

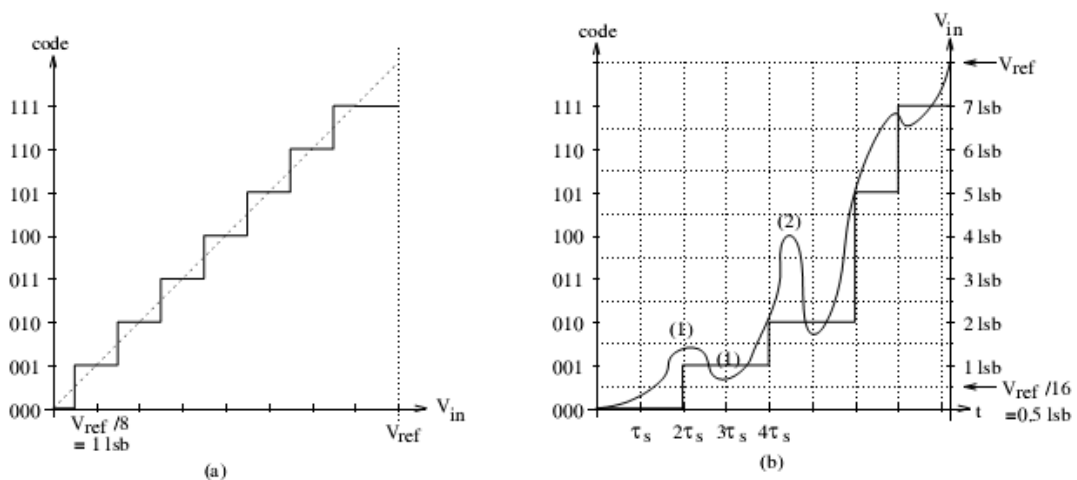
Analogas I/O

Iepriekšējā nodaļā tika apskatītas ciparu ieejas/izejas. Tur ciparu signāli tiek atspoguļoti kā divas diskrētas vērtības 0 un 1. Kaut gan tas ir izdevīgi, ir gadījumi, kad ir nepieciešams zināt precīzo līnijas spriegumu, piemēram, izmantojot fototranzistoru kā gaismas sensoru: sprieguma kritums uz tā izejas ir tieši proporcionāls uz tā krītošam gaismas daudzumam, un, lai pareizi novērtēt sensora rādījumu, mikrokontrollierim jāspēj strādāt ar analogām vērtībām. No citas puses, pēc būtības mikrokontrollieris ir ciparu iekārta, tāpēc ir nepieciešams veids, kā analogus signālus pārveidot ciparu formā un atpakaļ. Šī problēma tiek atrisināta ar analogo mikrokontrolliera moduli.

Analogais/ciparu pārveidotājs

Ja sprieguma vērtība ir svarīga, piemēram, ja mēs gribam izmantot foto tranzistoru, lai noteiktu un atspoguļotu spilgtumu, ar vienkāršu komparatoru nebūs pietiekami. Tajā vietā ir nepieciešams veids, kā pārveidot analogu lielumu ciparu formā. Šīm nolūkam daudzi mikrokontrollieri satur *analogu-ciparu pārveidotāju* (ACP) (analog-to-digital converter (ADC)), kas pārveido analogo ieejas lielumu binārā skaitlī.

Darbības princips



Zīmējums 2.17: Pamatideja analogs-ciparu pārveidošanā ($r=3$, $GND=0$). (a) Analoga sprieguma attēlošana ciparu kodā, (b) ieejas signāla piemērs un pārveidošanas neprecizitātes.

Zīmējums 2.17(a) parāda analogs-ciparu pārveidošanas pamatprincipu. Analogs ieejas sprieguma diapazons $[GND, V_{ref}]$ tiek sadalīts 2^r iedaļās, kur r ir bitu skaits, kas tiek izmantots ciparu vērtības atspoguļošanai. Katra iedaļa atbilst ciparu kodam no 0 līdz 2^r-1 . Analoga vērtībai tiek piešķirts kods, mūsu gadījumā pēc vidējā punkta, ar pārvades funkcijas palīdzību. Mēs saucām par *izšķirtspēju*, literatūrā tiek izmantots arī termins *vārda garums*. Tipiskas r vērtības ir 8 vai 10 biti, taču mēdz būt arī 12 biti un vairāk. Ciparu skaitļa lsb norāda mazāko sprieguma starpību $V_{ref}/2^r$ kas var tikt atšķirta droši. Šo vērtību sauc par ACP *granularitāti*, bet to mēdz saukt arī par *izšķirtspēju*. Gandrīz visu iedaļu lielums atbilst 1 lsb, izņemot pirmo iedaļu (0.5 lsb) un pēdējo iedaļu (1.5 lsb). Šāda asimetrija rodas no tā, ka ciparu kodam 0 ir jāatbilst 0V, tāpēc pirmajai iedaļai ir tikai puse no pārējo iedaļu lieluma, kamēr ciparu kodam 2^r-1 būtu jāatbilst $V_{ref}-1$ lsb, lai

