

Elettronica & Maker

n° 11 - 2019

***Giociamo con
i Regolatori Switching***

***AD8232 + Sensore IR
per ECG con Arduino***

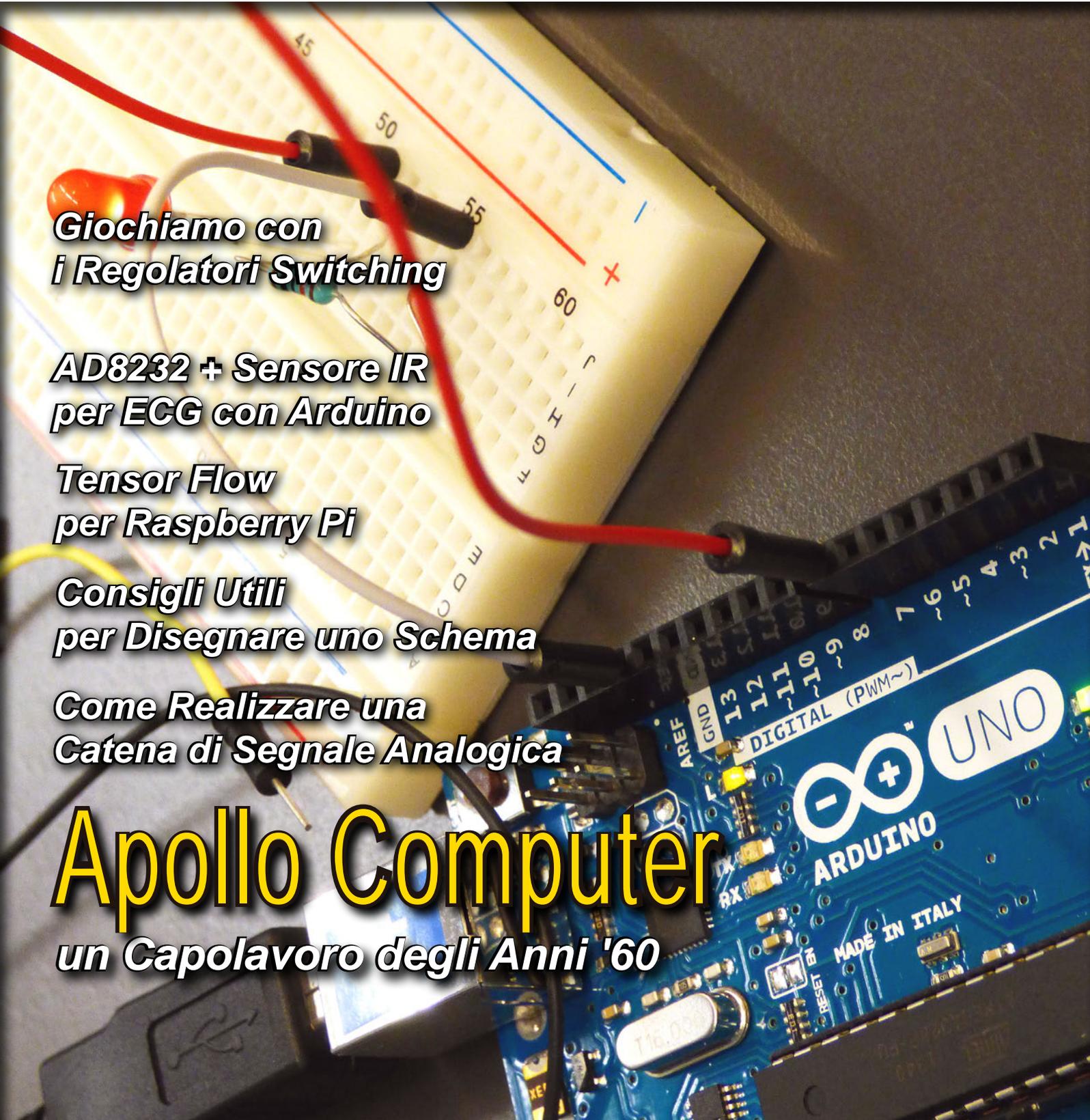
***Tensor Flow
per Raspberry Pi***

***Consigli Utili
per Disegnare uno Schema***

***Come Realizzare una
Catena di Segnale Analogica***

Apollo Computer

un Capolavoro degli Anni '60



Elettronica Maker

EDITORIALE

r.armani@elettronicaemaker.it



Elettroflop

Di questi tempi si parla molto di veicoli elettrici, forse fin troppo. In Italia, alle tante, troppe parole spese seguono soltanto pochi fatti concreti. Tanto per cambiare, nella migrazione verso i cosiddetti "sistemi di trasporto eco-sostenibili", in Europa il nostro paese si trova all'ultimo posto. A volte, però, essere ultimi può rappresentare un vantaggio; come quello, primo fra tutti, di poter analizzare a costo zero la situazione in Paesi che, da questo punto di vista, sono estremamente evoluti. Alludo alla Norvegia, dove la percentuale di veicoli elettrici ha superato il 50% delle immatricolazioni complessive, non senza problemi e con notevoli conseguenze per l'utenza. Il Governo Norvegese sta subendo l'impatto finanziario dell'incremento della spesa per gli eco-incentivi all'acquisto, al quale ha corrisposto una drastica riduzione del gettito fiscale che derivava dall'uso dei veicoli convenzionali con motore termico. Di conseguenza, lentamente ma progressivamente, per risanare i bilanci si stanno sopprimendo le agevolazioni applicate agli esordi, come ricarica batterie a costo zero o quasi, uso delle corsie preferenziali e accesso ad aree limitate, soste gratuite, esenzioni da tasse di possesso etc. Sono solo alcuni esempi dei vantaggi che sono in fase di eliminazione, aumentando così in modo indiretto il costo complessivo di gestione di questi automezzi. A rincarare la dose e peggiorare ulteriormente la reputazione del mezzo elettrico ci pensa la più grande compagnia di assicurazioni Norvegese, che ha riscontrato negli EV un livello di sinistrosità superiore del 20% rispetto ai classici diesel o benzina. Un fatto dovuto, molto probabilmente, alle alte capacità di accelerazione da fermo del motore elettrico che, se mal controllate, aumentano i rischi di collisione. Per non parlare, poi, delle grandi difficoltà di gestione, a volte quasi insormontabili, vissute ormai perennemente

dalla rete di distribuzione elettrica nazionale. Anche in Paesi extraeuropei come la Svizzera la situazione non è migliore; il gettito fiscale da carburanti diminuisce e il Consiglio Federale (il 18 Dicembre) ha ipotizzato l'istituzione di una "tassa al consumo", basata sui chilometri percorsi. Tutto lascia pensare che, almeno per il momento, valga la pena di...restare sul "buon vecchio pistone"!

Roberto Armani

Ora E&M è anche su:



Seguici!

Due Parole...

4 **Apollo Computer**

Maker

15 **AD8232 + Sensore IR per ECG con Arduino**

40 **Giochiamo con i Regolatori Switching**

Strumenti

56 **Consigli Utili per Disegnare uno Schema**

64 **Come Realizzare una Catena di Segnale Analogica**

Arduino, Raspberry & Co.

28 **Tensor Flow per Raspberry Pi**



Apollo Computer

di Mario Rotigni

m.rotigni@elettronicaemaker.it

A cinquant'anni dal primo sbarco sulla Luna, diamo un'occhiata da vicino al computer di bordo delle missioni Apollo, Apollo Guidance Computer (AGC). Cerchiamo anche di capire se veramente aveva meno potenza di calcolo di una banale calcolatrice tascabile dei giorni nostri, come a volte si sente dire.

Il lettore perdonerà l'esordio autobiografico, inusuale ma inevitabile in questo caso. L'autore aveva 11 anni quel 20 Luglio del 1969 e tutta la sua attenzione ed entusiasmo di bimbo erano catturati da quell'evento storico. Come molti in quel periodo, naturalmente voleva diventare astronauta, aspirazione presto logorata dalla progressiva conoscenza dei requisiti necessari. Certamente l'evento ha avuto influenza sulla scelta della professione nella vita, anche dopo aver constatato che ben difficilmente avrebbe sperimentato personalmente la formidabile spinta del Saturno V. A quel bimbo, a quell'entusiasmo e desiderio di "andare oltre", a tutti coloro che, bimbi o no, lo hanno condiviso e lo condividono tuttora, è dedicato questo articolo. Oggi Internet e la politica di divulgazione della NASA rendono possibile avere accesso diretto ad una sterminata quantità

Le immagini di questo articolo sono state ricavate dal sito NASA e utilizzate nel rispetto della politica di utilizzo della stessa.

di informazioni sulle missioni spaziali ed i mezzi impiegati. L'autore ha speso molte ore divertenti ed interessanti a zonzo negli archivi NASA ed invita il lettore a fare la stessa esperienza (si potrebbe iniziare da: <https://history.nasa.gov/computers/contents.html> e <https://history.nasa.gov/afj/ap11fj/index.html>).



Giugno 1969, Neil Armstrong si esercita nel simulatore del Modulo di Discesa Lunare

CONFRONTO TRA COMPUTER

Esploriamo un po' i termini della pretesa scarsità del computer di bordo delle missioni Apollo, per iniziare. In generale, confrontare computer è un'attività tutt'altro che semplice, come ben sa chi sia stato coinvolto anche soltanto in benchmark di microprocessori diversi. L'immaginario collettivo ha inoltre sviluppato aspettative molto

sofisticate per computer destinati ad applicazioni spaziali. Film e serie TV ci hanno abituato a computer che interagiscono con gli umani servendosi del linguaggio naturale, macchine in grado di svolgere i compiti più diversi e fornire risposte alla ricerca di informazioni qualsiasi. Questo scenario pone già in una luce dimessa il pezzo di tecnologia anni '60 di cui ci stiamo occupando. Per cercare di farci comunque un'idea, proviamo a confrontare le caratteristiche di AGC con un piccolo, moderno computer, Raspberry Pi4. La Tabella 1 in questo articolo accosta le performance e caratteristiche delle due ma-



	Raspberry Pi4	AGC Apollo Guidance Computer
Anno	2019	1966
CPU	Broadcom BCM2711B0 quad-core A72 (ARMv8-A) 64-bit + GPU Broadcom VideoCore VI	Cablata con Circuiti Integrati
Lunghezza parola binaria	64 bit	16 bit (15+1 parità)
Frequenza clock	1,5 GHz	2,048 MHz
Velocità esecuzione		85000 istruzioni al secondo
Tecnologia	CMOS 28nm	RTL (Resistor Transistor Logic)
RAM	1GB a 4GB LPDDR4 SDRAM	4096 byte magnetica
Memoria di massa	microSD	ROM 73,7 kbyte magnetica
General Purpose I/O	40	nessuno
Porte di Comunicazione	2 micro-HDMI 2.0 (doppia uscita video) 3 uscita video analogica più audio jack 3,5mm 2 USB 2.0 2 USB 3.0 Camera Serial Interface (CSI) Display Serial Interface (DSI)	DSKY IMU Hand Controller Rendezvous Radar (CM) Landing Radar (LM) Telemetry Receiver Engine Command Reaction Control System (Vedi Articolo)
Comunicazione Wireless	2,4GHz e 5 GHz 802.11b/g/n/ac wireless LAN	RTX VHF
	Bluetooth 5.0, (B Low Energy)	
Potenza consumata	7,6 W	55 W
Dimensioni e peso	88 x 58 x 19,5 mm, 46 g	61 x 32 x 17 cm, 32 kg
Unità prodotte	In produzione di serie	57

Tabella 1

chine. Chiaramente, le prestazioni di AGC sembrano fare ben misera figura, confrontate ad una macchina con 5 unità di calcolo (comprendendo anche la Graphics Unit), frutto di 40 anni di tecnologie di integrazione su silicio. Anche nelle prestazioni base tuttavia l'occhio attento può cogliere dettagli interessanti. La lunghezza di parola della CPU risulta non di 16 ma di 15 bit utili, destinando il sedicesimo al controllo di parità. Finezza non presente nella maggioranza dei computer in circolazione ancor oggi (dove in genere la parità viene applicata alle memorie ma non direttamente nell'unità di elaborazione stessa). La memoria di massa è una ROM, memoria non alterabile. Non è quindi prevista la possibilità di caricare ed eseguire un programma qualsiasi a posteriori, dopo la costruzione e collaudo del computer. Ma dove realmente le prestazioni divergono radicalmente, è nella serie di porte di comunicazioni gestite. Pi4 offre le classiche, più o meno sofisticate, interface general purpose o dedicate all'interfaccia utente. AGC offriva invece una serie di interface dedicate e specifiche alla sua missione spaziale, consentendo di dialogare con la sofisticata strumentazione di bordo. Qui AGC rivela la sua vera natura, quello che oggi definiremmo un "controllore embedded", dedicato alla gestione di un certo

numero di sottosistemi dell'astronave Apollo, non una unità di elaborazione e calcolo general purpose. Bisogna inoltre sottolineare come molti sottosistemi venivano gestiti simultaneamente (multi-tasking), tenendo conto delle rispettive priorità ed abbastanza rapidamente da reagire tempestivamente ad eventi segnalati dai sensori (real time). Un confronto corretto andrebbe eventualmente fatto con analoghi sistemi delle missioni Shuttle o apparecchiatura avionica di un velivolo, non certo con un calcolatore da tavolo. Anche nei pochi dati analizzati, è evidente che affidabilità, semplicità e sicurezza hanno avuto una considerazione fondamentale nella progettazione e costruzione, requisiti certo non rilevabili in PC ed affini. Al di là del folklore, un confronto ed un giudizio non sono così scontati. Vedremo in seguito come questo venga confermato da applicazioni posteriori alle missioni Apollo. Ora entriamo più in dettaglio, vedendo la composizione di AGC.

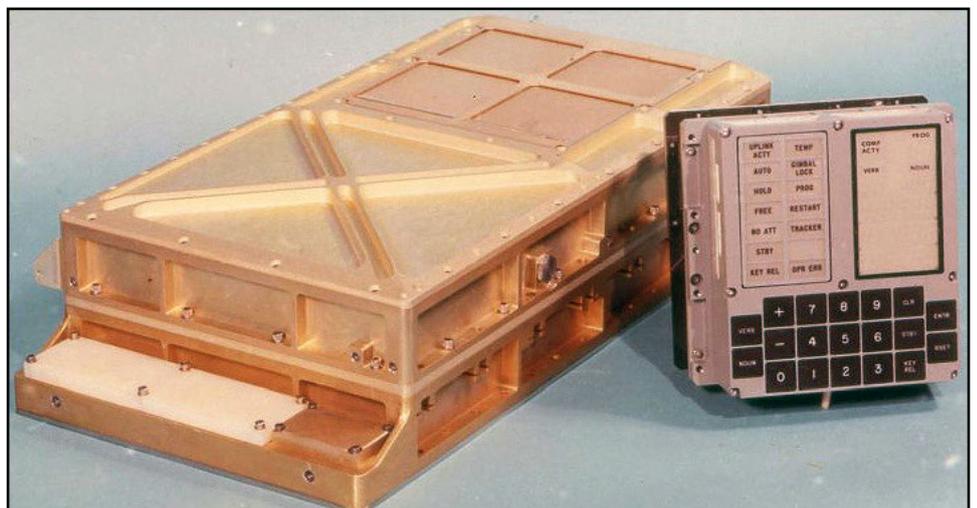


Figura 1: AGC e DSKY



APOLLO GUIDANCE COMPUTER (AGC)

AGC non è nato dal nulla per servire il programma Apollo, tutte le missioni spaziali hanno avuto a bordo qualche genere di controllo automatico, in forma di computer programmato o di circuiteria cablata. Lo stesso missile A4 (V2), considerato antenato di tutti i successivi missili per applicazioni militari e spaziali, aveva già un sistema di navigazione. AGC in particolare venne definito all'inizio del 1961 da NASA-Lan-

gley Space Center e Instrumentation Laboratory del Massachusetts Institute of Technology (il famoso MIT), a partire dalle esperienze dei sistemi di guida dei missili Polaris del decennio precedente.

La NASA voleva un computer a bordo per varie ragioni, non tutte pienamente realizzate nell'implementazione ed applicazione del progetto. Si voleva principalmente aumentare l'autonomia dell'astronave per tener conto del ritardo nelle comunicazioni

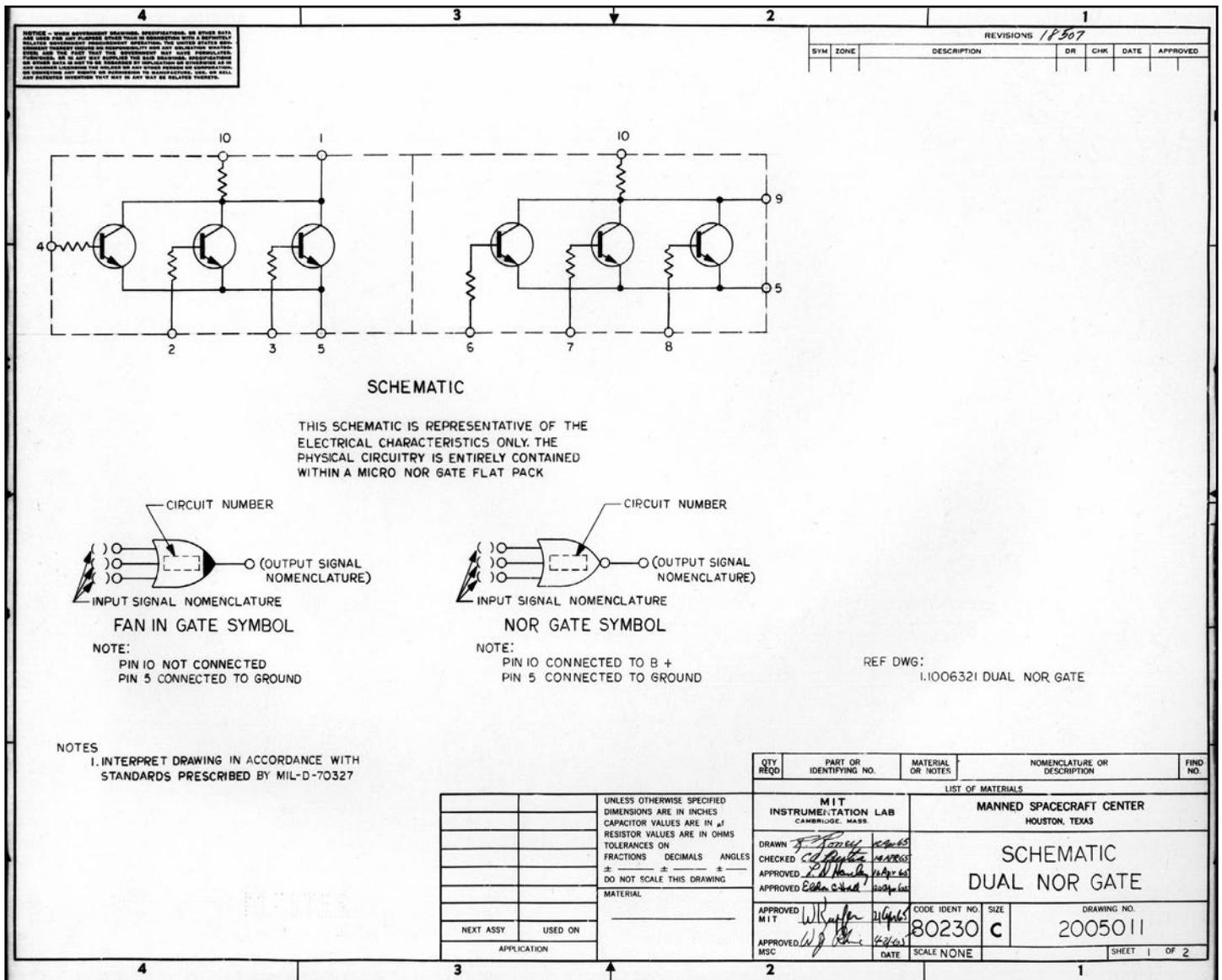


Figura 2: Dagli schemi elettrici originali, Circuito Integrato Dual NOR in logica RTL

con la Terra (fino ad 1,5 sec) e della possibilità di eseguire manovre fuori vista, nel cono d'ombra della Luna. In realtà i computer a Terra hanno avuto il ruolo principale nel calcolare la posizione e l'inserimento in orbita lunare, lasciando ad AGC la capacità di riportare a Terra autonomamente la capsula in caso di necessità. Pienamente operative sono state invece le altre funzioni assegnate ad AGC, gestione dei componenti del sistema di assetto, navigazione e guida, la generazione dei segnali di temporizzazione per almeno altri 20 sistemi dell'astronave. Secondo la stessa NASA, AGC è stato il primo computer al mondo realizzato con circuiti integrati, tecnologia introdotta da Fairchild Semiconductors solo tre anni prima. I circuiti integrati usati erano logiche NOR a tre ingressi, in tecnologia RTL (Resistor Transistor Logic). La pagina Wikipedia (https://en.wikipedia.org/wiki/Resistor%E2%80%93transistor_logic), dedicata alla tecnologia RTL, riporta fotografie del chip originale e dei circuiti integrati montati sui circuiti stampati di AGC. 5000 circuiti integrati NOR sono serviti a comporre AGC, permettendo di realizzare registri dati, registri di controllo (Program Counter, address pointer...), interfaccia di memoria, gestione interrupt, contatori, 6 timer, canali I/O ed un set di 41 istruzioni base. Due timer con risoluzione di 10 ms. erano riservati all'orologio di missione, che scandiva il tempo trascorso da un istante zero precedente di alcune ore il lancio.

Altri quattro timer avevano risoluzione rispettivamente di 10 ms, 10 ms, 100 ms e 625 μ s, per la gestione della schedulazione di esecuzione delle task principali e delle interfacce. La risoluzione dei timer potrebbe sorprendere con soli 625 μ s.

Tutto però era progettato in funzione delle esigenze reali dei sistemi sotto controllo, senza sprechi di risorse. I 625 μ s ad esempio erano stati scelti per permettere un controllo fine dei razzi di correzione dell'assetto che prevedevano un tempo di accensione minimo di 14 ms. Il set di istruzioni, il linguaggio assembler di AGC, comprendeva operazioni aritmetiche e logiche, istruzioni di controllo e sequenza (salti condizionati e non), lettura/scrittura/movimento dati, gestione I/O, gestione degli



Figura 3: Unità DSKY in funzione

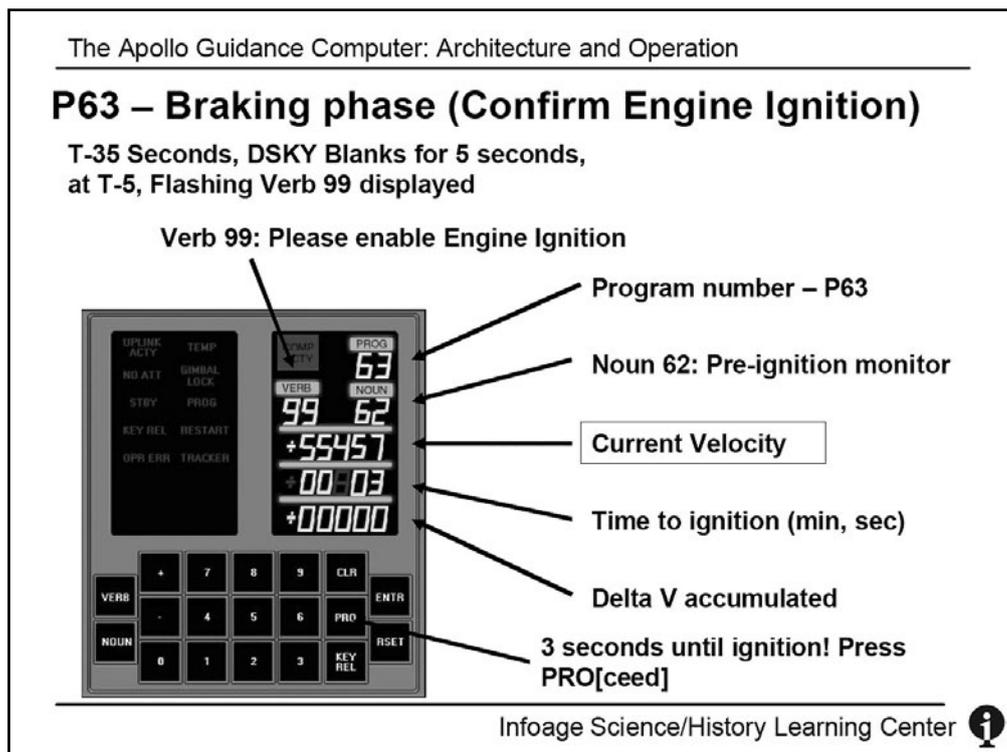


Figura 4: Note sul display durante l'esecuzione del programma 63

interrupt e dei contatori. Vista la missione di controllo affidata ad AGC, le operazioni logiche erano particolarmente flessibili sui dati dei canali I/O. Sorprendentemente per molti lettori, non era prevista una conferma

il passaggio di parametri a subroutine. Questo ha richiesto particolare cura e mole di lavoro nella stesura del software, obbligando all'uso di variabili globali, permanenti e visibili a tutto il programma, im-

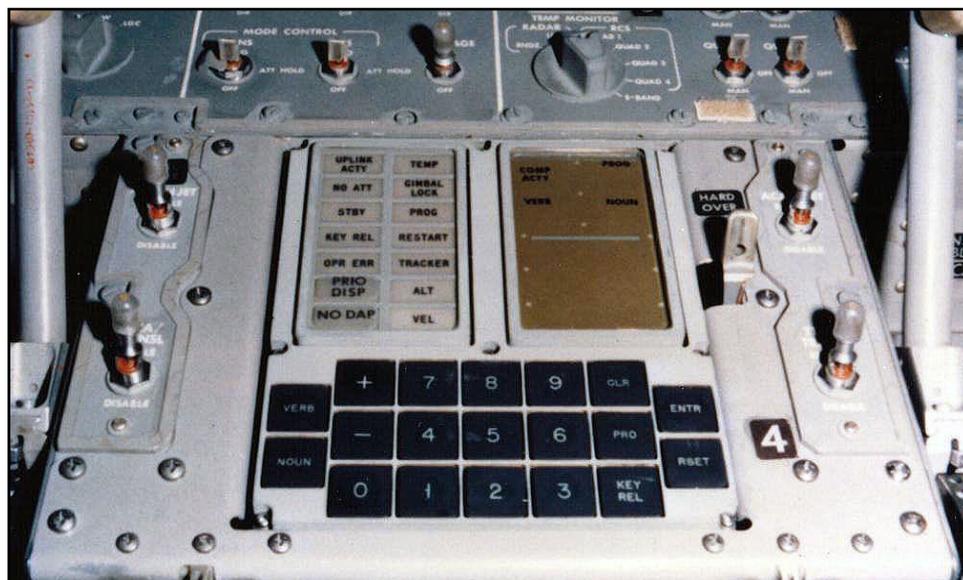


Figura 5: DSKY inserito nella console di comando

della corretta ricezione ed interpretazione dei dati/comandi inviati (c'è una sola eccezione, il RADAR di discesa del Modulo Lunare). Troviamo quindi quasi tutte le strutture ed istruzioni tipiche dei moderni microcontrollori integrati, con almeno una notevole eccezione. Lo stack pointer fu sacrificato per ragioni di semplificazione hardware,

non prevedendo strutture dedicate a supportare

il passaggio di parametri a subroutine. Questo ha richiesto particolare cura e mole di lavoro nella stesura del software, obbligando all'uso di variabili globali, permanenti e visibili a tutto il programma, imponendo anche economia nelle chiamate a subroutine.

