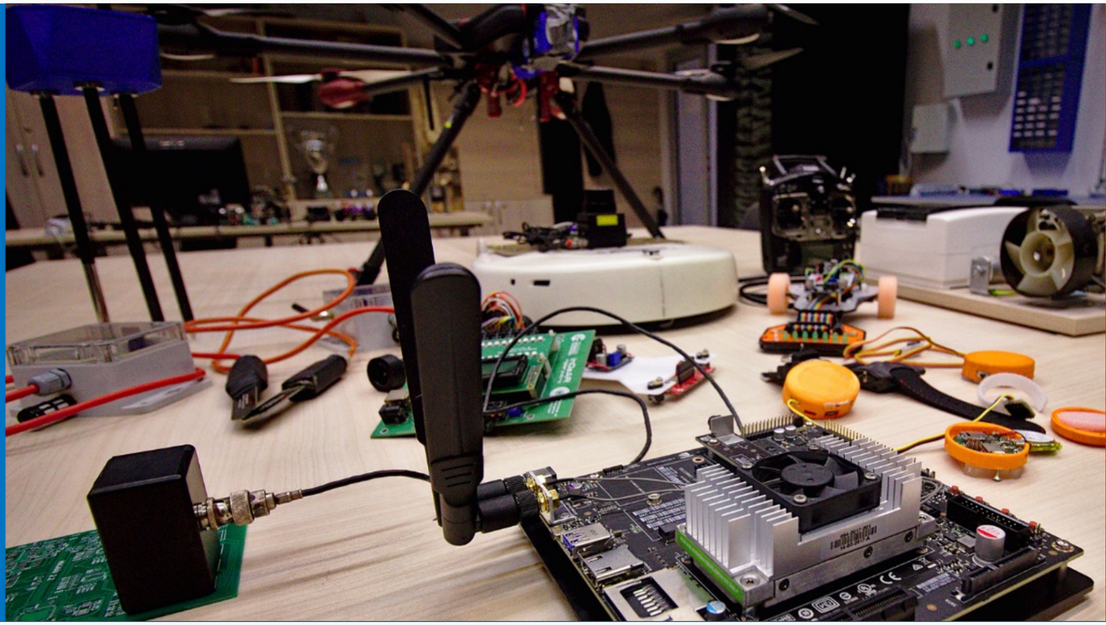
Cursul 8

Power Management

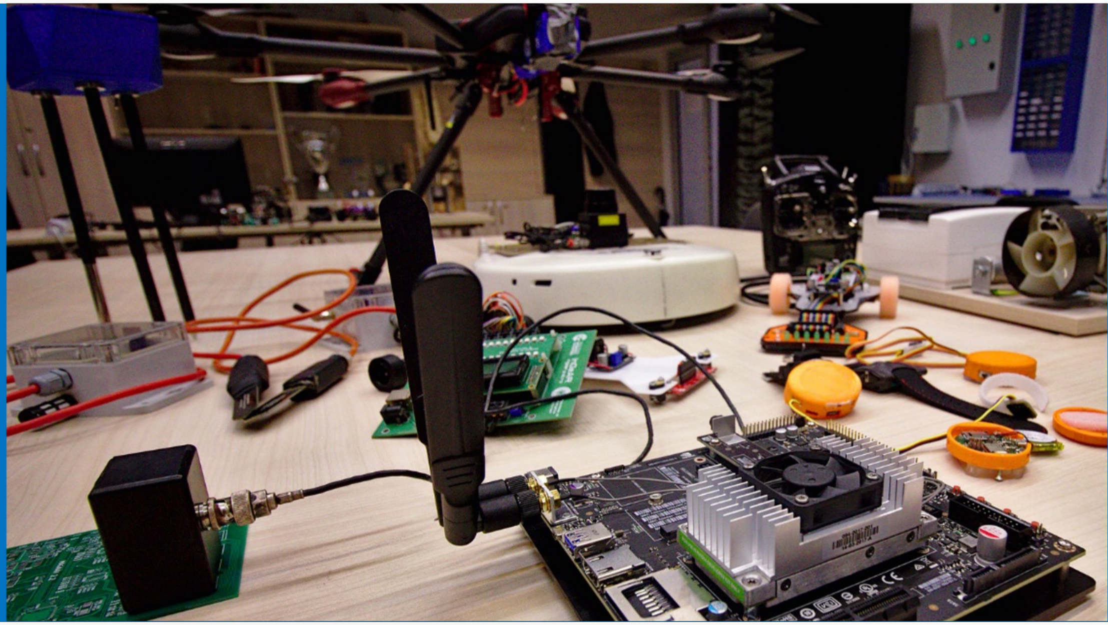


PM

Cuprins

- Stabilizatoare de tensiune
- LDO – Low Dropout
- Surse în comutație
- Alimentare AC
- Consum de energie
- Setări de lucru
- Mod idle / power save
- Optimizări
- Exemple implementări punte H
- Discuții despre proiect

Stabilizatoare de tensiune



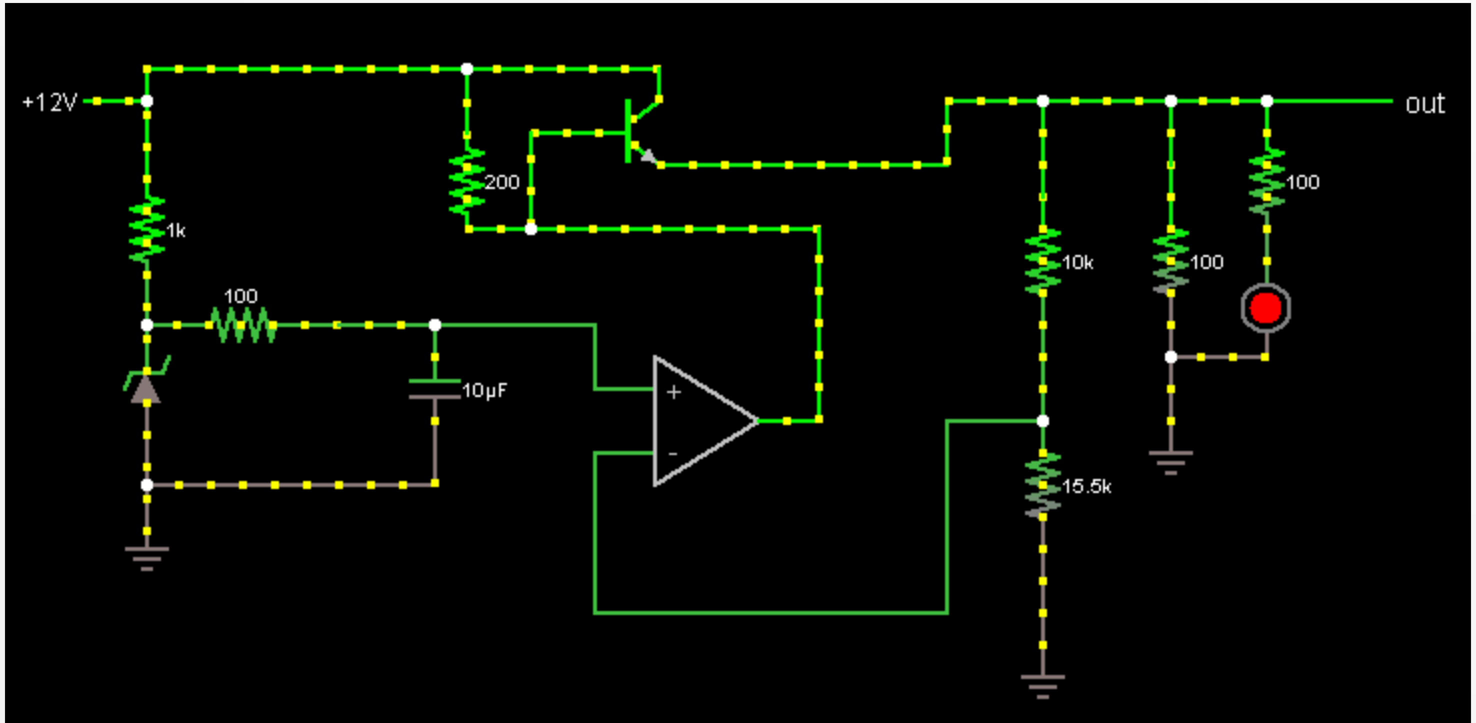
PM

Scenariu de utilizare

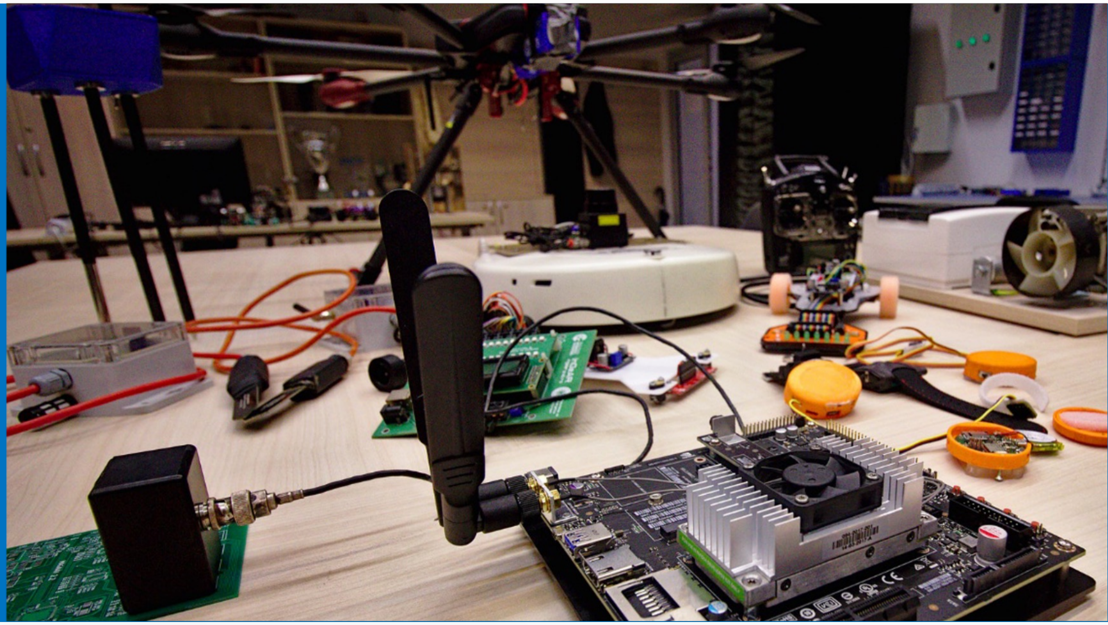
- Protejează împotriva fluctuațiilor de tensiune
- Sunt folosite oriunde avem circuite digitale
  - Centrale termice
  - Aer condiționat
  - Televizoare
  - Calculatoare
  - Etc.
- Sau in cadrul proiectelor de la PM (DC-to-DC)
- Previn defecțiuni nedorite sau funcționări defectoase

Stabilizator de tensiune liniar

- Schemă internă simplificată
- Utilizează o diodă (Zener) sau tranzistor ca element activ
- Funcție de rezistor variabil ce permite reglarea tensiunii de ieșire
- Excesul de energie e disipat prin căldură

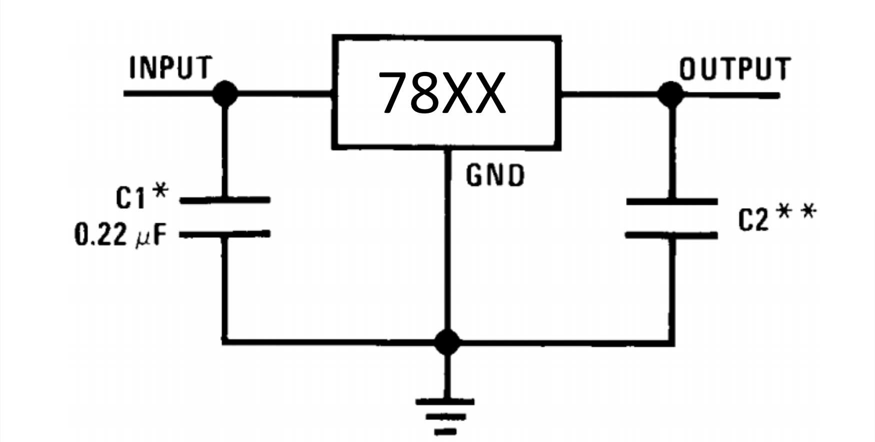


Stabilizatoare de tensiune – liniare integrate



PM

Descriere familia 78/79



- Stabilizator cu tensiune de ieșire fixă
- Ultimele două cifre indică tensiunea de ieșire
- Cel mai comun este 7805 (5V)
- Disponibil si pe Arduino Uno

Familia 78/79

Tensiuni pozitive:

- LM78xx - http://www.fairchildsemi.com/ds/LM/LM7805.pdf
- uA78xx – http://www.ti.com/lit/ds/symlink/ua7805.pdf
- uA78lxx - http://www.datasheetcatalog.org/datasheets/208/402911_DS.pdf

Tensiuni negative:

