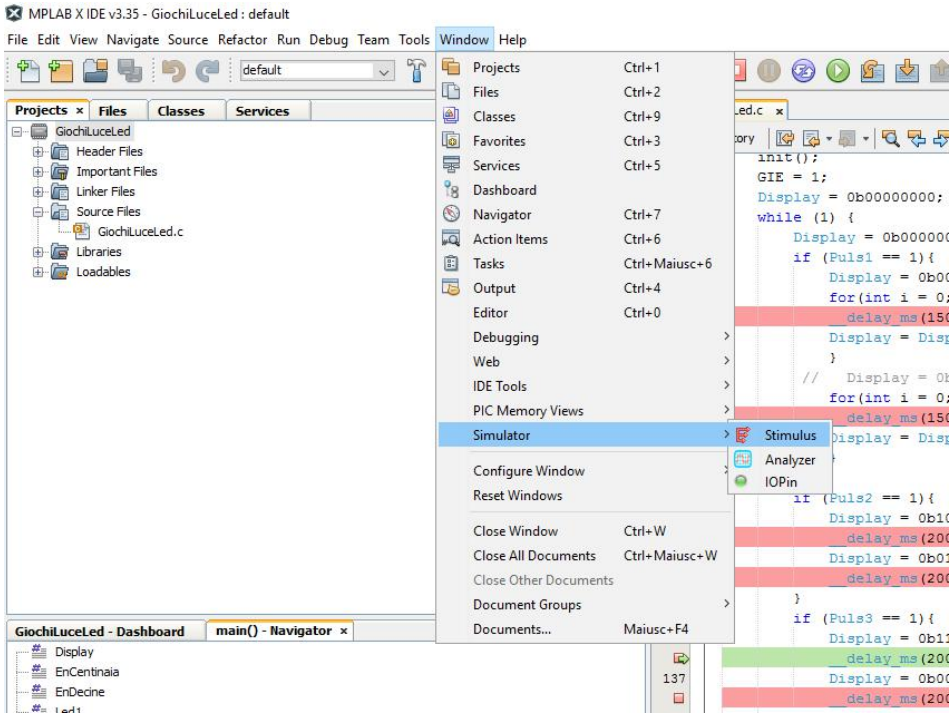
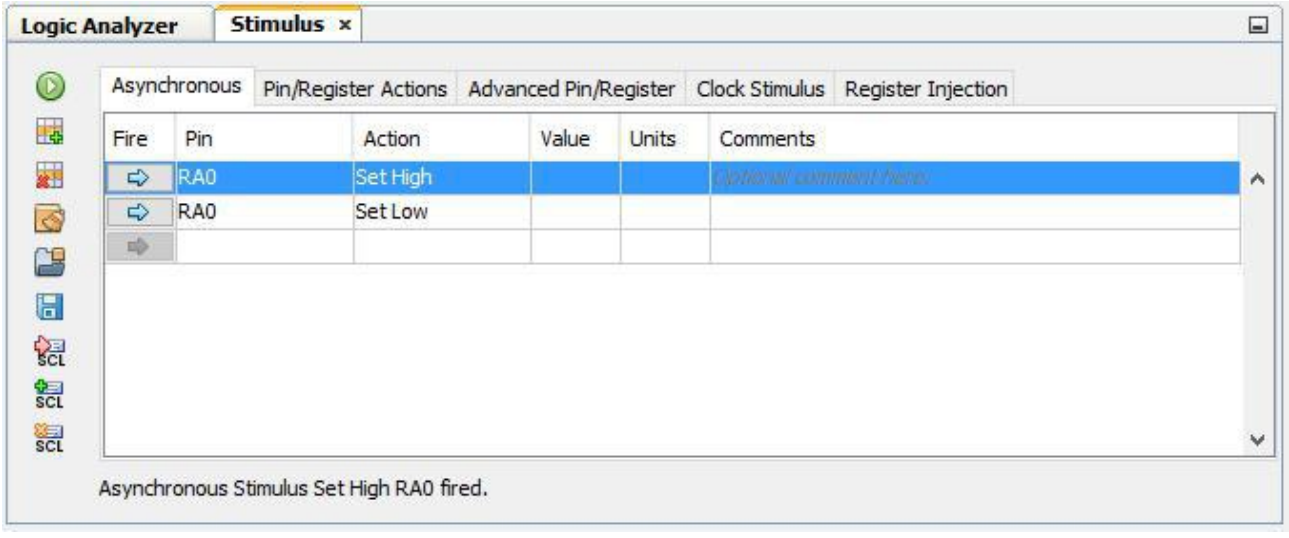
Guida simulazione MPLAB X 3.35

Per simulare un progetto in MPLAB X occorre verificare che il nostro progetto venga compilato correttamente.
Inseguito aprire due finestre: il Logic Analyzer e lo Stimulus.
Il primo serve per visualizzare lo stato delle uscite; mentre il secondo serve per modificare lo stato degli ingressi.