Un sistema di controllo è di per se cieco e sordo al mondo esterno. La sezione di interfaccia I/O lo mette in comunicazione con sensori e strumenti che gli forniscono i dati necessari per conoscere la posizione, intensità e direzione delle accelerazioni

cui il veicolo è sottoposto. La stessa sezione collega controllore e controllato cioè i vari attuatori che gestiscono i sistemi di propulsione e correzione per mantenere assetto e rotta desiderati. Il principale problema risolto dai progettisti è stato la riduzione della enorme massa di dati a priori necessaria per gestire un sistema così dinamico ed esposto a sollecitazioni. Invece che una misura diretta di ogni grandezza e successiva conversione in binario, cui si potrebbe di primo acchito pensare, ogni strumento forniva un'informazione binaria di incremento o decremento, rispetto al valore iniziale di calibrazione. Contando gli impulsi ricevuti, noto il peso di ogni impulso elementare ed il valore di calibrazione, in ogni istante è calcolabile il valore della grandezza di interesse con il minimo volume di dati gestiti e tempo di elaborazione. Ad ognuna delle grandezze fondamentali era quindi associata una risorsa hardware, uno o più contatori di opportune dimensioni. Molti attuatori richiedevano inoltre un

semplice consenso on/off, un singolo bit di registro. E' questo il caso del propulsore del Modulo di Comando, un consenso passato al sistema di gestione del combustibile abilitava il rilascio dei componenti conservati in serbatoi separati. La miscelazione innesca direttamente la combustione e la spinta. La sezione di I/O gestiva così in modo puramente digitale i motori principa-



Figura 6: Margaret Hamilton, responsabile dello sviluppo software, di fianco ad una parte dei listati del programma AGC Apollo 11



li, i razzi di correzione assetto, telemetria, comunicazione VHF, Inertial Measurement Unit (IMU), lo Star Direction and Navigation (sestante), i segnali di controllo manuale, il pannello principale ed il pannello di navigazione (DSKY). Vediamo più da vicino alcuni di questi strumenti. IMU era una piatta-

forma inerziale composta da tre giroscopi, in grado di riportare l'angolo di rotazione attuale in rapporto alla posizione di riferimento (calibrazione) su ogni asse (X,Y,Z). Tre accelerometri integrati consentivano di riportare l'eventuale accelerazione sull'asse stesso. Il VHF range communication

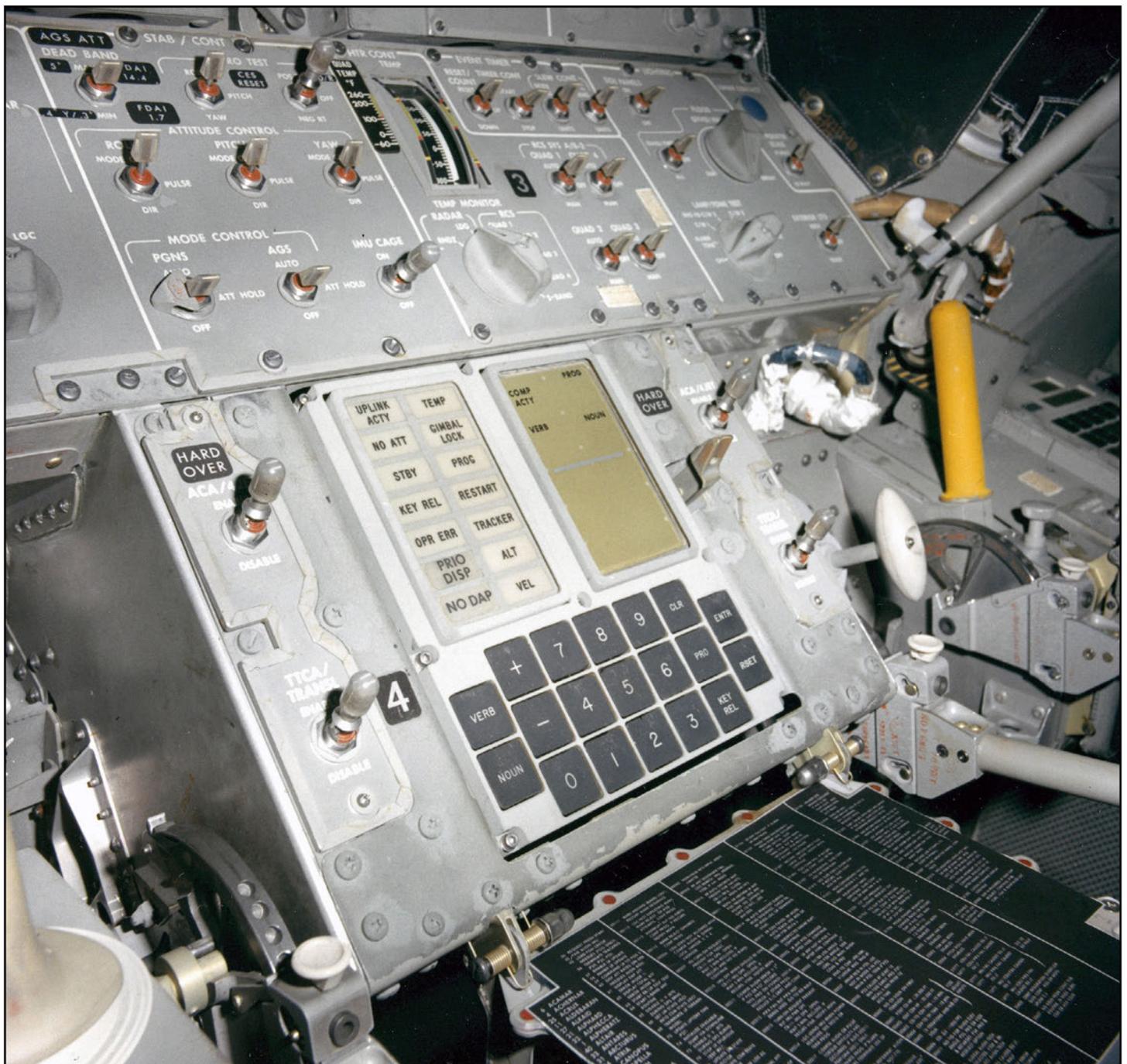


Figura 7: Parte inferiore del pulpito: si noti il supporto con le informazioni dettagliate su DSKY

comunicava la distanza tra Modulo di Comando e Modulo Lunare durante le manovre, svolgendo quindi un ruolo fondamentale nello sgancio prima della discesa sulla Luna e nel riaggancio a missione compiuta. Il pannello principale consentiva all'equipaggio la lettura dei vari parametri ricevuti

dagli strumenti. Per facilitarne l'interpretazione, localmente i dati digitali ad esempio relative all'altimetro, venivano convertiti in una presentazione analogica su strumenti con cursore, rendendo evidente la lettura del trend, più che del valore istantaneo. Il sestante permette di calcolare l'errore di

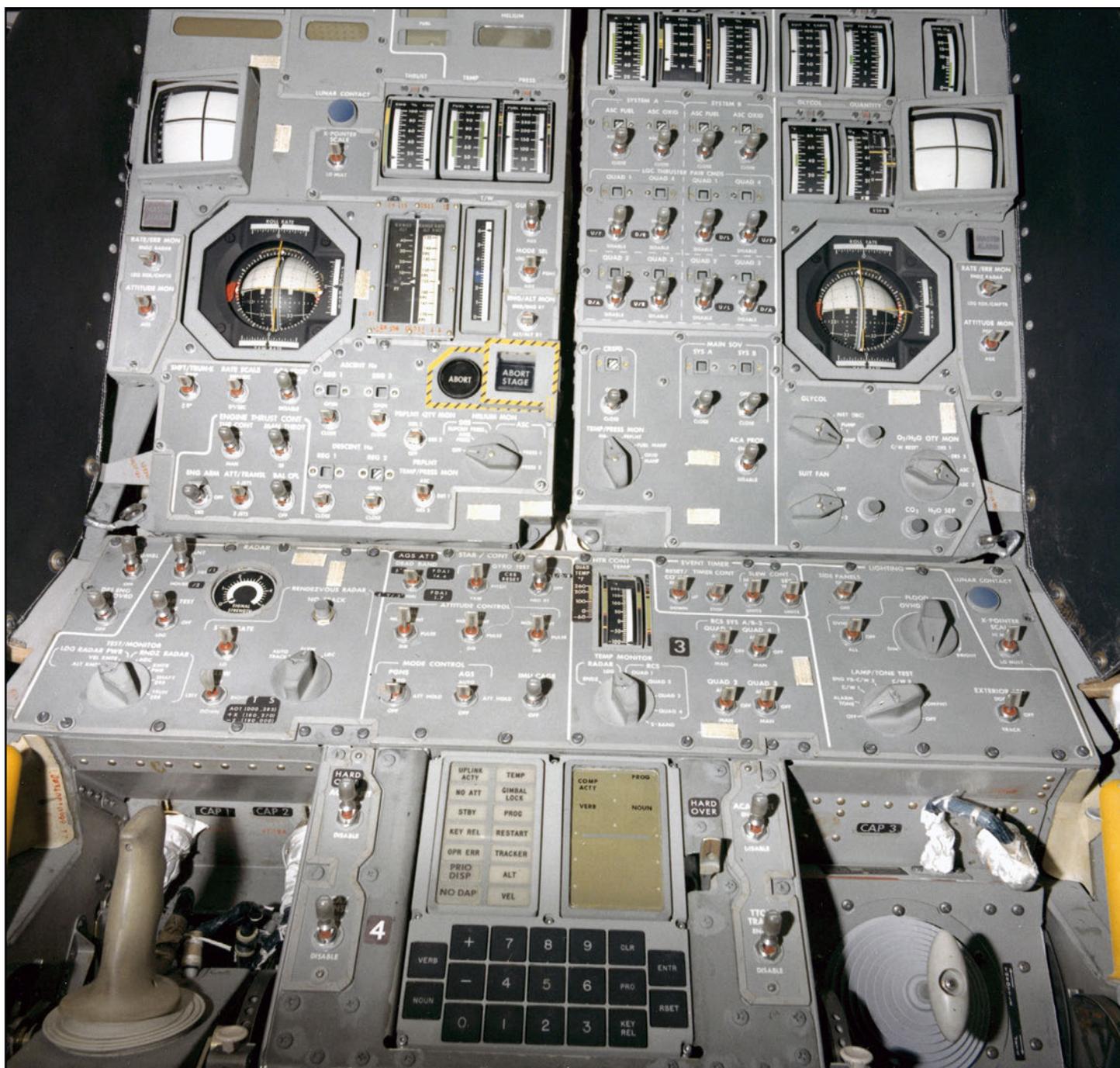


Figura 8: Vista complessiva della console di comando e controllo del modulo lunare



rotta servendosi delle posizioni di un certo numero di stelle, note ad AGC, contribuendo alla navigazione così come da secoli si fa in mare. L'unità Display and Keyboard (DSKY) era l'interfaccia principale tra AGC e l'equipaggio. La maggior parte delle fotografie riprodotte quando si parla di computer delle missioni Apollo, riproducono in realtà DSKY. Come si vede nell'immagine che li mostra affiancati, AGC ha molto poco appeal, ripreso da solo. L'interazione tra computer ed equipaggio via DSKY era normalmente iniziata dall'equipaggio e gestita da un modulo di programma dedicato. Quattro modalità operative erano disponibili:

- Display di dati
- Caricamento manuale di dati
- Lancio di programmi in background (che non richiedono lettura/scrittura di dati)
- Lancio dei programmi principali di gestione delle fasi della missione

Lo scambio di informazioni era gestito da una coppia di parole definite "verbo" e "nome". Come in grammatica, il verbo definisce l'azione da compiere, il nome definisce a che cosa l'azione è applicata. L'equipaggio poteva richiedere la visualizzazione di qualsiasi registro o locazione di memoria, in caso di eventuale interesse. Il pannello in alto a sinistra "ACTIVITY" riportava attraverso segnalazioni luminose l'attività del computer e della telemetria.

Sia il Modulo di Comando che il Modulo Lu-

nare erano dotati di AGC, con programmi diversi per la gestione dei differenti sistemi di propulsione, strumentazione e diverse operazioni da compiere. Oltre al suo AGC, il LEM aveva anche un computer supplementare, Abort Guidance System (AGS). In caso di guasto AGC durante la discesa, risalita o riaggancio, AGS poteva assumere il controllo, abortendo la discesa e assistendo il ricongiungimento con il Modulo di Comando. Non era previsto back-up per la manovra di discesa. Il sistema di computer Apollo comprendeva quindi tre computer, due AGC ed un AGS. Una copia di AGC era inoltre operata a Terra. In caso di guasto al sistema del Modulo di Comando, la guida poteva essere proseguita in manuale sulla base dei dati trasmessi da Terra.

LA CARRIERA DI AGC

Due versioni sono state realizzate, Block I e Block II. Block I era sostanzialmente analogo al sistema di bordo dei missili Polaris. Block II era più sofisticato ed affidabile, sviluppato per ovviare alcuni inconvenienti emersi nella carriera dei Polaris. AGC Block I è stato usato nelle prime missioni Apollo, AGC Block II è stato invece usato nelle missioni successive, da Apollo 7 ad Apollo 17 e nei relativi Moduli Lunari (LEM). Il suo uso non è però terminato con le missioni Apollo, è stato usato in tre missioni Skylab e nella missione congiunta Apollo-Soyuz. L'ultimo impiego si è avuto nel 1972 su un aeroplano Vought F-8C Crusa-

der, dove ha costituito il sistema di controllo primario del primo sistema fly-by-wires sperimentato in volo, vedi fly-by-wire research project using F-8 aircraft.

PORTIAMOCENE UNO A CASA...

Potrebbe sembrare uno scherzo ma la documentazione sostanzialmente completa oggi disponibile online, ha invogliato alcuni volenterosi a ricreare varie versioni hardware e software di AGC, vedi http://www.ibiblio.org/apollo/Pultorak.html#-Mike_Stewarts_Block_II_AGC_Project

Forse la più nota è la ricostruzione funzionante di Block I fatta da John Pultorak tra il 2000 ed il 2004, lavorando una media di 10 ore a settimana ed usando circa 2 km di filo per wire-wrap con 15000 connessioni. Vista l'irreperibilità di logiche RTL, John ha usato circuiti integrati della famiglia 74LS, arrivando ad usarne 500. Gli schemi completi di Block I non erano disponibili all'epoca, quindi la realizzazione è una sua interpretazione quanto più vicina possibile. Altre versioni conformi invece agli schemi originali sono descritte nel sito all'indirizzo fornito. Navigando in Internet, sono anche reperibili implementazioni meno fedeli ma meno monumentali, quale <https://blog.arduino.cc/2018/01/11/recreating-the-apollo-guidance-computer-display-and-keyboard-with-arduino/> basata su Arduino. Il kit completo offerto non è certo economico per togliersi uno sfizio ma la quantità di lavoro che ha richiesto è di tutto rispetto.

Un'alternativa decisamente economica, consiste nel rinunciare ad ogni hardware e giocare con un emulatore software che consente di riprodurre tutte le fasi di molte missioni Apollo. Troviamo Virtual AGC qui <http://www.ibiblio.org/apollo/index.html>

CONCLUSIONI

Abbiamo appena scalfito un argomento affascinante ed immenso, parlando tra l'altro soprattutto di hardware, con minimi cenni al software che pure ha avuto un ruolo altrettanto rilevante (e la sua parte di polemiche). Abbiamo anche rapidamente seguito le tracce lasciate da AGC fino ai giorni nostri, riconoscimento di un progetto ricco di soluzioni ancora attuali. In conclusione, si riporta il commento di John Pultorak, tradotto integralmente dal suo documento "Block I Apollo Guidance Computer (AGC) How to build one in your basement - Part 1: Overview", commento impregnato di autentico "spirito Maker".

"Perché costruire un AGC?

I primi computer sono interessanti. Poiché sono semplici, potete (se lo desiderate) comprendere l'intero computer dall'hardware al software. AGC è il più interessante perché:

- a) Ha portato il primo uomo sulla Luna.
- b) E' il primo computer al mondo realizzato con circuiti integrati.

La sua architettura presenta anche interessanti soluzioni.

(John Pultorak, Dicembre 2004)"



AD8232+Sensore IR per ECG con Arduino

di Roberto Vallini

r.vallini@elettronicaemaker.it

In questo articolo vedremo come visualizzare sul computer la nostra attività cardiaca utilizzando Arduino. Il tutto sarà attuato con due sistemi: il primo attraverso uno spettro luminoso che attraversa alcune parti del nostro corpo, il secondo monitorando gli impulsi elettrici con la shield Sparkfun AD8232, interfacciata tramite una porta di comunicazione I²C.

DOVEROSA CONSIDERAZIONE SULL' ECG

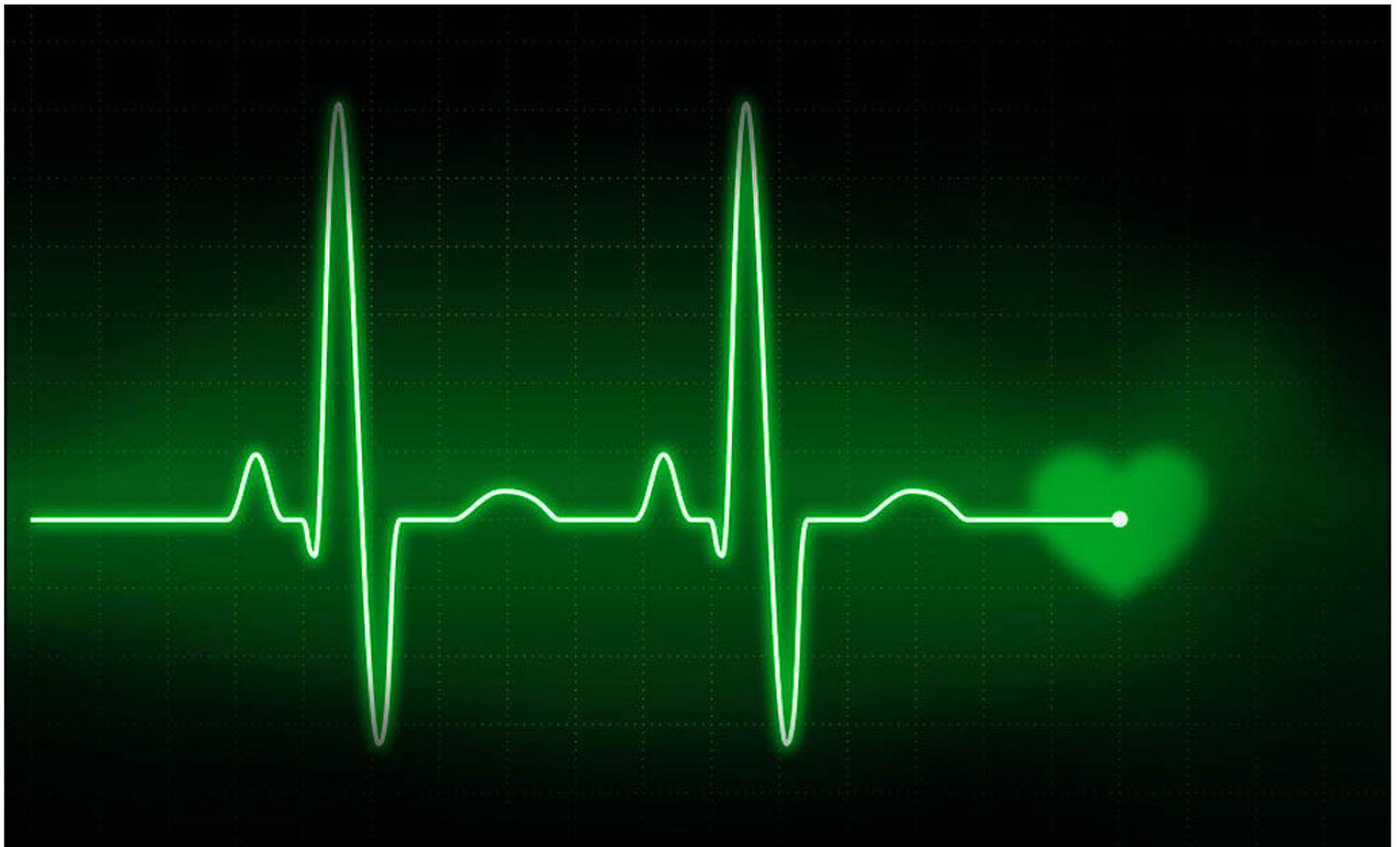
L'elettrocardiogramma, conosciuto con la sigla ECG è la riproduzione grafica dell'attività elettrica del cuore durante il suo funzionamento, rilevata su parti della superficie del corpo. Sulla superficie del corpo umano e di quello animale sono presenti e registrabili campi elettrici di bassa intensità, che nell'individuo a riposo sono principalmente dovuti alle periodiche depolarizzazioni e re-polarizzazioni del cuore. Con il termine polarizzazione ci si riferisce, praticamente, a tutta l'attività elettrica che si registra a livello del tronco. In effetti, i potenziali elettrici prodotti dal muscolo cardiaco rappresentano la sommatoria di minime quantità di elettricità, generate dalle singole cellule muscolari cardiache. Queste piccole correnti vengono registrate attraverso un apparecchio denominato elettro-

cardiografo. L'energia elettrica viene adeguatamente amplificata, in modo da poter trascrivere le escursioni in maniera chiara, consentendo la registrazione di un segnale leggibile. In questo progetto, andremo a rilevare tali mutazioni sullo schermo del nostro PC, attraverso il monitor seriale di Arduino Uno/ Nano. I sistemi utilizzati saranno quindi due. Il primo attraverso le mutazioni dello spettro luminoso, mentre nel secondo sfruttando i dipoli elettrici, rilevati con degli elettrodi opportunamente posizionati.

TEORIA DEL DIPOLO CALDO

Il concetto del triangolo equilatero di Einthoven (1912), o del dipolo caldo (vedi Figura 1), sviluppato da precedenti ricerche, è

importante per le rilevazioni del battito cardiaco nell'ambito elettrico. Esso considera tutto il cuore come un dipolo equivalente uniforme. Pensano il dipolo caldo equivalente come una fonte di correnti elettriche che si alzano e si abbassano, seguendo il flusso cardiaco, nel conduttore di volume sanguigno corrispondente. La distanza di un punto dagli altri due, considerati insieme, è la misura della corrente di quel punto. L'utilità di tale triangolo consiste nel fatto che una forza elettrica, originata a livello cardiaco, può essere rappresentata e risolta in una maglia elettrica. Da una rete lineare contenente n generatori indipendenti di tensione o di corrente, la tensione relativa ad una coppia di nodi o la corrente in un ramo può essere determinata come





somma algebrica degli n valori che quella tensione o quella corrente assume quando gli n generatori indipendenti si suppongono agenti separatamente, uno per volta. Questo piccolo approfondimento è utile, in quanto ci dà un'idea del perché gli elettrodi di rilevazione delle apparecchiature professionali vengano posizionati in precise zone del corpo. Per approfondire [1]. Lo stimolo che genera le contrazioni è di natura elettrica si origina involontariamente dai centri di controllo posti nel sistema nervoso centrale nell'encefalo e nel midollo spinale. Esso viene trasportato dal sistema nervoso centrale al cuore attraverso le vie efferenti parasimpatiche e simpatiche. Questo stimolatore denominato nodo senoatriale produce l'impulso nervoso simile ad una scarica elettrica (vedi Figura 2), che attiva la contrazione cardiaca. Questa particola-

re proprietà di produrre spontaneamente lo stimolo che regola la frequenza cardiaca, chiamato ritmo sinusale, è un vero e proprio "pacemaker" del cuore, che in Inglese indica qualcosa che "stabilisce il ritmo". È situato nell'atrio destro, vicino allo sbocco della vena cava superiore.

LA RADIAZIONE INFRAROSSA

È una radiazione elettromagnetica con lunghezza d'onda superiore a quella della luce visibile. Tutti i nostri tessuti, nell'insieme dei processi metabolici, producono calore. Di conseguenza, lo emanano in forma di radiazione infrarossa. La radiazione infrarossa penetra nel corpo e il calore si distribuisce naturalmente laddove è richiesto. Un fenomeno interessante è legato alla radiazione infrarossa che siamo in grado di percepire dal Sole anche in una fredda giornata invernale, se soleggiata.

Ogni anno noi tutti viviamo un paradosso interessante: quando d'inverno ci troviamo in una stanza con temperatura ambientale di 18°C , dobbiamo coprirci con indumenti pesanti, per non sentire freddo. Quando arriva l'estate invece possiamo stare nella stessa stanza di prima, alla stessa temperatura, ma in maglietta e non sentiamo freddo. Questo fenomeno è

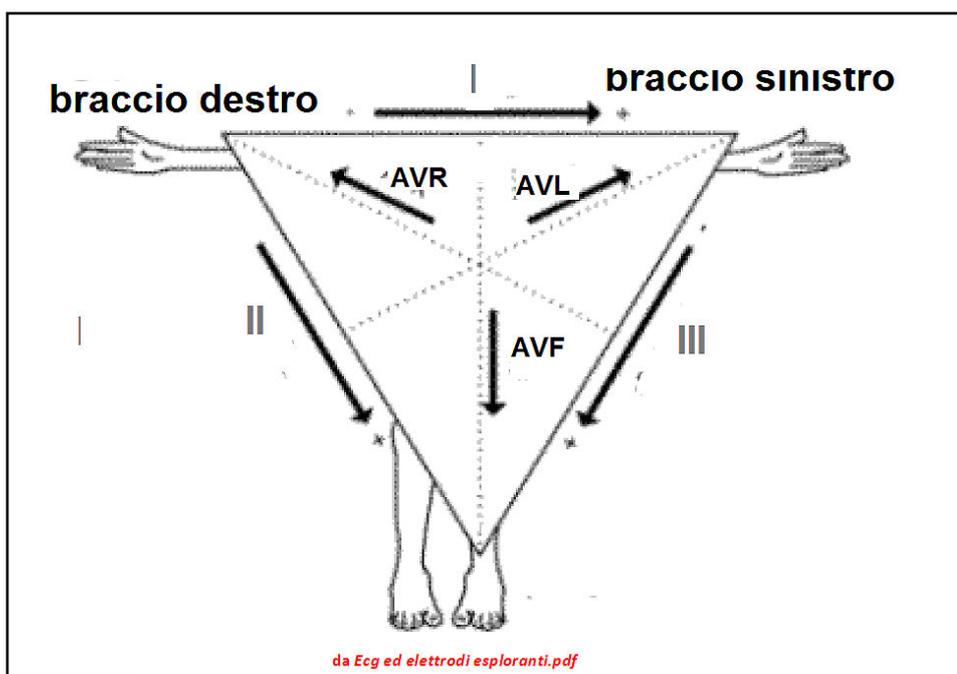


Figura 1: Triangolo equilatero di Einthoven

possibile grazie all'energia infrarossa. Essa riscalda gli oggetti circostanti, i quali a loro volta distribuiscono calore, anche al nostro corpo. Questo fenomeno viene definito “mutuo scambio del parenchima”. Negli organismi animali esso rappresenta un tessuto che compone la massa principale, attiva e funzionale di un agglomerato cellulare, parte di organo o organo in toto. È spesso strutturalmente associato ad uno o più tessuti con funzione nutritiva e di sostegno. Le onde infrarosse, onde di luce invisibile, vengono assorbite anche dalla pelle, e

penetrano nel corpo umano. Il nostro corpo, che con il proprio sistema metabolico produce calore, poi si raffredda emettendo radiazione infrarossa. I vasi sanguigni vicini alla superficie del derma, che sono soggetti a variazioni dimensionali in funzione del battito cardiaco, sono un importante fonte di informazioni, proprio grazie all'emissione Infrared Ray (I.R.) la quale viene sfruttata, in qualsiasi pronto soccorso ospedaliero, con l'applicazione di rilevatore a clip, chiamato saturimetro/pulsiossimetro. Quest'ultimo, durante lo screening di accettazione, viene

attaccato a un dito del paziente e fornisce il livello di saturazione di ossigeno del sangue, rilevando sia la temperatura che la frequenza cardiaca, sfruttando la diretta relazione tra assorbimento nelle lunghezze d'onda rosse e infrarosse. Quindi siamo nell'ambito della luce, sia visibile che invisibile. Questa precisazione è importante, dato che ho visionato molti progetti destinati a rilevare il battito cardiaco, dei quali parlerò brevemente. Il primo è stato fatto con un sensore piezoelettrico,

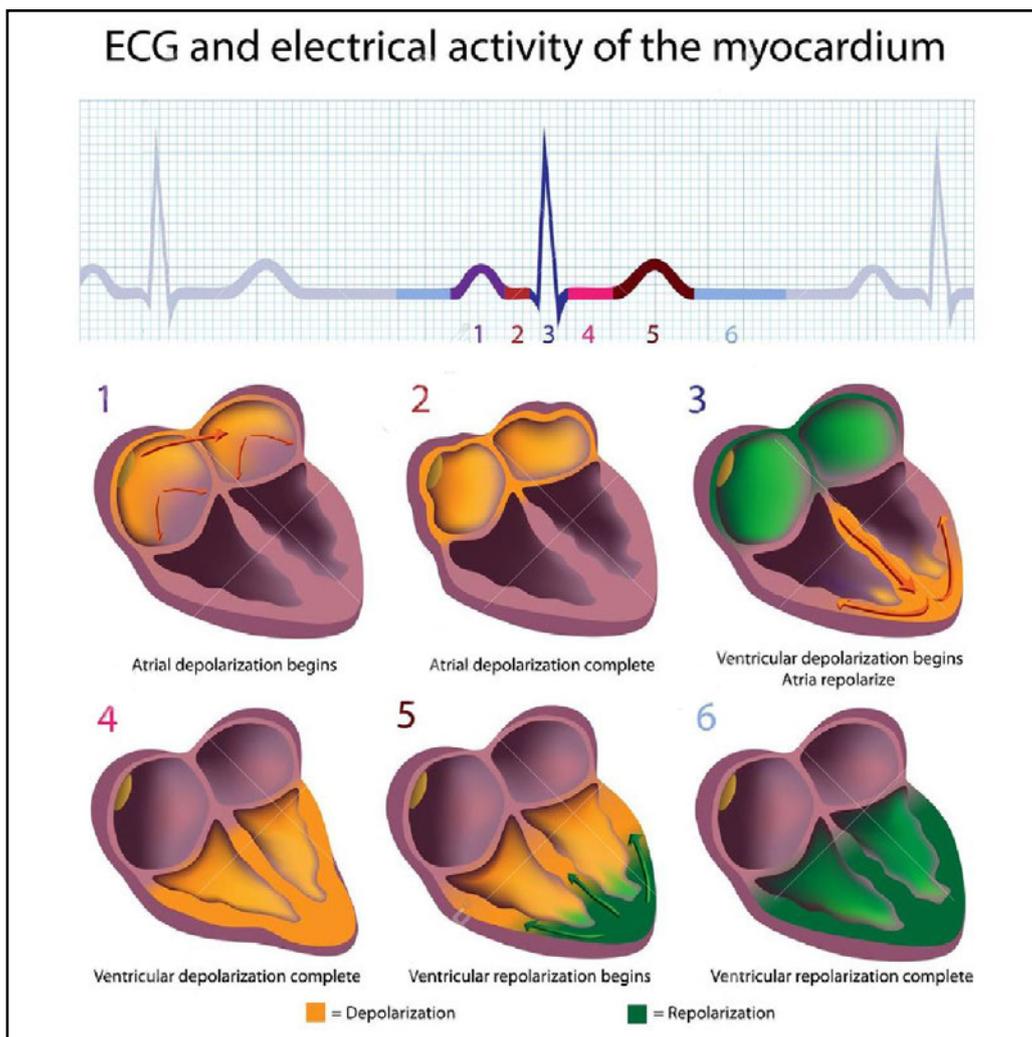


Figura 2: Relazione tra il tracciato dell'ECG e l'attività del miocardio



nastrato attorno ad un dito e poi collegato direttamente ad Arduino, altri con un semplice led ricevitore di infrarossi del genere TSOP4838. In linea generale, in tutti i casi il battito cardiaco viene rilevato in maniera ottica, per poi essere restituito attraverso vari tipi di visualizzazione. Tra le varie versioni ciò che fa la differenza è la precisione, ma soprattutto il discernimento tra tutti gli spettri luminosi emessi e rilevati. Comunque, il percorso di principio è il medesimo. A): rilevazione del battito tramite un sensore, B): trasformazione in una grandezza elettrica, C): amplificazione del segnale, D): elaborazione del segnale per renderlo riconoscibile in qualsivoglia maniera, sia essa sonora, che ottica, oppure grafica. L'importanza dell'intero percorso consiste soprattutto nel garantire e mantenere l'affidabilità di quanto rilevato, cercando di abbassare il più possibile la soglia di rumore, insita, inevitabilmente, in tutti gli ambienti sperimentali.

LED VISIBILI E LED INFRAROSSI

E' un led di materiale diverso dal silicio (ma sempre semiconduttore) formato da gallio, alluminio, fosforo, arsenico, indio in varie percentuali, in cui il passaggio di corrente in polarizzazione diretta produce principalmente emissione di luce (ogni elettrone perde una quantità ben precisa di energia, corrispondente a un fotone), attraverso la plastica più o meno trasparente del contenitore si può intravedere la diversa forma dei due elettrodi: il più grande è il catodo. Il corrispondente ricevitore infrarosso, in cui può entrare la luce (visibile o infrarossa), libera ulteriori portatori di carica, per cui equivale alla presenza di una corrente di polarizzazione diretta senza la presenza di un generatore di tensione. Cercherò di schematizzare il principio di rilevazione delle pulsazioni cardiache attraverso quello che viene definito tessuto parenchimatrico. Non essendo un cardiologo, mi affiderò a tutto quanto imparato finora, basando-

mi sulle numerosi tesi lette, anche più volte, presenti sul web. Lo farò con uno schema a blocchi, cercando di renderlo il più semplice possibile, (Figura 4). Dopo questa premessa, entriamo nell'applicazione elettronica vera e propria la quale, nello specifico, si può considerare dedicata alla conoscenza di tutte le manifestazioni che, una

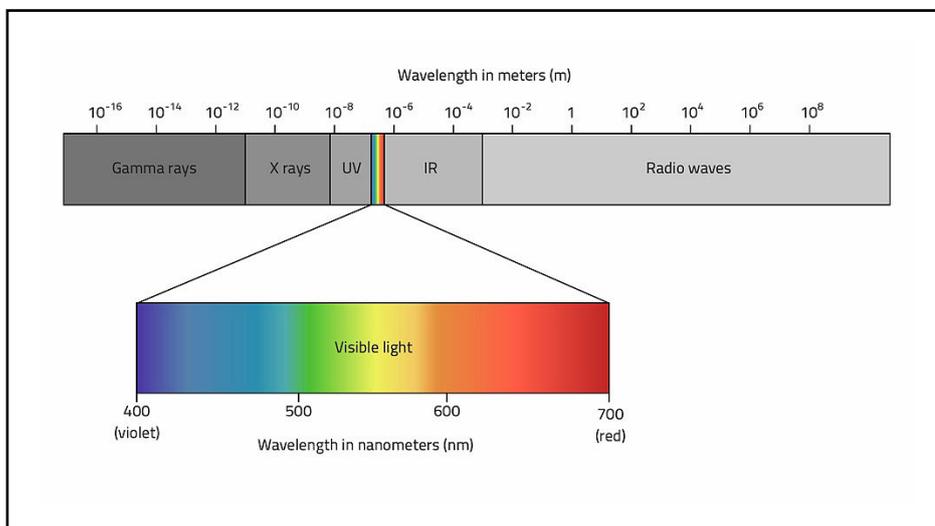


Figura 3: Spettro del visibile, seguito (a Dx) dalla banda IR

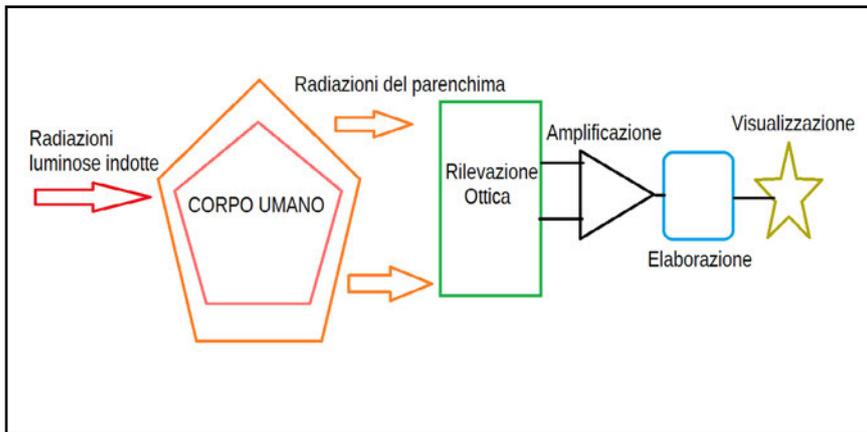


Figura 4: Principio di rilevazione delle pulsazioni.

volta rilevate in forma puramente conoscitiva e senza pretese mediche, ci potranno offrire un piccolo specchio dell'universo dei segnali di funzionamento che il nostro muscolo cardiaco invia verso l'esterno. Quello che ho sperimentato è un pulsiossimetro da dito, o da lobo auricolare. Misurare la saturazione è utile solo in caso di patologie gravi. Normalmente il livello di saturazione

è sempre corretto e, nel caso non lo fosse, si avrebbero dei sintomi seri, tali da richiedere le cure di un medico. Per realizzare un pulsiossimetro professionale ci vorrebbero componenti particolari e tarature che renderebbero l'esito incerto. Perciò ci affidiamo a una shield della Adafruit la quale garantisce comunque, se il circuito è opportunamente tarato, livelli di visualizzazione più che accettabili. Comunque, lo rammento ancora una volta, questi progetti non sono finalizzati a rilevare condizioni patologiche, ma solo a scopo puramente sperimentale.

