



DANCING ROBOT

EITF11, Grupp 10

Kajsa Salomonsson, Hugo Hedin och Petter Mårtensson

Abstract

As part of the course Digital Projects EITF11 at the Faculty of Engineering, Lund University, a prototype of a dancing robot has been constructed. Firstly a circuit diagram was produced and then all components connected on a breadboard. After testing that all connections were working coding begun simultaneously with testing of the code. As issues came up they were either solved or worked around arriving at a robot able to differentiate between three intervals of beats and performing a dance corresponding to each interval after listening for 8 beats, changing which LED is lit for every beat.

Innehållsförteckning

Inledning	3
Bakgrund	3
Problemformulering	3
Kravspecifikation	3
Hårdvara	3
Processor	3
Mikrofon	4
JTAG	4
Lysdioder	4
Motorer	4
H-brygga	4
Spänningsomvandlare	4
Spänningsförsörjning - Batteri	4
Spänningsförsörjning - Nätaggregat	4
Resistorer	5
Mjukvara	5
Arbetsprocess	5
Planering	5
Konstruktion	5
Programmering	6
Resultat	6
Diskussion	6
Vidareutveckling	7
Slutsats	7
Referenslista	8
Bilagor	9
Kopplingsschema	9
Källkod	9

Inledning

Bakgrund

Kursen Digitala Projekt ges som en del av teknikprofilen System- och programvaruutveckling på programmet Industriell Ekonomi. Avsikten är att ge eleverna en känsla för hur konstruktionsarbete går till och självständigt producera en fungerande prototyp med tillhörande dokumentation.

Problemformulering

Vår infallsvinkel på uppgiften var att sätta kopplingsbrädet med komponenter på hjul och med hjälp av en mikrofon få konstruktionen att reagera olika på ljud olika med olika takt, i praktiken en dansande robot.

Kravspecifikation

En robot som rör sig enligt ett förutbestämt mönster samtidigt som lysdioder blinkar i takt med musik som spelas i rummet.

- När musiken börjar ska roboten börja röra sig.
- När musiken slutar ska roboten sluta röra sig.
- Lysdioder kopplade till roboten ska kunna blinka i takt till musiken.
- Roboten ska kunna röra sig i takt till musiken.
- Roboten ska kunna röra sig fram och tillbaka samt kunna svänga enligt ett förutbestämt mönster.

I mån av tid:

- Roboten har en mun av en display som är ledsn om roboten inte hör musik och skiftar snabbt mellan ett streck och en cirkel (sjunger) om den hör musik.
- Roboten har en glad mun i 10 sekunder efter att den har slutat dansa.
- 4 ansikten: neutral, ledsn, glad, sjungande

Hårdvara

Processor

För att kunna styra robotens olika komponenter används en programmerbar processor. Processorn som används i roboten är ATmega16 8-bitars mikrokontroller med ett 16 kb flashminne och 40 I/O pinnar, varav 32 finns till förfogande för programmering. Av de 32 programmerbara pinnarna så finns det 8 st var av typerna A, B, C och D.

Alla programmerbara pinnar kan användas för att ta emot eller sända digitala signaler, men pinnarna har också vissa specialfunktioner. De specialfunktioner som vi har använt oss av är att pinnarna av typ A kan användas för A/D-omvandling och pinnarna av typ C används för att koppla in JTAG.

Mikrofon

En mikrofon används för att fånga upp ljudvågor. Vi har använt oss av en kondensatormikrofon som i praktiken är en kondensator som vi har spänningssatt på 2,5 V. När den träffas av ljudvågor så kommer kondensatorns spänning att variera och denna spänning tar vi in i processorn som en analog signal på en av A-portarna där den sedan omvandlas till ett digitalt värde, dvs en så kallad A/D-omvandling sker.

JTAG

För att kunna föra över vår programmerade programvara från en stationär dator till processorn, samt för debugging, så används interfacet JTAG som kopplas från datorns USB-uttag till C-portarna på processorn.

Lysdioder

För att sända ut ljus från roboten så används lysdioder. Detta är komponenter som bara kan leda ström i en riktning och när ström leds i denna riktning så avges ljus i en viss färg. Vi har använt oss av lysdioder i färgerna rött, gult, grönt och orange som vi har kopplat till B-portarna på processorn.

Motorer

Roboten drivs framåt av två larvfötter, en på höger sida och en på vänster sida. I varje larvfot finns en elektrisk motor som helst arbetar under spänningen 6 V, men de klarar även av att arbeta under spänningen 5 V. Beroende på vilket håll som strömmen flödar så kan motorerna köras både framåt och bakåt.

