

# DIY: Battery tester-decharger

*Un circuito per scaricare le batterie ?!*

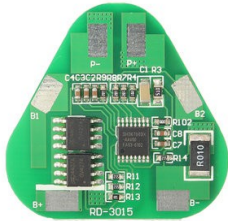
*Un problema si presenta spesso quando si usano batterie ricaricabili in serie/parallelo: valutare lo stato di carica del singolo elemento. Si pensi a due batterie ricaricabili NiMH, molto cariche messe in serie ad una terza quasi scarica: a vuoto possono presentare circa la stessa tensione, ma scaricandosi può succedere che la batteria più debole vada in overdecharge prima che la tensione totale scenda sotto i fatidici 2,7 V. (  $1,2+1,2+0,8 = 3,2$  V vedi per un'analisi più dettagliata).*

*In altre parole 3 batterie in serie devono essere non solo uguali (stessa marca, tipo e capacità) ma anche nello stesso stato di carica!*

*Il problema è molto reale per le batterie tipo Lilon, che possono incendiarsi e perfino scoppiare in casi estremi. Esistono batterie Lilon cosiddette 'protette' che hanno all'interno della batteria stessa una circuiteria elettronica per evitare overcharge ed overdecharge.[1]*

*I maggiori rischi si corrono quando si usano pacchi serie/parallelo di batterie Lilon: o si usano batterie 'protette', oppure esistono circuiti per gestire pacchi di batterie non protette.[2]*

*Esempio:*

	<u>4A 3S Lilon Lithium Battery Protection Circuit Board Three Cell PCB</u>	2,11 €
--	--	--------

*Le batterie NiMH sono più robuste e meno pericolose. Il danno tipico è la riduzione della capacità di carica, quasi sempre irreversibile e che può giungere a rendere inutilizzabile la batteria.*

*In generale le batterie NiMH non devono superare i limiti di overcharge e di overdecharge e devono essere conservate cariche (anche solo al 50%), rinfrescando la carica una volta al mese.*

Nei circuiti utilizzati in e3DHW-PMS per l'alimentazione di apparecchiature elettroniche sono usate, nei casi di minor potenza, 3 batterie al NiMH oppure una sola batteria Lilon. Le batterie NiMH, al contrario delle Lilon, possono essere tenute sempre sotto carica (carica 'lenta') e quindi essere usate come UPS. Per creare un pacco di 3 batterie NiMH in modo corretto, che garantisca una lunga vita alle batterie, si può usare la seguente strategia:





1. usare 3 batterie uguali: stessa marca, tipo e capacità
2. scaricare singolarmente le batterie, fino al valore di sicurezza (0,9 V)
3. metterle in serie e ricaricare l'intero pacco fino a 4,4 V
4. verificare che ogni batteria abbia la carica prevista (circa 1,47 V).

Questo tester permette di valutare lo stato di salute della singola batteria e di effettuare il passo 2 in modo affidabile.

In altri casi si può voler caricare una batteria in modo 'cieco', ad esempio per 1h a 1,2 C. Questa strategia è valida solo partendo da batterie sicuramente scariche: un altro caso d'uso per questo tester.

# Caratteristiche

- *Tester per batterie AA al NiMH a microcontroller.*
- *Funzionamento automatico: inserire la batteria AA al NiMH e verificare i LED.*
- *Scarica sicura delle batterie cariche, un Beep segnala il raggiungimento di 0,9 V, evitando il rischio di overdecharge.*
- *Circuito passivo, senza alimentazione: utilizza la carica residua della batteria sotto test.*

Tensione batteria	LED Test	Azione
$V < 0,6^1$	spenti 	Irrecuperabile Batteria da eliminare
$0,6^1 < V < 0,9^2$	Rosso fisso 	Overdecharge Forse utilizzabile a capacità ridotta
	Rosso lampeggiante 	Decharge Attendere la fine scarica
$V > 0,9^2$ (max 5 V)	Verde beep  +	Fine scarica (0,9 V) Togliere subito la batteria