IL PULSE SENSOR AMPED

Sulla parte anteriore del sensore, vi è il lato con il logo del produttore. Questo è il lato

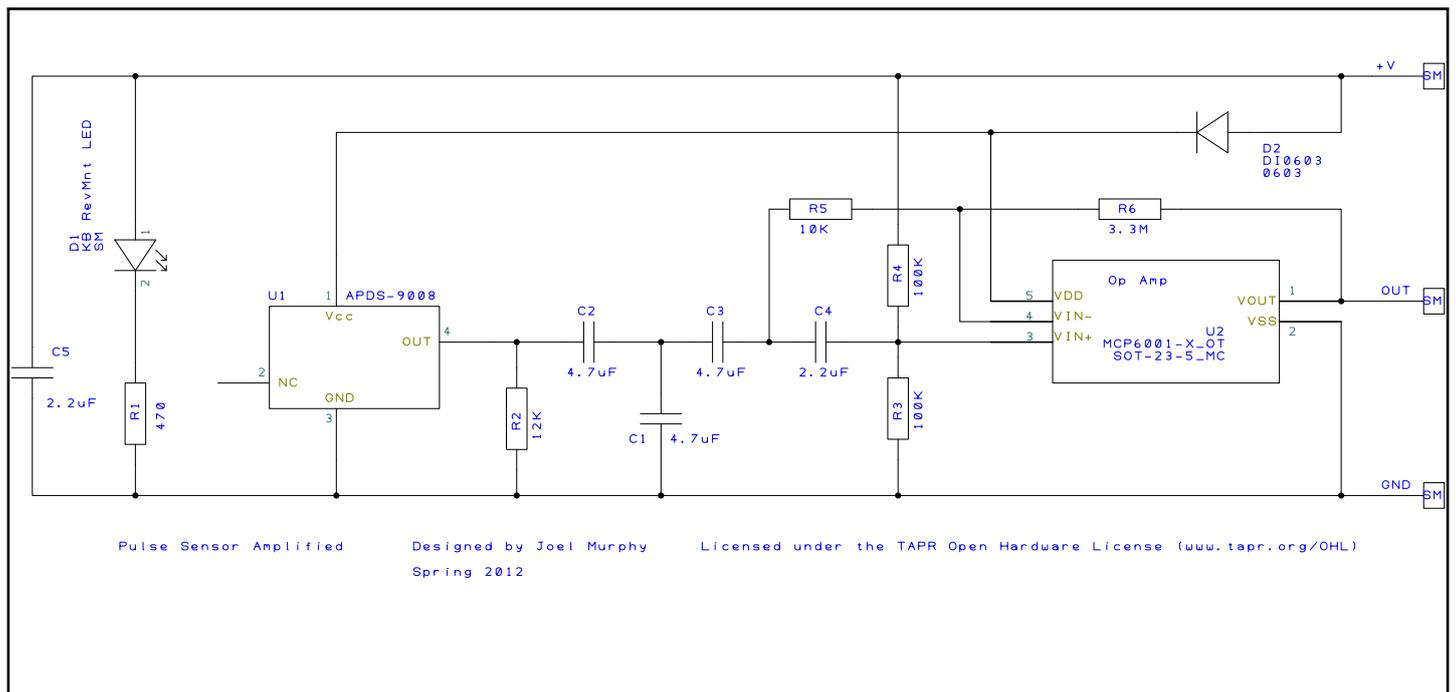


Figura 5: Schema elettrico del PulseSensor Amped

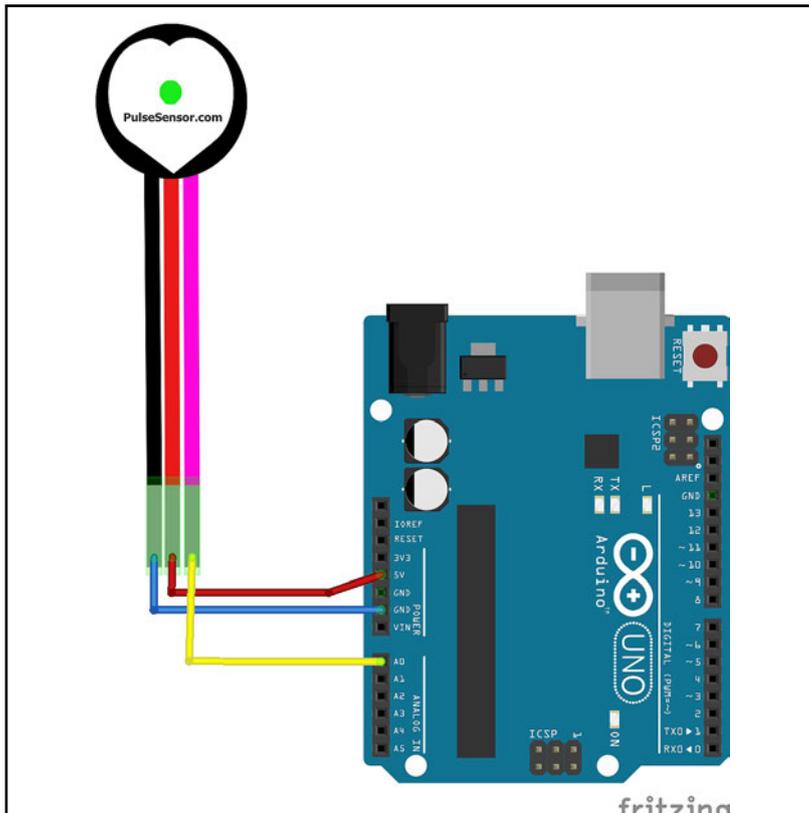


Figura 6: Collegamento del PulseSensor Amped

che entra in contatto con la pelle. Sempre nella parte anteriore, si nota il LED emettitore Kingbright AM2520ZGC09 [6] dotato di forte luminescenza verde, a campo ristretto, che ha lo scopo di permeare il tessuto umano e trasmettere la luce al ricevitore contrapposto, l'APDS-9008 [5], un chip della Avago Technologies con uscita analogica. Quest'ultimo dispositivo viene anche usato come sensore di luce per apparecchi fotografici. Riesce a rilevare uno spettro cromatico molto ampio ed è un sensore altamente reattivo per obiettivi molto vicini. Il LED illumina la punta delle dita o il lobo dell'orecchio o altri tessuti capillari e il sensore legge la quantità di luce che li attraversa. Sulla parte posteriore è

alloggiato il resto della circuiteria, la quale si dovrà interfacciare con il microcontrollore. Al momento del battito, la presenza di sangue nelle vene aumenta e ciò modifica la quantità di luce riflessa. Questa variazione, catturata dal sensore e inviata ad Arduino sotto forma di segnale analogico, consente di contare i battiti e quindi calcolarne il numero al secondo. Questo avviene attraverso i cavi AWG 24, con uscita maschio, lunghi circa 200 mm., rispettivamente VCC, GND, e Signal. Il Pulse Sensor Amped è una versione notevolmente migliorata del Pulse Sensor originale, un sensore di frequenza cardiaca plug-and-play per Arduino

e Arduino compatibili. Può essere utilizzato da studenti, artisti, atleti che desiderano incorporare facilmente i dati di frequenza cardiaca in tempo reale nei loro progetti. Il Pulse Sensor Amped aggiunge all'hardware circuiti di amplificazione e cancellazione del rumore. È notevolmente più veloce e risulta più facile ottenere letture di impulsi affidabili. Il sensore di impulsi amplificato funziona con un Arduino in versioni da 3V o 5V. Il kit comprende una clip auricolare, perfettamente dimensionata per il sensore. Volendo, può essere incollata con colla a caldo sul retro del sensore per ottenere la lettura da un lobo dell'orecchio. Vi sono alcune parti per realizzare un pratico cinturino in velcro. Questo è un altro metodo

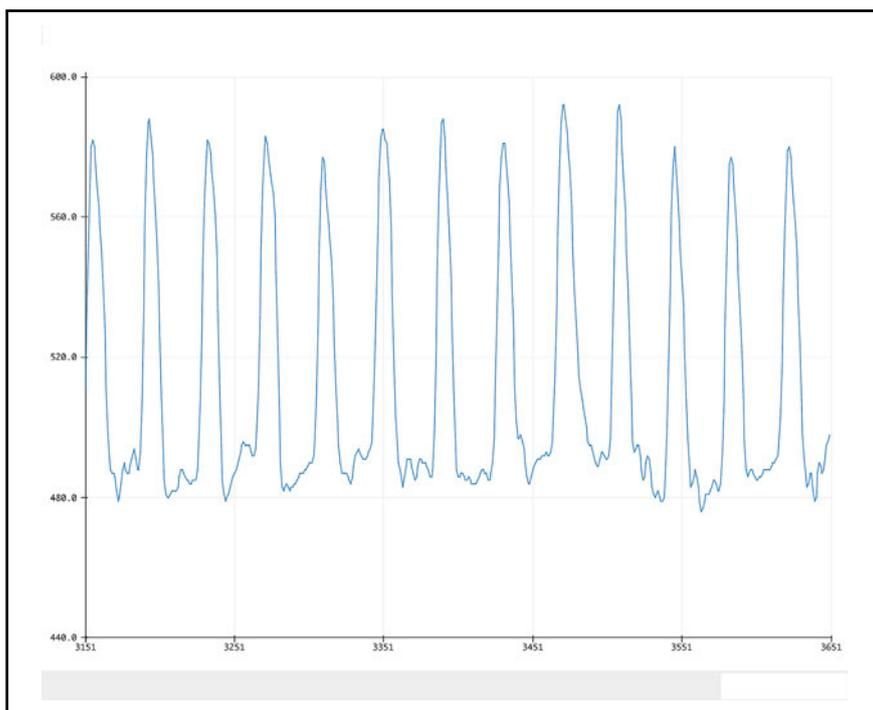


Figura 7: Tracciato del PulseSensor restituito da Arduino

per ottenere dati sulla frequenza cardiaca. In dotazione vi sono anche 4 adesivi trasparenti, per isolare la parte anteriore del sensore di impulsi dalle secrezioni dell'epidermide dei polpastrelli. Il Pulse Sensor non si limita a contare i battiti cardiaci, ma è dotato anche di una routine di “taratura” che analizza le pulsazioni per una decina di secondi, individua i valori massimi e minimi trasmessi dal sensore e definisce la soglia oltre la quale il valore trasmesso dal sensore può essere considerato un battito. Il sensore di impulsi ha 3 fori attorno al bordo esterno che consentono di cucirlo ad un supporto tessile. Il costo dell'originale Adafruit è di 25 €, ma personalmente, mi sono rivolto a Ebay che, a un prezzo che non supera i 5 \$, propone articoli equivalenti. Certo, con quest'ultimi occorre poi

cercare e, se necessario, modificare il software di utilizzo, ma questo ritengo possa rientrare nelle competenze del progettista. Lo schema di collegamento (Figura 6) è il seguente: il sensore viene alimentato dai 5 volt forniti da Arduino, poi connesso a GND e la parte di segnale ad un pin analogico, nel nostro caso, A0. Dal link [2] è possibile scaricare l'algoritmo di visualizzazione sul monitor del PC (Figura 7). Una volta lanciato lo sketch, attivate il monitor seriale, vedrete quindi il vostro battito cardiaco, rappre-

sentato in forma di tracciato. Finora abbiamo esaminato il funzionamento del Pulse Sensor Amped che è paragonabile, in linea di principio, al pulsiossimetro. Come scritto in precedenza, nella sua funzione base è paragonabile a quello della clip che ci viene innestata sul dito ogni qual volta ci rechiamo in una struttura di pronto soccorso sanitario, nei casi di necessità. Se in queste occasioni il primo, sommario controllo con il pulsiossimetro rivelasse qualche anomalia, gli assistenti sanitari passerebbero ad un controllo più approfondito, con un dispositivo chiamato elettrocardiografo. Le premesse fatte in apertura, ci hanno reso un'idea generale del suo principio di funzionamento. La shield che andremo a vedere funziona con lo stesso principio dell'ECG. Il modulo (Figura 7) può essere utilizzato



come cardiofrequenzimetro, per misurare l'attività elettrica del cuore. Questa attività elettrica può essere tracciata come un ECG o elettrocardiogramma (l'uscita è una lettura analogica). Il chip AD8232, Heart Rate Monitor comprende un amplificatore operazionale, per aiutare ad ottenere un segnale chiaro e leggibile. È stato progettato per estrarre, amplificare e filtrare piccoli segnali biopotenziali.

I SEGNALI BIOPOTENZIALI

Prima di approfondire il funzionamento del circuito, esaminiamo il modo in cui misuriamo il battito cardiaco. Come anticipato nella premessa, l'attività elettrica del cuore, produce un campo magnetico, il quale si risolve in segnali elettroacustici. Questi segnali cruciali provengono da una massa di cellule nel nostro cuore chiamata nodo

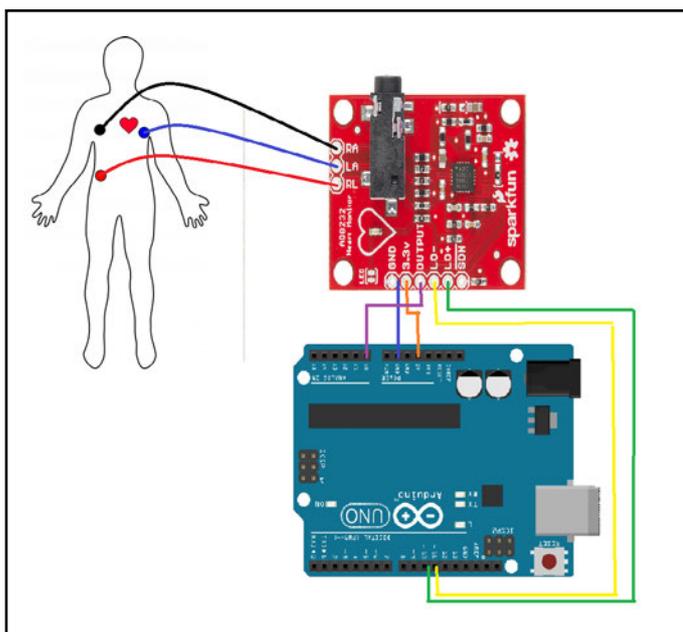


Figura 6: Schema di collegamento tra il modulo Sparkfun, Arduino e elettrodi sensori

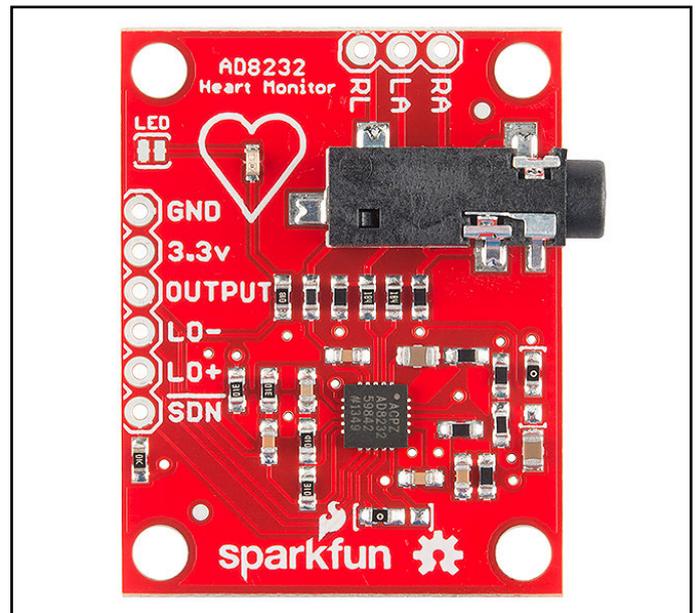


Figura 7: Il modulo Sparkfun AD8232

senoatriale o NSA. Esso si trova nell'atrio destro e controlla il ritmo del nostro battito cardiaco e il flusso di sangue che va dal cuore alle altre parti del corpo. Per formare un battito cardiaco, la regione NSA del cuore inizia ad eccitarsi, mentre gli atri, (le due camere superiori del cuore) si contraggono. Il nodo NSA ha quindi generato una piccola scarica elettrica, la quale stimola la contrazione sincronizzata. Per misurare l'ECG, posizioneremmo gli elettrodi sulla superficie del cuore in modo ottimale. Ricordandoci che non abbiamo competenze mediche, ma piuttosto quelle tipiche degli sperimentatori, per cui cercheremo di misurare, al meglio delle nostre possibilità, le variazioni di tensione sulla superficie dell'epidermide, che di solito non superano qualche millivolt. Occorrerebbe una lunga spiegazione tecnica, a livello medico, per espletare l'intero argomento, cosa che non

ci compete. Ciò che conta per noi sono i punti dove la shield AD8232 va collegata al nostro corpo, attraverso gli elettrodi in dotazione nel kit. Quest'ultimi sono, ne più ne meno, gli stessi che ci vengono applicati "in situ" durante un normale elettrocardiogramma. Rispettivamente, elettrodo RA, a destra del torace, elettrodo LA a sinistra del torace, leggermente in basso, mentre l'elettrodo RL più vicino possibile al basso-ventre dal lato destro, verso l'incrocio con la gamba. La shield è dotata di due sistemi di collegamento agli elettrodi; uno è a saldare, mentre l'altro dispone di un plug su quale sono già inserite le clip per i sensori. Giunti a questo punto, se vogliamo visualizzare i nostri battiti è necessario interfacciarsi con un microcontrollore, nel nostro caso quello di Arduino Uno/Nano.

Le connessioni con la scheda sono : SDN, LO +, LO-, USCITA, 3,3V, GND.

Le caratteristiche dichiarate dal costruttore sono:

- Tensione alimentazione : 3,3V
- Uscita : analogica
- Rilevamento sensori staccati.
- Pin di shutdown
- Indicatore LED battito
- Interfaccia sensori (RA, LA, RL) su 3 Pin connettore 2,54 o su Jack da 3,5mm
- Misure: 36mm * 31mm * 18mm
- Temp. lavoro : -40 °C— +85 °C
- Peso confezione: 45 gr

Nel caso in cui usiate gli elettrodi, i quali si connettono con un plug tramite il jack della scheda, il posizionamento è quello della Figura 8, dove ho utilizzato una maglietta per schematizzare i punti elettivi sul torace. Quello che leggete nella tabella di pagina seguente è il listato da caricare. Ne esistono più versioni, ma qui potete iniziare a testare il vostro Heart Rate Monitor.

Di conseguenza, per prima cosa dovrete installare la libreria I²C [3], necessaria per la comunicazione seriale tra la shield AD8232 con Arduino ed il vostro PC. Come nel precedente progetto, anche in questo caso, una volta caricato lo sketch, si dovrà aprire il monitor seriale per la visualizzazione del battito cardiaco.

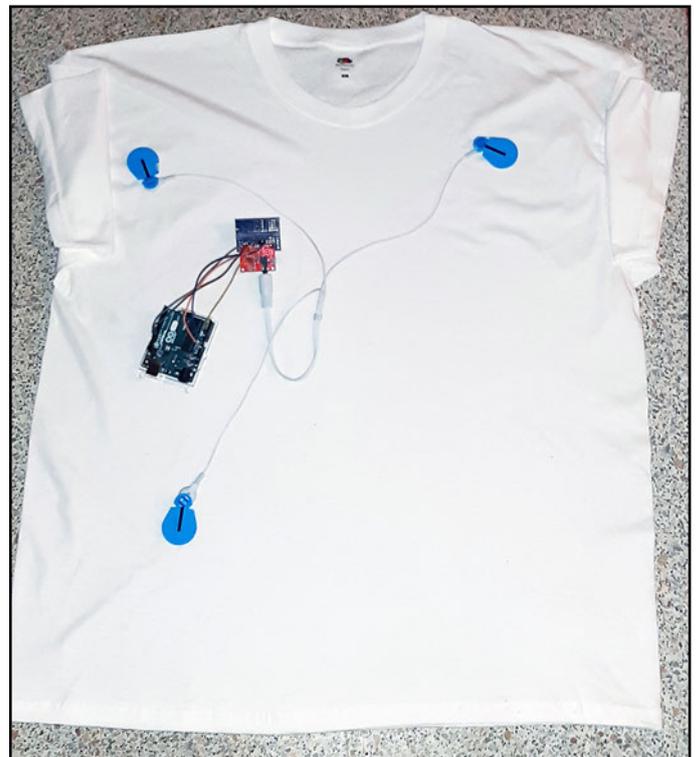


Figura 8: Posizionamento degli elettrodi in uscita dal modulo AD8232

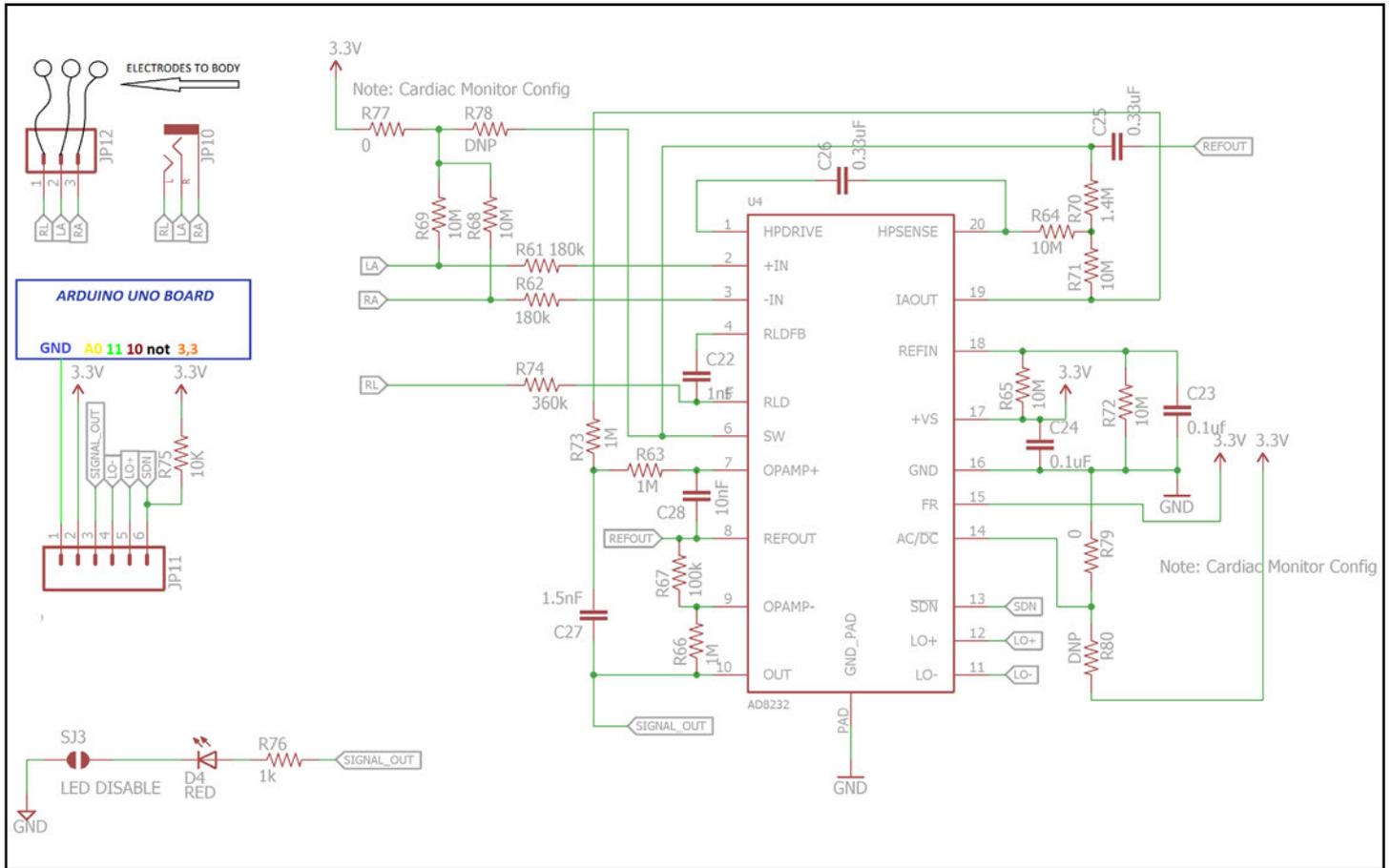


Figura 9: Schema elettrico del modulo Sparkfun

Demo Program for AD8232 Heart Rate sensor.

```
void setup() {
  // inizializzazione comunicazione seriale:
  Serial.begin(9600);
  pinMode(10, INPUT); // Impostazione per il livellamento di lead off LO +
  pinMode(11, INPUT); // Impostazione per il livellamento di lead off LO -
}

void loop() {
  if((digitalRead(10) == 1) || (digitalRead(11) == 1)){
    Serial.println('!');
  }
  else{
    // manda il segnale analogico a input 0:
    Serial.println(analogRead(A0));
  }
  //Attesa minima per impedire la saturazione dei dati seriali
  delay(1);
}
```

CONCLUSIONI

Lo scopo di questi due progetti deriva da una necessità: quella di colmare le notevoli lacune quasi sistematicamente presenti nelle varie versioni reperibili sul Web; carenze constatate e sofferte anche attraverso la lettura di innumerevoli blog redatti da maker di tutto il mondo. Ho notato, come sempre, una certa immediatezza nell'illustrare i collegamenti e le possibili varianti circuitali ma, di contro, nessun tipo di attenzione ai motivi che hanno stimolato la realizzazione dei progetti stessi. Lo scopo principale di un esercizio di questo genere non è che tutto funzioni (ammesso che riesca a farlo), ma piuttosto, come in questo caso, quello di comprendere "il perché delle cose", andando quindi a cercare la relazione stretta tra questi tracciati e il nostro corpo. Curiosità e approfondimento: queste sono le forze propulsive dei maker. Da parte

mia, ho letto molte tesi sul muscolo cardiaco e mi ritengo soddisfatto per aver avuto l'opportunità di scoprire tante nozioni che non conoscevo. Giusto per citarne una come esempio, tra tutti: ho scoperto come è distribuita l'elettricità nel nostro corpo e come essa possa essere tradotta in concetti di fisica elettronica. Concludo nella speranza che, così com'è stato per me, quanto ho scritto in questo articolo possa rappresentare per voi una finestra che si dischiude su un mondo di sperimentazione, affascinante e ancora tutto da esplorare.

Nota: Entrambi i progetti proposti in questo articolo hanno un puro scopo dimostrativo. Non sono quindi da considerarsi dispositivi medicali né devono essere destinati alla diagnosi o alla cura di qualsiasi tipo di condizioni patologiche, per le quali è sempre indispensabile rivolgersi a un medico.

RIFERIMENTI E LINK UTILI

[1] <https://gioaboffa.wordpress.com/tag/triangolo-di-einthoven/>

[2] https://github.com/WorldFamousElectronics/PulseSensor_Amped_Arduino

[3] <https://playground.arduino.cc/Main/SoftwareI2CLibrary/>

[4] https://cdn.shopify.com/s/files/1/0100/6632/files/Pulse_Sensor_Data_Sheet.pdf?14358792549038671331

[5] <https://www.broadcom.com/docs/AV02-1169EN>

[6] <http://www.kingbrightusa.com/images/catalog/SPEC/am2520zgc09.pdf>

[7] <https://www.ebay.it/itm/Modulo-con-sensore-battito-cuore-e-frequenza-cardiaca-per-Raspberry-Arduino-/262770966836>



TensorFlow

per Raspberry Pi

di Stefano Lovati
s.lovati@elettronicaemaker.it

TensorFlow, una delle piattaforme maggiormente utilizzate per lo sviluppo di applicazioni nel campo dell'intelligenza artificiale (e in particolare nel campo del deep learning), può essere installata in pochi semplici passi anche su un comune Raspberry Pi. Scopriamo come farlo in questo articolo, nel corso del quale testeremo anche un modello di classificatore per il riconoscimento delle immagini

Il progetto TensorFlow, nato nel 2015 all'interno dei laboratori di ricerca Google Brain, è una libreria software open source per il calcolo numerico basata sull'utilizzo di grafi dataflow. TensorFlow si focalizza dunque sui concetti di grafo e di flusso dei dati che stimolano il processing concentrato nei nodi della rete. Un esempio di grafo TensorFlow è rappresentato in Figura 1: i nodi rappresentano le operazioni matematiche, mentre gli archi rappresentano i flussi dati che scorrono tra i nodi. I dati scambiati tra i nodi, in particolare, vengono chiamati tensori, una forma particolare di array (vettore) multidimensionale.

Questa particolare architettura risulta molto flessibile, consentendo l'allocazione e distribuzione dei nodi computazionali su più unità di elaborazione (CPU single o multi-core e GPU) e su varie tipologie di piattaforma hardware (computer desktop,

server, dispositivo mobile oppure embedded) senza alcuna necessità di dover riscrivere il codice. A causa dell'utilizzo esteso di operazioni di calcolo vettoriale e matriciale, TensorFlow fornisce il migliore livello di prestazioni quando viene eseguito su una GPU. Google ha persino realizzato un proprio hardware specializzato per l'elaborazione eseguita sul cloud, noto come Tensor Processing Unit (TPU), in grado di garantire il migliore livello prestazionale tra tutti i differenti tipi di unità di elaborazione. TensorFlow è stato inizialmente sviluppato internamente a Go-

ogle per applicazioni di machine learning e deep learning, ma può essere utilizzato in un'ampia gamma di domini. Dal punto di vista della programmazione, le API ufficiali sono disponibili in ambiente Python e C, mentre esiste una compatibilità all'indietro (non garantita) con i linguaggi C++, Java, Javascript, Go e Swift. Sin dalla sua nascita, il progetto TensorFlow si è posto l'obiettivo di diventare un framework per il machine e il deep learning accessibile a tutti. Per soddisfare questo requisito, sono state rese disponibili versioni per i più diffusi sistemi operativi: Linux, Windows, Ma-



TensorFlow

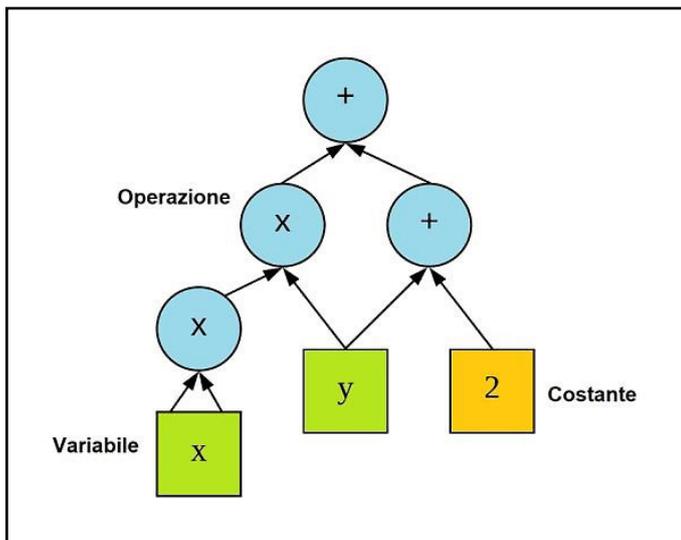


Figura 1: Un esempio di grafo TensorFlow

cOS, iOS e Android. Grazie a una stretta collaborazione tra il Google Brain Team e la Raspberry Foundation, è stato possibile aggiungere alla precedente lista anche una versione di TensorFlow (a partire dalla 1.9) appositamente “ritagliata” per il Raspberry Pi. Come vedremo tra poco, il vantaggio di questa soluzione è quello di semplificare drasticamente la procedura di installazione, resa possibile con l’utilizzo del gestore di pacchetti pip di Python e dalla disponibilità di binari precompilati. Per i maker e gli hobbisti si tratta sicuramente di un traguardo significativo: il Raspberry Pi è infatti ampiamente utilizzato non solo dagli appassionati di tecnologia e di elettronica, ma anche in ambito didattico per introdurre gli allievi al mondo della programmazione. Le frontiere del deep learning possono ora allargarsi ulteriormente, coinvolgendo anche gli appassionati che dispongono di un budget limitato.