H-brygga

För att kunna styra motorerna med en kraftigare ström än den från processorn så används en H-brygga med namn L298. Den tar emot den spänning som används för att driva motorerna och om komponenten släpper vidare spänningen till motorerna så kommer motorna att köra. Huruvida spänningen ska släppas vidare eller inte, samt vilket håll motorna ska drivas åt styrs av signaler från processorn in till H-bryggan. Vi har kopplat H-bryggan till D-portarna på processorn.

Spänningsomvandlare

Batteriet har spänningen 6 V, vilket även motorerna arbetar under. Processorn däremot har arbetsspänningen 5 V. För att kunna driva processorn så behöver därför den inkommande spänningen omvandlas från 6 V till 5 V och till detta används spänningsomvandlaren MCP1827.

Spänningsförsörjning - Batteri

För att kunna roboten ska kunna köra utan några inkopplade sladdar används ett batteri som ger ifrån sig likström med spänningen 6 V.

Spänningsförsörjning - Nätaggregat

För att kunna använda roboten under utvecklingsfasen utan att behöva använda batteri så används ett nätaggregat som ger ifrån sig likström med spänningen 5 V.

Resistorer

För att på lämpliga ställen i konstruktionen kunna begränsa strömmen för att till exempel undvika att skada komponenter eller för att kunna nå en viss funktion så används resistorer med lämpliga resistanser.

Mjukvara

Programmet skrevs i utvecklingsplattformen ATMEL Studio i programspråket C. JTAG används för att ladda upp kod till processorn och för att felsöka och testa programmet.

Koden består av en main-metod och två avbrott som sker självständigt från annan kod. Det första avbrottet sker när en timer går ut som startar när programmet startar. I den kod som då körs börjar en A/D-omvandling av signalen som läses från mikrofonen. När denna omvandling är färdig börjar timern om och samtidigt går det andra avbrottet igång som gör värdet från mikrofonen användbart i main-metoden och gör programmet redo för en ny omvandling. I main-metoden räknar processorn tiden för åtta slag och låter en ny lysdiod lysa för varje nytt slag. Efter dessa åtta slag körs ett av tre förutbestämda körschema, vilket beror på om takten var under 60 slag i minuten, mellan 60 och 120 eller över 120. Detta körschema varar lika länge som den tid det tog att komma upp i 8 slag. När det är över körs allt om från början.

Arbetsprocess

Planering

Under första veckan av kursen samlades alla deltagare och lade upp en plan för hur arbetet skulle gå till. Vilken konstruktion som skulle byggas och en övergripande tidsplan bestämdes samt vilka krav vi hade på prototypen. Det bestämdes även att alla i gruppen skulle vara insatta i alla delar av projektet.

Konstruktion

Efter att ha bestämt vilken prototyp som skulle byggas konstruerades ett kopplingsschema med de komponenter vi bestämt oss för att använda med hjälp av programmet PowerLogic. Eftersom vi byggde en prototyp användes ett bräde där man kan flytta komponenter och i de flesta fall virades komponenter istället för att lödas fast. I de fall då det skulle gå en kraftigare ström genom ledningen, i praktiken alla ledningar med strömförsörjning och jord till motorerna och voltomvandlaren, så användes en kraftigare ledning som fick lödas fast.

För att testa och felsöka användes en logikpenna som gör att man kan se om det är en hög eller låg signal på en ledning. Efter att ha sett att alla ledningar hade rätt spänning testades om mikrofonen gav utslag för ljud med hjälp av ett oscilloskop som visade hur signalen förändrades då mikrofonen tog emot ljud.

Spänningsomvandlaren började av okänd anledning att bli konstant överhettad vilket ledde till att den efter ett antal sekunder stängdes av. Efter mycket felsökning byttes omvandlaren ut och ersattes med en ny som fungerade, men efter en kortare tidsperiod började även denna omvandlare drabbas av samma fel. Eftersom att det inte fanns något enkelt sätt att felsöka detta så togs beslutet att koppla bort spänningsomvandlaren och köra hela konstruktionen med 5V spänning från nätaggregatet.

Programmering

För att ytterligare testa om hårdvaran var rätt kopplad skrevs några enkla rader kod i ATME Studio med syfte att testa om vi kunde tända LED, starta motorerna och om vi kunde läsa värdet mikrofonen gav oss och vilket detta var för typisk musik.

För att tidsavbrotten som satte igång A/D-omvandlaren skulle ha en lagom frekvens skalades frekvensen på timern ner med 8. Samtidigt sattes räknarens startvärde till 150, vilket gör att den räknar 96 steg upp till sin timeout som sker efter 256 steg, och då börjar den om igen. Detta ger oss 2604 tidsavbrott i sekunden. För att lamporna inte skulle tändas och släckas snabbare än ögat hinner uppfatta, samt för att undvika att fler än en lampa hinner tändas under ett och samma slag sattes även ett krav på att en ny lampa inte kan tändas förrän efter 50 tidsavbrott.

