Σύντομη Αναφορά

Σκοπός του Project μας είναι η δημιουργία ενός συστήματος που θα υποδεικνύει στον χρήστη τα χρώματα μέσω ηχητικών τόνων. Η ιδέα βασίστηκε στην ομιλία του Neil Harbisson στο TEDGlobal 2012 και η υλοποίηση θα γίνει μέσω :

- AVR microcontroller
- Atmel studio 7 ως προγραμματιστικό περιβάλλον του AVR
- Αισθητήρα χρωμάτων βασισμένο σε LDR αντίσταση

Πιο συγκεκριμένα θα εκμεταλευτούμε την ιδιότητα της LDR να αλλάζει αντίσταση ανάλογα με την ένταση του φωτός που πέφτει πάνω της. Έτσι παίρνοντας διαφορετικές τιμές τάσης για το κάθε χρώμα στην έξοδο του αισθητήρα μας θα είμαστε σε θέση να το αναγνωρίσουμε. Στη συνέχεια θα χρησιμοποιήσουμε τις τιμές των τάσεων που λαμβάνουμε ώστε να παράξουμε τους ηχιτικούς τόνους.

Βιβλιογραφία:

- Ομιλία του Neil Harbisson στο TEDGlobal 2012
- Αντιστοίχιση τάσεων σε ήχους
- Δημιουργία κυκλώματος αισθητήρα

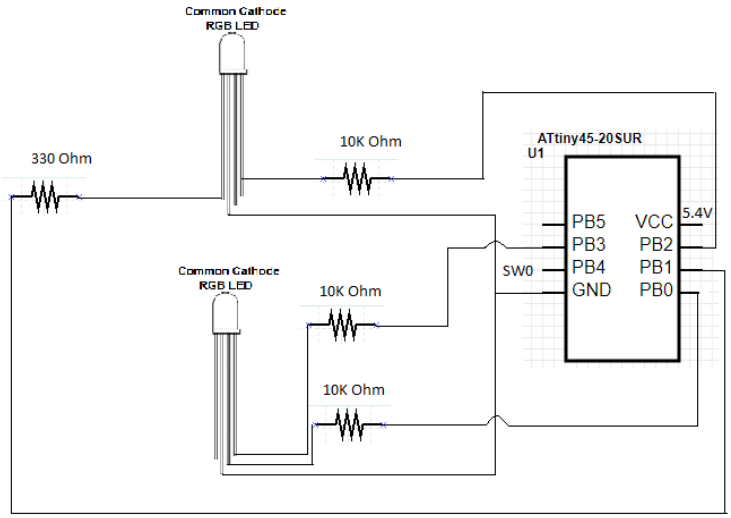
Milestone 1

Για το πρώτο Milestone συναρμολογήσαμε και ελέγξαμε το κύκλωμα του αισθητήρα με κάποια βασικά χρώματα. Οι μετρήσεις που πήραμε για το καθε χρώμα είναι οι παρακάτω:

- Μπλε - 4.73 V
- Κόκκινο - 4.61 V
- Πράσινο - 4.43 V
- Κίτρινο - 4.27 V
- Λευκό LED φλας - 3.7 V
- Σκοτάδι - 5.23 V (Τάση πηγής)

Επίσης υλοποιήσαμε κάποια μικρά, βασικά πρότζεκτ, στην προσπάθεια μας να εξοικειωθούμε με το Atmel Studio και τις δυνατότητες του. Αξίζει να σημειωθεί πως λόγω δυσκολιών που αντιμετωπίσαμε με τα 40πινα chip που μας δόθηκαν, δουλέψαμε με έναν 8πινο ATtiny85 που βάλαμε στο STK 500 που πήραμε από το εργαστήριο. Εστιάσαμε στα παρακάτω:

- Απλή χρήση ενός LED ως output στα πρώτα βήματα (on, off, blinking)
- Interrupts
- Timers/Counters
- Debugging/Simulation



Αφού φτιάξαμε το παραπάνω κύκλωμα καταφέραμε να ενεργοποιούμε και να απενεργοποιούμε το ένα LED μετά το άλλο, μέχρι να ανάψουν και να σβήσουν όλα τα LED από μία φορά. Αυτό το υλοποιήσαμε μέσω time counter. Πιο συγκεκριμένα ενεργοποιήσαμε το CTC1 που βρίσκεται στο 7ο bit του TCCR1 για να μηδενίσουμε τον μετρητή. Έπειτα ενεργοποιήσαμε τα CS13, CS12, CS11, CS10 του TCCR1 bits 3, 2, 1, 0 αντίστοιχα και βάλαμε το OCRC1=61 για να κρατάμε τα LED 1 δευτερόλεπτο αναμένα (προκύπτει από 10^(-6)*16384*61 περίπου ίσο με 1sec με F_CPU 1MHz). Δηλώσαμε μια μεταβλητή, την αρχικοποιήσαμε σε 1 και κάναμε αριστερή ολίσθηση (πολλαπλασμός με το 2) και έτσι στα PORTB βγαίνουν οι έξοδοι 1, 2, 4, 8 στο δυαδικό σύστημα. Τέλος υλοποιήσαμε ένα PCIE interrupt το οποίο ενεργοποιείται με τη μεταβολή του SW0 και αλλάζει την κατάσταση του μπλε LED (PB0) ενώ τα υπόλοιπα συνεχίζουν χωρίς αλλαγή στη ροή τους.