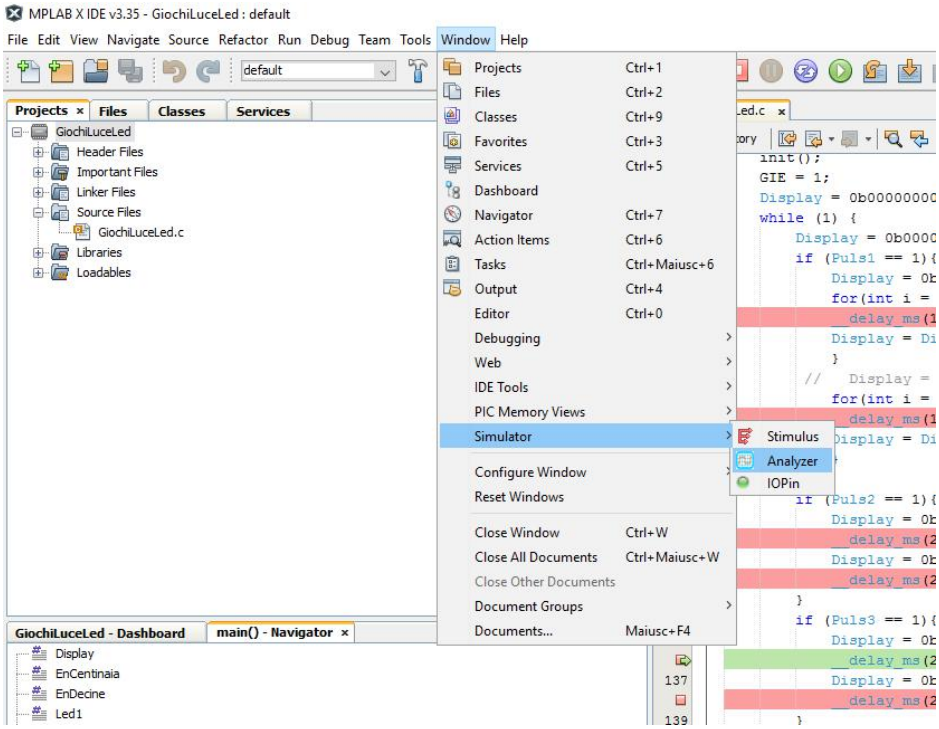
Per aprire lo stimulus il percorso è **Window -> Simulator -> Stimulus**



Per settare gli stati del pin occorre cliccare nella casella e selezionare il pin di nostro interesse poi nella casella a fianco selezionare l'ingresso che vogliamo dare al pin.
Se vogliamo cambiare lo stato dell'ingresso a nostro piacimento nella casella Action dobbiamo selezionare Toggle e poi durante la simulazione cliccare su FIRE (la freccia).
Ogni volta che premiamo la freccia lo stato dell'ingresso cambia.
Selezionando invece Set High o Set Low possiamo forzarlo a rimanere alto o basso.

