

Αναφορά Milestone 3

Ημερομηνία έκδοσης : 21/12/2017

Όνομα project : <<Ακούγοντας τα χρώματα >>

Ομάδα:

Δώρας Κωνσταντίνος 2014030060

Παλαιολόγκ Χαρίτων 2014030051

Σκοπός του Project μας ήταν η δημιουργία ενός συστήματος που θα υποδεικνύει στον χρήστη τα χρώματα (πράσινο, κόκκινο, μπλε, κίτρινο) μέσω ηχητικών τόνων. Η ιδέα βασίστηκε στην ομιλία του Neil Harbisson στο TEDGlobal 2012.

Η υλοποίηση έγινε σε τρία στάδια, τρία Milestones. Αρχικά ήρθαμε σε μια πρώτη επαφή με το εργαστηριακό υλικό που μας δόθηκε και εξοικειωθήκαμε με το προγραμματιστικό περιβάλλον του Atmel Studio 7. Στα πλαίσια του πρώτου Milestone λοιπόν, φτιάξαμε ένα απλό κύκλωμα με 2 RGB LED και τα κάναμε να αναβοσβήνουν τα χρώματά τους με τη σειρά, το ένα μετά το άλλο, καθώς επίσης προσθέσαμε στο κύκλωμα μας ένα push button με τη χρήση του οποίου ανάβαμε ανά πάσα στιγμή το μπλε χρώμα του ενός από τα δύο LEDs.

Προχωρώντας στο δεύτερο Milestone στόχος μας ήταν η δημιουργία διεπαφής μεταξύ του αισθητήρα μας και του μικροεπεξεργαστή, κάτι που καταφέραμε επιτυχώς. Έτσι στο τελικό παραδοτέο της δεύτερης φάσης του πρότζεκτ μας ήμασταν σε θέση να αναγνωρίσουμε τα χρώματα στην οθόνη του κινητού μας και να τα υποδείξουμε μέσω των RGB LED που είχαμε στη διάθεση μας.

Αφού υλοποιήσαμε τη διεπαφή μεταξύ του αισθητήρα μας και του μικροεπεξεργαστή, προχωρήσαμε στο τελευταίο βήμα, που ήταν η παραγωγή ηχητικών τόνων, για το κάθε χρώμα, στην έξοδο του συστήματός μας. Για να το πετύχουμε αυτό χρησιμοποιήσαμε ως ηχείο ένα piezoelectric speaker/buzzer, το οποίο συνδέσαμε σε κατάλληλο pin του MCU μας ώστε, χρησιμοποιώντας fast pulse width modulation στην πρώτη υλοποίηση και phase and frequency modulation στη δεύτερη, να το κάνουμε να βγάζει διαφορετικούς τόνους. Επιπλέον για να γίνει το πρότζεκτ μας πιο ρεαλιστικό, και να λειτουργεί το συστήμά μας και με αντικείμενα πέρα της οθόνης του κινητού μας, τροποποιήσαμε τον αισθητήρα προσθέτοντάς του 2 λευκά LED για να μπορούμε μέσω της αντανάκλασης του φωτός τους στα 'υπό αναγνώριση' αντικείμενα να αναγνωρίσουμε το χρώμα τους. Τέλος, μία ακόμα αλλαγή στη δεύτερη υλοποίησή μας ήταν η προσθήκη ενός push button με τη χρήση του οποίου ανάβαν τα λευκά LED του αισθητήρα μας και γινόταν η αναγνώριση των χρωμάτων και η παραγωγή των ηχητικών τόνων.

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 :

