

## *10.2. Sisteme de fișiere orientate memorie*

În sistemele embedded este obișnuită accesarea unei zone de memorie printre interfața unui sistem de fișiere. Sistemul de operare Linux oferă mai multe căi de realizare a acestui lucru. Cel mai des întâlnit este reprezentat de transformarea unei zone de memorie într-un dispozitiv bloc și apoi montarea peste acesta a unui sistem normal de fișiere (de exemplu *ext2*). Acest lucru se realizează activând opțiunea 'RAM disk support' (`CONFIG_BLK_DEV_RAM`), și specificând mărimea implicită a fiecărui disc RAM. Acest lucru oferă și facilitatea initializării conținutului discului RAM pe baza unei imagini a sistemului comprimat de fișiere la punerea în funcțiune a sistemului. (trebuie activată opțiunea 'Initial RAM disk (initrd) support' (`CONFIG_BLK_DEV_INITRD`) și furnizați nucleului citiva parametri pentru a ști în ce loc din memorie este plasată imaginea)

Deși este un lucru obișnuit utilizarea sistemelor de fișiere *ext2* sau *minix* peste un disc RAM, dacă nu este necesară scrierea în fișiere poate fi folosit mai eficient sistemul 'Compressed ROM file system' (`CONFIG_CRAMFS`). Fiind read-only și având câteva restricții legate de mărimea fișierelor și a sistemului de fișiere în general, sistemul CRAMFS utilizează foarte puțină memorie pentru metadatele fișierelor, ba mai mult, este capabil de compresie a datelor economisind și mai mult spațiu.

Alt mod de a păstra un sistem de fișiere în întregime în memorie este utilizarea *tmpfs* (trebuie activată opțiunea 'Virtual memory file system support' (`CONFIG_TMPFS`)). Acest mod de lucru are avantajul că sistemul de fișiere se va extinde sau restrânge în funcție de conținut, în timp ce un disc RAM are dimensiune fixă ce trebuie precizată la crearea acestuia.

### *10.2.1 Flash File Systems*

Utilizarea memoriei Flash în sistemele embedded permite modificarea pe teren a softului. Memoria Flash poate fi folosită și pentru susținerea unui sistem de fișiere, totuși cu anumite restricții. Dispozitivele flash se caracterizează printr-o viteză de scriere multumitoare, dar datele nu pot fi rescrise înainte ca celulele de memorie să fie sterse, iar memoriile Flash permit doar ștergerea unor blocuri mari, operație mare consumatoare de timp.

Cel mai simplu mod de a folosi memoria Flash ca suport pentru un sistem de fișiere este utilizarea sa pentru stocarea imaginii sistemului de fișiere, imagine care va fi accesată direct (utilizând dispozitivul Memory Technology Device - MTD pseudo block device) sau prin copierea într-un disc RAM. Dezavantajele acestor abordări este că sistemul de fișiere este read-only (prima variantă) sau read/write dar volatilă (în cazul versiunii cu disc RAM). În multe situații acest aspect este un avantaj – dacă apar date corupte în sistemul de fișiere prin resetarea sistemului se revine la starea corectă. Tehnica de mai sus nu mai poate fi folosită însă dacă sistemul de fișiere trebuie să fie actualizat pe teren.

Cel mai bun mod de a avea în memoria Flash un sistem de fișiere cu capabilități de scriere într-un sistem embedded Linux este de a utiliza Journaling Flash File System *JFFS2*. Acest sistem



UNIUNEA EUROPEANĂ

MINISTERUL MUNCII, FAMILIEI ȘI  
PROTECȚIEI SOCIALE  
AMFOSDRUFONDUL SOCIAL EUROPEAN  
POSDRU  
2007-2013INSTRUMENTE STRUCTURALE  
2007-2013

**Structuri hardware si algoritmi specifici microsistemelor EMBEDDED**

de fisiere utilizeaza tehnici de jurnalizare similare celor folosite de sisteme de fisiere de inalta performanta ca *ext3*. Intrucit jurnalizarea adauga toate modificarile la log-ul sistemului de fisiere, nu este necesara actualizarea datelor in locul in care sint plasate, ceea ce este ideal pentru Flash. JFFS2 este de asemenea capabil sa comprime datele, utilizind astfel foarte eficient spatiul disponibil.

