



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

Μικροεπεξεργαστές - DSPs

Εργαστηριακή Άσκηση 2:
Input/Output

Εαρινό Εξάμηνο
2022 – 2023

Ενότητα 1: Εισαγωγή

1-1 Σκοπός της άσκησης

Σκοπός της άσκησης είναι η εξοικείωση με το πως χειρίζεται την είσοδο και έξοδο ένας μικροελεγκτής, καθώς και το πως τα I/O Ports μπορούν να χρησιμοποιηθούν σε εφαρμογές.

1-2 Board setup και Δημιουργία Project.

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

Στη συνέχεια εκτελέστε το **e² studio** και επιλέξτε τις ίδιες ρυθμίσεις με την προηγούμενη άσκηση για να δημιουργήσετε ένα καινούργιο project.

(**“Create a new C/C++ project”, “Renesas Synergy C Executable Project”, “S7G2 SK”, BSP**).

Ενότητα 2 : I/O Ports

2-1: Εισαγωγικές έννοιες

Οι μικροελεγκτές διαθέτουν ακροδέκτες για να διαχειρίζονται τα εισερχόμενα και εξερχόμενα σήματα. Οι ακροδέκτες αυτοί μπορεί να είναι γενικής χρήσης ή να συνδέονται με συγκεκριμένες περιφερειακές συσκευές του μικροελεγκτή. Οι ακροδέκτες ομαδοποιούνται σε I/O Ports και η διαχείριση τους συνήθως γίνεται μέσω αυτών.

Ο μικροελεγκτής που χρησιμοποιείται στο εργαστήριο αποτελείται από 12 τέτοια ports (port0-portB) όπου το καθένα περιέχει μέχρι και 16 ακροδέκτες. Στην άσκηση αυτή θα χρησιμοποιήσετε τα I/O Ports για να διαβάσετε σήματα εξωτερικά του μικροελεγκτή και να τροφοδοτήσετε λυχνίες LED.

2-2 : Ενεργοποίηση και Ρύθμιση των I/O Ports στο γραφικό περιβάλλον.

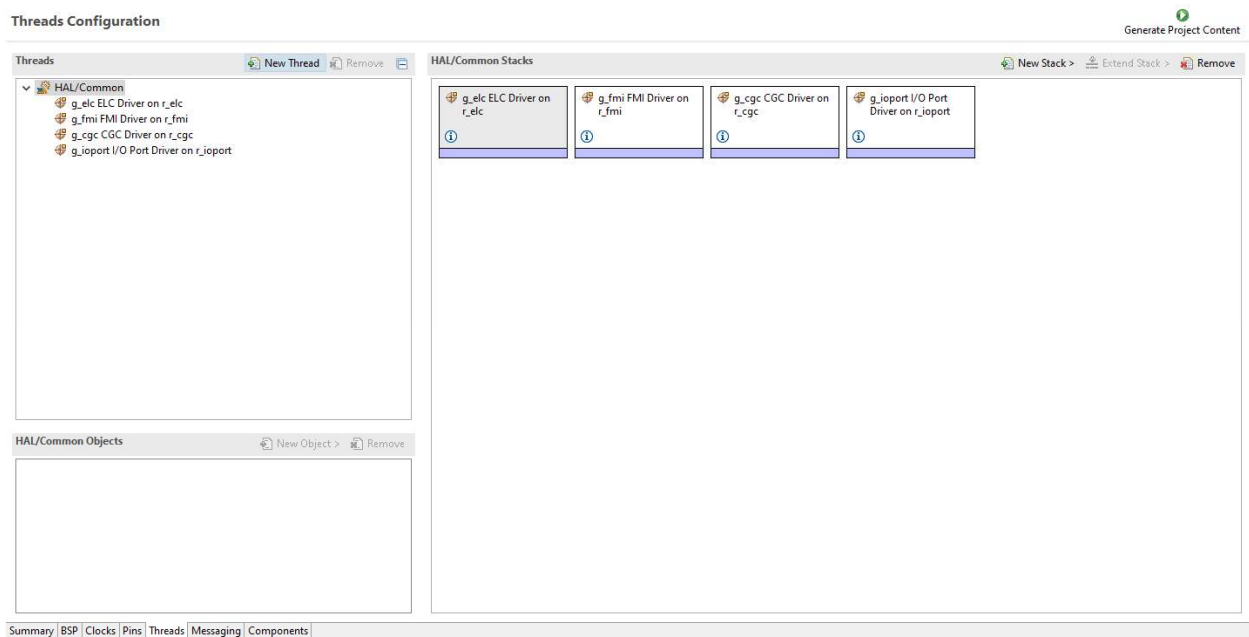
Στη καρτέλα Synergy Configuration->Pins του e² studio μπορείτε να δείτε πληροφορίες για τους ακροδέκτες και να ρυθμίσετε τη λειτουργία σε συγκεκριμένους από αυτούς.