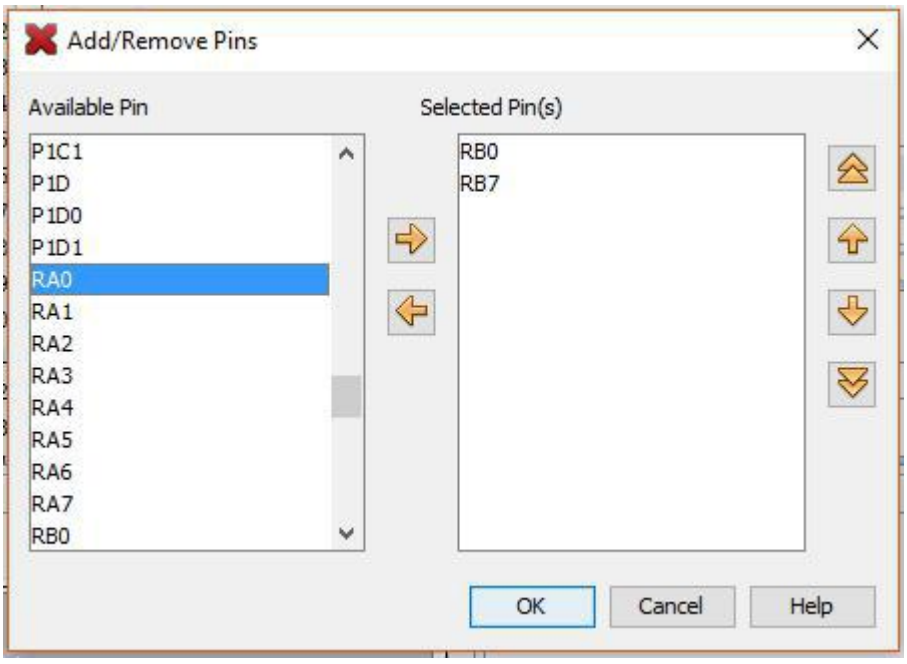


Per aprire il logic analyzer il percorso è **Window -> Simulator -> Analyzer**



Ora si aprirà in basso la finestra desiderata ora occorre selezionare i bit di cui vogliamo vedere lo stato, per fare ciò bisogna cliccare nel nell'icona seguente: 

Nella finestra che si aprirà bisogna selezionare dalla tabella di sinistra i bit che ci interessano e con la freccia spostarli nella tabella di destra ed infine dare l'ok.



Ora si passa alla fase di simulazione vera e propria per fare ciò bisogna cliccare nell'icona del debug () a questo punto il nostro progetto verrà simulato.
Ora simulando gli ingressi, premendo sulla freccia () posta a fianco alla stato che vogliamo dare all'ingresso nella finestra stimulus, osserveremo la variazione delle uscite tramite la finestra del Logic Analyzer.

ATTENZIONE: le variazione nel logic analyzer si vedranno solo quando la simulazione sarà o messa in pausa () o fermata( ).

Dopo aver fermato la simulazione per farla ripartire occorre cliccare nel pulsante start ( ).

Per facilitare la visualizzazione delle uscite si possono utilizzare i BREAKPOINTs, i quali bloccano la simulazione quando viene eseguita la riga di codice alla quale sono associati. Possono anche essere utilizzati per capire quando viene eseguita una riga di codice. Per collocare i BREAKPOINT basta cliccare sul numero della riga presso la quale vogliamo che venga fermata la simulazione, a questo punto al posto del numero ci sarà un quadratino rosso.
ATTENZIONE: se il quadratino è spezzato significa che il breakpoint non verrà eseguito perché non è possibile fermare la simulazione su quella riga di codice.

Esempio:

```
76          }
77          else
78          {
            IO_RB0_SetLow();
            }
81          // Add your application code
```

Il primo breakpoint è corretto il secondo no.

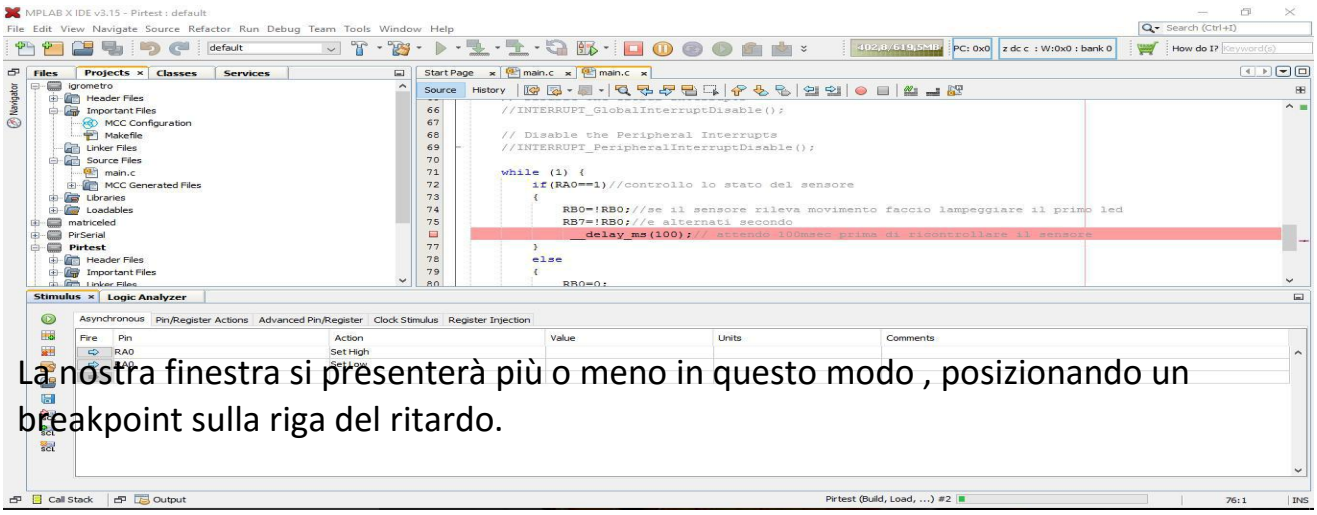
Quando viene eseguito il breakpoint verremo portati automaticamente a quel breakpoint che assumerà questo aspetto:

```
74          {
              IO_RB0_SetHigh();
76          }
```