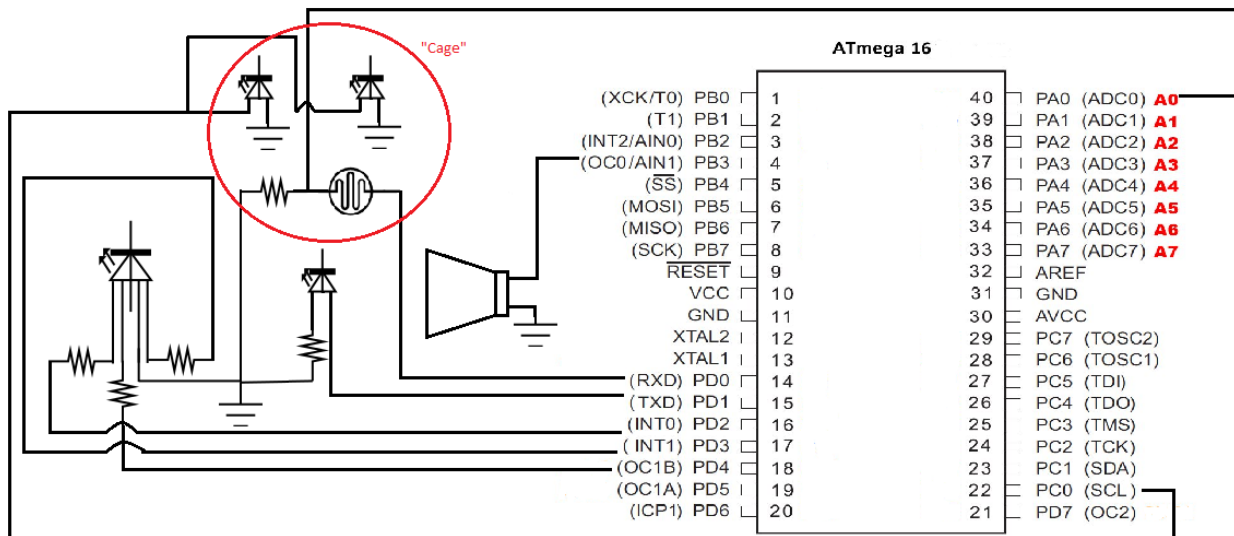
- Αντιστάσεις 10KΩ.
- LDR αντίσταση για την ανίχνευση των χρωμάτων.
- Ένα αυτοσχέδιο κουτί , με το οποίο απομονώναμε την LDR αντίσταση για να αποφύγουμε το θόρυβο από την ένταση του φωτός στον περιβάλλοντα χώρο.

Για την έξοδο του συστήματος χρησιμοποιήσαμε:

- Αντιστάσεις 10KΩ.
- Ένα κίτρινο και ένα RGB Led για την υπόδειξη των χρωμάτων.
- Ένα piezoelectric speaker/buzzer για την αναπαραγωγή ηχητικών τόνων.

Fast PWN

Το κύκλωμα για την υλοποίηση με Fast PWM φαίνεται στην εικόνα:



Περιγραφή:

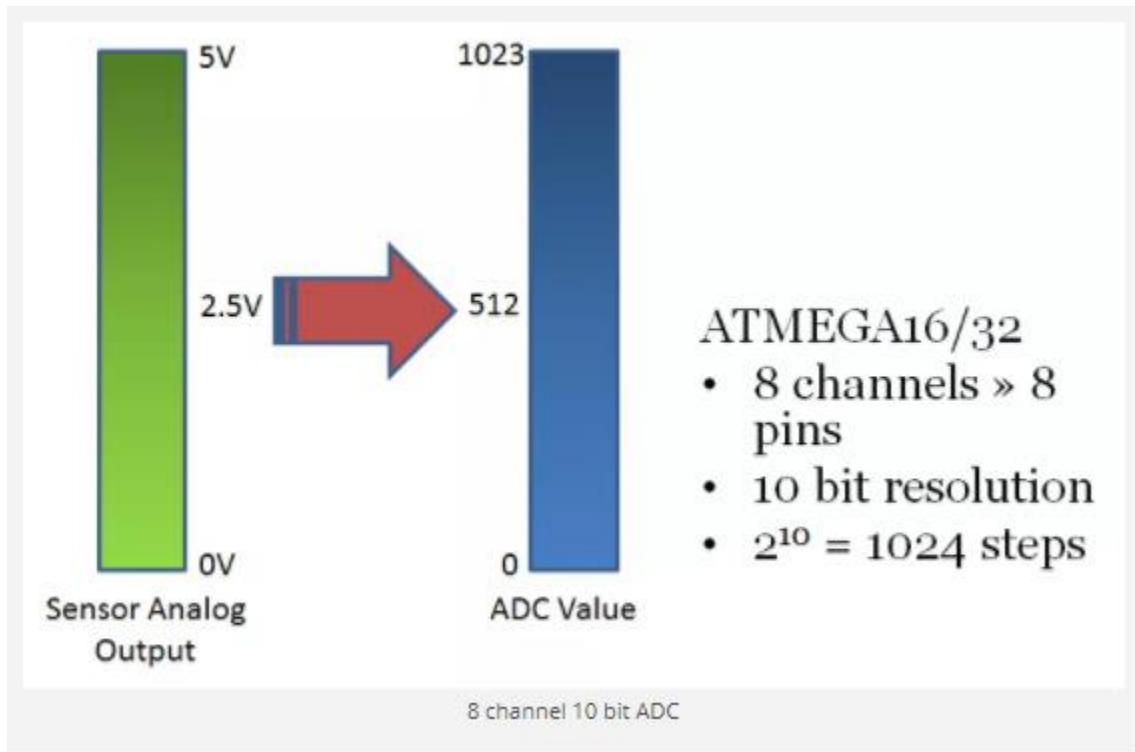
Η τροφοδοσία του αισθητήρα μας γίνεται μέσω του PD0 pin του μικροελεγκτή (5.24Volt), ενώ τα LED τα συνδέουμε στα PD1:4 και ενεργοποιούμε το ADC στο PA0. Η τροφοδοσία των 2 λευκών LED που έχουμε βάλει στον αισθητήρα μας γίνεται μέσω του pin PC0 (χρησιμοποιήσαμε χώρια τροφοδοσία για τα LED λόγω του push button που προσθέτουμε στη δεύτερη υλοποίηση). 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 $500 \leq X_{adc} \leq 550$
- Πράσινο $607 \leq X_{adc} \leq 660$
- Μπλε $343 \leq X_{adc} \leq 390$
- Κίτρινο $680 \leq X_{adc} \leq 725$

