Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Εργασία στο μάθημα
Μικροεπεξεργαστές και Περιφερειακά

των φοιτητών:

Νικόλαου Καγιάφα 8817

Αντώνιου Μυρσινιά 8873

Περιεχόμενα

Εισαγωγή

Στην παρακάτω αναφορά, παρουσιάζονται τα εξής στοιχεία που σχετίζονται με την εκπόνηση της προαιρετικής εργασίας με τον μικροελεγκτή ARDUINO UNO R3:

1. Παρουσίαση του προβλήματος και του εγχειρήματος που ζητήθηκε να υλοποιηθεί.
2. Περιγραφή της ροής του κώδικα και ανάλυση των αποφάσεων σχεδίασης που ελήφθησαν.
3. Περιγραφή των προβλημάτων που αντιμετωπίστηκαν κατά την ανάπτυξη του κώδικα ή και την σύνδεση των περιφερειακών.
4. Μετρήσεις και ανάλυσή τους που “αποδεικνύουν” ότι το σύστημα λειτουργεί σωστά.
5. Παρουσίαση του σχηματικού διαγράμματος που απεικονίζει το κύκλωμα και τον τρόπο διασύνδεσης των επιμέρους λειτουργικών διατάξεων.

1. Περιγραφή του προβλήματος

Το εγχείρημα που ζητήθηκε να υλοποιηθεί σχετίζεται με την δημιουργία ενός έξυπνου sθερμομέτρου-θερμοστάτη με την βοήθεια ενός μικροελεγκτή ARDUINO. Το θερμόμετρο αυτό θα έχει την δυνατότητα να συλλέγει δεδομένα σχετικά με την θερμοκρασία του

περιβάλλοντος στο οποίο βρίσκεται και να τα απεικονίζει σε μία οθόνη LCD. Ταυτόχρονα, θα δύναται να ελέγχει εάν η θερμοκρασία ξεπερνά ένα συγκεκριμένο κάτω ή άνω όριο εμφανίζοντας παράλληλα αντίστοιχο μήνυμα στην οθόνη LCD για την ενημέρωση του χρήστη αλλά και ανάβοντας τα κατάλληλα LEDs. Πιο αναλυτικά, ο αισθητήρας θερμοκρασίας θα λαμβάνει ερεθίσματα από το περιβάλλον του με περίοδο 5 δευτερολέπτων, θα υπολογίζει την μέση τιμή αυτών των δειγμάτων σε διάρκεια 2 λεπτών και θα την εμφανίζει για 10 δευτερόλεπτα στην οθόνη. Τέλος, ο μικροελεγκτής θα πρέπει μέσω ενός αισθητήρα εγγύτητας να ανιχνεύει εάν κάποιος προσπαθεί να προσεγγίσει την συσκευή και σε αυτήν την περίπτωση να τού παρουσιάζει την πιο πρόσφατη τιμή της θερμοκρασίας καθώς και την μέση τιμή των θερμοκρασιών που σημειώθηκαν κατά την διάρκεια των 2 προηγούμενων λεπτών.

2. Ανάλυση ροής και αποφάσεων σχεδίασης

Αρχικά, προτού εκκινήσει η κύρια λειτουργία του προγράμματος, γίνεται η απαραίτητη αρχικοποίηση των καθολικών(global) μεταβλητών, οι οποίες θα χρησιμοποιηθούν καθ' όλην την διάρκεια του προγράμματος. Για την ορθή λειτουργία του κώδικα υλοποιήθηκαν οι παρακάτω συναρτήσεις:

void setup(void)

Σε αυτήν την συνάρτηση, αρχικά, απενεργοποιούνται όλες οι διακοπές και στην συνέχεια τίθεται ο 16-bit timer1 που χρησιμοποιείται, στην κατάλληλη κατάσταση λειτουργίας του ως εξής:

```
TCCR1A = 0;
TCCR1B = 0;
TCCR1B |= (1<<CS12);  // Set the prescaler to 256.
TIMSK1 |= (1<<TOIE1);  // Enable timer1 overflow interrupt.
TCNT1 = Timer_initVal; // Start counting from 3035, until the timer overflows.
```

