# ΗΡΥ411- Ενσωματωμένα Συστήματα Μικροεπεξεργαστών

# Εργαστήριο 2

LAB41145851 15/10/2020

Εμμανουήλ Πετράκος ΑΜ 2014030009

Για το δεύτερο εργαστήριο έχει κατασκευαστεί το πρόγραμμα οδήγησης μιας 7-segment οθόνης οκτώ ψηφίων. Ως είσοδος του προγράμματος θεωρείται ένα κομμάτι μνήμης SRAM που περιέχει τα δεδομένα προς εμφάνιση. Η έξοδος του συστήματος είναι το σήμα 7-segment (A-G,DP) στο PORTA και το σήμα ενεργοποίησης των ψηφίων (AN7-AN0) στο PORTC.

## Επεξήγηση προσέγγισης

Το πρόγραμμα ξεκινάει με τις απαραίτητες αρχικοποιήσεις. Όπως και στο προηγούμενο εργαστήριο, αρχικοποιείται ο stack pointer για να μπορούν να χρησιμοποιηθούν interrupts και υπορουτίνες, καθώς και ο TIMERO ρυθμισμένος στα 2ms. Με αυτή την ρύθμιση, η συχνότητα ανανέωσης είναι  $1s/(2ms*8\psi\eta\varphii\alpha)\approx 62,5\,Hz/\psi\eta\varphiio$ , περίπου δύο φορές μεγαλύτερη από το ελάχιστο όριο. Οι θύρες A και C ορίζονται ως έξοδοι και αρχικοποιούνται με τέτοιο τρόπο ώστε να μην ενεργοποιηθεί ακόμα κάποιο LED και να ξεκινήσει η λειτουργία από το ANO. Το PORTC λειτουργεί σαν ένας μετρητής δακτυλίου, έχοντας ενεργοποιημένο μόνο ένα bit που αντιστοιχεί στο ψηφίο που εμφανίζεται.

Το πρόγραμμα χρησιμοποιεί 18 bytes στην SRAM, 8 για την είσοδο του και 10 για τις κωδικοποιήσεις 7\_segment. Χρησιμοποιώντας την ρουτίνα write\_data αποθηκεύεται η είσοδος, ξεκινώντας από την διεύθυνση 0x0060. Τα δεδομένα του ψηφίου 0 αποθηκεύονται στην πρώτη θέση, του ψηφίου 1 στην δεύτερη θέση κλπ. Με αυτό τον τρόπο, ο δείκτης μιας εξόδου μπορεί να χρησιμοποιηθεί ως offset πάνω στην αρχική διεύθυνση για να γίνει πρόσβαση στα αντίστοιχα δεδομένα. Η αποθήκευση των κωδικοποιήσεων γίνεται μέσω της ρουτίνας write\_7\_segments αμέσως μετά τα δεδομένα εισόδου, δηλαδή ξεκινώντας από την διεύθυνση 0x0068. Ακολουθείται παρόμοια λογική με πριν, η κωδικοποίηση του αριθμού 0 αποθηκεύεται στην πρώτη θέση, του αριθμού 1 στη δεύτερη κλπ. Σαν αποτέλεσμα, ο ίδιος ο αριθμός λειτουργεί ως offset κατά την πρόσβαση στην κωδικοποίηση του. Γενικά, η αποθήκευση των δεδομένων έχει γίνει με γνώμονα την ευκολότερη αναζήτηση και πρόσβαση τους.

Υπολογισμός δικτών, όπου x το ενεργοποιημένο ψηφίο (DATA.x) = 0x0060 + x (7 segment.y) = 0x0068 + DATA.x

Τέλος, ενεργοποιούνται τα interrupts και το πρόγραμμα μπαίνει σε ένα ατέρμον βρόχο.

Η κύρια λειτουργικότητα του προγράμματος υλοποιείται στην ρουτίνα εξυπηρέτησης του interrupt. Εεκινώντας, απενεργοποιούνται όλα τα LED για την αποφυγή εμφάνισης σκουπιδιών κατά την αλλαγή ψηφίου και δεδομένων. Κάνοντας τον μετρητή δακτυλίου στο PORTC κυκλικό shift προς τα αριστερά, απενεργοποιείται το τρέχων ψηφίο και ενεργοποιείται το επόμενο. Επίσης, η νέα τιμή του δίνει αρκετή πληροφορία για το πιο byte πρέπει να διαβαστεί από την μνήμη, καθώς μετατρέποντας τον αριθμό δακτυλίου σε δυαδική μορφή μπορεί να γίνει η πρόσβαση στα αντίστοιχα δεδομένα με τον τρόπο που αναφέρθηκε παραπάνω. Το επόμενο βήμα είναι η μετατροπή τους σε 7 segment μορφή. Γίνεται έλεγχος εγκυρότητας των δεδομένων, αν δεν είναι ένα από τα αποδεκτά ψηφία (0-9) απενεργοποιούνται όλα τα LED, αλλιώς διαβάζεται από την μνήμη η κατάλληλη κωδικοποίηση και κατευθύνεται στο PORTA.