Αναλυτική Περιγραφή:

- Από τη μάσκα DDRB ενεργοποιήσαμε τα bit 0:3 ώστε τα PIN5, PIN6, PIN7 και PIN2 στο ATTINY να γίνουν έξοδοι.

10.4.2 PORTB – Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x18	–	–	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

10.4.3 DDRB – Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x17	–	–	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

10.4.4 PINB – Port B Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x16	–	–	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	N/A	N/A	N/A	N/A	N/A	N/A	

- Έχουμε ορίσει το ρολόι μας να έιναι ***1 MHz***. Έτσι θέσαμε το ρολόι του timer/counter1 με prescaler 16384 (άρα το ρολόι του είναι περίπου 61,03 αν κάνουμε τη διαίρεση) συνεπώς εκχωρήσαμε στο OCR1C την τιμή 61 με σκοπό να αναβοσβήνουν τα LED ανά 1 δευτερόλεπτο ($16384*61 \approx 10^6$). Για να το πετύχουμε αυτό χρησιμοποιήσαμε τις παρακάτω μάσκες και ενεργοποιήσαμε τα κατάλληλα σήματα.
- Από τη μάσκα TCCR1 ενεργοποιήσαμε το CTC1 ώστε ο μετρητής να μηδενίζεται κάθε φορά που γίνεται ίσος με την τιμή του OCR1C.

12.3.1 TCCR1 – Timer/Counter1 Control Register

Bit	7	6	5	4	3	2	1	0	
0x30	CTC1	PWM1A	COM1A1	COM1A0	CS13	CS12	CS11	CS10	TCCR1
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	

12.3.6 OCR1C – Timer/Counter1 Output Compare RegisterC

Bit	7	6	5	4	3	2	1	0	
0x2D	MSB							LSB	OCR1C
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	1	1	1	1	1	1	1	1	

The output compare register C is an 8-bit read/write register.

The Timer/Counter Output Compare Register C contains data to be continuously compared with Timer/Counter1. A compare match does only occur if Timer/Counter1 counts to the OCR1C value. A software write that sets TCNT1 and OCR1C to the same value does not generate a compare match. If the CTC1 bit in TCCR1 is set, a compare match will clear TCNT1.

This register has the same function in normal mode and PWM mode.

Έπειτα ενεργοποιήσαμε τα CS13 , CS12 , CS11 και CS10 με σκοπό να θέσουμε το prescaling στο 16384.

• **Bits 3:0 - CS1[3:0]: Clock Select Bits 3, 2, 1, and 0**
The Clock Select bits 3, 2, 1, and 0 define the prescaling source of Timer/Counter1.

Table 12-5. Timer/Counter1 Prescale Select

CS13	CS12	CS11	CS10	Asynchronous Clocking Mode	Synchronous Clocking Mode
0	0	0	0	T/C1 stopped	T/C1 stopped
0	0	0	1	PCK	CK
0	0	1	0	PCK/2	CK/2
0	0	1	1	PCK/4	CK/4
0	1	0	0	PCK/8	CK/8
0	1	0	1	PCK/16	CK/16
0	1	1	0	PCK/32	CK/32

Table 12-5. Timer/Counter1 Prescale Select (Continued)

CS13	CS12	CS11	CS10	Asynchronous Clocking Mode	Synchronous Clocking Mode
0	1	1	1	PCK/64	CK/64
1	0	0	0	PCK/128	CK/128
1	0	0	1	PCK/256	CK/256
1	0	1	0	PCK/512	CK/512
1	0	1	1	PCK/1024	CK/1024
1	1	0	0	PCK/2048	CK/2048
1	1	0	1	PCK/4096	CK/4096
1	1	1	0	PCK/8192	CK/8192
1	1	1	1	PCK/16384	CK/16384

The Stop condition provides a Timer Enable/Disable function.

- Στη συνέχεια κάναμε άσσο το OCIE1A από τη μάσκα TIMSK για να ενεργοποιήσουμε το Interrupt σε περίπτωση που η τιμή του μετρητή και του OCRC1 γίνουν ίσες.

12.3.7 TIMSK – Timer/Counter Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x39	–	OCIE1A	OCIE1B	OCIE0A	OCIE0B	TOIE1	TOIE0	–	TIMSK
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R	
Initial value	0	0	0	0	0	0	0	0	

• **Bit 7 – Res: Reserved Bit**
This bit is a reserved bit in the ATtiny25/45/85 and always reads as zero.

• **Bit 6 – OCIE1A: Timer/Counter1 Output Compare Interrupt Enable**
When the OCIE1A bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Compare MatchA, interrupt is enabled. The corresponding interrupt at vector $003 is executed if a compare matchA occurs. The Compare Flag in Timer/Counter1 is set (one) in the Timer/Counter Interrupt Flag Register.

