

Village Telco e Mesh Potato

by David Rowe

Traduzione libera dal [blog](#) di David Rowe del 10 settembre 2008: Diego Serafin IK3HUK. Distribuito con licenza GNU FDL.

[Il Village Telco](#) è un kit per realizzare una compagnia telefonica fai-da-te, facile da usare, scalabile, basata su standard, wireless. Ci siamo trovati a Cape Town per capire come realizzarlo.

[Steve Song](#) della fondazione Shuttleworth ha aggregato un entusiasmante gruppo di persone che facevano parte delle comunità dello sviluppo software, del VOIP, del mesh networking, e del business. Il team era ristretto (circa 10 persone) ed era molto “pratico” come mentalità e abilità personali.

Una delle decisioni prese è stata quella di sviluppare il “*Mesh Potato*”.

Il *Mesh Potato* “in a nutshell”

Il *Mesh Potato* è un mesh router 802.11bg con una singola porta FXS (Porta a cui è possibile collegare un telefono analogico. N.d.t.). E' progettato per fornire telefonia VoIP e simultaneamente rendere possibile una “mesh cloud”. E' un progetto open hardware e open software. Funzionerà con una tensione nominale di 12VDC, fornita da un normale alimentatore da tensione di rete o da un sistema fotovoltaico, e il suo prezzo può essere stimato attorno ai prezzi correnti per un router WiFi (sotto i 100 dollari USA).

Caratteristiche salienti:

1. Utilizza il software di mesh routing [B.A.T.M.A.N.](#), Asterisk, il codec voce [Speex](#), e il cancellatore d'eco [Oslec](#) (Open Source Line Echo Cancellor – sviluppato dallo stesso David N.d.t.).
2. L'obiettivo finale è realizzare una rete VoIP “mesh routed”, in particolare (ma non solo) per le comunità in via di sviluppo. Si collega un telefono analogico al *Potato* attraverso la porta FXS. Quando si effettua una chiamata, il *Potato* comunica con un altro *Potato* sito nelle vicinanze, il quale a sua volta comunica con un altro *Potato*, e, finalmente, con il destinatario della chiamata. La rete mesh può essere potenziata utilizzando dei collegamenti di tipo “backbone” e può essere connessa con il resto del mondo attraverso gateway VOIP.
3. Non sono necessari tralicci di telefonia cellulare, linee telefoniche su cavo, né alcun grosso operatore telefonico. Un imprenditore locale può gestire il suo sistema “Village Telco” utilizzando un modesto server e qualche router *Mesh Potato*. La telefonia diventa così di proprietà della Comunità.
4. La rete mesh è auto-organizzante e auto-riparante, se un nodo cade B.A.T.M.A.N. automaticamente reindirizza le chiamate.
5. Stiamo costruendo hardware dedicato specificamente per le comunità in via di sviluppo utilizzando principi di hardware e software libero. Vorrei sottolineare il principio che è **possibile** sviluppare dispositivi open hardware custom – non è necessario utilizzare per forza quello che è disponibile in commercio. Come è noto la maggior parte dello sforzo per sviluppare qualsiasi dispositivo è nel software, così l'idea di appoggiarsi ad hardware chiuso, proprietario e non del tutto adatto allo scopo è **obsoleta**.
6. Il *Mesh Potato* è aperto, veramente **aperto**. Gli obiettivi chiave sono la minimizzazione dei “blob binari”, dei software proprietari e la realizzazione di un hardware aperto. Ciò implica che il *Potato* sarà probabilmente basato su chip Atheros, in quanto possiamo utilizzare il driver WLAN open source [madwifi](#). Significa anche [Speex](#) al posto del G.729, e [Oslec](#) al posto di un cancellatore d'eco proprietario. Come minimo saranno resi disponibili gli schemi

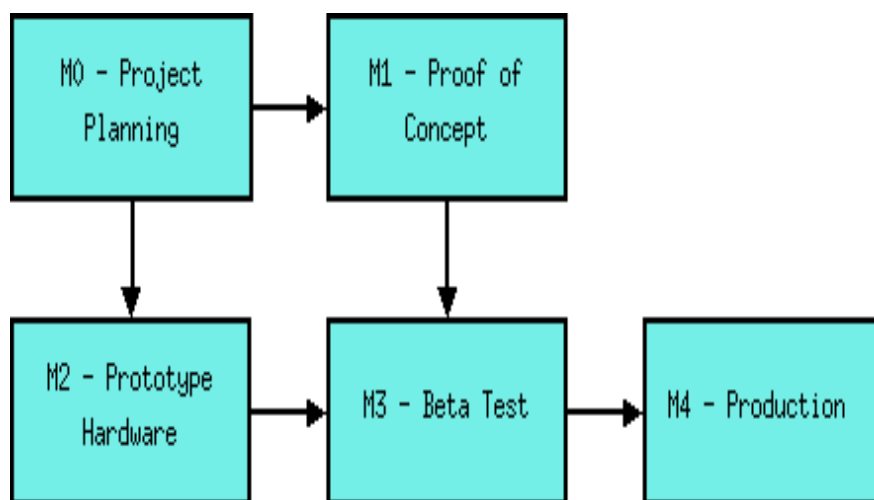
elettrici dell'hardware. Il *Potato* sarà prodotto in serie in grande quantità, quindi l'**open community** ha in questo caso l'opportunità di **dettare** lo standard, anziché dover sottostare a “standard” basati su hardware e software chiusi.

Per i curiosi, Steve descrive le origini del nome *Mesh Potato* in [questo](#) post (in inglese N.d.t.).