Γίνεται χρήση του συγκεκριμένου 16-bit timer για την μέτρηση ενός δευτερολέπτου, με την επιλογή της μικρότερης δυνατής τιμής του prescaler για την επίτευξη της μεγαλύτερης δυνατής ακρίβειας. Πιο συγκεκριμένα, στις τρεις πρώτες εντολές τίθεται ο timer1 στο κατάλληλο mode, ενώ επιλέγεται η τιμή του prescaler να είναι ίση με 256. Στην επόμενη εντολή, γίνεται ενεργοποίηση της δυνατότητας διακοπής ύστερα από υπερχείλιση(overflow) του καταχωρητή TCNT1. Στην τελευταία εντολή, ο TCNT1 αρχίζει να αυξάνεται ξεκινώντας από την τιμή 3035(2^16-(16*10^6)/256-1).
Στην συνέχεια, ορίζονται τα pins εισόδου-εξόδου του μικροελεγκτή, εκκινείται η διαδικασία συλλογής μετρήσεων από τον αισθητήρα θερμοκρασίας, λαμβάνεται και αποθηκεύεται η πρώτη μέτρηση και γίνεται ο έλεγχος για το αν αυτή ξεπερνά το κατώτερο ή το ανώτερο όριο θερμοκρασίας. Αναλόγως, ανάβουν τα αντίστοιχα LEDs. Τέλος, αρχικοποιείται η οθόνη LCD και ενεργοποιούνται όλες οι διακοπές.

void loop(void)

Η συνάρτηση αυτή συνιστά την κύρια ρουτίνα του προγράμματος και εκτελείται επαναλαμβανόμενα. Ανά μισό δευτερόλεπτο, καλεί την συνάρτηση checkProx(), για να ελεγχθεί εάν κάποιος έχει πλησιάσει την συσκευή(η λειτουργία της θα αναλυθεί εκτενώς στην συνέχεια). Αυτό το μισό δευτερόλεπτο μετριέται με την βοήθεια της συνάρτησης millis(), η οποία χρησιμοποιεί τον 8-bit timer0. Στην συνέχεια, εάν το σύστημα δεν βρίσκεται στην κατάσταση κατά την οποία εμφανίζονται τα αποτελέσματα στην οθόνη μετά το πέρας των δύο λεπτών(τιμή της powerLCD), ελέγχεται εάν βρίσκεται κάποιος κοντά στην συσκευή(τιμή της μεταβλητής powerLCDProx). Αν ναι, τότε καλείται η συνάρτηση LCDWhenClose()(η λειτουργία της θα αναλυθεί εκτενώς στην συνέχεια). Αν όχι, τότε καλείται η συνάρτηση tempLowHigh()(η λειτουργία της θα αναλυθεί εκτενώς στην συνέχεια).

void tempLowHigh()

Η συνάρτηση αυτή ελέγχει αν έχουν ξεπεραστεί οι οριακές τιμές θερμοκρασίας και αναλόγως τυπώνει στην οθόνη LCD τις κατάλληλες ενδείξεις. Κατ' αρχάς, εάν είναι η πρώτη φορά που ανιχνεύεται χαμηλή ή υψηλή θερμοκρασία, τότε γίνεται καθαρισμός(clear) της οθόνης από την προηγούμενη κατάστασή της. Στη συνέχεια, γίνεται παρόμοια ο έλεγχος για το αν η συνάρτηση καλείται για πρώτη φορά μετά την απομάκρυνση του χρήστη από την συσκευή. Στην περίπτωση αυτή, πραγματοποιείται καθαρισμός(clear) της οθόνης. Έπειτα, τίθεται ο κέρσορας στην κάτω γραμμή της οθόνης και διενεργείται έλεγχος για το αν η τελευταία τιμή της θερμοκρασίας(lastTempVal) έχει υπερβεί τα δοσμένα όρια. Αν ναι, τυπώνεται στην οθόνη η κατάλληλη ένδειξη, διαφορετικά γίνεται καθαρισμός και σβήσιμο της οθόνης.

```
if (situation == false)
{
lcd.clear();
}
if (situation2 == 1)
{
lcd.clear();
}
lcd.setCursor(0, 1);
if(lastTempVal >= 33.5)
{
lcd.backlight();
lcd.print("High Temp");
situation = true;
situation2 = 2;
}
```