Τέλος συνδέσαμε στο PB3 ένα piezoelectric speaker/buzzer. Το συγκεκριμένο pin χρησιμοποιείται σαν έξοδος στο Pulse Width Modulation (PWM). Έτσι μεταβάλλοντας το πλάτος των παλμών που το piezo παίρνει στην είσοδό του, μεταβάλουμε και τη συχνότητα ταλάντωσής του και κατά συνέπεια τον ηχητικό τόνο.

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 ότι αν η τιμή του ADC βρίσκεται μέσα στα παραπάνω όρια που υπολογίσαμε τότε ενεργοποιούμε το αντίστοιχο PORT (PORTD1:4 στην πρώτη περίπτωση και PORTB1:4 στη δεύτερη) 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 και το PC0 για την τροφοδοσία των λευκών LED αντίστοιχα.

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

- Έπειτα 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	ADIF	ADIF	ADIF	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.

- Για την παραγωγή ηχητικών τόνων χρησιμοποιήσαμε το Fast PWM του Timer/Counter0. Έτσι, στον καταχωρητή TCCR0, θέσαμε σε άσσο τα bits WGM00 και WGM01 για να ενεργοποιήσουμε τη συγκεκριμένη λειτουργία, τα COM00 και COM01 για να μηδενίζουμε την έξοδό μας σε περίπτωση ισότητας του OCR0 με το μετρητή μας και το bit CS00 για να μην έχουμε prescaling.

8-bit Timer/Counter Register Description

Timer/Counter Control Register – TCCR0

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	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 3, 6 – WGM01:0: Waveform Generation Mode

These bits control the counting sequence of the counter, the source for the maximum (TOP) counter value, and what type of Waveform Generation to be used. Modes of operation supported by the Timer/Counter unit are: Normal mode, Clear Timer on Compare Match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes. See [Table 38](#) and "Modes of Operation" on page 76.

Table 38. Waveform Generation Mode Bit Description⁽¹⁾

Mode	WGM01 (CTC0)	WGM00 (PWM0)	Timer/Counter Mode of Operation	TOP	Update of OCR0	TOV0 Flag Set-on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

Note: 1. The CTC0 and PWM0 bit definition names are now obsolete. Use the WGM01:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

• Bit 5:4 – COM01:0: Compare Match Output Mode

These bits control the Output Compare pin (OC0) behavior. If one or both of the COM01:0 bits are set, the OC0 output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0 pin must be set in order to enable the output driver.

- **Bit 5:4 – COM01:0: Compare Match Output Mode**

These bits control the Output Compare pin (OC0) behavior. If one or both of the COM01:0 bits are set, the OC0 output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0 pin must be set in order to enable the output driver.