The screenshot displays the 'Pins Configuration' window in the Synergy IDE. The left pane shows a tree view of pins, with P100 selected under port P1. The right pane shows the configuration for P100, including its symbolic name (SPI0_Pin_Option_A_MISOA_A), port capabilities (AGTIO, BUS0, GLCD, IIC, IRQ, KINT, PORG, SCI, SCK, SPI), and P100 configuration (Mode: Peripheral mode, Pull up: None, IRQ: None, Drive Capacity: Medium, Output type: CMOS). The chip input/output is set to SPI0_MISO.

Επιλέγοντας το “Generate Project Content”, παράγονται τα απαραίτητα configuration αρχεία και η πλακέτα ρυθμίζεται σύμφωνα με τις επιλογές που έχετε κάνει.

Για να χρησιμοποιήσετε τα I/O Ports σε μια εφαρμογή θα πρέπει να προσθέσετε στο project τους drivers για αυτά. Οι drivers για τα I/O Ports είναι προεγκατεστημένοι σε κάθε project. Στη περίπτωση που δεν συμβαίνει αυτό μπορείτε να τους προσθέσετε ως εξής:

Στη καρτέλα Synergy Configuration επιλέξτε την επιλογή Threads και μετά επιλέξτε HAL/Common Stack. Στη συνέχεια επιλέξτε τις επιλογές New Stack > Driver > System > I/O Port Driver on r_ioport.



2-3: I/O Port HAL

Η χρήση των I/O Ports σε κάποια εφαρμογή, μπορεί να γίνει μέσω του I/O Port HAL (hardware abstraction layer) module. Το I/O Port HAL ορίζει συναρτήσεις για τη ρύθμιση και χρήση των ακροδεκτών. Η κλήση των συναρτήσεων γίνεται μέσω του api. Μερικές από τις συναρτήσεις που ορίζονται στο I/O Port HAL module είναι οι ακόλουθες:

Όνομα	Κλήση και περιγραφή λειτουργίας
.init	<pre>g_ioport.p_api->init(g_ioport.p_cfg);</pre> <p>Αρχικοποίηση των ακροδεκτών σύμφωνα με τις ρυθμίσεις και τα χαρακτηριστικά που έχουν επιλεγθεί στο γραφικό περιβάλλον.</p>
.pinCfg	<pre>g_ioport.p_api->pinCfg(IOPORT_PORT_00_PIN_00, IOPORT_CFG_IRQ_ENABLE IOPORT_CFG_PORT_DIRECTION_INPUT);</pre> <p>Ρύθμιση λειτουργίας ενός ακροδέκτη.</p>
.portDirectionSet	<pre>g_ioport.p_api->portDirectionSet(IOPORT_PORT_00, direction_values, mask);</pre> <p>Ρύθμιση λειτουργίας εισόδου/εξόδου σε πολλούς ακροδέκτες ενός port μέσω χρήση μάσκας.</p>
.pinDirectionSet	<pre>g_ioport.p_api->pinDirectionSet(IOPORT_PORT_00_ PIN_00, IOPORT_DIRECTION_INPUT);</pre> <p>Ρύθμιση λειτουργίας εισόδου/εξόδου ενός ακροδέκτη.</p>