Όπως φαίνεται παραπάνω, η μεταβλητή situation χρησιμοποιείται, για να γίνει καθαρισμός της οθόνης μόνο την πρώτη φορά που ανιχνεύεται τιμή της θερμοκρασίας που βρίσκεται εκτός των προκαθορισμένων ορίων(τιμή false της μεταβλητής). Η μεταβλητή αυτή γίνεται true όταν τυπωθεί για πρώτη φορά στην οθόνη η κατάλληλη ένδειξη, οπότε όταν ξαναεκτελεστεί μετά

από μισό δευτερόλεπτο δεν θα ξαναγίνει καθαρισμός της οθόνης, με αποτέλεσμα αυτή να "τρεμοπαίζει", γεγονός που συμβαίνει αν εκτελούνται συνεχόμενοι καθαρισμοί(clear). Επιπλέον, η μεταβλητή situation2 χρησιμοποιείται, για να γίνει καθαρισμός της οθόνης έπειτα από την πρώτη φορά που ανιχνεύεται η απομάκρυνση του χρήστη από την συσκευή(τιμή 1 της μεταβλητής). Η μεταβλητή αυτή γίνεται ίση με 2, εάν τυπωθεί για πρώτη φορά μετά την απομάκρυνση του χρήστη η ένδειξη για υψηλή ή χαμηλή θερμοκρασία(σε περίπτωση που αυτή σημειώνεται).

void LCDWhenClose()

Η συνάρτηση αυτή εκτελείται, όταν κάποιος χρήστης έχει πλησιάσει την συσκευή(στην συγκεκριμένη υλοποίηση σε απόσταση μικρότερη από 15cm). Κατ' αρχάς, εντείνεται η φωτεινότητα της οθόνης LCD και ο κέρσορας τοποθετείται στην άνω γραμμή στην αριστερή πλευρά της οθόνης. Στην συνέχεια, τυπώνεται η τιμή της μέσης τιμής που αφορά τα δείγματα των προηγούμενων 2 λεπτών. Η μέση τιμή είναι αποθηκευμένη στην μεταβλητή lastMeanVal. Αμέσως μετά, αφού ο κέρσορας της οθόνης τοποθετηθεί στην κάτω γραμμή στην αριστερή της πλευρά, τυπώνεται η τελευταία τιμή της θερμοκρασίας που καταγράφηκε, τιμή η οποία είναι αποθηκευμένη στην μεταβλητή lastTempVal.

void checkProx()

Η συνάρτηση αυτή χρησιμοποιείται για να ελέγξει εάν κάποιος χρήστης βρίσκεται σε απόσταση μικρότερη από 15cm από την συσκευή και σε αυτήν την περίπτωση ενεργοποιεί την αντίστοιχη κατάσταση λειτουργίας θέτοντας την τιμή της μεταβλητής powerLCDProx ίση με true. Εν πρώτοις, τίθεται το trigPin σε κατάσταση Low για 2 microseconds, μετά για 10 microseconds στην κατάσταση High και τέλος στην κατάσταση Low. Οι ενέργειες αυτές πραγματοποιούνται με την βοήθεια των παρακάτω εντολών:

```
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
```

