

Αναφορά Milestone 2

Ημερομηνία έκδοσης : 14/12/2017

Όνομα project : <<Ακούγοντας τα χρώματα >>

Ομάδα:

Δώρας Κωνσταντίνος 2014030060

Παλαιολόγκ Χαρίτων 2014030051

Στο milestone 2 είχαμε θέσει ως στόχο να υλοποιήσουμε τη διεπαφή μεταξύ του μικροεπεξεργαστή και του αισθητήρα χρωμάτων. Στη παρούσα φάση η έξοδος του συστήματός μας είναι τέσσερα LED (κόκκινο, πράσινο, μπλε, κίτρινο) αφού η χρήση ηχητικών τόνων ήταν καθορισμένη για το επόμενο Milestone.

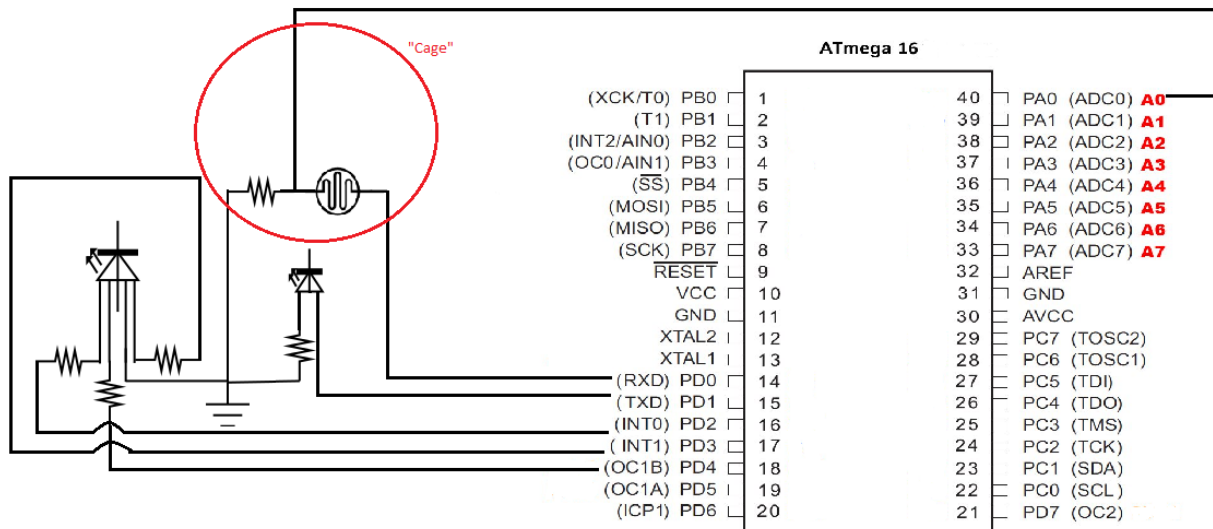
Για να πραγματοποιήσουμε τα παραπάνω χρησιμοποιήσαμε :

- ATMega16 microcontroller.
- Atmel Studio 7 για τον προγραμματισμό του μικροελεγκτή.
- STK500 για τη σύνδεση του υπολογιστή με τον μικροεπεξεργαστή.

Για τον αισθητήρα χρωμάτων χρειαστήκαμε :

- Αντιστάσεις 10KΩ.
- LDR αντίσταση για την ανίχνευση των χρωμάτων.
- Ένα κίτρινο και ένα RGB Led για την υπόδειξη των χρωμάτων.
- Ένα αυτοσχέδιο κουτί , με το οποίο απομονώναμε την LDR αντίσταση για να αποφύγουμε το θόρυβο από την ένταση του φωτός.

