

ARDUINO

Beh, non è mai troppo tardi, forse, ho scoperto in ritardo Arduino Uno un microprocessore programmabile, già montato su basetta con 14 ingressi/uscite digitali e 6 ingressi/uscite analogiche .

E' programmabile in C e per programmarlo basta collegarlo con un cavo tipo stampante a una porta USB, non si deve smontare il microprocessore dalla basetta per programmarlo e questo semplifica il procedimento Programmazione/Verifica funzionamento.

Per le prove si può alimentare anche con la USB del PC.

Il sistema di sviluppo (si dice così ?) viene fornito gratuitamente insieme all'editor.

A proposito dell'editor mi sono dato da fare subito a modificare colori e grandezza font.

Volevo font dai colori vivaci rosso, verde e blu su sfondo nero per risparmiare un po' la vista.

ROSSO : strutture e controllo strutture: void setup() , void loop () , if ... else , for
tipo di variabili : int , double ecc.

BIANCO : il valore delle variabili

BLU : le costanti tipo: HIGH ; LOW ; INPUT ; OUTPUT

VERDE : i commenti

Le parole in Rosso e Blu devono essere scritte come nelle "Reference " rispettando maiuscole e minuscole.

Le parole in bianco sono scelte da noi.

**Per digitare una parentesi graffa aperta: { o chiusa: } , premere
contemporaneamente**

In Windows:

AltGr + ↑ (Maiuscola) + [per la graffa aperta , AltGr + ↑ (Maiuscola) +]
per la graffa chiusa

In MacOSx:

Alt + ↑ (Maiuscola) + [per la graffa aperta, Alt + ↑ (Maiuscola) +] per la
graffa chiusa

Ricordarsi di digitare il punto e virgola ; alla fine di ogni istruzione!!

Ho modificato il file " theme.txt " e anche il " preference.txt " .

In Windows, credo di ricordare, siano file nascosti, quindi bisogna abilitarne la visualizzazione.

In MacOSx bisogna per il " theme.txt " "Mostra contenuto pacchetto" della :
"Arduino.app " , clic col tasto destro del mouse su: " Arduino.app"

Poi aprire: Contents – Resources – Java – Lib – Theme - theme.txt

Sempre in MacOSx per il “preference.txt” bisogna trovarlo nella directory indicata in Preferenze.

Allego comunque i due file che vanno bene sia per Windows che per MacOSx.

Ho notato che l’istruzione “goto” è mostrata in caratteri bianchi, ma funziona egualmente.

Nelle “References” si fa intendere che un bravo programmatore non vi dovrebbe far ricorso ma la tentazione è grande. Comunque è possibile farne a meno.

Ad esempio avevo la necessità di far partire un procedimento ciclico con un minuto di ritardo, alla fine ci sono riuscito senza il “goto” con un ciclo “for”.

Altra cosa che ho notato e non sono riuscito a spiegarmi: dovevo generare una forma d’onda impulsiva ciclica diciamo alta per 1000 us e bassa per 20000 us.

Per il tempo high nessun problema , delayMicroseconds(1000); ma imponendo un delayMicroseconds(20000); per il tempo low non ottenevo la forma d’onda desiderata.

Il massimo valore del delayMicroseconds è risultato essere 16383 us , circa, invece di 32767 us, visto che il numero in parentesi è un intero (int) . Boh !

Ho ovviato a ciò con due delayMicrosecond(10000); uno di seguito all’altro.