TENSORE

In questo articolo introdurremo alcuni concetti basilari di TensorFlow, necessari per una completa comprensione di questo framework di sviluppo.

Non ci addenteremo quindi negli aspetti meramente programmatici, riservandoci di esplorare questo campo in articoli futuri, più orientati a un approccio implementativo. Il primo concetto che introduciamo è quello di tensore, a cui il framework sviluppato da Google deve il nome (il termine Tensorflow serve proprio ad indicare un flusso di tensori).

Il tensore è la struttura utilizzata per rappresentare i dati nell’ambito del deep learning ed equivale a un array multidimensionale: avremo così tensori a una dimensione (vettori), tensori a due dimensioni (matrici) e più in generale tensori di dimensione n.

Un esempio esplicativo è visibile in Figura 2, in cui sono mostrati differenti tipi di tensori a una, due e tre dimensioni.

I tensori si suddividono in tre categorie principali:

- costante: è un tensore immutabile, equivalente a un nodo privo di qualsiasi ingresso ma in grado di produrre un valore costante in uscita;
- variabile: è un tensore il cui valore può variare nel tempo, a seguito del flusso dati che attraversa il grafo. Nelle applicazioni di deep learning, le varia-

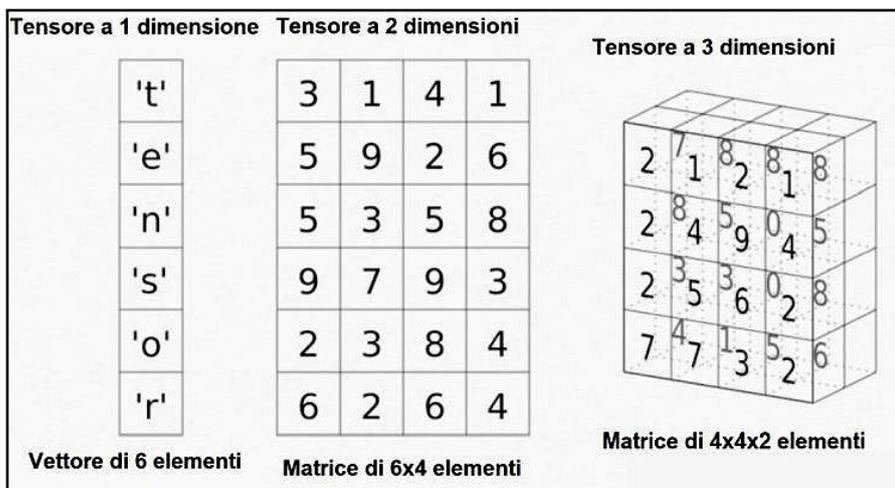


Figura 2: Esempi di tensori a 1, 2 e 3 dimensioni

bili contengono i parametri della rete che devono essere ottimizzati, cioè i pesi associati ai modelli di neurone che compongono la rete. Le variabili devono essere inizializzate prima di attivare il grafo;

- placeholder: sono un tipo particolare di tensore che memorizza i dati provenienti da sorgenti esterne. Nelle applicazioni di deep learning, i placeholder vengono normalmente utilizzati per fornire dei dati in ingresso al modello di apprendimento (learning). Un placeholder può anche essere definito come un nodo il cui valore verrà inizializzato in futuro.

Fondamentalmente, il processo di creazione di un programma Tensorflow implica i seguenti due passi:

- creazione di un grafo computazionale;
- esecuzione del grafo computazionale precedentemente creato.

GRAFO

Per grafo computazionale si intende una serie di operazioni TensorFlow associate ai vari nodi del grafo. Ciascun nodo può avere zero oppure più tensori in ingresso, ma produce sempre un tensore in uscita. A titolo di esempio, si consideri ora il grafo computazionale di Figura 3, composto da tre nodi a, b e c.

Possiamo subito osservare come i nodi a e b siano dei nodi costanti, con valori 5 e 6, rispettivamente. Il nodo c rappresenta invece l'operazione di moltiplicazione tra le costanti associate ai nodi a e b. Ne consegue che l'esecuzione del nodo c equivale alla moltiplicazione di a per b. Un grafo computazionale può essere pensato come un modo alternativo per rappresentare i calcoli matematici eseguiti in un program-

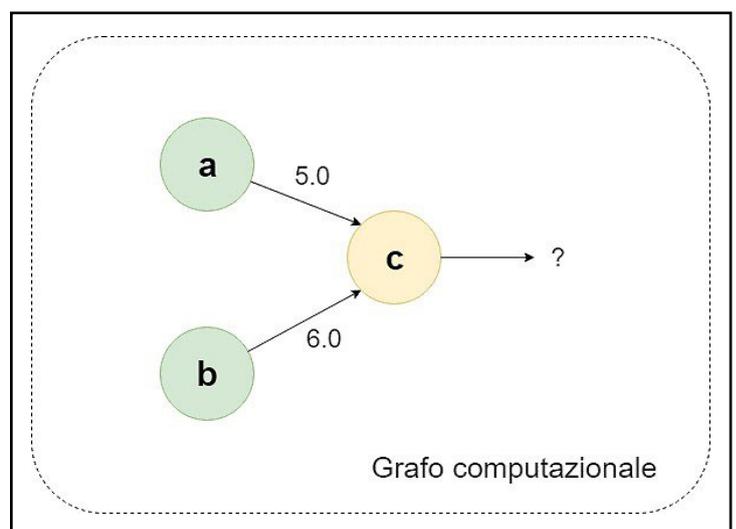


Figura 3: Un semplice grafo computazionale composto da tre soli nodi

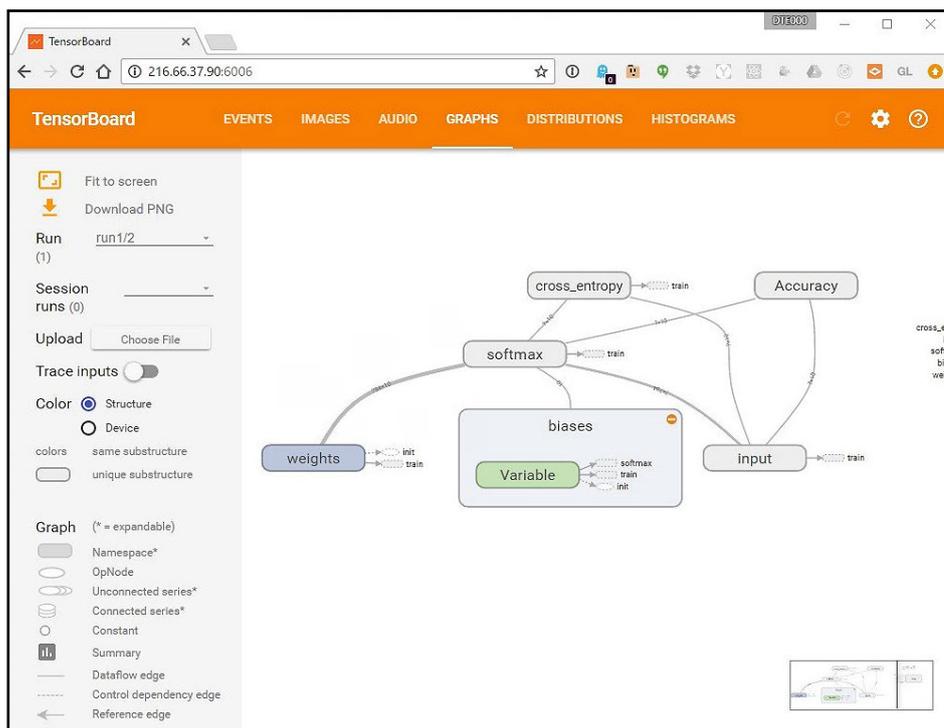


Figura 4: Screenshot dello strumento di analisi e debug TensorBoard

ma TensorFlow. Le operazioni assegnate a diversi nodi del grafo possono inoltre essere eseguite in parallelo, con enormi benefici a livello di prestazioni computazionali. Il frammento di codice Python che descrive il semplice grafo considerato nell'esempio di Figura 3 è il seguente:

```
import tensorflow as tf
```

```
# Costruzione di un grafo
```

```
a = tf.constant(5.0)
```

```
b = tf.constant(6.0)
```

```
c = a * b
```

I calcoli matematici che un'applicazione TensorFlow può comportare (come ad esempio eseguire l'addestramento di una rete neurale deep) possono essere notevolmente complessi e generare confusio-

ne. Al fine di semplificare la comprensione, il debug e l'ottimizzazione delle applicazioni TensorFlow, i progettisti del framework hanno incluso uno strumento per la visualizzazione grafica delle operazioni, chiamato TensorBoard. TensorBoard (di cui in Figura 4 possiamo osservare una screenshot) può essere utilizzato per visualizzare i grafi, per rappresentare metriche numeriche relative all'esecuzione del grafo e altro ancora.

Per parlare di esecuzione del

grafo occorre prima introdurre l'importante concetto di sessione, che ora vedremo.

SESSIONE

Per ottenere il valore in uscita dal nodo *c*, occorre eseguire il grafo computazionale all'interno di una sessione. La sessione ha infatti il compito di allocare le operazioni del grafo sui dispositivi fisici (CPU o GPU, tipicamente) e fornire dei meccanismi per poterle eseguire. Una sessione include dunque sia lo stato che il controllo di un programma TensorFlow: determina infatti l'ordine con cui le operazioni devono essere eseguite e inoltre i risultati di un'operazione verso lo stadio successivo della pipeline di operazioni. Con riferimento al grafo di Figura 3 possiamo ora esamina-

re il seguente script Python, in cui le singole fasi relative alla creazione di una sessione sono commentate direttamente nel codice:

```
# Creazione di un oggetto sessione
```

```
sess = tf.Session()
```

```
#Esecuzione del grafo all'interno della sessione e
```

```
#memorizzazione del risultato in una variabile
```

```
output_c = sess.run(c)
```

```
#Visualizza l'uscita del nodo c
```

```
print(output_c)
```

```
#La sessione viene chiusa e tutte le risorse allocate sono rilasciate
```

```
sess.close()
```



Figura 6: Programmazione della scheda SD

Eseguendo lo script si ottiene il seguente risultato: 30

INSTALLAZIONE DI TENSORFLOW

Dopo una doverosa introduzione ai principali concetti che stanno alla base di TensorFlow, è giunto il momento di procedere con l'installazione del framework sul Raspberry Pi. Per eseguire questa operazione è

sufficiente disporre di un Raspberry Pi 3 e di una scheda di memoria SD da almeno 16 Gbyte. Procediamo anzitutto con la preparazione dell'immagine di Raspbian Stretch, scaricabile dal sito ufficiale Raspberry [1] (si osservi la Figura 5).

Per la programmazione della scheda SD possiamo utilizzare il tool Etcher [2]. In Figura 6 è mostrata una fase della programmazione della memoria

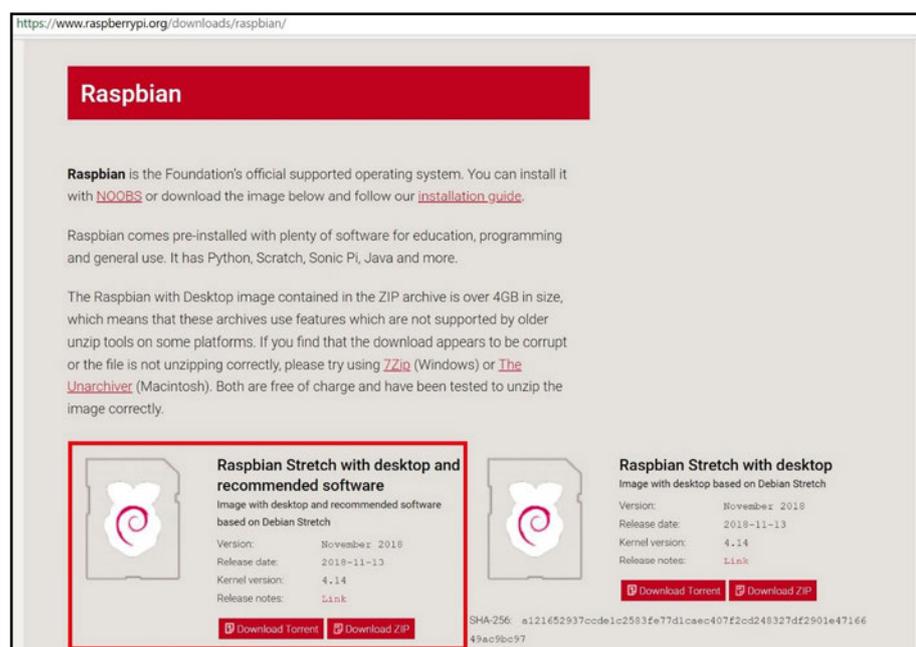


Figura 5: Download della distribuzione Raspbian Stretch



flash su scheda di memoria SD.

Successivamente, dovremo eseguire l'aggiornamento della distribuzione, verificando inoltre che Python sia correttamente installato (il comando `python` fa riferimento alla versione 2.x, mentre `python3` alla versione 3 di Python).

```
sudo apt-get update
python --version
python3 --version
```

TensorFlow ha come prerequisito la presenza della libreria di algebra lineare Atlas (Automatically Tuned Linear Algebra Software), che andrà installata tramite il seguente comando (l'output del comando è visibile in Figura 7):

Possiamo a questo punto installare TensorFlow tramite il gestore dei pacchetti di Python. Il comando da impartire alla shell è

```
pi@raspberrypi:~$ sudo apt-get install libatlas-base-dev
r/lib/libblas.so.3 (libblas.so.3) in auto mode
update-alternatives: using /usr/lib/atlas-base/atlas/liblapack.so.3 to provide /
usr/lib/liblapack.so.3 (liblapack.so.3) in auto mode
Setting up libgfortran-6-dev:armhf (6.3.0-18+rpil+deb9u1) ...
Processing triggers for libc-bin (2.24-11+deb9u3) ...
Processing triggers for man-db (2.7.6.1-2) ...
Setting up gfortran-6 (6.3.0-18+rpil+deb9u1) ...
Setting up gfortran (4:6.3.0-4) ...
update-alternatives: using /usr/bin/gfortran to provide /usr/bin/f95 (f95) in au
to mode
update-alternatives: using /usr/bin/gfortran to provide /usr/bin/f77 (f77) in au
to mode
Setting up libblas-dev (3.7.0-2) ...
update-alternatives: using /usr/lib/libblas/libblas.so to provide /usr/lib/libbl
as.so (libblas.so) in auto mode
Setting up libatlas-dev (3.10.3-1+rpil) ...
Setting up libatlas-base-dev (3.10.3-1+rpil) ...
update-alternatives: using /usr/lib/atlas-base/atlas/libblas.so to provide /usr/
lib/libblas.so (libblas.so) in auto mode
update-alternatives: using /usr/lib/atlas-base/atlas/liblapack.so to provide /us
r/lib/liblapack.so (liblapack.so) in auto mode
pi@raspberrypi:~$
```

Figura 7: Installazione della libreria Atlas

il seguente:

```
pip3 install --user tensorflow
```

Con una certa sorpresa possiamo constatare come l'installazione sia stata portata a termine in tempi molto ristretti e senza problemi, grazie al lavoro congiunto svolto dal Google Brain Team e dalla Raspberry Foundation. In Figura 8 possiamo osservare l'output prodotto dalla procedura di installazione di TensorFlow.

Per verificare che l'installazione sia andata a buon fine, creiamo un breve script di test, l'equivalente del classico "Hello World!". Creiamo quindi lo script `tftest.py` utilizzando l'editor `vi` (in alternativa si può utilizzare anche `nano` o un qualunque altro editor):

```
sudo vi tftest.py
```

Inseriamo poi il seguente codice all'interno del file:

```
import tensorflow as tf
hello = tf.constant('Hello, TensorFlow!')
sess = tf.Session()
print(sess.run(hello))
```

Con l'editor `vi` i comandi per inserire le linee di codice Python nel file script sono i seguenti:

- copiare nel blocco appunti il contenuto dello script;
- aprire `vi`, premere `Esc` e

```

pi@raspberrypi ~
File Edit Tabs Help
Downloading https://www.piwheels.org/simple/termcolor/termcolor-1.1.0-py3-none-any.whl
Collecting werkzeug>=0.11.10 (from tensorboard<1.12.0,>=1.11.0->tensorflow)
  Downloading https://files.pythonhosted.org/packages/20/c4/12e3e56473e52375aa29c4764e70d1b8f3efa6682bef8d0aae04fe335243/Werkzeug-0.14.1-py2.py3-none-any.whl (322kB)
  100% |#####| 327kB 690kB/s
Collecting markdown>=2.6.8 (from tensorboard<1.12.0,>=1.11.0->tensorflow)
  Downloading https://files.pythonhosted.org/packages/7a/6b/5600647404ba15545ec37d2f7f58844d690baf2f81f3a60b862e48f29287/Markdown-3.0.1-py2.py3-none-any.whl (89kB)
  100% |#####| 92kB 1.7MB/s
Collecting h5py (from keras-applications>=1.0.5->tensorflow)
  Downloading https://www.piwheels.org/simple/h5py/h5py-2.9.0-cp35-cp35m-linux_armv7l.whl (3.5MB)
  100% |#####| 3.5MB 79kB/s
Installing collected packages: six, wheel, numpy, werkzeug, setuptools, protobuf, grpcio, markdown, tensorboard, keras-preprocessing, absl-py, h5py, keras-applications, astor, gast, termcolor, tensorflow
Successfully installed absl-py-0.7.0 astor-0.7.1 gast-0.2.2 grpcio-1.18.0 h5py-2.9.0 keras-applications-1.0.7 keras-preprocessing-1.0.8 markdown-3.0.1 numpy-1.16.1 protobuf-3.6.1 setuptools-39.1.0 six-1.12.0 tensorboard-1.11.0 tensorflow-1.11.0 termcolor-1.1.0 werkzeug-0.14.1 wheel-0.32.3
pi@raspberrypi:~ $

```

Figura 8: Output prodotto dall'installazione di TensorFlow

poi i (per abilitare la modalità di inserimento del testo);

- nella finestra terminale di vi selezionare la voce di menu Edit->Paste;
- premere ancora Esc e poi :wq per salvare in modo permanente le modifiche al file.

Eseguiamo quindi lo script con Python3 (assumiamo ovviamente che il precedente test sulla presenza di Python3 abbia dato esito positivo):

```
python3 tftest.py
```

Se si sta utilizzando Python 3.5 si potranno a questo punto osservare diversi "runtime warning". Non preoccupatevi di ciò, è un

fatto già noto ed il consiglio ufficiale degli sviluppatori di TensorFlow è quello di ignorare tali warning. Se tutto è andato per il verso giusto (come è lecito aspettarsi), potremo osservare un output del comando analogo a quello visualizzato in Figura 9.

ESEMPIO: CLASSIFICATORE DI IMMAGINI

TensorFlow dispone su Gi-

tHub di un repository git in cui sono presenti dei modelli di rete neurale già preconfezionati, pronti per essere utilizzati all'interno del framework. Cominciamo con il creare sul Raspberry Pi una cartella (tf1) in cui andremo a versare il modello di esempio pre-

```

pi@raspberrypi ~
File Edit Tabs Help
pi@raspberrypi:~ $ python3 tftest.py
/usr/lib/python3.5/importlib/_bootstrap.py:222: RuntimeWarning: compiletime version 3.4 of module 'tensorflow.python.framework.fast_tensor_util' does not match runtime version 3.5
  return f(*args, **kwargs)
/usr/lib/python3.5/importlib/_bootstrap.py:222: RuntimeWarning: builtins.type size changed, may indicate binary incompatibility. Expected 432, got 412
  return f(*args, **kwargs)
b'Hello, TensorFlow!'
pi@raspberrypi:~ $ python3 --version
Python 3.5.3
pi@raspberrypi:~ $

```

Figura 9: Test della corretta installazione di TensorFlow



```
pi@raspberrypi: ~/tf1
File Edit Tabs Help
pi@raspberrypi:~$ python3 tftest.py
/usr/lib/python3.5/importlib/_bootstrap.py:222: RuntimeWarning: compiletime version 3.4 of module 'tensorflow.python.framework.fast_tensor_util' does not match runtime version 3.5
  return f(*args, **kwargs)
/usr/lib/python3.5/importlib/_bootstrap.py:222: RuntimeWarning: builtins.type size changed, may indicate binary incompatibility. Expected 432, got 412
  return f(*args, **kwargs)
b'Hello, TensorFlow!'
pi@raspberrypi:~$ python3 --version
Python 3.5.3
pi@raspberrypi:~$ cd
pi@raspberrypi:~$ mkdir tf1
pi@raspberrypi:~$ cd tf1
pi@raspberrypi:~/tf1$ git clone https://github.com/tensorflow/models.git
Cloning into 'models'...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 24298 (delta 53), reused 31 (delta 16), pack-reused 24216
Receiving objects: 100% (24298/24298), 563.39 MiB | 1.92 MiB/s, done.
Resolving deltas: 100% (14381/14381), done.
Checking out files: 100% (2768/2768), done.
pi@raspberrypi:~/tf1$
```

Figura 10: Clonazione del repository git contenente i modelli

levato da GitHub.

```
mkdir tf1
cd tf1
```

Possiamo ora procedere con la clonazione, nella nuova cartella appena creata, del repository contenente i modelli TensorFlow:

```
git clone https://github.com/tensorflow/models.git
```

L'output del comando di clonazione è visibile in Figura 10.

Di questi modelli siamo interessati ad utilizzare il classificatore di immagini, ovvero uno strumento che permette di classificare un'immagine fornita in ingresso, stabilendo a quale categoria (tra un gruppo di immagini utilizzato per il training della rete neurale del modello) essa appartiene. Il classificatore

di immagini è contenuto nella cartella `models/tutorials/image/imagenet` e dovremo pertanto impartire da terminale il comando `cd` per cambiare la directory corrente:

```
cd models/tutorials/image/imagenet
```

Lo script del classificatore di immagini viene eseguito di default con l'immagine di un simpatico panda, un file in formato jpeg con dimensione 100 x 100 pixel visibile

(ingrandita per esigenze tipografiche) nella sottostante Figura 11. Per eseguire il classificatore di immagini fornendo in ingresso l'immagine del panda occorre digitare nel terminale il seguente comando:

```
python3 classify_image.py
```



L'effetto è quello di alimentare la rete neurale con l'immagine del panda: in uscita potremo poi osservare la "predizione" sulla classe di appartenenza dell'oggetto identificato eseguita dalla rete neurale. Sappiamo che tale predizione viene espressa in termini di probabilità (dove 1.0 equivale alla certezza assoluta). L'uscita di un classificatore

di immagini è pertanto solitamente rappresentata da più classi di oggetti, ciascuna con la sua probabilità (il totale delle stesse deve essere pari a 1). Il risultato prodotto dal classificatore è visibile in Figura 12: come era lecito aspettarsi (non dimentichiamo che l'immagine utilizzata è quella utilizzata di default dall'esempio del classificatore) la rete neurale ha operato correttamente, attribuendo una probabilità quasi pari al 90% al panda. Tra le altre possibili scelte, notiamo la presenza di una mela zuccherina (custard apple), alla quale è tuttavia assegnato un livello di confidenza quasi nullo (0.00147 per l'esattezza). Il riconoscimento (o classificazione delle immagini) è stato eseguito dalla rete neurale utilizzando ImageNet [3], un database di immagini preconfezionate organizzate in una struttura gerarchica. Il progetto ImageNet contiene attualmente oltre 14 milioni di immagini, ognuna delle quali è



Figura 13: Immagine di un calice utilizzata per il secondo test

stata annotata a mano per indicare gli oggetti in essa contenuti (ad esempio panda, oppure mela zuccherina, ecc.) e almeno un milione di queste immagini contiene anche dei rettangoli di selezione (bounding box) attorno ai singoli oggetti in essa presen-

ti. ImageNet include oltre 20000 categorie di oggetti, con ciascuna categoria composta da diverse centinaia di immagini.

Test con un'altra immagine
Diciamoci la verità, eseguire il test del classificatore con l'immagine utilizzata dagli stessi sviluppatori del modello non ci basta: vogliamo mettere alla prova le capacità della rete neurale con un'altra immagi-

```

pi@raspberrypi: ~/tf1/models/tutorials/image/imagenet
File Edit Tabs Help
return f(*args, **kwargs)
>> Downloading inception-2015-12-05.tgz 100.0%
Successfully downloaded inception-2015-12-05.tgz 88931400 bytes.
2019-02-01 13:12:30.632887: W tensorflow/core/framework/op_def_util.cc:355] Op BatchNormWithGlobalNormalization is deprecated. It will cease to work in GraphDef version 9. Use tf.nn.batch_normalization().
2019-02-01 13:12:33.231090: W tensorflow/core/framework/allocator.cc:113] Allocation of 3981312 exceeds 10% of system memory.
2019-02-01 13:12:33.335727: W tensorflow/core/framework/allocator.cc:113] Allocation of 6193152 exceeds 10% of system memory.
2019-02-01 13:12:33.376409: W tensorflow/core/framework/allocator.cc:113] Allocation of 3670016 exceeds 10% of system memory.
2019-02-01 13:12:33.383744: W tensorflow/core/framework/allocator.cc:113] Allocation of 6193152 exceeds 10% of system memory.
2019-02-01 13:12:33.408965: W tensorflow/core/framework/allocator.cc:113] Allocation of 8257536 exceeds 10% of system memory.
giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca (score = 0.89107)
indri, indris, Indri indri, Indri brevicaudatus (score = 0.00779)
lesser panda, red panda, panda, bear cat, cat bear, Ailurus fulgens (score = 0.00296)
custard apple (score = 0.00147)
earthstar (score = 0.00117)
pi@raspberrypi:~/tf1/models/tutorials/image/imagenet $

```

Figura 12: Output prodotto dal classificatore di immagini



ne, scelta da noi. La cosa importante è che l'immagine sia ben chiara, di dimensioni non eccessive e soprattutto contenga il target da riconoscere all'interno di un riquadro ben definito (nel gergo dell'elaborazione delle immagini, si dice che l'immagine stessa deve essere "cropped"). Come ulteriore immagine ho scelto un soggetto abbastanza comune, un bicchiere (in particolare si tratta di un calice), visibile in Figura 13.

Diamo quindi nuovamente il comando di attivazione al classificatore di immagini, utilizzando però questa volta non l'immagine di default, ma quella da noi scelta (specificata tramite il parametro `--image_file`):

```
python3 classify_image.py --image_file=/home/pi/bicchiere.jpg
```

Il risultato prodotto da questo secondo test è visibile in Figura 14. Non male! Possiamo osservare come la rete neurale abbia assegnato una probabilità superiore al 90% all'oggetto goblet (calice).

CONCLUSIONI

Abbiamo visto in questo articolo come TensorFlow, una delle più diffuse piattaforme per lo sviluppo di applicazioni machine e deep learning, sia installabile e utilizzabile anche su un pic-

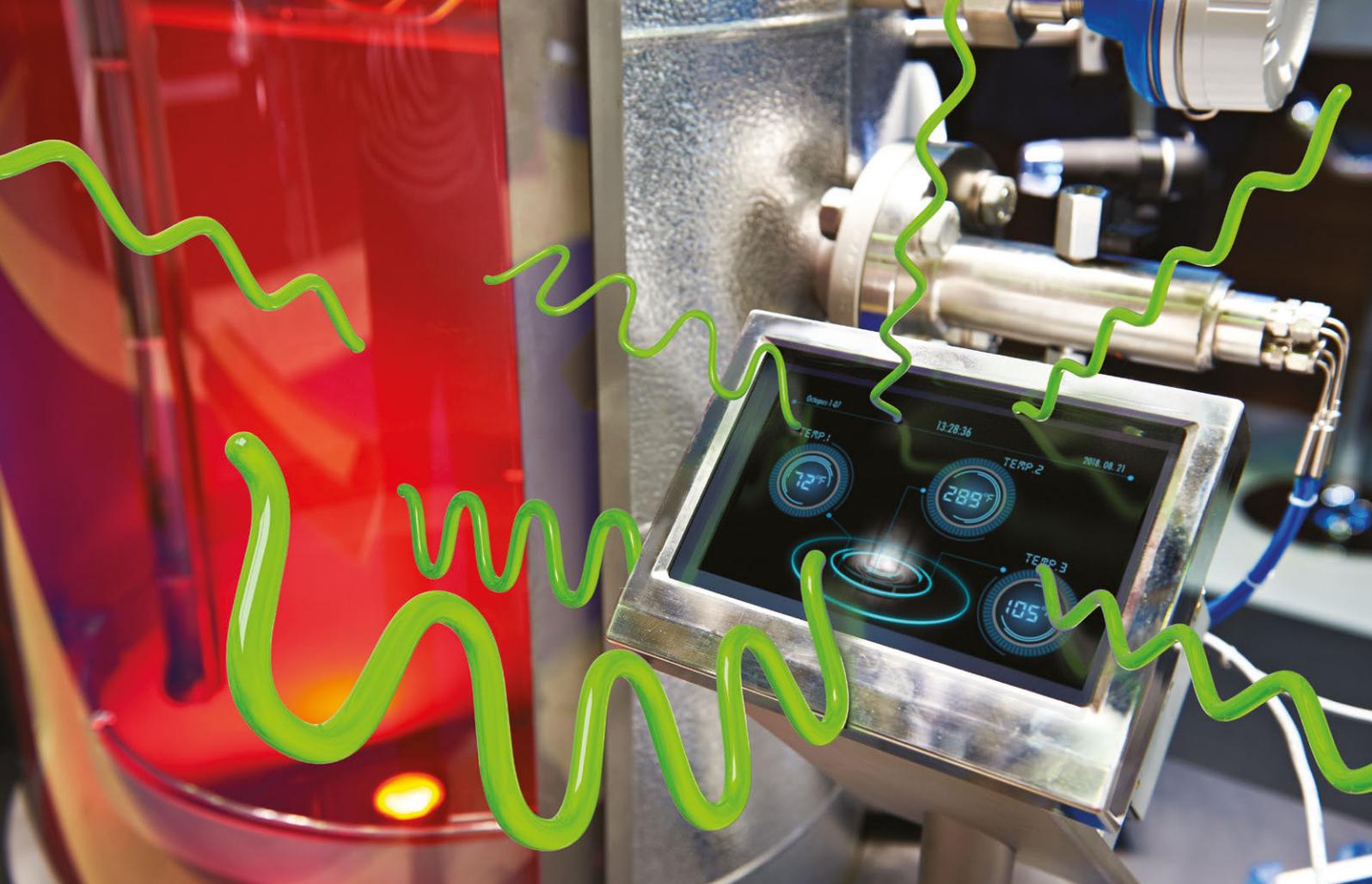
colo sistema hardware embedded come il Raspberry Pi. L'applicazione di esempio utilizzata, anche se basilare, dimostra un grande potenziale. Il riconoscimento degli oggetti è stato infatti eseguito interamente sul Raspberry Pi, senza necessità di alcuna connessione alla rete internet. Ciò significa che aggiungendo un sistema autonomo di acquisizione delle immagini (come il Pi camera module o una normale webcam USB) il progetto diventerebbe completamente autonomo, in grado di acquisire e riconoscere immagini del mondo reale. Ma questo lo vedremo in un prossimo articolo.

