

2.3 Ciparu I/O

Digitālā ievade/izvade (input/output I/O) jeb arī lai būtu nedaudz precīzāk, novērošanas un kontrolēšanas datortehnika, ir galvenā mikrokontroleros. Praktiski visi mikrokontrolieri ir ar vismaz 1-2 digitāliem I/O savienojumiem kas var tik tieši savienoti ar datortehniku (kontroliera limitu ietvaros). Kopumā lielākoties kontrolieriem ir 8 – 32 savienojumi un dažiem pat vēl vairāk (Motorola HCS12 vairāk par 90).

I/O savienojumi galvenokārt ir sagrupēti portos pa 8 savienojumiem, kam var piekļūt ar viena baita piekļuvi. Savienojumi var gan tikai ievades, tikai izvades, vai visbiežāk sastopamie ir abpusējie, tie ir gan ar ievades gan izvades funkciju. Neskaitot to I/O iespējas lielākoties savienojumi ir ar vienu vai vairākām mainīgām funkcijām lai ietaupītu savienojumu skaitu un saglabātu čipa izmērus mazus. visi pārējie kontrolieru moduļi kas pieprasa I/O savienojumus, tādi kā analogais modulis vai taimeris, izmanto mainīgās funkcijas no digitālajiem I/O savienojumiem. Aplikācijas programmētājs var izvēlēties kuras funkcijas vajadzētu izmantot priekš savienojumiem tās aktivizējot izmantojot attiecīgo moduli. Protams ja savienojums tiek izmantots priekš analogā moduļa tad tas tiek zaudēts priekš digitālā I/O, tādēļ datortehnikas izstrādātājiem ir jāizvēlas uzmanīgi kurus savienojumus izmantot kurām funkcijām.

Šajā sadaļā Jūsu uzmanība tiks vērsta uz digitālo I/O iespējām. Vēlākās sadaļas aptvers mainīgās funkcijas. No sākuma paskaidrosim ko mēs domājam ar terminu digitālais. Kad tiek nolasīta savienojuma vērtība uz voltmetra, mēs redzēsim analogu voltāžu. Tomēr mikrokontrolieris digitalizē šo voltāžu to kartējot vienā no diviem režīmiem, loģiskajā 0 vai 1. Tātad kad tiek runāts par digitālo I/O mēs ar to domājam ka savienojuma vērtība no kontroliera perspektīvas ir vainu 0 vai 1. Der atcerēties ka pozitīvi loģiskais 1 attiecas pret augstu līnijas stāvokli kamēr 0 atbilst zemam līnijas stāvoklim. Negatīvajā loģikā tas ir otrādāk, 1 atbilst zemam līnijas stāvoklim bet 0 – augstam.

Trīs reģistri kontrolē savienojumu uzvedību:

Datu virziena reģistrs (Data directional register – DDR): katrs divvirzienu ports ir ar savu DDRm kas satur vienu bitu katram porta savienojumam. Savienojuma funkcionalitāte (ievade vai izvade) tiek noteikta notīrot vai iestatot tā bitu iekš DDR. Dažādi porta savienojumi var tikt konfigurēti dažādi, tā ka tas ir pilnīgi normāli ja trīs savienojumi no porta ir konfigurēti kā izejas porti bet pārējie kā ieejas. Pēc atiestatīšanas DDR biti tiek inicializēti uz ievadi. Reģistra nolasīšana atgriež tā vērtību.

Porta reģistrs (port register - PORT) : šis reģistrs tiek izmantots lai regulētu voltāžas līmeni izejas savienojumos. Pieņemot ka savienojums ir nokonfigurēts kā izejas, tad ja tā bits PORT reģistrā ir uzstādīts, tad savienojums būs augsts; ja bits ir notīrīts, tad savienojums būs zems. Lai izvairītos no pārējo bitu pārrakstīšanas kad tiek rediģēts bits ir ieteicams izmantot kontroliera bitu operācijas. Citādi ir jāizmanto Lasīšanas-modifikācijas-rakstīšanas piekļuve un ir jānodrošina ka šī piekļuve netiek pārtraukta.

Priekš izejas savienojumiem, reģistra nolasīšana atgriež vērtību kas ir ierakstīta. Priekš ievada savienojumiem funkcionalitāte atšķiras no kontroliera. Daži kontrolieri atļauj lasīt ievada savienojumus caur porta reģistru. Citi kontrolieri, tādi kā, ATmega16, izmanto porta daļas priekš

citiem nolūkiem ja savienojumi ir uzstādīti uz ievades režīmu, tātad šeit tiks nolasīta vērtība kas tika ierakstīta reģistrā.