Αξίζει να τονιστεί ότι οι προαναφερθείσες καθυστερήσεις δεν επηρεάζουν αρνητικά τον κώδικα, εάν σκεφτούμε παράλληλα ότι η τιμή του timer1 αλλάζει κάθε 16 microseconds. Ύστερα, ακολουθεί η ανάγνωση του echoPin και η καταχώρηση της χρονικής διάρκειας του "ταξιδιού" του ηχητικού κύματος, το οποίο αποτελείται από την εκπομπή του έως την ανάκλασή του στην επιφάνεια πρόσπτωσης που βρίσκεται πάνω από τον αισθητήρα και από την ανάκλασή του έως την επιστροφή του στον δέκτη του αισθητήρα. Κατ' αυτόν τον τρόπο, γνωρίζοντας την ταχύτητα του ήχου μπορούμε να υπολογίσουμε την απόσταση της επιφάνειας πρόσπτωσης του ηχητικού κύματος από τον αισθητήρα. Μετέπειτα, ελέγχεται εάν είναι η πρώτη φορά που ανιχνεύεται η απομάκρυνση ενός χρήστη από την συσκευή και σε αυτήν την

περίπτωση, η τιμή της tristate μεταβλητής situation2 τίθεται ίση με 1, προκειμένου να γίνει καθαρισμός(clear) της οθόνης για μία μόνο φορά και όχι επαναλαμβανόμενα στην συνάρτηση tempLowHigh(), όπως αναφέρθηκε λεπτομερώς προηγουμένως στην περιγραφή της. Τέλος, εάν ανιχνευτεί ότι κάποιος χρήστης έχει πλησιάσει την συσκευή, αποθηκεύεται αυτή η κατάσταση λειτουργίας στην μεταβλητή powerLCDProx και η τιμή της μεταβλητής situation2 γίνεται ίση με 0, ώστε όταν απομακρυνθεί ο χρήστης να δύναται να τεθεί ίση με 1, για τους λόγους που αναφέρθηκαν λίγο παραπάνω.

void LCDafter2mins(double meanTempVal, double lastTempVal)

Η συνάρτηση αυτή χρησιμοποιείται, για να τυπώσει στην οθόνη LCD την μέση τιμή meanTempVal των θερμοκρασιών των 2 προηγούμενων λεπτών, μετά το πέρας των 2 αυτών λεπτών, για χρονικό διάστημα 10 δευτερολέπτων. Επιπλέον, πραγματοποιείται έλεγχος για το αν η τελευταία τιμή lastTempVal της θερμοκρασίας βρίσκεται πέρα από τα επιτρεπτά όρια και αναλόγως τυπώνεται στην κάτω γραμμή για 10 δευτερόλεπτα η αντίστοιχη ένδειξη μαζί με την τιμή της μέσης τιμής που αναφέρθηκε προηγουμένως.

ISR(TIMER1_OVF_vect)

