Μικροεπεξεργαστές και Περιφερειακά

Arduino Project

"Εξυπνος θερμοστάτης"

Ιούλιος 2019

Υπατία Δάμη 8606

Στόχος εργασίας:

Στόχος της ομάδας μας ήταν η δημιουργία ενός συστήματος ικανού να αντιλαμβάνεται την θερμοκρασία και την υγρασία του περιβάλλοντος ,την κινηση σε κοντινή απόσταση,την προβολή μετρήσεων θερμοκρασίας σε οθόνη lcd και την αποστολή των μετρήσεων αυτών με email ,όντας συνδεδεμένο σε τοπικό δίκτυο wifi. Για την πραγματοποίηση του συστήματος χρησιμοποιήσαμε έναν μικροελεγκτή *Wemos D1R2* ,ένα ράστερ, έναν αισθητήρα θερμοκρασίας και υγρασίας *DHT11* ,έναν αισθητήρα κίνησης *Ultrasonic* ,μία οθόνη LCD *LiquidCrystal_I2C* , τρία leds και συνδετικά καλώδια. Πιο συγκεκριμένα , μέσω του αισθητήρα θερμοκρασίας το σύστημα παίρνει ανά τακτά χρονικά διαστήματα (5 δευτερόλεπτα) μετρήσεις θερμοκρασίας και υγρασίας .Ανα δύο λεπτά ο μέσος όρος των 24 τελευταίων μετρήσεων παρουσιάζονται στην οθόνη lcd,καθώς και αποστέλλεται με email . Ο αισθητήρας κίνησης λειτουργεί παράλληλα με τον αισθητήρα θερμοκρασίας και αντιλαμβάνεται αν παρουσιαστεί κάτι σε απόσταση μικρότερη από 1 μέτρο. Αν αυτό συμβεί τότε εμφανίζεται στην lcd οθόνη η τρέχουσα θερμοκρασία του δωματίου καθώς και ο μέσος όρος των τελευταίων 2 λεπτών. Σε ακραίες θερμοκρασίες ανάβουν 2 led, ένα για χαμηλές θερμοκρασίες και ένα για υψηλές και εμφανίζεται ανάλογο μήνυμα στην οθόνη. Αν η θερμοκρασία ξεπεράσει μια συγκεκριμένη τιμή ανάβει ένα τρίτο led, το οποίο σβήνει αν η θερμοκρασία κατέβει ξανά κάτω από την τιμή αυτή(αναπαράσταση διακόπτη ρελέ μέσω του led) .Να σημειωθεί ότι η οθόνη lcd συνδέθηκε στο σύστημα με έναν κονέκτορα I2C για την αποδέσμευση όσο περισσότερων pins για τους αισθητήρες.

Υλοποίηση προγράμματος:

Αρχικά για την υλοποίηση του προγράμματος χρησιμοποιήσαμε τις εξής βιβλιοθήκες:

<Ticker.h> - βιβλιοθήκη για τον timer που χρησιμοποιήσαμε

<DHTesp.h> - βιβλιοθήκη για τον αισθητήρα θερμοκρασίας και υγρασία

<LiquidCrystal_I2C.h> , <Wire.h> - βιβλιοθήκες για την οθόνη lcd

<ESP8266WiFi.h> - βιβλιοθήκη για το wifi

Στην αρχή του προγράμματος , γίνεται η δήλωση των μεταβλητών μας και η δημιουργία των αντικειμένων αισθητήρων , της οθόνης lcd και του wifi Client;

DHTesp dht;

WiFiClient espClient;

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7);

Οι παράμετροι στον Constructor της lcd οθόνης παριστούν τα pins του μικροελεγκτή που αντιστοιχούν στα pins της οθόνης στη συνδεσμολογία.

Στη συνέχεια γίνεται το setup του wifi , όπου εισάγεται στο σύστημα το όνομα και ο κωδικός του wifi και έπειτα εκτελείται η σύνδεση . Ο χρήστης ενημερώνεται όταν το σύστημα συνδεθεί στο δίκτυο και για τον IP αριθμό που του αντιστοιχεί.

