Styrning av lysdioder med hjälp av accelerometer

LTH - Digitala Projekt -05 VT1

Handledare: Bertil Lindvall

Per Nordbeck E01
Poyan Daneshnejad E02

Innehållsförteckning

Bilagor

3. 1. Inledning

Denna rapport behandlar genomförandet av ett digitalt projekt i kursen med samma namn vid LTH. Projektets syfte var att se hur man kan använda en givare tillsammans med en enchipsprocessor för att utföra önskade tillämpningar. Vi använde oss av en accelerometer ADXL202 som givare från Analog Devices och en AVR ATmega16 från Atmel som processor.

2. Bakgrund

Vi har länge funderat på vad det finns för typer av sensorer som kan känna av om ett objekt ligger på det sätt som man önskar och hur den märker om objektet rör på sig. Svaret är en accelerometer. Anledningen till vårt intresse av en accelerometern är att man kan göra så oändligt många applikationer med hjälp av den. I vårt projekt tänkte vi använda denna givare för att skapa en alternativ mus till datorn. Tanken var att detta skulle vara en bra applikation för rörelsehindrade. Vi tänkte att man kunde ha givaren på huvudet, benet eller annan kroppsdel som personen i fråga har kontroll över för att utföra önskade tillämpningar, till exempel arbeta med en dator. Vi planerade först att vi skulle styra fyra stycken lysdioder för att sedan tillämpa den som en datormus.

3. Komponenter

3.1 Accelerometern

Vissa dielektriska material har egenskapen att en mekanisk deformation alstrar elektrisk laddning på ytan. Denna egenskap kallas piezoelektrisk effekt. Då deformationen byter riktning byter laddningen tecken. Då ett piezoelektriskt material utsätts för en kraft erhålls på dess yta en laddning. Piezoelektrisk accelerometer kopplas till en laddningsförstärkare dvs. en operationsförstärkare återkopplad med en kapacitans. På så sätt kan man alltså få olika signaler för olika rörelser.
Vi använde oss av en accelerometer från Analog Devices med typbeteckning ADXL202. Denna accelerometer har två kanaler, en i X-ledd och en i Y-ledd. Accelerometerns utgång är en fyrkantsvåg för varje kanal. Om hela perioden för fykantsvågen är T_2 så är halva den perioden aktiv medan den andra halvan 0V. Då vi matar kretsen med 5V så är den aktiva delen av signalen även den 5V. Vi betecknar den aktiva delen av signalen som T_1.
Om vi antar att den i viloläge står platt på ett bord ger den oss en signal som ser ut som fig.1 nedan.

$$T_1 = \frac{T_2}{2}$$

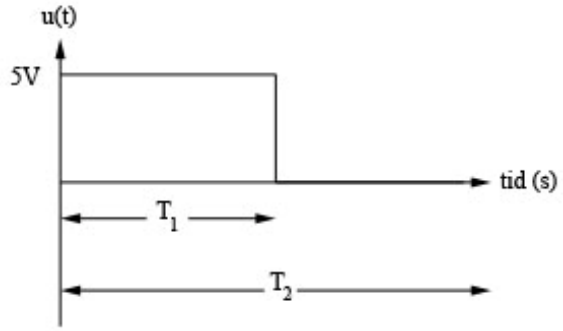


Fig.1

När man sedan vinklar chippen som accelerometern är byggd på åt ett håll så kommer den aktiva delen av pulsen, T_1 att skjutas åt till exempel höger, se fig.2. Hur mycket den förskjuts beror på hur mycket vi vinklar sensorn.

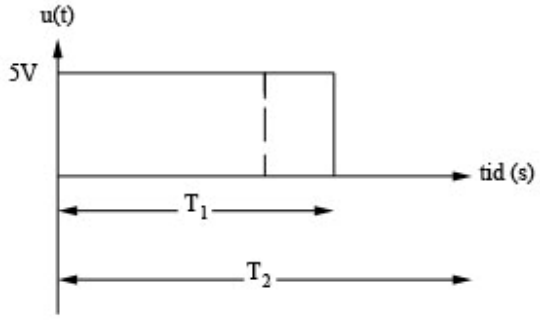


Fig.2

Samma sak gäller naturligtvis åt andra hållet, då man vinklar sensorn åt motsatt håll så förskjuts pulsen åt andra hållet, se fig.3.