Το κύκλωμα φαίνεται στην εικόνα:



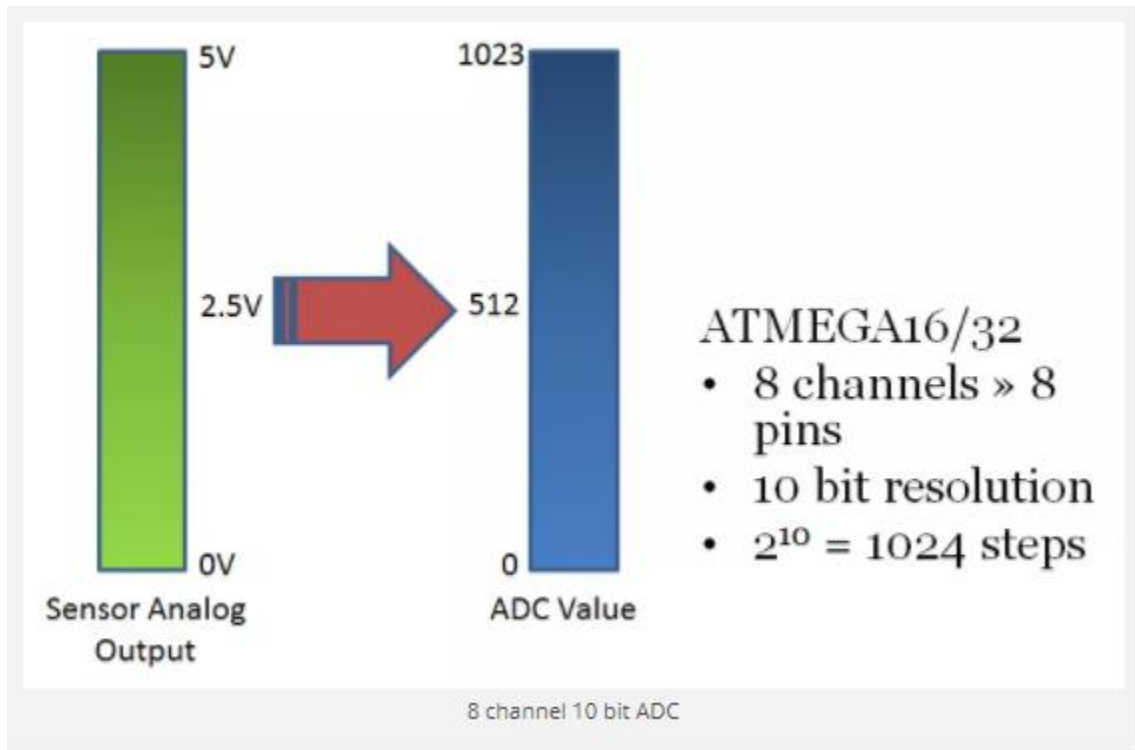
Περιγραφή:

Η τροφοδοσία του αισθητήρα μας γίνεται μέσω του PD0 pin του μικροελεγκτή (5.24Volt), ενώ τα Led τα συνδέουμε στα PD1:4 και ενεργοποιούμε το ADC στο PA0. Αρχικά μετρήσαμε την τάση εξόδου του αισθητήρα για κάθε χρώμα. Στη συνέχεια αυτήν την τάση την αντιστοιχίσαμε σε αναπαράσταση 10 bit (ο καταχωρητής του ADC είναι 10 Bit) για να ψηφιοποιήσουμε την αναλογική τιμή της τάσης μέσω της μαθηματικής σχέσης $X_{adc}=1024 \frac{V_{ldr}}{V_{cc}}$ όπου $V_{cc}=5.24V$ (προκύπτει από απλή μέθοδο των τριών, στα 5.24Volt αντιστοιχίζεται ο αριθμός 1023 και στην V_{ldr} που μετρήσαμε μέσω βολτόμετρου αντιστοιχίζεται η X_{adc} , **εικόνα 1**). Για το κάθε χρώμα υπολογίστηκε:

- Κόκκινο $273 \leq X_{\text{adc}} \leq 307$
- Πράσινο $308 \leq X_{\text{adc}} \leq 348$
- Μπλε $107 \leq X_{\text{adc}} \leq 147$
- Κίτρινο $382 \leq X_{\text{adc}} \leq 422$

ΣΗΜΕΙΩΣΗ: Τα εύρη τιμών δεν υπολογίστηκαν μέσω κάποιας μαθηματικής σχέσης αλλά μετά από testing και πολλές δοκιμές, δηλαδή πήραμε μετρήσεις σε διάφορες ώρες της μέρας, με διαφορετικό φωτισμό, με αποτέλεσμα να έχουμε μετρήσεις με ποικιλία εξωτερικού θορύβου από το περιβάλλον και καταλήξαμε στα παραπάνω όρια.