RIFERIMENTI

- [1] <https://www.raspberrypi.org/downloads/raspbian/>
- [2] <https://www.balena.io/etcher/>
- [3] <http://www.image-net.org>

```
pi@raspberrypi: ~/tf1/models/tutorials/image/imagenet
File Edit Tabs Help
runtime version 3.5
  return f(*args, **kwargs)
/usr/lib/python3.5/importlib/_bootstrap.py:222: RuntimeWarning: builtins.type si
ze changed, may indicate binary incompatibility. Expected 432, got 412
  return f(*args, **kwargs)
2019-02-01 14:03:13.460384: W tensorflow/core/framework/op_def_util.cc:355] Op B
atchNormWithGlobalNormalization is deprecated. It will cease to work in GraphDef
 version 9. Use tf.nn.batch_normalization().
2019-02-01 14:03:16.074000: W tensorflow/core/framework/allocator.cc:113] Alloca
tion of 6193152 exceeds 10% of system memory.
2019-02-01 14:03:16.122883: W tensorflow/core/framework/allocator.cc:113] Alloca
tion of 6193152 exceeds 10% of system memory.
2019-02-01 14:03:16.146788: W tensorflow/core/framework/allocator.cc:113] Alloca
tion of 8257536 exceeds 10% of system memory.
2019-02-01 14:03:16.639562: W tensorflow/core/framework/allocator.cc:113] Alloca
tion of 5531904 exceeds 10% of system memory.
2019-02-01 14:03:16.770262: W tensorflow/core/framework/allocator.cc:113] Alloca
tion of 5531904 exceeds 10% of system memory.
goblet (score = 0.90722)
red wine (score = 0.01875)
cup (score = 0.00410)
wine bottle (score = 0.00285)
corkscrew, bottle screw (score = 0.00216)
pi@raspberrypi:~/tf1/models/tutorials/image/imagenet $
```

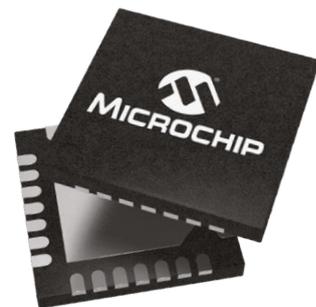
Figura 14: Output prodotto dall'esecuzione del modello con l'immagine da noi fornita



Misura il mondo attorno a te

Soluzioni per condizionare ogni segnale

Colmare il divario tra il mondo in cui viviamo e il mondo digitale è un compito complesso. I prodotti standalone mixed-signal, lineari e di interfaccia di Microchip consentono ai progettisti di raccogliere, misurare, monitorare e convertire in modo semplice ed accurato un'ampia gamma di segnali per migliorare la progettazione dei prodotti. Oltre ad una vasta gamma di prodotti di uso generale, è disponibile anche una selezione di dispositivi ad alte prestazioni per aiutarti a soddisfare i tuoi severi requisiti di progettazione.



Guarda la mostra offerta per il condizionamento di segnali!

Esplora la nostra offerta su www.microchip.com/Signal





Giochiamo con i Regolatori Switching

di Walter Ribbert

redazione@elettronicaemaker.it

Era da un po' che volevo far delle prove con i regolatori switching, tipo step-down e step-up; mi è venuta l'occasione quando ho trovato dei caricabatterie da auto per telefoni cellulari, ancora nuovi in blister a due pezzi per un euro su una bancarella, sono datati, hanno il connettore adatto ai vecchi iPhone. Ne ho acquistati quattro, due euro per giocare mi è parso un buon prezzo.

Questi caricabatterie sono dei DC/DC converter switching step-down non isolati, ingresso 12-24V e uscita 5V / 650mA, basati sull'integrato MC34063, che sembra molto popolare a giudicare dal numero di schemi pubblicati sul web. L'idea è stata di realizzare un paio di semplici circuiti, utilizzando i componenti recuperati, testarli con l'ausilio di multimetro e oscilloscopio e farci qualche ragionamento teorico; approfittando anche delle correnti in gioco relativamente basse.

L'INTEGRATO MC34063

Vediamo innanzi tutto com'è fatto per sommi capi l'integrato MC34063. Come si vede dagli schemi l'integrato ha un design interno minimale: un oscillatore, un riferimento di 1,25V, un comparatore e un flip-flop RS che pilota l'attuatore di uscita (i due transistor switching collegati in

darlington). La disponibilità sui vari pin di queste funzionalità permette una facile configurazione e diverse soluzioni circuitali, il datasheet, ma più ancora le note applicative (AN920/D) offrono molti spunti di utilizzo. L'approccio utilizzato per il calcolo dei componenti cruciali del circuito è anche auto esplicativo sul principio di

funzionamento di questa tecnologia e non fa uso di tabelle per la scelta di induttanze e capacità, ma solo semplici formule accessibili a tutti. Vediamo un po' meglio questi elementi: l'oscillatore, si basa sulla carica/scarica asimmetrica a corrente costante di un condensatore esterno CT (Figura 2), la carica dura 6 volte la scarica.

Durante la carica di CT il transistor Q1 può essere chiuso (set flip-flop RS se l'uscita del comparatore è alta) ed è aperto durante la scarica (reset flip-flop RS). Due caratteristiche lo rendono diverso da altri regolatori simili (ad esempio LM2574), il limitatore di corrente e il comparatore riferimento vs. feedback non agiscono direttamente sul transistor in uscita (vedi AN920/D). La limitazione di corrente rileva la caduta di tensione su un resistore shunt, R_{sc} posto tra i pin 6 e 7, generalmente in serie al transistor finale. Quando la caduta di tensione arriva agli 0,3V circa agisce sull'oscillatore, aumen-



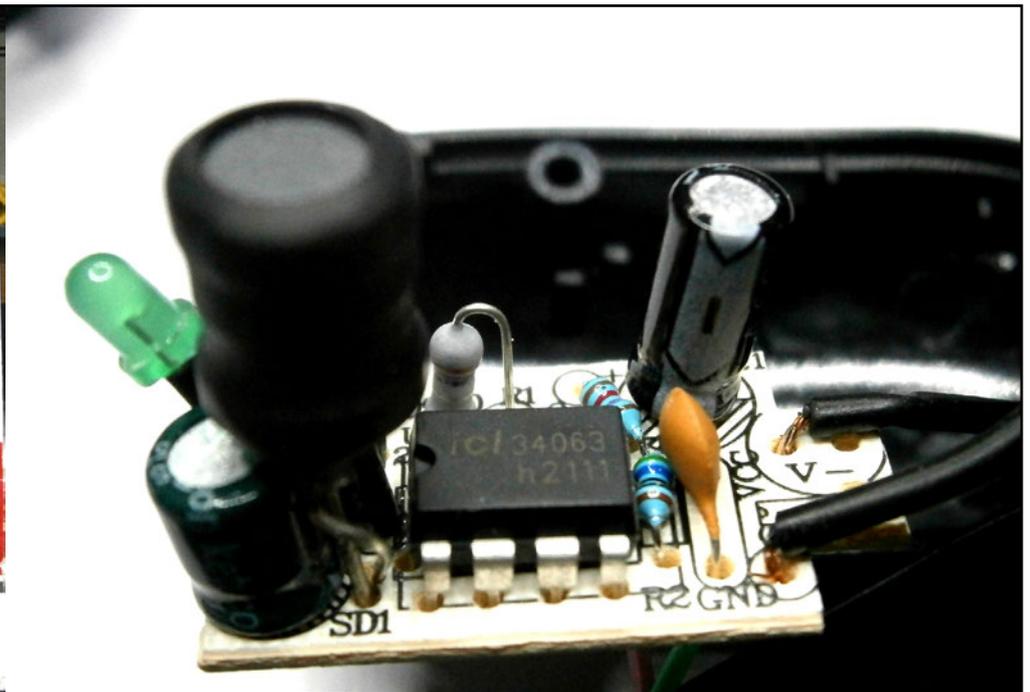


Figura 1. Caricatore in blister e circuito interno

ta la corrente di carica di CT avvicinandola al valore di quella di scarica, accorciando di molto il tempo di conduzione del transistor (Figura 3). Anche il comparatore agisce in modo indiretto, se la tensione di feedback è minore del riferimento (1,25V), la sua uscita va alta, in AND con quella dell'oscillatore setta il flip-flop RS durante la fase di carica e quindi chiude il transistor finale. Il reset del flip-flop è comandato esclusivamente dall'oscillatore, se la tensione di feedback supera il riferimento il comparatore blocca il set del flip-flop RS, ma al ciclo successivo, quello in corso viene terminato. In Figura 3 sono illustrate le forme d'onda tipiche relative alle funzioni descritte, si noti il cambio di pendenza della carica di CT che evidenzia l'intervento della limitazione di corrente.

I° CIRCUITO: PICCOLO ALIMENTATORE STABILIZZATO REGOLABILE

In Figura 4 lo schema dell'alimentatore, si tratta praticamente dello stesso circuito del carica batteria (Step-Down) con il partitore resistivo (R1-R2) variabile per variare la tensione in uscita e con l'aggiunta di ponte raddrizzatore e condensatore di livellamento. Il circuito va completato di un trasformatore, il cui secondario deve poter fornire almeno 1,5 A e almeno 5V rms di tensione maggiore della massima tensione di uscita del regolatore. Attenzione! La massima tensione in ingresso è di 28Vac, cioè circa 40V ($28 * \sqrt{2}$) in dc sul condensatore C_i che sono il massimo che l'MC34063 sopporta tra i pin e massa e ai capi di Q1 e Q2. Nel calcolo, per sicurezza, è meglio non considerare la caduta di tensione sui

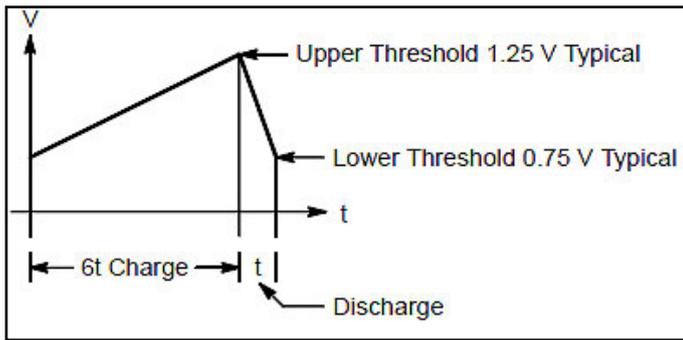


Figura 2. Forma d'onda asimmetrica su CTMaster - Slave

diodi D1 - D4, è variabile in funzione della corrente. Si può usare il circuito come riduttore di tensione eliminando il ponte a diodi, in questo caso la tensione V_{dc} d'ingresso dovrebbe essere almeno 5Vdc superiore a quella d'uscita e il condensatore C_i può essere ridotto a $100\mu F$. Con il potenziometro R2 da $10k\Omega$ la tensione massima in uscita è $= V_{ref} \cdot (R1+R2)/R1 = 1,25V \cdot 11 = 13.75V_{dc}$. Con il resistore Rsc da 0.27Ω la limitazione di corrente interviene a $1,2A$ circa ($0,33V$ di soglia max. tra V_{cc} pin 6 e I_{pk} sense pin 7), la corrente massima in uscita al circuito, a tensione costante, è quindi la metà $600mA$. La corrente in corto circuito è di circa $1,5A$, perché come detto la limitazione riduce il tempo di carica di C_T ad un sesto, ma non lo azzerava e quindi c'è un minimo tempo di conduzione di Q1.

Del circuito originale ho 'riciclato': l'MC34063, l'induttanza L, il diodo schottky D5, il condensatore C_T e il resistore Rsc. Il diodo D5 deve

essere 1N5819 da 40V di tensione inversa (alcuni caricabatterie usano 1N5818, reverse voltage = 30V). Ho dovuto utilizzare condensatori elettrolitici diversi, per via del valore (C_i è passato da $47\mu F$ a $1000\mu F$), ma soprattutto per il cambiamento delle tensioni d'ingresso e uscita del circuito (C_o è rimasto da $220\mu F$ ma ora è da 40V anziché 10V). I resistori R1 e R2 sono ovviamente cambiati, R3 rappresenta un carico minimo in assenza di quello esterno. Due parole sul layout: il diodo D5, l'induttanza L e il condensatore C_o dovrebbero essere montati vicini con collegamenti corti (come schema), C_i e C_o dovrebbero essere a bassa resistenza interna (ESR); questo per limitare le perdite di energia agevolando il passaggio di corrente fra C_i , L, C_o e D5 e anche limitare le emissioni elettromagnetiche. Lo schema raffigura come dovrebbero essere eseguiti i collegamenti;

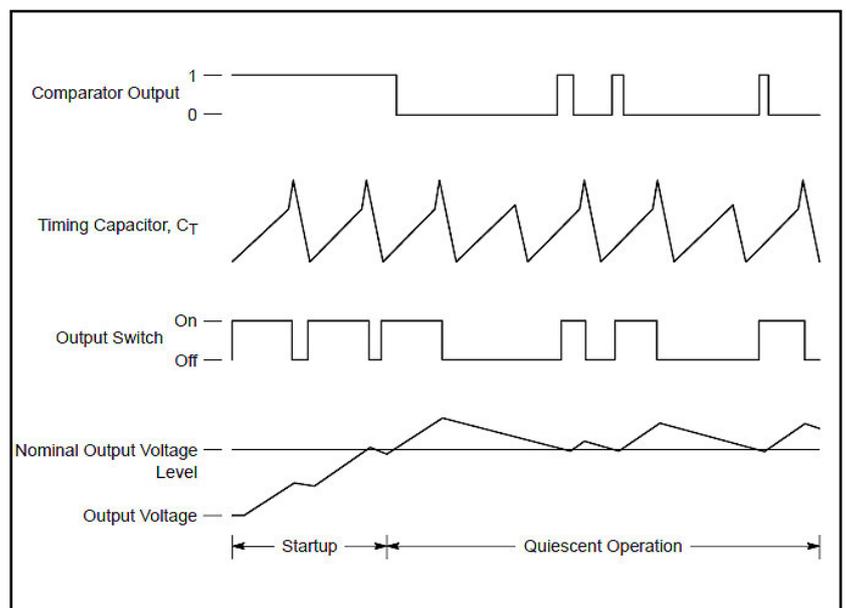


Figura 3. Forme d'onda tipiche del circuito

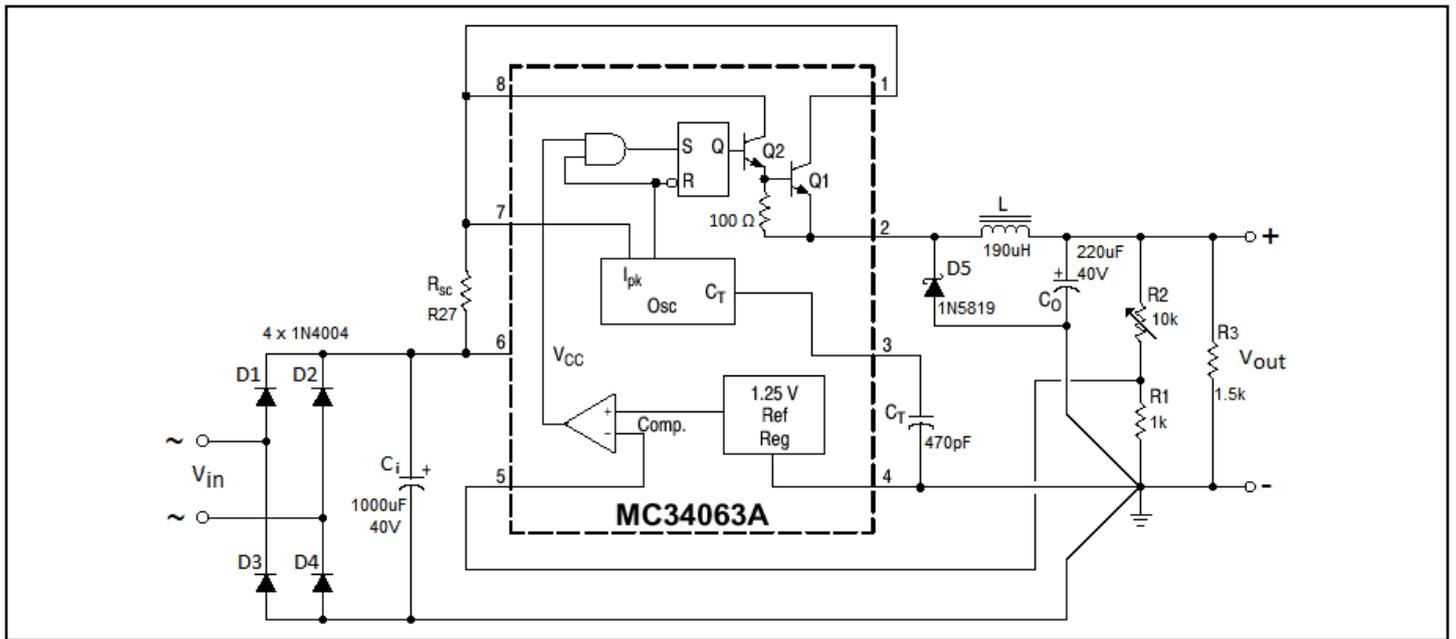


Figura 4. Schema dell'alimentatore regolabile

notare il punto comune di massa all'uscita, serve per evitare alterazioni al valore della tensione di feedback (pin 5) e al valore del GND di riferimento del C.I. (pin 4) per effetto delle cadute di tensione dovute alle correnti di picco circolanti nel ritorno di massa.

PROVA DELL'ALIMENTATORE

Ho alimentato il circuito con un trasformatore da campanelli da 12Vac / 15VA (14,3Vac a vuoto) e poi con un alimentatore esterno in corrente continua, come carico ho usato dei resistori da 47 Ohm 5W collegati in parallelo in vario numero. Nella Tabella 1

sono riportate le misure fatte nelle varie condizioni con il trasformatore (sfondo arancio) e con l'alimentatore esterno (sfondo azzurrino). La tensione V_{in} e la corrente I_{in} sono misurate all'ingresso dopo il ponte su C_i . Le righe con sfondo rosa si riferiscono a prove di corto circuito, la riga rossa è quando l'integrato ha ceduto 'cuocendosi' (credo per una disattenzione) prima di riuscire a misurare il valore della corrente d'ingresso. Come si vede dalla prima riga della Tabella 1, con il trasformatore il regolato-

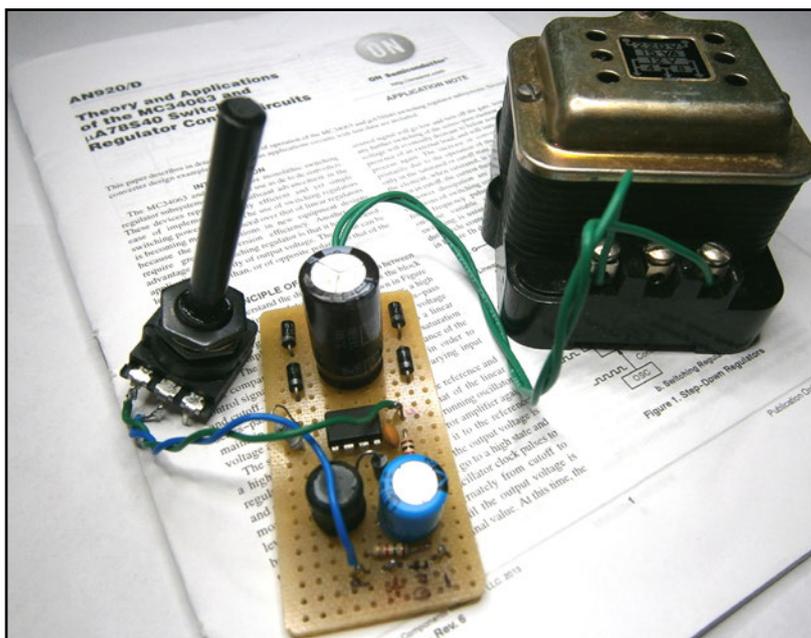


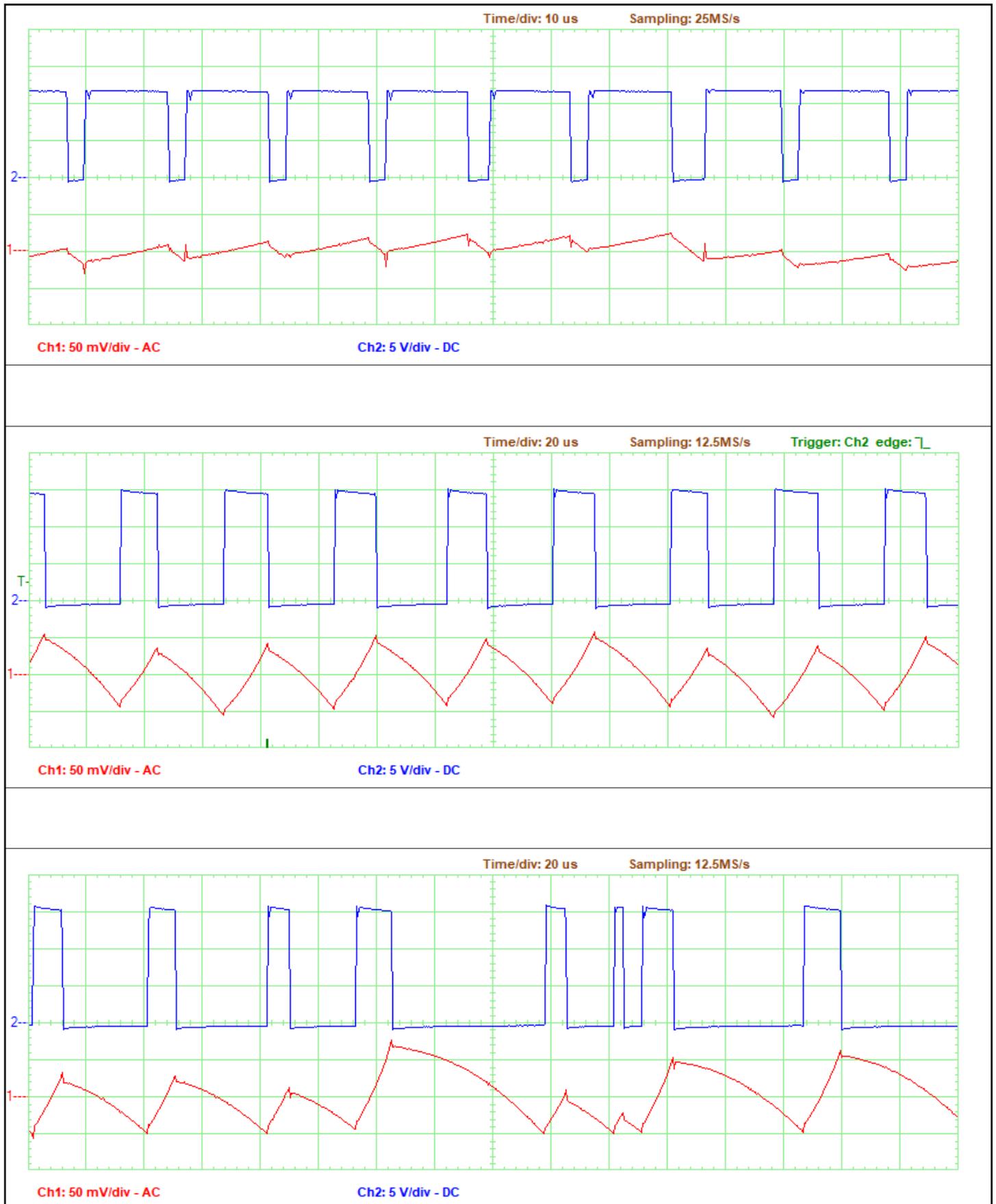
Figura 5. Alimentatore regolabile assemblato

Vin (dc)	Iin (dc)	Win	Vo (dc)	Io (dc)	Wo	W diss.	Rend. %	W diss. lineare
13,8 V	470 mA	6,486 W	9 V	560 mA	5,04 W	1,446 W	77,7 %	2,69 W
16 V	210 mA	3,36 W	5 V	520 mA	2,6 W	0,76 W	77,4 %	5,72 W
17 V	120 mA	2,04 W	3 V	490 mA	1,47 W	0,64 W	72,7 %	6,86 W
Vin (dc)	Iin (dc)	Win	Vo (dc)	Io (dc)	Wo	W diss.	Rend. %	W diss. lineare
12 V	345 mA	4,14 W	5V	620 mA	3,1 W	1,04 W	74,8 %	4,34 W
12 V	172 mA	2,06 W	3 V	490 mA	1,47 W	0,68 W	71,3 %	4,41 W
18 V	464 mA	8,35 W	13 V	540 mA	7,02 W	1,33 W	84 %	2,7 W
18 V	420 mA	7,56 W	10 V	620 mA	6,2 W	1,36 W	82 %	4,96 W
18 V	225 mA	4,05 W	5 V	620 mA	3,1 W	0,95 W	76,5 %	8,06 W
18 V	115 mA	2,07 W	3 V	490 mA	1,47 W	0,6 W	71 %	7,35 W
24 V	350 mA	8,4 W	13 V	540 mA	7,02 W	1,38 W	83,5 %	5,94 W
24 V	315 mA	7,56 W	10 V	620 mA	6,2 W	1,36 W	82 %	8,68 W
24 V	168 mA	4,03 W	5 V	620 mA	3,1 W	0,93 W	76,9 %	11,78 W
24 V	90 mA	2,16 W	3 V	490 mA	1,47 W	0,69 W	68 %	10,29 W
Vin (dc)	Iin (dc)	Win	Vo (dc)	Io (dc)	Wo	W diss.		
12 V	216 mA	2,58 W	0,3 V	1,24 A	0,372 W	2,21 W		
18 V	170 mA	3,06 W	0,3 V	1,3 A	0,39 W	2,67 W		
24 V	?	?	0,3 V	1,35 A	0,405 W	?		

Tabella 1

re arriva a fornire in uscita 9V con 560mA di carico, la tensione sul secondario sotto carico scende a 13Vac (misura non rms) e risulta molto distorta, dopo il ponte di diodi su Ci il valore medio è di circa 13,8Vdc, la corrente media d'ingresso Iin di 470 mA (credo che al picco sia circa il doppio), il punto più basso dell'ondulazione a 100Hz è 12V. A tensioni di uscita minori va meglio, la tensione sul secondario sale perché diminuisce la corrente da erogare anche se

la corrente in uscita praticamente non varia, in questo il regolatore a commutazione è migliore rispetto ad un lineare; la corrente di uscita di un lineare sarebbe uguale alla corrente entrante. Il trasformatore per campanelli utilizzato non è proprio il massimo, è probabilmente realizzato per servizio intermittente non adatto a fornire picchi di corrente, ha un buon nucleo per limitare le perdite nel ferro e scaldare poco ma probabilmente è più scarso nel rame.



Oscillogrammi 1, 2, 3 (dall'alto)

Nelle varie colonne della Tabella 1 sono indicati i dati di tensione, corrente, potenza d'ingresso e di uscita, la potenza dissipata nei componenti, il rendimento in potenza e la potenza ipoteticamente dissipata se il regolatore fosse un lineare (colonna grigia [$W = (V_{in} - V_o) \cdot I_o$]). Diversamente dai regolatori di tensione serie che dissipano in calore l'energia in surplus per regolare la tensione in uscita V_o , negli switching l'energia dispersa è data quasi esclusivamente dalla caduta di tensione su Q_1 , D_5 e dalle resistenze ohmiche di L (filo di rame) e di C_o (ESR) che sono molto basse. Per questo gli switching sono vantaggiosi nei casi in cui il divario fra le tensioni d'ingresso e uscita è medio-alto. Negli oscillogrammi 1, 2 e 3 si possono vedere l'andamento della tensione sul catodo di D_5 a monte di L (Canale 2 traccia blu) e l'espansione del valore di cresta della tensione di uscita V_o a valle di L (Canale 1 traccia rossa) evidenziando il ripple. Il ripple in uscita risulta essere minore se il divario tra tensione d'ingresso e d'uscita ($V_{ci} - V_o$) è bassa e la corrente del carico I_o è alta, nell'oscillogramma 1 è di circa 30 mV p-p, negli oscillogrammi 2 e 3 si vede che aumenta molto se aumenta il divario ($V_{ci} - V_o$). La condizione peggiore di ripple molto alto si ha con alta differenza di tensione e basso carico. I regolatori lineari hanno ripple molto basso (praticamente inesistente) mentre gli switching sono di gran lunga peggiori e necessitano, a seconda dell'uso, di filtri LC aggiuntivi all'uscita.

UN PO' DI TEORIA SUL REGOLATORE STEP-DOWN

Ora un po' di teoria, ma senza annoiarsi troppo. Dunque, in Figura 6 si vede il circuito ideale teorico del regolare step-down (detto anche buck converter) di Figura 4. Bene, vediamo come funziona: l'interruttore Q_1 (il transistor del nostro alimentatore) viene chiuso e aperto alternativamente generando un'onda quadra con ampiezza V_{in} e con un dato valore di duty cycle, in funzione della tensione di uscita V_o . Quando Q_1 si chiude circola corrente (linea rossa) attraverso l'induttanza L che si carica elettromagneticamente attraverso il condensatore C_o e il carico R_L . Quando Q_1 si apre la corrente indotta per autoinduzione dall'induttanza L si scarica sul condensatore C_o e il carico R_L attraverso il diodo D_5 (linea blu). Il filtro 'attivo' L/C estrae il valore medio V_o dall'onda quadra generata da Q_1 modulando la V_{in} , per cui [$V_o = V_{in} \cdot \text{DutyCycle}$]. Quanto detto è valido per un ben preciso valore delle tensioni V_{in} ,

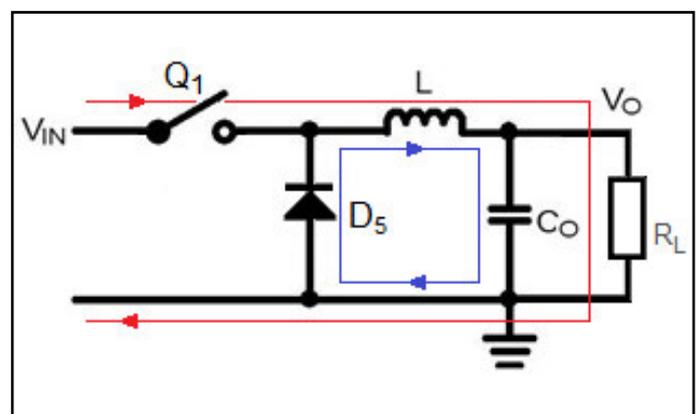
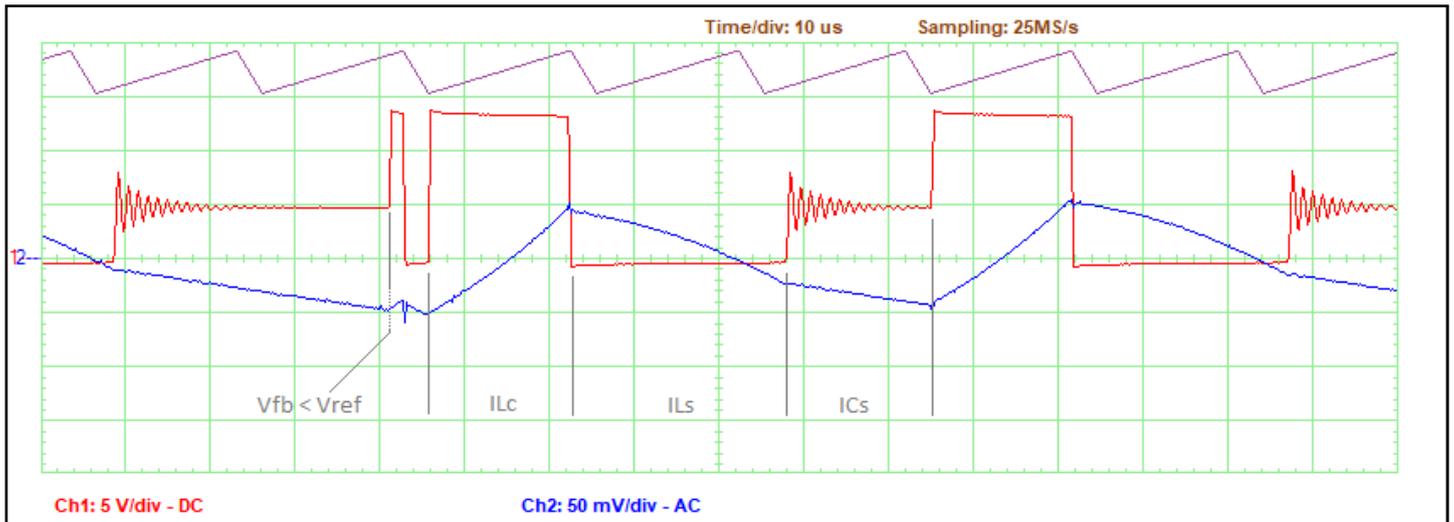


Figura 6; Schema ideale del regolatore step-down



Oscillogramma 4

V_o e corrente di carico I_o . Se V_{in} e/o I_o variano si deve fare in modo che la tensione V_o si discosti il meno possibile dal valore teorico scelto. Questo avviene attraverso il comparatore [Comp] interno all'integrato che mette in relazione la tensione di feedback con quella del riferimento interno (1,25V) e lo fa agendo sul flip-flop RS di comando di Q1 come descritto nel paragrafo dedicato al MC34063 variando il duty cycle. Nell'oscillogramma 4 si vede bene come ciò avviene, dove in rosso (Canale 1) abbiamo la tensione sul catodo di D5, in blu (Canale 2) è evidenziata la cresta della tensione di uscita e in violetto la traccia indicativa della carica / scarica di CT dell'oscillatore di base. Quando la tensione V_o si abbassa e la tensione di feedback diventa minore della V_{ref} (punto $V_{fb} < V_{ref}$) viene comandato Q1 che si chiude. Il tratto ILc (Q1 chiuso) evidenzia la carica in corrente dell'induttanza L e di tensione

del condensatore C_o . Nel tratto ILs (Q1 aperto), l'induttanza L si scarica completamente sul condensatore e sul carico RL attraverso la conduzione del diodo D5 (la cui tensione risulta negativa di circa 0,5 V). Dato che il carico non richiede molta corrente la tensione su V_o rimane ancora alta e si vede nella fase ICs, ora è il condensatore C_o ad alimentare il carico RL mentre l'induttanza L è inattiva avendo esaurito la carica elettromagnetica. Al momento in cui si verifica nuovamente $V_{fb} < V_{ref}$ il ciclo si ripete. Negli oscillogrammi 1, 2 e 3 il carico è più alto, le fasi ILs e ICs sono contemporanee e la condizione $V_{fb} < V_{ref}$ si verifica prima.