JFFS2 utilizeaza interfata MTD Flash in loc sa se plaseze peste interfata standard a dispozitivelor cu acces bloc. Acest lucru da vizibilitate caracteristicilor Flash si permite o operare mai eficienta a memoriilor Flash, de exemplu prin abilitatea de a scrie la o granularitate mai fina decit un bloc intreg de memorie si prin efectuarea procesului de 'wear leveling' – incercarea de a scrie in zone de memorie care au mai putine cicluri de stergere-scriere, uniformizind astfel numarul de scrieri in celulele de memorie - , ce imbunatateste performantele la scriere si prelungeste viata memoriei Flash.

Principalele dezavantaje ale utilizari sistemului de fisiere sint legate de timp. La pornire sistemul trebuie sa scaneze memoria Flash si sa-si cosntruiasca tabelele de stari, proces ce poate lua timp. De asemenea, sistemul trebuie sa stearga periodic sectoare Flash, ceea ce poate lua citeva secunde, iar in timpul acestui proces accesul la sistemul de fisiere este blocat. Acesta problema poate fi rezolvata totusi prin utilizarea unui proces sau fir de executie buffer, decuplat de partea de timp real a sistemului.

Tehnologia Flash ofera de asemenea interfata Flash Translation Layer (FTL). Aceasta este o tehnica de mapare a unui sistem de fisiere orientat bloc intr-un dispozitiv Flash. Desi pare foarte utila la prima vedere, interfata FTL nu ofera wear leveling si efectueaza mai multe cicluri de stergere incercind sa emuleze comportamentul unui disc magnetic. In plus, pot exista probleme cu patentele in anumite tari. Din aceste motive nu se recomanda in practica FTL ci numai JFFS2.

### ***10.2.2 Utilizarea combinata a diferitelor tipuri de sisteme de fisiere***

Dupa cum s-a remarcat anterior, exista un numar mare de optiuni pentru sistemul de fisiere, fiecare cu avantajele si dezavantajele proprii. Oricum, tinind cont ca este posibil sa se monteze mai multe sisteme de fisiere aproape oriunde in ierarhia directoarelor, se poate obtine o structura utilizator ce corespunde tuturor cerintelor.

Un exemplu tipic ar fi un sistem ce nu are nevoie de acces in scriere la fisiere exceptind citeva fisiere de configurare ce trebuie sa fie scrise si salvate(nevolatile) si citeva fisiere temporare ce pot fi regenerate la pornirea sistemului. In acest caz sistemul de fisiere root poate fi *CramFs* (comprimat si acces doar in citire), la care se adauga o partitie *JFFS2* de dimensiuni reduse pentru directorul `/etc/config` ce contine fisierele de configurare a sistemului si o partitie *TmpFs* montata in punctul `/tmp` pentru fisierele temporare.



### 10.2.3 Optimizarea consumului de energie la microsistemele integrate

Minimizarea consumului de curent este de o însemnată deosebită pentru sistemele alimentate de la baterii, dar este utilă a fi avută în vedere și pentru sistemele alimentate din rețeaua de curent alternativ, cel puțin din două motive:

- cu cât mai scăzut este consumul de curent, cu atât mai mică și deci mai ieftină este sursa de alimentare a aplicației
- același rationament se aplică și pentru condensatorul de filtraj care trebuie să susțină scurtele scurte scăderi ale tensiunii de alimentare aparute la comutările părții digitale

### 10.2.4 Profilul consumului unui microcontroler Low-power

Un exemplu tipic de procesor embedded de tip low power este reprezentat de familia Texas Instruments MSP430. Pentru a minimiza consumul unui sistem bazat pe un astfel de microcontroler, trebuie ca acesta să fie plasat cât mai mult posibil în modul de curent redus LPM3 (aproximativ 99.99% din timp procesorul trebuie să se afle în acest mod); numai circuitul BasicTimer1, circuitul de comandă și control al afișajului cu cristale lichide și circuitul de întreruperi sunt active, iar toate celelalte module, inclusiv unitatea CPU sunt oprite; unitatea CPU este trezită la intervale regulate programabile (de exemplu la fiecare secundă) de către circuitul BasicTimer 1. Un astfel de sistem în fiecare secundă este trezit și incrementează un contor de timp și la fiecare minut efectuează o achiziție de semnal folosind convertorul analog-digital integrat și apoi procesează rezultatul măsurătorii.