Εικόνα 1



Η λειτουργία του κυκλώματος ακολουθεί τη λογική ό,τι αν η τιμή του ADC βρίσκεται μέσα στα παραπάνω όρια που υπολογίσαμε τότε ενεργοποιούμε το αντίστοιχο PORT (PORTD1:4) του χρώματος που ανιχνεύσαμε.

Αναλυτική περιγραφή :

- Αρχικά στο initialization από τη μάσκα DDRD θέτουμε τα pins 0:4 ως εξόδους, καθώς επίσης αρχικοποιούμε το PD0 μονίμως σε άσσο για να το χρησιμοποιήσουμε για την τροφοδοσία του αισθητήρα.

Port D Data Register – PORTD

Bit	7	6	5	4	3	2	1	0	
	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port D Data Direction Register – DDRD

Bit	7	6	5	4	3	2	1	0	
	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port D Input Pins Address – PIND

Bit	7	6	5	4	3	2	1	0	
	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

- Στη συνέχεια κάναμε την αρχικοποίηση του ADC μέσω της συνάρτησης setupADC(), όπου ενεργοποιούμε από τη μάσκα ADMUX τα κατάλληλα bit ώστε να θέσουμε το pin PA0 ως ADC και την τάση αναφοράς μας το Vcc του μικροελεγκτή. Πιο συγκεκριμένα κάνουμε άσσο το REFS0 για να πάρουμε την τάση αναφοράς και αφήνουμε μηδέν όλα τα MUX4:0 για να κάνουμε το PA0 ADC pin.

ADC Multiplexer Selection Register – ADMUX

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7:6 – REFS1:0: Reference Selection Bits

These bits select the voltage reference for the ADC, as shown in Table 83. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set). The internal voltage reference options may not be used if an external reference voltage is being applied to the AREF pin.

Table 83. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal Vref turned off
0	1	AVCC with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 2.56V Voltage Reference with external capacitor at AREF pin

- **Bits 4:0 – MUX4:0: Analog Channel and Gain Selection Bits**

The value of these bits selects which combination of analog inputs are connected to the ADC. These bits also select the gain for the differential channels. See [Table 84](#) for details. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set).

Table 84. Input Channel and Gain Selections

MUX4..0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
00000	ADC0	N/A		
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			

- Ύστερα από τη μάσκα ADCSRA κάναμε άσσο το ADEN για ενεργοποιήσουμε το ADC, το ADIE για να ενεργοποιήσουμε το interrupt του ADC και θέσαμε μέσω του ADPS το prescaler σε

$$\frac{F_{CPU}}{ADPS} = \frac{16000000}{128} = 125000 \text{ Hz.}$$

ADC Control and Status Register A – ADCSRA

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – ADEN: ADC Enable**

Writing this bit to one enables the ADC. By writing it to zero, the ADC is turned off. Turning the ADC off while a conversion is in progress, will terminate this conversion.

- **Bit 3 – ADIE: ADC Interrupt Enable**

When this bit is written to one and the I-bit in SREG is set, the ADC Conversion Complete Interrupt is activated.

- **Bits 2:0 – ADPS2:0: ADC Prescaler Select Bits**

These bits determine the division factor between the XTAL frequency and the input clock to the ADC.

ATmega16(L)

Table 85. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

- Τέλος καλούμε τη startADC() για να ξεκινήσει η μετατροπή της αναλογικής τάσης σε ψηφιακή 10-bit τιμή. Για να το πετύχουμε αυτό ενεργοποιήσαμε το ADSC.

