



ΑΘΗΝΑ, 12 Νοεμβρίου 2022

4<sup>η</sup> ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ  
ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Εργαστήριο Μικροϋπολογιστών"  
Χρήση ADC και Οθόνης 2×16 Χαρακτήρων στον AVR

Εξέταση – Επίδειξη: Τετάρτη 23/11/2022  
Προθεσμία για παράδοση Έκθεσης: Κυριακή 27/11/2022 (23:59)

Μετατροπέας αναλογικής σε ψηφιακή μορφή **ADC**.

Βασικό περιφερειακό των μικροελεγκτών είναι ο μετατροπέας από Αναλογική σε Ψηφιακή μορφή (ADC). Αν  $V_{in}$  είναι η αναλογική τάση εισόδου του ADC με ανάλυση  $n$ -bit,  $V_{REF}$  μία τάση αναφοράς βάσει της οποίας γίνεται η μετατροπή και  $V_{in} \leq V_{REF}$  τότε η ψηφιακή έξοδος του ADC μετατροπέα είναι:

$$ADC = \frac{V_{in} \cdot 2^n}{V_{REF}}$$

Στην παραπάνω σχέση όλες η ποσότητες είναι ακέραιοι θετικοί αριθμοί.

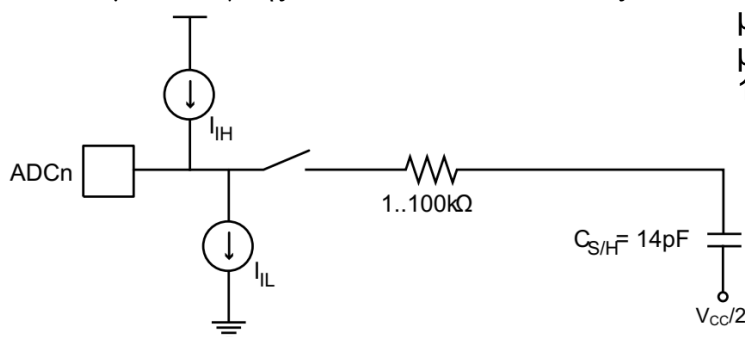
$$V_{ref} = V_{max} \quad \mu \quad 0 \dots V_{ref} \quad "$$

Αναλογικό κύκλωμα εισόδου

$$n = 10$$

$$\mu \quad \mu \quad 1024 \quad \mu \quad V_{ref}/1024.$$

Το κύκλωμα αναλογικής εισόδου του ADC απεικονίζεται στο παρακάτω σχήμα:



$$\mu \quad \mu \quad 1024/V_{ref} = ADC, \quad ADC \quad V_{in}/(V_{ref}/1024) = V_{in}^* \quad \mu \quad V_{in}.$$

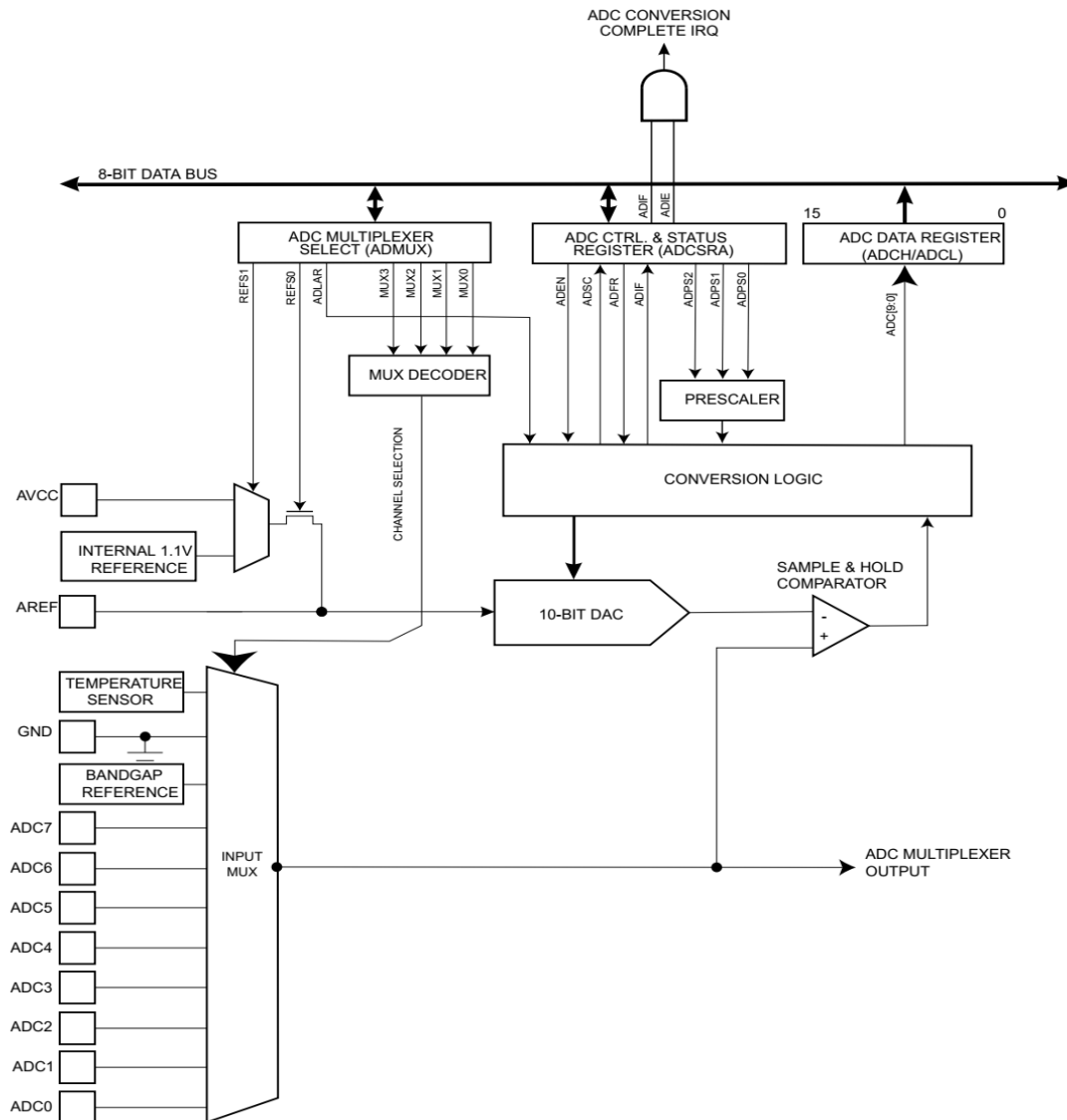
Σχήμα 4.1: Κύκλωμα αναλογικής εισόδου στον ADC

Μια αναλογική πηγή που εφαρμόζεται στον ακροδέκτη  $ADCn$  επηρεάζεται από τη χωρητικότητα του ακροδέκτη αυτού και από το ρεύμα διαρροής του. Όταν επιλεγεί το κανάλι αυτό για μετατροπή τότε πρέπει το σήμα του να φορτίσει τον πυκνωτή S/H δια μέσου της αντίστασης σειράς (Ολική αντίσταση στη διαδρομή εισόδου).

Το ADC είναι βελτιστοποιημένο για αναλογικά σήματα εισόδου με σύνθετη αντίσταση μέχρι 10 KΩ, όπου ο χρόνος δειγματοληψίας είναι αμελητέος. Εάν χρησιμοποιηθεί πηγή με υψηλότερη τιμή αντίστασης τότε ο χρόνος δειγματοληψίας αυξάνεται και εξαρτάται από το χρόνο που απαιτείται για να φορτιστεί ο πυκνωτής S/H.

