

# Μικροεπεξεργαστές και Περιφερειακά

## Αναφορά Εργαστηρίου Ομάδας 4:

Ονοματεπώνυμο:Φωτεινή Κοκκώνη ΑΕΜ:10852

Ονοματεπώνυμο: Μπίνος Γεώργιος ΑΕΜ:9333

## Πρώτο Σκέλος(Uart stuff):

Όπως και στο προηγούμενο εργαστήριο χρησιμοποιήσαμε την ίδια λογική για υλοποίηση της uart με μια Boolean μεταβλητή για έλεγχο του κωδικού{σωστός/λάθος} ώστε να υπάρχει Loop μέχρι να μπει ο σωστός κωδικός που θέσαμε {1234(Αυτό το αλλαξα όπως το ζητησατε και το εκανα να παιρνει a,b,c,d και status command)}.Επειτα επειδή ΔΕΝ κάνουμε store τον κωδικό χρησιμοποιούμε τον ΙΔΙΟ buffer για το AEM και την ίδια διαδικασία όμως κάνουμε store το AEM για χρήση αργότερα.Αργότερα με μια συνάρτηση στήσαμε το visualization του Menu όπως το όριζε οι οδηγίες της εκφώνησης.

## Δεύτερο Σκέλος(Sensor):

Μετά από πολλή μελέτη των datasheets για τον αισθητήρα, ψάχιμο online και πολλά tutorial + trials and error runs καταφέραμε να κάνουμε τον αισθητήρα να διαβάζει και να τυπώνει αποτελεσματικά.Θέσαμε 2 modes ΜΕΣΑ στην main μας,εχουμε το Mode A και όταν !=Mode A ενεργοποιείται το Mode B που είναι φτιαγμένο σε συνάρτηση.{Καλύτερη εξήγηση στα comments}

## Τρίτο Σκέλος(TickInterval,Synchronisation):

Μπορείτε να δείτε στα comments που κάναμε initialize τον timer + enable, τώρα στο read και print που είναι κοινό και στα 2 mode θέσαμε μια συνάρτηση όπου διαβάζει και τυπώνει κανονικά τις τιμές των display mode που έχουμε θέση {typedef enum για initialization και cases στον Handler}.Ο handler είναι η συνάρτηση όπου καλείται όποτε έχουμε input από την uart και αναλόγως το case κάνει αύξηση/μείωση του Interval ΠΟΥ είναι υπεύθυνο για την συχνότητα εκτύπωσης στην συνάρτηση read και print του αισθητήρα που αναφέραμε προηγουμένως.Επίσης αυτό κάνει και handle τα display modes αλλά και την εμφάνιση

των status{Υλοποιήση με switch κυρίως και στο readnprint και στον handler

### **Τέταρτο Σκέλος(Led Blink):**

Στο Mode B έχουμε μια Boolean μεταβλητή που είναι ορισμένη false οπού μέχρι οι μετρήσεις βγουν εκτός και αρχίσει να αναβοσβήνει το Led. Εκεί η Boolean γίνεται true και έπειτα όταν οι ενδείξεις επαναφερθούν σε χαμηλότερα επίπεδα από τα conditions για led blink τότε πρώτα περνάνε από ένα if που επιτρέπει το led να συνεχίσει να αναβοσβήνει μέχρι τις 5 πρώτες normal μετρήσεις μέσα από μία μεταβλητή times όπου αυξάνεται κάθε φορά που ο κώδικας μας «μπαίνει» σε εκείνο το if

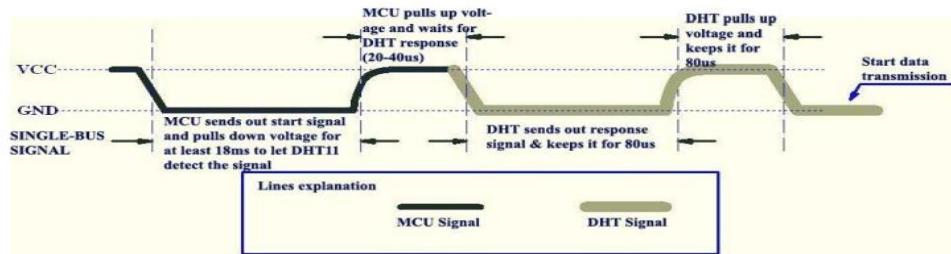
### **Πεμπτο Σκέλος(Button Interrupt Και Panic Reset):**

