REPORT

Εργαστήριο Μικροεπεξεργαστές και Περιφερειακά

Γκούμα Βασιλική 9755

ΚΩΣΤΑΣ ΑΝΔΡΟΝΙΚΟΣ 9754

Περιεχόμενα

[Εισαγωγή 2](#_Toc136011564)

[Υλοποίηση συνάρτησης *start\_signal()* 2](#_Toc136011565)

[Υλοποίηση συνάρτησης *check\_response()* 3](#_Toc136011566)

[Υλοποίηση συνάρτησης *readBit()* 4](#_Toc136011567)

[Υλοποίηση συνάρτησης *readByte()* 4](#_Toc136011568)

[Υλοποίηση συνάρτησης *get\_packet()* 5](#_Toc136011569)

[Υλοποίηση συνάρτησης *check\_sum*() 5](#_Toc136011570)

[Δηλώσεις και Αρχικοποιήσεις Μεταβλητών 6](#_Toc136011571)

[UART Communication 6](#_Toc136011572)

[Timer Initialization and Logic 8](#_Toc136011573)

[Υλοποίηση συνάρτησης *timer\_ISR()* 9](#_Toc136011574)

[Υλοποίηση συναρτήσεων για το control των LED 10](#_Toc136011575)

[Υλοποίηση της *BUTTON\_ISR()* 10](#_Toc136011576)

# Εισαγωγή

Στο συγκεκριμένο παραδοτέο χρησιμοποιούμε τον αισθητήρα DHT11 (αισθητήρας θερμοκρασίας και υγρασίας) για να μετρήσουμε την θερμοκρασία με χρήση ISR καθώς και να τυπώνουμε την θερμοκρασία με διαφορετική περίοδο από την περίοδο που διαβάζουμε τον αισθητήρα. Για τη χρήση του συγκεκριμένου αισθητήρα χρειάστηκε η υλοποίηση των αντίστοιχων drivers. Αρχικά θα περιγραφούν οι συναρτήσεις που είναι απαραίτητες για να λάβουμε δεδομένα από τον αισθητήρα και έπειτα η λογική του κύριου προγράμματος του εργαστηρίου

**DHT DRIVERS**

# Υλοποίηση συνάρτησης *start\_signal()*

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, οθόνη, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματα Η επικοινωνία μεταξύ του μικροελεγκτή και του αισθητήρα DHT11 ξεκινάει όταν θέλουμε να λάβουμε δεδομένα από τον αισθητήρα. Για να επιτευχθεί αυτό πρέπει να ενημερώσουμε τον αισθητήρα το οποίο σύμφωνα με το datasheet πρέπει να τεθεί το voltage level από HIGH σε LOW για 20ms της γραμμής μεταφοράς δεδομένων. Έπειτα τον θέτουμε σε ΗΙGH και περιμένουμε 30us (μεταξύ 20 και 40us) για την απάντηση του.

# Υλοποίηση συνάρτησης *check\_response()*

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμμή, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματαΕικόνα που περιέχει κείμενο, ηλεκτρονικές συσκευές, στιγμιότυπο οθόνης, λογισμικό

Περιγραφή που δημιουργήθηκε αυτόματαΌταν ο DHT ανιχνεύσει το σήμα εισόδου στέλνει ένα σήμα χαμηλής τάσης για περίπου 80 us και μετά ένα σήμα υψηλής τάσης για 80us όπως φαίνεται στην Εικόνα 1. Στη συγκεκριμένη συνάρτηση εξετάζουμε αν αυτό επιτελείται.

Εικόνα 1 : σωστή ανταπόκριση του DHT

# Υλοποίηση συνάρτησης *readBit()*

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματαΕφόσον όλα λειτουργούν σωστά τώρα μπορούμε να διαβάσουμε δεδομένα. Πριν ο αισθητήρας από κάθε bit πληροφορίας, στέλνει το 0 για 50us ως start bit . Έπειτα έχουμε υψηλή τάση όπου αν είναι για διάρκεια 70 us παίρν1 ενώ αν είναι μεταξύ 26 - 28us παίρνουμε 0,όπως φαίνεται και στις παρακάτω εικόνες .Στην συγκεκριμένη συνάρτηση περιμένουμε 40us και ανάλογα με το αν έχουμε υψηλή ή χαμηλή τάση την στιγμή 40 us τότε έχουμε 1 ή 0 αντίστοιχα.

# Υλοποίηση συνάρτησης *readByte()*

Αποθηκεύονται σε μία μεταβλητή byte τα bit που διαβάσαμε με την συνάρτηση *readBit() .*

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά, αριθμός

Περιγραφή που δημιουργήθηκε αυτόματα

# Υλοποίηση συνάρτησης *get\_packet()*

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματαΑποθηκεύουμε σε έναν πίνακα τα byte με τα δεδομένα για θερμοκρασία και υγρασία που πήραμε από τον αισθητήρα.