- LM79xx - http://www.fairchildsemi.com/ds/LM/LM7905.pdf

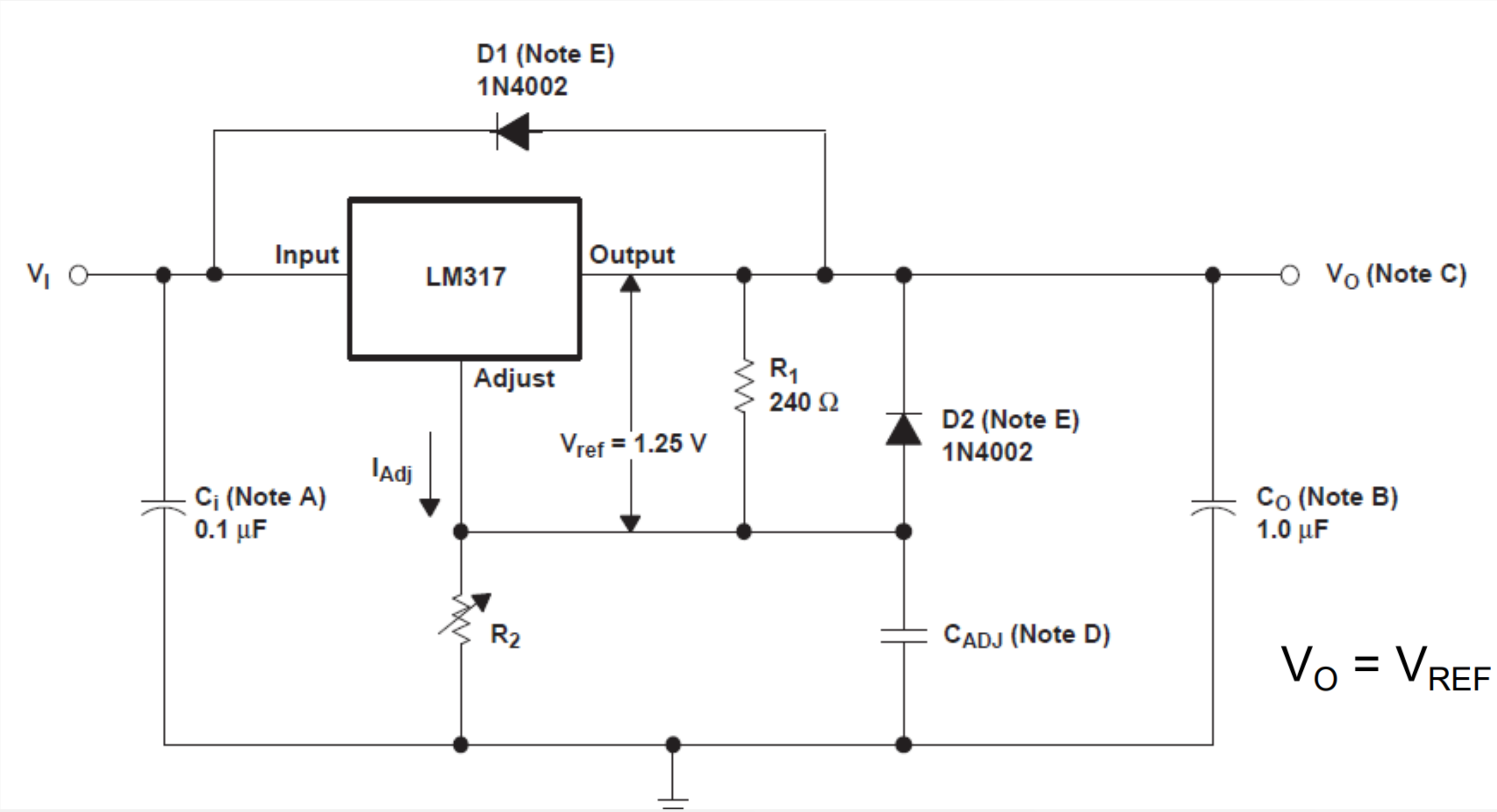
Parametri relevanți

- Voltage dropout
- Quiescent current
- Current limit
- PSSR (Power-supply rejection ratio)
- Output noise

$$PSRR(dB) = 20 \log \frac{V_{ripple(in)}}{V_{ripple(out)}}$$

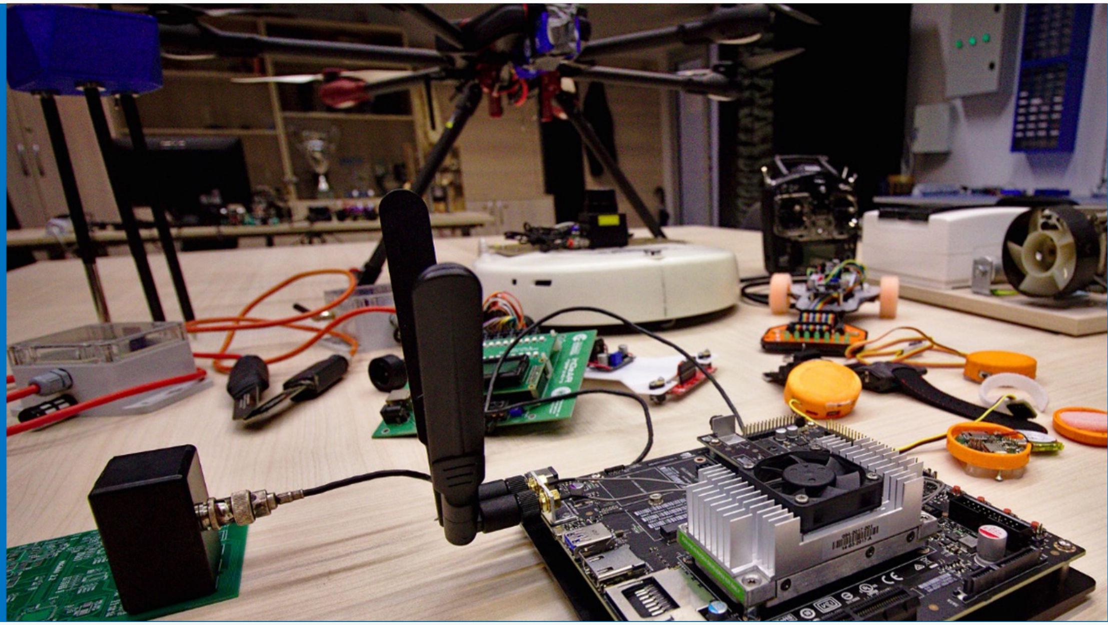
Stabilizator reglabil: LM317

- http://www.ti.com/lit/ds/symlink/lm317.pdf (pag7)



$$V_O = V_{REF}\,(1 + R2 / R1) + (I_{ADJ} \times R2)$$

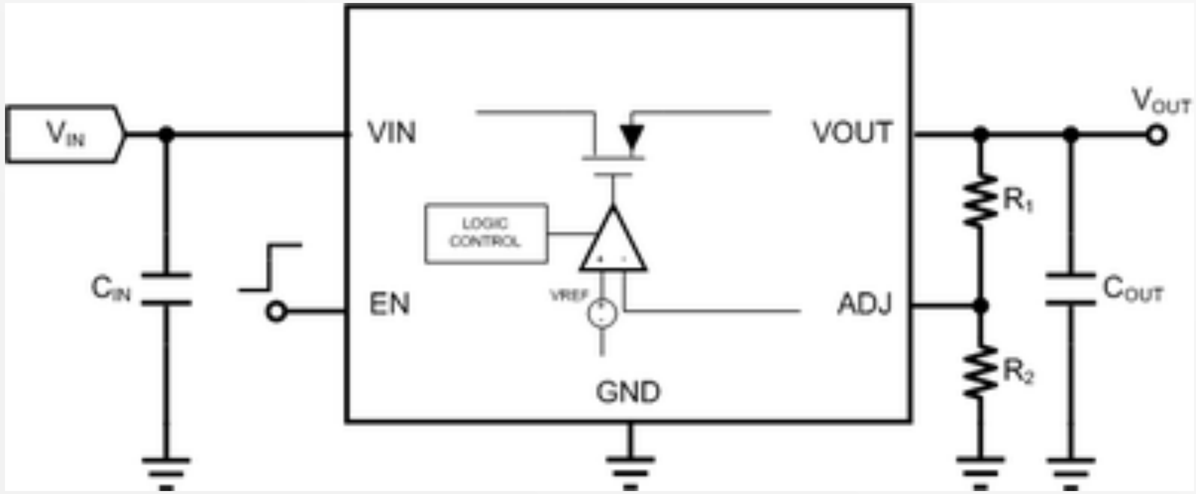
LDO – Low Dropout



PM

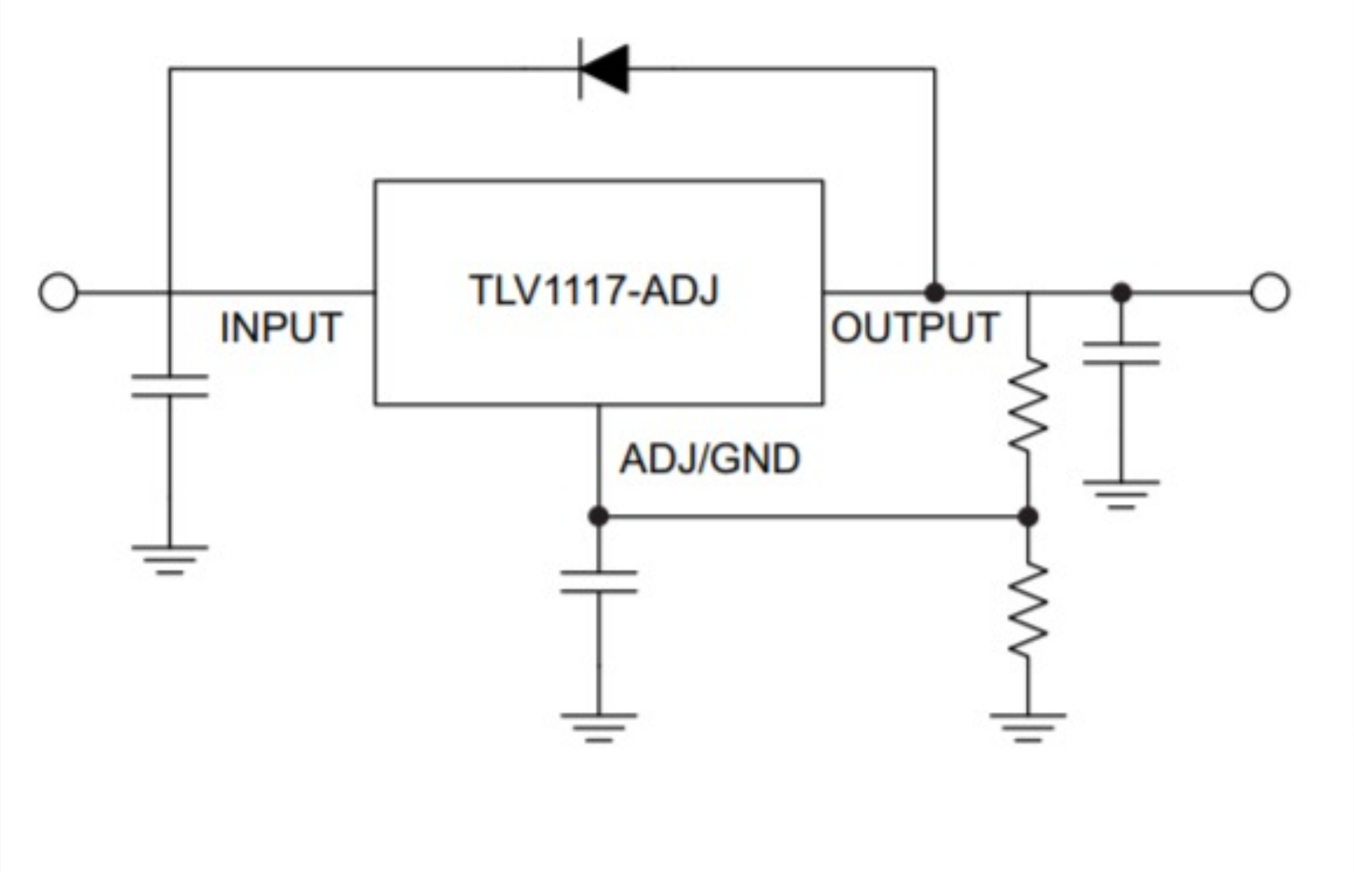
Motivație

- Poate regla tensiunea de ieșire chiar și atunci când tensiunea de alimentare e apropiată de cea de ieșire
- Zgomotul de comutare lipsește în cazul lor
- De asemenea disipă căldură în procesul de stabilizare a tensiunii de ieșire – este tot un stabilizator liniar



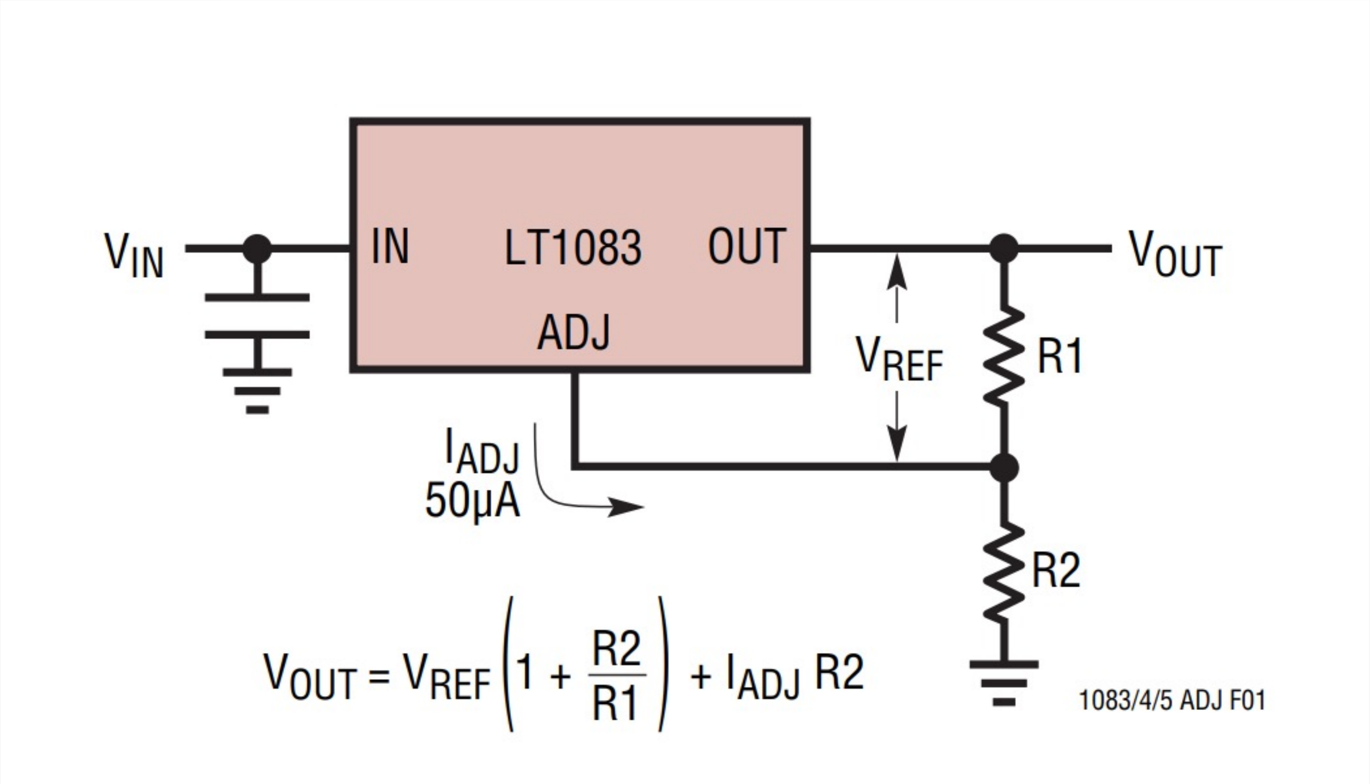
$$V_{\mathrm{OUT}} = \left(1 + \frac{R_1}{R_2}\right) V_{\mathrm{REF}}$$