Το Button το ορίσαμε όπως και στο προηγούμενο εργαστήριο, βάλαμε μια μεταβλητή ώστε να μετράμε τα presses και να μπορέσουμε να κάνουμε %3 ώστε κάθε τρίτο πάτημα του κουμπιού το TickInterval μας να γίνεται όσο το άθροισμα των 2 τελευταίων ψηφίων του ΑΠΟΘΗΚΕΥΜΕΝΟΥ ΑΕΜ μας. Για το Panic Reset απλά βάλαμε στην sensor read n print μια συνθήκη που ελέγχει αν οι μετρήσεις είναι πάνω από τα «υπέρτατα» όρια και τότε μέσα από την μεταβλητή PanicCounter που αυξάνεται κάθε φορά που μπαίνουμε μέσα στο από εκεί αν φτάσουμε PanicCounter=3 τότε κάνουμε NVIC\_SystemReset();

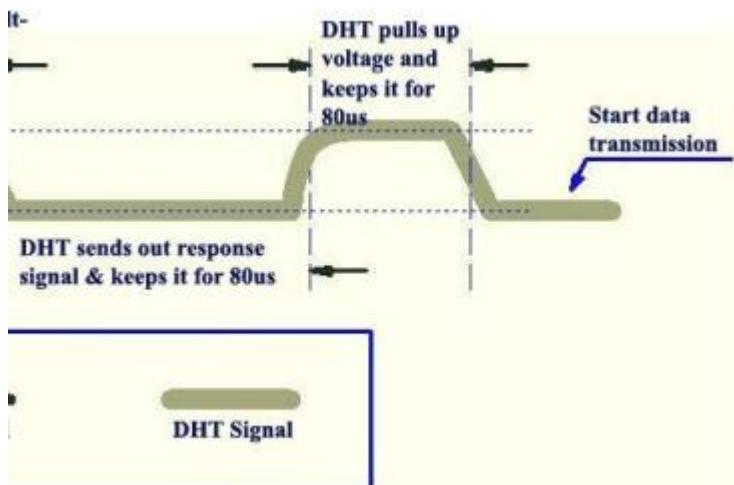
### **Έκτο Σκέλος(DHT11 Sensor):**

Ο αισθητήρας χρησιμοποιεί πρωτόκολλο μονής γραμμής (one-wire) και απαιτεί πολύ ακριβή χρονισμό για να πραγματοποιηθεί σωστά η επικοινωνία. {1} Στη συνάρτηση sensor\_start\_signal(), οδηγούμε τη γραμμή δεδομένων σε χαμηλή στάθμη (LOW) για τουλάχιστον 18ms και μετά την αφήνουμε σε υψηλή (HIGH) για περίπου 30μs. Θέλουμε το λιγότερο 18ms για να σιγουρευτούμε για το detection signal του DHT11 μετά HIGH για 20-40 ώστε να περιμένουμε για το response to DHT11. Αυτό είναι το σήμα εκκίνησης που απαιτείται για να ξεκινήσει η επικοινωνία με τον DHT11, όπως ορίζεται στο datasheet. {2} Στη συνέχεια, ο αισθητήρας απαντά με παλμούς LOW και HIGH των 80μs, κάτι που ελέγχουμε στη sensor\_check\_response() για να επιβεβαιώσουμε ότι ανταποκρίνεται. Στη συνέχεια, διαβάζουμε 5 bytes (40 bits) δεδομένων με τη συνάρτηση sensor\_read\_byte(), όπου κάθε bit αναγνωρίζεται με βάση τη διάρκεια του παλμού HIGH — σύντομο HIGH σημαίνει '0', μεγάλο HIGH σημαίνει '1'. Η συνάρτηση sensor\_read() διαχειρίζεται όλη τη διαδικασία και επαληθεύει τα δεδομένα με checksum

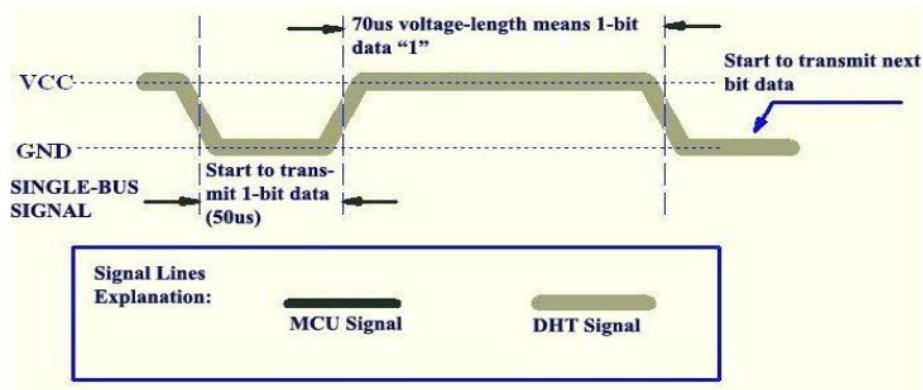
DHT11 begins, the programme of MCU will set Data Single bus voltage level from high to low and this process must take at least 18ms to ensure DHT's detection of MCU's signal, then MCU will pull up voltage and wait 20-40us for DHT's response.



{1}



{2}



{3}