.portRead	<pre>g_ioport.p_api->portRead(IOPORT_PORT_00, &pin_levels);</pre> <p>“Διαβάζει” τη κατάσταση των ακροδεκτών σε ένα συγκεκριμένο port και αποθηκεύει τη τιμή στη μεταβλητή pin_levels.</p>
.pinRead	<pre>g_ioport.p_api->pinRead(IOPORT_PORT_00_PIN_0 0, &pin_level);</pre> <p>“Διαβάζει” την κατάσταση ενός ακροδέκτη και την αποθηκεύει στη μεταβλητή pin_level.</p>
.portWrite	<pre>g_ioport.p_api->portWrite(IOPORT_PORT_00, pin_levels, mask);</pre> <p>Αλλάζει την κατάσταση πολλαπλών ακροδεκτών σε ένα συγκεκριμένο port, με χρήση μάσκας.</p>
.pinWrite	<pre>g_ioport.p_api->pinWrite(IOPORT_PORT_00_PIN_0 0, IOPORT_PIN_LEVEL_HIGH);</pre> <p>Αλλάζει τη κατάσταση ενός ακροδέκτη.</p>

2-4: Εργαστηριακή Άσκηση

Αναπτύξτε εφαρμογή, στην οποία θα χρησιμοποιήσετε τα I/O Ports. Στην εφαρμογή θα χρησιμοποιήσετε τις λυχνίες LED και τους διακόπτες S4,S5 της πλακέτας. Η εφαρμογή θα πρέπει να “διαβάζει” την κατάσταση των διακοπών και ανάλογα με αυτή να τροφοδοτεί τα αντίστοιχα LED. Πιο συγκεκριμένα, η λειτουργία της εφαρμογής θα πρέπει να είναι η ακόλουθη:

- Αν ο διακόπτης S4 έχει πατηθεί τότε το Led1 θα ανάβει.
- Αν ο διακόπτης S5 έχει πατηθεί τότε το Led2 θα ανάβει.
- Αν και οι 2 διακόπτες S4,S5 έχουν πατηθεί τότε το Led3 θα ανάβει.

Δημιουργήστε ένα νέο project σύμφωνα με τις οδηγίες της 1ης άσκησης και αναπτύξτε την εφαρμογή σας στο αρχείο ‘hal_entry.c’. Η εφαρμογή σας αρχικά θα πρέπει να ορίζει τη λειτουργία (input/output) των ακροδεκτών που συνδέονται με τις συσκευές που θα χρησιμοποιήσετε στην άσκηση αυτή. Η αντιστοίχιση είναι η ακόλουθη:

- Led1: port6, pin0 (IOPORT_PORT_06_PIN_00)
- Led2: port6, pin1 (IOPORT_PORT_06_PIN_01)
- Led3: port6, pin2 (IOPORT_PORT_06_PIN_02)
- Button S4: port0, pin6 (IOPORT_PORT_00_PIN_06)
- Button S5: port5, pin5 (IOPORT_PORT_00_PIN_05)

Για να ορίσετε τη λειτουργία χρησιμοποιήστε τη συνάρτηση `pinDirectionSet()`. Στη συνέχεια δημιουργήστε μια `super-loop` η οποία θα διαβάζει το επίπεδο των διακοπών και θα ανάβει τις αντίστοιχες λυχνίες. Για να διαβάσετε το επίπεδο τάσης στους διακόπτες χρησιμοποιήστε τη συνάρτηση `pinRead()`. Για να ανάψετε ένα LED χρησιμοποιήστε τη συνάρτηση `pinWrite()`. Υπενθυμίζουμε πως ένας διακόπτης που έχει πατηθεί βρίσκεται στην κατάσταση ‘0’ και πως για να ανάψει μια λυχνία της πλακέτας πρέπει να βρίσκεται στη κατάσταση ‘0’.