TLV1117



https://www.ti.com/lit/ds/symlink/tlv1117.pdf

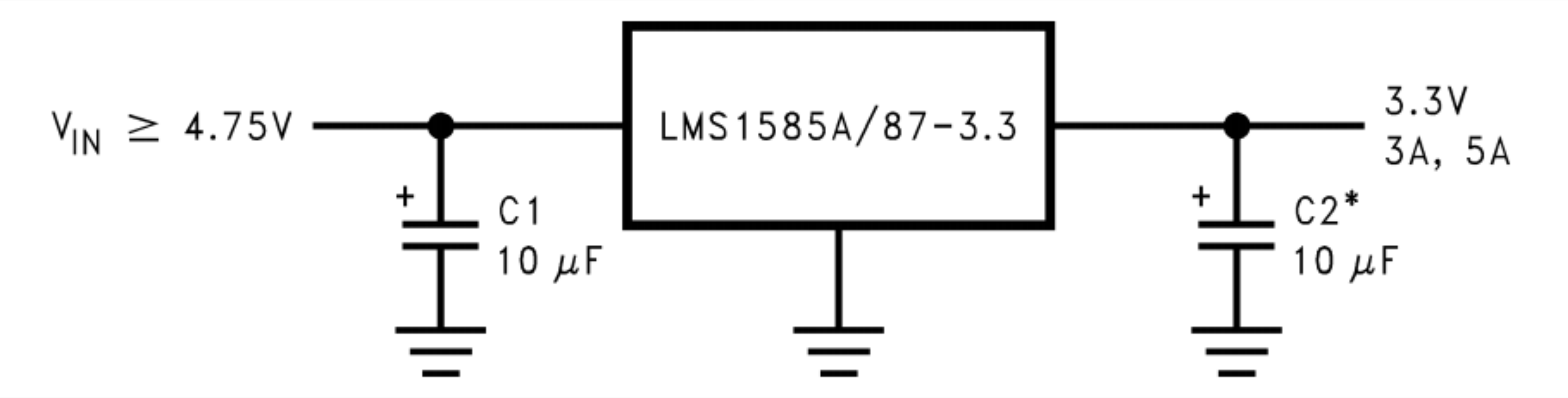
Discutie ADJ



https://www.analog.com/media/en/technical-documentation/data-sheets/108345fh.pdf

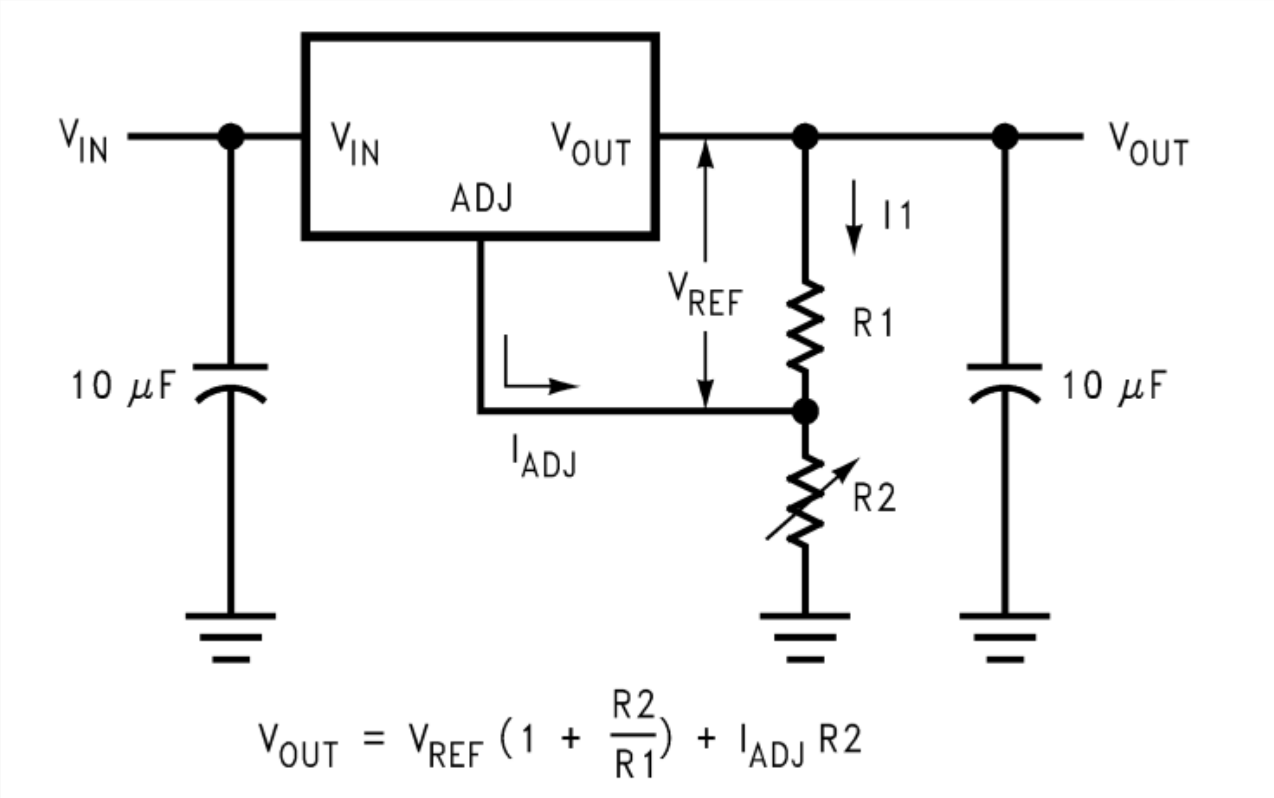
Variante pentru curenți mari

- Tensiuni mici (<3.3V) & curenți mari (5A)
  https://www.ti.com/lit/ds/symlink/lms1585a.pdf



Variante pentru curenți mari

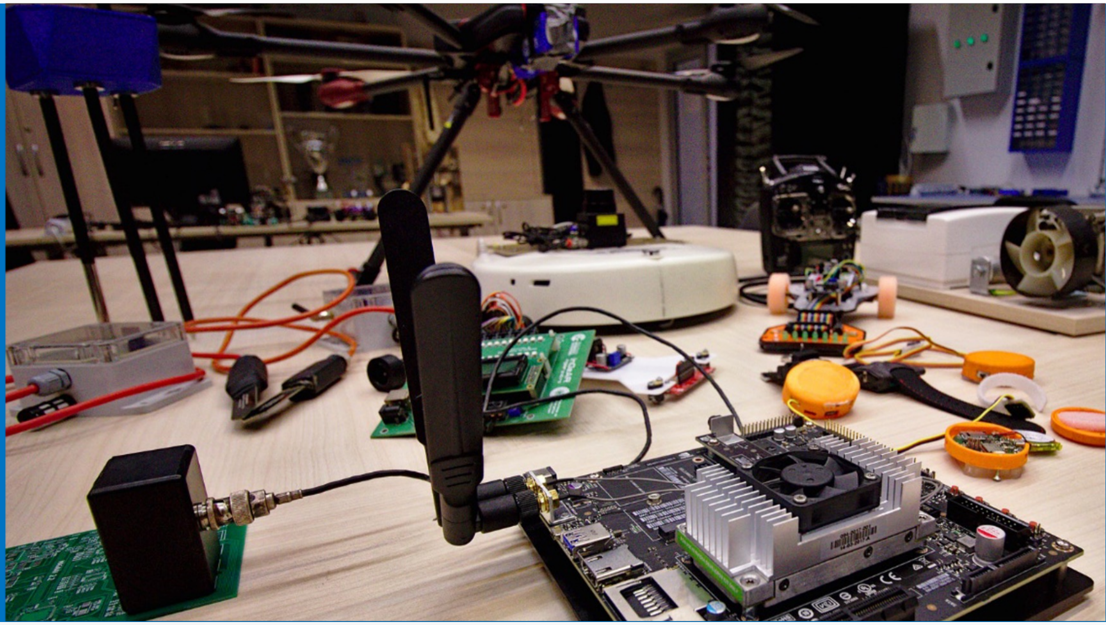
- Tensiuni mici (<3.3V) & curenți mari (5A) https://www.ti.com/lit/ds/symlink/lms1585a.pdf



Variante pentru curenți mici

- http://www.ti.com/lit/ds/symlink/lp3964.pdf
    - **dropout – 340mV la 800mA**

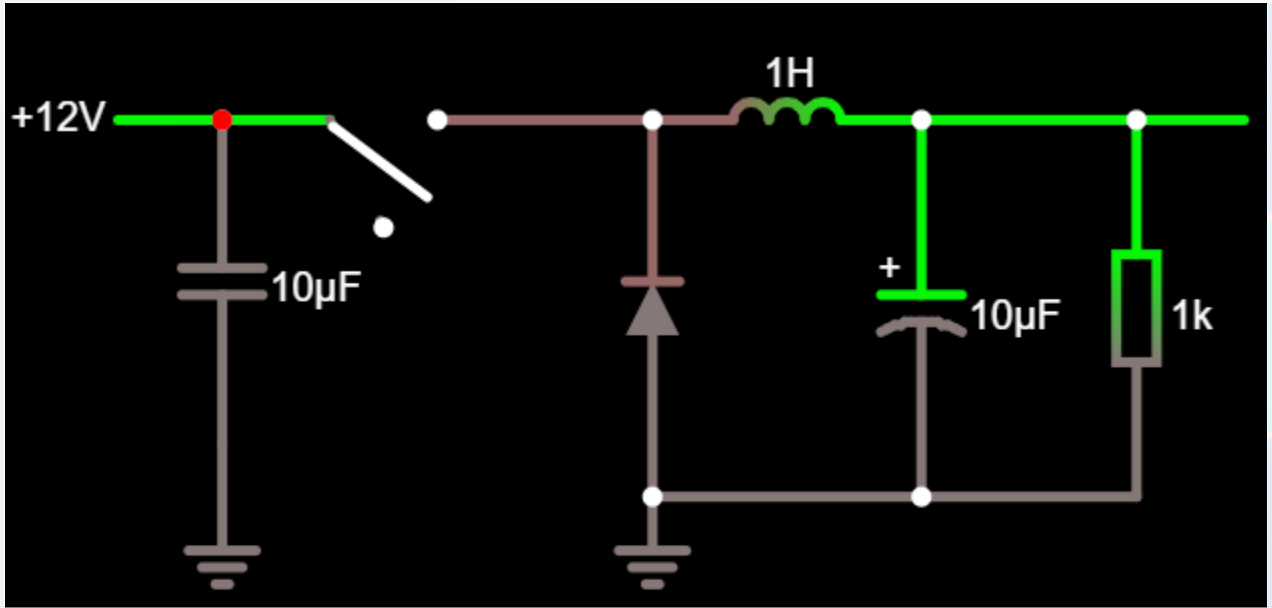
Surse în comutație

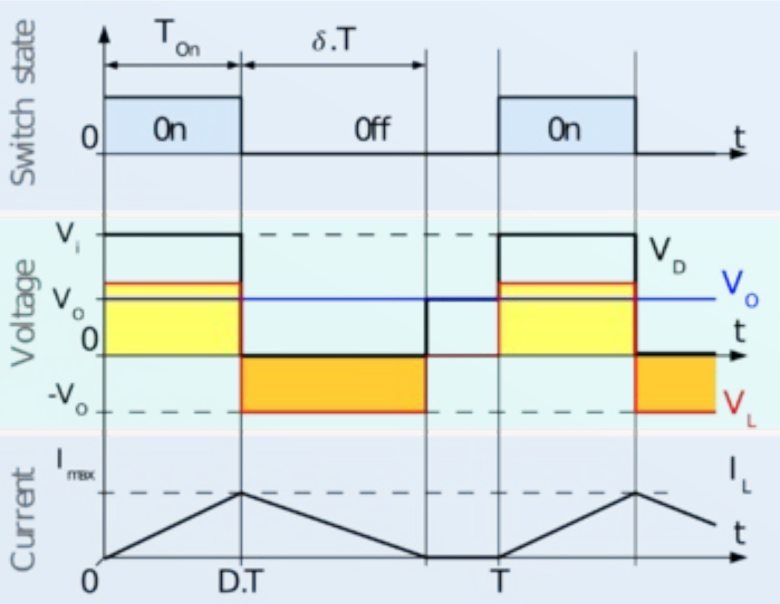


PM

Caracteristici

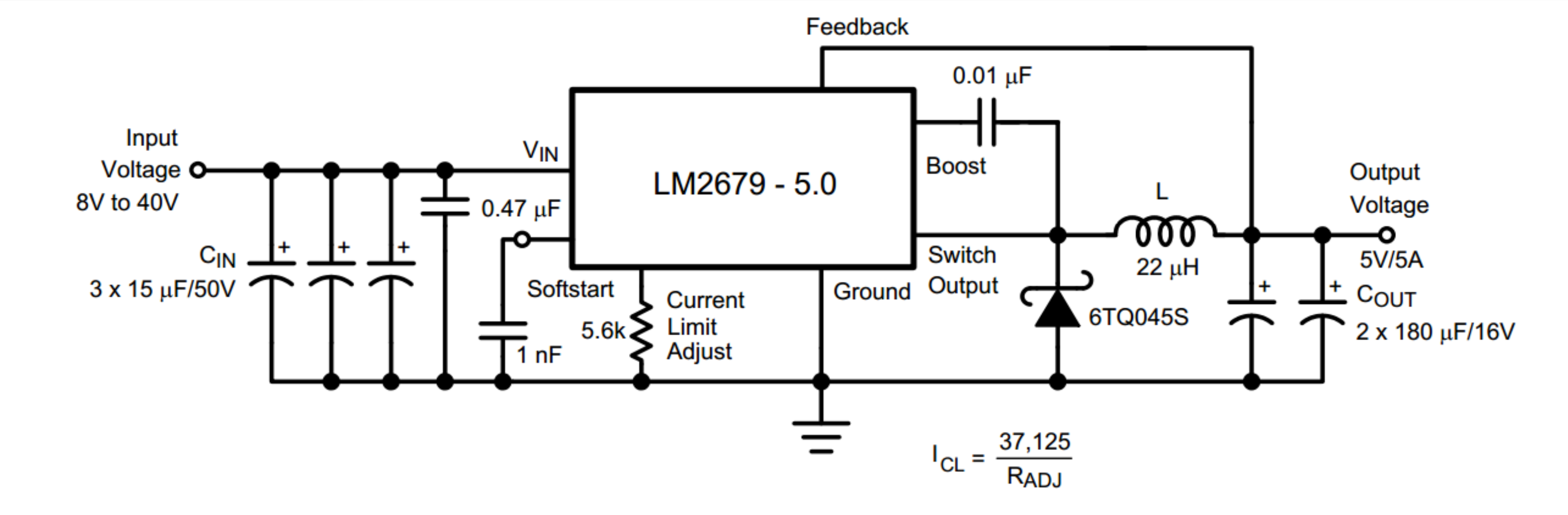
- Transfer de putere de la o sursă DC sau AC la una DC
- Reglarea tensiunii prin raportul dintre timpul de pornire și oprire (duty cycle)
- Eficiență mai ridicată decât regulatoarele de tensiune liniare
- Mai complexe și mai scumpe