ATmega16(L)

When OC0 is connected to the pin, the function of the COM01:0 bits depends on the WGM01:0 bit setting. Table 39 shows the COM01:0 bit functionality when the WGM01:0 bits are set to a normal or CTC mode (non-PWM).

Table 39. Compare Output Mode, non-PWM Mode

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Toggle OC0 on compare match
1	0	Clear OC0 on compare match
1	1	Set OC0 on compare match

- Τέλος θέσαμε το pin PB3 ως έξοδο για να τους παλμούς μας και ενεργοποιήσαμε τα overflow interrupts του Timer/Counter0 θέτοντας σε άσσο το TOIE0.

Timer/Counter Interrupt Mask Register – TIMSK

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
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 1 – OCIE0: Timer/Counter0 Output Compare Match Interrupt Enable

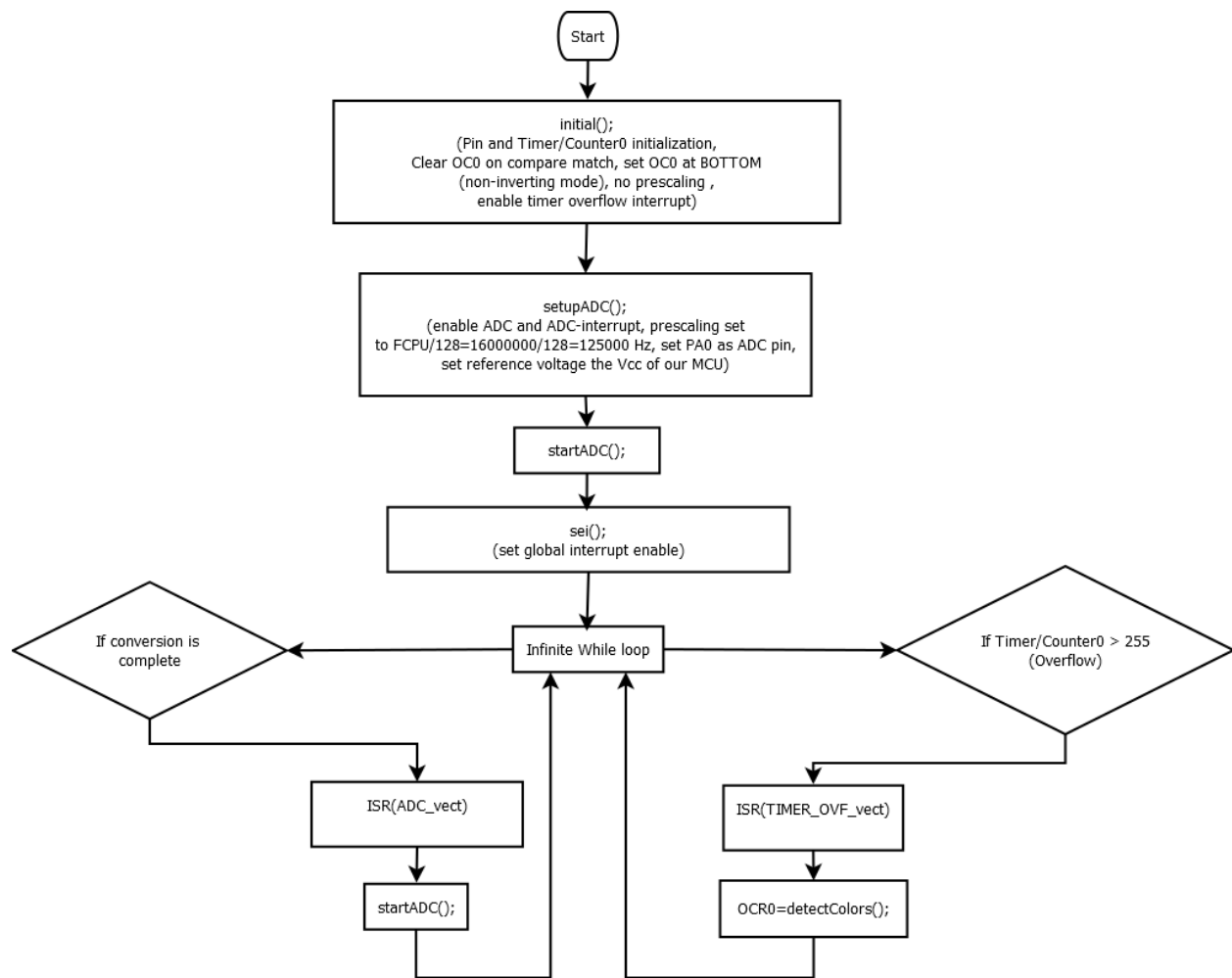
When the OCIE0 bit is written to one, and the I-bit in the Status Register is set (one), the Timer/Counter0 Compare Match interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter0 occurs, that is, when the OCF0 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.