## AVR ATmega328PB ADC

Ο μικροελεγκτής AVR ATmega328PB διαθέτει έναν ADC, ανάλυσης **n=10 bit**, που βασίζεται στη μέθοδο των διαδοχικών προσεγγίσεων. Στο παρακάτω σχήμα φαίνεται το μπλοκ διάγραμμά του ADC μετατροπέα:



Σχήμα 4.2: Μπλοκ διάγραμμα του ADC

### Αναλογικός πολυπλέκτης του ADC

Ο ADC είναι συνδεδεμένος με ένα αναλογικό πολυπλέκτη 16 εισόδων. Το κανάλι αναλογικής εισόδου που είναι κάθε φορά συνδεδεμένο στο κύκλωμα μετατροπής του ADC επιλέγεται από τα bit MUX[3:0] του καταχωρητή ADMUX, όπως φαίνεται παρακάτω:

### ADMUX (Offset 0x7C). Multiplexer Selection Register

| Bit    | 7     | 6     | 5     | 4 | 3    | 2    | 1    | 0    |
|--------|-------|-------|-------|---|------|------|------|------|
|        | REFSn | REFSn | ADLAR |   | MUXn | MUXn | MUXn | MUXn |
| Access | R/W   | R/W   | R/W   |   | R/W  | R/W  | R/W  | R/W  |
| Reset  | 0     | 0     | 0     |   | 0    | 0    | 0    | 0    |

Σχήμα 4.3: Καταχωρητής ADMUX

| MUX[3:0] | ΚΑΝΑΛΙ ΕΙΣΟΔΟΥ     |
|----------|--------------------|
| 0000     | ADC0               |
| 0001     | ADC1               |
| 0010     | ADC2               |
| 0011     | ADC3               |
| 0100     | ADC4               |
| 0101     | ADC5               |
| 0110     | ADC6               |
| 0111     | ADC7               |
| 1000     | Temperature sensor |
| 1001     | Reserved           |
| 1010     | Reserved           |
| 1011     | Reserved           |
| 1100     | Reserved           |
| 1101     | Reserved           |
| 1110     | 1.1V (VBG)         |
| 1111     | 0V (GND)           |

Πίνακας 4.1: Επιλογή αναλογικών εισόδων του ADC

Στον παρακάτω πίνακα εμφανίζονται οι ακροδέκτες του μικροελεγκτή AVR ATmega328PB που μπορούν να δεχτούν αναλογικά σήματα και τα κανάλια του ADC που αντιστοιχούν. Όσοι από αυτούς τους ακροδέκτες θα χρησιμοποιηθούν από τον ADC πρέπει να οριστούν ως είσοδοι, κάνοντας χρήση του αντίστοιχου καταχωρητή DDRx.

| ΑΚΡΟΔΕΚΤΗΣ | ΚΑΝΑΛΙ ΤΟΥ ADC |
|------------|----------------|
| PC0        | ADC0           |
| PC1        | ADC1           |
| PC2        | ADC2           |
| PC3        | ADC3           |
| PC4        | ADC4           |
| PC5        | ADC5           |
| PE2        | ADC6           |
| PE3        | ADC7           |

Πίνακας 4.2: Αναλογικές είσοδοι του ADC

Τα bit REFS[1:0] χρησιμοποιούνται για την επιλογή του  $V_{REF}$  σύμφωνα με τον επόμενο πίνακα:

| REFS[1:0] | Voltage reference selection   |
|-----------|---|
| 00        | AREF, internal $V_{REF}$ turned OFF                                 |
| 01        | <b>AVCC</b> with external capacitor at AREF pin (5V)                |
| 10        | Reserved  |
| 11        | Internal 1.1V voltage reference with external capacitor at AREF pin |

AVCC pin and VCC pin should be connected

**Πίνακας 4.3:** Επιλογή του ADC  $V_{REF}$

### Έλεγχος της λειτουργίας του ADC

Ο έλεγχος της λειτουργίας του ADC γίνεται δια μέσω των καταχωρητών ADCSRA και ADCSRB.

Ο καταχωρητής ADCSRA φαίνεται στο παρακάτω σχήμα:

#### **ADCSRA (Offset 0x7A). ADC Control and Status Register A**

| Bit    | 7    | 6    | 5     | 4    | 3    | 2     | 1     | 0     |
|--------|------|------|-------|------|------|-------|-------|-------|
|        | ADEN | ADSC | ADATE | ADIF | ADIE | ADPSn | ADPSn | ADPSn |
| Access | R/W  | R/W  | R/W   | R/W  | R/W  | R/W   | R/W   | R/W   |
| Reset  | 0    | 0    | 0     | 0    | 0    | 0     | 0     | 0     |

**Σχήμα 4.4:** Καταχωρητής ADCSRA

Στη συνέχεια παρατίθενται οι λειτουργίες αυτού του καταχωρητή:

#### **ADEN: ADC Enable**

Όταν η σημαία αυτή τεθεί τότε ο ADC ενεργοποιείτε.

#### **ADSC: ADC Start Conversion**

Στη λειτουργία μεμονωμένης μετατροπής (Single Conversion mode), η σημαία αυτή πρέπει να τεθεί για να ξεκινήσει μία μετατροπή. Στη λειτουργία Free Running, η σημαία αυτή πρέπει να τεθεί για να ξεκινήσει η πρώτη μετατροπή. Η πρώτη μετατροπή διαρκεί 25 κύκλους ρολογιού του ADC ενώ οι επόμενες μετατροπές διαρκούν 13 κύκλους ρολογιού. Η πρώτη μετατροπή εκτελεί την αρχικοποίηση του ADC.

#### **ADATE: ADC Auto Trigger Enable**

Όταν η σημαία αυτή τεθεί, ενεργοποιείται η λειτουργία Auto Triggering του ADC. Το ADC θα ξεκινήσει αυτόματα μια μετατροπή σε μια θετική ακμή του επιλεγμένου σήματος σκανδαλισμού. Η πηγή σκανδαλισμού επιλέγεται ρυθμίζοντας τα bit ADTS[2:0] στον καταχωρητή ADCSRB.

### ADIF: ADC Interrupt Flag

Η σημαία αυτή τίθεται αυτόματα όταν ολοκληρωθεί μια μετατροπή και ενημερωθούν οι καταχωρητές δεδομένων. Αν η τιμή της σημαίας ADIF καθώς και της σημαίας I στον SREG είναι 1 τότε θα προκληθεί διακοπή ολοκλήρωσης μετατροπής του ADC. Το ADIF μηδενίζεται από το υλικό κατά την εκτέλεση του αντίστοιχου διανύσματος χειρισμού διακοπών. Εναλλακτικά, το ADIF μηδενίζεται από τον κώδικα γράφοντας σε αυτό λογικό 1.

### ADIE: ADC Interrupt Enable

Αν η τιμή της σημαίας ADIF καθώς και της σημαίας I στον SREG είναι 1 τότε θα προκληθεί διακοπή ολοκλήρωσης μετατροπής του ADC.

### ADPS[2:0]: ADC Prescaler Select bits

Τα bit αυτά θέτουν ένα συντελεστή διαίρεσης μεταξύ του κεντρικού ρολογιού και της συχνότητας λειτουργίας του ADC όπως φαίνεται στον επόμενο πίνακα:

| ADPS[2:0] | Division Factor |
|-----------|-----------------|
| 000       | 2               |
| 001       | 2               |
| 010       | 4               |
| 011       | 8               |
| 100       | 16              |
| 101       | 32              |
| 110       | 64              |
| 111       | 128             |

Πίνακας 4.4: ADC Prescaler Select bits

Το κύκλωμα διαδοχικής προσέγγισης του ADC, για μέγιστη ανάλυση 10 Bits, απαιτεί συχνότητα ρολογιού εισόδου μεταξύ 50 KHz και 200 KHz.