varētu viegli palielināt bitu skaitu. Lai izvairītos no asimetrijas, varētu, piemēram, izmantot iedaļas apakšējo robežu kā atskaites punktu. Taču šajā gadījumā sliktākā gadījumā kļūda būtu +1 lsb. Izmantojot viduspunktu, kļūda ir tikai ± 0.5 lsb.

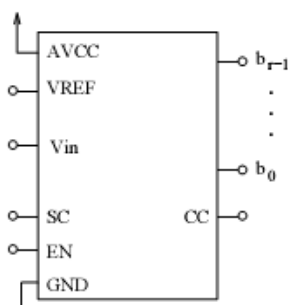
Kā var redzēt no Zīmējums 2.17(b), pārveidošana ievieš neprecizitātes analogas vērtības atspoguļojumā mikrokontrollierī. Pirmkārt, analogas vērtības atspoguļošana iedaļās ievieš informācijas zudumu rezultātā. Analoga signāla svārstības vienas iedaļas ietvaros pāriet neievērotas, t.i. abi punkti (1) attēlā tiek attēloti ar vienādu ciparu kodu 001. Šāda situācija var tikt uzlabota samazinot granularitāti. Viens veids, kā to izdarīt ir palielināt r , palielinot vārda garumu. Otrs veids, kā var uzlabot granularitāti ir samazināt V_{ref} , bet tas samazinās arī ieejas sprieguma intervālu.

Otrkārt, *pārveidošanas laiks*, kas ir laiks no pārveidošanas sākuma līdz brīdim, kad pārveidojuma rezultāts ir pieejams, nav nulle. Kā sekas, mēs iegūstam noteiktu minimālo diskretizācijas periodu T_s starp diviem veiksmīgiem pārveidojumiem, kas rada informācijas zudumu laikā. Vērtības izmaiņas starp diviem pārveidojumiem tiek zaudētas, kā var redzēt attēla (2) punktā. Augšējā ieejas frekvences f_{max} robeža, kas var tikt diskretizēta un rekonstruēta ar ACP ir *Šanona diskretizācijas teorēmu (Naikvista kritērijs)*:

$$f_{max} < \frac{f_d}{2} = \frac{1}{2T_d}$$

Teorēma nosaka, ka maksimālai ieejas signāla frekvencei f_{max} ir jābūt mazākai par pusi no diskretizācijas frekvences f_s . Acīmredzami, ka tas nozīmē, ka augstām ieejas frekvencēm minimālais diskretizācijas periodam T_d , kas ir atkarīgs no izmantojamā pārveidošanas metodes, ir jābūt mazam.

Error: Reference source not found parāda vienkāršu ACP kā melnu kastī. AVCC un GND pini nodrošina barošanu pārveidotājam. V_{ref} nodrošina ACP ar maksimālo spriegumu, kas tiks izmantots pārveidošanai. Pie V_{in} tiek pieslēgts mērāmais spriegums. EN (enable) ieeja un pielaišanas ieeja jaunas pārveidošanas sākšanai pabeidz ieejas pusi no mūsu vienkāršā ACP. Izejas pusē atrodas pārveidotā vērtība un signāla līnija, kas norāda uz pabeigto pārveidojumu.

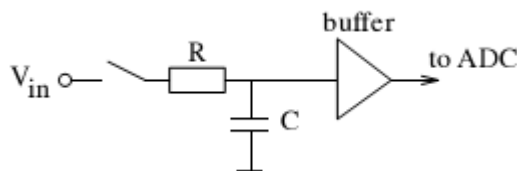


Zīmējums 2.18: ACP kā melnā kaste.