#### Πειραματική Διαδικασία

Αποθηκεύοντας στην μνήμη τους αριθμούς 1 έως 8, η οθόνη πρέπει να δείξει 87654321. Επειδή δεν υπάρχει διαθέσιμο υλικό, η επαλήθευση της λειτουργικότητας του προγράμματος γίνεται μέσω του simulation του Atmel Studio 7. Παρακολουθώντας τα PORTA και PORTC στον simulator είναι δυνατόν να αποφανθούμε τι θα εμφάνιζε η οθόνη την συγκεκριμένη χρονική στιγμή. Αυτό γίνεται με ένα breakpoint στο τέλος της ρουτίνας εξυπηρέτησης του interrupt και τα παράθυρα I/O και Processor Status.

	Address			Address		Cycle Counter	118
I/O PINA	0x39	0xFF	 ₩ PINC	0x33	0x80	Frequency	10.000 MHz
1/0 DDRA	0x3A	0xFF	₩ DDRC	0x34	0xFF	Stop Watch	0.01 ms
I/O PORTA	0x3B	0xFF	 <b>₩</b> PORTC	0x35	0x80	Stop waten	0,01 ms

Εκκίνηση προγράμματος, PORTA = 0b11111111 (τίποτα)

	Address			Address		Cycle Counter	20029
<mark>⊮o</mark> PINA	0x39	0x9F	⊮o PINC	0x33	0x01	 Frequency	10,000 MHz
1/O DDRA	0x3A	0xFF	<mark>⊮⊙</mark> DDRC	0x34	0xFF	Stop Watch	2,00 ms
VO PORTA	0x3B	0x9F	1/0 PORTC	0x35	0x01	Stop Water	2,001115

10 interrupt: PORTA = 0b10011111 (1), PORTC = AN0

	Address			Address		 Cycle Counter	40001
I/O PINA	0x39	0x25	PINC	0x33	0x02	 Frequency	10,000 MHz
						Stop Watch	4.00 ms
I/O PORTA	0x3B	0x25	PORTC	0x35	0x02	Stop Water	4,001115

20 interrupt: PORTA = 0b00100101 (2), PORTC = AN1

	Address			Address		 Cycle Counter	79945
<mark>⊮⊙</mark> PINA	0x39	0x99	₩ PINC	0x33	0x08	Frequency	10,000 MHz
1/O DDRA	0x3A	0xFF	 1/0 DDRC	0x34	0xFF	 Stop Watch	7.99 ms
✓ PORTA	0x3B	0x99	VO PORTC	0x35	0x08	Stop Water	1,55 1115

4o interrupt: PORTA = 0b10011001 (4), PORTC = AN3

	Address			Address		 Cycle Counter	59973
I/O PINA	0x39	0x0D	I/O PINC	0x33	0x04	Frequency	10,000 MHz
₩ DDRA						 Stop Watch	6.00 ms
<b>₩</b> PORTA	0x3B	0x0D	I/O PORTC	0x35	0x04	Stop Water	0,001113

*3o interrupt: PORTA = 0b00001101 (3), PORTC = AN2* 

	Address			Address		Cycle Counter	99917
I/O PINA	0x39	0x49	I/O PINC	0x33	0x10	Frequency	10,000 MHz
<mark>⊮o</mark> DDRA	0x3A	0xFF	 <mark>⊮o</mark> DDRC	0x34	0xFF	 Stop Watch	9.99 ms
1/0 PORTA	0x3B	0x49	PORTC	0x35	0x10	Stop Water	3,33 1113

50 interrupt: PORTA = 0b01001001 (5), PORTC = AN4

	Address			Address		Cycle Counter	119889
I/O PINA	0x39	0x41	VO PINC	0x33	0x20	Frequency	10,000 MHz
₩ DDRA	0x3A	0xFF	<mark>⊮○</mark> DDRC	0x34	0xFF	 Stop Watch	11,99 ms
1/0 PORTA	0x3B	0x41	<mark>⊮⊙</mark> PORTC	0x35	0x20	Stop Water	11,55 1115