Ο καταχωρητής ADCSRB φαίνεται στο παρακάτω σχήμα:

### ADCSRB (Offset 0x7B). ADC Control and Status Register B

| Bit    | 7 | 6    | 5 | 4 | 3 | 2     | 1     | 0     |
|--------|---|------|---|---|---|-------|-------|-------|
|        |   | ACME |   |   |   | ADTSn | ADTSn | ADTSn |
| Access |   | R/W  |   |   |   | R/W   | R/W   | R/W   |
| Reset  |   | 0    |   |   |   | 0     | 0     | 0     |

Σχήμα 4.5: Καταχωρητής ADCSRB

Στη συνέχεια παρατίθενται οι λειτουργίες αυτού του καταχωρητή:

**ACME: Analog Comparator Multiplexer Enable**

Χρησιμοποιείται για την επιλογή της αναλογικής εισόδου στον analog comparator, ο οποίος είναι μια περιφερειακή συσκευή, ενσωματωμένη στον μικροελεγκτή ATmega328PB.

**ADTSn [2:0]: ADC Auto Trigger Source**

Όταν η σημαία ADATE του καταχωρητή ADCSRA είναι 1 τότε ενεργοποιείται η λειτουργία Auto Triggering του ADC. Το ADC θα ξεκινάει αυτόματα μια μετατροπή στη θετική ακμή του σήματος σκανδαλισμού το οποίο επιλέγεται σύμφωνα με τον παρακάτω πίνακα:

| ADTS[2:0] | Trigger Source                 |
|-----------|--------------------------------|
| 000       | Free Running mode              |
| 001       | Analog Comparator              |
| 010       | External Interrupt Request 0   |
| 011       | Timer/Counter0 Compare Match A |
| 100       | Timer/Counter0 Overflow        |
| 101       | Timer/Counter1 Compare Match B |
| 110       | Timer/Counter1 Overflow        |
| 111       | Timer/Counter1 Capture Event   |

**Πίνακας 4.5: ADC Auto Trigger Source**

**Καταχωρητές δεδομένων του ADC**

Το αποτέλεσμα του ADC έχει μήκος 10-bit και παρουσιάζεται στους καταχωρητές δεδομένων ADCH (high byte) και ADCL (low byte). Από προεπιλογή, το αποτέλεσμα παρουσιάζεται δεξιά προσαρμοσμένο(right adjusted), αλλά μπορεί προαιρετικά να τεθεί η σημαία ADLAR του καταχωρητή ADMUX και να παρουσιαστεί αριστερά προσαρμοσμένο (left adjusted).

Κατά την ανάγνωση του ADCL, οι καταχωρητές δεδομένων του ADC δεν ενημερώνονται μέχρι να διαβαστεί το ADCH. Εάν το αποτέλεσμα είναι αριστερά προσαρμοσμένο(Left adjusted) και δεν απαιτείται ακρίβεια μεγαλύτερη από 8 bit, αρκεί να διαβαστεί ο καταχωρητής ADCH. Σε διαφορετική περίπτωση, πρέπει πρώτα να διαβαστεί το ADCL και μετά το ADCH.

Οι καταχωρητές δεδομένων του ADC, ADCL και ADCH έχουν offset 0x78 και 0x79 αντίστοιχα. Παρουσιάζονται στο επόμενο σχήμα:

#### ADC Data Register Low and High Byte (ADLAR=0, Right adjusted)

Low?

| Bit    | 15 | 14 | 13 | 12 | 11 | 10 | 9    | 8    |
|--------|----|----|----|----|----|----|------|------|
|        |    |    |    |    |    |    | ADC9 | ADC8 |
| Access |    |    |    |    |    |    | R    | R    |
| Reset  |    |    |    |    |    |    | 0    | 0    |

| Bit    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|--------|------|------|------|------|------|------|------|------|
|        | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 | ADC1 | ADC0 |
| Access | R    | R    | R    | R    | R    | R    | R    | R    |
| Reset  | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

#### ADC Data Register Low and High Byte (ADLAR=1, Left adjusted)

| Bit    | 15   | 14   | 13   | 12   | 11   | 10   | 9    | 8    |
|--------|------|------|------|------|------|------|------|------|
|        | ADC9 | ADC8 | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 |
| Access | R    | R    | R    | R    | R    | R    | R    | R    |
| Reset  | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

| Bit    | 7    | 6    | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------|------|---|---|---|---|---|---|
|        | ADC1 | ADC0 |   |   |   |   |   |   |
| Access | R    | R    |   |   |   |   |   |   |
| Reset  | 0    | 0    |   |   |   |   |   |   |

Σχήμα 4.6: Καταχωρητές ADCL και ADCH

#### DIDR0 (Offset: 0x7E). Digital Input Disable Register 0

| Bit    | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
|        | ADC7D | ADC6D | ADC5D | ADC4D | ADC3D | ADC2D | ADC1D | ADC0D |
| Access | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |
| Reset  | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

Σχήμα 4.7: Καταχωρητής DIDR0

Όταν ένα bit αυτού του καταχωρητή τίθεται, τότε ο απομονωτής του αντίστοιχου ακροδέκτη του ADC απενεργοποιείται. Εάν ένα αναλογικό σήμα εφαρμόζεται σε έναν από του ακροδέκτες ADC[7:0] και η ψηφιακή είσοδος από αυτόν τον ακροδέκτη δεν χρειάζεται τότε το αντίστοιχο bit ADCxD θα πρέπει να τεθεί για να μειωθεί η κατανάλωση ενέργειας στον απομονωτή της ψηφιακής εισόδου.

#### Παραγωγή αναλογικών τάσεων στο ntuAboard\_G1

Η κάρτα ntuAboard\_G1 διαθέτει κυκλώματα για την παραγωγή αναλογικών τάσεων, οι οποίες μπορούν να συνδεθούν στις εισόδους 0 έως 3 του ADC μετατροπέα. Ποιο συγκεκριμένα η κάρτα ntuAboard\_G1 διαθέτει 4 ποτενσιόμετρα, 3 αναλογικά φίλτρα τα οποία παράγουν μεταβαλλόμενες DC τάσεις από τις PWM εξόδους των χρονιστών και ένα μετατροπέα ψηφιακού σήματος σε αναλογικό(DAC).

Τα κυκλώματα που αναφέρθηκαν απεικονίζονται στο παρακάτω σχήμα:



**Σχήμα 4.8:** Κυκλώματα για την παραγωγή αναλογικών τάσεων στην κάρτα ntuAboard\_G1.

Η σύνδεση καθεμιάς από τις πρώτες 4 αναλογικές εισόδους του ADC με τις αναλογικές τάσεις που παράγονται στην κάρτα ntuAboard\_G1 μπορεί να επιτευχθεί χρησιμοποιώντας βραχυκυκλωτήρες:

- Η αναλογική είσοδος 0 του ADC μπορεί να συνδεθεί είτε με το ποτενσιόμετρο POT1 είτε με το αναλογικό φίλτρο PD6\_PWM.
- Η αναλογική είσοδος 1 του ADC μπορεί να συνδεθεί είτε με το ποτενσιόμετρο POT2 είτε με το αναλογικό φίλτρο PB1\_PWM.
- Η αναλογική είσοδος 2 του ADC μπορεί να συνδεθεί είτε με το ποτενσιόμετρο POT3 είτε με το αναλογικό φίλτρο PD3\_PWM.
- Η αναλογική είσοδος 3 του ADC μπορεί να συνδεθεί είτε με το ποτενσιόμετρο POT4 είτε με την έξοδο του DAC.