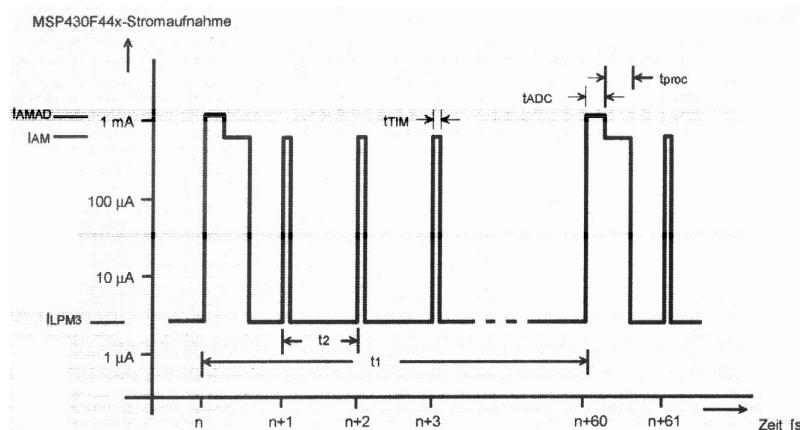


Figura 10.9. Optimizarea consumului pentru un convertor AD

Se poate calcula în condițiile date consumul de curent al sistemului, considerând următoarele măriri:

**Structuri hardware si algoritmi specifici microsystemelor EMBEDDED**

t1	intervalul intre doua achizitii succesive cu modulul ADC12 (aici 60s)
t2	intervalul de timp intre doua perioade active (1s)
tTim	timpul activ dupa trezire. Tipic este 25microsecunde pina la 1 milisecunda. Este consumat cu incrementarea contorului de timp si procesarea valorii achizitionate
tADC	timpul activ cu procesorul si circuitul ADC pornite. Tipic 3-13 microsecunde pe conversie cu ADC12
tproc 100	timpul activ cu procesorul pornit pentru procesarea masuratorii. Tipic inte 1 si milisecunde
IAM	consumul in mod activ cu circuitul ADC12 dezactivat. 0.42mA tipic
IAMAD	consumul in mod activ cu circuitul ADC12 activat. 0.42+0.8mA=1.22mA tipic
ILPM3	consumul in modul 3 low power. Tipic 1.6μA

Curentul mediu consumat din baterie de microcontrolerul MSP430 este calculabil cu formula:

$$I_{CC} = \frac{1}{t_1} \left( \frac{t_1}{t_2} \times t_{Tim} \times I_{AM} + t_{proc} \times I_{AM} + t_{ADC} \times I_{AMAD} + \left( t_1 - \frac{t_1}{t_2} \times t_{Tim} - t_{ADC} - t_{proc} \right) \times I_{LPM3} \right)$$

Formula de mai sus se poate simplifica daca timpii tTim, tADC si tproc sint mult mai mici decit t1, ceea ce este cazul normal:

$$I_{CC} \approx \frac{1}{t_2} \times t_{Tim} \times I_{AM} + \frac{1}{t_1} (t_{proc} \times I_{AM} + t_{ADC} \times I_{AMAD}) + I_{LPM3}$$

Considerind ca sistemul lucreaza cu urmatoorii parametri: AVcc=DVcc=3V, TA=25 grade Celsius, t1=60s, t2=1s, tTim=0.5ms, tADC=0.15ms, tproc=10ms, fMCLK=1MHz, doua masuratori ADC12 de cite 13μs. Intrucit , tADC si tproc sint nesemnificativi fata de t1 se poate lucra cu formula aproximativa si se obtine o valoare medie a consumului:

$$I_{CC} \approx \frac{1}{1s} \times 0.5ms \times 0.42mA + \frac{1}{60s} (10ms \times 0.42mA + 0.026ms \times 0.80mA) + 1.6\mu A$$

$$I_{CC} \approx 1.88\mu A$$

La aplicatia analizata curentul procesorului MSP430F4xx creste fata de modul low power numai cu 17.5%, practic nesemnificativ.