Η συνάρτηση αυτή καλείται, όταν γίνεται διακοπή λόγω της υπερχείλισης του timer1(εξού και τα αρχικά της συνάρτησης **I**nterrupt **S**ervice **R**outine και του ορίσματος που δέχεται **TIMER1_OV**er**F**low_**vect**). Μέσα σε αυτήν την συνάρτηση, αρχικοποιείται ξανά ο timer1 στην τιμή 3036, επειδή πέρασε ένα δευτερόλεπτο και αυξάνονται οι απαριθμητές counter5sec, counter2mins που είναι υπεύθυνοι για την μέτρηση των 5 δευτερολέπτων και των 2 λεπτών αντίστοιχα. Όταν περάσουν 5 δευτερόλεπτα, μηδενίζεται ο counter5sec και δίνεται εντολή στον αισθητήρα για την λήψη νέας τιμής θερμοκρασίας, η οποία αποθηκεύεται στον πίνακα temps τύπου double. Παράλληλα, πραγματοποιείται ένας έλεγχος στην νέα τιμή της θερμοκρασίας που ανεγνώσθη, για το εάν βρίσκεται εκτός των προκαθορισμένων οριακών τιμών. Αν ναι, τότε ανάβει το κατάλληλο LED για την ενημέρωση του χρήστη, καθώς και το άσπρο LED (που προσομοιώνει έναν ανεμιστήρα), αν η τελευταία ένδειξη υπερβαίνει την άνω οριακή τιμή θερμοκρασίας. Έπειτα, γίνεται έλεγχος για το εάν έχουν περάσει τα 2 λεπτά. Αν ναι, τότε μηδενίζεται ο counter2mins, μεταβάλλεται η τιμή της μεταβλητής powerLCD τύπου boolean από false σε true, υπολογίζεται ο μέσος όρος των τιμών θερμοκρασίας που ελήφθησαν σε αυτήν την περίοδο(ο συνολικός αριθμός των δειγμάτων είναι 24, αφού λαμβάνεται ένα δείγμα κάθε 5 δευτερόλεπτα για 2 λεπτά συνολικά), μηδενίζεται ο πίνακας temps και καλείται η συνάρτηση LCDafter2mins(double meanTempVal, double lastTempVal). Μετά από την εν λόγω μεταβολή της μεταβλητής powerLCD σε true, ο απαριθμητής counter10sec αυξάνεται έως το πέρας των επόμενων 10 δευτερολέπτων, όπου τότε καθαρίζει και σβήνει την οθόνη.

3. Παρουσίαση προβλημάτων

Αν και εκ πρώτης όψεως το προς υλοποίηση εγχείρημα φαίνεται απλό στην δόμηση και την οργάνωσή του, παρουσιάστηκαν λίγα προβλήματα προς επίλυση για την ορθή λειτουργία του

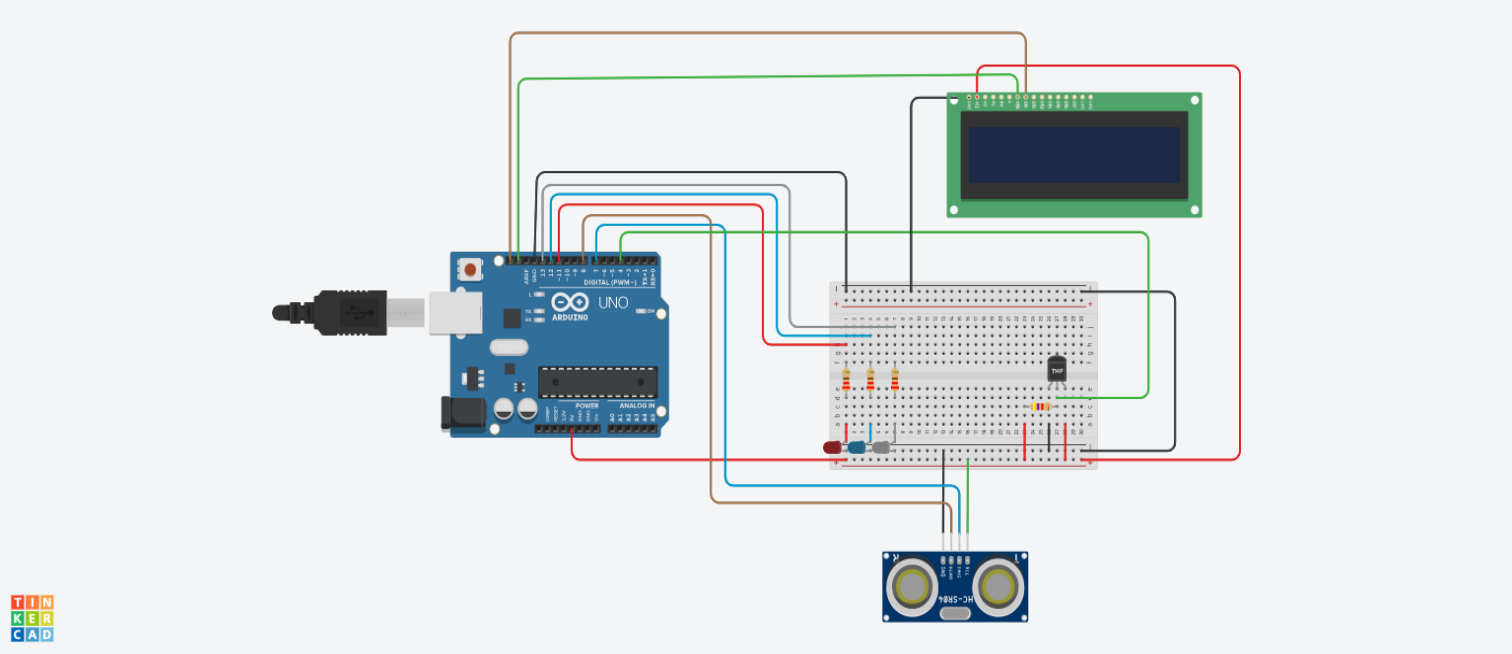
συστήματος και την ικανοποίηση όλων των προδιαγραφών που απαιτούνται να εκπληρωθούν. Πιο συγκεκριμένα, έπρεπε να βρεθούν οι σωστές λύσεις για τα παρακάτω ζητήματα:

- Τα προς μέτρηση χρονικά διαστήματα που θα χρησιμοποιηθούν είναι τα 5, 10 και 120 δευτερόλεπτα. Για να μετρηθούν σωστά, αποφασίσθηκε να γίνει χρήση μόνο ενός timer και συγκεκριμένα του 16-bit timer1 σε λειτουργία διακοπής ύστερα από υπερχείλιση κάθε 1 δευτερόλεπτο. Κατ' αυτόν τον τρόπο, ορίζονται τρεις μεταβλητές που έχουν τον ρόλο απαριθμητή δευτερολέπτων για την ακριβή μέτρηση των παραπάνω χρονικών διαστημάτων.
- Ένα πιο δύσκολο τεχνικό πρόβλημα αφορούσε το γεγονός ότι σε μία λούπα, όπου πραγματοποιείται εκτέλεση των ίδιων εντολών κατ' επανάληψη, πρέπει να γίνεται καθαρισμός της οθόνης στα σημεία που πρέπει για να μην συγχέονται τα προς επίδειξη δεδομένα. Κρίνεται όμως απαραίτητη η προσθήκη κατάλληλων ελέγχων σε αυτήν την επαναλαμβανόμενη αλληλουχία εντολών, ώστε να μην εκτελείται συνέχεια καθαρισμός της οθόνης με αποτέλεσμα αυτή να "τρεμοπαίζει". Ειδικότερα, όταν βρισκόμαστε στην κατάσταση όπου η τιμή της θερμοκρασίας έχει ξεπεράσει ορισμένα προκαθορισμένα όρια, τυπώνεται στην οθόνη κάθε φορά η κατάλληλη ένδειξη ενημέρωσης του χρήστη. Σε αυτήν την περίπτωση, πρέπει να γίνει καθαρισμός της οθόνης μόνο την πρώτη φορά και όχι συνέχεια. Αυτό το πρόβλημα λύνεται με την προσθήκη της μεταβλητής situation τύπου boolean.
- Παρόμοιο πρόβλημα σχετικά με τον καθαρισμό της οθόνης συναντάται όταν κάποιος απομακρύνεται από την συσκευή και βρισκόμαστε σε τιμή θερμοκρασίας έξω από τα προβλεπόμενα όρια. Τότε, ο καθαρισμός της οθόνης θα πρέπει να γίνει μόνο την πρώτη φορά που ανιχνεύεται η απομάκρυνση του χρήστη. Για την επίλυση αυτού του προβλήματος, εισάγεται η tristate μεταβλητή situation2, η οποία στην προαναφερθείσα κατάσταση έχει λάβει την τιμή 0(όντας πριν σε κατάσταση όπου κάποιος βρισκόταν κοντά στην συσκευή), γίνεται 1 κατά την πρώτη ανίχνευση απομάκρυνσης, κάνει τον καθαρισμό και τίθεται ίση με 2, ώστε όταν ξαναγίνει ο έλεγχος εγγύτητας, να μην είναι 0, ώστε να ξαναγίνει 1 και άρα να ξανακάνει καθαρισμό.