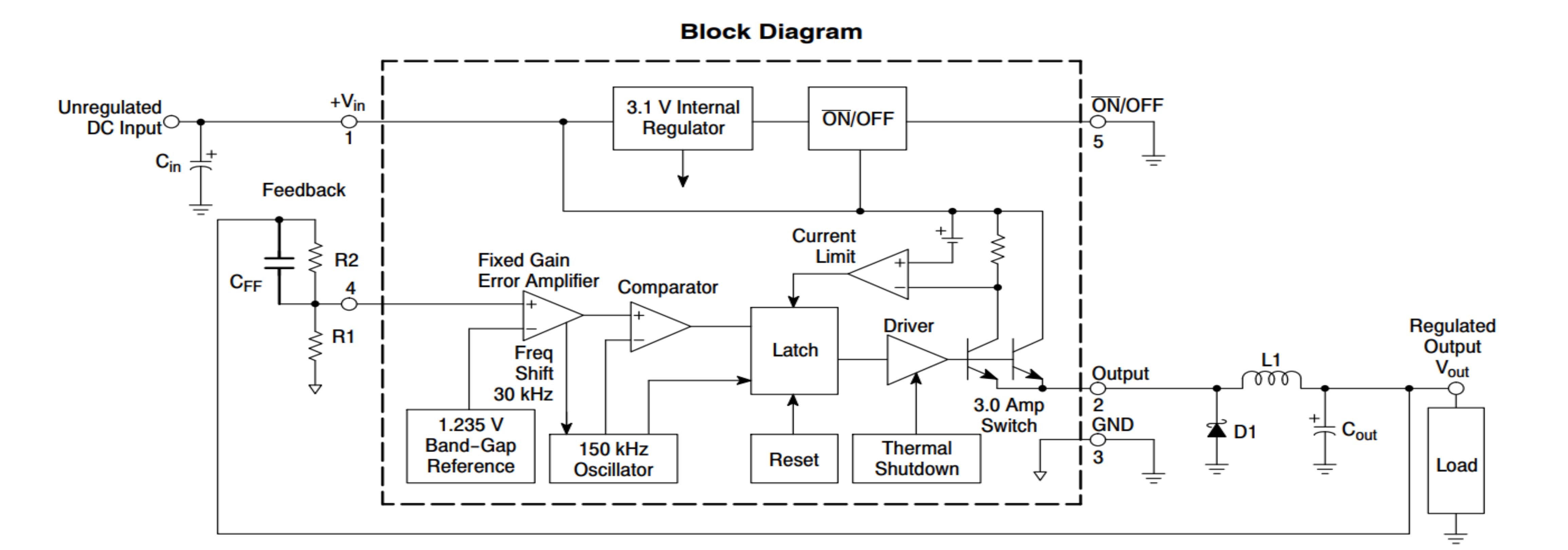


LM2679

- http://www.ti.com/product/lm2679

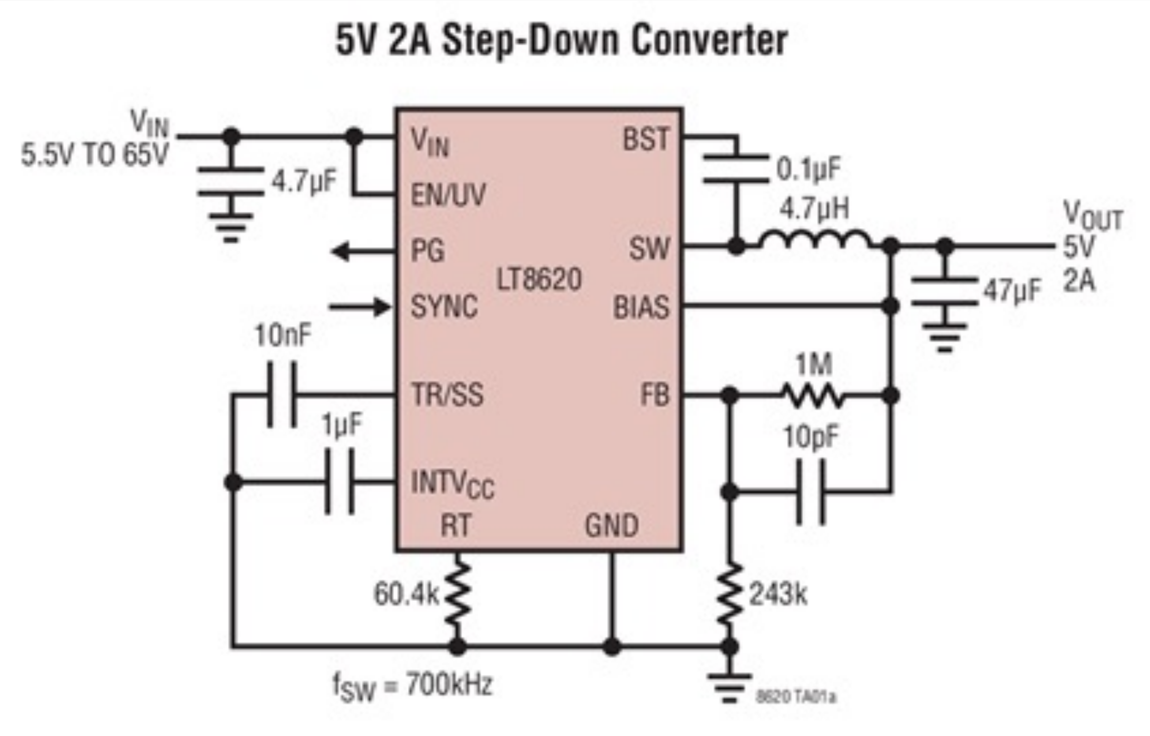


Schema bloc LM2679



Variante

- http://www.ti.com/product/lm2679
- http://www.linear.com/product/LT8620

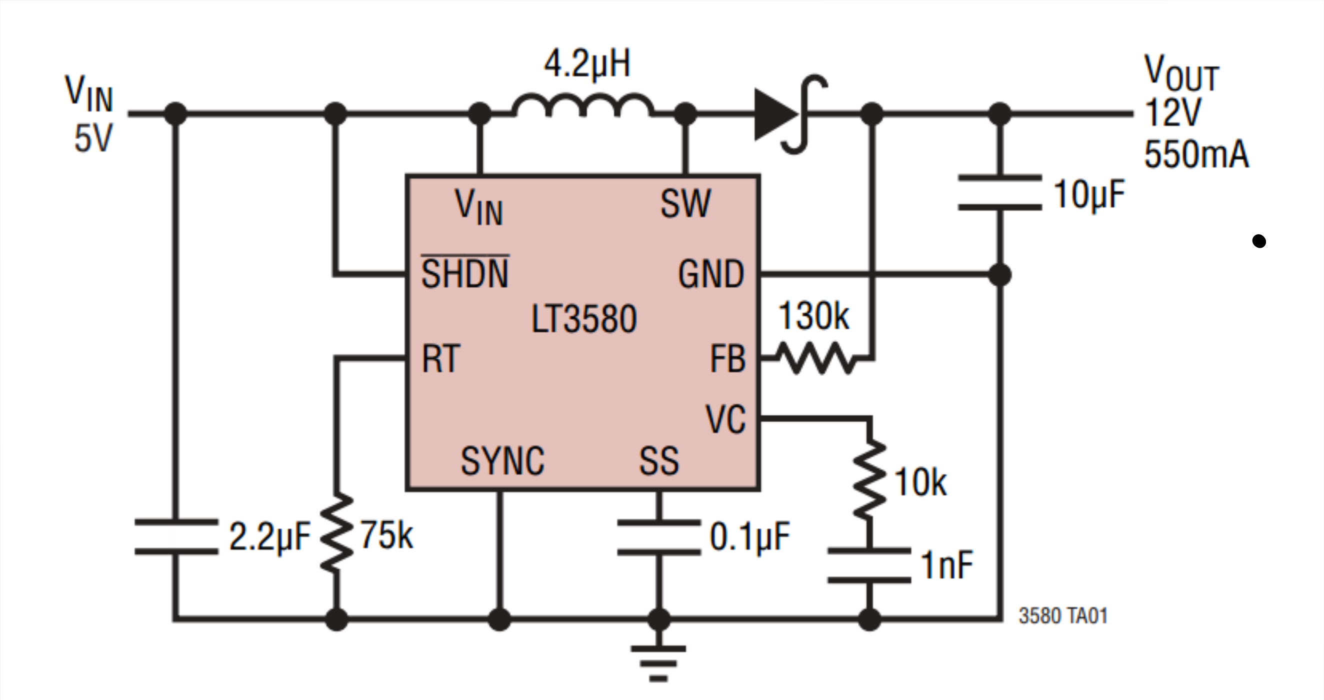


Exemple module “ready to buy”

- https://www.optimusdigital.ro/ro/surse-coboratoare-reglabile/12258-modul-dc-dc-step-down-lm2596-albastru-intrare-3-35-v-smd.html
- https://www.optimusdigital.ro/ro/surse-coboratoare-reglabile/150-modul-dc-dc-step-down-lm296.html
- https://ro.farnell.com/texas-instruments/pth08t241wad/dc-dc-converter-1-o-p-0-69-5-5v/dp/3009735?st=dc-dc
- https://ro.farnell.com/cui/vxo7803-500-m-tr/dc-dc-converter-fixed-3-3v/dp/3595427?st=dc-dc
- https://ro.farnell.com/tracopower/tsr-1-2433/converter-dc-to-dc-3-3v-1a-sip/dp/1696319


Boost – LT3580

- http://cds.linear.com/docs/en/datasheet/3580fg.pdf

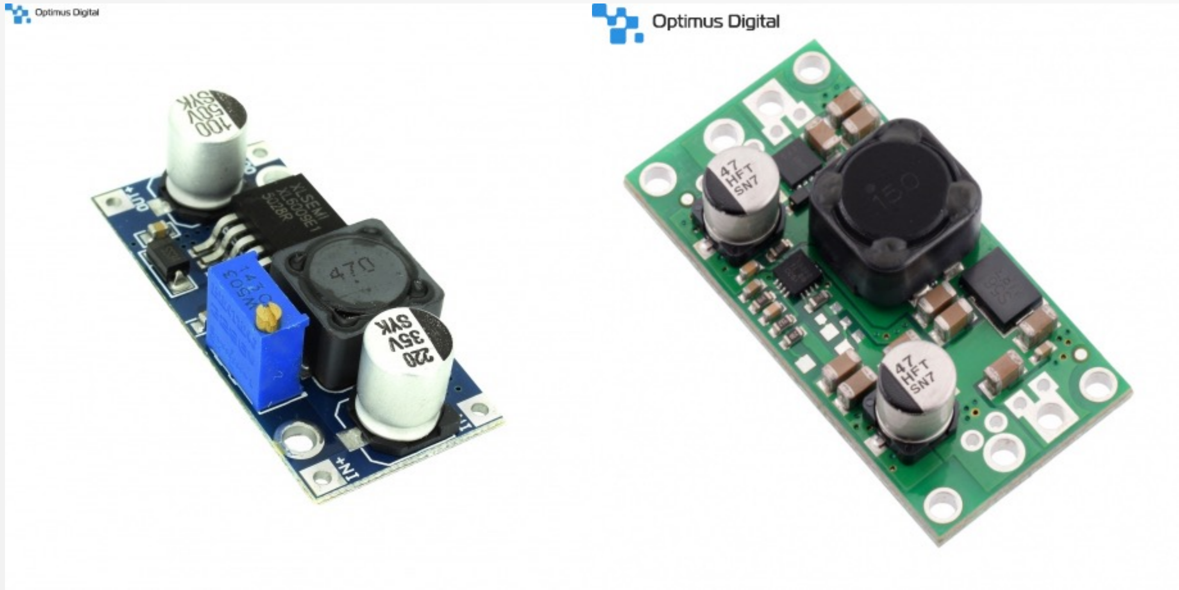


Exemple module “ready to buy”

- https://www.optimusdigital.ro/ro/surse-ridicatoare-reglabile/160-modul-dc-dc-boost-xl6009.html

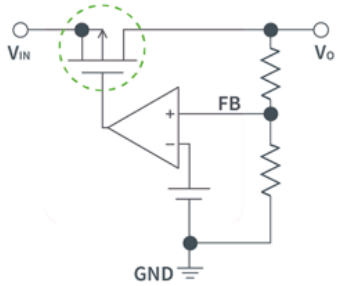
Buck-boost:

- https://www.optimusdigital.ro/ro/surse-ridicatoare-coboratoare/8275-sursa-coboratoareridicatoare-de-tensiune-de-24v-cu-regulator-s18v20f24.html



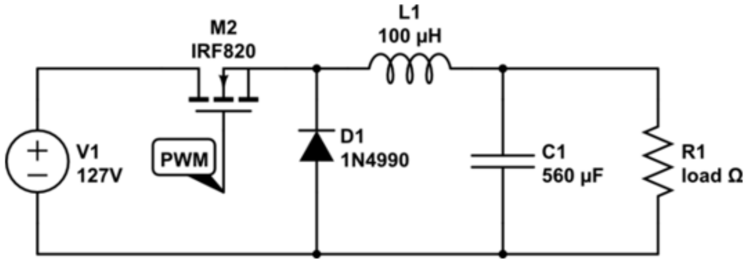
Comparație

Linear Regulators



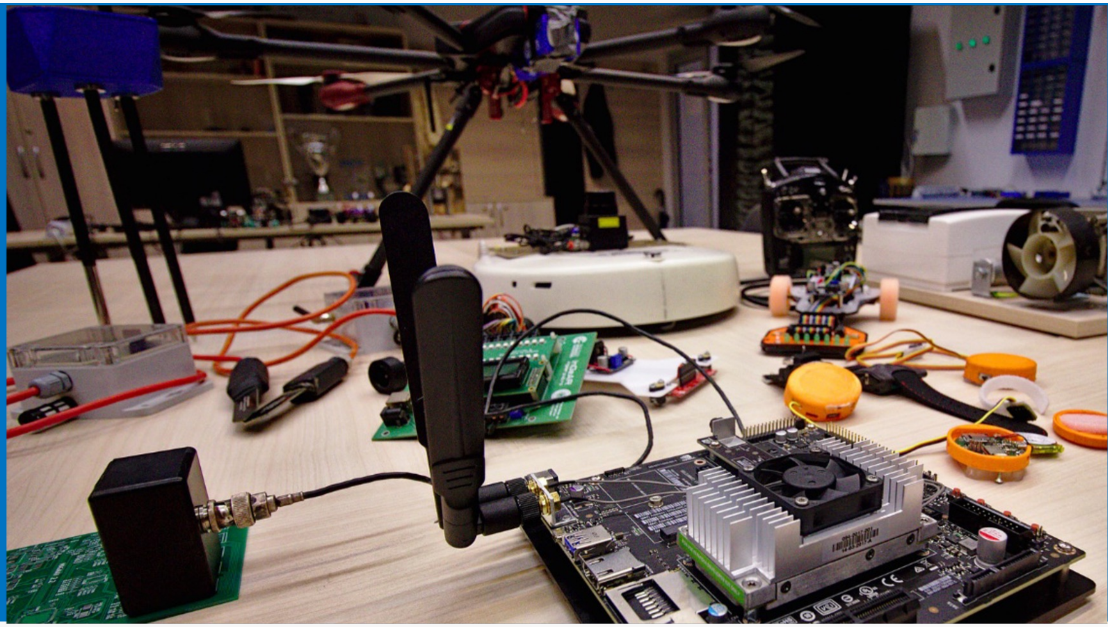
- Low Cost (low power DC applications)
- Simple to design
- High PSRR (low noise)
- Low efficiency if $V_O \ll V_I$
- Bigger heat losses than SMPS