Ακολουθεί το σετάρισμα των υπόλοιπων στοιχείων του προγράμματος, εντός του loop setup(). Τα pins D3,D4 και D5 τίθενται ως pins εξόδου και το καθένα αντιστοιχεί σε ένα από τα προαναφερθέντα leds, που ανάβουν όταν το σύστημα πάρει συγκεκριμένες μετρήσεις θερμοκρασιών. Στη συνέχεια ορίζονται τα pins 12 kai 13 ως είσοδος και έξοδος, τα οποία αντιστοιχουν στα pins που χρησιμοποιεί ο αισθητήρας κίνησης Ultrasonic.Το ένα pin τίθεται ως έξοδος για να στέλνει ένα σήμα και το άλλο ως είσοδος για να δέχεται το σήμα που επιστρέφει αφού ανακλαστεί σε ένα εμπόδιο. Τέλος σετάρουμε με τις κατάλληλες συναρτήσεις την οθόνη lcd και τέλος θέτουμε σε λειτουργία τον timer, να ενεργοποιείται κάθε 5 δευτερόλεπτα με την παρακάτω συνάρτηση.

blinker.attach(5, changeState);

Αφού περάσουν 5 δευτερόλεπτα καλείται η συνάρτηση διαχείρισης του timer “changeState()”.

Στη συνέχεια ξεκινάει ο κύριος βρόχος του προγράμματος μας loop().Στην μεταβλητή *time* αποθηκεύεται η μέτρηση του αισθητήρα κίνησης. Ο αισθητήρας ουσιαστικά μετράει την ώρα που έκανε το σήμα που έστειλε για να ταξιδέψει μέχρι να επιστρέψει στον δέκτη του αισθητήρα , και έτσι υπολογίζεται η απόσταση του από το εμπόδιο, αν λάβουμε υπόψιν ότι το σήμα ταξιδεύει με την ταχύτητα του φωτός(γνωστός τύπος x=C*t).Έχοντας υπόψην ότι θέλουμε να ενημερωθούμε σε περίπτωση που κάτι πλησιάσει το σύστημα πάνω από ένα μέτρο, υπολογίζουμε από τον τύπο x=U*t αυτό συμβαίνει όταν η μεταβλητή time παίρνει τιμή κάτω από 3000.Οι μεταβλητές avg και sum αφορούν τις μετρήσεις του αισθητήρα θερμότητας και παριστάνουν τον μέσο όρο και το άθροισμα των 24 τελευταίων μετρήσεων. Στην αρχή του προγράμματος έχουν οριστεί οι μεταβλητές t1 και t2 οι οποίες χρησιμοποιούνται στον χειρισμό του timer. Αρχικά είναι μηδενισμένες και οι 2. Όταν περάσουν 5 δευτερόλεπτα από την ενεργοποίηση του timer , καλείται η ρουτίνα changeState() ,(timer Handler) ,εντός της οποίας αυξάνεται η μεταβλητή t2. Όταν το πρόγραμμα επιστρέψει στο κύριο loop , στο εσωτερικό for loop, επειδή

πλέον ισχύει η συνθήκη του if βρόχου t1!=t2 , μπαίνουμε στον if βρόχο και παίρνουμε την επόμενη μέτρηση θερμοκρασίας, μέσω του αντικειμένου dht. Επίσης η μεταβλητή t2 παίρνει πάλι τιμή 0 , για να μην εισέρθουμε ξανά στον if βρόχο αν δεν περάσουν 5 δευτερόλεπτα και κληθεί η ρουτίνα του timer. Η μεταβλητή counter αποθηκεύει τον αριθμό των τελευταίων μετρήσεων που έχουν πραγματοποιηθεί. Αν φτάσει τις 24 μετρήσεις μηδενίζεται και ξεκινάει από την αρχή. Ο κώδικας του βρόχου if που πραγματοποιούνται οι μετρήσεις θερμοκρασίας :

```
if (t1 != t2) {

    t2 = t1;

    t = dht.getTemperature();

    counter++;

    sum = sum + t;

    Serial.println(t);

}
```

Έπονται άλλοι 2 if βρόχοι που ελέγχουν αν η θερμοκρασία που μετρήθηκε έχει τιμή πάνω από 30 βαθμούς ή κάτω από -10. Αν ισχύει κάποια από αυτές τις συνθήκες εμφανίζεται το κατάλληλο μήνυμα στην οθόνη lcd και στέλνεται ενημερωτικό email. Επίσης ανάβει το κατάλληλο led. Ο κώδικας της περίπτωσης που έχουμε πάνω από 30 βαθμούς :

```
if (t > 30) {

    digitalWrite(D3, HIGH);

    digitalWrite(D4, LOW);

    lcd.clear();

    lcd.print("Temp too high");

    flag2 = 1;

    if (Wflag == 0) {

      byte ret = sendEmail();

      Wflag=1;

    }
```

