SLIDE 1

Facciamo una piccola introduzione per dare un po’il contesto. Abbiamo immaginato di essere dipendenti presso un’azienda e stiamo proponendo al nostro superiore l’idea di un progetto che può diminuire i tempi necessari a scrivere la documentazione. Siamo Davide Salvetti e Matteo Verzeroli e siamo Firmware & Hardware Engineers presso l’azienda Omega.

SLIDE 2

Omega è un’azienda che lavora nell’ambito dello sviluppo di elettronica, in particolare ha grande esperienza nello sviluppo di prodotti IoT, Wearable e dispositivi per smart home. Cosa fa Omega? Omega di occupa di sviluppo Hardawre, Simulazioni di circuiti elettronici con LTSpice, sviluppo di firmware, analisi dei dati e sviluppo di applicazioni desktop per la visualizzazione dei dati raccolti. Abbiamo anche delle certificazioni e abbiamo diversi clienti anche all’estero.

SLIDE 4

Il problema con cui ci scontriamo ogni giorno è con la stesura della documentazione. Ogni volta che sviluppiamo un progetto dobbiamo corredarlo con una documentazione ricca e scritta in un certo modo. Questa documentazione include ovviamente anche circuiti elettronici. La rappresentazione dei circuiti elettronici all’interno della documentazione deve seguire delle regole ben precise che sono per la maggior parte dettate dal cliente, per fare in modo che la documentazione sia coerente con quella degli altri progetti e con le regole aziendali, ma anche per soddisfare degli standard.

Ovviamente, nella nostra azienda sono presenti diversi ingegneri ed ognuno utilizza software diversi in funzione della mansione specifica: per esempio l’hardware engineer utilizza Eagle e quindi ha la rappresentazione del circuito in Eagle, l’ingegnere che si occupa delle simulazioni invece utilizza LTSpice, il firmwarista utilizza invece altri software come draw.io. Quindi durante lo sviluppo ci sono rappresentazioni differenti tra lor che son mostrate qua di fianco.

Ovviamente per inserire questi circuiti nella documentazione non è possibile fare uno screenshoot, visto che utilizzano tutti convenzioni diverse e rappresentano i componenti in modo diverso.

Per questo motivo si utilizza un package di Latex, CircuiTikz, che permette di inserire il circuito in un documento latex in modo da seguire tutte le convenzioni e da usare un’unica rappresentazione per tutti. Questo processo, manuale, ha ovviamente i suoi svantaggi:

* È molto dispendioso in termini di tempo (e quindi anche di soldi). Ogni ingegnere quando scrive il suo pezzo di documentazione deve tradurre il circuito inserendo a mano i comandi in latex per la generazione del circuito;
* Il linguaggio del package CircuiTikz non è banale e può portare spesso ad errori durante la trascrizione;
* Inoltre, clienti diversi necessitano di utilizzare convenzioni e regole differenti, quindi gli ingegneri devono stare attenti alle richieste del cliente per la rappresentazione del circuito. Ad esempio, potrebbero esserci certificazioni che richiedono di seguire uno standard di rappresentazione.

SLIDE 5

Cosa proponiamo noi?

Proponiamo un prodotto con diversi vantaggi:

1. Il primo tra tutti, si tratta di una semplicissima interfaccia grafdica che può essere utilizzata da tutti i membri dell’azienda per la generazione dei circuiti da inserire nella documentazione.
2. Automatizza il processo di trascrizione dei circuiti, diminuendo il tempo necessario alla stesura della documentazione e diminuendo il numero degli errori che possono essere commessi.
3. È possibile rendere l’applicazione open source: questo porterebbe con se dei vantaggi, primo tra tutti quello che la community può contribuire migliorando il software e rendere più semplice il processo di integrazione nell’applicazione di nuove regole di rappresnetazione. In questo modo, se ci sono delle modifiche ad alcune regole di rappresentazione del circuito, sarà sufficiente scaricare la nuova versione del programma e i nostri ingegneri non devono preoccuparsi di altro.

Questo prodotto fantastico che noi proponiamo e che risolve tutti i nostri problemi lo abbiamo intitolato LTSpice2CircuiTikz. È ora Matteo vi spiega più nel dettaglio il suo funzionamento.

SLIDE 12

Qui abbiamo riportato un ipotetico diagramma di Gannt. Per avere una prima versione dell’applicazione, che magari non ha tutti i comandi possibili ma che sia comunque utilizzabile, abbiamo previsto circa 5/6 settimane, seguendo un modello di sviluppo classico a cascata con una fase di analisi dei requisiti, una fase di progettazione seguita dallo sviluppo, dal collaudo dai test di integrazione e poi dalla fase di manutenzione.

SLIDE 13

Inoltre, si può pensare comunque di aumentare i guadagni andando a commercializzare tale applicazione. Al momento non ci sono software che convertono in codice latex usando il package CircuiTikz e questo sicuramente è un vantaggio.

SLIDE 14

Abbiamo poi pensato che per lo sviluppo son sufficienti due persone, in particolare un software engineer per la progettazione del sistema e un developer per la sua implementazione.