

# Introduzione a ktechlab

programma open source per disegno di schemi elettronici, simulazione e programmazione di PIC, anche tramite flow chart, adatto per didattica ed uso hobbistico.

**ktechlab** è un programma per disegnare schemi di elettronici e simularli in tempo reale.

La caratteristica che però lo rende particolare rispetto ad altri simulatori è che comprende anche la simulazione di microcontrollori PIC che si possono programmare direttamente nell'ambiente, e tra i vari modi, si può utilizzare un **flow chart** per inserire le istruzioni e collegarle in **modo molto semplice e comprensibile anche ai meno esperti** (un ragazzo digiuno di programmazione è riuscito in un pomeriggio a realizzare un programma di segnalazione del numero di posti liberi – da 0 a 99 - di un parcheggio, con pulsanti per l'entrata e per l'uscita delle auto e visualizzazione su due display a 7 segmenti).

L'utilizzo dei PIC è limitato ai tipi 16F84, 16F627 e 16F628 .

Il programma è disponibile solo per GNU/Linux, mi sa che gli utenti di window\$ si mangeranno le mani :). Ah, dovrebbe essere possibile installare Ubuntu in Windows e poi installare ktechlab.

## Installazione

il programma è nativo per KDE ma funziona anche sotto GNOME.

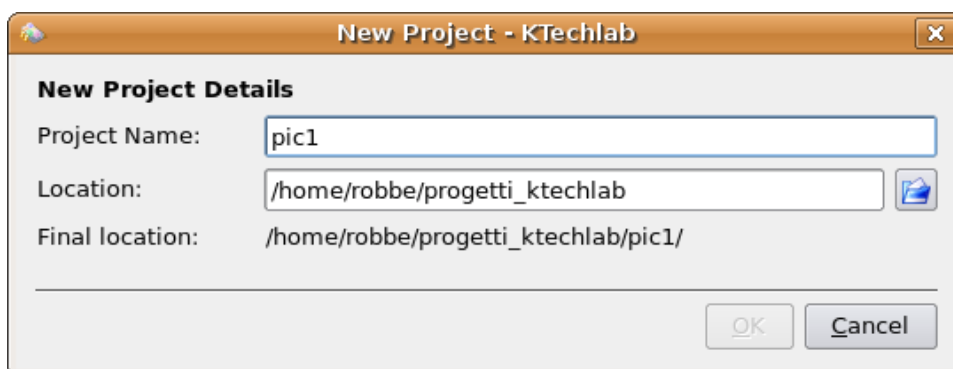
Se utilizzate Ubuntu/KUbuntu/Debian per installarlo basta dare da terminale: *sudo apt-get install ktechlab* o dal menu *Sistema-Amministrazione-Gestione pacchetti Synaptic* cercare il pacchetto *ktechlab* ed installarlo.

Altrimenti si può scaricare da [www.ktechlab.org](http://www.ktechlab.org) il sorgente o il pacchetto per Fedora, OpenSuse, Gentoo ed alte distribuzioni.