Porta ievades reģistrs (port input register - PIN) : PIN reģistrs galvenokārt ir tikai nolasāms un satur sevī šā brīža statusu (augsts vai zems) visiem savienojumiem, neatkarīgi no tā vai tie ir konfigurēti kā izejas vai ieejas. Tas tiek izmantots lai lasītu ieejas savienojumu statusum bet var tik izmantots lai lasītu izejas savienojumu stāvokli lai pārliecinātos ka izejas ir nolasītas pareizi. Rakstīšana šajā reģistrā ir bez sekām.

Katrs bits šajos reģistros ir piesaistīts vienam savienojumam, ja ir vēlme mainīt uzstādījumus tikai vienam savienojumam tad tas ir jāveic nemainot pārējo bitu vērtības reģistros. Labākais veids kā to izdarīt ir izmantot bitu operācijas, ja kontrolieris atbalsta šādu opciju, ja ne tad jāizmanto lasīšanas-rediģēšanas-rakstīšanas operācija visam reģistram un tā nedrīkst tikt pārtraukta.

Digitālā ievade

Digitālās ievades funkcionalitāte tiek izmantota vienmēr kad novērotais signāls būtu jāpārtrauc digitāli, tas ir tikai tad ka tas maina savu statusu no 0 uz 1 vai otrādi. Vienalga vai dotais signāls tiek interpretēts kā augsts vai zems tas ir atkarīgs no voltāžas līmeņa kontrolierī. Piemēram operēšanas voltāža V_{cc} priekš ATmega16 ir jābūt intervālā starp [4.5, 5.5]V, tā ievades zemā voltāžai jābūt starp [0.5, 0.2V_{cc}]V un tā ievades augstajai voltāžai jābūt starp [0.6V_{cc}, V_{cc}+0.5]V. Tas atstāj intervālu (0.2V_{cc}, 0.6V_{cc}) kurā signāls nav definēts.

Digitālā paraugu ņemšana

Tā kā digitālais signāls ir tikai voltāžas vērtība, parādās jautājums, kā šī voltāžas vērtība tiek pārvērsta binārajās vērtībās iekš reģistra. Kā pirmais risinājums ir vienkārši izmantot aizbīdņus priekš PIN reģistra un noslēgt tekošo savienojuma stāvokli reģistrā. Ja aizbīdnis tiek aktivizēts izmantojot sistēmas pulksteņi tad tas saglabās tekošo stāvokli katra cikla sākumā. Dabīgi ka kopš mēs varam tikai paņemt paraugu sistēmas pulksteņa iedaļu vērtībās, tas nozīmē ka var noteikt atšķirību tikai novēloti. Var pat nokavēt impulsus tā kā tie ir īsāki par pulksteņa cikliem. Apskatīt 2.9 attēlu.

Aizture kas ir iespaidojoties no paraugu ņemšanas sadalījuma ir $d_{latch} = (0,1]$ pulksteņa ciklā. Jāievēro ka nulles tiek izlaistas, jo nav noteiktības tajā kas notiek kad signāls mainās un tajā pat laikā ir parauga ņemšanas brīža mala. Tas var tik paņemts kā paraugs, bet var arī netikt paņemts. Tāpēc ir saprātīgi atstāt nulli ārpus intervāla. Ar to pašu pamatojumu impulsiem vajadzētu būt ilgākiem kā pulksteņa cikliem lai būtu atpazītiem ar noteiktību. Atlikušajā tekstā tiks izmantots $d_{in} = (d_{min.in}, d_{max.in}]$ lai apzīmētu ievades aiztures intervālu, kur $d_{min.in}$ ir kā zemākā intervāla robeža bet $d_{max.in}$ kā augšējā intervāla robeža.

Lai arī paraugu ņemšanas tehnika izskatās noderīga un formē pamata kontroliera ievades shēmu, tā neder situācijām kas bieži parādās reālajās sistēmās: kas notiek ja signāls mainās lēni? Galu galā signāls tiek ģenerēts izmantojot datortehniku, kas var uzvesties neprognozējami, tā ka mums nav nekādas garantijas, ka signāla maiņas būs tik ātras un parādīsies problēmas ka signāls nav identificēts kad tas mēģina mainīties. Šajā gadījumā vienkāršais variants ir meta stabilitāte: aizture kas saņem nenoteiktu voltāžas līmeni kā ievadi ir ar noteiktu iespējamību p ka tā ieies un paliks meta-stabilā stāvoklī kurā tā var izdot augstu vai zemu vai nenoteiktu vērtību vai

oscilēties. Acīmredzami pēdējās divas iespējas nav pieņemamas kontrolierim un no tām aplikācijās ir jāizvairās, it īpaši svarīgās drošības sistēmās. Lai samazinātu iespējamību tādiem gadījumiem, digitālā ievades kontroliera shēmā sākotnēji izmanto schmitta triggeri lai iegūtu labi definētus stūrus un atsijātu svārstības ievades voltāžā. Tas ierobežo problēmu īsākā periodā, kura laikā schmitta triggeris samaina izejas.