Να σημειωθεί ότι για να μπορέσει να λειτουργήσει το αναλογικό φίλτρο στο PD6\_PWM πρέπει να συνδεθεί ένας βραχυκυκλωτήρας στον κονέκτορα J3. Παρομοίως το φίλτρο PB1\_PWM απαιτεί ένα βραχυκυκλωτήρα στον κονέκτορα J5 και το φίλτρο PD3\_PWM απαιτεί ένα βραχυκυκλωτήρα στον κονέκτορα J7.

Για παράδειγμα στο σχήμα 4.8 η αναλογική είσοδος A0 έχει συνδεθεί στο ποτενσιόμετρο POT1, η αναλογική είσοδος A1 έχει συνδεθεί στο PB1\_PWM φίλτρο και οι αναλογικές εισόδους A2 και A3 παραμένουν ασύνδετες.

Για να λειτουργήσουν σωστά οι αναλογικές εισόδους της PORTC πρέπει να αποσυνδεθεί η θύρα αυτή από τα LED δια μέσω του DIP switch.

#### **Παράδειγμα 4.1**

Στο παρακάτω παράδειγμα ο ADC έχει ρυθμιστεί έτσι ώστε να διαβάζει συνεχώς την αναλογική είσοδο 0. Το αποτέλεσμα της μετατροπής είναι αριστερά προσαρμοσμένο (Left adjusted) και τα 8 σημαντικότερα bit απεικονίζονται στα LED του PORTD.

Η τάση αναφοράς έχει ρυθμιστεί στα 5 volt.

Η συχνότητα του ADC ισούται με 125 KHz και προκύπτει από τη διαίρεση της συχνότητας λειτουργίας του μικροελεγκτή (16MHz) με το συντελεστή 128.



```

.include "m328PBdef.inc"           ;ATmega328P microcontroller definitions

.def temp = r16
.def ADC_L = r21
.def ADC_H = r22

.org 0x00
    jmp reset

.org 0x2A                           ;ADC Conversion Complete Interrupt
    reti

reset:
    ldi temp, high(RAMEND)
    out SPH,temp
    ldi temp, low(RAMEND)
    out SPL,temp

    ldi temp, 0xFF
    out DDRD, temp                 ;Set PORTD as output

    ldi temp, 0x00
    out DDRC, temp                 ;Set PORTC as input

    ; REFSn[1:0]=01 => select Vref=5V, MUXn[4:0]=0000 => select ADC0(pin PC0),
    ; ADLAR=1 => Left adjust the ADC result
    ldi temp, 0b01100000 ;
    sts ADMUX, temp

    ; ADEN=1 => ADC Enable, ADSC=0 => No Conversion,
    ; ADIE=0 => disable adc interrupt, ADPS[2:0]=111 => fADC=16MHz/128=125KHz
    ldi temp, 0b10000111
    sts ADCSRA, temp

Start_conv:
    lds temp, ADCSRA               ;
    ori temp, (1<<ADSC)           ; Set ADSC flag of ADCSRA
    sts ADCSRA, temp              ;

    wait_adc:                      0 ;
    lds temp, ADCSRA               ;
    sbrc temp,ADSC                 ; Wait until ADSC flag of ADCSRA becomes 0
    rjmp wait_adc                  ;

    lds ADC_L,ADCL                 ; Read ADC result(Left adjusted)
    lds ADC_H,ADCH                 ;

    out PORTD, ADC_H               ; Output ADCH to PORTD

    rjmp Start_conv

```

## Αλφαριθμητική Οθόνη 2×16 Χαρακτήρων

Η οθόνη είναι υγρού κρυστάλλου (LCD) με χαμηλή κατανάλωση και έχει ενσωματωμένο ένα μικροελεγκτή. Ο ενσωματωμένος μικροελεγκτής εμπεριέχει δύο μνήμες για τη δημιουργία χαρακτήρων: την CG-ROM (Character Generation ROM) και την CG-RAM (Character Generation RAM). Επίσης έχει μία μνήμη RAM στην οποία κρατούνται προσωρινά οι χαρακτήρες που απεικονίζονται κάθε φορά στην οθόνη την DD\_RAM (Display Data RAM).

### Ακροδέκτες της οθόνης

Στη συνέχεια παρατίθενται οι ακροδέκτες της οθόνης και οι αντίστοιχες λειτουργίες της:

**DB[0:7]** Τρικατάστατος αμφίδρομος διάδρομος δεδομένων 8-γραμμών, με τον οποίο η οθόνη ανταλλάσσει δεδομένα με τον εξωτερικό μικροεπεξεργαστή. Υπάρχει δυνατότητα χρήσης μόνο 4 γραμμών, οπότε η ανταλλαγή των δεδομένων γίνεται σε δύο φάσεις, δια μέσω των ακροδεκτών DB[7:4] ενώ οι ακροδέκτες DB[0:3] δεν χρησιμοποιούνται. Τα 4 περισσότερο σημαντικά bit πρέπει να μεταφέρονται πρώτα.

**E'** Σήμα που ενεργοποιεί την οθόνη. Η ενεργοποίηση γίνεται Αν  $E'=0$ .

**R/W'** Σήμα που καθορίζει αν πρόκειται να γίνει εγγραφή ή διάβασμα από την οθόνη ( $R/W'=0$  εγγραφή,  $R/W'=1$  διάβασμα).

**RS** Το σήμα RS καθορίζει αν το περιεχόμενο των DB[0:7] είναι διεύθυνση/εντολή ή αν είναι δεδομένα που πρόκειται να διαβαστούν ή να εγγραφούν ( $RS=0$  διεύθυνση/εντολή,  $RS=1$  δεδομένα)

**V0** Ρυθμίζει τη φωτεινότητα της οθόνης.

**VDD** Συνδέεται με την τάση τροφοδοσίας (3,3 volt).

**VSS** Συνδέεται με τη γη (0 volt).

### Καταχωρητές

Ο εσωτερικός μικροελεγκτής της LCD οθόνης ενσωματώνει δύο καταχωρητές των 8-bit τον καταχωρητή εντολών (instruction register -IR) και τον καταχωρητή δεδομένων (data register- DR).

Ο καταχωρητής IR αποθηκεύει τον κώδικα των εντολών, ή τη διεύθυνση μιας θέσης της μνήμης DD-RAM ή της μνήμης CG-RAM. Στον καταχωρητή IR γίνεται μόνο εγγραφή και όχι ανάγνωση από τον εξωτερικό μικροελεγκτή. Ο καταχωρητής DR αποθηκεύει προσωρινά τα δεδομένα που διαβάζονται ή γράφονται στην DD-RAM ή στην CG-RAM.

### Σημαία απασχόλησης (Busy Flag-BF)

Όταν η σημαία απασχόλησης είναι 1, η οθόνη είναι σε κατάσταση εσωτερικής λειτουργίας και δεν γίνεται δεκτή καμία άλλη εντολή. Όταν  $RS=0$  και  $R/W'=1$ , η σημαία απασχόλησης εμφανίζεται στην έξοδο DB7. Η επόμενη εντολή πρέπει να είναι δοθεί, μετά την εξασφάλιση ότι η σημαία απασχόλησης είναι 0.

### Μετρητής Διεύθυνση (Address Counter -AC)

Ο μετρητής διεύθυνση (AC) παρέχει διευθύνσεις σε αμφοτέρους τις μνήμες DD-RAM και CG-RAM. Όταν μια διεύθυνση μιας εντολής εγγράφεται στον IR, η διεύθυνση στέλνεται από τον IR στον AC. Η επιλογή είτε της DD-RAM ή της CG-RAM καθορίζεται επίσης από την εντολή.