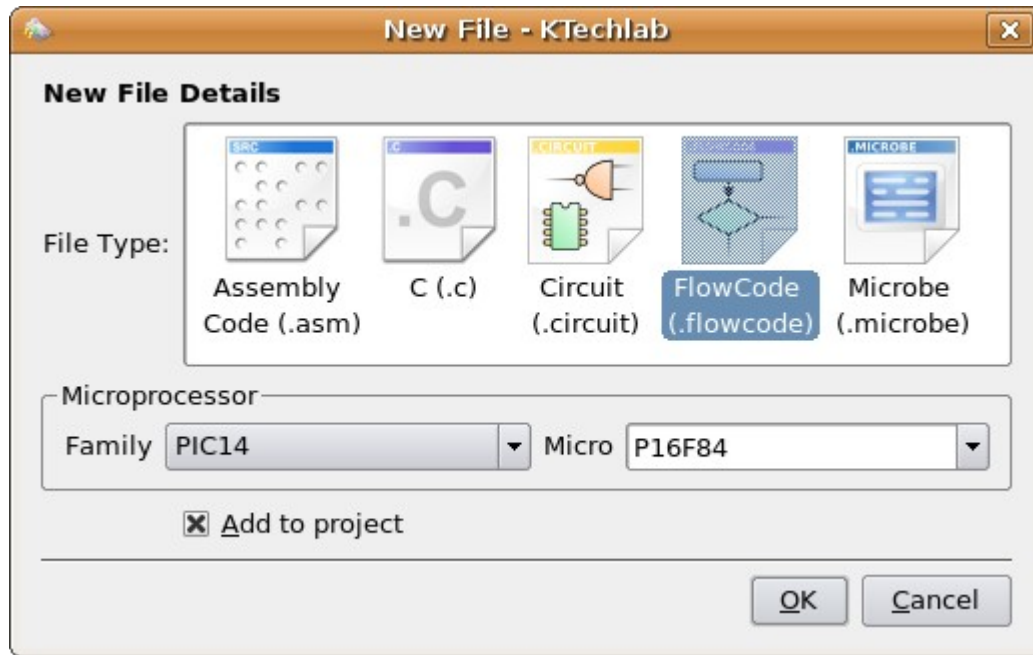
## Esempio di uso semplificato: programmazione con flow-chart lampeggio di un led

### creazione del progetto

Dal menu *Project – New Project* , dare un nome al progetto che sarà anche il nome della directory che verrà creata e che conterrà tutti i file del progetto. In questo caso *pic1* .