Lai samainītu iespējamību meta stabilitātei arī vēlāk viens vai vairāki aizturi var tikt novietoti starp schmitta triggeriem un PIN reģistra aizturēm. Šādas konstrukcijas sauc par sinhronizatoriem. 2.11 attēls parāda bloka diagrammu rezultējošai shēmai. Katrs papildus sinhronizators aiztur iespējamību p no ienākšanas meta stāvoklī ja parādīts ar nenodefinētu ieeju, tā ka visa ķēde ar k aizturēm ieskaitot PIN aizturi ir ar iespējamību $p_k \ll 1$ ka tā šķērsos meta-stabilo stāvokli līdz pat pašam kontrolierim. Praktiski pietiek ar vienu sinhronizatoru lai samazinātu iespējamību līdz pieļaujamai robežai (tā nekada nebūs nulle neskatoties uz to cik daudz aiztures tiks ievietotas).

Varbūt jūs jau pamanījāt mīnus sinhronizācijas posmā: tas pagarina laika vienību kurā signāla maiņa ir pagājusi uz kontrolieri pēc konstantes d_{sync} , kas ir vienkārši ciklu skaits kas ir nepieciešami lai pārietu aizturēto vērtību no pirmās sinhronizācijas aiztures līdz PIN aizturai.

Traucējumu izslēgšana.

Lai arī PIN kontroliera reģistram vajadzētu normāli sekot ievada savienojumiem tik cieši cik vien iespējams, tad tas ir diezgan neiespējami ja signāls ir ar daudz traucējumiem. Šeit elektromagnētiskie traucējumi no apkārtējās vides rada īsus voltāžas lēcienus uz līnijas, un uz šīm voltāžas izmaiņām normālos gadījumos nevajadzētu kontrolierim reaģēt, tā kā tas varētu radīt kļūdainas reakcijas it īpaši saistībā ar pārtraukumiem.

Tāpēc daži kontrolieri piedāvā trokšņu izslēgšanu. Ja tā ir ieslēgta, tad kontrolieris ņem paraugus no savienojumiem ne tikai vienu reizi bet vairākas k reizes un ņem jaunu vērtību tikai tad k paraugi ir vienādi. Acīmredzami tas pievieno jaunu konstanti $d_{canc} = k-1$ cikli pie vispārējās ievades aizkaves, tātad robežas pie aizkaves sanak $d_{in} = d_{latch} + d_{sync} + d_{canc}$ pulksteņa cikli.

Atsienošie rezistori

Daudzi kontrolieri tiek iebūvēti ar atsienošiem rezistoriem. Lielākajā daļā, tie ir kā pul-up rezistori, bet dažos pull-down (HCS12). Atsienošs rezistors uzdevums ir savienot ievades pinu ar nedefinēto voltāžu ja tā netiek vadīta ar ārējo tehniku. Atsienošais rezistors tiek vadīts ar reģistru palīdzību, tie var tikt iedarbināti vai izslēgti katram pinam atsevišķi. Citos kontrolieros ir speciāli reģistri šim mērķim.

Diezgan bieži var būt tā, ka ievades pins visu laiku netiek vadīts ar datortehnikas palīdzību, visbiežāk kad tiek izmantoti vienkārši mehāniski slēdži, tādi kā DIP slēdži un pogas. Šeit ievades pini ir savienoti lai noteiktu vērtību tik ilgi kamēr vien slēdzis ir noslēgts, bet atstāt tādu kāds tas ir kad slēdzis ir atvērts. Tā ka atvērtie pini ir diezgan slikts variants (tiem ir liels trokšņu līmenis), vilcēj rezistors ir jāizmanto lai noteiktu pina iepriekš iestatīto līmeni kamēr slēdzis ir atvērts. Tas redzams 2.12 attēlā.

Attēlā redzams ka ir savienoti slēdži ar ievadi ar aktīvo pull-up, un reizēm saukts par atvērto-nosūcēj ievadi. Kamēr slēdzis ir atvērts, ievades pins ir pievienots pie Vcc un kontrolieris nolasīs 1. Aizverot slēdzi tiks savienots pins ar zemējumu un kontrolieris nolasīs 0 vērtību.