Στον ίδιο αέναο βρόχο for, αφού περάσουμε το κομμάτι που αφορά την πιθανή μέτρηση θερμοκρασίας, φτάνουμε στο κομμάτι που αφορά τη μέτρηση του αισθητήρα κίνησης.Θέτουμε το pin εξόδου του αισθητήρα σε HIGH για 10 μseconds και μετά από μια καθυστέριση αποθηκεύουμε τη μέτρηση στη μεταβλητή time . Ο κώδικας για τη διαδικασία αυτή φαίνεται παρακάτω:

```
digitalWrite(trigPin, LOW);
   delayMicroseconds(2);
   digitalWrite(trigPin, HIGH);
   delayMicroseconds(10);
   digitalWrite(trigPin, LOW);
   time = pulseIn(echoPin, HIGH);
```

Έπειτα σε έναν if βρόχο ελέγχεται αν η τιμή της μεταβλητής time είναι μικρότερη από 3000. Αν ναι, σημαίνει πως κάτι πλησίασε πάνω από ένα μέτρο το σύστημα , επομένως εμφανίζεται στην οθόνη lcd ο μέσος όρος της θερμοκρασίας των τελευταίων 2 λεπτών καθώς και η τρέχουσα θερμοκρασία. Μέσω της συνάρτησης millis() αποθηκεύουμε στη μεταβλητή startime την στιγμή που άναψε η οθόνη . Έπειτα ελέγχουμε την τιμή της μεταβλητής συνεχώς και αφότου ξεπεράσει την τιμή 10000(10 δευτερόλεπτα) κλείνουμε την lcd οθόνη. Στον τελευταίο βρόγχο ελέγχου του for loop το πρόγραμμα εισέρχεται όταν η τιμή της μεταβλητής counter πάρει την τιμή 24, έχουν δηλαδή πραγματοποιηθεί οι 24 τελευταίες μετρήσεις θερμοκρασίας. Μηδενίζεται ο counter, υπολογίζεται ο μέσος όρος των μετρήσεων και μηδενίζεται το άθροισμα(sum) για την επόμενη σειρά μετρήσεων. Έπειτα εμφανίζονται οι μετρήσεις στην οθόνη lcd και στέλνεται email.

Τέλος , ακολουθεί η συνάρτηση sendEmail() που είναι υπεύθυνη για την αποστολή των emails . Το πρόγραμμα στέλνει 25 email με μετρήσεις θερμοκρασίας την ώρα . Για την αποστολή των emails χρησιμοποιούμε τον server “SMTP2GO” με το σύστημά μας να λειτουργεί ως client. Αρχικά επιχειρούμε σύνδεση με τον server στη θύρα 2525 με την συνάρτηση :

```
espClient.connect(server, 2525)
```

Στη συνέχεια επικοινωνούμε με τον server μέσω της συνάρτησης

```
espClient.println("String")
```

και ελέγχουμε την τιμή που επιστρέφει η emailResp() ,δηλαδή την απάντηση του server στο μύνημά μας.Αν η τιμή είναι 0 , δεν υπάρχει σύνδεση με τον server , ενώ

αν είναι 1 η σύνδεση είναι επιτυχής και μπορούμε να συνεχίσουμε την επικοινωνία . Η διαδικασία επαναλαμβάνεται πολλές φορές μέχρι να στείλουμε όλα τα απαραίτητα στοιχεία στον server. Στέλνουμε δηλαδή μια φράση και ελέγχουμε την απόκριση του server. Αν η σύνδεση είναι εντάξει συνεχίζουμε . Παράλληλα με Serial Print ενημερωνόμαστε από την οθόνη για το τι στέλνεται κάθε στιγμή στον server. Αρχικά στέλνονται τα στοιχεία του λογαριασμού από τον οποίο θα σταλεί το email (ιστότοπος, username, password) με παράδειγμα τον κώδικα για αποστολή του κωδικού:

```
Serial.println(F("Sending Password"));

  espClient.println("elByamJwd1V6eXcx");

  if (!emailResp())

    return 0;
```

Έπειτα στέλνονται τα δεδομένα του email , και ανάλογα την τιμή της μεταβλητής AVflag που πήρε την τιμή της στο βασικό loop, εισάγονται ως δεδομένα του mail οι κατάλληλες μετρήσεις θερμοκρασίας και προειδοποιητικά μηνύματα. (πχ το σπίτι είναι πολύ κρύο ή πολύ ζεστό.)

Η επικοινωνία με τον server σταματάει με την συνάρτηση:

```
espClient.stop();
```

και έτσι ολοκληρώνεται η διαδικασία αποστολής email και το πρόγραμμα επιστρέφει στο βασικό loop, το οποίο εκτελεί συνεχώς την ίδια ακολουθία ενεργειών μέχρι να απενεργοποιηθεί το σύστημα.