Μετά την εγγραφή ή την ανάγνωση από τις DD-RAM ή CG-RAM, η AC αυτόματα αυξάνεται κατά 1 (ή ελαττώνεται κατά 1). Το περιεχόμενο του καταχωρητή AC δίνεται στη έξοδο μέσω των DB[0:6] όταν RS = 0 και R/W' = 1.

| RS | R/W | Operation  |
|----|-----|--|
| 0  | 0   | IR write as an internal operation (display clear, etc.)    |
| 0  | 1   | Read busy flag (DB7) and address counter (DB0 to DB6)      |
| 1  | 0   | DR write as an internal operation (DR to DD-RAM or CG-RAM) |
| 1  | 1   | DR read as an internal operation (DD-RAM or CG-RAM to DR)  |

**Πίνακας 4.6:** Επιλογή Καταχωρητή

#### Εσωτερική μνήμη DD-RAM

Η DD-RAM αποθηκεύει τα δεδομένα που πρόκειται να απεικονιστούν στην οθόνη. Η χωρητικότητά της είναι 80 byte. Ο κώδικας της διεύθυνσης μιας θέσης μνήμης DD-RAM είναι των επτά Bit. Οι χαρακτήρες που οι κώδικες της βρίσκονται στις θέσεις 0x00 έως 0x0F της DD-RAM τοποθετούνται(απεικονίζονται) στη γραμμή 1 της οθόνης, ενώ οι κώδικες που βρίσκονται στις θέσεις 0x40 έως 0x4F απεικονίζονται στη γραμμή 2 όπως φαίνονται στον παρακάτω πίνακα:

|          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Γραμμή_1 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F |
| Γραμμή_2 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 4A | 4B | 4C | 4D | 4E | 4F |

**Πίνακας 4.7:** Απεικόνιση χαρακτήρων

Αν πρέπει να απεικονιστούν και άλλοι χαρακτήρες που οι κώδικες τους είναι αποθηκευμένοι στη DD-RAM, εκτός από τους 32 χαρακτήρες που αναφέρθηκαν πιο πάνω, τότε πρέπει να γίνει ολίσθηση στο περιεχόμενο της οθόνης δεξιά ή αριστερά.

#### Εσωτερική μνήμη CG-ROM

Η CG-ROM χρησιμοποιείται για τη δημιουργία εικόνων χαρακτήρων 5×8 κουκίδων από κωδικούς χαρακτήρων των 8 bit. Η αντιστοιχία αριθμών των 8 bit και χαρακτήρων φαίνεται στον ακόλουθο πίνακα.

| Upper<br>4 bit<br>Lower<br>4 bit | LLLL             | LLLH | LLHL | LLHH | LHLL | LHLH | LHHL | LHHH | HLLL | HLLH | HLHL | HLHH | HHLL | HHLH | HHHL | HHHH |
|----------------------------------|------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| LLLL                             | CG<br>RAM<br>(1) | ±    |      | 0    | P    | '    | P    | S    | E    | Δ    | '    | r    | P    | P    | τ    |      |
| LLLH                             | CG<br>RAM<br>(2) | ≡    | !    | 1    | A    | Q    | a    | q    | Q    | a    | i    | '    | J    | +    | γ    | υ    |
| LLHL                             | CG<br>RAM<br>(3) | 7    | "    | 2    | B    | R    | b    | r    | é    | R    | é    | Δ    | '    | ∞    | §    | Σ    |
| LLHH                             | CG<br>RAM<br>(4) | L    | #    | 3    | O    | S    | c    | s    | Δ    | Δ    | Δ    | '    | P    | W    | E    | ψ    |
| LHLL                             | CG<br>RAM<br>(5) | 7    | *    | 4    | D    | T    | t    | t    | Δ    | Δ    | t    | '    | *    | r    | ζ    | ω    |
| LHLH                             | CG<br>RAM<br>(6) | 7    | %    | 5    | E    | U    | e    | u    | Δ    | Δ    | e    | '    | *    | ↑    | Δ    | η    |
| LHHL                             | CG<br>RAM<br>(7) | 7    | 8    | 6    | F    | V    | v    | v    | Δ    | Δ    | v    | '    | w    | ↓    | E    | Θ    |
| LHHH                             | CG<br>RAM<br>(8) | 7    | '    | 7    | G    | W    | w    | w    | Δ    | Δ    | w    | '    | x    | *    | Δ    | Δ    |
| HLLL                             | CG<br>RAM<br>(1) | 7    | (    | 8    | H    | X    | x    | x    | Δ    | Δ    | x    | '    | ÷    | *    | E    | K    |
| HLLH                             | CG<br>RAM<br>(2) | 7    | )    | 9    | I    | Y    | i    | y    | Δ    | Δ    | i    | '    | Δ    | Γ    | Π    | Δ    |
| HLHL                             | CG<br>RAM<br>(3) | *    | *    | #    | J    | Z    | z    | z    | Δ    | Δ    | z    | '    | Δ    | Δ    | Δ    | Δ    |
| HLHH                             | CG<br>RAM<br>(4) | 7    | +    | #    | K    | C    | k    | c    | Δ    | Δ    | c    | '    | *    | L    | Γ    | Δ    |
| HHLL                             | CG<br>RAM<br>(5) | ≡    | .    | <    | L    | \    | l    | l    | Δ    | Δ    | Δ    | '    | *    | Δ    | Δ    | Δ    |
| HHLH                             | CG<br>RAM<br>(6) | ∞    | —    | ≡    | M    | ]    | m    | ]    | Δ    | Δ    | Δ    | '    | *    | .    | ψ    | π    |
| HHHL                             | CG<br>RAM<br>(7) | #    | .    | >    | N    | ^    | n    | ^    | Δ    | Δ    | Δ    | '    | Δ    | Δ    | ρ    | Δ    |
| HHHH                             | CG<br>RAM<br>(8) | ≡    | /    | ?    | O    | _    | o    | Δ    | Δ    | Δ    | Δ    | '    | Δ    | Δ    | σ    | Δ    |

Πίνακας 4.8: Αντιστοιχία αριθμών των 8 bit και χαρακτήρων.

## Εντολές της οθόνης

Επειδή δεν υπάρχει δυνατότητα ανάγνωσης της σημαίας BUSY FLAG για να γνωρίζουμε πότε η οθόνη είναι έτοιμη να δεχτεί νέα εντολή πρέπει να εισάγουμε καθυστέρηση μεταξύ των διαδοχικών εντολών. Ο χρόνος που χρειάζεται για την εκτέλεση κάθε εντολής είναι μεταβλητός και εξαρτάται από την κάθε εντολή. Το σύνολο εντολών της οθόνης περιέχεται στον ακόλουθο πίνακα.

