

Η χρήση της εντολής `delay()` (βλέπε σκαρίφημα 1) στα πλαίσια ενός προγράμματος για το `arduino` υπο είναι εύκολα κατανοητή και δεν απαγορεύεται η χρήση της για απλές εφαρμογές που ελέγχουν μία έξοδο (π.χ. αναβοσβήσιμο ενός LED ή πολλών LEDs που είναι συνδεδεμένα στην ίδια έξοδο, ή πολλών LEDs συνδεδεμένα σε διαφορετικές εξόδους που αναβοσβήνουν διαδοχικά).

Το πρόβλημα με τη χρήση αυτής της εντολής είναι ότι ο μικροελεγκτής κατά την εκτέλεση της, «σταματά» για το χρονικό διάστημα του ορίσματος της, εκφρασμένο σε `msec`. Κατ' αυτό το χρονικό διάστημα ο μικροελεγκτής δεν «κάνει τίποτα» (εκτός κάποιων ειδικών εξαιρέσεων) και έτσι αν συμβεί κάποια αλλαγή τιμής σε κάποιο π.χ. αισθητήρα, αυτή θα γίνει αισθητή στην επόμενη περιόδευση του προγράμματος που τρέχει ο μικροελεγκτής (αν δεν έχει αλλάξει εν τω μεταξύ). Γενικά η εντολή αυτή δεν χρησιμοποιείται σε σύνθετα προγράμματα που ελέγχουμε πολλές εισόδους /εξόδους όπου συχνά έχουμε καταστάσεις “multitasking”.

Συνηθέστατα, και σ' ένα πιο αναβαθμισμένο επίπεδο προγραμματισμού χρησιμοποιούμε την εντολή `millis()` (βλέπε σκαρίφημα 2). Η εκτέλεση της εντολής αυτής μας επιστρέφει τα `milliseconds` που έχουν παρέλθει από την έναρξη λειτουργίας (power on) ή το reset του `Arduino`.

Η συνήθης προσέγγιση για τη χρήση αυτής της εντολής είναι να ορίσουμε ένα χρονικό διάστημα για το οποίο θέλουμε να συμβαίνει κάτι, και με τη χρήση της `millis()` ελέγχουμε πότε έχει ξεπεραστεί αυτό το χρονικό διάστημα (όπως φαίνεται στο παρακάτω sketch).

```
unsigned long previousMillis = 0; //Stores the last time the LED was
updated
const unsigned long period = 1000; //the value is a number of milliseconds
const byte ledPin = 13; //using the built in LED

void setup()
{
  pinMode(ledPin, OUTPUT);
  previousMillis = millis(); //initial start time
}

void loop()
{
  //get the current "time" (actually the number of milliseconds since the
  program started)
  unsigned long currentMillis = millis(); // Get the current time
  //test whether the period has elapsed
  if (currentMillis - startMillis >= period)
  {
    //if so, change the state of the LED.
    digitalWrite(ledPin, !digitalRead(ledPin));
    //IMPORTANT to save the start time of the current LED state.
    previousMillis = currentMillis;
  }
}
```

Το παραπάνω sketch λήφθηκε από την παρακάτω ιστοσελίδα.

<https://forum.arduino.cc/t/using-millis-for-timing-a-beginners-guide/483573>

Εάν χρησιμοποιήσουμε διαφορετικά `periods` και διαφορετικά `startMillis` μέσα στο `loop()` μπορούμε να ελέγξουμε διαφορετικά συμβάντα χωρίς να περιμένουμε να λήξει η επενέργεια του ενός συμβάντος για να συμβεί το επόμενο όπως θα έπρεπε να γίνει με τη χρήση της εντολής `delay()`.