Attēlā maksimālais spriegums V_{ref} , *balstspriegums* kas tiek izmantots pārveidošanas intervāla definēšanai, tiek pieslēgts ar ārējo pinu. Tomēr, daži kontrolleri piedāvā arī iekšējos balstspriegumus. ATmega16, piemēram, ļauj lietotājam izvēlēties starp iekšējo 2.56V balstspriegumu, analogo barošanas spriegumu AVCC vai ārējo balstspriegumu. Ja analogs ieejas signāls ir lielāks par V_{ref} , tas tiek atspoguļots ar $2^r - 1$. Sarežģītāki ACP var arī norādīt šādu

pārpildi atsevišķā pārpildes bita. Līdzīgi, signāls, kas ir mazāks par GND tiek atspoguļots ar kodu 0.

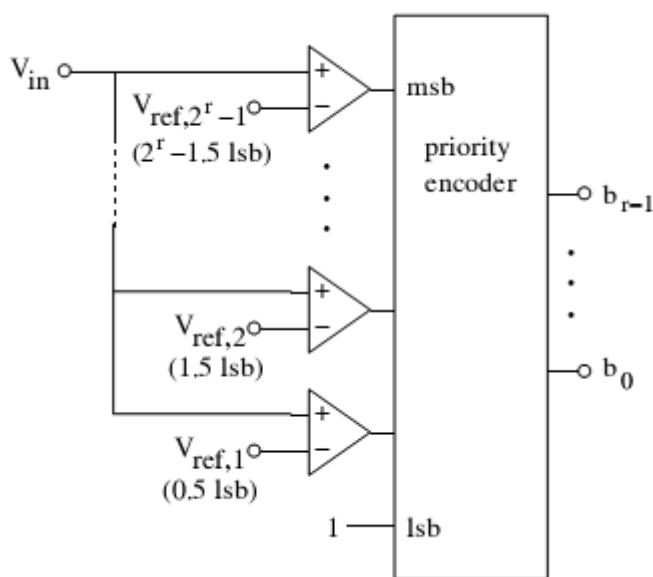
Ieejas signāla svārstības pārveidošanas laikā var pasliktināt rezultāta kvalitāti, tāpēc, lai pārveidošanas laikā ieejas signālu uzturētu stabili, tiek izmantota parauga tveršanas ķēde (Error: Reference source not found). Pārveidošanas sākumā kondensators tiek uzlādēts no ieejas signāla. Pēc noteikta laika tas tiek atslēgts no ieejas signāla un tiek izmantots kā ACP ieeja, nodrošinot, ka pārveidošanas laikā spriegums paliek nemainīgs.



Zīmējums 2.19: ACP parauga tveršanas ķēde

Pārveidošanas metodes

Ir dažas dažādas analoģ-ciparu pārveidošanas metodes. Visvienkāršākā ir *flash konvertors* (Error: Reference source not found).



Zīmējums 2.20: Flash konvertora darbības princips.

Ideja ir ļoti vienkārša: ieejas spriegums V_{in} tiek salīdzināts ar dažiem balstspriegumiem $V_{ref,i}$, kur

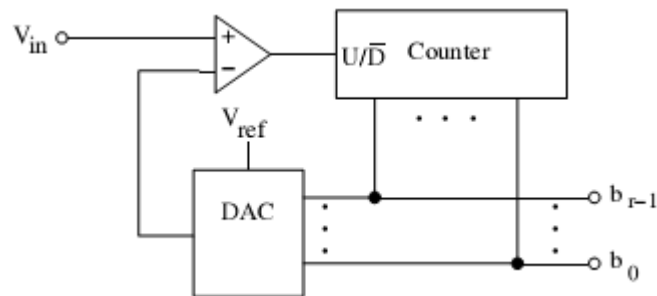
$$V_{ref,i} = \frac{V_{ref} * (2i - 1)}{2^{r+1}}, 1 \leq i \leq 2^r - 1$$

Ja ieejas spriegums ir lielāks par komparatora balstspriegumu, tas izejā izdos 1. Visas komparatoru izejas ir pieslēgtas pie prioritātes kodētāja, kas izejā izvadīs bināru ciparu, kas atbilst msb no ieejas vērtības. Kodētāja lsb ir pievienots pie 1, tāpēc, ja neviens no komparatoriem nav aktīvs, kodētājs izejā izdos kodu 0.