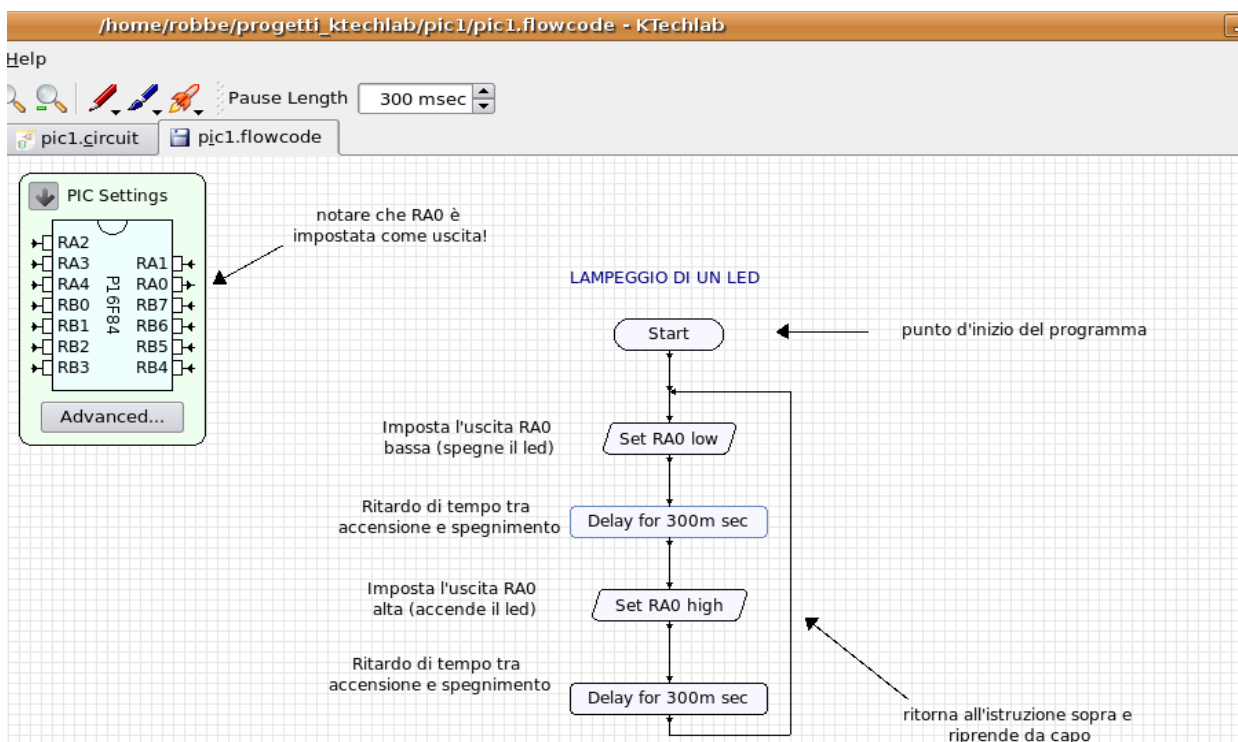
## Creazione di un nuovo file



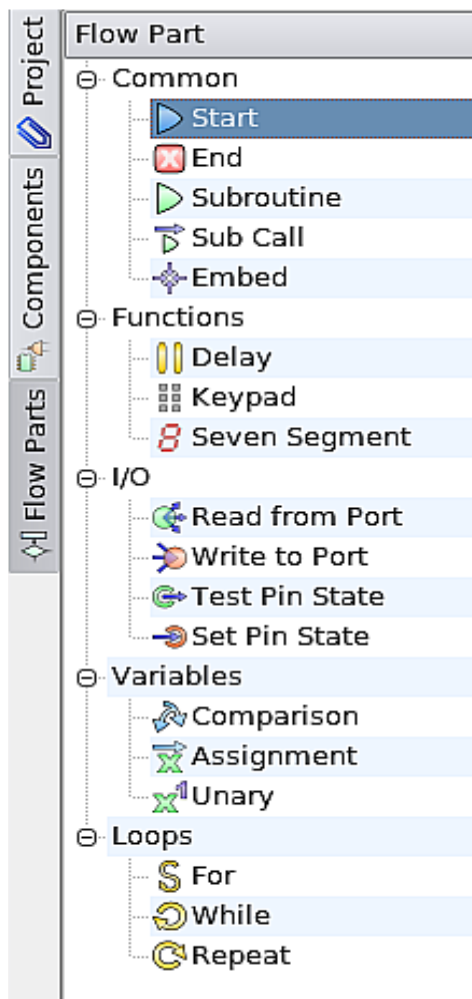
Qui si può scegliere il tipo del file: *circuit* per lo schema, *flowcode* per programmare il PIC con il flow-chart. Avremo bisogno di crearli entrambi e dare loro dei nomi significativi. In alternativa al *flowcode* si può programmare in *c*, in *assembler* o in *microbe* (una specie di basic), ma per i principianti è sicuramente più adatto il *flowcode*.

## Programmazione con il flow chart

Nel file *pic1.flowcode* andremo a creare con il mouse un diagramma di flusso del genere che non fa altro che accendere e spegnere un led collegato al pin RA0 del PIC.



La prima cosa da fare è, nel grafico della “piedinatura” del PIC in alto a sinistra, cliccare sul piedino RA0 e trascinare il mouse verso l'esterno del Pic: così si imposta il **piedino come uscita**. In maniera analoga imposteremo i piedini che servono come uscita o, trascinandoli verso l'interno del PIC, come ingresso.



A sinistra nella schermata apriamo il pannello denominato *Flow Parts* dove troviamo le “istruzioni” in forma di blocchi.

Basta trascinarle con il mouse nella zona di destra e sistemarle dove vogliamo.

Ogni programma inizia con **Start** ma, se è un programma iterativo, non occorre l'istruzione **End**.

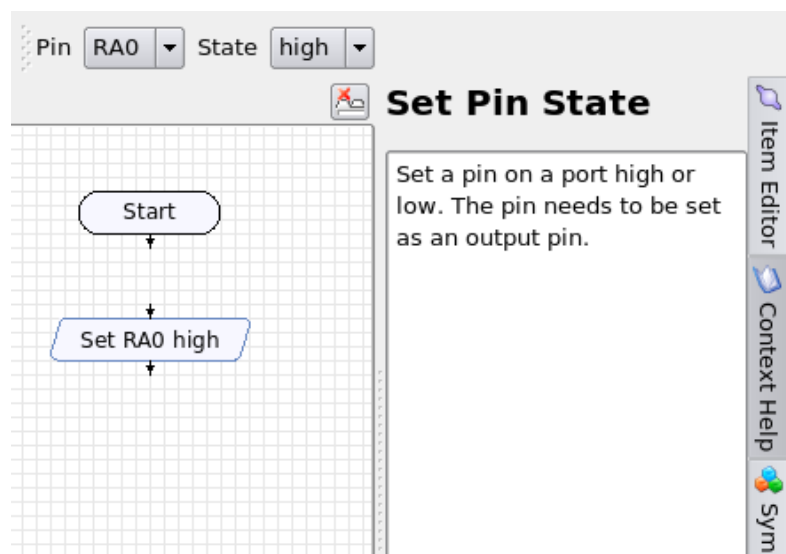
I blocchi più importanti sono **Test Pin State** per leggere il bit su un pin di ingresso, **Set Pin State** per impostare un pin di uscita alto o basso.