# Υλοποίηση συνάρτησης *check\_sum*()

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματαEλέγχουμε ότι στάλθηκαν σωστά τα δεδομένα σύμφωνα με το datasheet.

**Main Program**

# Δηλώσεις και Αρχικοποιήσεις Μεταβλητών

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά, αριθμός

Περιγραφή που δημιουργήθηκε αυτόματα

# UART Communication

Αρχικοποιούμε την επικοινωνία στα 115200bps και θέτουμε τον επεξεργαστή να είναι σε θέση να ακούσει το interrupt από τoν UART transceiver μέσω της *uart\_set\_rx\_callback()*

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά, λευκό

Περιγραφή που δημιουργήθηκε αυτόματα

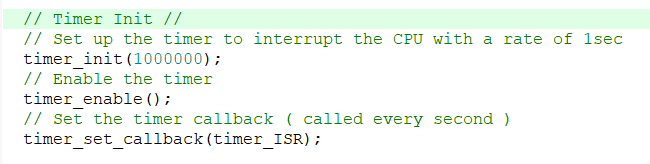
Aκολουθεί η ISR που καλείται κάθε φορά που λαμβάνεται από τον μικροελεγκτή ένα byte από τον χρήστη που στέλνει μέσω σειριακής κονσόλας . Αποθηκεύουμε τον κάθε χαρακτήρα σε έναν πίνακα AEM αφού ελέγξουμε ότι λαμβάνουμε από τον χρήστη αριθμούς. Μόλις πατηθεί το «enter» από τον χρήστη θα σταλεί το ‘10’, οπότε γνωρίζουμε ότι έχουμε λάβει σωστά όλο το ΑΕΜ. Υπολογίζουμε και το άθροισμα των ψηφίων καθώς θα χρειαστεί στην *BUTTON\_ISR()* όπως θα δούμε παρακάτω.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά, αριθμός

Περιγραφή που δημιουργήθηκε αυτόματα

# Timer Initialization and Logic

Στην εργασία πρέπει να χειριστούμε διαφορετικά events και όπου η εκτέλεση του αντίστοιχου κώδικα για το κάθε event πρέπει να εκτελείται με διαφορετική συχνότητα. Για παράδειγμα, η θερμοκρασία πρέπει να διαβάζεται ανά 2s, ενώ η τύπωση της δεν είναι κατά ανάγκη να συμβαίνει στον ίδιο χρόνο αλλά εξαρτάται από το AEM και το πάτημα του user\_button. Επειδή θα χρησιμοποιήσουμε έναν timer, τον αρχικοποιούμε να εκτελεί την ISR του ανά 1s και με κατάλληλους counters μπορούμε να ξέρουμε πότε μεσολάβησαν 2s ή όποια άλλη συχνότητα επιθυμούμε.

 Γνωρίζουμε ότι κάθε κύκλος που εκτελεί ο επεξεργαστής μια συγκεκριμένη εντολή επιτελείται με συχνότητα 16MHz (System CPU Clock). Για να περάσει 1s πρέπει να 6μεσολαβήσουν 16 \* 106  κύκλοι ρολογιού. Επιλέγουμε ως όρισμα το 106  το οποίο προκύπτει αν κατανοήσουμε την *timer\_init()* που φαίνεται παρακάτω. Τέλος, αρχικοποιούμε κα την callback *timer\_ISR().*

Εικόνα που περιέχει κείμενο, γραμματοσειρά, γραμμή, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

# Υλοποίηση συνάρτησης *timer\_ISR()*

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματαΕφόσον έχουμε αρχικοποιήσει τον timer να προκαλεί interrupt ανά δευτερόλεπτο και εμείς θέλουμε να ελέγχουμε τον ρυθμό της εκτύπωσης της θερμοκρασίας δημιουργήθηκε η μεταβλητή print\_timing\_counter. Αντίστοιχα, το διάβασμα του αισθητήρα γίνεται ανεξαρτήτως της τύπωσης ανά 2s μέσω της μεταβλητής temperature\_timing\_counter. Σημαντικό πριν τυπώσουμε την θερμοκρασία ή πριν τον έλεγχο των LED πρέπει να βεβαιωθούμε ότι το πακέτο που λάβαμε από τον αισθητήρα είναι σωστό. Αυτό ελέγχεται από την *check\_sum()* .

# Υλοποίηση συναρτήσεων για το control των LED

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματαΧρησιμοποιήθηκε το user\_led για την ενημέρωση του χρήστη σχετικά με την θερμοκρασία.

Εικόνα που περιέχει κείμενο, γραμματοσειρά, στιγμιότυπο οθόνης, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματα

# Υλοποίηση της *BUTTON\_ISR()*

Εδώ φαίνεται ότι το user button επηρεάζει την συχνότητα τύπωσης της θερμοκρασίας καθώς μεταβάλει την όριο PRINT\_FREQUENCY που ελέγχεται εντός της timer\_ISR.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά, αριθμός

Περιγραφή που δημιουργήθηκε αυτόματα