Switch Mode Power Supplies



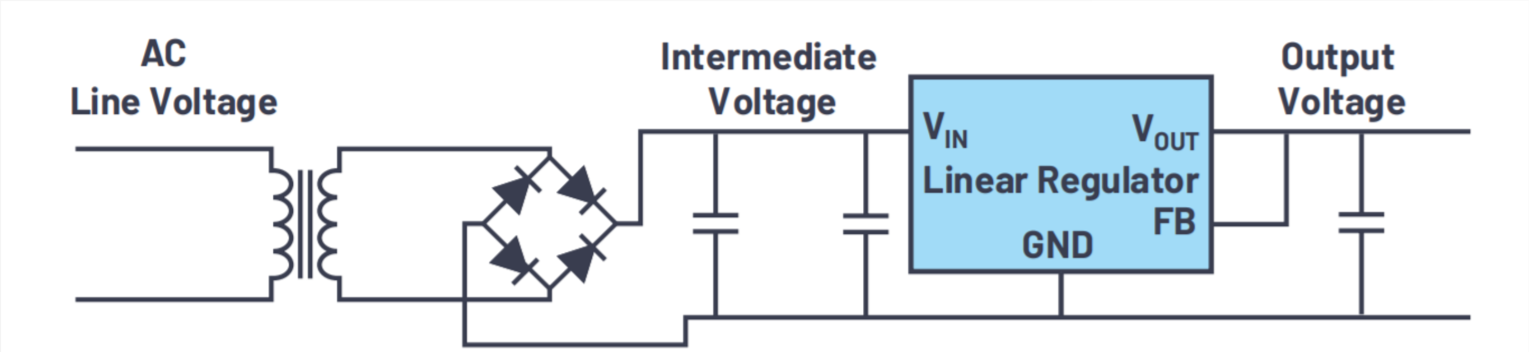
- Higher efficiency
- More W/cm^3
- Lower cost than linear supplies when used in AC-DC conversion (especially for high power)
- Source of EMI and can cause EMC problems
- Complex design (requires skilled engineer)
- Slower transient response

AC mains -> low DC



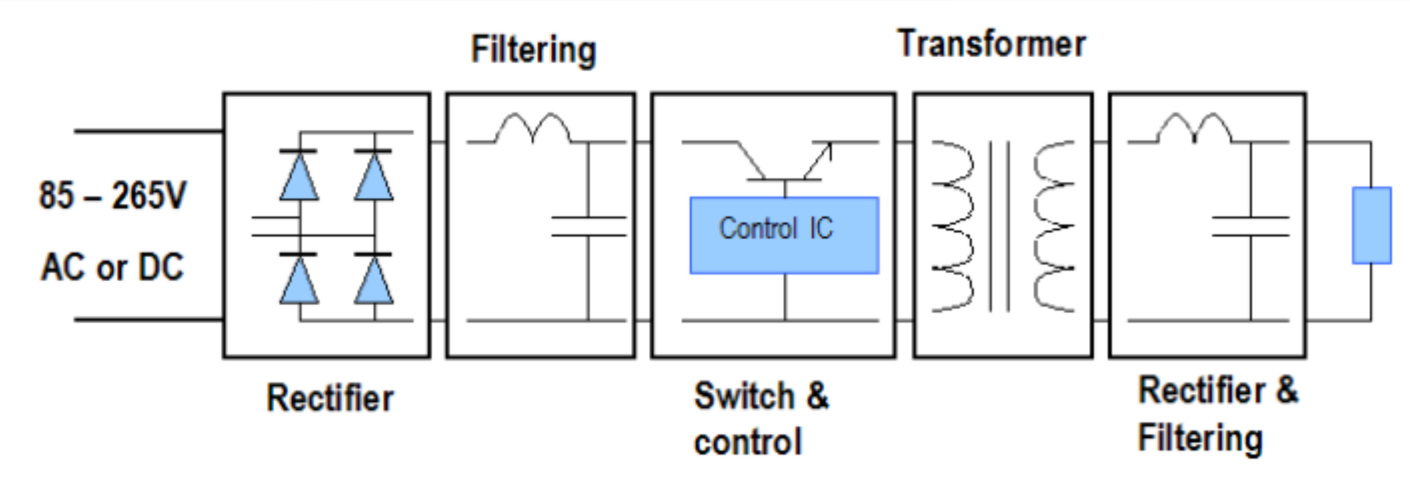
PM

Surse AC – DC liniare



Surse AC – DC în comutație

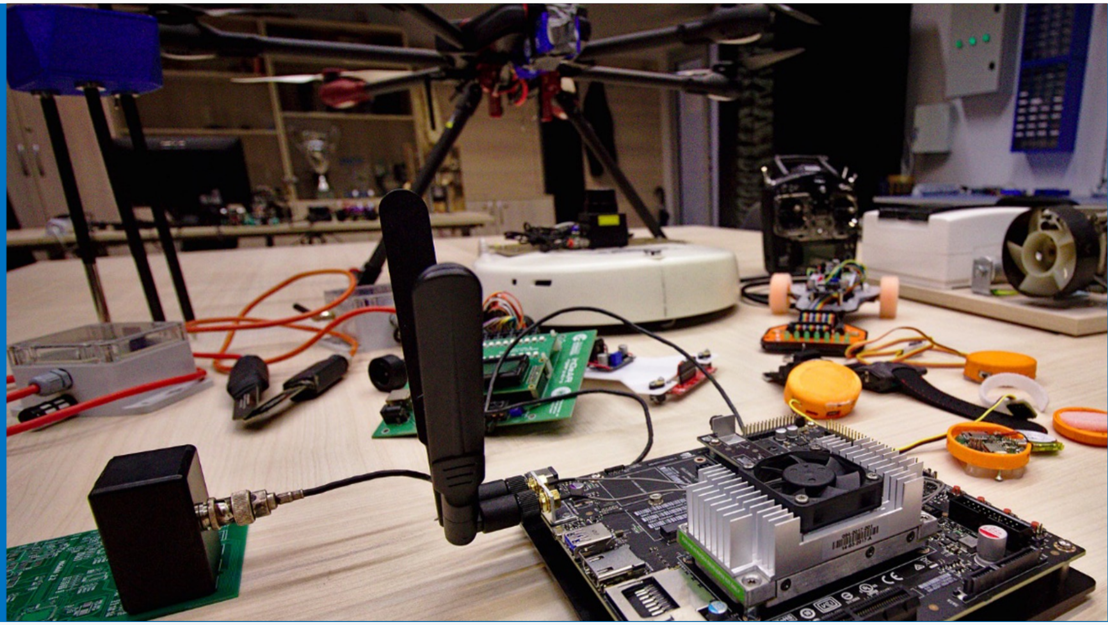
- http://ro.farnell.com/myrra/47152/power-supply-4-5w-5vdc-reg/dp/1825779



Example surse AC-DC

- https://ro.farnell.com/tdk-lambda/cus100me12-a/power-supply-medical-ac-dc-12v/dp/2913441
- https://ro.farnell.com/stontronics/pd-100-12/power-supply-enclosed-12v-100w/dp/3375267
- https://ro.farnell.com/multicomp-pro/mp003282/power-supply-ac-dc-24v-3a/dp/3408561

Diverse



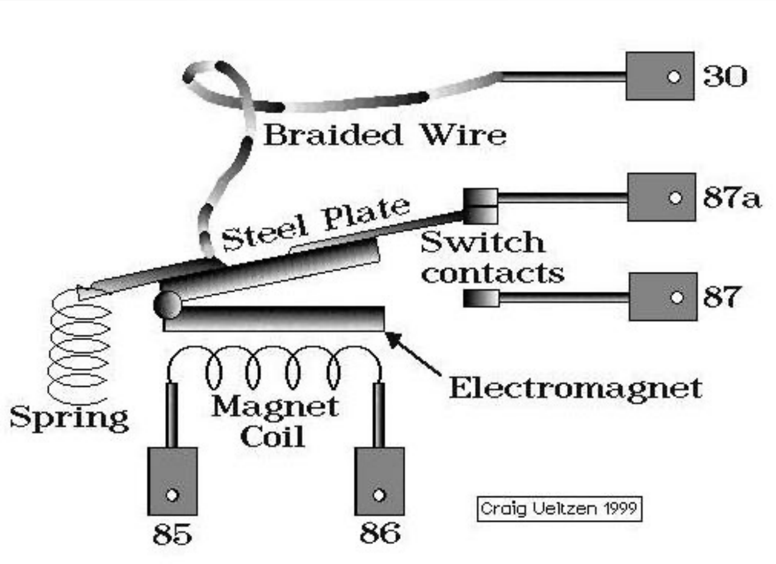
PM

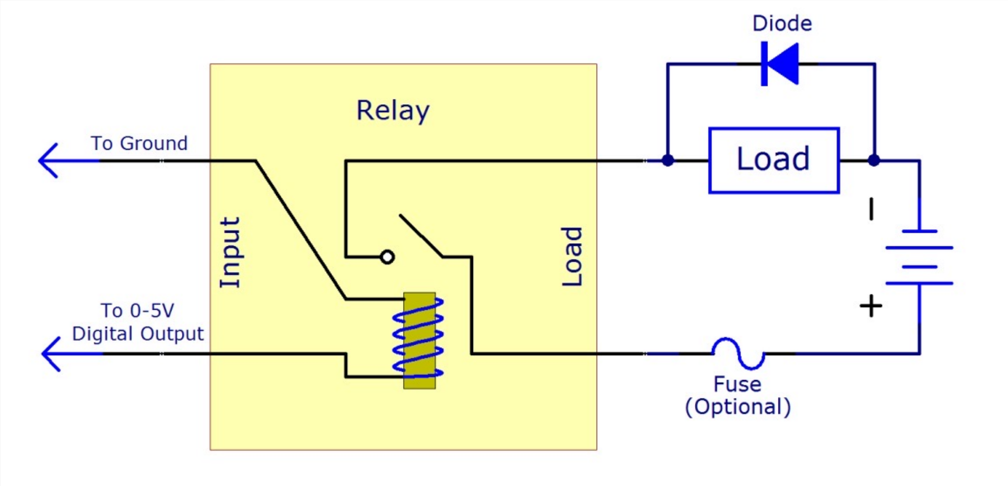
Releul

- Element electromecanic

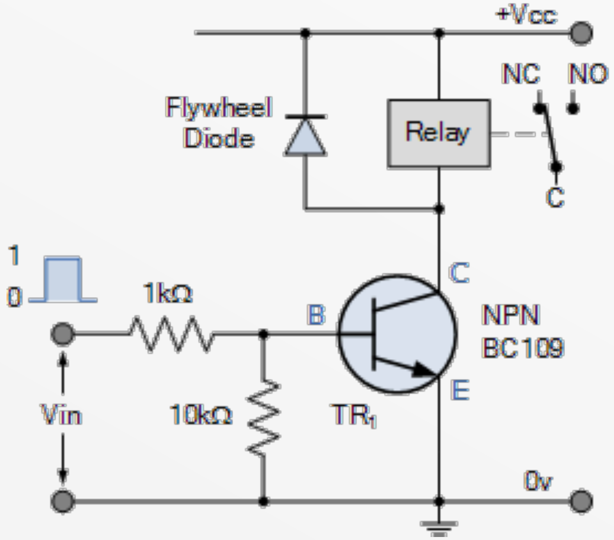
- Când un curent electric străbate bobina, aceasta atrage o lamelă metalică, care realizează un contact electic

- Când curentul ce străbate bobina dispare, lamela metalică este readusă de către un arc în poziția inițială