Si possono usare anche variabili con nomi impostabili a piacere ed assegnarne il valore, fare dei test, incrementarne il valore ecc.

Sono possibili anche i cicli **for**, **while** e **repeat**.

Una volta inserito un blocco possiamo selezionarlo con un click (il suo bordo diventa azzurro) e, nel pannello di destra, denominato *Context Help*, avremo una descrizione del blocco.

Ancora, una volta selezionato il blocco, nella parte alta vedremo le possibili azioni associate ad esso ad es. nella figura sotto possiamo scegliere quale pin settare (RA0) e se alto o basso (high).



Per unire i blocchi basta cliccare il tasto sx vicino alla freccina di uscita di un blocco e trascinare il mouse vicino alla freccina di ingresso di un altro blocco.

Per cancellare un blocco o un collegamento basta selezionare (provate con la selezione trascinando il mouse da destra verso sinistra su un oggetto) e premere il tasto *canc* sulla tastiera.

Per il blocco **Delay**, una volta inserito e selezionato, potete modificare il valore del ritardo temporale agendo sul controllo che appare nella parte alta della schermata.

Il programma è ancora in fase di sviluppo, e lamenta un bug nei controlli: attenzione a non portare a zero il valore del controllo pena blocco del programma stesso!

Alla fine dovreste avere un *flow chart* simile a quello riportato a pagina 2.

Ora salvate (! :) e passate al file del circuito, creato precedentemente.

### Disegno dello schema

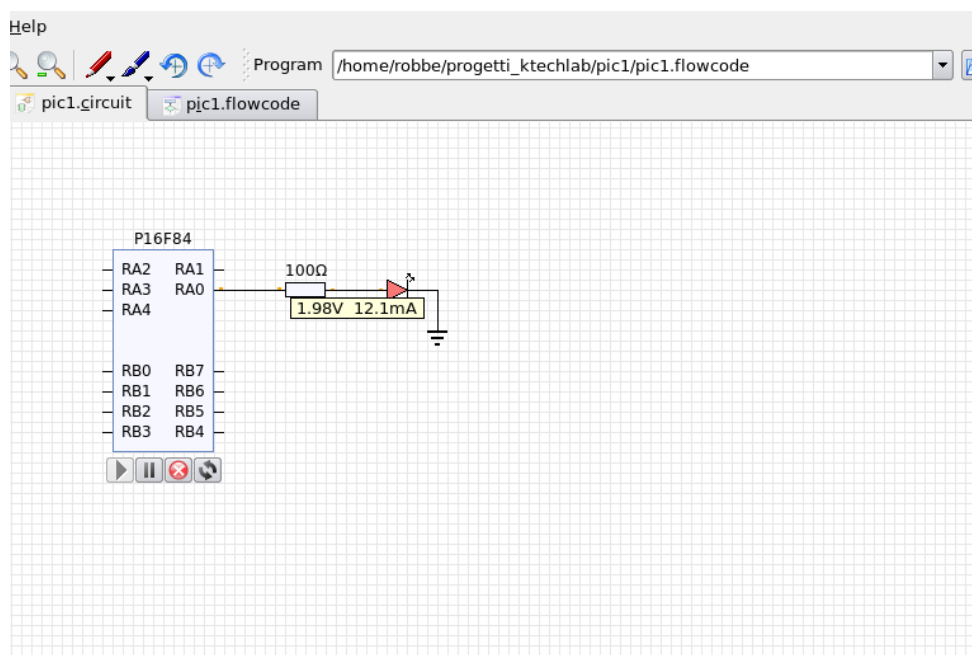
Nel pannello di sinistra selezionate **Components** e troverete un elenco di componenti elettronici, tra cui il PIC, che trascinerete nella zona di centrale. Altrettanto fate con una resistenza, un led e un simbolo di massa (Ground).