UNIUNEA EUROPEANĂ

MINISTERUL MUNCII, FAMILIEI ȘI  
PROTECȚIEI SOCIALE  
AMFOSDRUFONDUL SOCIAL EUROPEAN  
POSDRU  
2007-2013INSTRUMENTE STRUCTURALE  
2007-2013

**Structuri hardware si algoritmi specifici microsistemelor EMBEDDED**

Minimizarea consumului unei aplicatii bazate pe microcontrolere MSP430

Consumul total al unui sistem MSP430 alimentat de la baterie cuprinde mai multe componente:

1. Consumul procesorului MSP430
2. Curentul de descarcare propriu al bateriei
3. Consumul cuartului
4. Consumul afisorului cu cristale lichide
5. Consumul componentelor electronice externe

Fiecare din aceste componente va fi studiata in detaliu. Analizele vizeaza un sistem care lucreaza pe baza modului de consum redus LPM3, intrucit modul LPM4, desi are un consum si mai redus, nu permite masurari de timp – in acest caz este oprit si circuitul de ceas auxiliar ACLK si numai intreruperile de la porturile 1 si 2 ale microcontrolerului il mai pot trezi din acest mod.

### ***10.2.5 Consumul procesorului MSP430***

Modul de lucru de sonsum redus LPM3 trebuie sa fie modul normal de lucru al microcontrolerului MSP430. Modul activ si modul activ cu modulul ADC12 activat se utilizeaza doar atunci cind sint absolut necesare. Citeva reguli importante pentru minimizarea consumului de curent sint:

1. Modul LPM3 trebuie parasit cit mai rar posibil, de exemplu doar la fiecare 2 secunde prin intrerupere de la modulul BasicTimer1
2. Programul executat dupa trezirea din modul LPM3 trebuie sa fie cit mai scurt posibil, de exemplu incrementarea unui numarator si verificarea necesitatii efectuarii altor calcule; daca acestea nu sint necesare se reintra imediat in modul LPM3
3. Intervalele de timp intre doua perioade active – adica perioadele in care se masoara si se efectueaza calcule – trebuie sa fie cit mai lungi posibil, de exemplu 60s sau mai mult
4. Trebuie tinute active numai perifericele cu adevarat necesare si numai cind au sarcini de efectuat. De exemplu convertorul ADC trebuie tinut activ numai pe durata unei masuratori, si indata ce aceasta a fost efectuata modulul trebuie dezactivat prin resetarea bitului ADC12ON din registrul ADC12CTL0. In aceasta situatie utilizarea intreruperilor este foarte eficienta – subrutina de tratare a intreruperii asociate ADC12 opreste functionarea acestuia dupa o masurare reusita
5. Evitarea evaluarii modificarii unui prin de intrare prin tehnica polling. In locul acestei metode consumatoare de timp procesor se recomanda utilizarea facilitatii de generare de intreruperi la porturile 1 si 2 pentru a detecta modificarea starii acestora. Intrucit aceste intrari pot reactiona la ambele fronturi ale semnalului de intrare, poate fi detectata orice modificare de stare fara testari permanente ale intrarii. Aceasta facilitate simplifica mult implementarea citirii contactelor si tastelor.
6. Evitarea calculelor foarte lungi si inlocuirea lor pe cit posibil cu calculul tabelat. Cele 7 moduri de adresare ale procesorului MSP430 asigura un lucru rapid si transparent din punct de vedere al codului cu tabelele.



UNIUNEA EUROPEANĂ

MINISTERUL MUNCII, FAMILIEI ȘI  
PROTECȚIEI SOCIALE  
AMFOSDRUFONDUL SOCIAL EUROPEAN  
POSDRU  
2007-2013INSTRUMENTE STRUCTURALE  
2007-2013

**Structuri hardware si algoritmi specifici microsistemelor EMBEDDED**

7. Trebuie evitata plasarea portiunilor des folosite de cod in subrutine apelate prin instructiunea CALL pentru ca aceasta necesita un numar mare -7- de cicli MCLK pentru un apel normal ca si pentru reintoarcerea in programul apelant. In locul acestor subrutine se recomanda utilizarea in-line a codului respectiv, operatiune ce poate fi usurata prin definirea aceluia cod ca MACRO. Evident consumul de memorie de program creste, dar scade numarul de cicli CPU.
8. Buclele scurte efectuate de un numar redus de ori trebuie evitate, intrucit controlul buclei reprezinta un procent important din timpul total de prelucrare si de fapt acesta nu produce rezultate utile. In locul buclelor scurte se poate utiliza o secventa liniara de cod, obtinuta prin repetarea secventei din interiorul buclei – desfacerea buclei.
9. Pentru portiuni de cod prea lente se recomanda folosirea registrilor R4..R15 pentru stocarea variabilelor intrucit instructiunile ce utilizeaza registrii sint mai rapide decit cele ce lucreaza cu locatiile de memorie si ocupa spatiu mai redus in memoria program; in acest caz, se reduce si necesarul de memorie RAM.
10. Terminarea imediata a calculului la o operatie de inmultire la detectarea unui factor nul.
11. Utilizarea multiplicatorului hardware, daca acesta este disponibil, pentru calcule matematice complexe. In acest fel, o inmultire pe 16 biti va dura 12 cicli MCLK in loc de circa 170 cicli necesari inmultirii realizate in program. La utilizarea bibliotecii cu suport pentru calcule in virgula mobila utilizarea multiplicatorului hardware reduce de asemenea timpul de calcul