60 interrupt: PORTA = 0b01000001 (6), PORTC = AN5

	Address		Name			Cycle Counter	139861
PINA DDRA	0x39 0x3A	0x1F 0xFF	I/O PINC	0x33 0x34	0xC0 0xFF	Frequency Stop Watch	10,000 MHz 13,99 ms
PORIA	0x3B	0x1F	PORIC	0x35	0x40		·

*7o interrupt: PORTA = 0b00011111 (7), PORTC = AN6* 

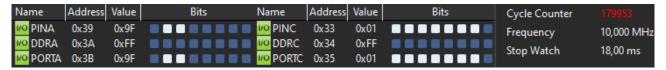
		Address			Address		Cycle Counter	159833
	<mark>⊮o</mark> PINA	0x39	0x01	₩ PINC	0x33	0x80	Frequency	10,000 MHz
ı	<u>™</u> DDRA	0x3A	0xFF	 MO DDRC	0x34	0xFF	Stop Watch	15.98 ms
ı	I/O PORTA	0x3B	0x01	1/0 PORTC	0x35	0x80	Stop materi	.5,505

80 interrupt: PORTA = 0b00000001 (8), PORTC = AN7

	Address			Address		,	179773
I/O PINA	0x39	0x9F	 WO DDRC	0x33	0x01		10,000 MHz
						Stop Watch	17,98 ms

90 interrupt: PORTA = 0b10011111 (1), PORTC = AN0

Πέρα από την σωστή αλλαγή των εξόδων, πρέπει να παραμένουν σταθερές όσο το πρόγραμμα είναι στον ατέρμον βρόχο και να μην εμφανίζονται σκουπίδια όσο εξυπηρετείται το interrupt.



Ατέρμον βρόχος: PORTA = 0b10011111 (1), PORTC = AN0

Name	Address	Value	Bits	Name	Address	Value	Bits	Cycle Counter	199719
⊮o PINA	0x39	0xFF		I/O PINC	0x33	0x02		Frequency	10,000 MHz
⊮o DDRA	0x3A	0xFF		⊮o DDRC	0x34	0xFF		Stop Watch	19.97 ms
⊮o PORTA	0x3B	0xFF		VO PORTC	0x35	0x02		Stop Water	15,57 1113

Μέσα στο επόμενο Interrput: PORTA = 0b11111111 (τίποτα)

Τέλος, εμφανίζεται η έξοδος όταν η είσοδος έχει μη αποδεκτά δεδομένα.

N	lame	Address	Value	Bits	Name	Address	Value	Bits	Cycle Counter	199739
L	PINA	0x39	0xFF		<mark>⊮○</mark> PINC	0x33	0x02		Frequency	10,000 MHz
_	=								Stop Watch	19.97 ms
L	PORTA	0x3B	0xFF		1/0 PORTC	0x35	0x02		Stop Water	15,51 1115

Θέτοντας στην θέση 2 την τιμή 10: PORTA = 0b11111111 (τίποτα)

### Ανάλυση & Παρατηρήσεις

Μεταξύ δύο διαδοχικών αλλαγών παρέρχονται ~20000 κύκλοι ρολογιού, όπου ~60 από αυτούς χρειάζονται για την εξυπηρέτηση του interrupt. Δηλαδή, ο driver απαιτεί ~0.3% του υπολογιστικού χρόνου του επεξεργαστή.

Για το παρόν εργαστήριο τα δεδομένα προς εμφάνιση ορίζονται στην ρουτίνα write\_data κατά την εκκίνηση του προγράμματος και μένουν σταθερά, κάτι που πρέπει να αλλάξει στα επόμενα εργαστήρια για να είναι ο driver χρήσιμος. Πέρα από αυτό, το πρόγραμμα είναι ολοκληρωμένο.

#### Πηγές

7 Segment Encoding www.elprocus.com/bcd-to-seven-segment-display-decoder-theory/

Using SRAM in AVR www.avr-asm-tutorial.net/avr\_en/beginner/SRAM.html

AVR Instruction Set Manual <a href="http://ww1.microchip.com/downloads/en/devicedoc/atmel-0856-avr-instruction-set-manual.pdf">http://ww1.microchip.com/downloads/en/devicedoc/atmel-0856-avr-instruction-set-manual.pdf</a>