No formulas iegūstamie balstspriegumi ir ar iedaļas lielumu 0.5 lsb kodam 0, lielumu 1.5 kodam 2^r-2 un liekumiem 1 lsb visam pārējam iedaļām.

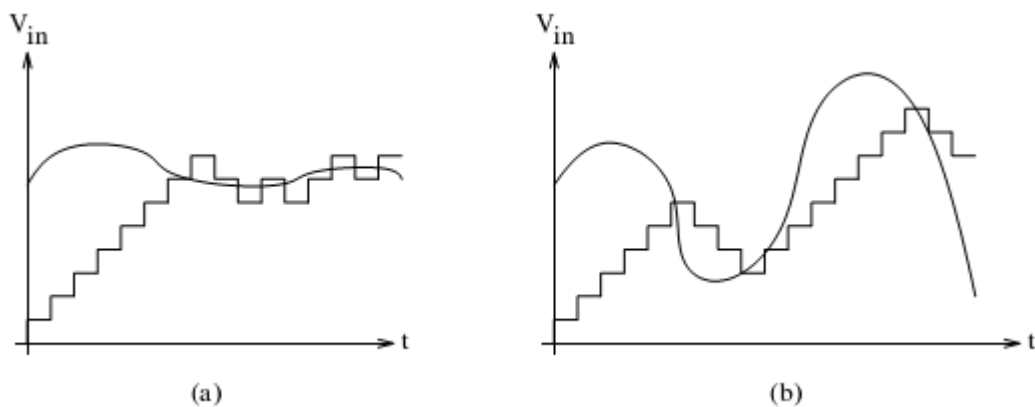
Flash konvertora lielā priekšrocība, no kā ir cēlies tā nosaukums, ir tā ātrums: pārveidošana tiek paveikta vienā solī, visas iespējamās iedaļas, kurām var atbilst ieejas spriegums tiek pārbaudītas vienlaikus. No tā izriet, ka tā laika sarežģītība ir $O(1)$. Taču ātrs pārveidošanas laiks ir sasniegts ar ļoti lielu aparatūras sarežģītību: ir nepieciešami 2^r-1 komparatori un vēl viena bita pievienošana koda vārdam dubulto aparatūras prasības. Tāpēc flash konvertori ir samērā dārgi.

Pavisam cita pieeja tiek izmantota sekojošā konvertorā (Error: Reference source not found)



Zīmējums 2.21: Sekojošā konvertora darbības princips.

Šāds konvertors ir interesants ar to, ka tam ir nepieciešams ciparu-analogs pārveidotājs, lai veiktu analogs-ciparu pārveidojumu. Princips atkal ir ļoti vienkāršs: Sekošanas konvertora pamatā ir skaitītājs, kas glabā tekošo ieejas signāla ciparu tuvinājumu. Skaitītāja vērtība tiek pārveidota par analogu signālu, izmantojot DAC un salīdzināts ar ieejas spriegumu. Ja ieejas spriegums ir lielāks par tekošo skaitītāja vērtību, tad skaitītājs tiek palielināts, pretējā gadījumā to samazina.

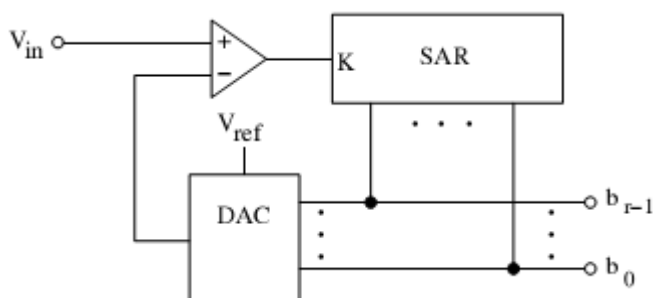


Zīmējums 2.22: Sekošanas konvertors darbībā.

Kā var redzēt Error: Reference source not found(a), sekošanas konvertoram ir nepieciešams liels laiks, kamēr tas sasniegs signālu, bet tiklīdz tas atrod signālu, pārveidošana ir diezgan ātra. Diemžēl, tas ir patiesi tikai lēni mainīgiem signāliem. Ja signāls mainās pārāk ātri, kā tas ir parādīts attēla (b) daļā, konvertors atkal pavada lielāko laiku daļu signāla atrašanai. Pareizas pārveidojuma vērtības ir tikai punktos, kur skaitītājs maina skaitīšanas virzienu.