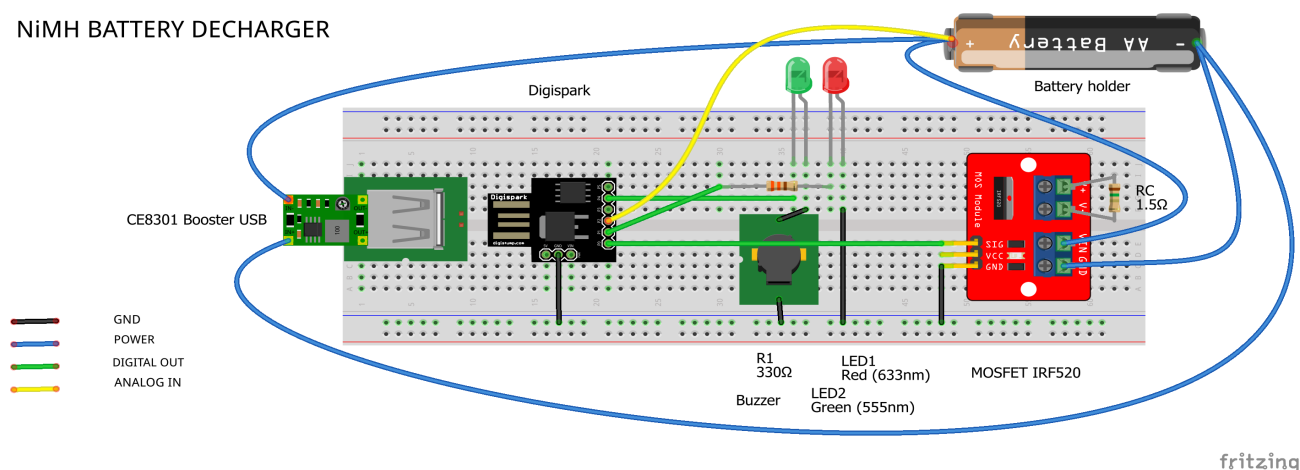
(1) indicativo, limite inferiore di funzionamento del circuito Booster CE8301

(2) dipende dalla  $V_{ref}$  interna di ATtiny85. Tarabile modificando la costante ADC\_ADJ nello sketch (verificare con un voltmetro digitale).

Nota: i LED si accendono dopo 5 secondi dall'inserimento della batteria, a causa della procedura di avvio di ATtiny85.