För att en lampa ska tändas måste värdet från mikrofonen vara mindre eller lika med ett visst tröskelvärde. Detta bestämdes genom att använda debuggern för att se i vilken storleksordning insignalerna var och sedan genom att prova oss fram för att se vilket värde som fungerade bäst.

Våra två gränser för snabbt respektive långsam takt räknade vi ut genom genomsnittligt antal tidsavbrott per slag. Antal avbrott per sekund är 2604. 60 bpm (beat per minute) motsvaras då av 2604 avbrott per slag, och 120 bpm motsvaras av 1302 avbrott per slag. Dessa värden visade sig ge något fel gränser och nära intervallgränserna kördes fel program. Genom att prova oss fram kom vi fram till att vi kom närmare de önskade gränserna om vi justerade 2604 till $2604 \cdot 0,9$ och $1302 \cdot 0,8 + 50$.

Resultat

Vårt arbete resulterade i en robot som kan höra starka ljud nära mikrofonen och med hjälp av dessa räkna ut takten och köra ett av tre olika program beroende på ljudets tempo. Den initiala idén om att plocka ut takten ur musik visade sig vara mycket komplicerad och därför har vi fokuserat på att få roboten att köra i takt till en metronom.

På grund av problem med voltomvandlaren så körs roboten enbart med ström från nätaggregat, dvs ej batteri.

Ett okänt fel har gjort att roboten måste ha JTAG inkopplad för att programmet ska köras på rätt sätt.

Diskussion

I början togs inte ljudet ifrån motorerna med i beräkningen. Det blev dock snabbt tydligt att detta ljud togs upp av mikrofonen och hade en stark påverkan på signalen, vilket ledde till att vi inte kunde lyssna på signalen samtidigt som motorerna kördes. Vi löste detta genom att variera mellan att roboten lyssnar på takten och att den kör omkring i ett förprogrammerat mönster enligt den senast avlyssnade takten. Den här metoden fungerar bra då alla taktslag avlyssnas, men om något eller några taktslag inte avlyssnas blir uträkningen av takten inte korrekt vilket påverkar körmönstret.

Olika metoder för att filtrera ut bastonerna ur musik som spelas har diskuterats, då detta var vår ursprungliga vision. Med den begränsade utrustningen, förkunskaperna samt tiden vi hade kom vi dock inte fram till någon genomförbar lösning.

Vidareutveckling

Det finns ett antal förbättringar som bör kunna göras utan några större insatser. Det första är att få konstruktionen att fungera utan att vara inkopplad med JTAG. Det andra är att hitta felet till att spänningomvandlaren blir överhettad och därefter koppla in den igen. Dessa problem bör inte uppstå ifall man bygger roboten igen med fräscha komponenter från grunden. I så fall har vi en robot som fri från sladdar kan dansa.

I nuläget reagerar roboten på starka ljud i allmänhet och inte på musikens takt i synnerhet som var ett av våra ursprungliga krav. Eftersom vi har en så svag mikrofon fungerar det bra med en metronom eftersom den endast tar in ljud från den och är för svag för att störas av ljud runt om (med undantag för motorerna). Till skillnad från om den kör till vanlig musik då den reagerar på andra starka toner än bara bastonerna. Det hade varit önskvärt att ha en mikrofon som är känsligare så att man kan ha högtalaren på ett längre avstånd än cirka två cm från mikrofonen. Detta hade som sagt även krävt att koden utvecklats ytterligare med metoder för att urskilja takt och inte reagera på alla ljud. Kanske skulle ett lågpasfilter kunna användas för att försöka att endast lyssna på basen.

Ett annat område för vidareutveckling är taktkänsligheten. Denna skulle kunna utvecklas så att roboten känner igen olika takter med bättre precision, och fler taktintervall än tre. Dansmönstrena skulle även kunna utvecklas, varieras och anpassas bättre för de olika takterna.

Slutsats

Projektet har gett oss en förståelse för arbetet som ligger bakom att ta fram nya produkter och hur man kan lägga upp arbetet. Framför allt har vi förstått hur mycket som kan gå fel och vikten av tydlig dokumentation och effektiv felsökning. Kursen är uppbyggd på ett något annorlunda sätt jämfört med andra då meningen är att själv finna kunskap utan kurslitteratur och på ett iterativt vis prova, misslyckas, tänka till och lyckas. Det är även uppfriskande att den är så mycket mer praktisk än mycket annat vi har läst.

Referenslista

Skriftliga

Datablad för ATMega16

<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Processors/ATmega16.pdf>

Datablad för H-brygga

<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Analog/linear/l298.pdf>

Datablad för voltregulator

<http://www.eit.lth.se/fileadmin/eit/courses/edi021/datablad/Analog/voltage/mcp1827.pdf>

Muntliga

Bertil Lindvall

Bilagor

Kopplingsschema

Se hemsida.

Källkod

Se hemsida.