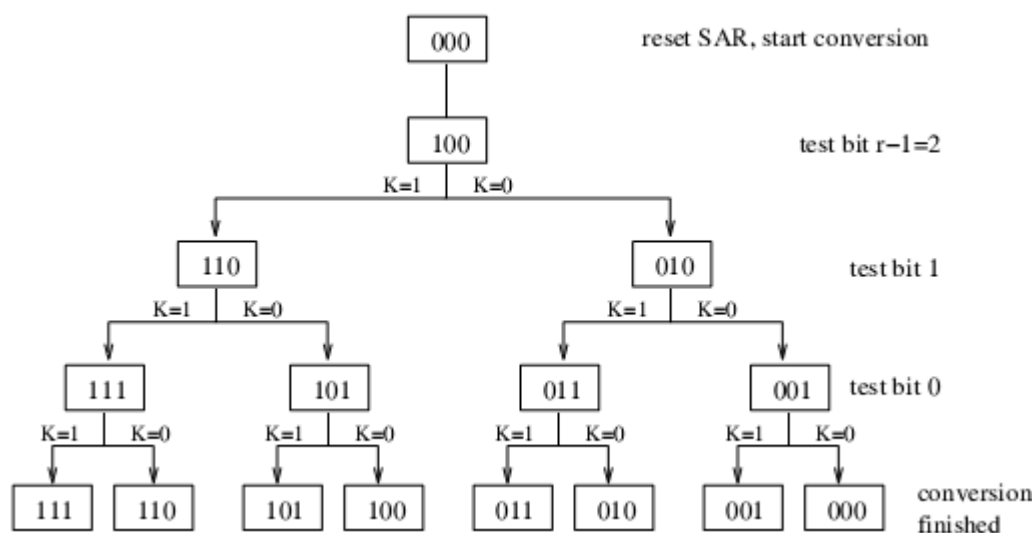
Tā kā sliktāka gadījuma laika sarežģītība sekojošām konvertoram ir $O(2^r)$, tas ir pārāk lēns daudziem pielietojumiem.

Nedaudz izmainot sekojošā konvertora darbības principu, iegūst *secīgas tuvināšanas konvertoru* (Error: Reference source not found).



Zīmējums 2.23: Secīgas tuvināšanas konvertora darbības princips.

Kā var redzēt, vienīga atšķirība ir tā, ka skaitītājs tika aizstāts ar *secīgas tuvināšanas reģistru* (STR) (*successive approximation register* (SAR)). STR izmanto bināras meklēšanas paņēmieni, nevis vienkāršu skaitīšanu augšup/lejup: kad ir sāta pārveidošana, STR msb (b_{2^r-1}) tiek iestatīts un iegūta analogā vērtība tiek salīdzināta ar ieejas spriegumu. Ja ieejas spriegums ir lielāks par iegūto, b_{2^r-1} tiek atstāts, pretējā gadījumā to uzstāda uz 0. Tad uzstāda nākamo b_{2^r-2} bitu un tā tālāk, līdz tiek pārbaudīts pēdējais bits. Pēc r salīdzināšanām, STR vērtība atbilst ieejas spriegumam. Error: Reference source not found demonstrē darbību.



Zīmējums 2.24: Secīgas tuvināšanas reģistrs darbībā ($r=3$).

Secīgas tuvināšanas konvertors ar tā lineāro sarežģītību $O(r)$ ir labs kompromiss starp flash konvertora ātrumu un sekojošā konvertora vienkāršību. Cita priekšrocība par sekošanas konvertoru ir tas, ka tā pārveidošanas laiks nav atkarīgs no ieejas sprieguma un ir konstants. Lai izvairītos no kļūdām pārveidošanas laikā, ir nepieciešamā parauga tveršanas ķēde.

Secīgas tuvināšanas konvertori tiek bieži izmantoti mikrokontrollieros.