ADC Control and Status Register A – ADCSRA

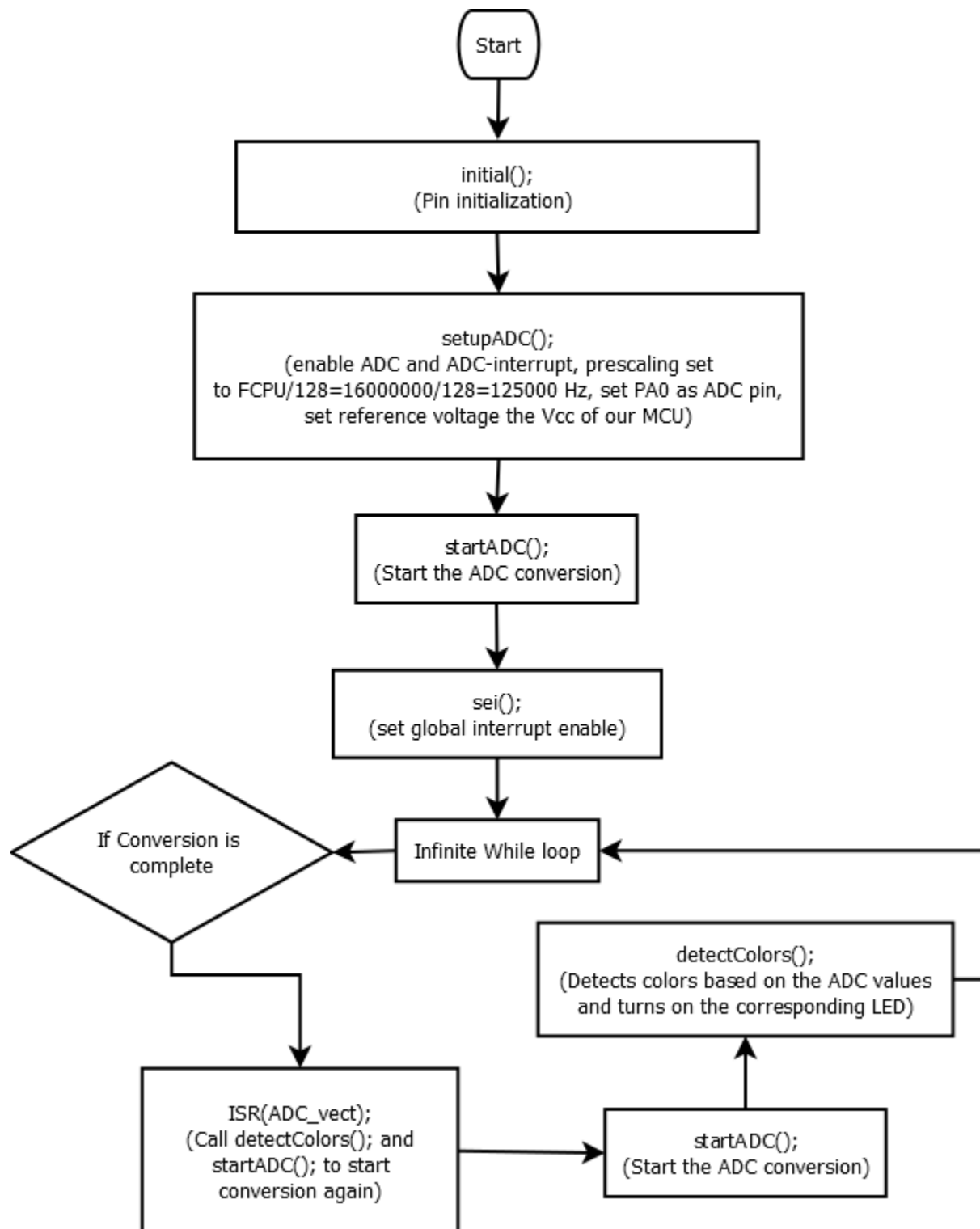
Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 6 – ADSC: ADC Start Conversion**

In Single Conversion mode, write this bit to one to start each conversion. In Free Running Mode, write this bit to one to start the first conversion. The first conversion after ADSC has been written after the ADC has been enabled, or if ADSC is written at the same time as the ADC is enabled, will take 25 ADC clock cycles instead of the normal 13. This first conversion performs initialization of the ADC.

ADSC will read as one as long as a conversion is in progress. When the conversion is complete, it returns to zero. Writing zero to this bit has no effect.

Flowchart



Βιβλιοθήκες :

- <avr/io.h>
- <avr/interrupt.h>

Πηγές / βιβλιογραφία :

- [Τα βασικά για τη χρήση του ADC](#)
- [ATMega16 datasheet](#)