# Circuito


## NIMH BATTERY DECHARGER

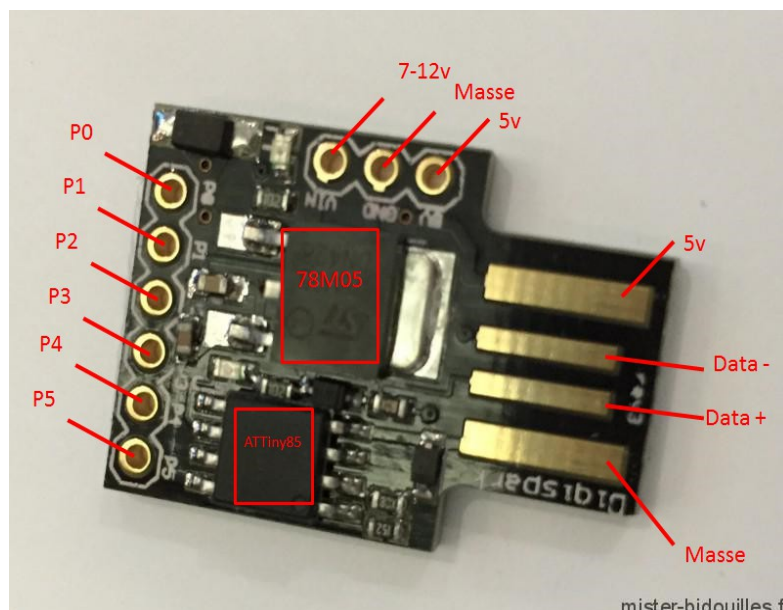
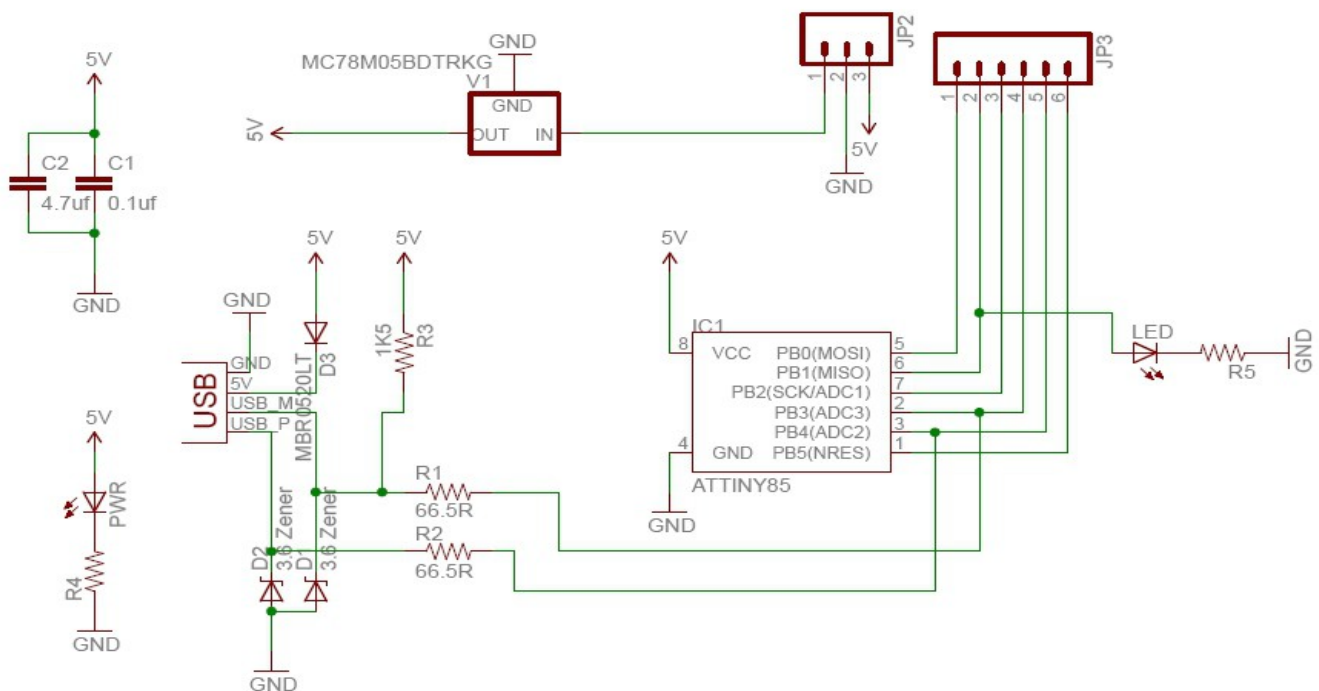


Il circuito è molto semplice: 4 moduli e 3 componenti. L'uso di economici moduli premontati rende superfluo l'uso di circuiti stampati ad hoc e questo semplifica molto sia i test che il montaggio finale.

## Microcontroller

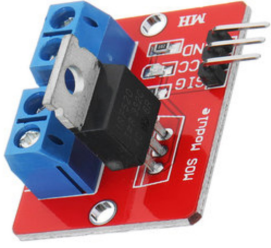
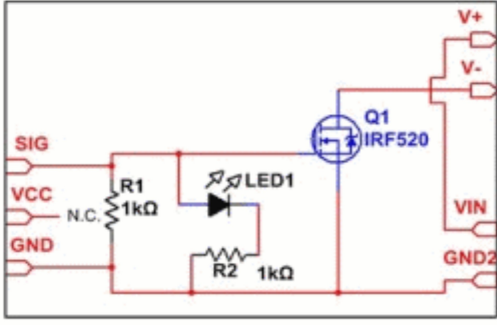
Il componente principale è il Microcontroller ATtiny85 (AVR, 8 bit, 8K) contenuto della scheda Digispark [3]

	<p><u>Digispark Micro scheda di sviluppo ATTINY85</u></p>	<p>1,21 €</p>
---	---	---------------



nota: I pin di I/O, indicati sullo schema elettrico precedente come JP3 [1..6] sullo stampato corrispondono a **P0..P5** e sono così usati:

**P0:** OUTPUT DIGITALE, collegato all'ingresso SIG di una scheda MOSFET (IRF520), che comanda la chiusura della batteria sotto test (GND2 e VIN) sul resistore di carico (RC). Quando P0 è alto RC è collegato alla batteria.