Una volta osservata la variazione possiamo riavviare la simulazione tramite il pulsante start ( ).

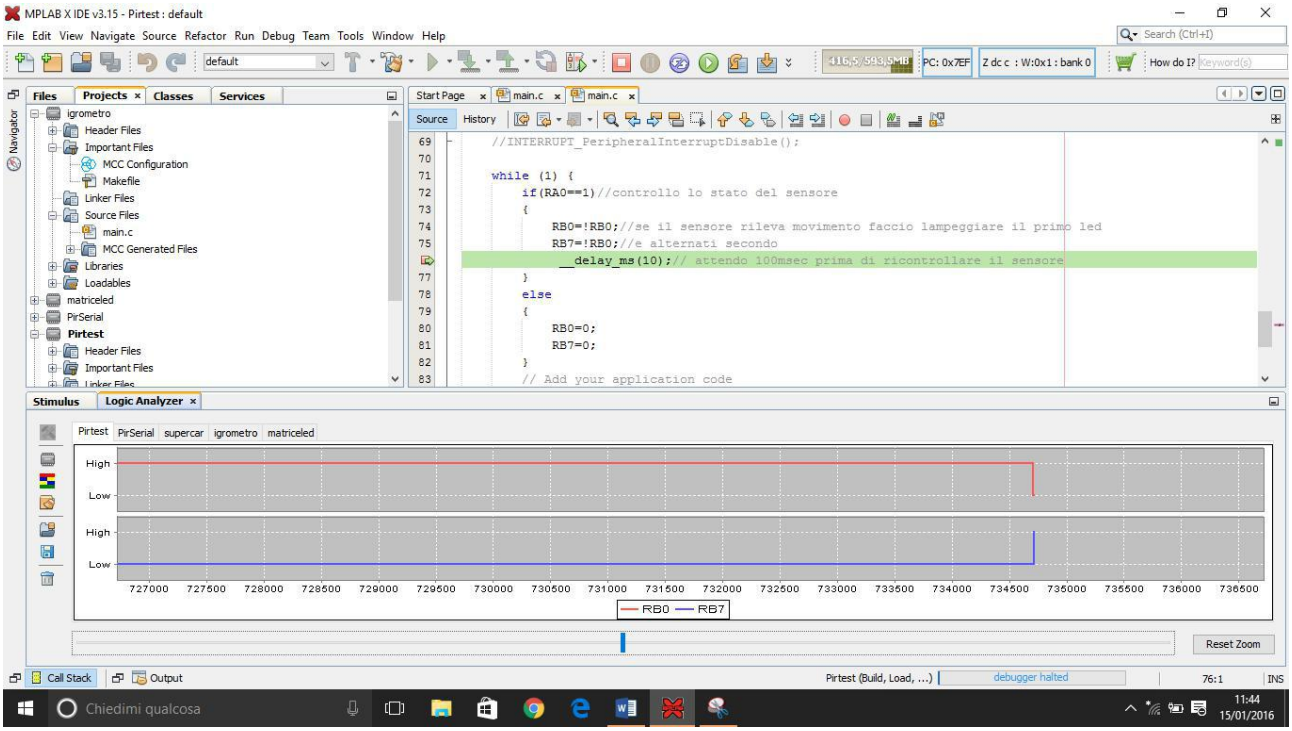
Esempio di un programma simulato:

Il programma in questione si chiama Pirtest ,prevede di collegare al pin RA0 un sensore di movimento PIR, il nostro sensore darà un segnale alto quando rileverà movimento e basso quando non rileva niente, perciò se il segnale in ingresso è alto il primo e l'ultimo led lampeggeranno in modo alterato altrimenti saranno spenti.