Respectind recomandarile anterioare, consumul de curent in modul activ devine nesemnificativ fata de consumul modului low-power LPM3 datorita timpului extrem de scazut de lucru in modul activ. Puterea de calcul deosebita a procesorului MSP430, evaluata la 1623 MIPS/W permite ignorarea efectului unei singure comenzi asupra consumului mediu.

### ***10.2.6 Autodescargarea bateriei***

Aceasta parte a consumului de curent dintr-un sistem MSP430 poate fi putin influentata. Important este aici sa se respecte cu strictete recomandarile producatorului.

Se recomanda plasarea bateriei intr-un loc relativ rece din carcasa. Aceasta inseamna nu direct pe sau in vecinatatea partilor fierbinti, de exemplu exact pe calorifer la un repartitor de costuri electronic.

O valoare frecvent utilizata pentru estimarea descarcarii unei baterii de-a lungul unui timp de functionare de 10 ani este lucrul cu o capacitate de numai 70% din valoarea nominala. Aceasta valoare presupune o autodescargare anuala pSe de:

$$p_{SE} = 1 - \sqrt[10]{0.7} = 0.03504 \approx 3.5\%$$

Curentul de autodescargare ISE are astfel o valoare de aproximativ 2μA



UNIUNEA EUROPEANĂ

MINISTERUL MUNCII, FAMILIEI ȘI  
PROTECȚIEI SOCIALE  
AMFOSDRUFONDUL SOCIAL EUROPEAN  
POSDRU  
2007-2013INSTRUMENTE STRUCTURALE  
2007-2013



**Structuri hardware si algoritmi specifici microsistemelor EMBEDDED***Consumul cuartului*

Calitatea ridicata a cristalului de cuarț și frecvența sa de rezonanță de valoare scăzută (32768Hz) determină o funcționare sigură și eficiență a oscilatorului la o putere necesară de 1μW până la maxim 10μW. La o alimentare de 3V, această putere implică un consum de curent al cuartului între 333nA și 3.33μA. Valoarea medie a consumului cuartului este deci  $I_{Qu}=1\mu A$  iar acest curent circulă în permanență, pentru că funcționarea oscilatorului de 32.768KHz ca bază de timp în modul low-power LPM3 este esențială.

*Consumul afisorului cu cristale lichide*

O calitate bună de fabricație și o frecvență scăzută de lucru (128Hz) asigură un consum scăzut al afisorului cu cristale lichide, aproximativ 13nA/mm<sup>2</sup> de suprafață activă a afisajului. Pentru un afisor cu o suprafață de 100mm<sup>2</sup> curentul consumat va fi de:

$$I_{LCD} = 100mm^2 \times 13nA / mm^2 = 1.3\mu A$$

Microcontrolerul MSP430 permite optimizarea adaptării la un anumit afisor cu cristale lichide, prin utilizarea unor rezistoare externe la punctele R03, R13, R23 și R33. Curentul IRLCD prin aceste rezistoare se adaugă curentului consumat de afisor, astfel că pentru un afisor cvadruplu multiplexat și deci cu 4 rezistoare de 1.5M, se consumă suplimentar:

$$I_{RLCD} = \frac{V_{CC}}{4 \times R_{LCD}} = \frac{3.0V}{4 \times 1.5M\Omega} = 0.5\mu A$$

*Consumul componentelor electronice externe*

Tastele și butoanele conectate la porturile de intrare ale microcontrolerului pot rămâne închise pe perioade lungi de timp – de exemplu, contactul de la morisca unui contor mecanic de apă în timpul concediului locatarilor. În acest caz va circula timp îndelungat un curent prin rezistența de pull-up obligatorie asociată portului și, datorită valorii relativ scăzute a acesteia (sute de kilohmi), se obține un consum mult crescut față de cazul normal de funcționare. Astfel de contacte trebuie să fie prevăzute cu posibilitatea comutării software, ca în figura alăturată. În acest fel se poate evita trecerea curentilor prin rezistoarele de pull-up sau pull-down.