		<p><b>IRF520 MOS Driver Module</b></p> <p>1.26 €</p>
---	--	--

Il resistore di carico può essere facilmente cambiato (è collegata ai due morsetti V+ e V-) se necessario per velocizzare la scarica.

RC non ha un valore critico. La corrente di scarica ( $x_C$ ) deve essere compresa tra 0.1C e 2C (dove C è la capacità della batteria in Ah) ed il resistore deve essere di adeguata potenza (2W – 4W). Non dovrebbero esserci problemi di riscaldamento eccessivo se l'aria può circolare attorno al resistore. Calcolare che il circuito, con i vari LED, consuma circa 80 mA.

Per la precisione, in formule:

$$RC \approx \frac{1.2}{x_C - 0.08} \quad [\Omega]$$

$$P > 1.4 \cdot x_C \quad [W]$$


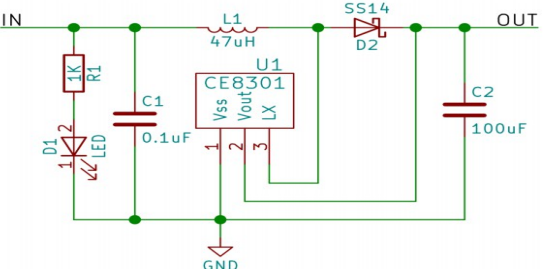
**P1:** OUTPUT DIGITALE. Controlla il LED rosso (LED1), con in serie una resistenza limitatrice di corrente di 330  $\Omega$  (R1). Quando P1 è alto il LED è acceso.

**P2:** INPUT ANALOGICO. Collegato al positivo della batteria per misurare la tensione con l'ADC interno di Attiny85 (10 bit). Poiché il Microcontroller ha un'alimentazione bassa e variabile, l'ADC deve usare la tensione interna di riferimento di 1,1 V.

**P4:** OUTPUT DIGITALE. Controlla il LED verde (LED2), a massa tramite un Buzzer (o Beeper) elettromagnetico. Mandando un segnale ad onda quadra, il led si accende ed il Buzzer suona,

**P3, P5:** NC

## Alimentazione

		<p><b>PFM Control DC DC 0 9V 5V To USB Step-up</b></p> <p>1,30 €</p>
---	--	--

Un booster con CE8301 (step-up) è usato per elevare la tensione fornita dalla batteria (0,6...1,4 V).