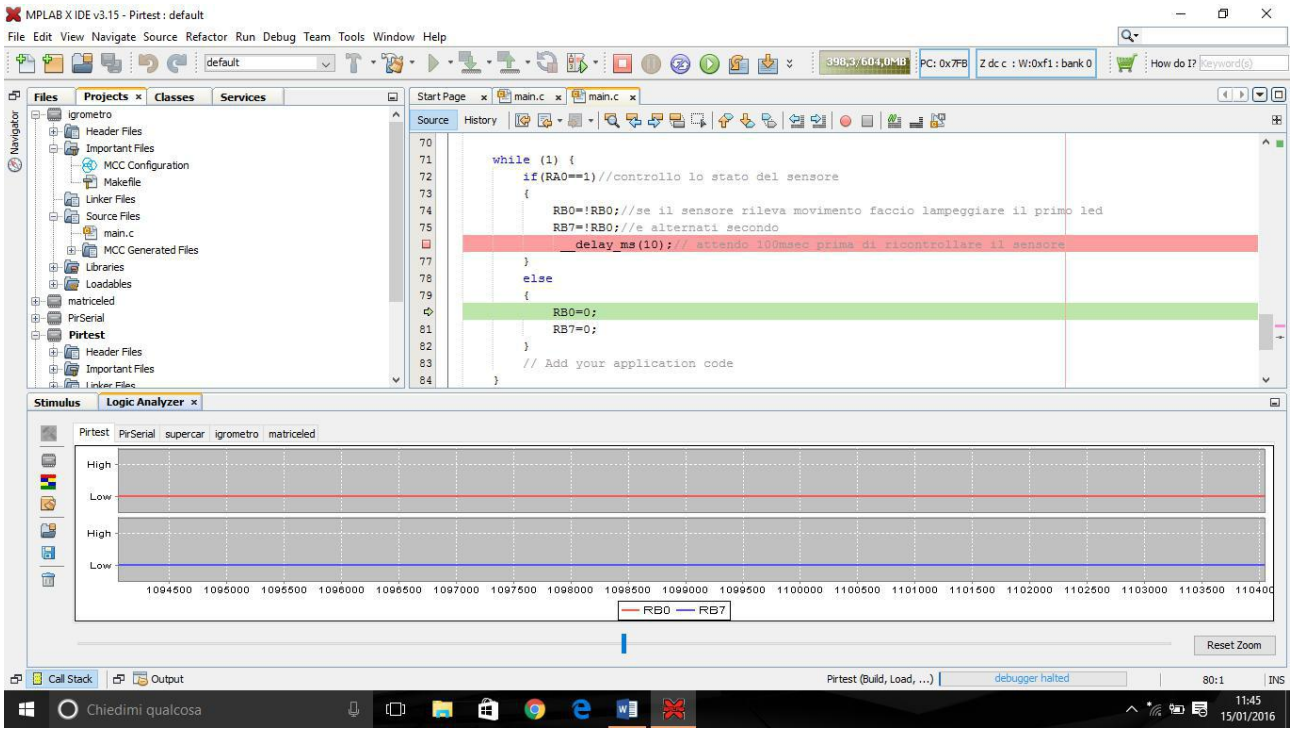


La nostra finestra si presenterà più o meno in questo modo , posizionando un breakpoint sulla riga del ritardo.