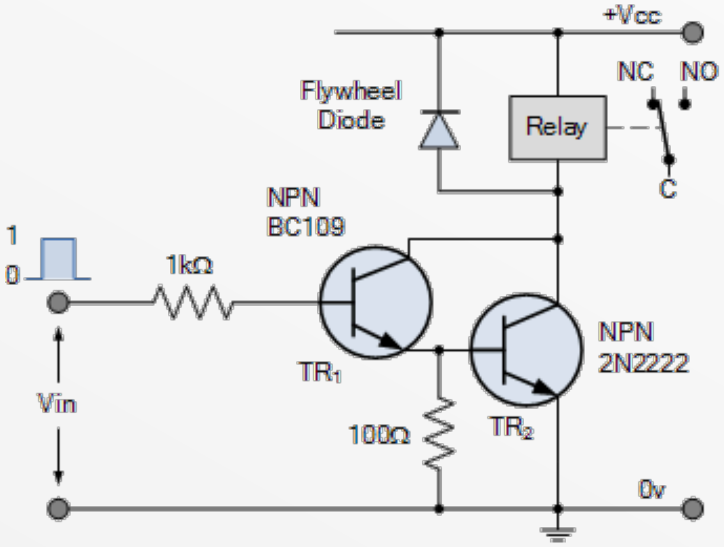




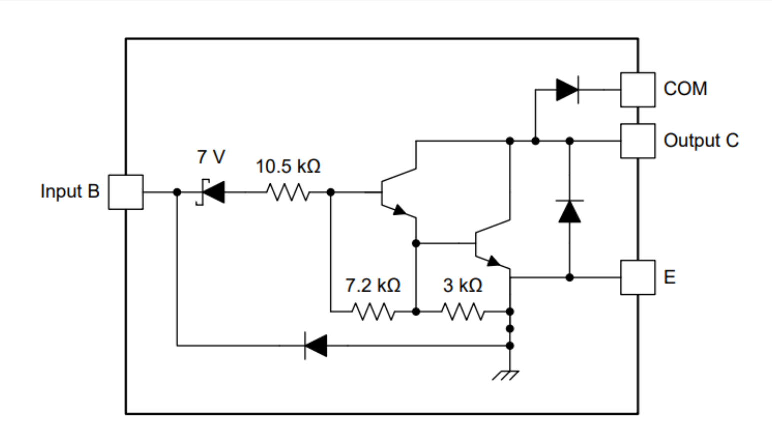
NPN

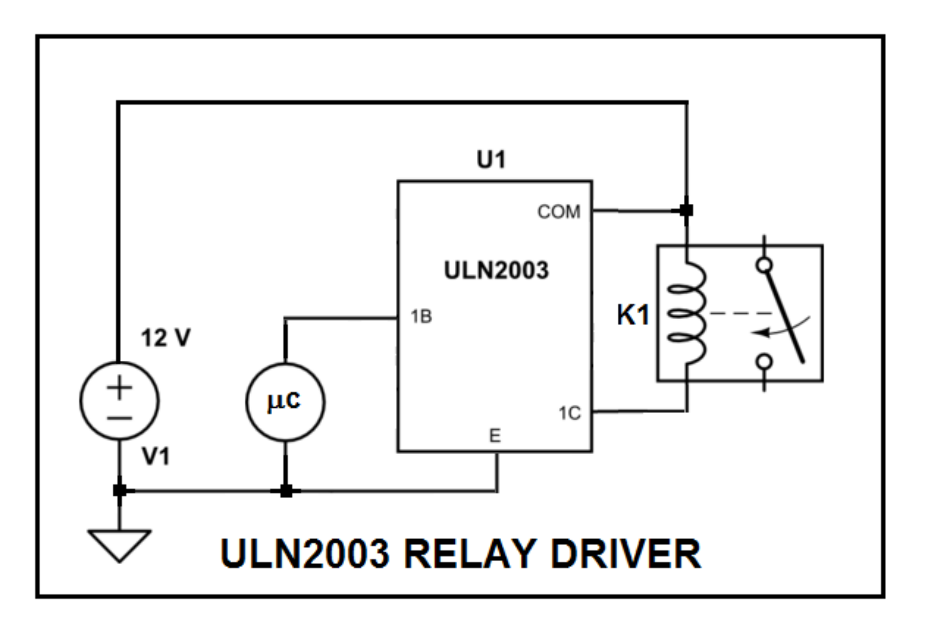

+Vcc
NC
NO
Flywheel
Diode
Relay
C
1
0
1kΩ
B
C
NPN
BC109
E
Vin
10kΩ
TR1
0v

Darlington NPN



ULN 2003

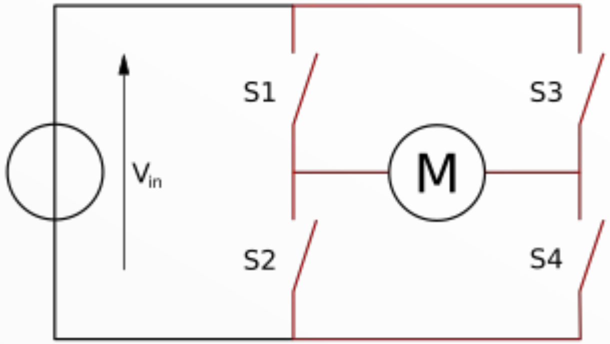




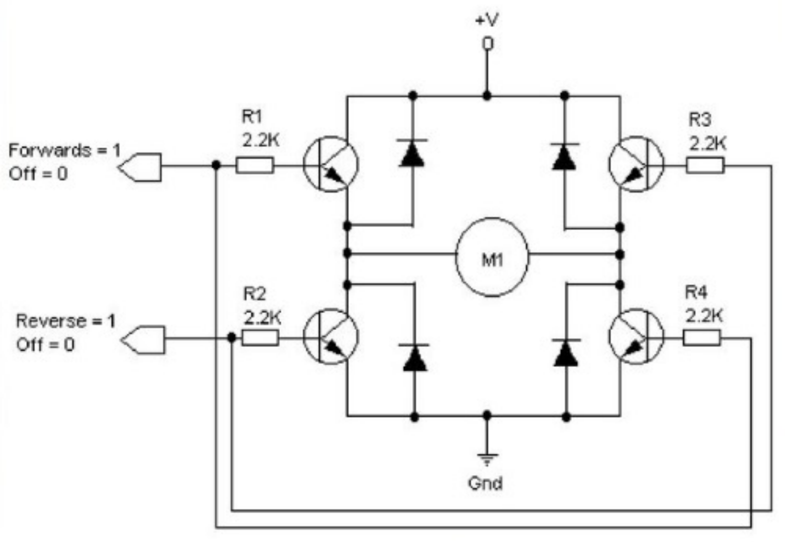
© *Texas Instruments ULN2003 DS*

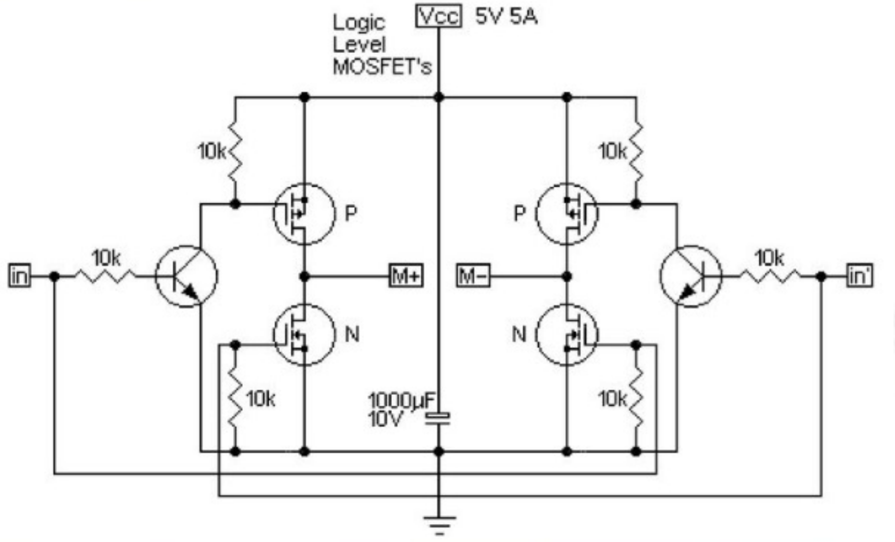
© https://electronics.stackexchange.com/

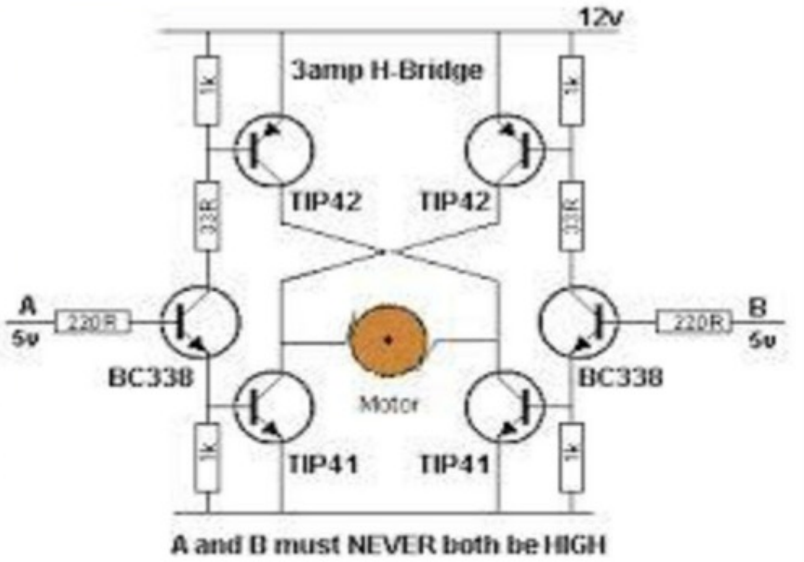
“Drivere motoare”



- Puntea H
- Comută polaritatea unei tensiuni aplicată sarcinii
- Funcționare înainte și înapoi
- Exemple de implementare

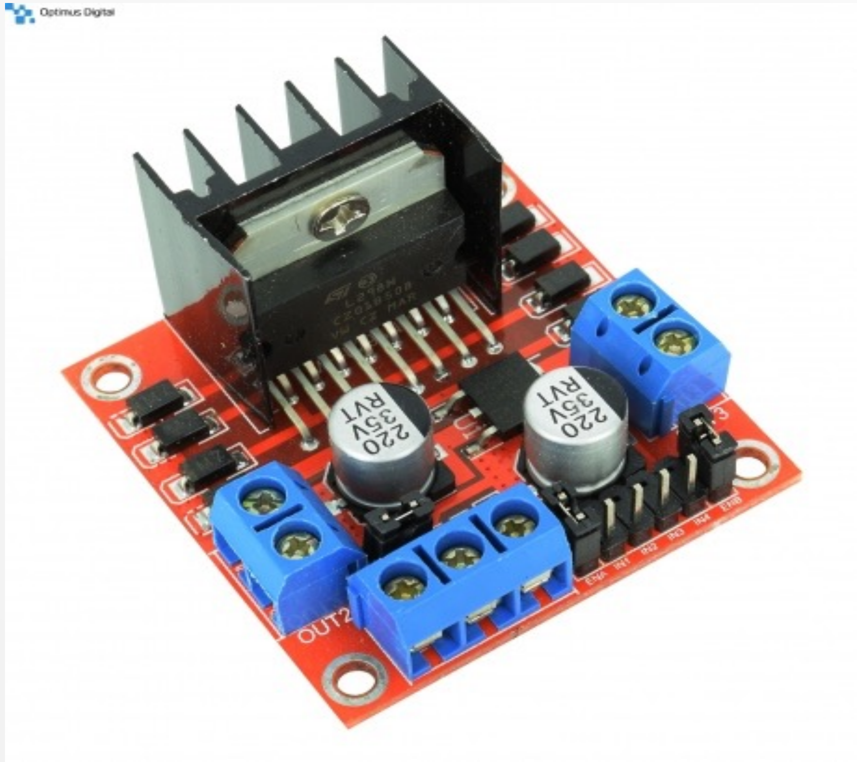






Brushed Motor Drivers

- https://www.optimusdigital.ro/ro/drivere-de-motoare-cu-perii/145-driver-de-motoare-dual-l298n.html?search_query=l298&results=5

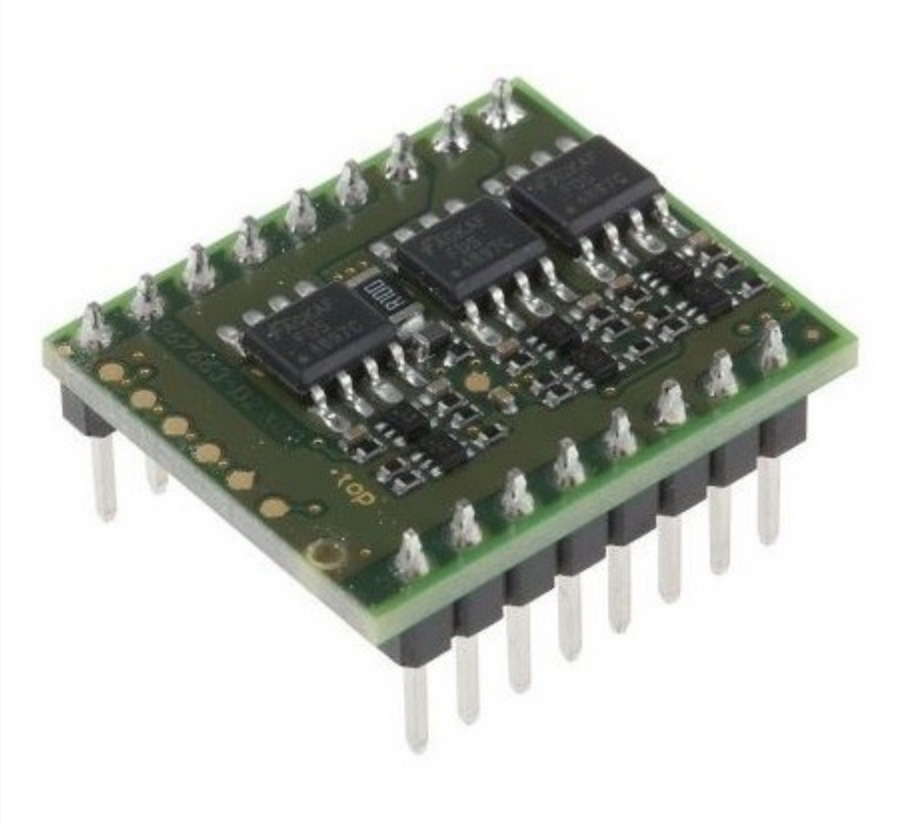


Stepper Motor Drivers

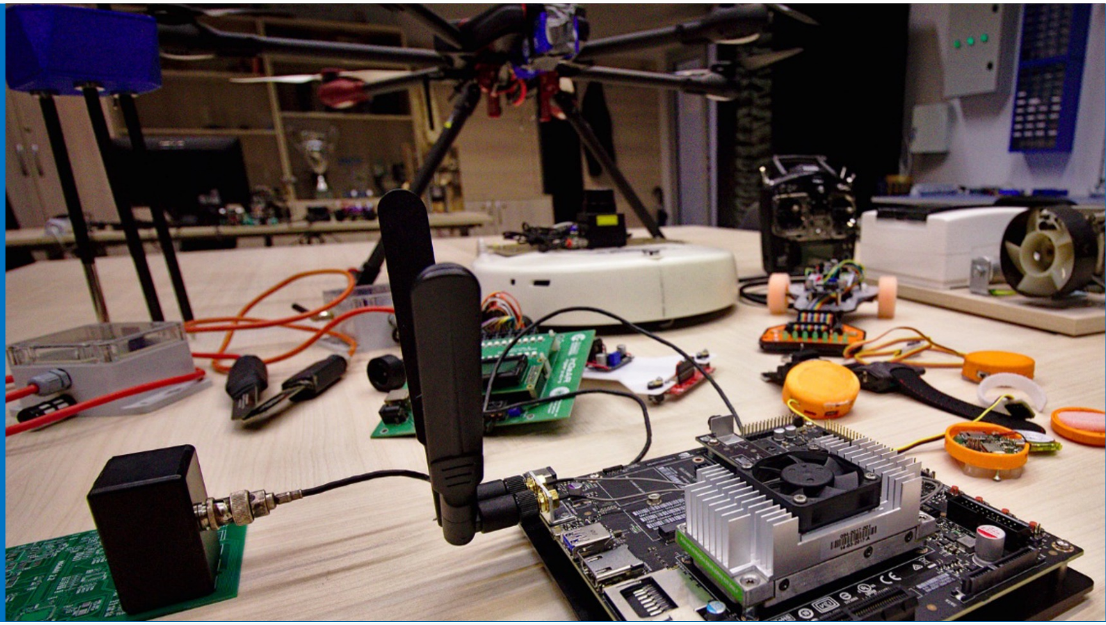
- https://www.sigmanortec.ro/Driver-motor-pas-cu-pas-ULN2003-p126284051

Brushless Motor Drivers

- https://cleste.ro/driver-motor-dc-brushless-maxon-motor.html?utm_medium=GoogleAds&utm_campaign=&utm_source=



Consum de energie



PM

Consumul de energie

- În tehnologia CMOS: P ~ $V_{DD}^2 \cdot f$
  - Tranzițiile costă
- Ce tranziții avem când în cod avem while(1); ?
  - Încă se execută instrucțiuni, PC se modifică etc.
  - Toate timerele active numără
  - ADC-ul poate să fie activ
  - Seriala poate transmite/recepționa
  - Circuitele de input GPIO sunt active
- Power management - procedeul prin care restricționăm ce poate face procesorul

ATMega328P Power Management

- Mai multe stări de power management
  - Mai puține periferice
  - Mai puține ceasuri active
  - Mai puține metode de revenire
- Periferice dezactivabile manual
  - PRR - Power Reduction Register
    - Un bit pt fiecare periferic: ADC, USART0, SPI, Timer/Counter1, Timer/Counter0, Timer/Counter2, TWI(I^2C)
    - One reserved bit (bit 4)

Stări de lucru

Table 9-1. Active Clock Domains and Wake-up Sources in the Different Sleep Modes.

Sleep Mode	Active Clock Domains					Oscillators		Wake-up Sources							Software BOD Disable
	clk_{CPU}	clk_{FLASH}	clk_{IO}	clk_{ADC}	clk_{ASY}	Main Clock Source Enabled	Timer Oscillator Enabled	INT1, INT0 and Pin Change	TWI Address Match	Timer2	SPM/EEPROM Ready	ADC	WDT	Other/O	
Idle			X	X	X	X	X[(2)]	X	X	X	X	X	X	X	
ADC noise Reduction				X	X	X	X[(2)]	X[(3)]	X	X[(2)]	X	X	X		
Power-down								X[(3)]	X				X		X
Power-save					X		X[(2)]	X[(3)]	X	X			X		X
Standby[(1)]						X		X[(3)]	X				X		X
Extended Standby					X[(2)]	X	X[(2)]	X[(3)]	X	X			X		X

Notes:
1. Only recommended with external crystal or resonator selected as clock source.
2. If Timer/Counter2 is running in asynchronous mode.
3. For INT1 and INT0, only level interrupt.