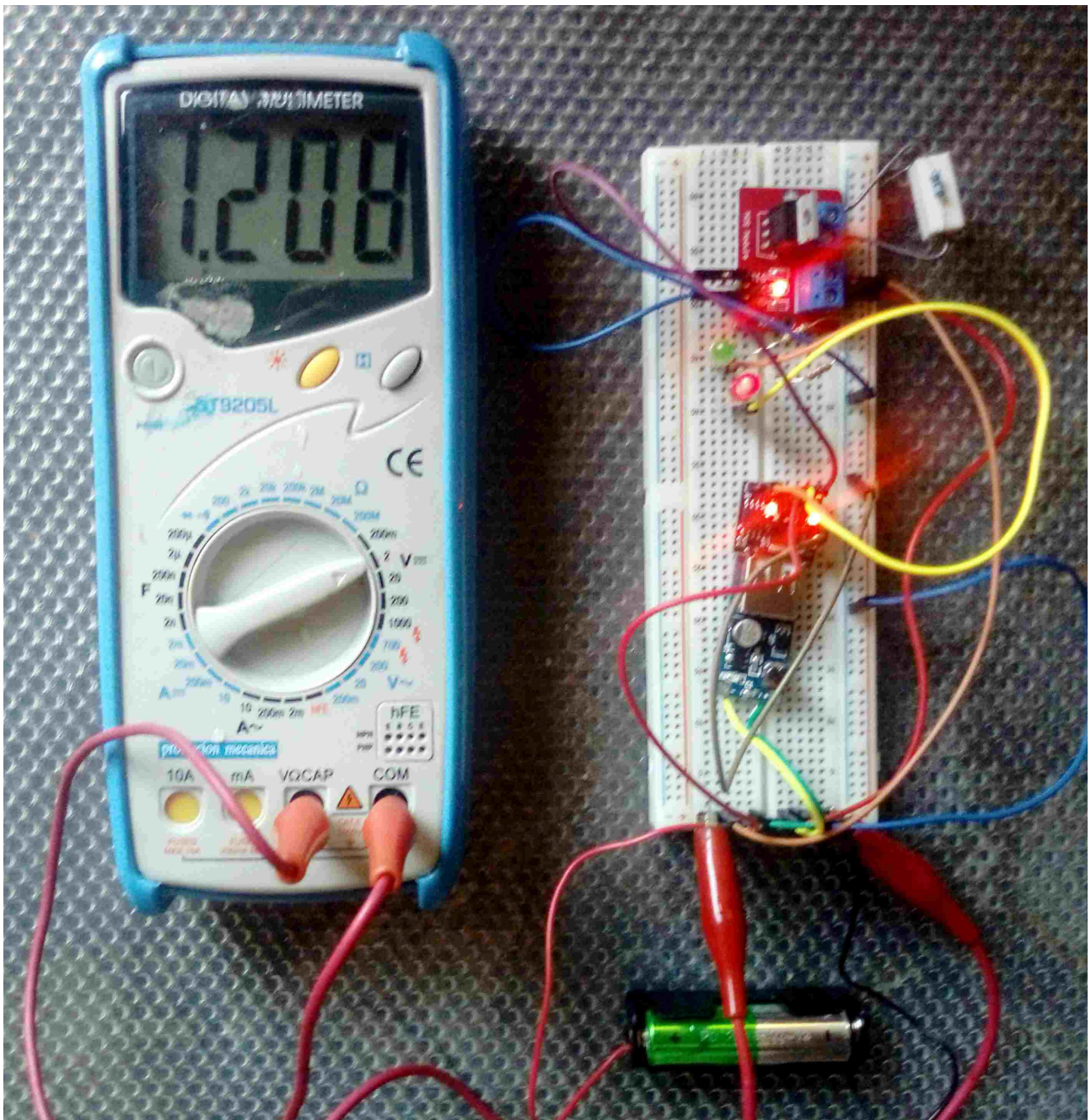
A bassissimi input (0.6V-1V) questo circuito sotto carico non fornisce 5V, ma svolge sempre la sua funzione step-up fornendo circa 3 V, sufficienti per ATtiny85 [5]

La scheda Digispark è direttamente inserita nella presa USB del booster.

Semplicemente estraendo la scheda dal booster ed usando in sua vece un cavo USB connesso al PC si può programmare 'in-circuit' il microcontroller Attiny85, opzione molto comoda per gli aggiornamenti del firmware.



# Prototipo



Il prototipo, dove si fanno misure e modifiche, l'ho realizzato su breadboard con l'uso di cavetti da 20 cm. con connettori (3 tipi: M-M, M-F, F-F).



## Realizzazione

La realizzazione finale è stata fatta in un contenitore di plastica Teko piccolo: 11 x 7 x 5.5 cm.

Rispetto allo schema sono state aggiunte due boccole collegate alla batteria:

1. rendono semplice collegare un voltmetro digitale per misurare la tensione della batteria
2. permettono di usare altre batterie, non di tipo AA

Primo passo la foratura del coperchio superiore: per non scaldare la plastica uso un avvitatore a batteria, con un mandrino per le punte.