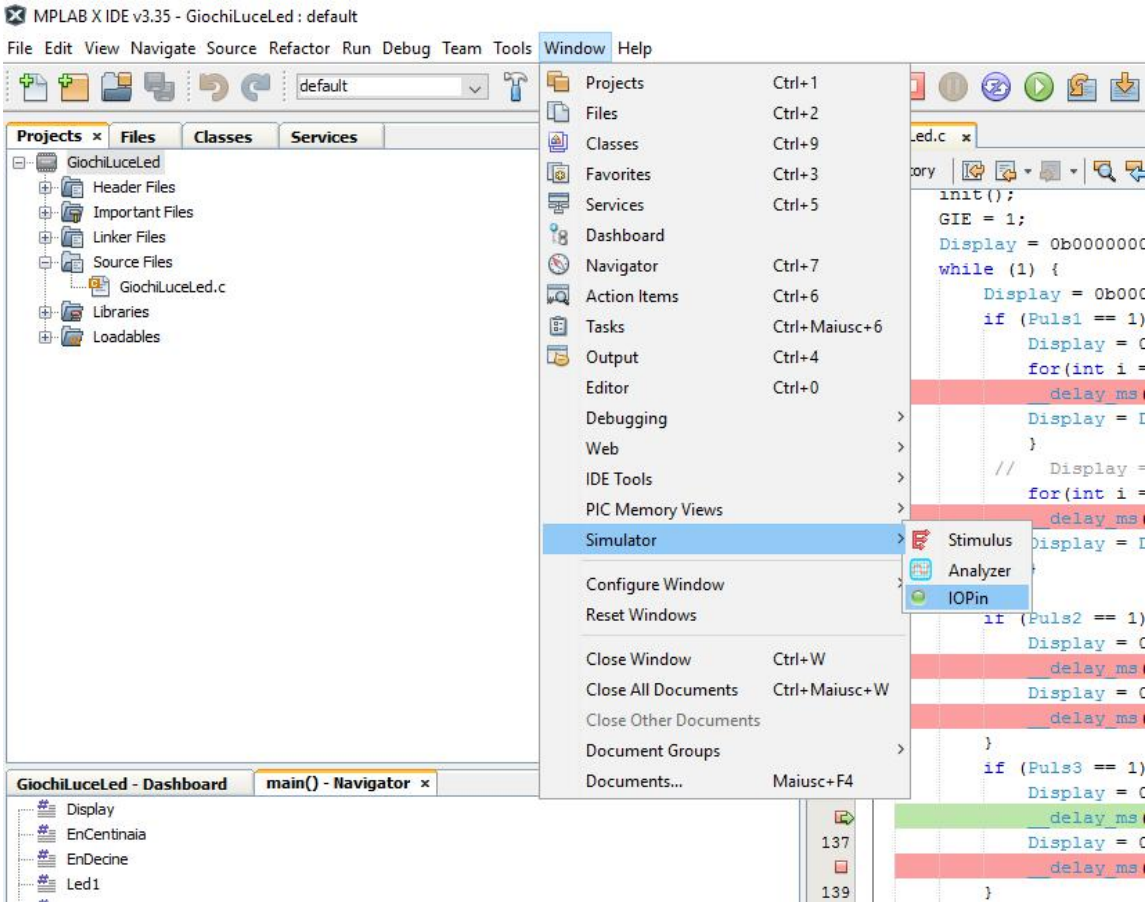
Portando lo stato del RA0 ad alto i due led inizieranno a lampeggiare e lo vedremo tramite il logic analyzer perché il breakpoint fermerà la simulazione.



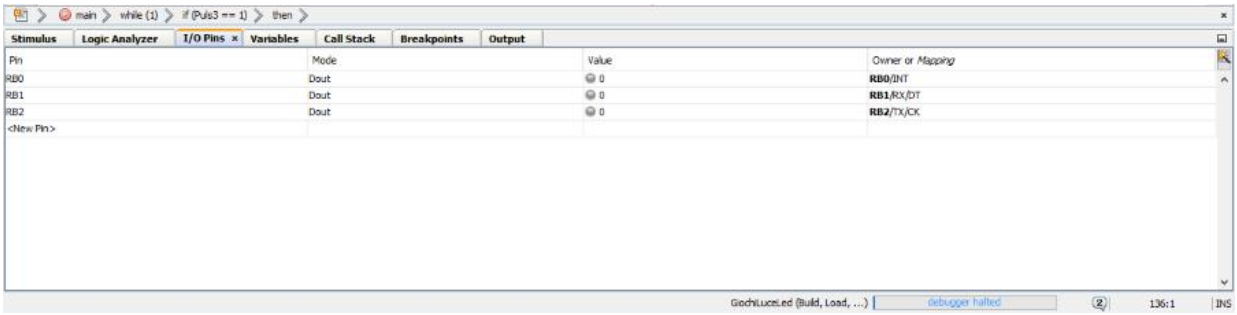
Portando lo stato dell'ingresso a 0, le due uscite saranno basse.