UNIUNEA EUROPEANĂ

MINISTERUL MUNCII, FAMILIEI ȘI  
PROTECȚIEI SOCIALE  
AMFOSDRUFONDUL SOCIAL EUROPEAN  
POSDRU  
2007-2013INSTRUMENTE STRUCTURALE  
2007-2013

## Structuri hardware si algoritmi specifici microsistemelor EMBEDDED

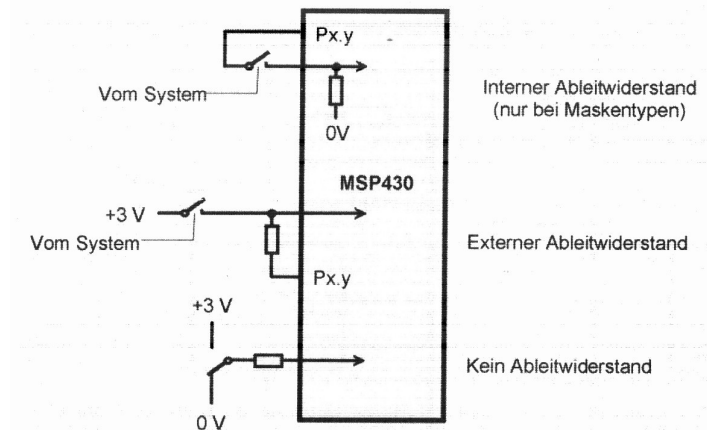


Figura 10.10. Elementele de conectica verificate pentru reducerea consumului

Programul trebuie sa verifice daca un contact este tinut inchis pentru o durata mai mare decit un timp predefinit si, daca da, sa comute contactul in asa fel incit sa se anuleze curentul prin el. Dupa aceasta este necesara inasa o testare periodica a starii contactului, pentru a detecta o eventuala deschidere a sa – daca s-a deschis, atunci trebuie sa se revina la situatia initiala. Intrucit aceste testari ale starii contactului sint foarte scurte, de ordinul microsecundelor, consumul suplimentar de curent datorat lor este nesemnificativ.

In cazul in care intrarile microcontrolerului MSP430 nu sint la un potential definit, in vecinatatea valorilor Vcc sau Vss, atunci prin circuitul de intrare al portului va circula un curent suplimentar pentru ca tranzistoarele de intrare(ambe) nu sint complet blocate. Figura anterioara prezinta trei modalitati de a da un potential definit unui port de intrare:

- intrare cu rezistenta pull-down interna; contactul poate fi activat/dezactivat cu o linie de iesire. Aceasta linie va fi plasata in starea Vss sau hi-Z dupa caz
- intrare cu rezistenta pull-up externa. In acest caz rezistenta de pull-up va fi comutata la acelasi potential ca si contactul, adica rezistenta la Vss cind contactul este deschis si la Vdd cind contactul este inchis
- fara rezistenta la intrare- contactul leaga intrarea la un potential bine definit (Vss sau Vdd)

Circuitele externe, de exemplu senzorii, trebuie sa fie oprite atunci cind nu sint utilizate. Acest lucru se poate realiza fie alimentind senzorul dintr-un port, daca curentul consumat nu depaseste posibilitatile portului (acesta are o rezistenta RDSon de ordinul 100 de ohmi) sau din mai multe linii ale portului, plasate in paralel, pentru cresterea capabilitatii de curent, fie utilizand un tranzistor PNP extern pe post de comutator.