L'oscillazione smorzata sul catodo di D5, è prodotta dalla risonanza dell'induttanza L (190 μ H) con le capacità parassite di D5 (~100 pF), del cablaggio e della sonda dell'oscilloscopio (~15 pF). La frequenza è di circa 900 kHz.

COME CALCOLARE I VALORI DI L, CO E CT

Ho realizzato il circuito assemblando i componenti recuperati senza nessun calcolo, ma come saltano fuori i valori dei componenti tipici del circuito, L, Co e CT?

Proviamo a seguire le note applicative e ricalcolare i componenti del circuito, e facendolo capiamo un po' di più come dimensionare un circuito switching step-down (detto anche buck regulator).

Ok, il primo vero passo (che definisco 0) è stabilire i dati caratteristici del convertitore:

- $V_{out} = 5V$; $I_{out} = 650mA$ max; Valori dichiarati sulla targhetta del caricabatteria;
- $V_{in} = 10V_{min} - 26V$ max; +/-2V di margine sulle tensioni 12V e 24V nominali.
- Freq. Osc. = 50kHz; frequenza oscillatore. Questo è il valore utilizzato nelle note applicative ed è la metà del massimo valore di funzionamento del MC34063.
- $V_{ripple}(p-p) = 0,5\% V_{out} = 25mV$;

Passo 1, il filtro LC converte in valore medio (V_o) l'onda quadra a tensione V_{in} che gli viene applicata, il cui duty cycle è proporzionale al rapporto V_o / V_{in} (al netto delle cadute di tensione su transistor e diodo). Quindi: $V_o / V_{in} = t_{on} / (t_{on} + t_{off}) = \text{duty cycle}$. Il MC34063 però regola il tempo t_{on} variando il tempo totale del ciclo di commutazione. Per cui come esemplificato

nelle note applicative, si parte calcolando il rapporto fra $t_{on(max)} / t_{off}$, dove t_{on} è il tempo massimo di conduzione di Q1 con la V_{in} al valore minimo:

$$t_{on(max)} / t_{off} = (V_o + V_{D5}) / (V_{in(min)} - V_{Q1(sat)} - V_o);$$

$$\Rightarrow (5V + 0,8V) / 10V - 0,8V - 5V = 1,38;$$

Passo 2, ora stabiliamo il tempo del ciclo completo, cioè l'inverso della frequenza scelta per l'oscillatore:

$$t_{on} + t_{off} = 1 / 50kHz = 20 \mu s.$$

Passo 3, ora ricaviamo i due tempi:

$$t_{off} = (t_{on} + t_{off}) / (t_{on} / t_{off}) + 1;$$
$$t_{on(max)} = (t_{on} + t_{off}) - t_{off};$$
$$\Rightarrow t_{off} = 20 \mu s / 1.38 + 1 = 8,40 \mu s;$$
$$t_{on(max)} = 20 \mu s - 8,40 \mu s = 11,6 \mu s$$

Passo 4, da qui possiamo calcolare il valore del condensatore dell'oscillatore:

$$C_T = I_{chg(min)} * (dt/dV);$$

dove I_{chg} è la minima corrente costante di carica di C_T , dal datasheet si prende il valore minimo (20 uA suggerito nelle App. Note), dt è il tempo di carica (cioè $t_{on(max)}$) e dV è la differenza tra le soglie di tensione minima e massima di carica e scarica di C_T , 0,5 V. Quindi:



$$C_T = I_{\text{chg}(\text{min})} * (dt/dV) = 20 \mu\text{A} * (11,6 \mu\text{s} / 0,5 \text{V}) = 464 \text{pF}, \text{ (valore commerciale } 470\text{pF)}$$

Passo 5, ora dobbiamo stabilire la corrente I_{pk} , cioè la corrente massima che deve commutare il transistor Q1.

È facile, come si vede dalle forme d'onda riportate sulle note applicative, la corrente in Q1 è la stessa di L che ha un andamento triangolare il cui valore medio è I_{out} , cioè 650mA.

$$\text{Quindi } I_{pk} = 2 * I_{out} = 1,3 \text{ A.}$$

Passo 6, conoscendo I_{pk} e t_{on} troviamo il valore dell'induttanza utilizzando il valore di V_{in} massima:

$$L = (V_{in(\text{max})} - V_{Q1} - V_o) / I_{pk} * t_{on} = (26\text{V} - 0,8\text{V} - 5\text{V}) / 1,3\text{A} * 11,6 \mu\text{s} = 180 \mu\text{H}$$

Passo 7, il resistore per la limitazione di corrente è:

$$R_{sc} = V_{ipk(\text{max})} / I_{pk} = 0,33\text{V} / 1,3\text{A} = 0,253 \text{ Ohm}$$

Il valore utilizzato è 0,27 Ohm, con questo valore il limite è 1,23A che significa una I_{out} di 615mA circa, è più basso del valore indicato in targhetta.

Passo 8, e ora l'ultimo componente che caratterizza il convertitore, C_o .

Va dimensionato in funzione del ripple scelto: 25mV.

L'equazione semplificata è:

$$C_o = I_{pk} * (t_{on} + t_{off}) / 8 * V_{\text{ripple}(\text{p-p})} \\ \Rightarrow C_o = 1,3\text{A} * (20\mu\text{s}) / 8 * 25\text{mV} = 130 \mu\text{F} \\ \text{(montato } 220 \mu\text{F, valore molto più elevato).}$$

A quanto pare i valori quadrano abbastanza con quanto montato nel circuito originale del riduttore, spulciando le note applicative ho trovato un circuito (AN920/D, Figura 21) che ha i valori di L e C_o troppo simili a quelli montati. Sarà un caso? Mah. Le note applicative AD920/D sono ben fatte e spiegano molto bene il perché dei calcoli a patto di conoscere almeno un poco di elettrotecnica.

Un altro parametro da considerare per scegliere l'induttanza L è la corrente, in questo caso almeno 2A per minimizzare la resistenza ohmica dell'avvolgimento e si può trovare già avvolta. Per il condensatore C_o va considerata la tensione che deve tenere e che deve avere una bassa ESR (Equivalent Serial Resistant), così come per C_i . Entrambi comunque possono essere acquistati su RS, Mouser, Farnell o altri fornitori di elettronica.

II° CIRCUITO: TORCIA A LED

Il secondo circuito realizzato è un elevatore di tensione (step-up o boost) utile per alimentare in corrente costante tre serie di 8 led bianchi in parallelo partendo da una tensione di batteria di 4,5 Vdc; in pratica una torcia a led, con lo scopo di sfrutta-

Vbatt.	Valore R2	lin	I _{LED}	Nr. LED
2,9 V	47 R	270 mA	26 mA	24 LED 8 x 3 //
3 V	47 R	260 mA	26 mA	
4,5 V	47 R	190 mA	26 mA	
6 V	47 R	147 mA	26 mA	
7,5 V	47 R	110 mA	27 mA	
9 V	47 R	92 mA	27 mA	
3 V	23,5 R	327 mA	37 mA	
3,4 V	23,5 R	470 mA	51 mA	
4,5V	23,5 R	342 mA	52 mA	
6 V	23,5 R	267 mA	52 mA	
9 V	23,5 R	225 mA	53 mA	
9 V	23,5 R	182 mA	53 mA	
4,5 V (3 x AAA)			50 mA	Torcetta a 3 LED //
4,5 V (3 x AAA) => 3V			450 mA 50 mA	Torcia a 9 LED //

Tabella 2

re al meglio l'energia fornita dalla batteria. Il circuito, in figura 7, composto da soli 9 componenti eleva la tensione da 4,5V a circa 22V, che corrisponde alla serie di 8 led bianchi, ciascuno dei quali a circa 2,7V ($2.7 * 8 = 21,6V$). La regolazione in realtà avviene in corrente, determinata dal valore della resistenza shunt R2 sulla quale cade la tensione di feedback misurata dal comparatore (pin 5). Il valore della corrente I_{LED} segue quindi la legge di Ohm: $[I_{LED} = V_{ref} / R2]$, dove V_{ref} è 1,25V. Con il valore di R2 indicati nello schema le correnti sono rispettivamente circa: 26 mA e (52 mA).

Nella Tabella 2 sono riportate le prestazioni del circuito, valori di corrente d'ingresso e uscita al variare della tensione d'ingresso. Nello schema è indicata una batteria da 4,5V, ma come si vede dalla tabella il circuito funziona bene fino a 9V e la corrente assorbita, ovviamente, diminuisce all'aumentare della tensione di alimentazione. Quello di 3V è anche il limite minimo di funzionamento del MC34063, si è considerata la tensione di 3V come livello di batteria scarica (1V per elemento). Le ultime righe della Tabella 2 (colore rosa) riportano gli assorbimenti di due torce commerciali, da

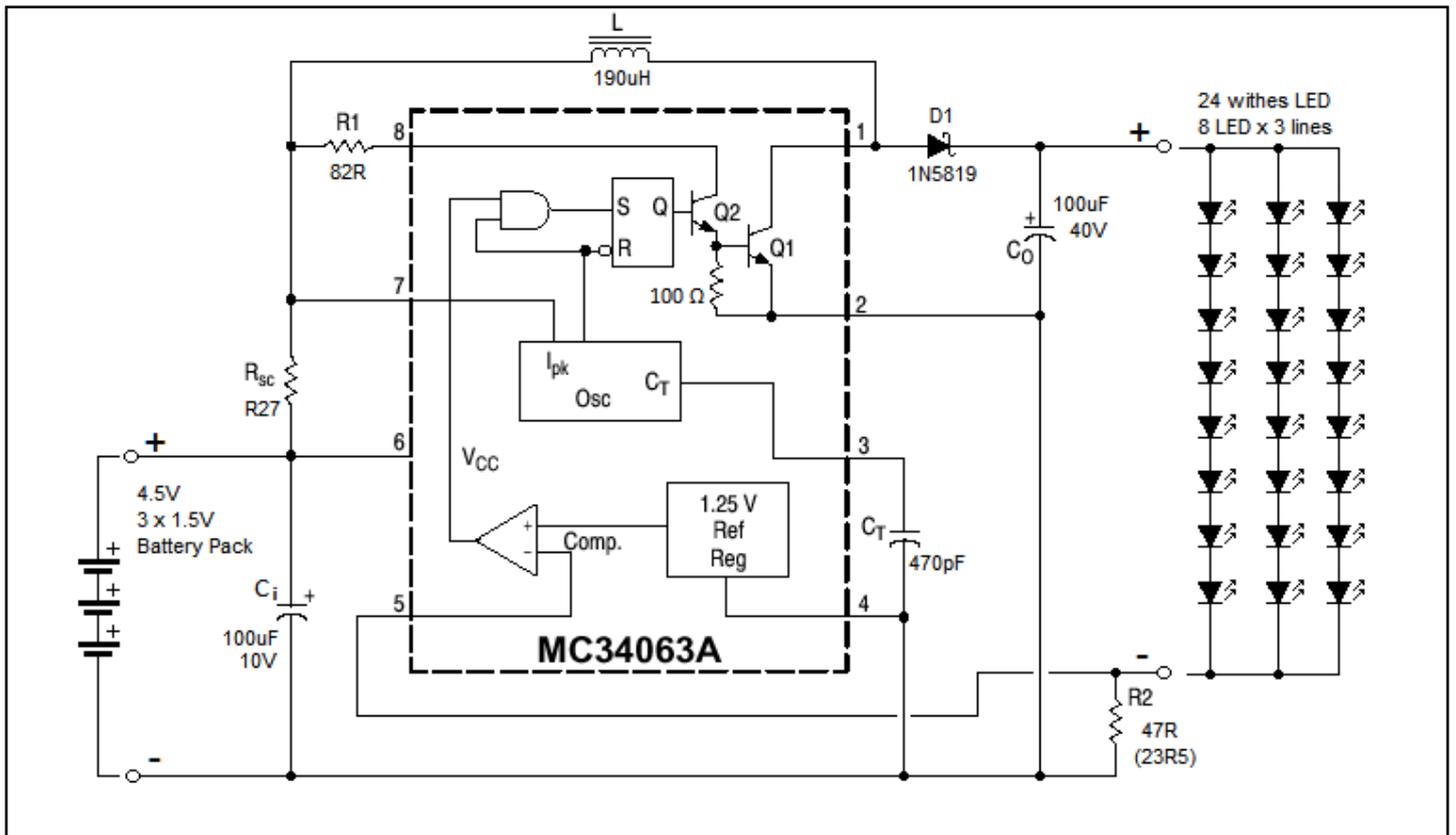


Figura 7. Schema dell'alimentatore per torcia a LED

3 e da 9 led connessi in parallelo, che utilizzano tre batterie ministilo (AAA) in serie.

Le correnti sono misurate alla tensione nominale di 4,5V a pile nuove, appena la carica delle batterie diminuisce anche la corrente / luminosità dei led diminuisce. La torcia a nove led ha un assorbimento importante, la luce è vivida con le pile nuove ma poi, pur durando abbastanza la luce diventa presto fioca. Prendendo invece i primi tre dati della tabella 2 si vede, come con questo circuito, l'assorbimento è più basso e quindi più alta la durata delle batterie pur alimentando ben 24 led, e durante la scarica la corrente e quindi la luminosità nei led rimane praticamente costante per poi scen-

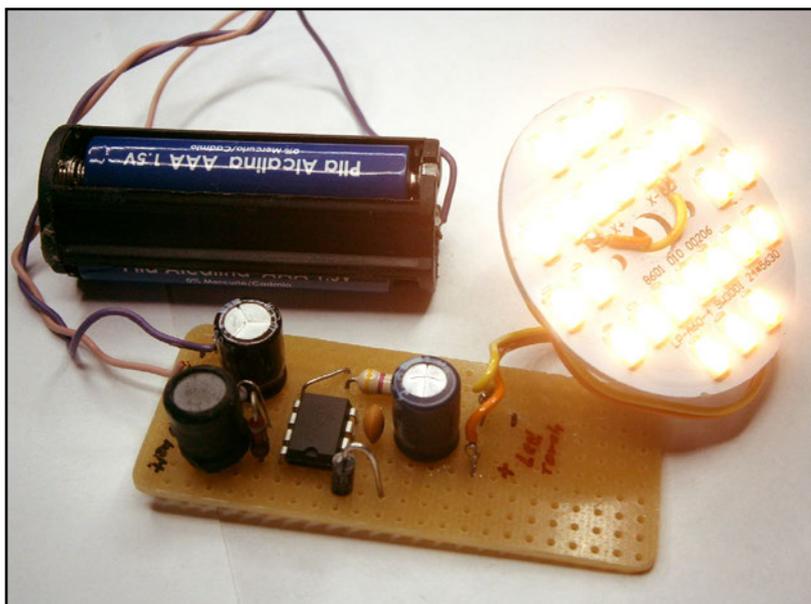


Figura 8. Alimentatore per torcia a LED assemblato

dere sotto i 3V. Questi dati confermano che il circuito è in grado di sfruttare adeguatamente quasi tutta l'energia delle pile, per meglio usufruire di questa caratteristica si dovrebbero utilizzare 3, o meglio 4 batterie alcaline (6V) in serie del tipo stilo (serie AA), mezza torcia (serie C) o torcia (serie D). Questo tipo di pile consentono un tempo di uso continuativo anche molto lungo (stima di: 8h, 20h e 40h rispettivamente).

Ci si chiederà perché si è usata una 'lampadina' formata da 24 led, tre serie da 8 led in parallelo? La risposta è molto semplice, è il pezzo interno recuperato da una lampada a led da 230V, credo intorno ai 400 lumen a luce calda, inutilizzabile perché si è rotto il vetro protettivo cadendo a terra (sigh, era nuova). Con le correnti utilizzate la 'lampadina' ha una luminosità più bassa rispetto all'originale alimentato a 230V, ma come torcia funziona egregiamente.

In figura 8 si vede il circuito montato con il pacco batterie (3 ministilo AAA) e i led accesi.

III° CIRCUITO: INTERMITTENZA PER CATENA DI LED NATALIZIA

Questo è l'ultimo circuito realizzato utilizzando i componenti recuperati dai carica-batterie, ed è dedicato al Natale.

Tempo addietro acquistai, sempre sulle solite bancarelle, due catene luminose lunghe 2 m composte di 20 led di quattro colori (RGYB) a luce fissa alimentate a 3V, mi servivano dei led a basso costo per modi-

ficare un albero natalizio grande. Me n'è rimasta una e ho pensato di utilizzarla per un alberello piccolo da scrivania, 25 cm circa di altezza, da collegare alla USB del PC o ad un alimentatore da cellulare. Provando ad alimentare la catena di led con l'alimentatore da banco ho fatto una sorprendente scoperta, tipo acqua calda, essendo i led di colori diversi hanno soglie di accensione diverse. I Rossi e i Gialli si accendono già a circa 2V mentre i Verdi e i Blu si accendono a circa 3V, infatti i led rossi e gialli hanno (come si vede nello schema) un resistore di limitazione in serie da 68 Ohm.

Da qui è nata l'idea di una 'intermittenza' natalizia come quelle di una volta, a tre livelli: tutti spenti, lampeggio [Rossi / Gialli] e [Verdi / Blu] spenti, [Rossi / Gialli] accesi e lampeggio [Verdi / Blu]. I tempi di accensione / spegnimento si sovrappongono senza un particolare sincronismo, i lampeggi brevi sono da circa 2,5 s quelli lunghi da circa 13 s. Il circuito (Figura 9) è basato sul regolatore step-down, i tempi sono ottenuti con dei semplici oscillatori astabili realizzati con un chip di inverter CD40106 CMOS, con ingressi triggerati, alimentati dalla tensione d'ingresso (5V- 10V).

Le uscite degli oscillatori pilotano dei transistor che con dei resistori da 1k alterano, quando sono in conduzione, il partitore di feedback del regolatore. A transistor interdetti la tensione di uscita è di 1,7 V circa, con un transistor in saturazione passa a 2,4 V e con entrambi i transistor chiusi

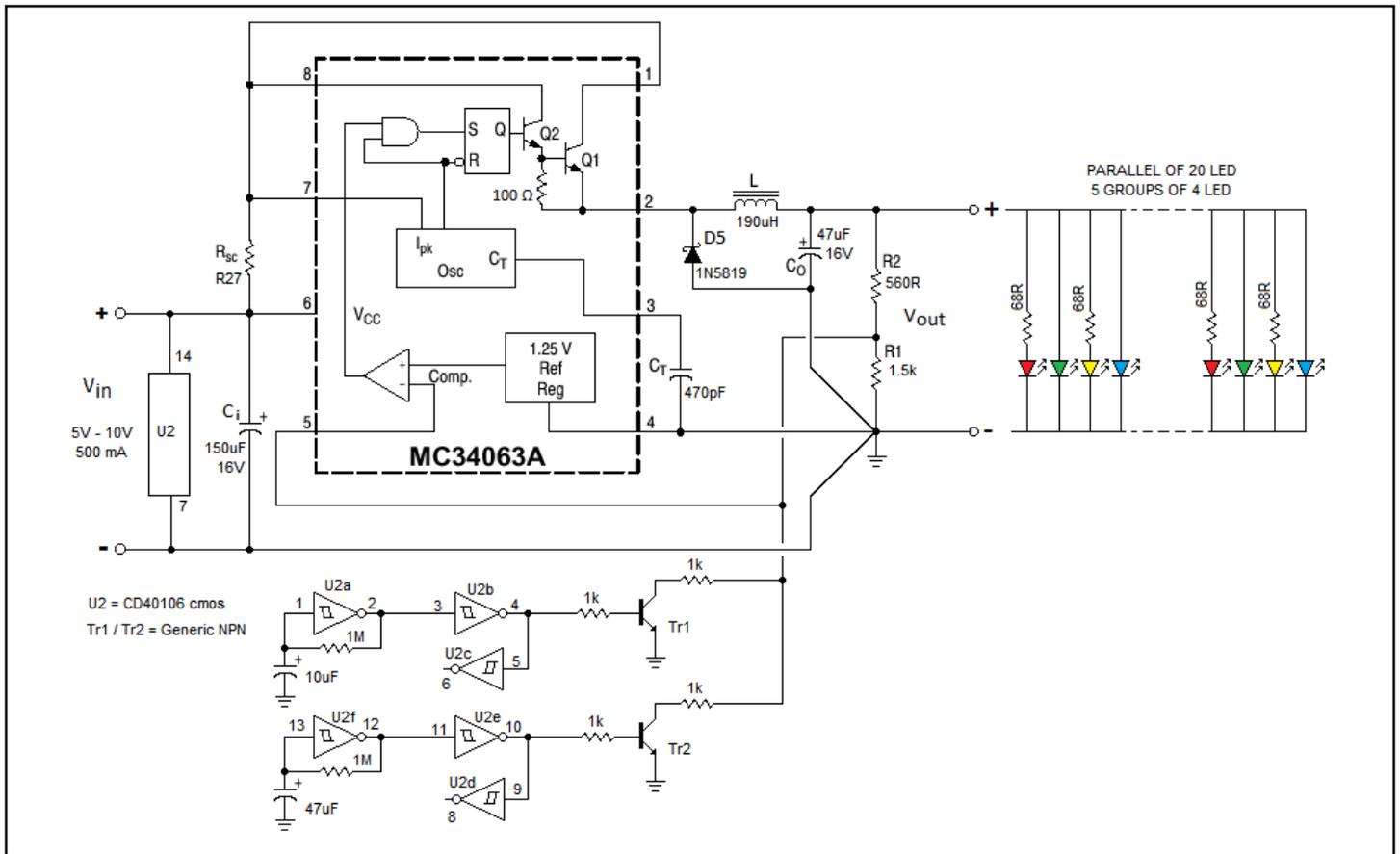


Figura 9. Schema dell'intermittenza natalizia

raggiunge 3,1 V. Nell'immagine di apertura di questo articolo si vede l'alberello da scrivania in funzione collegato alla USB

del PC, mentre in Figura 10 la catena di led multicolori utilizzata.

Nota: il condensatore d'ingresso Ci deve essere almeno da 100µF per evitare sfarfallamenti della tensione di alimentazione.

CONCLUSIONE

Bene! Giunti al termine, che dire?

Buon divertimento e... non dimenticate di sfogliare la raccolta dei datasheet e delle note applicative che potete scaricare dal sito di E&M cliccando qui.



Figura 10. Catena di led multicolori



**GUARDA SUL SITO
GLI ALTRI TITOLI!**

**NOVITA'
TUTTI I MESI**

OLTRE 500 LIBRI A CATALOGO

Pier Calderan
Capire e Usare LoRa e LoRaWAN
CREARE UNA RETE LoRaWAN PER IOT

Progetto Pratico di un dispositivo LoRa e LoRaWAN con Arduino, ATmega328 e Raspberry Pi

libri SANDIT

Cod. 5597 - Pag.236 - € 17,90
ISBN 9788869283239
Capire e Usare LoRa e LoRaWAN

MAKER BOOK
Con CD
Bruno Bertucci
DOMOTICA LOW COST con ARDUINO

Guida pratica alla realizzazione di impianti domotici con Arduino controllati da touch screen Weintek e da smartphone

NEI CD-ROM SPERIMENTARE PER ANCHE SENZA PANNELLO TOUCH-SCREEN

libri SANDIT

Cod. 5541 - Pag.148 - € 15,50
ISBN 9788869282454
Domotica Low Cost con Arduino

GIUSEPPE ZELLA
ARDUINO & MATRICI LED
Sperimentazione, Grafica, Didattica e Alfabetizzazione con Arduino e Matrici LED

libri SANDIT

Cod. 5575 - Pag.178 - € 14,90
ISBN 9788869282942
Arduino & Matrici LED

A Colori!
STEFANO ANTONIAZZI
GUIDA INNOVATIVA AL LINGUAGGIO C
PER SISTEMI ARDUINO E MAC/PC

libri SANDIT

Cod. 5456 - Pag.254 - € 23,50
ISBN 9788869281211
Guida innovativa al linguaggio C

A Colori!
ESP8266
LA SCHEDA E LE APPLICAZIONI

IL CONTROLLO A DISTANZA VIA INTERNET

- SENSORI ED ATTUATORI SEMPLI
- PROGETTI FACILI PER LA DOMOTICA
- IOT: INTERNET DELLE COSE

PROGRAMMAZIONE TRAMITE IDE ARDUINO
ESPERIMENTI DI UTILIZZO ED INTERFACCIA

I FASCICOLI DEL MAKER

libri SANDIT
PAOLO CAPOBUSSI

Cod. 5497 - Pag.64 - € 9,90
ISBN 9788869281655
ESP8266

A Colori!
NODEMCU
LE SCHEDE E LE APPLICAZIONI

PROGRAMMAZIONE FACILE COME UN ARDUINO

- DOMOTICA A PORTATA DI MANO
- PROGETTI PER TANTE SOLUZIONI
- SCOPRIRE LA SCHEDA GIUSTA

USARE L'IDE DI ARDUINO
ESEMP. SKETCH E MOLTO ALTRO
- IOT: INTERNET DELLE COSE

I FASCICOLI DEL MAKER

libri SANDIT
PAOLO CAPOBUSSI

Cod. 5569 - Pag.80 - € 9,90
ISBN 9788869282843
NodeMCU

Paolo Di Leo
101 esercizi guidati di elettronica
Componenti discreti

libri SANDIT

Cod. 5560 - Pag.200 - € 13,90
ISBN 9788869282720
101 esercizi guidati di elettronica

Davide Scullino
Rework SMD e BGA Le tecniche

libri SANDIT

Cod. 5581 - Pag.122 - € 12,90
ISBN 9788869283024
Rework SMD e BGA Le tecniche

MAKER BOOK
Marco Buttollo
ROBOTICA Principi e Applicazioni

3 VOLUMI IN UNO
Principi di Robotica - Robotica Industriale - Robotica Umanoidale

libri SANDIT

Cod. 5475 - Pag.710 - € 29,90
ISBN 9788869281419
ROBOTICA Principi e Applicazioni

Con CD
Bruno Bertucci
PLC IN PRATICA

Guida alla programmazione dei controllori di ultima generazione per Industry 4.0 e IoT con sistema di sviluppo e simulatore integrato

versione aggiornata

libri SANDIT
CD ALLEGATO

Cod. 5429 - Pag.180 - € 17,50
ISBN 9788869280153
PLC in Pratica

Con CD
Cosimo Morello
PIC MCU
MICROCONTROLLORI PIC E PICBASICPRO

Utilizzare la potenza del PIC e usare i sensori e accessori pensati per Arduino con PICBASIC PRO PBP3

Teoria e Pratica

libri SANDIT
CD ALLEGATO

Cod. 5591 - Pag.468 - € 24,90
ISBN 9788869283147
PIC MCU: Utilizzare i PIC e sensori con Arduino

MAKER BOOK
Paolo Di Leo
Guida d'uso, shell e programmazione C di **Raspberry Pi**

Da Raspberry Pi Zero a Raspberry Pi 3 passando per Raspberry 1 Model B+ e Raspberry Pi 2

libri SANDIT

Cod. 5530 - Pag.310 - € 16,50
ISBN 9788869282348
Raspberry Pi - Guida d'uso, shell e programmazione in C



Consigli Utili per Disegnare uno Schema

di Marco Castiglioni

info@masterista.it

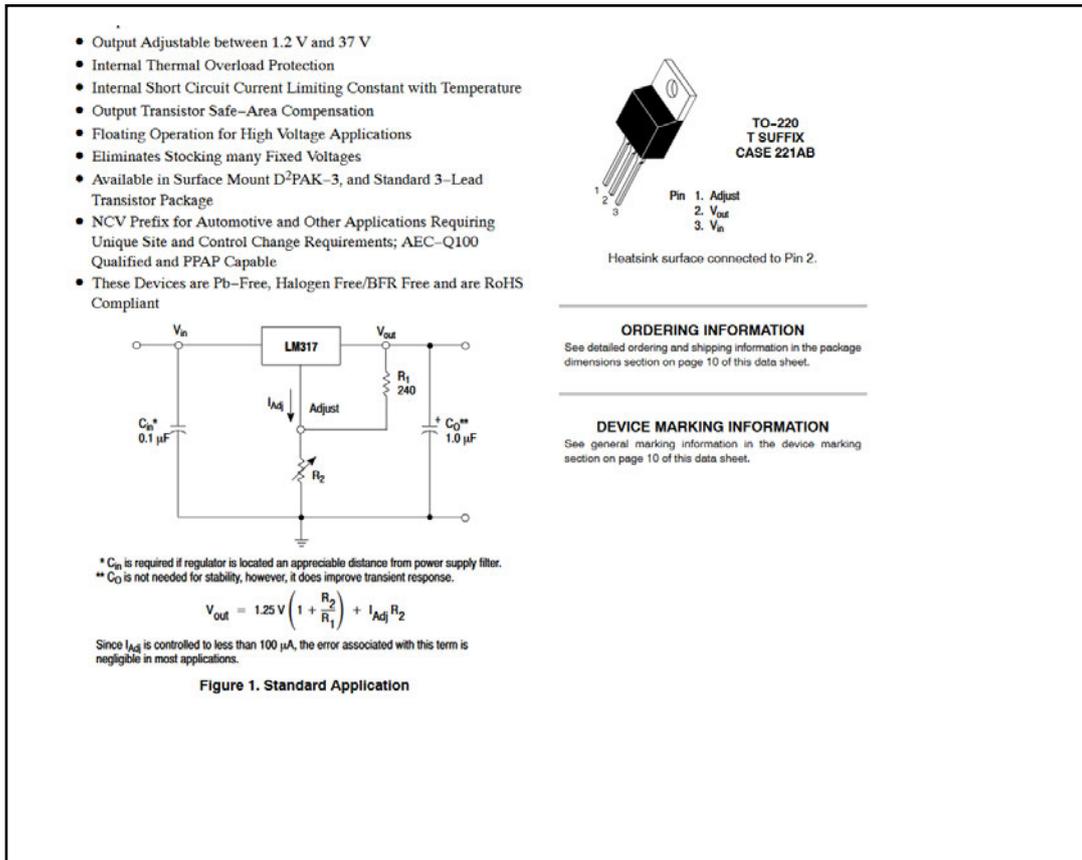
In qualità di Masterista Professionale, nel corso degli anni ho potuto vedere ogni sorta di schematici, realizzati a volte anche in maniera molto fantasiosa. Con questi articoli, vorrei spiegare e consigliare come deve essere disegnato uno schema elettrico in modo univoco: sia per chi lo disegna, sia per chi lo deve leggere o utilizzare.

