

## 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 ηχητικούς τόνους.

### Βιβλιογραφία:

- [Ομιλία](#) του Neil Harbisson στο TEDGlobal 2012
- [Αντιστοίχιση τάσεων σε ήχους](#)
- [Δημιουργία κυκλώματος αισθητήρα](#)

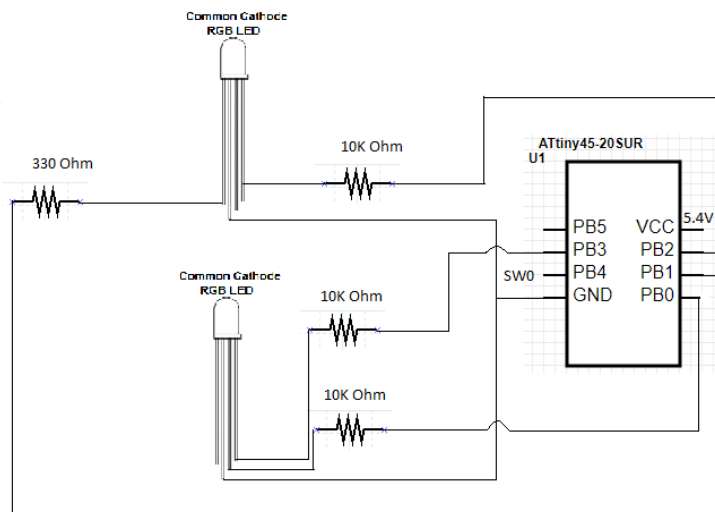
## Milestone 1

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 40πινια chip που μας δόθηκαν, δουλέψαμε με έναν 8πिनo 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  $10^{(-6)} \cdot 16384 \cdot 61$  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.

### **Βιβλιογραφία:**

- [Debugging/Simulation](#)
- Timers/Counters/Interrupts [link1](#), [link2](#).