- Για το switch button interrupt που υλοποιήσαμε χρησιμοποιήσαμε τη μάσκα GIMSK και ενεργοποιήσαμε το PCIE ώστε να ενεργοποποιηθούν τα Interrupts σε περίπτωση αλλαγής της λογικής κατάστασης των PIN.

9.3.2 GIMSK – General Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x3B	–	INT0	PCIE	–	–	–	–	–	GIMSK
Read/Write	R	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 5 – PCIE: Pin Change Interrupt Enable**

When the PCIE bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt is enabled. Any change on any enabled PCINT[5:0] pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI Interrupt Vector. PCINT[5:0] pins are enabled individually by the PCMSK0 Register.

- Τέλος από τη μάσκα PCMSK ενεργοποιήσαμε το PCINT4 και συνδέσαμε το PB4 στο SW0 ώστε να το χρησιμοποιήσουμε ως push button.

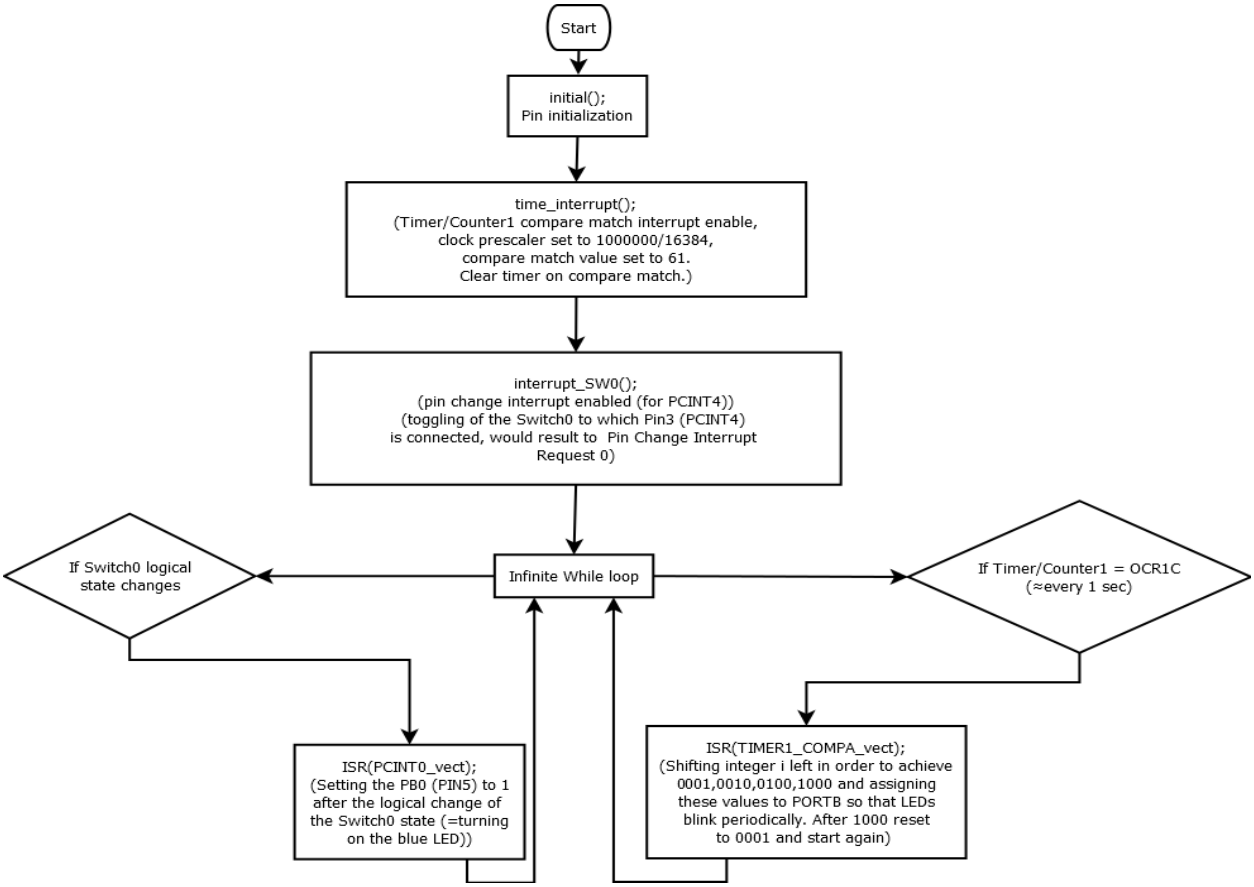
9.3.4 PCMSK – Pin Change Mask Register

Bit	7	6	5	4	3	2	1	0	
0x15	–	–	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 5:0 – PCINT[5:0]: Pin Change Enable Mask 5:0**

Each PCINT[5:0] bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT[5:0] is set and the PCIE bit in GIMSK is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT[5:0] is cleared, pin change interrupt on the corresponding I/O pin is disabled.

Flowchart:



Βιβλιοθήκες:

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

Βιβλιογραφία:

- Debugging/Simulation

- Timers/Counters/Interrupts link1, link2.