La [Shuttleworth Foundation](#) ha generosamente deciso di finanziare le fasi iniziali dello sviluppo del *Mesh Potato*. La Shuttleworth Foundation è presente in molti progetti degni di nota, come il [Freedom Toaster](#) e l'avvio del progetto [Ubuntu](#). Lo sviluppo del *Potato* sarà la mia attività principale per tutto il resto del 2008!

Progetto

Abbiamo deciso di dividere il progetto in 5 *milestone*, M0-M4. Cominciamo con lo 0 perchè siamo geek e ringraziateci che non abbiamo usato la notazione binaria. Abbiamo in gran parte completato la fase di pianificazione e abbiamo avviato lo sviluppo. Speriamo di avere prototipi del *Potato* hardware all'inizio del 2009.



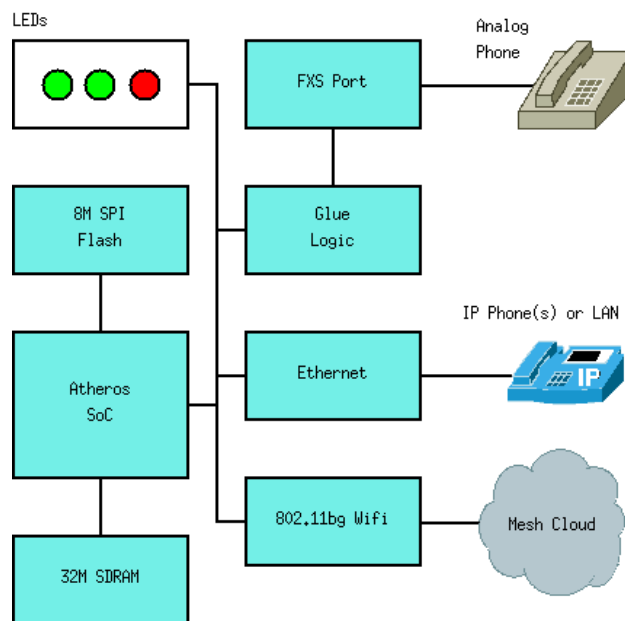
Attualmente stiamo lavorando su M1 - *Proof of Concept*. L'idea è di usare hardware disponibile in commercio - *Commercial Off The Shelf* (COTS) – come l'Ubiquity Nanostation 2 per sviluppare un prototipo che dimostri come tutte le componenti software funzionino assieme. Svilupperemo molte componenti software *custom* nella fase iniziale del progetto utilizzando una piattaforma hardware matura. Le principali componenti software dovranno essere abbastanza mature quando saranno integrate nell'hardware prototipale, risparmiando tempo e abbassando il rischio complessivo.

Nelle ultime settimane ho sperimentato OpenWRT sulla Nanostation 2 – lanciando programmi di test per Speex ed Oslec e caratterizzando il carico della CPU.

[Atcom](#) è entusiasta di lavorare sui prototipi hardware, e di essere il produttore di massa del *Mesh Potato*. L'Atcom è stata pioniera nello supportare prodotti Open Hardware come il centralino telefonico [IP04](#).

Architettura Hardware

Ecco uno schizzo del progetto hardware:



Una bella sfida sarà collegare il chipset FXS al SoC (SystemOnChip) Atheros (la *glue logic* nella figura qui sopra). Il chipset FXS richiede un collegamento seriale sincrono TDM più un bus SPI. Il bus TDM è tipicamente un bus seriale a 2.048MHz che fornisce in ogni time slot di 8-bit un campione di voce ogni 125µs (8kHz). Il bus SPI è utilizzato per il controllo e la configurazione del chip set FXS (per esempio per rilevare lo sgancio della cornetta, attivare il generatore di tensione di squillo, etc.).

Questa logica di collegamento (*glue logic*) è stata molto facile da realizzare per l'IP04 in quanto la CPU Blackfin offre un insieme ricchissimo di interfacce con un buon supporto DMA. Il SoC Atheros non è così ricco. Ha un bus SPI bus per dialogare con la memoria flash SPI. Quest'ultima ha, però, una sola linea di chip select anche se prevedo di poter utilizzare qualche segnale GPIO libero per accedere ad altri device collegati al bus SPI.

Dobbiamo poi costruire un'interfaccia TDM di qualche tipo. Per ridurre il sovraccarico di interrupt e l'*I-cache thrashing* sarebbe interessante inserire 1-20 ms di campioni voce in un buffer (da 8 a 160 campioni da 8-bit) prima di interrompere l'Atheros SoC. Questa interfaccia potrebbe essere realizzata con un po' di logica o forse addirittura con un piccolo [micro-controllore](#) come un PIC. Un'altra possibilità è interfacciare il bus TDM ad una UART RS232. Il SoC Atheros ha infatti un UART 16550 compatibile con un po' di FIFO hardware, comunque avremmo bisogno di qualche test per stabilire se il sovraccarico della CPU dato dall'utilizzo dell'UART è accettabile.

Partecipa

Il *Mesh Potato* promette di essere qualcosa di veramente speciale – un mesh router WiFi open hardware/software, progettato dalla *community*, equipaggiato con Asterisk ed una porta FXS. Ragazzi, sono rimasto senza fiato nel dirlo. Non serve chiedersi perché lo chiamiamo solo *Mesh Potato*. Il concetto di *Village Telco* che supporta potrebbe portare il telefono a milioni di persone. Per favore aderisci liberamente alla nostra *community* – iscrivendoti al [Village Telco Google Group](#).