Stări de lucru - consum

28.3 DC Characteristics

T_A = –40°C to +125°C, V_{CC} = 2.7V to 5.5V (unless otherwise noted)

Parameter	Condition	Symbol	Min.	Typ.[2]	Max.	Units
Power supply current[1]	Active 4MHz, V_{CC} = 3V	I_{CC}		1.5	2.4	mA
	Active 8MHz, V_{CC} = 5V			5.2	10	mA
	Active 16MHz, V_{CC} = 5V			9.2	14	mA
	Idle 4MHz, V_{CC} = 3V			0.25	0.6	mA
	Idle 8MHz, V_{CC} = 5V			1.0	1.6	mA
	Idle 16MHz, V_{CC} = 5V			1.9	2.8	mA
Power-down mode[3]	WDT enabled, V_{CC} = 3V				44	µA
	WDT enabled, V_{CC} = 5V				66	µA
	WDT disabled, V_{CC} = 3V				40	µA
	WDT disabled, V_{CC} = 5V				60	µA

Notes:
1. Values with Section 9.10 “Minimizing Power Consumption” on page 36 enabled (0xFF).
2. Typical values at 25°C.
3. The current consumption values include input leakage current.

Registre

- SMCR - registru control sleep
  - biți SM - modul în care să intre
  - SE - sleep enable (se poate intra in sleep)
- *sleep* – instrucțiunea care intră în sleep efectiv
- Cu ajutor / Fără ajutor

```
void main ()
{
  while(1)
  {
    do_stuff();
    // ...

    set_sleep_mode(SLEEP_MODE_IDLE);
    sleep_mode();
  }
}
```

```
void main ()
{
  while(1)
  {
    do_stuff();
    // ...

    SMCR = (1 << SE); // pentru bitii SM2..0 — 0 este IDLE
    __asm__ __volatile__ ( "sleep" "\n\t" :: );
    SMCR = 0;
  }
}
```

Exemplu aplicații

- Un termometru cu afișaj digital
- O dată la 1s măsoară și schimbă afișajul
- Afișaj conectat prin SPI foarte eficient
- Cum minimizăm consumul microcontroller-ului?

Datele problemei

- Presupunem un consum mediu de 100μA pentru senzor și display
- Avem la dispoziție o baterie de ceas CR2032 de 200mAh și 3.3V
- Presupunem că putem folosi toată capacitatea
- Avem un cristal de cuart, de 8MHz
- ADC configurat inițial cu prescaler maxim 128

$$t_{conversie} = 13 \text{ cicliADC} \cdot 128 \cdot t_{cicluCPU}$$
$$t_{conversie} = 208\text{μs} \simeq 200\text{μs}$$

- Consumul pe SPI presupune schimbul a 100 bytes la 4MHz.

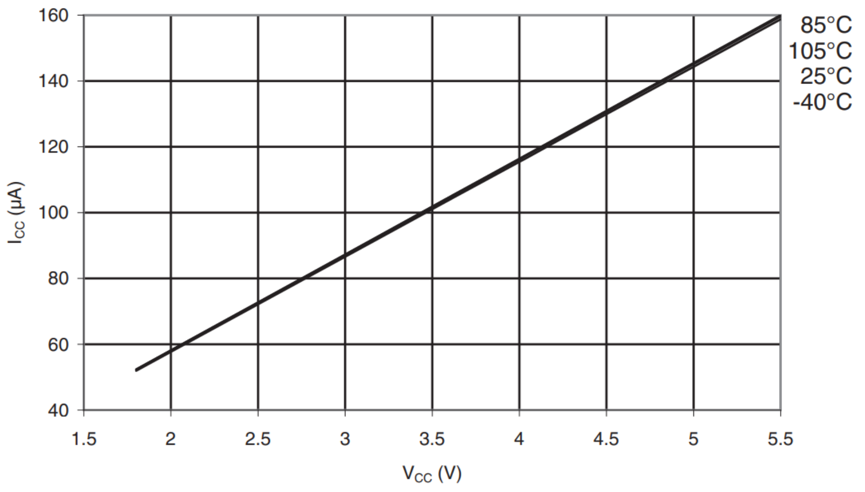
$$t_{\text{comunicare}} = \frac{100 * 8}{4000000}\text{s}$$
$$t_{\text{comunicare}} = 200\text{μs}$$

- Neglijăm restul operațiilor

Calcul consum activ

- $I_{CC_{\text{total}}} = I_{CC_{\text{Activ,8}MHz\text{,3.3}V}}(1 + \text{I}_{\text{ADC}} + \text{I}_{\text{SPI}} + \text{I}_{\text{TIM1}})$
- $I_{CC_{\text{total}}} = 3\text{mA} \cdot (1 + 0.042 + 0.027 + 0.016)$
- $I_{CC_{\text{total}}} = 3\text{mA} \cdot (1 + 0.042 + 0.027 + 0.016) = 3.255\text{mA}$

Figure 35-40.ATmega328P: ADC Current vs. V_{CC} (AREF = AV_{CC})

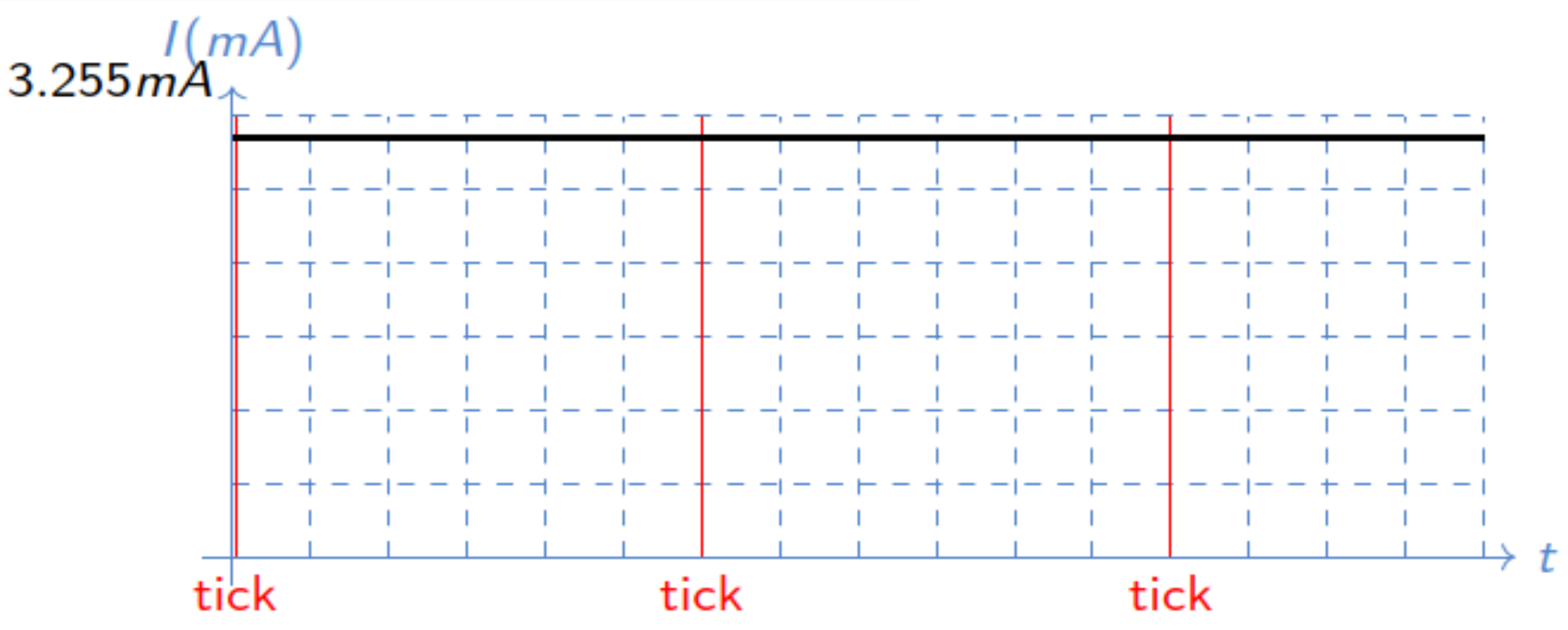


PRR bit	Additional Current consumption compared to Active with external clock (See Figure 30-1 and Figure 30-2)
PRTIM2	3.8%
PRTIM1	2.6%
PRTIM0	1.6%
PRADC	4.8%
PRSPI	2.8%

https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf

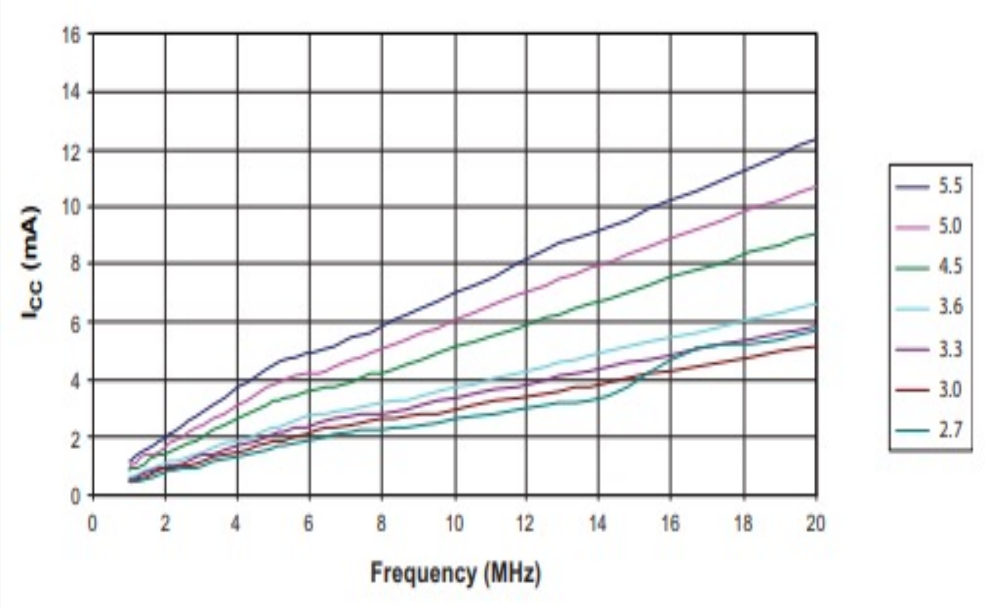
Grafic de consum

- $t_{\text{baterie}} = \frac{Capacitate}{Consum} = \frac{200}{3.26 + 0.01} = 61.1\text{h}$



Calcul consum activ optimizat

- *Dacă folosim PRR să închidem ADC și SPI când nu avem nevoie?*
- $I_{CC_{\text{adc}}} = I_{CC_{\text{Activ},8MHz,3.3V}}(1 + \text{I}_{\text{ADC}} + \text{I}_{\text{TIM1}}) = 3.174\text{mA}$
- $I_{CC_{\text{comunicatie}}} = I_{CC_{\text{Activ},8MHz,3.3V}}(1 + \text{I}_{\text{SPI}} + \text{I}_{\text{TIM1}}) = 3.129\text{mA}$
- $I_{CC_{\text{rest}}} = I_{CC_{\text{Activ},8MHz,3.3V}}(1 + \text{I}_{\text{TIM1}}) = 3.048\text{mA}$



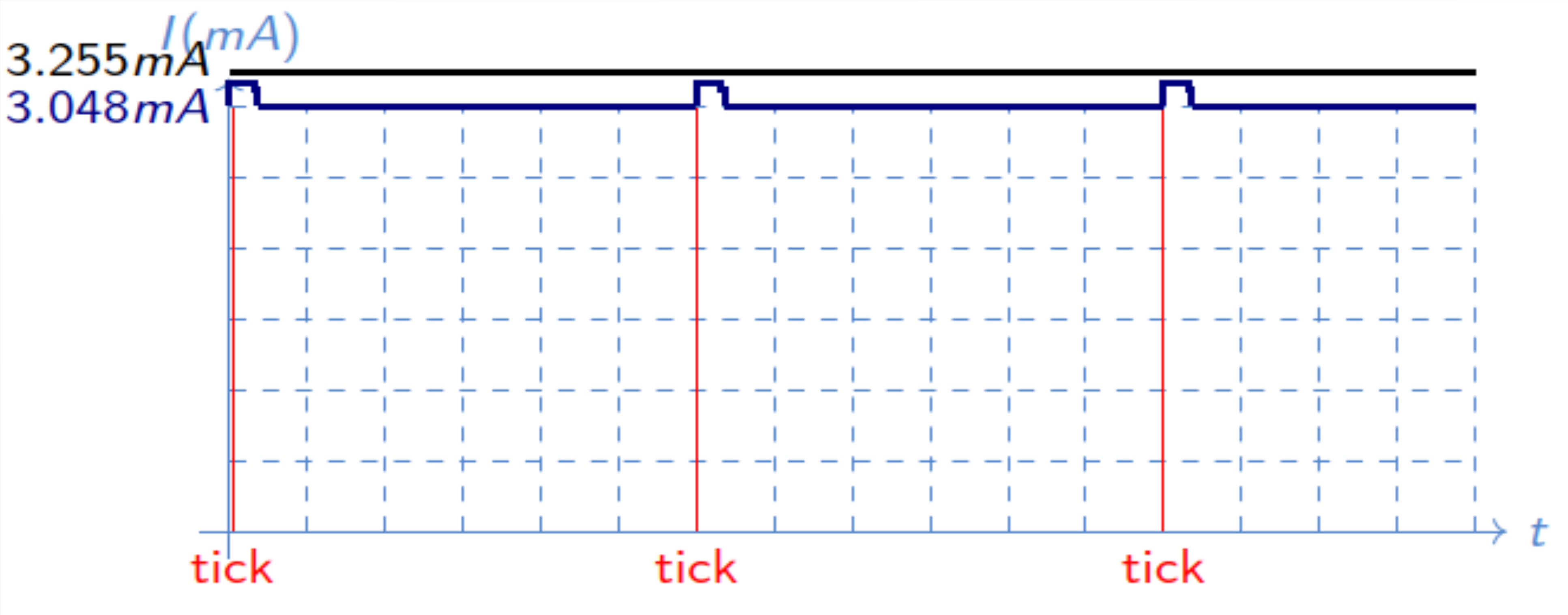
PRR bit	Additional Current consumption compared to Active with external clock (See Figure 30-1 and Figure 30-2)
PRTIM2	3.8%
PRTIM1	2.6%
PRTIM0	1.6%
PRADC	4.8%
PRSPI	2.8%

Curent mediu

- Curentul mediu este media ponderată pe timpi
- $I_{CC_{\mathrm{mediu}}} = \frac{\sum I_{CC_{\mathrm{x}}} * t_{\mathrm{x}}}{\sum t_{\mathrm{x}}}$
- $I_{CC_{\mathrm{mediu}}} = \frac{I_{CC_{conversie}} * t_{conversie} + I_{CC_{comunicatie}} * t_{comunicatie} + I_{CC_{rest}} * t_{rest}}{t_{total}}$
- $I_{CC_{\mathrm{mediu}}} = \frac{3.174 * 200 + 3.129 * 200 + 3.048 * 999600}{1000000}$
- $I_{CC_{\mathrm{mediu}}} = 3.048 \text{ mA}$

Grafic de consum optimizat

- $t_{\text{baterie}} = \frac{Capacitate}{Consum} = \frac{200\text{mAh}}{3.05\text{mAa} + 0.01\text{mA}}$ = 65.3h