Ir vēl kāda interesanta lieta ko var pieminēt pie šī zīmējuma: sakarā ar to ka pull-up rezistors, vienalga kad slēdzis ir slēgts, plūsma virzās no kontroliera caur ieejas pinu uz ārējo zemējumu. Bez vilcēj rezistora, nebūtu neviens ievērtības cienīgs signāls no vai uz kontroliera pinu, tā ka kontrolierim nevajadzētu iespaidoties no ārējā loka tikai to nolasot. Ar pull-up aktivizētu, lai arī kontrolierim ir galvenā loma nosakot līnijas stāvokli un tomēr plūsma dosies starp ievadi un pinu pie ārējā loka.

Ja plūsma iet no kontroliera uz ārējo loku ievade tiek saukta par avota ievadi tādēļ ka tā sniedz stāvu. Ja strāvas plūsma ir no datoriekārtām uz kontrolieri, tad tas tiek saukts par slīkstošo ievadi. Kontrolieri ir ļoti specifiski saistībā ar strāvas apjomu kuru var noslīcināt, un pārsniegt robežas kas ir noteiktas dokumentos var iznīcināt pinu un pat pašu kontrolieri. Kontrolieri ir spējīgi tikt galā ar 4 – 20 mA, un ja starpība starp avotu un slīkšanu tad var tik novadīts lielāks strāvas daudzums kā tiek padots.

Digitālā izeja

Digitālā izeja tiek izmantota lai uzstādītu izejas pinos noteiktus voltāžas līmeņus. Līmeņi kas ir atbilstīgi par augstu un zemo līniju ir katram kontrolielīm atšķirīgi un atkarīgi no kontroliera operējamo voltāžu. Priekš ATmega16 pie Vcc=5V maksimālā izejas zemā voltāža ir 0,7V un minimālā augšējās līnijas voltāža ir 4,2V.

Vienalga kad pins ar DDR tiek pataisīts par izeju, kontrolieris vada šo pienu attiecīgajai vērtībai kāda tiek norādīta PORT reģistrā.

Izejas pini ir daudz kritiskāki par ieejas piniem, ņemot vērā to, ka tie lielā mērā balstās uz ārējās strāvas aizsardzību. Galu galā izejas pinu varētu pieslēgt pie zemējuma un tad uzstādīt to uz 1, tādā veidā radot īssavienojumu. Tomēr kontrolieri cenšas ņemt vērā šādus gadījumus neilgos laika momentos (mazāk par sekundi), īssavienojums var sabojāt kontrolieri, tā ka datoraļu izstrādājam ir jānodrošina ka ārējā datortehnika nevar to radīt. Ja tomēr var vai arī aplikācijas programmētājs dod priekšroku būt nodrošinātam, kontrolieris piedāvā variantu nolasīt pina šābrīža stāvokli izmantojot PIN reģistru. Protams PIN reģistrs iespaidojas no ievades aiztures, tātad īssavienojums PIN reģistrā parādīsies ar [din] pulksteņa ciklu aizturi pēc izejas pina uzstādīšanas. Tā kā aplikācijai ir jānogaida šis laiks pirms tā var pārbaudīt PIN. Ja nesaderība ir konstatēta, tad aplikācijai vajadzētu nekavējoties uzstādīt pinu uz ievades režīmu un par to paziņot lietotāju.

Visbeidzot vēlamies pievērst jūsu uzmanību jautājumam kurš reģistrs tiek uzstādīts kā pirmais izejas piniem, PORT vai DDR. Pēc pārlādes pins tiek uzstādīts kā ieeja. Ja kontrolieris neizmanto PORT bitus priekš citiem nolūkiem (ATmega16 tos izmanto lai kontrolētu augšā stūmējus) un ja kontrolieris atļauj izmantot PORT bitus ievadei, tad atbilde ir acīmredzama: kā pirmais tiek uzstādīts PORT un tad DDR.

Priekš Atmel AVR kontrolieriem tādiem kā ATmega16, no sākuma ir jātiek galā ar augšā stūmējiem lai tie neietekmētu reģistru darbību.

Ir jāievēro ka piesardzīgi izstrādātāji pārliecināsies ka ir iespējams uzstādīt abus reģistrus jebkurā gadījumā izstrādājot ārējo shēmu šādā veidā ka standarta stāvoklis datortehnikai netiek vadīts ar kontrolieri ir tas pats kad kontroliera pins izvada savas PORT standarta vērtības. Šādā veidā, ja jūs uzstādat pinu uz izeju tad nenotiks nekādas izmaiņas viņa vērtībā un būs iespējams uzstādīt PORT pinu kādu vien gribēsīt.