Fori: LED:  $\varnothing$  5 mm, boccole:  $\varnothing$  8 mm, viti:  $\varnothing$  3 mm.

Montati gli accessori meccanici (boccole e porta-batteria) i led e le schede premontate sono fissate al retro del pannello, usando soprattutto colla calda.

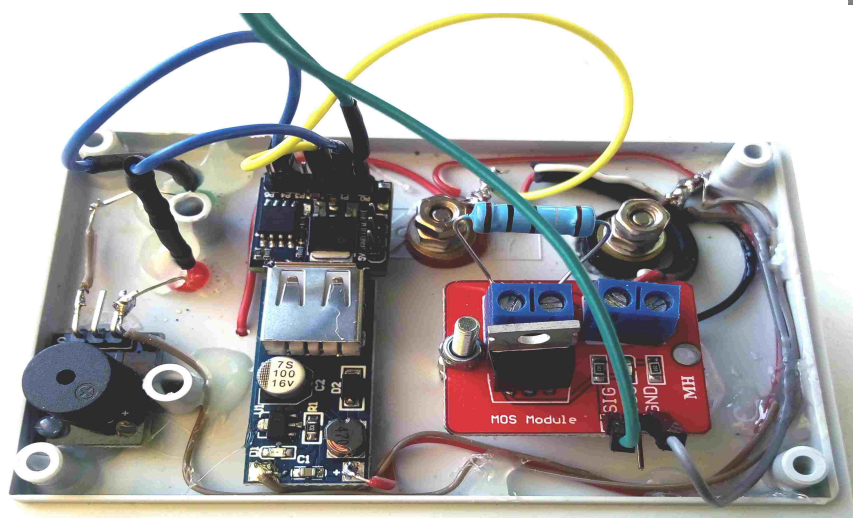
Il microcontroller Digispark non è fissato, ma solo infilato nella presa USB del booster.



Ho iniziato la filatura partendo dal porta-batterie e dalle boccole, usando saldature o cavetti con connettori F.

I fili e i connettori sono resi stabili sempre con colla calda. Un unico componente è volante: R1, montato sul filo e coperto da una guaina termorestringente.

I quattro collegamenti di I/O al microcontroller sono lasciati un po' lunghi per poter estrarre facilmente la scheda Digispark.



# Firmware

Il circuito è molto semplice: tutta la logica è nel programma che è eseguito su ATtiny85. Una grande comodità è che questo Microcontroller può essere sviluppato con Arduino IDE.

I passi necessari sono semplici e perfettamente illustrati in [3]. Inutile ripeterli qui. Con la versione Arduino IDE 1.8.9 non ho avuto nessun problema.

Lo sketch da usare è il file `NiMH_discharger.ino`.

In linea di massima non sono necessarie modifiche allo sketch.

E' comunque facile ridefinire la logica, perché il firmware è strutturato come un semplice ed intuitivo automa di Moore.

Si può tarare con buona esattezza il valore 0,9 V che individua la fine discarica. Controllate con un voltmetro digitale a che tensione scatta il vostro circuito e modificare la costante `ADC_ADJ` all'inizio dello sketch: un paio di prove e la taratura è fatta.

Buon lavoro !

# Riferimenti

1. *Matt Mielke, Simple Undervoltage and Overcurrent Protection for Lithium-Ion and Lithium Polymer Batteries*,  
<https://www.digikey.com/eewiki/display/Motley/Simple+Undervoltage+and+Overcurrent+Protection+for+Lithium-Ion+and+Lithium+Polymer+Batteries>
2. *Texas Instruments, Protect your battery*, <http://www.ti.com/power-management/battery-management/protectors/overview.html>
3. *Adriarobot, Digispark*, <https://www.adriarobot.it/arduino/digispark/digispark.htm>
4. *Matt's Electronics CE8301 Booster Converter* <https://www.matts-electronics.com/wp-content/uploads/2018/06/CE8301.pdf>
5. *MPJA.com, CE8301 datasheet*, <https://www.mpja.com/download/ce830.pdf>
6. *Atmel, Attiny85 datasheet*, [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet.pdf)