Che sia per Hobby o per lavoro, sarà capitato a volte di dover disegnare degli schemi elettrici utilizzando dei software più o meno professionali. Tali Cad permettono di realizzare quella sequenza di lavorazioni che terminano con la campionatura di un circuito stampato in versione "casalinga" oppure "professionale".

I SOFTWARE DISPONIBILI

Esistono in commercio software gratuiti oppure a pagamento, con prezzi e funzioni decisamente diversi tra loro, e molti di questi, hanno già internamente delle Librerie a cui attingere componenti. Ma c'è una cosa che li accumuna tutti, e sono quelle che io chiamo "Le Regole di Schematico".

In questo primo articolo disegneremo un semplice alimentatore con LM317 (Figura 1) utilizzando [KiCad](#), un programma gratuito di facile utilizzo con una Libreria com-



la prima volta che lo utilizziamo, verificiamo se i pin sono coerenti con il datasheet fornito dalla Casa produttrice:

- Pin 1 = ADJ
- Pin 2 = Vout
- Pin3 = Vin

Sopra l' LM317 aggiungiamo anche qualche informazione utile per l' utilizzo, come la formula da utilizzare per il calcolo delle

Figura 1: Estratto dal foglio specifiche dell'LM317

ponenti molto fornita. Utilizzeremo come base di partenza lo "Standard Application".

Per Prima cosa apriamo un nuovo progetto, che chiameremo "Power Supply" (Figura 2). Apriamo la pagina di schematico appena creata (Figura 3) e posizioniamo l' LM317 in versione TO220 al centro (Figura 4), e dato che e'

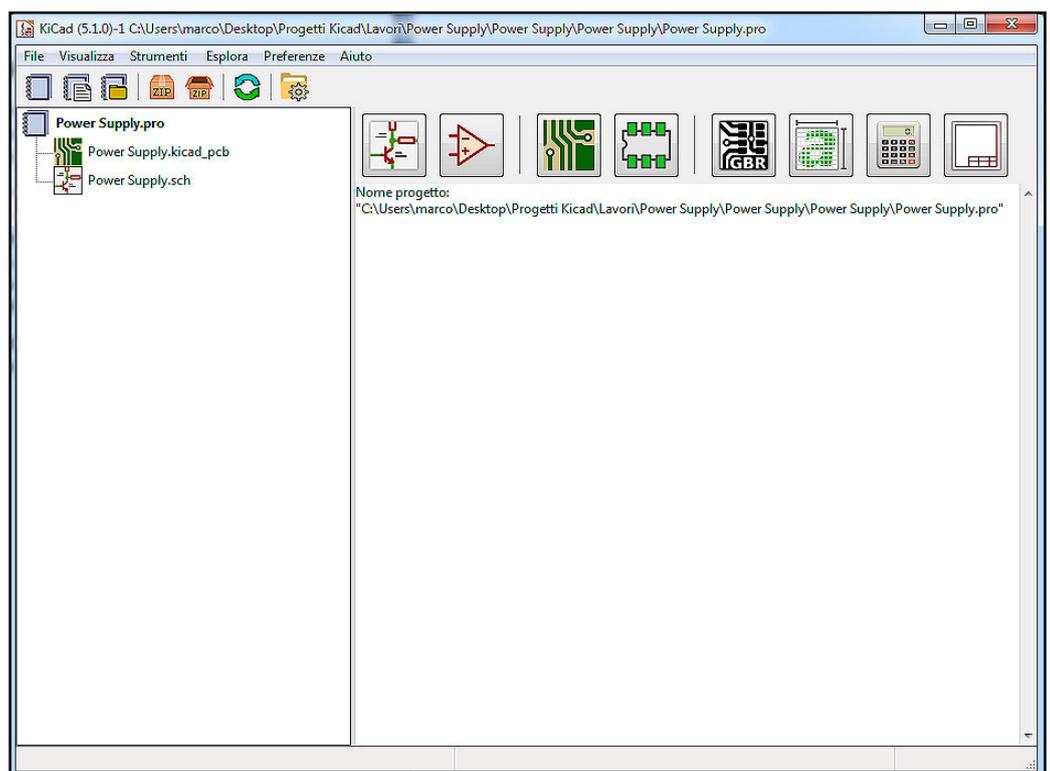


Figura 2: Apertura del nuovo progetto con KiCad

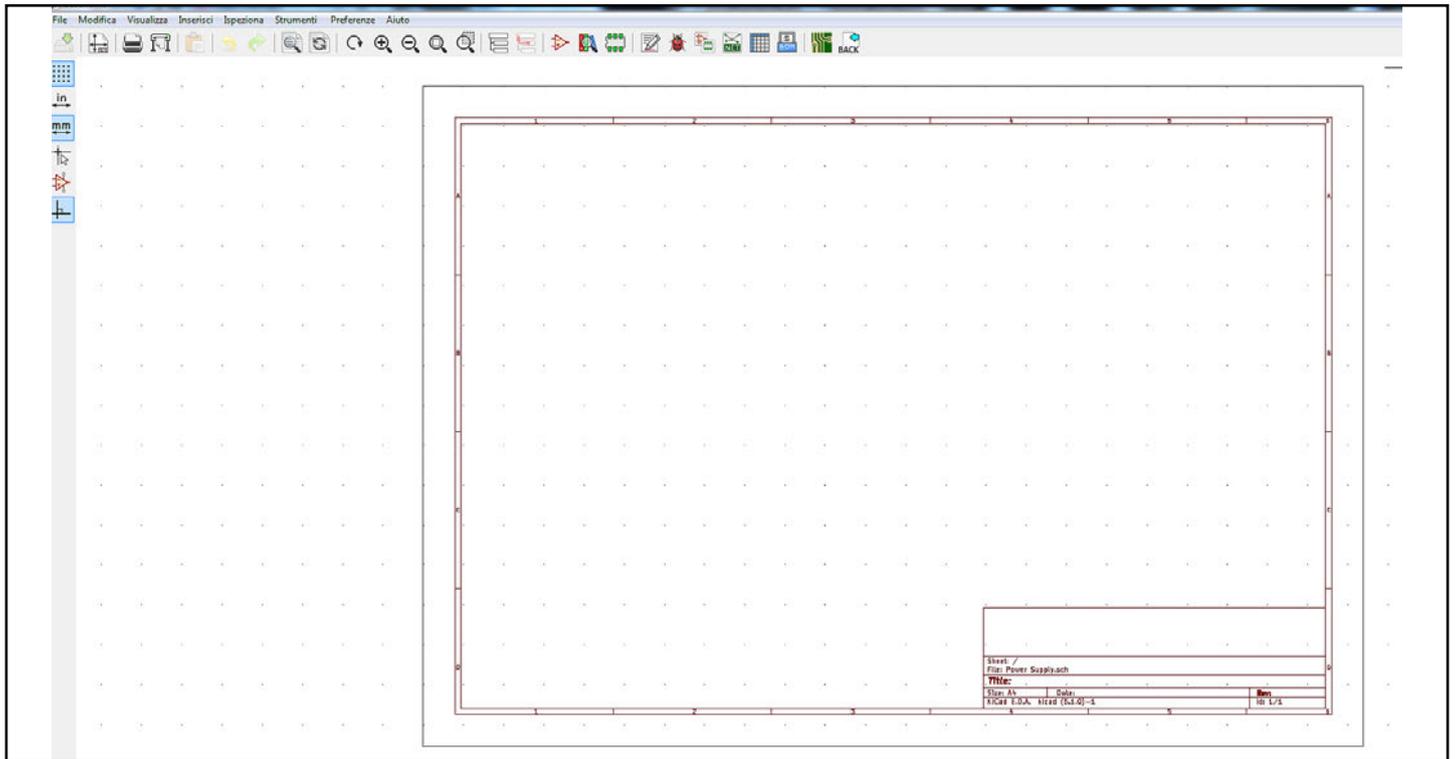


Figura 3: Pagina iniziale del progetto appena creato

resistenze ed il dato della V_{ref} (tensione di riferimento).

Inseriamo ora la R1 del valore consigliato di 240 ohm, valore difficile da trovare, quindi passiamo alla 220 ohm.

Ora è il turno della R2, resistenza variabile da 5K, e valutiamo di utilizzare un Trimmer Multigiri con regolazione verticale, per una maggiore precisione nella taratura della tensione in uscita. Andiamo a vedere nella Libreria dei simboli schematici e troviamo due tipi diversi di Trimmer (Figura 5).

Il simbolo di sinistra è realizzato con 3 pin, mentre quello di destra ha 2 pin, e per decidere quale utilizzare dobbiamo confrontarlo con il componente "reale" che corrisponde ad un trimmer Bourns serie 3296W (Figura 6) con piedinatura evidenziata nel

datasheet. La prima specifica che notiamo è che questo componente ha 3 pin, di conseguenza eliminiamo quello da 2. La seconda cosa che dobbiamo verificare è

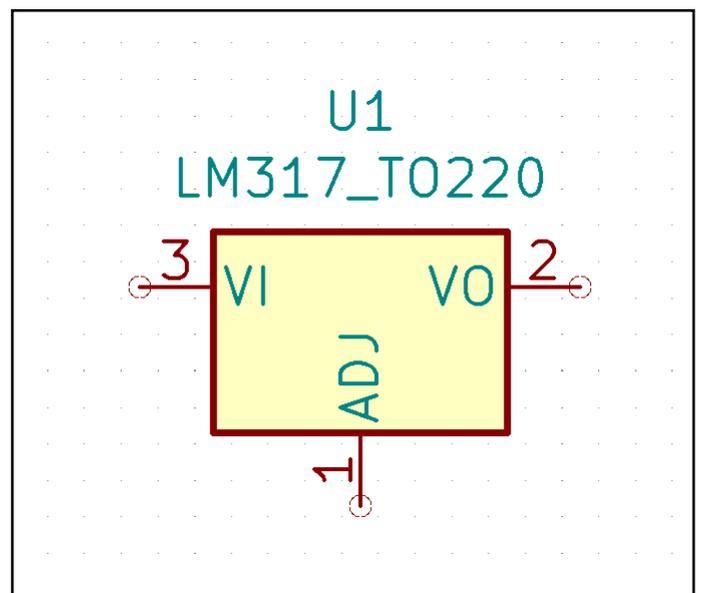


Figura 4: Simbolo dell'LM317-T0220 posizionato al centro della pagina di lavoro.

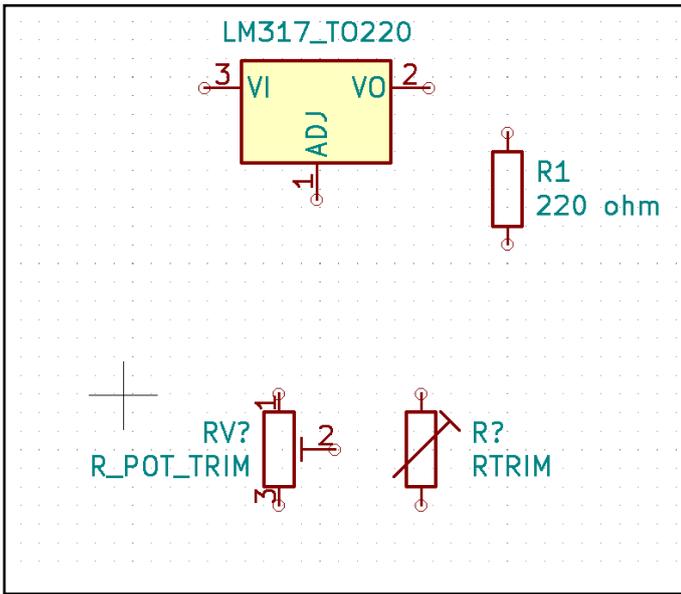


Figura 5: La libreria presenta due tipi di trimmer

il numero dei pin, sia del simbolo, sia del componente, e corrisponde a:

- Pin esterno = 1
- Pin centrale o di regolazione = 2
- altro Pin esterno = 3

Altra informazione che ricaviamo dal data-sheet è la specifica del senso di rotazione del trimmer :

"Se ruotiamo in senso orario (CLOCKWISE) la vite di regolazione, avviciniamo il pin (2) centrale al pin (3) esterno ".

Ci interessa saperlo perché vogliamo che quando ruotiamo in senso orario, la tensione in uscita "aumenti" e quando inve-



BOURNS®

*ROHS COMPLIANT VERSIONS AVAILABLE

Features

- Multiturn / Cermet / Industrial / Sealed
- 5 terminal styles
- Tape and reel packaging available
- Chevron seal design
- Listed on the QPL for style RJ24 per MIL-R-22097 and RJR24 per High-Rel Mil-R-39035

- Mounting hardware available (H-117P)
- RoHS compliant* version available
- For trimmer applications/processing guidelines, [click here](#)

3296 - 3/8" Square Trimpot® Trimming Potentiometer

Electrical Characteristics

Standard Resistance Range 10 ohms to 2 megohms (see standard resistance table)

Resistance Tolerance ±10 % std. (tighter tolerance available)

Absolute Minimum Resistance 1 % or 2 ohms max. (whichever is greater)

Contact Resistance Variation 1.0 % or 3 ohms max. (whichever is greater)

Adjustability

 Voltage ±0.01 %

 Resistance ±0.05 %

Resolution Infinite

Insulation Resistance 500 vdc. 1,000 megohms min.

Environmental Characteristics

Power Rating (300 volts max.)

 70 °C 0.5 watt

 125 °C 0 watt

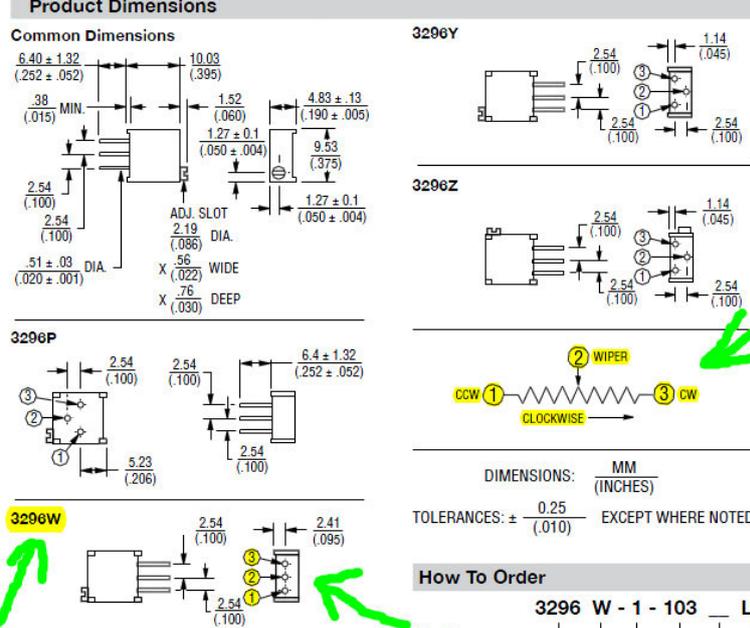
Temperature Range ... -55 °C to +125 °C

Temperature Coefficient ... ±100 ppm/°C

Seal Test 85 °C Fluorinert†

Product Dimensions

Common Dimensions



3296Y

3296Z

3296P

3296W

CCW ① ← WIPER → ③ CW

CLOCKWISE →

DIMENSIONS: MM (INCHES)

TOLERANCES: ± 0.25 (0.010) EXCEPT WHERE NOTED

How To Order

3296 W - 1 - 103 _ LF

Model _____

Figura 6: Identificazione delle specifiche di rilievo sul data-sheet del trimmer Bourns 3296W.

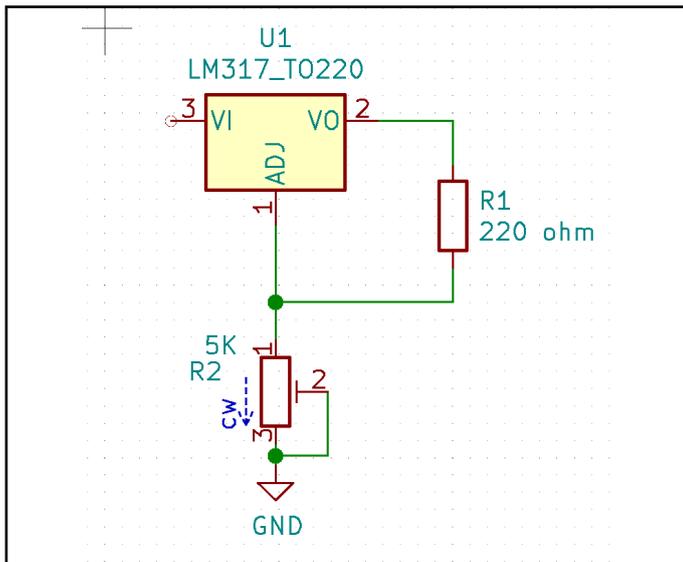


Figura 7: Collegamento a terra del Pin 2

ce ruotiamo in senso antiorario la tensione "diminuisca". Utilizzando la formula dell' LM317 risulta che:

- Trimmer a "zero ohm" = tensione in uscita "MIN"
- Trimmer a "5K ohm" = tensione in uscita "MAX"

Collegiamo il pin 2 a Massa (Figura 7) ed otteniamo:

- rotazione della vite in senso orario = pin 2 verso pin 3 = 5K = incremento della tensione in uscita
- rotazione della vite in senso antiorario = pin 2 verso pin 1 = 0 ohm = decremento della tensione in uscita

Aggiungiamo un condensatore elettrolitico da 470uF 50V in ingresso ed uno da 100uF 50V in uscita.

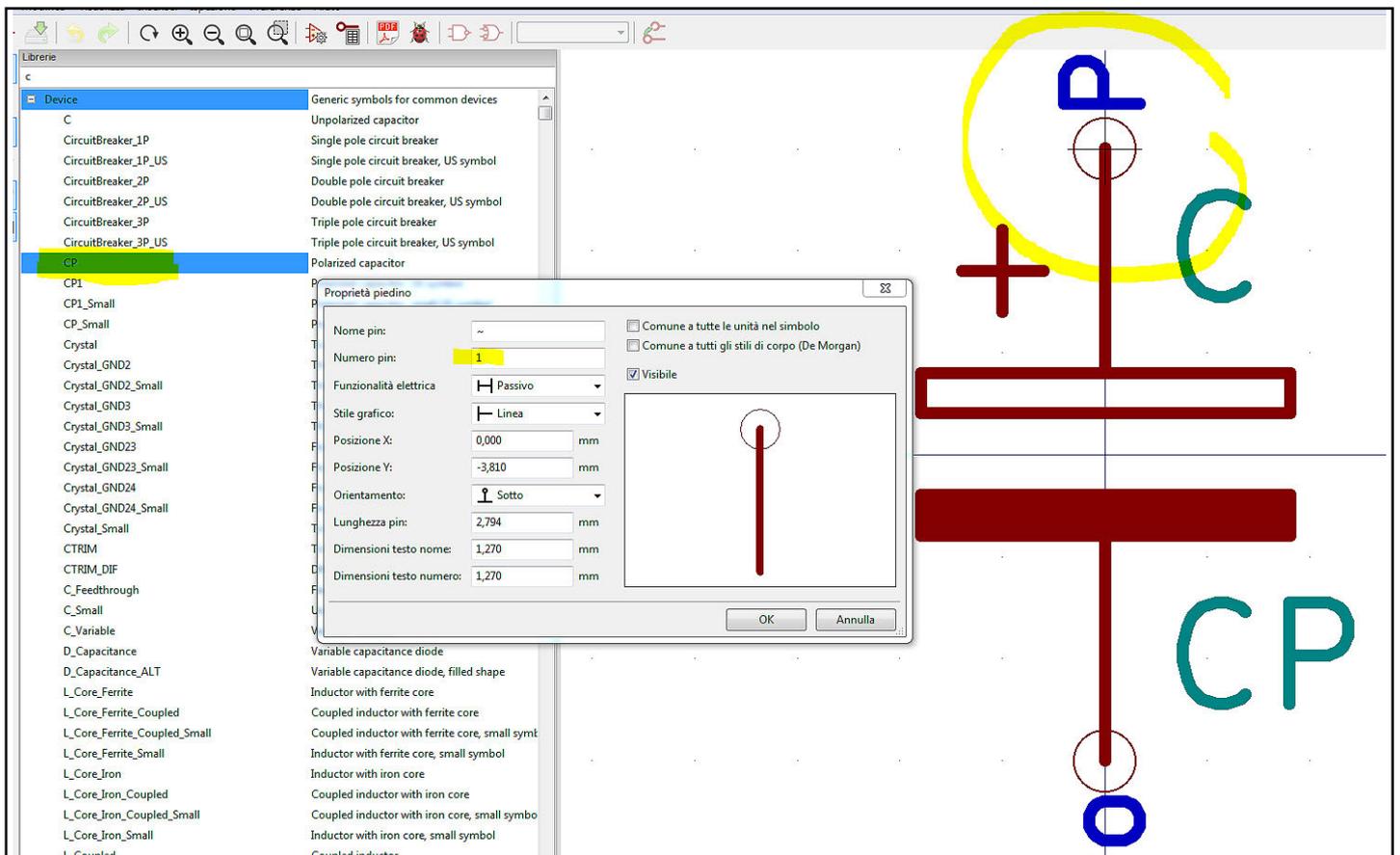


Figura 8: Verifica della numerazione dei Pin del condensatore

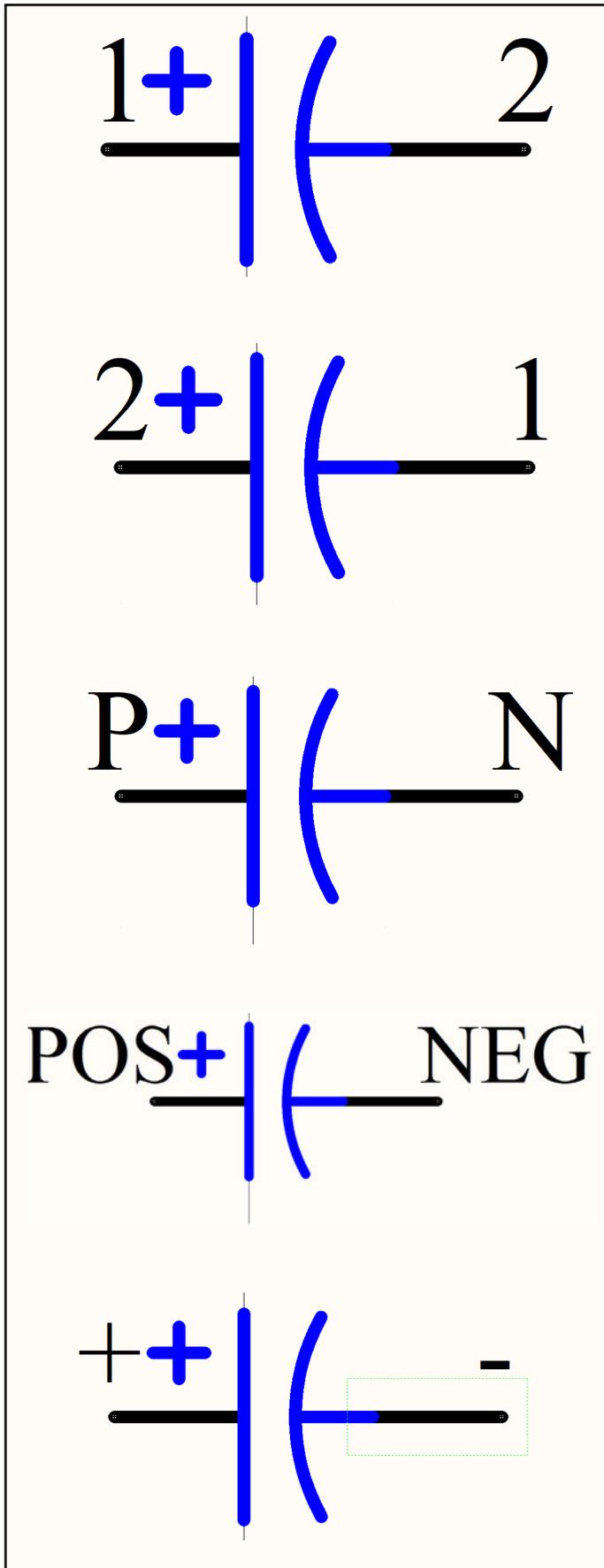


Figure 9-13: Diversi tipi di assegnazione

NOTE SUI CONDENSATORI

ELETTROLITICI:

La prima volta che utilizziamo questo simbolo, dobbiamo verificare in Libreria le assegnazioni dei pin, perché' nel caso di realizzazione di un nuovo Condensatore polarizzato con diametro diverso da quelli già esistenti, questa informazione ci permetterà di stabilire quale sarà il pin Positivo e quello Negativo. Il nostro ha la seguente numerazione:

- Pin Positivo = 1
- Pin Negativo = 2 (Figura 8)

Questa osservazione potrebbe sembrare banale, purtroppo nel mio mestiere ho potuto vedere ogni tipo di assegnazione sui condensatori elettrolitici (Figura 9,10,11,12,13). Nel caso in cui, nel nostro schematico, ci siano diversi tipi di simbologia, quando inseriamo il package o footprint PCB, ci troveremo dei montaggi errati. Normalmente queste “disattenzioni” avvengono quando si effettuano dei “copia/incolla” da altri schematici, oppure usando le librerie generiche fornite dai vari CAD.

UN ESEMPIO

Se scarichiamo un componente da www.snapeda.com (sito dove e' possibile scaricare gratuitamente simboli schematici, PCB e 3D) perché' mancante nella nostralibreria (Figure 14 e 15), bisogna sempre controllare il numero associato ai pad per essere sicuri che, una volta aggiunto nello schematico, il pin positivo sia uguale al pad positivo ed il

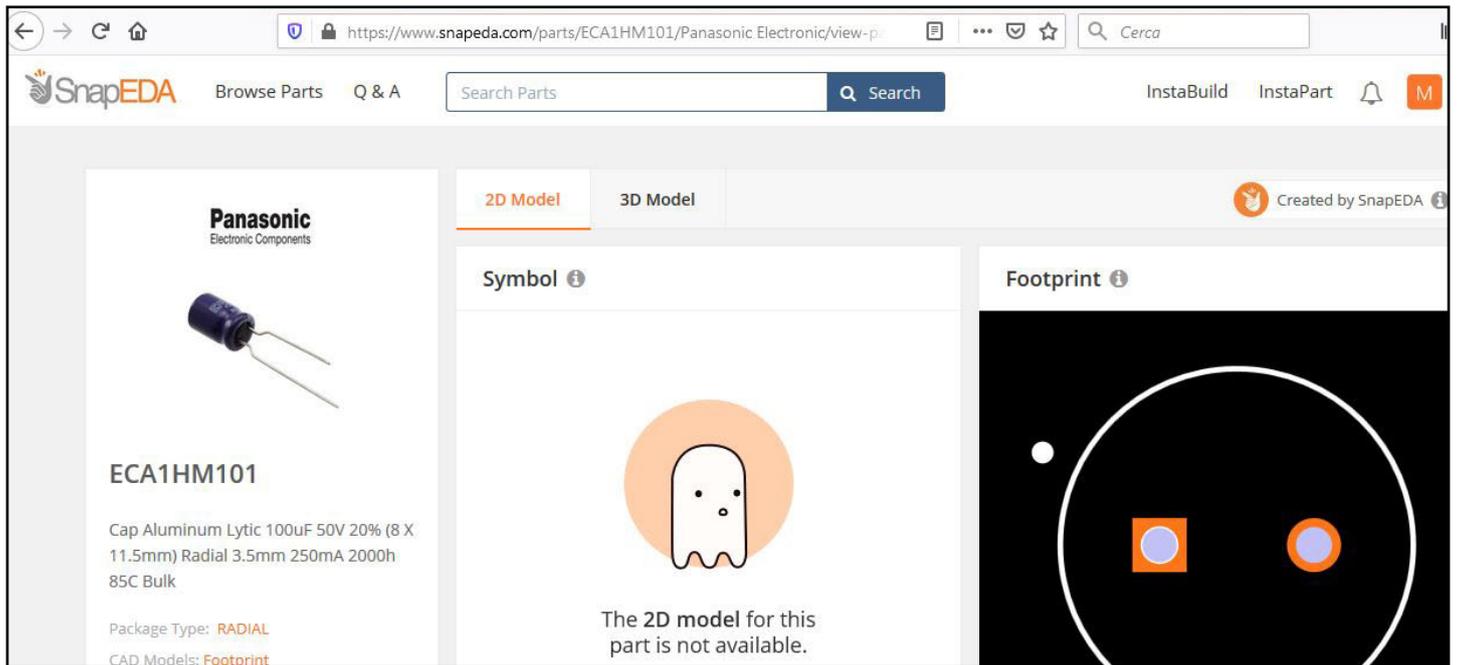


Figura 14: Condensatore trovato sul sito di SnapEDA

pin negativo al pad negativo

Altra indicazione per i condensatori polarizzati e' quella di inserire sempre la tensione di lavoro, affinché sia montato quello giusto e non corriamo il rischio di vederlo esplodere

perché' di tensione di lavoro troppo bassa o al limite delle specifiche.