| Instruction                | Instruction Code |     |     |     |     |     |     |     |     |     | Description   | Execution time<br>(fosc=270Khz)  |        |
|----------------------------|------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|--|--------|
|                            | RS               | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |   |  |        |
| Clear Display              | 0                | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | Write “00H” to DDRAM and set DDRAM address to “00H” from AC | 1.53ms   |        |
| Return Home                | 0                | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | —   | Set DDRAM address to “00H” from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.   | 1.53ms |
| Entry Mode Set             | 0                | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | I/D | SH  | Assign cursor moving direction and enable the shift of entire display.   | 39 μ s |
| Display ON/OFF Control     | 0                | 0   | 0   | 0   | 0   | 0   | 0   | 1   | D   | C   | B   | Set display (D), cursor (C), and blinking of cursor (B) on/off control bit.  | 39 μ s |
| Cursor or Display Shift    | 0                | 0   | 0   | 0   | 0   | 0   | 1   | S/C | R/L | —   | —   | Set cursor moving and display shift control bit, and the direction, without changing of DDRAM data.                                | 39 μ s |
| Function Set               | 0                | 0   | 0   | 0   | 0   | 1   | DL  | N   | F   | —   | —   | Set interface data length (DL:8-bit/4-bit), numbers of display line (N:2-line/1-line)and, display font type (F:5×11 dots/5×8 dots) | 39 μ s |
| Set CGRAM Address          | 0                | 0   | 0   | 1   | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 |   | Set CGRAM address in address counter.  | 39 μ s |
| Set DDRAM Address          | 0                | 0   | 1   | AC6 | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 |   | Set DDRAM address in address counter.  | 39 μ s |
| Read Busy Flag and Address | 0                | 1   | BF  | AC6 | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 |   | Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.             | 0 μ s  |
| Write Data to RAM          | 1                | 0   | D7  | D6  | D5  | D4  | D3  | D2  | D1  | D0  |   | Write data into internal RAM (DDRAM/CGRAM).  | 43 μ s |
| Read Data from RAM         | 1                | 1   | D7  | D6  | D5  | D4  | D3  | D2  | D1  | D0  |   | Read data from internal RAM (DDRAM/CGRAM).   | 43 μ s |

\* "—" : don't care

**Πίνακας 4.9:** Το σύνολο εντολών της οθόνης

Τέλος για να γίνει δεκτή μια εντολή από τον ελεγκτή πρέπει να τηρούνται οι χρόνοι του παρακάτω πίνακα:

Ta=25°C, VDD=5.0V

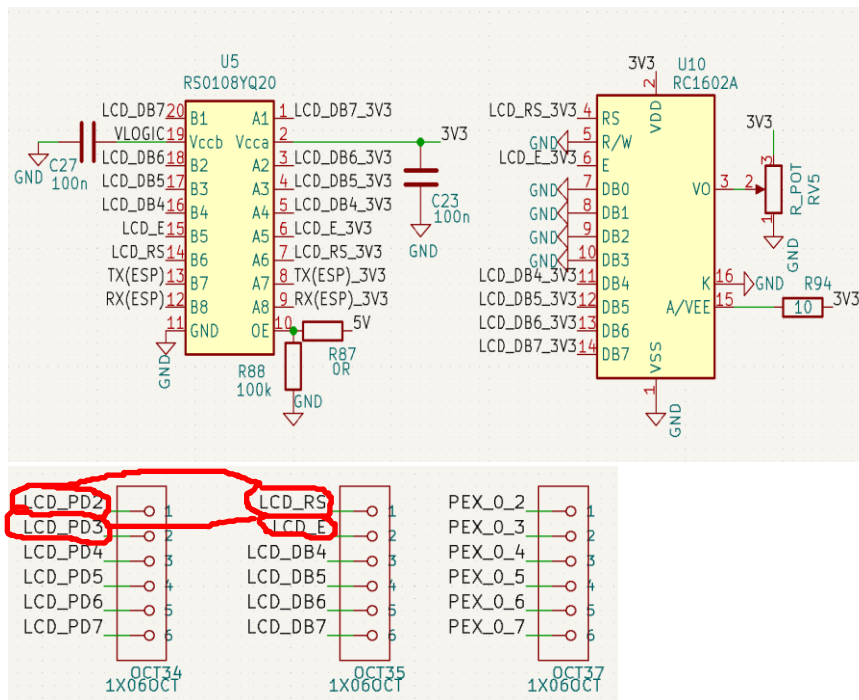
| Item                               | Symbol                          | Min  | Typ | Max | Unit |
|------------------------------------|---------------------------------|------|-----|-----|------|
| Enable cycle time                  | T <sub>C</sub>                  | 1200 | —   | —   | ns   |
| Enable pulse width                 | T <sub>PW</sub>                 | 140  | —   | —   | ns   |
| Enable rise/fall time              | T <sub>R</sub> , T <sub>F</sub> | —    | —   | 25  | ns   |
| Address set-up time (RS, R/W to E) | t <sub>AS</sub>                 | 0    | —   | —   | ns   |
| Address hold time                  | t <sub>AH</sub>                 | 10   | —   | —   | ns   |
| Data set-up time                   | t <sub>DSW</sub>                | 40   | —   | —   | ns   |
| Data hold time                     | t <sub>H</sub>                  | 10   | —   | —   | ns   |

**Πίνακας 4.10:** Χρονισμός εντολών οθόνης

#### Χρήση της οθόνης LCD 2x16 της κάρτας ntuAboard\_G1.

Η κάρτα ntuAboard\_G1 διαθέτει μία LCD οθόνη 2x16 χαρακτήρων. Μεταξύ της οθόνης και του μικροελεγκτή παρεμβάλλεται το ολοκληρωμένο RS0108YQ20 (8-Bit Bidirectional Voltage Level Translator) για προσαρμογή του επιπέδου της τάσης του μικροελεγκτή (5Volt) με το επίπεδο της τάσης της οθόνης (3.3 Volt).

Η συνδεσμολογία της οθόνης LCD με τον Μικροελεγκτή ATmega328PB φαίνεται στο παρακάτω σχήμα:



**Σχήμα 4.9:** Συνδεσμολογία της οθόνης LCD

Κάνοντας χρήση των κονεκτόρων OCT34, OCT35 και OCT37 μπορούμε να επιλέξουμε εάν οι ακροδέκτες ελέγχου της LCD οθόνης (OCT35) θα συνδεθούν στο PORTD του ATmega328PB (OCT34) ή στο PORT EXPANDER (OCT37). Ανάλογα με τη σύνδεση, οι ακροδέκτες [LCD\_PD2 : LCD\_PD7] ή οι ακροδέκτες [PEX\_0\_7 : PEX\_0\_2] πρέπει να είναι ρυθμισμένοι για έξοδο. **Η επικοινωνία μεταξύ οθόνης και του μικροελεγκτή γίνεται με λέξεις των 4 bit.**

#### Κώδικας επικοινωνίας μεταξύ μικροελεγκτή και οθόνης

Αρχικά χρειαζόμαστε μια ρουτίνα που θα μεταφέρει τα δύο τμήματα των 4 bit κάθε εντολής. Η ρουτίνα θα πρέπει να αφήνει ανεπηρέαστους τους ακροδέκτες που επιλέγουν μεταξύ καταχωρητή εντολών και καταχωρητή δεδομένων, ώστε να μπορεί να χρησιμοποιηθεί και για τις δύο λειτουργίες.

#### **Ρουτίνα: write\_2\_nibbles**

**write\_2\_nibbles:** argument register (r24)

```

push r24                ; στέλνει τα 4 MSB
in r25 ,PIND             ; διαβάζονται τα 4 LSB και τα ξαναστέλνουμε
andi r25 ,0x0f           ; για να μην χαλάσουμε την όποια προηγούμενη κατάσταση
andi r24 ,0xf0           ; απομονώνονται τα 4 MSB και
add r24 ,r25             ; συνδυάζονται με τα προϋπάρχοντα 4 LSB
out PORTD ,r24           ; και δίνονται στην έξοδο
sbi PORTD ,PD3           ; δημιουργείται παλμός Enable στον ακροδέκτη PD3      μ   enable???
cbi PORTD ,PD3           ; PD3=1 και μετά PD3=0
pop r24                  ; στέλνει τα 4 LSB. Ανακτάται το byte.
swap r24                 ; εναλλάσσονται τα 4 MSB με τα 4 LSB
andi r24 ,0xf0           ; που με την σειρά τους αποστέλλονται
add r24 ,r25
out PORTD ,r24
sbi PORTD ,PD3           ; Νέος παλμός Enable
cbi PORTD ,PD3
ret

```

Στη συνέχεια, με βάση την προηγούμενη ρουτίνα θα δημιουργήσουμε δύο άλλες. Η μία θα στέλνει εντολές στην οθόνη και η άλλη δεδομένα.