ADC izmantošana

Mikrokontrollieri, kas ir aprīkoti ar analogām ieejām, parasti piedāvā 4-16 analogas ieejas kanālus, kas caur multiplexoru ir savienoti ar vienu iekšējo ACP. Rezultātā vienlaikus no vairākām analogām ieejām vērtības nolasīt nevar, bet tos ir jālasa secīgi, citu pēc cita. Praktiski, tas nozīmē, ka ir jāizvēlas kanālu un tad jāsāk pārveidošanu. Kad pārveidošana ir pabeigta, nolasa vērtību, izvēlas nākamo kanālu un sāk jaunu pārveidošanu. Daži ACP ļauj uzstādīt nākamo kanālu kamēr notiek pārveidošana un izmanto jaunus iestatījumus tiklīdz pārveidošana ir pabeigta. Sarežģītāki pārveidošanas moduļi pat piedāvā automātiskās palielināšanas iespējas, kur nepieciešams uzstādīt pirmo kanālu un ACP automātiski pārslēgs kanālus ar katru jaunu pārveidošanu.

Pārveidošanu var uzsākt lietotājs, bet parasti mikrokontrollieriem ir vairāki ACP pielaišanas avoti. Bez atsevišķa pina ACP statusa reģistrā, kas ļauj lietotājam uzsākt pārveidošanu (*vienas pārveidošanas režīmā*), liekākai ACP daļai ir arī *nepārtrauktas pārveidošanas režīms*, kur jauna pārveidošana tiek uzsākta tiklīdz pēdējā ir pabeigta. Papildus arī citiem avotiem, tādiem kā taimeris, ieejas notikums, vai kāds ārējais signāls var būt iespēja sākt pārveidošanu.

Pēc pārveidošanas uzsākšanas, ACP ir nepieciešams laiks, lai uzlādētu parauga tveršanas ķēdi un tad vēl nedaudz vairāk laika, lai izdarītu pārveidošanu. Tā kā mikrokontrollieri parasti izmanto secīgas tuvināšanas konvertorus, pārveidošanas laiks ir nemainīgs. Pirmais pārveidojums pēc pārslēgšanas uz jaunu kanālu var aizņemt vairāk laika, jo konvertoram būs nepieciešams no jauna uzlādēt ieejas ķēdi.

Pareizai darbībai, ACP ir nepieciešams takts signāls, kas ir noteiktās robežās. Ja signāls tiek ņemts no ārēja takts avota, kā piemēram, ATmega16 gadījumā, ir iespēja uzstādīt priekšdalītāju,

lai samazinātu frekvenci līdz tā būs piemērota izmantošanai pārveidotājā. Priekšdalītāja takts frekvence tiek izmantota konvertora vadībai un tāpēc nosaka pārveidošanas laiku. Protams, var izmantot frekvenci, kas ir ārpus noteiktām robežām. Ja takts frekvence ir zemāka par noteikto, pārveidošanas laiks būs ilgāks, bet nevajadzētu būt nelabvēlīgu ietekmju uz pārveidošanas rezultātu. Ja takts frekvence ir pārāk liela, pārveidošanas precizitāte samazinās, jo pēdējie biti vairs nebūs pareizi. Jo lielāka frekvence, jo sliktāka būs precizitāte. Ja frekvence būs pārāk augsta, rezultāts var būt pilnīgi nepareizs.

Pēc pārveidošanas pabeigšanas, ACP statusa reģistrā tiek uzstādīts karodziņš. Analogs modulis, ja nepieciešams, var izraisīt arī pārtraukumu. Pārveidošanas rezultāts tiek glabāts datu reģistrā. Tā kā ACP izšķirtspēja var būt lielāka par mikrokontroliera vārda garumu, piemēram, kad uz 8-bitu mikrokontroliera tiek izmantots 10-bitu ACP, kļūst svarīga atomara pieeja datu reģistram. Parasti kontrolieri piedāvā mehānismu atomarai pieejai, piemēram, ATmega16 iesaldē datu reģistra saturu kad tiek nolasīts apakšējais baits. Reģistra atjauninājumi tiek atlikti līdz brīdim, kad tiek nolasīts augšējais baits.

Vērts atzīmēt, ka ja spriegums ir ārpus atļautām robežām, tas tiks atspoguļots uz robežas. Tā negatīvas vērtības tiks atspoguļotas kā 0, bet vērtības, lielākas par V_{ref} tiks atspoguļotas kā 2^r-1 . Tomēr, lai izvairītos no analoga moduļa bojājumiem, spriegumiem ir jābūt robežās, kas ir norādītas kontroliera datu lapā.