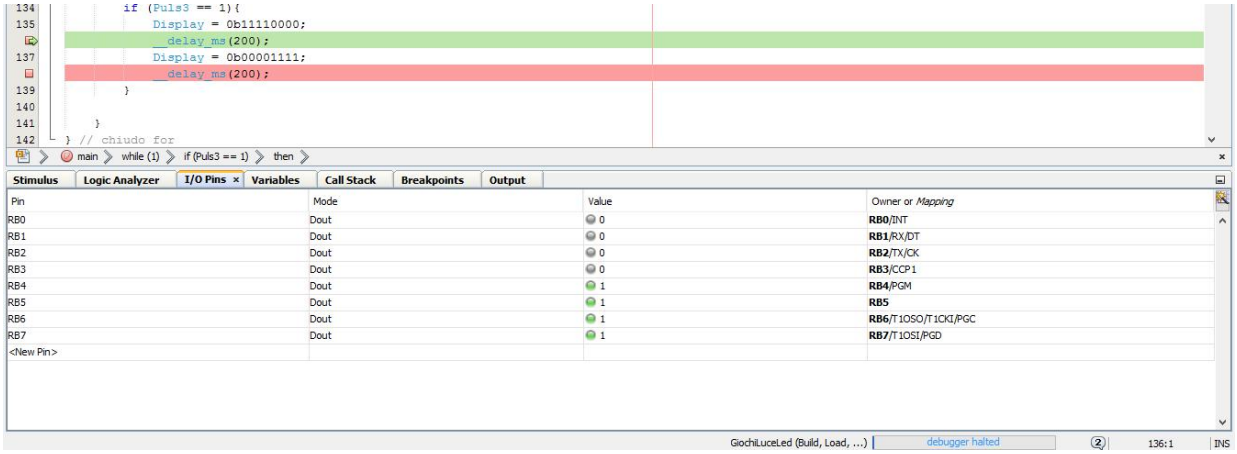
Esiste anche un'altra modalità di simulazione chiamata I/O Pins.
Il percorso è **Window -> Simulator -> IOPin.**

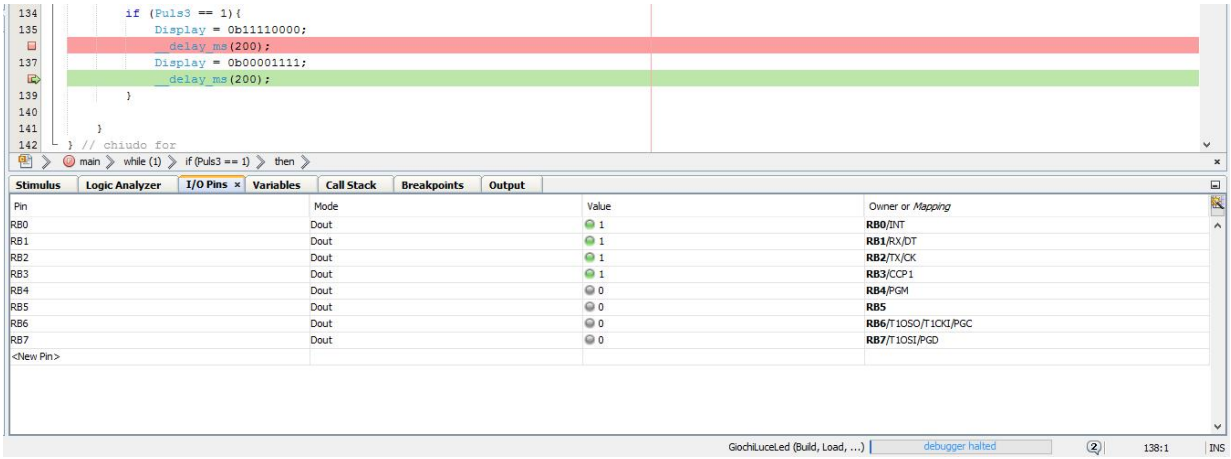


Si aprirà in basso una finestra dove dobbiamo selezionare i pin che ci interessano Cliccando su New Pin



Poi riavviare la simulazione col tasto Play( ) , usando sempre lo stesso metodo dell'Analyzer(Breakpoint) osserveremo i valori in uscita dai Pin





Porcheddu Lorenzo 4A 2016