#### **Ρουτίνα: lcd\_data**

Αποστολή ενός byte δεδομένων στον ελεγκτή της οθόνης lcd. **Ο ελεγκτής πρέπει να βρίσκεται σε 4 bit mode. Το byte που μεταδίδεται είναι αποθηκευμένο στον καταχωρητή r24**

#### **lcd\_data:**

```

sbi PORTD ,PD2           ; επιλογή του καταχωρητή δεδομένων (PD2=1)
rcall write_2_nibbles    ; αποστολή του byte

ldi r24 ,43              ; αναμονή 43μsec μέχρι να ολοκληρωθεί η λήψη
ldi r25 ,0                ; των δεδομένων από τον ελεγκτή της lcd
rcall wait_usec

ret

```

### Ρουτίνα: lcd\_command

Αποστολή μιας εντολής στον ελεγκτή της οθόνης lcd.

Ο ελεγκτής πρέπει να βρίσκεται σε 4 bit mode.

Η εντολή που μεταδίδεται είναι αποθηκευμένη στον καταχωρητή r24

#### lcd\_command:

```
cbi PORTD ,PD2      ; επιλογή του καταχωρητή εντολών (PD2=D)
rcall write_2_nibbles ; αποστολή της εντολής και αναμονή 39μsec

ldi r24 ,39          ; για την ολοκλήρωση της εκτέλεσης της από τον ελεγκτή της lcd.
ldi r25 ,0            ; ΣΗΜ.: υπάρχουν δύο εντολές, clear display και return home,
rcall wait_usec      ; που απαιτούν σημαντικά μεγαλύτερο χρονικό διάστημα.

ret
```

Τώρα που μπορούμε να στείλουμε εντολές και δεδομένα στην οθόνη μένει να την αρχικοποιήσουμε στην επιθυμητή κατάσταση, ώστε να μπορεί να χρησιμοποιηθεί. Όταν η οθόνη τροφοδοτείται με ρεύμα για πρώτη πραγματοποιείται μια εσωτερική αρχικοποίηση και για αυτό απαιτείται να περιμένουμε 40 ms. Στη συνέχεια ο ελεγκτής βρίσκεται σε 8 bit mode και είναι έτοιμος να λάβει εντολές. Ο κώδικας που θα κάνει την αρχικοποίηση δεν πρέπει να βασίζεται στο ότι ο ελεγκτής βρίσκεται σε 8 bit mode, διότι αυτό δεν είναι πάντα αληθές. Κάθε φορά που προγραμματίζουμε τον Μικροελεγκτή, αυτός ξεκινάει την εκτέλεση του κώδικα από την αρχή, η οθόνη όμως βρίσκεται στην κατάσταση που την αφήσαμε την προηγούμενη φορά. Για να οδηγήσουμε την οθόνη σε 4 bit mode στέλνουμε δύο φορές την εντολή 0x30 (function set) για 8 bit mode. Η συγκεκριμένη εντολή (μεταξύ άλλων) ρυθμίζει τον ελεγκτή να δέχεται εντολές και δεδομένα σε ένα ενιαίο κομμάτι των 8 bit. Τα 4 λιγότερο σημαντικά bit μας είναι αδιάφορα. Αν ο ελεγκτής είναι σε 8 bit mode δεν θα αλλάξει κάτι, αν όμως είναι σε 4 bit mode θα μεταβεί σε 8 bit mode. Μόλις είμαστε βέβαιοι για την μορφή που πρέπει να στέλνουμε τις εντολές μπορούμε να προχωρήσουμε με την αρχικοποίηση.

### Ρουτίνα: lcd\_init

Αρχικοποίηση και ρυθμίσεις της οθόνης LCD όπως παρουσιάζεται παρακάτω:

DL = 0 4 bit mode 4.9 DL = 0 μ 8bit mode?

N = 1 2 lines

F = 0 5×8 dots

D = 1 display on

C = 0 cursor off

B = 0 blinking off

I/D = 1 DD-RAM address auto increment

SH = 0 shift of entire display off

Είσοδος: -

Έξοδος: -

Καταχωρητές: r25:r24

Καλούμενες υπορουτίνες: wait\_msec, wait\_usec, lcd\_command



## lcd\_init:

```
ldi r24 ,40          ; Όταν ο ελεγκτής της lcd τροφοδοτείται με
ldi r25 ,0            ; ρεύμα εκτελεί την δική του αρχικοποίηση.
rcall wait_msec       ; Αναμονή 40 msec μέχρι αυτή να ολοκληρωθεί.

ldi r24 ,0x30         ; εντολή μετάβασης σε 8 bit mode
out PORTD ,r24        ; επειδή δεν μπορούμε να είμαστε βέβαιοι
sbi PORTD ,PD3        ; για τη διαμόρφωση εισόδου του ελεγκτή
cbi PORTD ,PD3        ; της οθόνης, η εντολή αποστέλλεται δύο φορές
ldi r24 ,39
ldi r25 ,0            ; εάν ο ελεγκτής της οθόνης βρίσκεται σε 8-bit mode
rcall wait_usec       ; δεν θα συμβεί τίποτα, αλλά αν ο ελεγκτής έχει διαμόρφωση
                      ; εισόδου 4 bit θα μεταβεί σε διαμόρφωση 8 bit

ldi r24 ,0x30
out PORTD ,r24
sbi PORTD ,PD3
cbi PORTD ,PD3
ldi r24 ,39
ldi r25 ,0
rcall wait_usec

ldi r24 ,0x20         ; αλλαγή σε 4-bit mode
out PORTD ,r24
sbi PORTD ,PD3
cbi PORTD ,PD3
ldi r24 ,39
ldi r25 ,0
rcall wait_usec

ldi r24 ,0x28         ; επιλογή χαρακτήρων μεγέθους 5x8 κουκίδων
rcall lcd_command     ; και εμφάνιση δύο γραμμών στην οθόνη

ldi r24 ,0x0c         ; ενεργοποίηση της οθόνης, απόκρυψη του κέρσορα
rcall lcd_command

ldi r24 ,0x01         ; καθαρισμός της οθόνης
rcall lcd_command

ldi r24 ,low(1530)
ldi r25 ,high(1530)
rcall wait_usec

ldi r24 ,0x06         ; ενεργοποίηση αυτόματης αύξησης κατά 1 της διεύθυνσης
rcall lcd_command     ; που είναι αποθηκευμένη στον μετρητή διευθύνσεων και
                      ; απενεργοποίηση της ολίσθησης ολόκληρης της οθόνης

ret
```

## Παράδειγμα 4.2

Απεικόνιση του χαρακτήρα 'A' στην οθόνη:

main:

```
ldi r24, low(RAMEND)
out SPL, r24
ldi r24, high(RAMEND)
out SPH, r24      ; αρχικοποίηση stack pointer
ser r24
out DDRD, r24     ; αρχικοποίηση PORTD που συνδέεται η οθόνη, ως έξοδος
clr r24
```

main1: rcall lcd\_init ; αρχικοποίηση οθόνης

```
ldi r24, low(2)
ldi r25, high(2)  ; Αναμονή 2 msec
rcall wait_msec
```

```
ldi r24, 'A'
rcall lcd_data    ; αποστολή ενός byte δεδομένων στον ελεγκτή της οθόνης lcd
```

```
ldi r24, low(2000)
ldi r25, high(2000) ; Αναμονή 2 sec
rcall wait_msec
```

jmp main1

## Τα ζητούμενα της 4<sup>ης</sup> εργαστηριακής άσκησης

### Ζήτημα 4.1

Στο πλαίσιο της άσκησης θα μελετηθούν δύο τρόποι χειρισμού του ADC.