Idle Mode

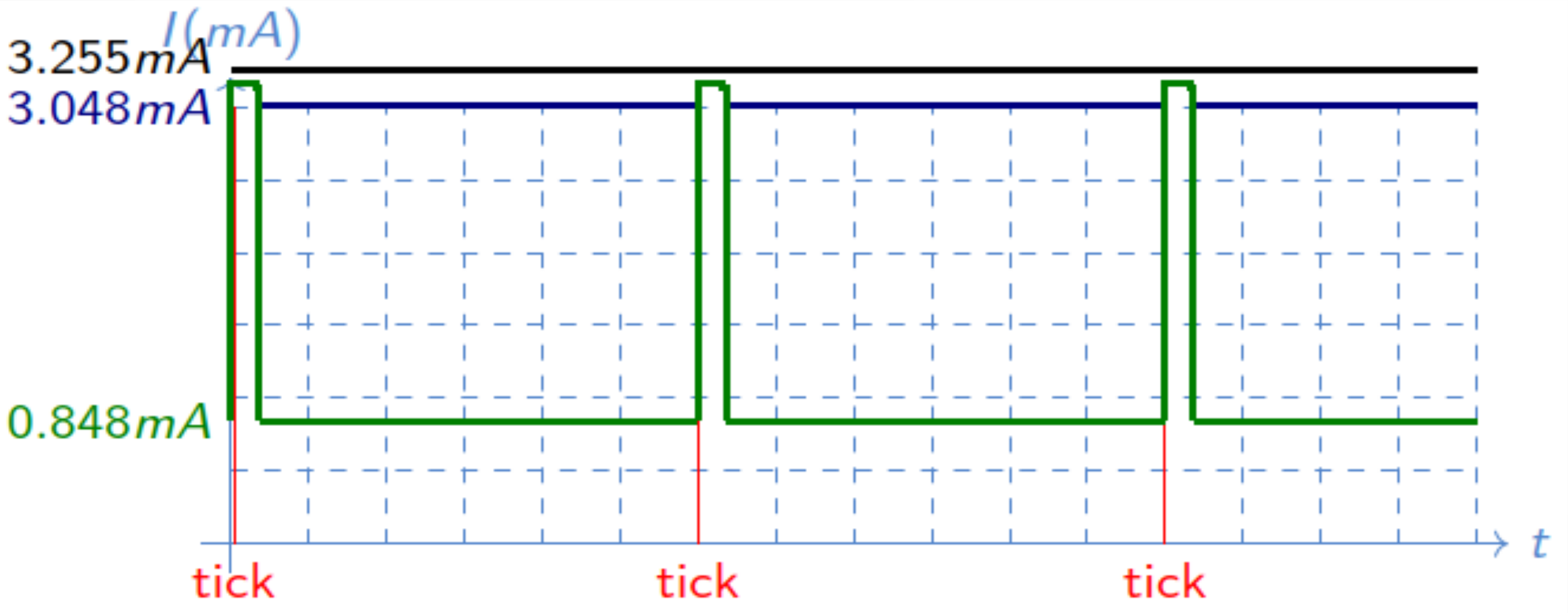
- Putem folosi Idle Mode, cu wake-up pe întreruperea de Timer1
- Putem număra maxim 4s
- Alegem întrerupere de Timer1 la 1s
- Neglijăm timpii de intrare/ieșire din Idle
- $I_{CC_{\mathrm{Idle}}} = I_{CC_{\mathrm{Idle,8M}Hz\mathrm{,3.3}V}}(1 + \mathrm{I_{TIM1,Idle}}) = 0.8 \cdot 1.06 = 0.848\mathrm{mA}$

Curent mediu Idle Mode

- $I_{CC_{\text{mediu}}} = \frac{\sum I_{CC_\text{x}} * t_\text{x}}{\sum t_\text{x}}$
- $I_{CC_{\text{mediu}}} = \frac{I_{CC_{conversie}} * t_{conversie} + I_{CC_{comunicatie}} * t_{comunicatie} + I_{CC_{\text{Idle}}} * t_{rest}}{t_{total}}$
- $I_{CC_{\text{mediu}}} = \frac{3.174 * 200 + 3.129 * 200 + 0.848 * 999600}{1000000}$
- $I_{CC_{\text{mediu}}} = 0.848 \text{ mA}$

Grafic de consum Idle Mode

- $t_{\mathrm{baterie}} = \frac{Capacitate}{Consum} = \frac{200mAh}{0.85\mathrm{mA} + 0.01\mathrm{mA}}$ = 232.56h



Alte stări de power saving

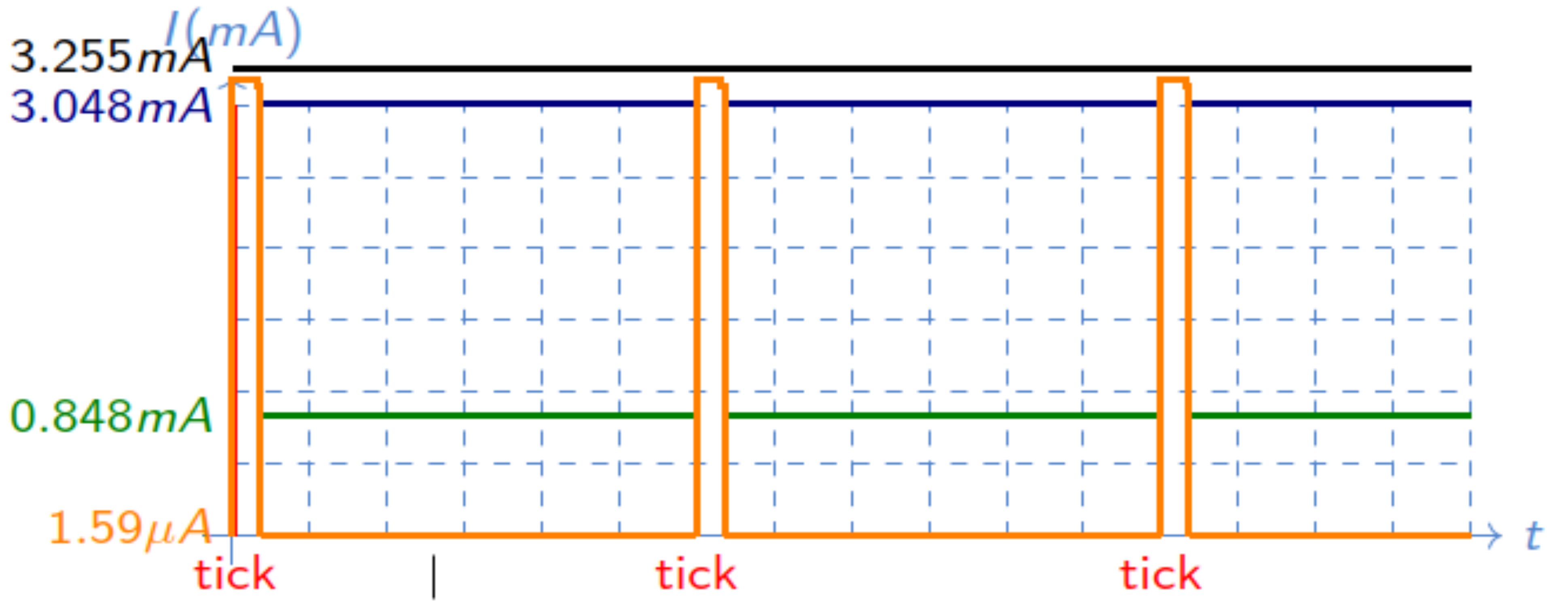
- Power-down - cel mai eficient mod de (ne)lucru
  - Poate fi trezit cu: întrerupere externă, I^2C, watchdog
  - Consum la 25°C de 0.2μA fără watchdog
  - Noi am avea nevoie de watchdog să-l trezim → 4μA
- Power-save - Adaugă timer-ul 2 dacă funcționează cu cristal propriu
  - Consum la 25°C, 3.3V de 0.75μA fără watchdog

Curent mediu Power-save Mode

- Curentul mediu este media ponderată pe timpi
- $I_{CC_{\mathrm{mediu}}} = \frac{\sum I_{CC_{\mathrm{x}}} * t_{\mathrm{x}}}{\sum t_{\mathrm{x}}}$
- $I_{CC_{\mathrm{mediu}}} = \frac{I_{CC_{conversie}} * t_{conversie} + I_{CC_{comunicatie}} * t_{comunicatie} + I_{CC_{\mathrm{Power-save}}} * t_{rest}}{t_{total}}$
- $I_{CC_{\mathrm{mediu}}} = \frac{3.174 * 200 + 3.129 * 200 + 0.00075 * 999600}{1000000}$
- $I_{CC_{\mathrm{mediu}}} = 1.59\ \mu\mathrm{A}$

Grafic de consum Power-save Mode

- $t_{\text{baterie}} = \frac{Capacitate}{Consum} = \frac{200mAh}{0.00159\text{mA} + 0.01\text{mA}}$ = 17256h ≃ 719zile
- Consumul MCU-ului este de 2047 ori mai mic



Optimizări ADC

- Senzorul oferă o scară de 10mV/1℃ , cu o precizie de 0.5 ℃
- Rezoluția ADC-ului este de 10 biți, la 3.3V → 3.22mV
  - Nu putem reduce rezoluția ADC-ului
  - Frecvența ADC-ului trebuie să fie între 50 și 200kHz
  - Cu PS 128 la 8Mhz este la 62.5kHz → putem reduce prescaler-ul de 2 ori
  - Pentru rezoluție mai mică, puteam reduce la PS 16 sau 32

Curent mediu cu ADC rapid

- Am redus prescalar – ul de la 128 la 64

$$t_{conversie} = 13 \text{ cicliADC} \cdot 64 \cdot t_{cicluCPU}$$

$$t_{conversie} = 104\mu s \simeq 100\mu s$$

- Curentul mediu este media ponderată pe timpi
- $I_{CC_{\text{mediu}}} = \frac{\sum I_{CC_{\text{x}}} * t_{\text{x}}}{\sum t_{\text{x}}}$
- $I_{CC_{\text{mediu}}} = \frac{I_{CC_{conversie}} * t_{conversie} + I_{CC_{comunicatie}} * t_{comunicatie} + I_{CC_{\text{Power-save}}} * t_{rest}}{t_{total}}$
- $I_{CC_{\text{mediu}}} = \frac{3.174 * 100 + 3.129 * 200 + 0.00075 * 999700}{1000000}$
- $I_{CC_{\text{mediu}}} = 1.27\mu\text{A}$

Optimizări frecvență

- Dar dacă am face totul mai încet?
- 1MHz MCU, SPI 500kHz (minim /2 prescaler)
- ADC prescaler 8 → 125000

$$t_{conversie} = 13 \text{ cicliADC} \cdot 8 \cdot t_{cicluCPU}$$

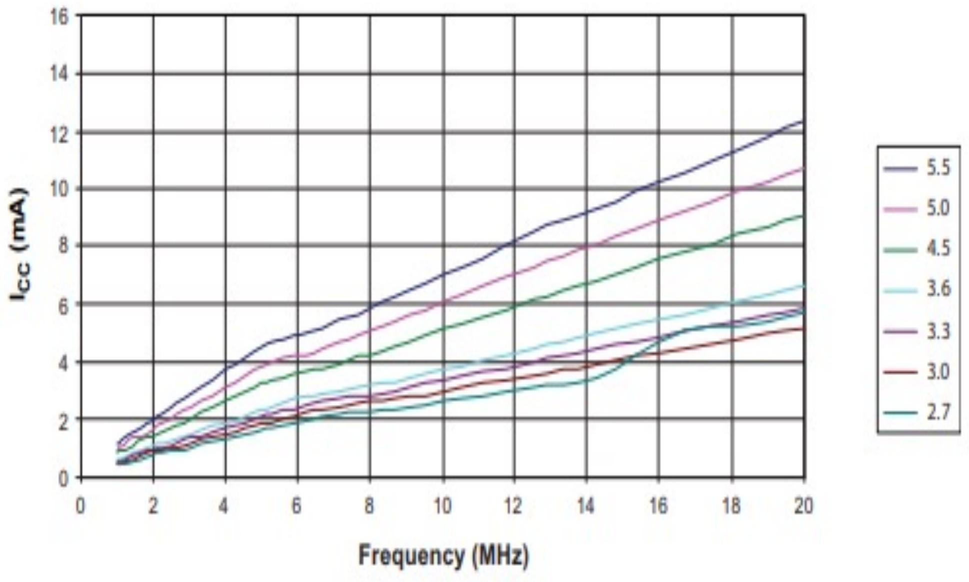
$$t_{conversie} = 104\mu s \simeq 100\mu s$$

- SPI durează mai mult

$$t_{\text{comunicare}} = \frac{100 * 8}{500000} s = 1.6ms$$

Calcul consum activ 1MHz

- $I_{CC_{\text{total}}} = I_{CC_{\text{Activ},1MHz,3.3V}}(1 + \mathrm{I}_{\text{ADC}} + I_{\text{SPI}} + \mathrm{I}_{\text{TIM1}})$
- $I_{CC_{\text{total}}} = 0.55\text{mA} \cdot (1 + 0.042 + 0.027 + 0.016) = 0.597\text{mA}$
- $I_{CC_{\text{conversie}}} = I_{CC_{\text{Activ},1MHz,3.3V}}(1 + I_{\text{ADC}} + \mathrm{I}_{\text{TIM1}}) = 0.588\text{mA}$
- $I_{CC_{\text{comunicare}}} = I_{CC_{\text{Activ},1MHz,3.3V}}(1 + \mathrm{I}_{\text{SPI}} + \mathrm{I}_{\text{TIM1}}) = 0.573\text{mA}$



PRR bit	Additional Current consumption compared to Active with external clock (See Figure 30-1 and Figure 30-2)
PRTIM2	2.5%
PRTIM1	1.6%
PRTIM0	0.7%
PRADC	4.2%
PRSPI	2.7%

Curent mediu cu 1MHz

- Curentul mediu este media ponderată pe timpi
- $I_{CC_{\mathrm{mediu}}} = \frac{\sum I_{CC_{\mathrm{x}}} * t_{\mathrm{x}}}{\sum t_{\mathrm{x}}}$
- $I_{CC_{\mathrm{mediu}}} = \frac{I_{CC_{conversie}} * t_{conversie} + I_{CC_{comunicatie}} * t_{comunicatie} + I_{CC_{rest}} * t_{rest}}{t_{total}}$
- $I_{CC_{\mathrm{mediu}}} = \frac{0.588 * 100 + 0.573 * 1600 + 0.00075 * 998300}{1000000}$
- $I_{CC_{\mathrm{mediu}}} = 0.98\ \mu\mathrm{A}$
- Consumul mediu al MCU-ului este de 3300 ori mai mic
- *Q: cat consuma celelalte componente?*

Alte optimizări

- Estimarea întregului cod (am ignorat mult)
- Scăderea tensiunii de alimentare (altă baterie?)
- Modelarea consumului pentru nivele de baterie diferite
- Schimbarea cerințelor pentru optimizări în plus
    - Trebuie neapărat la 1s pentru temperatură?
    - Trebuie neapărat făcut update la ecran dacă nu s-a schimbat nimic?
- Optimizări pe părțile ignorate (senzor, LCD)

Concluzii

- Cele mai mari economii - Idle sau Power-save
- Premisa - avem deja închise toate perifericele nefolosite
- Premisa2 - Avem un timp de repaus mare între procesări
- ATMega328P în general e mai eficient per MIPS la frecvențe mai mari pentru aceeași tensiune
  - Depinde și de ce alte operații mai facem
  - Degeaba reducem frecvența dacă nu reducem și tensiunea

OBS – limitari practice!