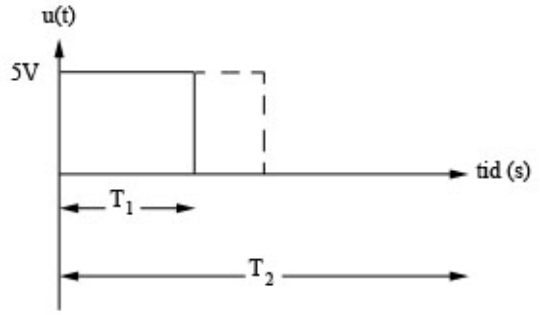


Fig.3

Hur lång hela perioden T_2 ska vara bestämmer vi genom kringkomponenter runt accelerometer. Genom att bläddra i datablad valde vi ut de resistorer och kondensatorer som var nödvändiga för att få period till 10ms. Detta är den längsta perioden som denna typ av accelerometer klarar av. Vi ansåg att vi inte behövde mer noggrant än så för vår applikation.

3.2 Atmel AVR Mega16 enchipsprocessor

ATmega 16 är en 8-bitars enchipsdator som bygger på Atmels AVR-RISC-arkitektur vilket ger mycket snabb programexekvering. Kretsen har 32 I/O-pinnar och innehåller bland annat en programmerbar seriell UART, en 10-bit 8-kanals A/D-omvandlare, watchdog-timer, komparator, tre timer/räknare, två 8-bit och en 16-bit, JTAG-interface och realtidsklocka med separat oscillator. Program lagras i Flash EPROM och kan programmeras då kretsen är monterad i ett färdigt system.

4. Mjukvaran

4.1 Programmet

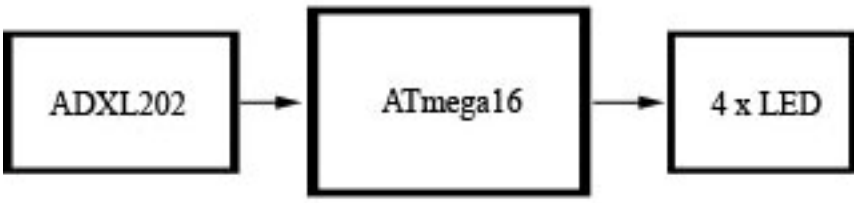
Atmega16 programmeras genom ett eget interface som kallas JTAG. Programmeringen kan ske även efter att processorn har placerats i det färdiga systemet. Programspråket är C och påminner mycket om Java. Vi använde oss av ett program som heter Studio 4 och det var väldigt smidigt att använda den för att implementera koden på chippen. Det tog dock ett tag innan vi förstod oss på Studio 4 ordentligt.

4.2 Förklarning av koden

ADXL202 sänder ut två olika fyrkantssignaler. En signal är för Y-ledd och en för X-ledd. Dessa tas emot i AVR:n separat, X-ledd på portD pin 5 och Y-ledd på portD pin 4. Först måste signalerna kalibreras i de olika lederna där utgångsläget skall vara för de olika mätningarna. Så ifall man vill ha huvudet lite framåt lutad så skall detta sättas till utgångsvärdet för mätningarna. Det görs genom att vi trycker ”reset” knappen och ser lysdioderna tändas till och då är den kalibrerad efter det läget som ADXL:n är i. Då utgångsläge för mätningarna är satta så börjar man att röra på huvudet. Då kommer de olika signalerna att ändras beroende åt vilket håll huvudet ändras åt. Hur mycket huvudet har flyttats så skall de olika lamporna tändas. Att få fram hur mycket huvudet har förflyttat sig görs genom att sampla den höga och låga pulsen. Där efter beräknas differensen mellan dem vilket visar vilken som är den dominerande och hur dominerande den är. Dessa värden används senare för att se hur mycket huvudet var flyttat och åt vilket håll. Därefter tänds lysdioden eller lysdioderna för den riktning huvudet hade förflyttats mot.
Då sampling sker så samplas inte båda signalerna samtidigt utan det kan vara så att vi missar en våglängd ibland. Detta gör inget ansåg vi eftersom ADXL:n sänder ut med en snabb frekvens. Så det finns gott om våglängder att mäta på. Kalibrering sker endast då ”reset” knapp är nedtryckt.