Ο πρώτος τρόπος θα είναι με την χρήση της διακοπής ολοκλήρωσης της μετατροπής του ADC. Η διακοπή αυτή (ADC) μεταφέρει τον έλεγχο στην διεύθυνση **0x02A**, αν είναι ενεργοποιημένη η αντίστοιχη διακοπή (από το bit ADIE του ADCSRA) καθώς και οι γενικές διακοπές.

Ο δεύτερος τρόπος είναι το πρόγραμμα να αναμένει να ολοκληρωθεί η μετατροπή. Η αναμονή αυτή γίνεται μέσω προγράμματος, ελέγχοντας το bit ADSC του ADCSRA το οποίο γίνεται 0 μόλις ολοκληρωθεί η μετατροπή (Polling).

(α) Να γραφεί πρόγραμμα σε assembly για τον ATmega328PB το οποίο θα ξεκινάει μια μετατροπή του ADC και θα αυξάνει έναν μετρητή ο οποίος θα εμφανίζεται σε δυαδική μορφή στα LED της PORTB κάθε 1Sec. Η ανάγνωση των δεδομένων του ADC πρέπει να γίνεται μέσα στην ρουτίνα εξυπηρέτησης της διακοπής ολοκλήρωσης μετατροπής του ADC και τα δεδομένα αυτά πρέπει να μετατρέπονται σε τάση και να εκτυπώνονται στην LCD οθόνη, αρχίζοντας κάθε φορά από τον πρώτο χαρακτήρα της πρώτης γραμμής, με ακρίβεια δύο δεκαδικών ψηφίων (δεν χρειάζεται στρογγυλοποίηση).

Η τάση δίνεται από τον τύπο:

$$V_{IN} = \frac{ADC}{1024} V_{REF}$$

Όπου:

- $V_{IN}$  η τάση στην αναλογική είσοδο A2 του μικροελεγκτή.
- ADC η τιμή που διαβάζεται από τον ADC (10bit, από 0-1023)
- $V_{REF}$  η τάση αναφοράς που έχει οριστεί στα 5V.

(β) Να γραφεί σε C πρόγραμμα για τον ATmega328PB το οποίο θα ξεκινάει μια μετατροπή, θα περιμένει να ολοκληρωθεί η μετατροπή και θα εκτυπώνει, όπως πριν, την τάση στην LCD με ακρίβεια δύο δεκαδικών ψηφίων

### Ζήτημα 4.2

Να δημιουργηθεί κώδικας επιτήρησης ενός χώρου όπου υπάρχει αυξημένος κίνδυνος ύπαρξης μονοξειδίου του άνθρακα (CO). Ο αισθητήρας CO είναι συνδεδεμένος στην αναλογική είσοδο A3 του μικροελεγκτή.

Καθ' όλη την διάρκεια πρέπει να διαβάζεται η τιμή του αισθητήρα ανά 100 ms (μικρές αποκλίσεις είναι αποδεκτές) και να εμφανίζεται μια ένδειξη του επιπέδου του αερίου στα LED PB0-PB5.

Αν οποιαδήποτε στιγμή η συγκέντρωση του CO ξεπεράσει τα 70ppm να τυπώνεται στην LCD το μήνυμα GAS DETECTED και να αναβοσβήνουν τα αντίστοιχα LED στα PB0-PB5 αναλόγως του επιπέδου του αερίου.

Το επίπεδο θα πρέπει να συνεχίζει να διαβάζεται (και να εμφανίζεται στα LED) και αν επανέλθει σε επίπεδο χαμηλότερο των 70ppm τα LEDs να σταματούν να αναβοσβήνουν και να τυπώνεται στην LCD το μήνυμα CLEAR.

**4.2α:** Υλοποίηση σε Assembly. Στην assembly η ανάγνωση των δεδομένων του ADC πρέπει να γίνεται μέσα στην ρουτίνα εξυπηρέτησης της διακοπής ολοκλήρωσης μετατροπής του ADC.

**4.2β:** Υλοποίηση σε C. Στην C θα χρησιμοποιήσετε την μέθοδο polling να διαβάσετε τον ADC.

Είναι καλή πρακτική να διατηρείτε τις ρουτίνες εξυπηρέτησης διακοπών όσο το δυνατόν μικρότερες σε χρονική

διάρκεια. Παρόλα αυτά αν το πρόγραμμα σας είναι λειτουργικό δεν θα υπάρξει αρνητική επίπτωση στην βαθμολογία.

Για την μέτρηση της συγκέντρωσης του CO χρησιμοποιείται αισθητήρας του τύπου ULPSM-CO-968-001 ([https://helios.ntua.gr/pluginfile.php/29155/mod\\_folder/content/0/ULPSM-CO-968-001.pdf?forcedownload=1](https://helios.ntua.gr/pluginfile.php/29155/mod_folder/content/0/ULPSM-CO-968-001.pdf?forcedownload=1)), ο οποίος βρίσκεται σε σταθερές συνθήκες θερμοκρασίας (20°C) και υγρασίας (40%). Δίνεται ότι  $V_{gas0}=0.1V$  και sensitivity code 129nA/ppm. Για να προσομοιώσετε την έξοδο που θα είχε ο αισθητήρας CO να χρησιμοποιήσετε το ποτενσιόμετρο POT3 στην κάρτα ntuAboard\_G1 το οποίο παράγει μια DC τάση στο εύρος 0-5V.

### Ζήτημα 4.3

Να δημιουργηθεί κώδικας σε γλώσσα C, ο οποίος να παράγει μία PWM κυματομορφή στον ακροδέκτη PB1 με συχνότητα 5KHz όταν είναι πατημένο κάποιο από τα πλήκτρα PB2 – PB5, ενώ όταν δεν είναι πατημένο κάποιο από αυτά τα πλήκτρα τότε δεν παράγεται κυματομορφή. Το duty cycle των κυματομορφών που παράγονται, εξαρτώνται από το πλήκτρο που είναι πατημένο και πρέπει να είναι όσο πιο κοντά γίνεται στις τιμές του παρακάτω πίνακα:

| ΠΛΗΚΤΡΟ | duty cycle |
|---------|------------|
| PB.2    | 20%        |
| PB.3    | 40%        |
| PB.4    | 60%        |
| PB.5    | 80%        |

Στην εκπαιδευτική κάρτα ntuAboard\_G1, συνδέστε ένα βραχυκυκλωτήρα μεταξύ A1 και PB1\_PWM και άλλον ένα βραχυκυκλωτήρα στον κονέκτορα J5 έτσι ώστε η έξοδος του φίλτρου της PWM κυματομορφή στον ακροδέκτη PB1 να οδηγείται στην αναλογική είσοδο 1 του ADC.

Στην LCD οθόνη να απεικονίζεται στην πρώτη γραμμή η τιμή του duty cycle και στη δεύτερη γραμμή η τιμή της τάσης που μετρά ο ADC με ακρίβεια 2 δεκαδικών ψηφίων.

Θεωρητικά η DC τάση στην έξοδο του φίλτρου πρέπει να ισούται με  $V_{out}=DC \cdot V_{in}$ , όπου  $V_{in}$  είναι η τάση του παλμού και DC είναι το duty cycle της PWM κυματομορφής.