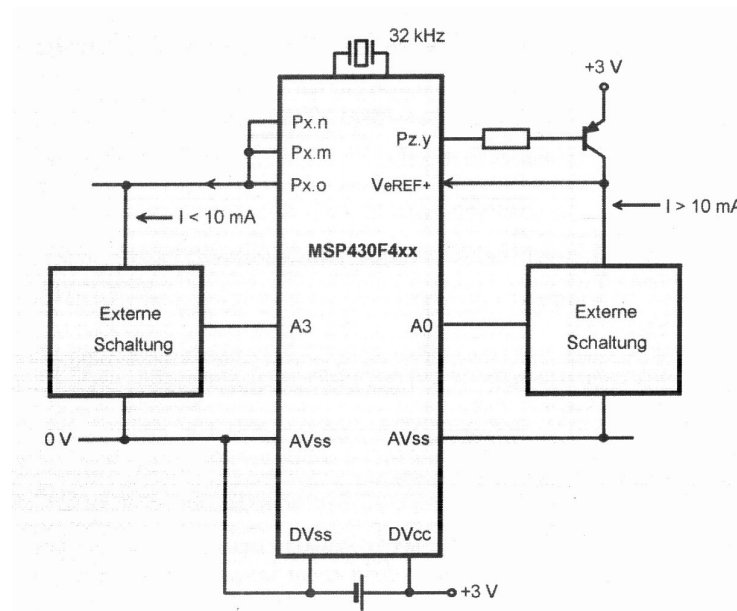


Figura 10.11. Schema unui senzor cu optimizarea consumului

De exemplu, un senzor de  $1K\Omega$  consuma 3mA (enorm!) alimentat la 3V. Acelasi senzor consuma un curent mediu  $I_{senzor}$  cind este activat doar o data la 60 de secunde in timpul masuratorii efectuate cu modulul ADC12 (timp de 3-13  $\mu s$ ):

$$I_{senzor} = \frac{3V \times 13\mu s}{1K\Omega \times 60s} = 0.506nA$$

Curentul mediu prin senzor este acum numai 0.5nA, de 5930000000 ori mai redus.

Calculul duratei de viata a bateriei

Considerind valorile analizate anterior se poate estima durata de viata a bateriei (baterie cu litium):

$$t_{Bat} = \frac{Q_{Bat}}{I_{CC} + I_{Qu} + I_{LCD} + I_{RLCD} + I_{Sys}}$$

in care intervin urmatoarele elemente:

- tBatt            durata de viata a bateriei in ore
- Qbatt            capacitatea utila a bateriei (70% din capacitatea nominala,  $Q_{nom}=0.5Ah$ )
- ICC            curentul mediu consumat de microcontrolerul MSP430F44x (1.88 $\mu A$  in cazul analizat)

**Structuri hardware si algoritmi specifici microsystemelor EMBEDDED**

- IQu               curentul mediu prin cuart( $1\mu A$  in cazul de fata)
- ILCD            curentul prin afisorul cu cristale lichide ( $1.3\mu A$  in cazul de fata)
- IRLCD          curentul prin rezistentele LCD externe ( $0.5\mu A$ )
- Isys            curentul mediu prin circuitele externe(nu este luat in calcul fiind puternic dependent de aplicatia concreta)

Pentru o temperatura a mediului ambiant de 25 de grade Celsius si considerind valorile calculate anterior se obtine o valoare estimata a duratei de functionare a bateriei de:

$$t_{Bat} = \frac{0.7 \times 0.5 Ah}{1.88\mu A + 1.0\mu A + 1.3\mu A + 0.5\mu A} = 74786h$$

Acest numar de ore corespunde unei durate de viata a bateriei de 8.53 ani. Prin utilizarea unei tensiuni de alimentare de 2.2V se poate creste acest timp, pentru ca toti curentii se vor reduce.

**Bibliografie:**

1. Istvan Sztojanov, Sever Pașca, Elisabeta Buzoianu, Aplicații hardware și software cu microcontrolerul PIC12F675, Editura Cavallioti, ISBN 978-973-7622-54-9, Bucuresti 2008
2. Istvan Sztojanov, Alexandru Vasile, Elisabeta Buzoianu, Sever Pașca, *Programarea microcontrolerelor din familia Intel, Aplicații practice hardware cu 80C552*, Editura Man-Dely, ISBN 973-85681-5-3, București 2004.
3. <http://vega.unitbv.ro/~romanca/EmbSys/>
4. <http://facultate.regielive.ro/cursuri/electronica/>
5. [www.microcip.com](http://www.microcip.com)
6. Andrei Drumea, Teza de doctorat, UPB 2009
7. BARUCH Zoltan, Translatarea limbajelor de descriere a unităților hardware, Referat de doctorat Cluj Napoca 2008

