	GTIOCB Stop Level	Pin Level Low
	Callback	timer0_callback
	Overflow Interrupt Priority	Priority 4

Ρυθμίσεις για το κανάλι 1 :

g_timer1 Timer Driver on r_gpt		
Settings	Property	Value
API Info	Common	
	Parameter Checking	Default (BSP)
	Module g_timer1 Timer Driver on r_gpt	
	Name	g_timer1
	Channel	1
	Mode	Periodic
	Duty Cycle Range (only applicable in PWM mode)	Shortest: 2 PCLK, Longest: (Period - 1) PCLK
	Period Value	3
	Period Unit	Seconds
	Duty Cycle Value	50
	Duty Cycle Unit	Unit Raw Counts
	Auto Start	True
	GTIOCA Output Enabled	False
	GTIOCA Stop Level	Pin Level Low
	GTIOCB Output Enabled	False
	GTIOCB Stop Level	Pin Level Low
	Callback	timer1_callback
	Overflow Interrupt Priority	Priority 4

Κατασκευάστε τις συναρτήσεις timer_callback ως εξής :

```
void timer0_callback (timer_callback_args_t * p_args0)
{
    //your code
}
```

```
void timer1_callback (timer_callback_args_t * p_args1)
{
    //your code
}
```

Ενότητα 4: Παιχνίδι χρόνου απόκρισης.

4-1: Εργαστηριακή Άσκηση

Συνδυάστε τις δύο παραπάνω μονάδες, των interrupts και των timers ώστε να αναπτύξετε εφαρμογή που θα μετράει το χρόνο απόκρισης του χρήστη σε ένα οπτικό ερέθισμα.

Το οπτικό ερέθισμα στην προκειμένη περίπτωση θα είναι μια λυχνία LED, η οποία θα ανάβει και ο χρήστης θα πρέπει να πατήσει όσο πιο γρήγορα γίνεται ένα διακόπτη, ώστε να μετρηθεί ο χρόνος απόκρισης του.

4-2: Οδηγίες Υλοποίησης

Η εφαρμογή θα πρέπει να υλοποιεί τα παρακάτω βήματα :

1. Σβήσιμο των LED.
2. Αναμονή 2 δευτερολέπτων.
3. Ενεργοποίηση ενός από τα LED, που θα λειτουργήσει ως το οπτικό ερέθισμα.
4. Αναμονή μέχρι ο χρήστης να πατήσει το διακόπτη.
5. Με το πάτημα του διακόπτη, θα πρέπει να σταματάει ο timer και να καταγράφεται η τιμή στην οποία σταμάτησε.
6. Υπολογισμός χρόνου απόκρισης
7. Επιστροφή στο πρώτο βήμα.

Υποδείξεις για την άσκηση:

1. Δημιουργείτε καινούργιο project, προσθέστε τις απαραίτητες μονάδες και επιλέξτε τις κατάλληλες ρυθμίσεις από το γραφικό περιβάλλον.
2. Ο timer που θα μετράει το χρόνο απόκρισης, δεν θα πρέπει να ξεπερνάει τα 3 δευτερόλεπτα. Ακόμη δεν θα πρέπει να ξεκινάει αυτόματα, αλλά με κλήση της συνάρτησης 'start'.
3. Χρησιμοποιείτε τη συνάρτηση counterGet() για να πάρετε την τιμή του timer.
4. Για τον υπολογισμό του χρόνου απόκρισης θυμηθείτε πως η μονάδα GPT μετράει παλμούς του ρολογιού PCLKD (120 MHz).
5. Χρησιμοποιείτε μεταβλητές flag στις συναρτήσεις callback που θα ειδοποιούν το κυρίως πρόγραμμα για γεγονότα (interrupt, λήξη timer, κλπ).
6. Στο παράθυρο debug, ορίστε ένα breakpoint μετά τον υπολογισμό του χρόνου απόκρισης, ώστε να εξετάσετε τη τιμή του.