4. Παρουσίαση μετρήσεων

Αξίζει στο σημείο αυτό να σημειωθεί ότι το κάτω και το άνω όριο θερμοκρασίας ελήφθη 31 και 33.5 αντίστοιχα για διευκόλυνση της διενέργειας δοκιμών της ορθής λειτουργίας του προγράμματος σε σπίτι, στο οποίο τυχαίνει να μην διατίθεται κλιματισμός του χώρου. Πράγματι, τα κατάλληλα LEDs άναβαν, όταν η θερμοκρασία ήταν εκτός αυτών των ορίων, αλλά και ο αισθητήρας εγγύτητας διαπιστώθηκε ότι ανταποκρίνεται σε κάθε αίτημα του χρήστη που ενδέχεται να τον πλησιάσει σε απόσταση μικρότερη από 15cm (τιμή, η οποία ελήφθη αυθαίρετα). Τέλος, θα πρέπει να σημειωθεί ότι τα χρονικά διαστήματα έχουν υλοποιηθεί με απόλυτη ακρίβεια, γεγονός που διαπιστώθηκε με την λειτουργία χρονομέτρου κατά την λειτουργία του προγράμματος.

5. Προβολή σχηματικού διαγράμματος



Παραπάνω, παρουσιάζεται το σχηματικό διάγραμμα του κυκλώματος που χρησιμοποιήθηκε για την εκπόνηση του project. Για την κατασκευή του έγινε χρήση:

- ενός αισθητήρα θερμοκρασίας (DS18B20 Temperature Sensor)
- ενός αισθητήρα εγγύτητας (HC-SR04 Ultrasonic Module Distance)
- μίας οθόνης LCD I2C (LCD Display 16x2 Module HD44780)
- τριών LEDs, ενός κόκκινου, ενός μπλε και ενός άσπρου
- τεσσάρων αντιστάσεων, τριών των 220 ohm και μία των 4.7 kohm

Οι συνδέσεις που πραγματοποιήθηκαν:

- ο αισθητήρας θερμοκρασίας (DS18B20 Temperature Sensor) στο Digital Pin 4 μέσω μίας αντίστασης 4.7 kohm. Το άλλο άκρο της αντίστασης συνδέθηκε στα 5V(Vcc). Τα άλλα δύο άκρα του αισθητήρα συνδέονται το ένα στα 5V και το δεύτερο στη γείωση.
- ο αισθητήρα εγγύτητας (HC-SR04 Ultrasonic Module Distance) έχει τέσσερα άκρα. Το TrigPin συνδέθηκε στο Digital Pin 7, το EchoPin συνδέθηκε στο Digital Pin 8. Τα άλλα δύο άκρα συνδέθηκαν στα 5V και στη γείωση.
- η οθόνη LCD I2C (LCD Display 16x2 Module HD44780) έχει επίσης τέσσερα άκρα. Τα δύο από αυτά συνδέθηκαν στα SCL, SDA Pins του μικροελεγκτή και τα άλλα δύο στα 5V και στην γείωση.
- τα LEDs έχουν δύο άκρα (κάθοδος, άνοδος). Η κάθοδος του κόκκινου LED συνδέθηκε στην γείωση και η άνοδος συνδέθηκε σε σειρά με μία αντίσταση 220 ohm, η οποία έπειτα συνδέθηκε με το Digital Pin 11 του μικροελεγκτή. Όμοια συνδέονται και τα άλλα δύο LED, δηλαδή το μπλε και το άσπρο, στα Digital Pins 12,13 αντίστοιχα.