• Bit 0 – TOIE0: Timer/Counter0 Overflow Interrupt Enable

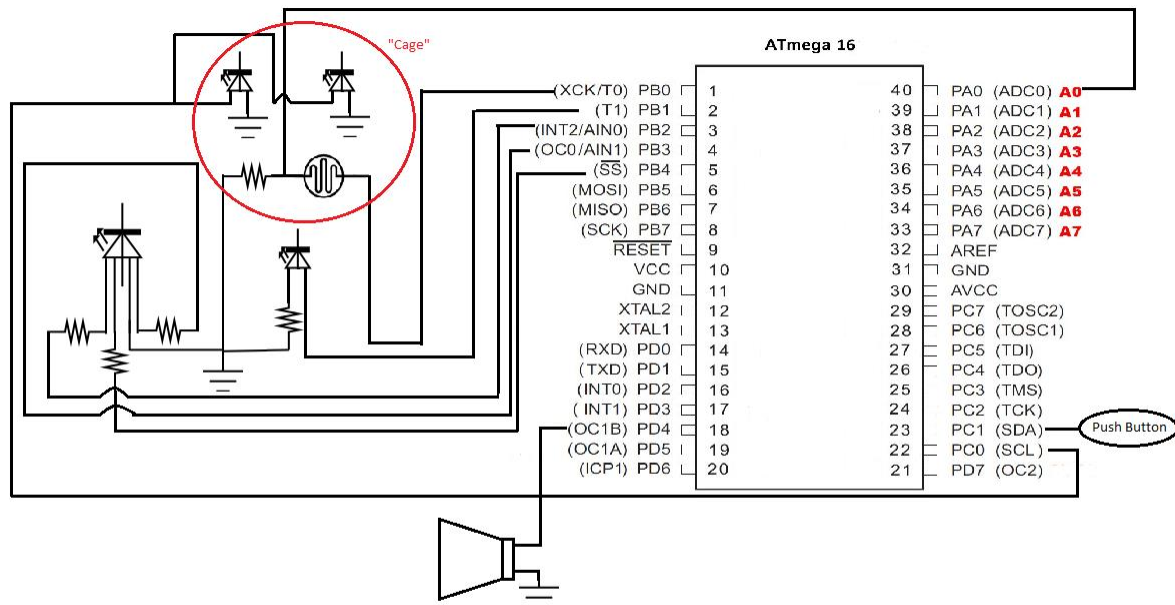
When the TOIE0 bit is written to one, and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs, that is, when the TOV0 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

Flowchart



Phase and Frequency Modulation

Το κύκλωμα για την υλοποίηση με Phase and Frequency Modulation φαίνεται στην εικόνα:



Η διαφοράς με την Fast PWN υλοποίηση είναι :

1. Χρησιμοποιούμε Phase and Frequency Modulation αντί για Fast PWN.
2. Χρησιμοποιούμε τον Timer/Counter 1 όπου είναι 16 bit , έτσι μας δίνει τη δυνατότητα να παράγουμε περισσότερους διαφορετικούς ήχους.
3. Προσθέσαμε Push Button (στο PC1) δίνοντας την ευχέρεια στο χρήστη να πατήσει ένα κουμπί για να ανιχνεύσει το χρώμα και αυτό έχει ως αποτέλεσμα να εξοικονομούμε ενέργεια αφού το σύστημα μας θα λειτουργεί μόνο όταν το κουμπί πατιέται.
4. Επιπλέον αλλάξαμε τη συνδεσμολογία ,δηλαδή βάλαμε τα RGB ,το κίτρινο LED και την τάση εισόδου στα PD0:4 και το piezzo στο PD4.

Αναλυτική περιγραφή

- Η μόνη μάσκα που αλλάξαμε ήταν αυτές όπου σχετίζονταν με τον Timer/Counter. Πιο συγκεκριμένα αντικαταστήσαμε τον Timer/Counter0 με τον Timer/Counter1 και ενεργοποιήσαμε από τον καταχωρητή TCCR1A το COM1B1 ώστε να μηδενίσουμε τους μετρητές OC1A και OC1B.