5. Konstruktion

Hjärnan i vår konstruktion är Atmega16 processorn. Till den har vi kopplat en sensor som ska känna av olika rörelser. Denna sensor är en accelerometer. För att läsa av åt vilket håll accelerometern vinklas använder vi oss av fyra lysdioder, en för upp, en för ner och ytterligare två för höger och vänster.



Accelerometer **enchipsprocessor** **lysdioder**

Som blockschemat visar så är accelerometern kopplad till processorns ingång och som utgång från processorn har vi fyra LED.
Funktionsmässigt är tanken att man fäster accelerometern på en lämplig plats, förslagsvis huvudet. Sedan trycker man in reset knappen för att nollställa sensorns läge. Detta gör vi för att om man eventuellt skulle hålla sensorn snett i grundställningen så ska inte mjukvaran tro att den är vinklad till exempel neråt hela tiden. Sedan ska processorn hela tiden läsa av accelerometerns position, jämföra med grundläget och utefter det tända lämplig lysdiod.

6. Problem som uppstått

Att finna den rätta signalen ur ADXL:n var inte så svår att tyda då vi med hjälp av ett oscilloskop fick se hur kurvan såg ut och hur den reagerade vid olika rörelser.
Då vi startade programmera så fick vi inte lysdioderna till att tändas vilket var ett stort problem trodde vi men då vi tittade lite noggrannare så var problemet mindre. Diodens anod och katod satt på fel håll vilket förklarade problemet att den inte villa lysa.
Därefter startade programmering för att använda avbrottsfunktion ICP (Interrupt Capture Pin). Vi kopplade in X signalen här för att börja någonstans. Den fungerade utmärkt och avbrott genererades då vi rörde på huvudet. Dock blev det problem då vi kopplade in Y signalen och trodde vi skulle kunna använda en Timer Interrupt för denna signal. Dessa krockade med varandra och vi fick endast den ena signalen till att fungera som satt på ICP pinnen. Då kom vi fram till det beslut som vi kändes vara rätt att sampla signalen. Vilket var ett nytt tänkande. Men det gjorde det lättare att få kontroll över de olika signalerna på ett mer lättbegripligt sätt. Här fanns även lite små problem som kröp in och det största var att vi inte fick in någon bra samplingstid utan den var för snabb. Då var det bara att sätta igång med oscilloskopet och mäta och se med hjälp av lysdioderna vilken tid som var den bästa för att få de resultat vi ville ha.

7. Komponenter

processor	AVR ATmega16
accelerometer	ADXL202
återfjädrandekontakt	
diverse resistorer och kondensatorer	
lysdioder	

8. Källor

http://www.it.lth.se/courses/digp/information.asp

http://www.it.lth.se/datablad/sensors/ADXL202.pdf

http://www.it.lth.se/datablad/Processors/ATmega16.pdf

http://www.it.lth.se/digp/libc-manual/index.html

http://www.avrbeginners.net/

http://www.it.lth.se/digp/libc-manual/index.html

http://www.elfa.se/se/index0.html

http://www.it.lth.se/digp/sammanfattning/default.asp

Samtliga länkar är aktuella mars 2005.

Ett stort tack till Bertil Lindvall och Carl Hakenäs för deras hjälp i detta projekt.