Selezionate il PIC e, nel controllo che apparirà nella parte alta della schermata, scegliete il file *flowcode* che avete salvato in precedenza. Questo è il programma creato in precedenza.

Subito il PIC prenderà un altro aspetto e sarà possibile collegare il pin RA0 alla resistenza, sempre trascinando con il mouse un “filo” di collegamento tra i terminali.

Appena terminato il circuito lanciamo il programma con la *freccia* (run) sotto il simbolo del PIC e partirà immediatamente la simulazione con tanto di accensione e spegnimento del led.

Inoltre lasciando il mouse su un collegamento avremo un'indicazione della tensione e dalla corrente in quel punto del circuito.



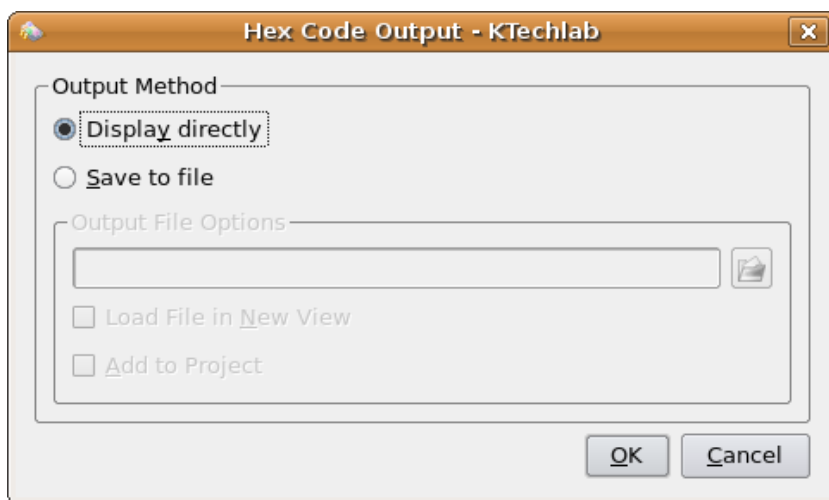
Tenete conto che la simulazione ha dei tempi che non corrispondono esattamente alla realtà e bisogna fare delle prove prima di portare il programma su un PIC reale.

## Programmazione del PIC

Una volta che il programma e la simulazione diano un risultato soddisfacente si preme, nel file del flowcode, il pulsante a forma di razzo.

Dovrebbe essere possibile, avendo collegato un programmatore compatibile, compilare e programmare direttamente il PIC.

Io purtroppo non ho potuto farlo ma ho risolto scegliendo **Convert to Hex**: si ottiene questa finestra.



Credo per un bug del programma non è possibile salvare il file .hex ma, se ve lo fate mostrare sullo schermo, poi è facile copiarlo ed incollarlo dentro un editor di testo.

Il file sarà da salvare con un'estensione .hex e da sparare nel PIC tramite qualche altro programma software compatibile con il vostro .

## Note finali

Probabilmente oltre a *ktechlab* bisogna, o sarebbe utile, installare anche altri pacchetti come *gpsim*, *gpasm* (assemblatore), *picprog* (programmatore) e se si usa la programmazione in c anche un compilatore come *sdcc* .

Nel wiki di [www.ktechlab.org](http://www.ktechlab.org) trovate qualche esempio d'uso del programma ed indicazioni varie.

L'uso che ne ho fatto non è stato approfondito ma mi è sembrato comunque un programma utile e soprattutto semplice da apprendere; considerando poi che è un software open source si può anche soprassedere ai piccoli bug che presenta.

Questo testo è stato realizzato da Roberto De Colle usando solamente software libero e rilasciato sotto Creative Commons Attribuzione 2.5 Italia License nella speranza che sia utile a qualcuno.