**Timer/Counter1
Control Register A –
TCCR1A**

Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	W	W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• **Bit 7:6 – COM1A1:0: Compare Output Mode for Channel A**

• **Bit 5:4 – COM1B1:0: Compare Output Mode for Channel B**

The COM1A1:0 and COM1B1:0 control the Output Compare pins (OC1A and OC1B respectively) behavior. If one or both of the COM1A1:0 bits are written to one, the OC1A output overrides the normal port functionality of the I/O pin it is connected to. If one or both of the COM1B1:0 bit are written to one, the OC1B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the *Data Direction Register* (DDR) bit corresponding to the OC1A or OC1B pin must be set in order to enable the output driver.

When the OC1A or OC1B is connected to the pin, the function of the COM1x1:0 bits is dependent of the WGM13:0 bits setting. [Table 44](#) shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to a normal or a CTC mode (non-PWM).

Table 44. Compare Output Mode, non-PWM

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	Toggle OC1A/OC1B on compare match
1	0	Clear OC1A/OC1B on compare match (Set output to low level)
1	1	Set OC1A/OC1B on compare match (Set output to high level)

[Table 45](#) shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to the fast PWM mode.

- Έπειτα από τον καταχωρητή TCCR1B ενεργοποιήσαμε το WGM13 και το CS10 έτσι ώστε να ορίσουμε το mode να είναι phase and frequency modulation καθώς και το ρολόι να είναι non-prescaling.

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1X	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	Reserved	–	–	–
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}/1$ (No prescaling)
0	1	0	$clk_{I/O}/8$ (From prescaler)
0	1	1	$clk_{I/O}/64$ (From prescaler)
1	0	0	$clk_{I/O}/256$ (From prescaler)
1	0	1	$clk_{I/O}/1024$ (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

- Στη συνέχεια , από τον καταχωρητή TIMSK θέσαμε άσσο το TOIE1 με αποτέλεσμα να ενεργοποιηθεί το overflow interrupt σε περίπτωση που το TOV1 είναι set στο 0.

Timer/Counter Interrupt Mask Register – TIMSK⁽¹⁾

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
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	

Note: 1. This register contains interrupt control bits for several Timer/Counters, but only Timer1 bits are described in this section. The remaining bits are described in their respective timer sections.

• Bit 5 – TICIE1: Timer/Counter1, Input Capture Interrupt Enable

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Input Capture Interrupt is enabled. The corresponding Interrupt Vector (See "Interrupts" on page 45.) is executed when the ICF1 Flag, located in TIFR, is set.

• Bit 4 – OCIE1A: Timer/Counter1, Output Compare A Match Interrupt Enable

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare A match interrupt is enabled. The corresponding Interrupt Vector (See "Interrupts" on page 45.) is executed when the OCF1A Flag, located in TIFR, is set.

• Bit 3 – OCIE1B: Timer/Counter1, Output Compare B Match Interrupt Enable

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare B match interrupt is enabled. The corresponding Interrupt Vector (See "Interrupts" on page 45.) is executed when the OCF1B Flag, located in TIFR, is set.

• Bit 2 – TOIE1: Timer/Counter1, Overflow Interrupt Enable

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Overflow Interrupt is enabled. The corresponding Interrupt Vector (See "Interrupts" on page 45.) is executed when the TOV1 Flag, located in TIFR, is set.

- Τέλος θέσαμε από τον καταχωρητή TIFR το TOV1 μηδέν.

Timer/Counter Interrupt Flag Register – TIFR

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
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	