Aggiungiamo un condensatore ceramico da 100KpF in ingresso ed uno identico in uscita, per gli eventuali disturbi sull' alimen-

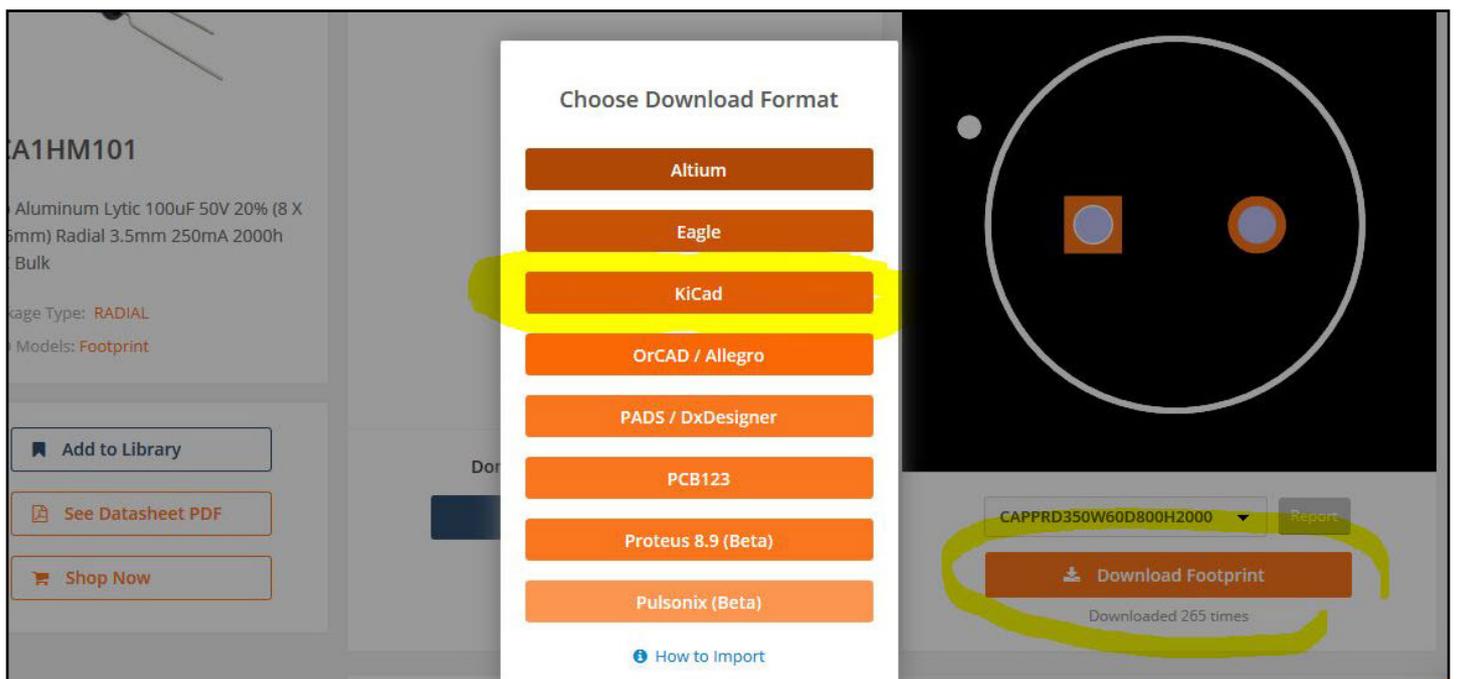


Figura 15: Selezione del formato CAD e download del footprint

tazione. (Figura 16). Con i condensatori Ceramici o Poliestere, non vi sono problemi perché non hanno un "verso di montaggio". Ora non rimane che inserire due connettori di alimentazione, uno per i cavetti di Ingresso ed uno di Uscita. Scegliamo di usare dei classici Morsetti a Vite (Figura 17), però inseriamo una specifica:

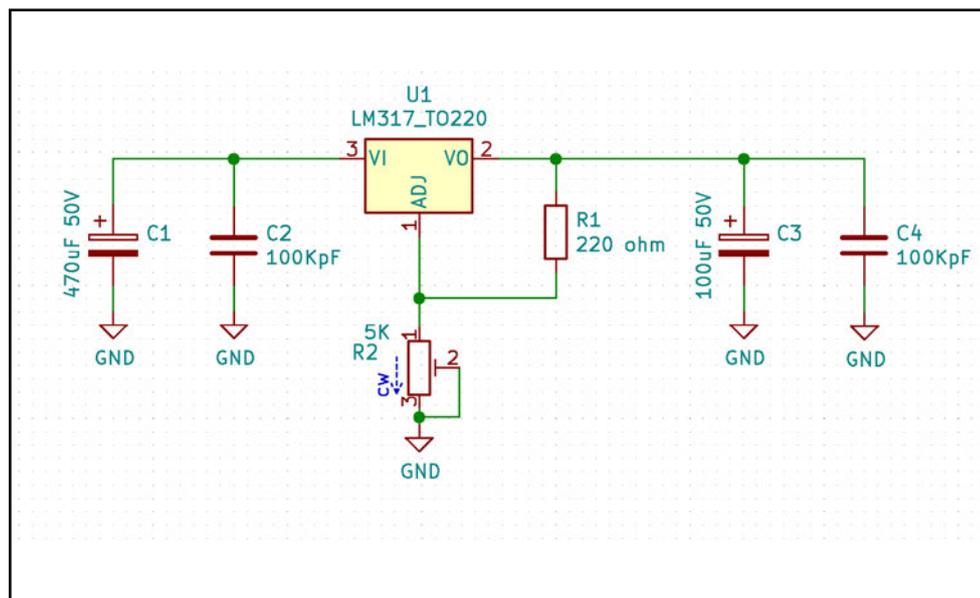


Figura 16: Inserimento dei condensatori ceramici antidisturbo



Figura 17: Morsetto a vite

" il Pin di Sinistra, quello che identificheremo come PAD numero 1, sarà sempre il Positivo e di conseguenza i PAD numero 2 saranno sempre Negativi (GND). Tenendo conto anche di questo , andiamo ad inserire due connettori nel nostro schematico (Figura 18). A questo punto il no-

stro schematico e' pronto per l' assegnazione dei simboli PCB (Footprint) e per lo sbroglio delle piste.

Nel prossimo articolo , implementeremo il nostro alimentatore aggiungendo altri componenti e funzioni , valutando i vari aspetti e specifiche , sempre riferiti a problematiche di schematico.....

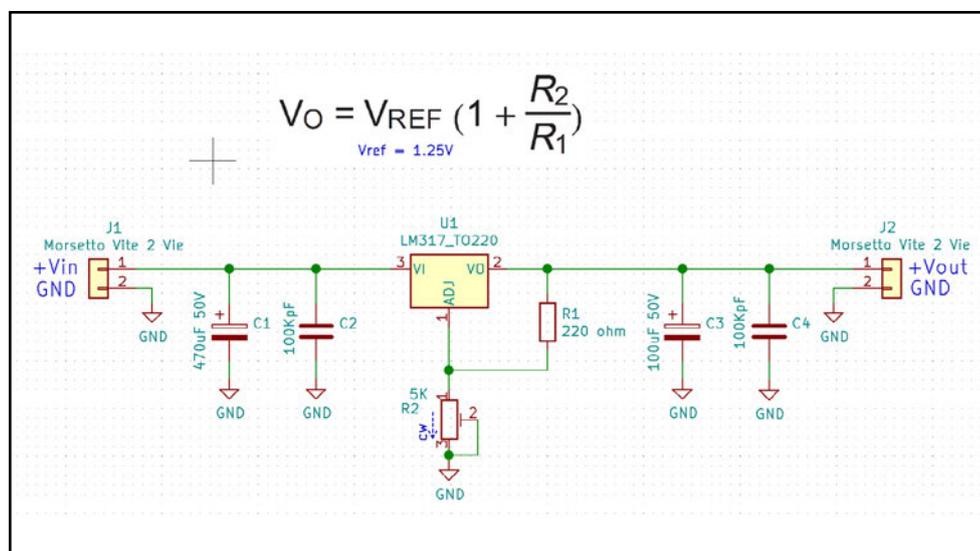


Figura 18: Inserimento dei due conettori sullo schema.



Come Realizzare una Catena di Segnale Analogica

di Mark Patrick
Mouser Electronics

Il termine "Digitale" ricorre sempre più spesso nel linguaggio della comunicazione di tutti i giorni. A volte viene anche utilizzato a sproposito, quasi a voler enfatizzare le caratteristiche di un prodotto commerciale, come se la componente analogica avesse una valenza minore. In realtà, come si legge in questo articolo, il segnale analogico risulta sempre di fondamentale importanza nei processi di acquisizione.

Il mondo in cui viviamo è prevalentemente digitale e, molto verosimilmente, lo sarà sempre più nel prossimo futuro vista la propensione a controllare in maniera automatica un gran numero di aspetti della nostra vita quotidiana. In passato, la maggior parte dei sistemi di controllo era di natura prevalentemente analogica. I moderni sistemi di controllo si basano su microcontrollori a basso costo e facilmente reperibili, che sono in grado di fornire funzioni di controllo a livello di sistema flessibili e aggiornabili in modo molto semplice.

A questo punto si potrebbe ipotizzare che la progettazione analogica abbia esaurito il suo compito. La realtà è alquanto differente in quanto si può a ragione affermare l'opposto, in quanto i sistemi di controllo utilizzano ingressi provenienti dai sensori che quasi sempre

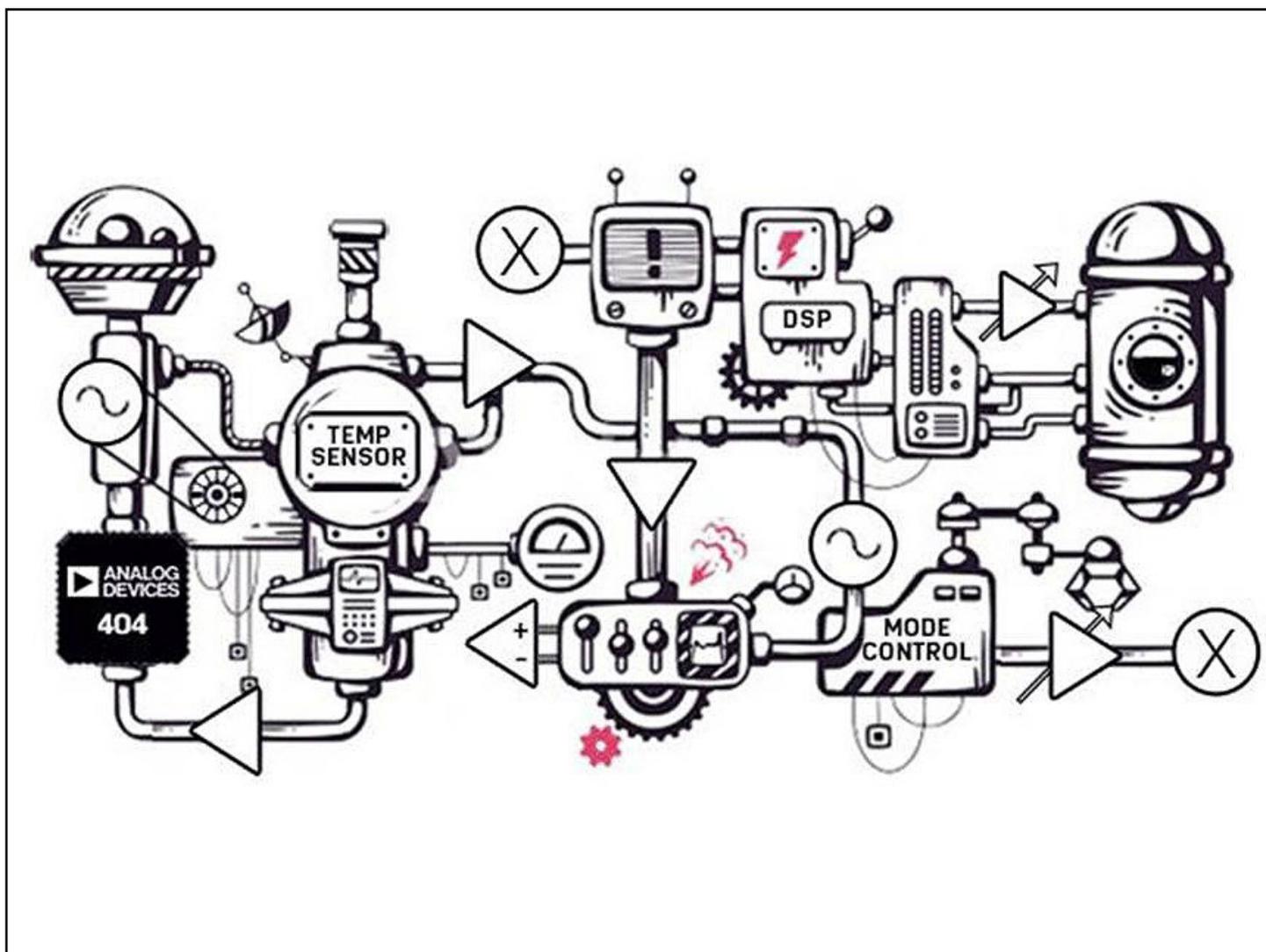
sono di tipo analogico.

Anche se esistono sensori digitali, essi sono solitamente sottosistemi che integrano un sensore analogico, circuiti per il condizionamento dei segnali e un convertitore A/D (ADC) per cui possono essere considerati dispositivi basati fondamentalmente sull'analogica. Di conseguenza, la possibilità di elaborare in maniera adeguata un segnale analogico a bassa frequenza e di trasferirlo a un ADC che lo converta nel dominio digitale per la successiva elaborazione è un requisi-

to chiave per molti progettisti, in particolare per quelli impegnati nello sviluppo di sistemi basati su sensori come ad esempio i nodi per le applicazioni IoT (Internet of Things).

LA CATENA DEL SEGNALE ANALOGICA

Una catena del segnale (signal chain) analogica prevede solitamente una sorgente del segnale (spesso sotto forma di sensore), oltre a circuiti di buffering (memorizzazione temporanea), filtraggio,



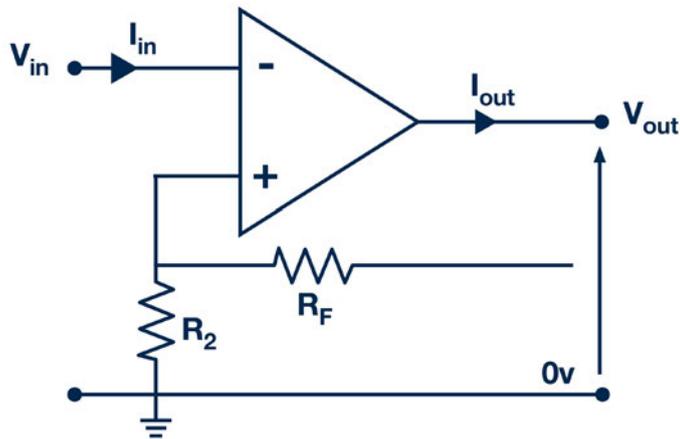


Figura 1: Schema di un amplificatore non invertente basato su un amplificatore operazionale

amplificazione e, infine, un convertitore A/D. Ciascun elemento attivo, unitamente ai relativi componenti passivi, riveste un ruolo importante nell'assicurare il mantenimento del segnale prima che questo venga convertito fedelmente nel dominio digitale.

Nella fase iniziale del processo di progettazione è necessario comprendere la natura del segnale sorgente in termini di ampiezza e di sensibilità al rumore. La maggior parte delle uscite dei sensori è rappresentata da una tensione continua (DC) proporzionale al parametro che viene misurato, che può essere luce, temperatura, pressione o un altro tra gli innumerevoli parametri fisici. Molto spesso i sensori producono un segnale relativamente debole che può risultare

influenzato dal rumore, con conseguenze negative sulla misura.

Per questo motivo è necessario predisporre un circuito follower (inseguitore), come ad esempio un semplice amplificatore non invertente basato su un amplificatore operazionale, il più vicino possibile al sensore. La prossimità è importante perché piste lunghe agiranno alla stregua di antenne e quindi "raccolglieranno" il rumore ambiente indesiderato. Nel caso non fosse possibile evitare il ricorso a piste lunghe, richieste in alcune applicazioni, è quasi sempre necessario prevedere un'adeguata schermatura.

Il guadagno di un circuito di questo tipo è determinato dal rapporto tra i resistori per cui qualsiasi differenza tra i valori di resistenza avrà sicuramente un impatto

sul guadagno del circuito e sull'ampiezza del segnale che viene trasferito lungo la catena del segnale. In alcuni casi ciò non rappresenta un problema, ma nel caso lo fosse è necessario specificare resistori con tolleranze molto strette.

Laddove fosse richiesto un elevato grado di precisione, sono disponibili resistori adattati come ad esempio il partitore resistivo [MAX5491LC](#) di Maxim.

Esso è composto da due resistori ospitati in package SOT23 adattati in maniera estremamente precisa (0,035%) e caratterizzati da una deriva di soli 2 ppm/°C sull'intero intervallo di temperatura operativa compreso tra -40 e +85 °C.

Il semplice circuito di un amplificatore

operazionale riportato in Figura 1 viene utilizzato come buffer ideale, in special modo per il pilotaggio di segnali proveniente da sensori remoti lungo linee di segnale di una certa lunghezza. Nonostante la semplicità di configurazione e di utilizzo, nel caso degli amplificatori operazionali è necessario prendere in considerazione vari aspetti relativi alle prestazioni al fine di ottenere gli elevati livelli di precisione richiesti per amplificare in maniera accurata piccoli segnali. In linea teorica, se entrambi gli ingressi di un amplificatore operazionale si trovano al medesimo potenziale, l'uscita dovrebbe essere nulla. Ciò in realtà non si verifica e in uscita ci sarà sempre una

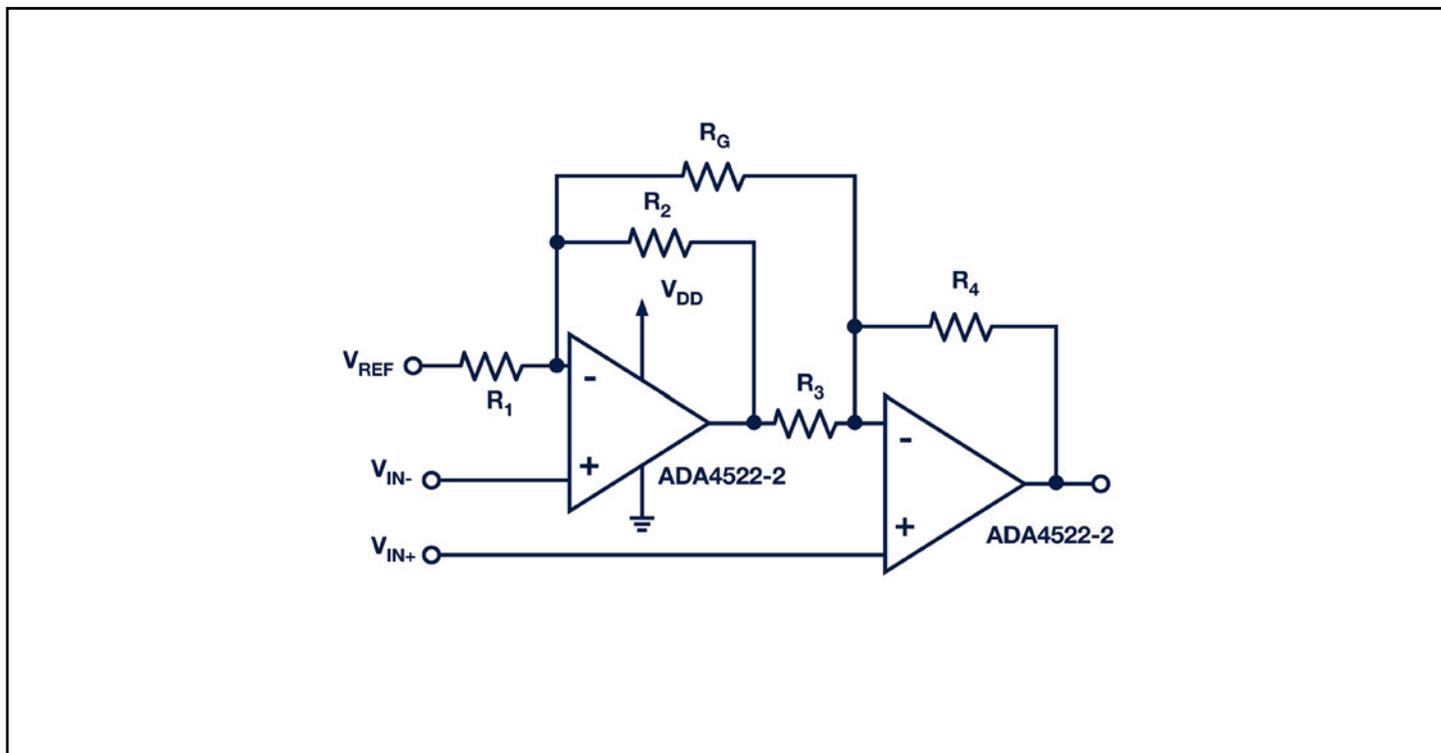


Figura 2: Un circuito con un amplificatore operazionale doppio può servire a superare alcune delle imprecisioni intrinseche legate all'impiego di un amplificatore operazionale singolo (fonte: Analog Devices).

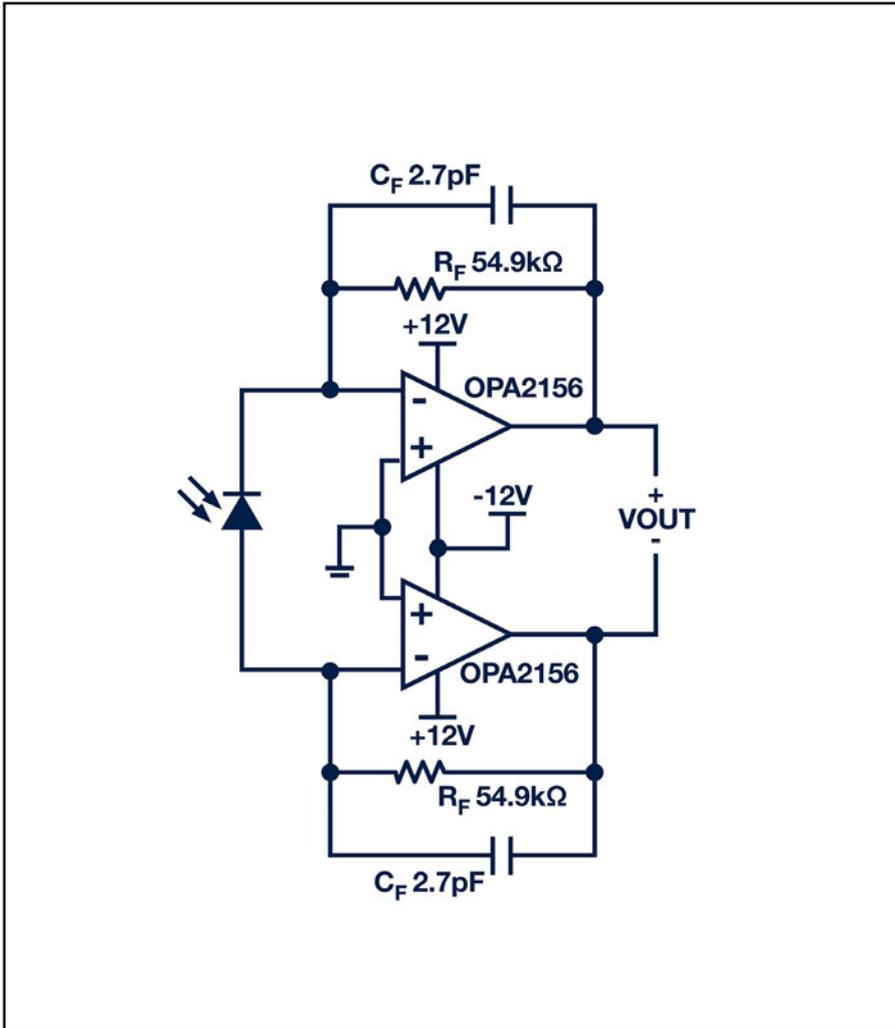


Figura 3: Un esempio applicativo di un amplificatore per fotodiodi differenziale (fonte: Texas Instruments)

tensione di errore di valore ridotto nota come tensione di offset di ingresso.

L'amplificatore operazionale, inoltre, potrebbe influenzare segnali molto deboli provenienti dai sensori (come ad esempio quelli prodotti dai sensori che operano senza batteria, tipici delle applicazioni IoT) nel caso esso richieda una potenza troppo elevata per il suo funzionamento. Gli altri fattori che i progettisti devo-

no tenere in considerazione sono l'impedenza di ingresso dell'amplificatore operazionale (che essendo dell'ordine dei MOhm non dovrebbe rappresentare un problema) e la corrente di polarizzazione di ingresso (ovvero la corrente richiesta per il funzionamento dell'amplificatore operazionale) in relazione al segnale del sensore che deve essere memorizzato nel buffer.

Per ovviare alle limitazioni legate all'uso di un singolo amplificatore operazionale, gli amplificatori di precisione spesso usano una coppia di amplificatori operazionali con caratteristiche simili (matched) integrati in un unico package. La configurazione circuitale riportata in Figura 2 è comunemente utilizzata con

amplificatori operazionali doppi come quelli della serie [ADA4522](#) di Analog Devices.

Ottimizzato per l'uso con qualsiasi tipo di sensore che produce piccoli segnali, questo circuito si propone come una soluzione a basso rumore con deriva nulla. Grazie a una corrente di polarizzazione di ingresso di appena 50 pA e una tensione di offset di ingresso di soli 0,7 μ V, qualsiasi impatto su segnali ad alta impedenza

è trascurabile, consentendo un'accurata amplificazione dei segnali provenienti da qualsiasi tipo di sensore.

Utilizzando un amplificatore operazionale dedicato per ciascun ingresso, l'impedenza di ingresso del circuito sarà definita dall'impedenza di ingresso di modo comune che assume un valore pari all'incirca a 100 GOhm. Poiché gli amplificatori operazionali sono contenuti in un singolo package, qualsiasi errore di offset viene compensato internamente, semplificando in tal modo il processo di sviluppo di soluzioni contraddistinte da un elevato grado di precisione. Un altro amplificatore operazionale doppio è il mod. [OPA2156](#) ad alta precisione di Texas Instruments.

Questo dispositivo della prossima generazione è caratterizzato da valori estremamente ridotti di tensione di offset ($\pm 200 \mu\text{V}$ max.), corrente di polarizzazione ($\pm 5 \text{ pA}$) e deriva ($\pm 3 \mu\text{V}/^\circ\text{C}$) che, combinati, permettono di ottenere una soluzione che abbina elevata precisione a livelli di rumore estremamente bassi. Questo dispositivo si distingue per alcune peculiarità tra cui la possibilità di commutare sia l'uscita sia l'ingresso tra i terminali (rail), estesa ampiezza di banda (25 MHz) ed elevato slew rate ($40 \text{ V}/\mu\text{s}$) che lo rendono particolarmente adatto per applicazioni di rilevamento in rapido movimento. Esso viene fornito in un

package SOIC a 8 pin e opera sull'intervallo di temperatura industriale, compreso tra -40 e $+125$ $^\circ\text{C}$.

Alcuni sensori prevedono differenti tipi di uscite, che possono quindi alternare differenti configurazioni di amplificatori operazionali per amplificare correttamente le loro uscite.

L'esempio riportato in figura 3 è basato su un fotodiodo con un intervallo di corrente compreso tra 0 e $90 \mu\text{A}$ progettato per fornire un'uscita nell'intervallo da -5 e $+5 \text{ V}$.

Grazie all'integrazione di un condensatore nell'anello di retroazione, si viene a creare un filtro che permette di ridurre la necessità di ricorrere a filtri addizionali di tipo discreto. Nell'esempio in questione è stato specificato un condensatore di $2,7 \text{ pF}$ per fornire un frequenza di taglio (cut-off) di 1 MHz , che permette di eliminare il rumore ad alta frequenza indesiderato. A valle della sezione analogica della maggior parte delle catene di segnale è presente un convertitore A/D che converte il segnale nel dominio digitale per la successiva elaborazione (tramite DSP) o per la memorizzazione dei dati. Una volta memorizzato nel buffer, filtrato e successivamente amplificato il segnale, è importante scegliere il convertitore A/D più idoneo per la particolare applicazione considerata.



CONVERSIONE DEI DATI: ALCUNE CONSIDERAZIONI

I convertitori A/D sono sotto-sistemi relativamente complessi per cui è necessario prendere in considerazione un certo numero di parametri per assicurare il conseguimento degli obiettivi di progetto del sistema complessivo. I quattro principali parametri di un convertitore A/D sono i seguenti: risoluzione, velocità, accuratezza e rumore. Spesso la risoluzione viene confusa con l'accuratezza, ma è importante sottolineare il fatto che si tratta di due parametri completamente differenti tra di loro.

La risoluzione si riferisce al numero di bit che vengono generati per ogni campione o conversione. Essa definisce il più piccolo segnale che può essere convertito così come la più piccola variazione incrementale del segnale che provocherà una variazione percepibile in uscita. L'accuratezza definisce invece il livello di fedeltà con il quale l'uscita è in grado di rappresentare l'ingresso e include qualsiasi offset e non-linearità presenti all'interno del convertitore stesso. In linea generale, un incremento della risoluzione comporta un aumento dell'accuratezza della conversione. La velocità può rappresentare un problema in certi tipi di applicazione (come ad esempio l'elaborazione di segnali video) poiché in questi casi la conversione deve essere effettuata a una

velocità pari ad almeno il doppio del valore della frequenza più elevata presente nel segnale sorgente. Nelle applicazioni di rilevamento, specialmente quelle che prevedono la misura di parametri fisici (come ad esempio la temperatura), raramente la velocità rappresenta un problema. Tutti i convertitori A/D generano una qualche forma di rumore di quantizzazione, un inevitabile effetto collaterale della conversione. Se, ad esempio, l'ingresso è un'onda sinusoidale pura, l'uscita sarà un'onda sinusoidale formata da una serie di piccoli gradini e le non linearità presenti sono note sotto il nome di rumore di quantizzazione. L'incremento della risoluzione contribuirà ad attenuare gli effetti del rumore di quantizzazione all'interno di un sistema.

Una volta analizzati i segnali che influenzano l'integrità del segnale, i progettisti devono esaminare gli altri parametri del dispositivo, compresa l'interfaccia digitale verso il sistema host e il numero di canali disponibili, oltre a qualsiasi altra caratteristica specifica richiesta sia dal sistema sia dall'applicazione. [ADS1219](#) di Texas Instruments è un convertitore A/D di precisione a 24 bit che si distingue per l'elevato livello di flessibilità grazie alla possibilità di configurare i suoi ingressi come quattro ingressi a terminazione singola (single ended) in multiplex o come coppia di ingressi differenziali. Questo dispositivo, che richiede un nu-

mero minimo di componenti esterni, integra a bordo un buffer che consente il collegamento diretto con sorgenti a elevata impedenza.

A bordo è anche presente uno stadio di guadagno programmabile per un'ulteriore condizionamento del segnale. Ospitato in package WQFN o TSSOP a 16 pin, il convertitore ADS1219 integra un'interfaccia I²C-compatibile a due fili che consente il trasferimento dei dati al sistema host a una velocità massima di 1Mbps.

CONSIDERAZIONI CONCLUSIVE

La progettazione analogica rimane una componente molto importante del mondo moderno. Per continuare a sfruttare i vantaggi derivati da un monitoraggio affidabile e accurato su base continua, è indispensabile disporre di una catena del segnale analogica ad alte prestazioni. Per questo motivo, in prima istanza è necessaria una completa comprensione della natura dei sensori e, in particolare, dei loro segnali di uscita. Questa catena del segnale sarà formata da amplificatori operazionale, componenti discreti e da un elemento per la conversione dei dati. I progettisti si impegneranno sicuramente a specificare in modo corretto i componenti attivi, ma non devono dimenticare di riservare un'identica attenzione nello specificare i componenti passivi che definiranno le prestazioni del sistema.



Mark Patrick - Mouser Electronics

Mark lavora per Mouser Electronics dal Luglio 2014, dopo aver ricoperto ruoli senior nel settore marketing di RS Components. Nel periodo precedente a quest'ultima, per 8 anni il suo impegno ha riguardato attività di Applications Support e Technical Sales in Texas Instruments. Mark Patrick ha conseguito un Dottorato di ricerca di primo grado in ingegneria elettronica presso la Coventry University in UK.

Hai la passione dell'elettronica e della sperimentazione?

Ti senti anche piuttosto portato a "mettere nero su bianco" i tuoi progetti ed esperimenti in forma chiara e comprensibile? Ti piacerebbe pubblicare un tuo articolo su queste pagine e sul sito web di EM?

Scrivi a

redazione@elettronicaemaker.it

indicando i tuoi campi d'interesse e allegando un elaborato sull'argomento d'elettronica preferito: ci risentiremo!



Elettronica & Maker

DIRETTORE

Roberto Armani

Art Director

Shylock-58

Hanno collaborato a questo numero:

Marco Castiglioni,
Stefano Lovati, Mark Patrick,
Walter Ribbert, Mario Rotigni,
Roberto Vallini

AVVERTENZE

Chiunque decida di fare uso delle nozioni riportate in questi articoli o decida di realizzare i circuiti esposti, è tenuto a prestare la massima attenzione in osservanza alle normative in vigore sulla sicurezza. Gli Autori di Elettronica&Maker sopracitati, che hanno collaborato alla realizzazione degli articoli pubblicati in questo numero, declinano ogni responsabilità per eventuali danni causati a persone, animali o cose derivante dall'utilizzo diretto o indiretto del materiale, dei dispositivi o del software presentati. Si avverte inoltre che quanto riportato negli articoli viene fornito così com'è, a solo scopo hobbistico, senza garanzia alcuna di correttezza e di funzionamento certo. L'editore e gli autori ringraziano anticipatamente per la segnalazione di ogni eventuale errore.

Su Elettronica & Maker

Elettronica & Maker è una testata pubblicata in formato esclusivamente elettronico, in formato portatile sfogliabile elettronicamente, diffusa esclusivamente per via telematica, non soggetta all'obbligo di registrazione presso il Tribunale, su sito web <https://www.elettronicaemaker.it> ed in fné al R.O.C. né agli obblighi dell'AgCom n° 666/08 del 26/11/08, a fronte del D.L. n° 63 del 18 Maggio 2012.

© Copyright

Tutti i diritti di riproduzione o di traduzione degli articoli pubblicati sono riservati. Manoscritti, disegni e fotografie sono di proprietà di E&M. È vietata la riproduzione anche parziale degli articoli salvo espressa autorizzazione scritta dell'editore. I contenuti pubblicitari sono riportati senza responsabilità, a puro titolo informativo.

Collaborare con Elettronica & Maker

Le richieste di collaborazione vanno indirizzate all'attenzione di Roberto Armani (r.armani@elettronicaemaker.it) e accompagnate, se possibile, da una breve descrizione delle vostre competenze tecniche e/o editoriali, oltre che da un elenco degli argomenti e/o progetti che desiderate proporre.

La grafica in copertina è di Tomasz Mikolajczyk