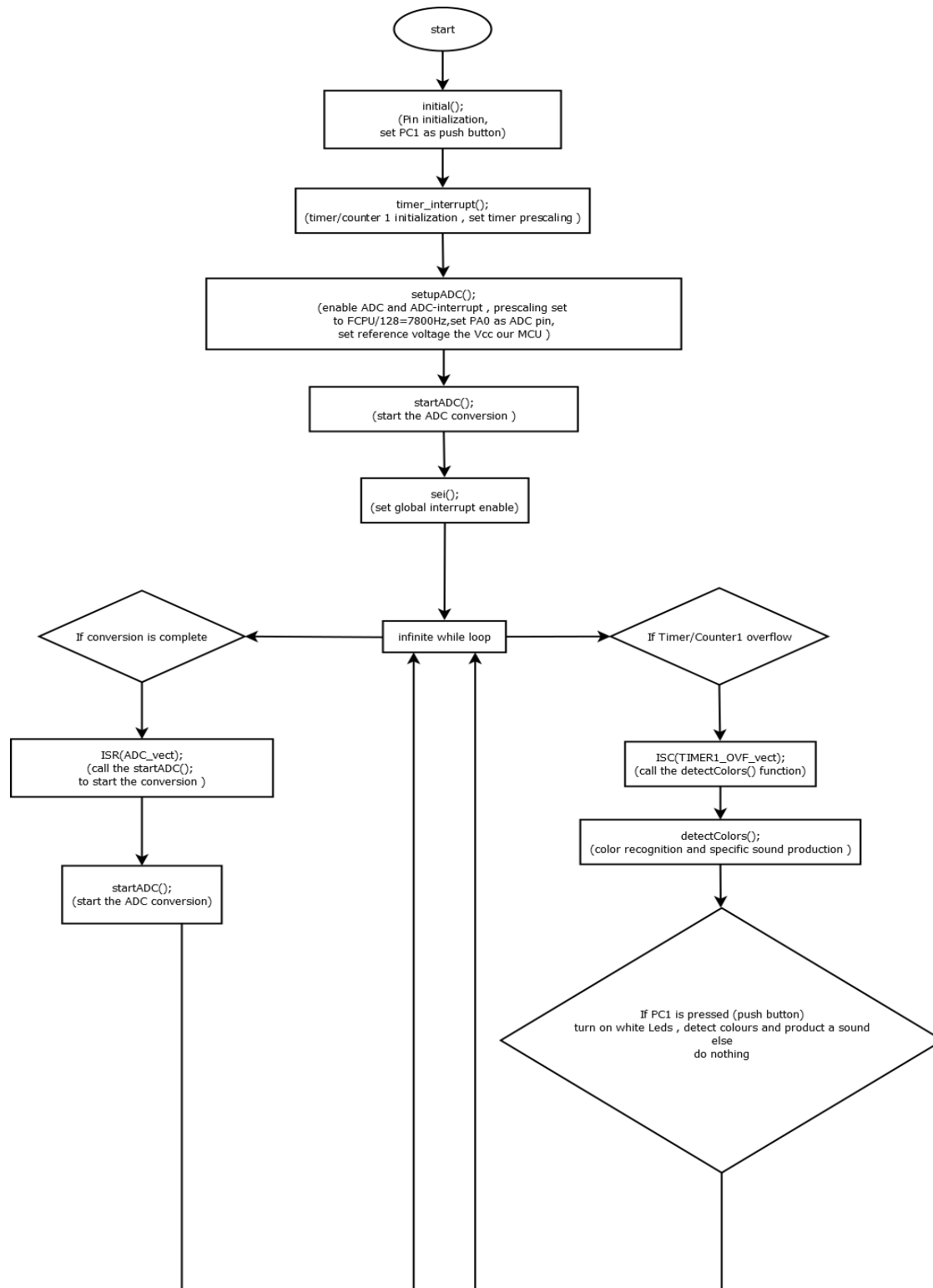
Note: This register contains flag bits for several Timer/Counters, but only Timer1 bits are described in this section. The remaining bits are described in their respective timer sections.

• Bit 2 – TOV1: Timer/Counter1, Overflow Flag

The setting of this flag is dependent of the WGM13:0 bits setting. In normal and CTC modes, the TOV1 Flag is set when the timer overflows. Refer to Table 47 on page 112 for the TOV1 Flag behavior when using another WGM13:0 bit setting.

TOV1 is automatically cleared when the Timer/Counter1 Overflow interrupt vector is executed. Alternatively, TOV1 can be cleared by writing a logic one to its bit location.

Flowchart:



Βιβλιοθήκες :

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

Πηγές / βιβλιογραφία :

- [Ομιλία του Neil Harbisson στο TEDGlobal 2012](#)
- [Η ιδέα για τον αισθητήρα βασίστηκε σε αυτήν την υλοποίηση](#)
- [Τα βασικά για τη χρήση του ADC](#)
- [ATMega16 datasheet](#)