



LUNDS TEKNISKA
HÖGSKOLA
Lunds universitet

Institutionen för
Informationsteknologi

Digitala projekt 5p HT 2004

Telephone controlled mouse

EMIL SÖGAARD e03es
PIERRE FRÖJD e03pf

Abstract

The possibility to communicate and control devices remote via an ordinary telephone can have many advantages e.g. intelligent home. Make a phone call and turn on the heater in your winter cottage the day before you arrive.

This project report describes how you can control your PC mouse via any telephone. Although controlling the mouse has no real practical usage, it's an example that shows the course of action to implement such a system. Another example of a system that is controlled via a telephone is today's "telephone banking".

This project includes choice of components, source code and the work in general.

Innehållsförteckning

ABSTRACT	2
1 INLEDNING	4
2 KRETSENS KOMPONENTER	4
2.1 DTMF RECEIVER.....	4
2.2 PS/2 MUS	5
2.3 RELÄ.....	6
2.4 DISPLAY.....	6
2.5 ATMEGA16	6
3 MJUKVARAN	7
3.1 UTVECKLINGS MILJÖN	7
3.2 KÄLLKODEN	7
3.3 MAIN.C	7
3.3.1 <i>Init_interrupt()</i>	7
3.3.2 <i>Interrupt(sig_output_compare1a)</i>	7
3.3.3 <i>Mus aktivitet</i>	8
3.3.4 <i>Main()</i>	9
3.4 DISPLAY.C	9
3.4.1 <i>Init_Display()</i>	9
3.4.2 <i>setScrollMode()</i>	10
3.4.3 <i>setNormalMode()</i>	10
3.4.4 <i>resetDisplay()</i>	10
3.4.5 <i>writeChar()</i>	11
3.4.6 <i>changeRow()</i>	11
3.4.7 <i>getCode()</i>	11
3.4.8 <i>printWelcome() / printReady() / printLeft / Right / Up / Down</i>	11
4 RESULTAT OCH SLUTSATSER	11
APPENDIX 1 KRETSSCHEMA	12
APPENDIX 2 SPECIFIKATION AV DTMF-SIGNALERING	13
APPENDIX 3 TECKENKODER FÖR DISPLAY	15

1 Inledning

Vår idé till kursen ”Digitala projekt” var skapa logik för att styra mus och tangentbord via en vanlig telefon. Under projektets gång fick vi dock begränsa oss till att bara styra musen. Musen styrs via knappsatsen på en telefon och de grundläggande musfunktionerna är implementerade. En display visar vilket nummer som har ringt upp datorn, samma display visar även det senaste kommandot som skickats till datorn. Denna rapport kommer att beskriva arbetet bakom projektet samt de ingående komponenterna och källkoden.

2 Kretsens komponenter

Vår krets baseras på en AVR 8-bit Mikrocontroller från Atmel. Microcontrollern är ”hjärnan” och sköter kommunikationen med övriga kretsar. Eftersom projektets omfattning täcktes väl av enchipsdatorn så behövde vi aldrig fundera på att använda en mikroprocessor med tillhörande kring kretsar.

2.1 DTMF receiver

De DTMF-toner som kan skickas över telenätet representeras av knapparna på en vanlig telefon samt bokstäverna A, B, C och D. Varje DTMF-ton är sammansatt av två stycken frekvenser, en radfrekvens och en kolumnfrekvens enligt tabellen nedan:

DTMF		Kolumner			
		1209 Hz	1336 Hz	1447 Hz	1633 Hz
Rader	697 Hz	1	2	3	A
	770 Hz	4	5	6	B
	852 Hz	7	8	9	C
	941 Hz	*	0	#	D

För att avkoda dessa frekvenser till motsvarande siffra eller tecken används en färdig IC från Mitel med beteckningen MT8870DE. IC-kretsen är implementerad med ett differentialsteg på ingången samt två kondensatorer som skydd mot likspänningar. Kretsen använder sig av en extern kristall på 3.57 MHz, denna frekvens krävs för att DTMF-avkodaren ska fungera korrekt. Eftersom det finns 16 olika kombinationer av tecken använder kretsen sig av 4 utgångar för att representera dessa.

2.2 PS/2 mus

Dagens datorer använder sig numera av PS/2 kommunikation framför den lite äldre RS-232 standarden. PS/2 standarden är ett tvåvägs synkront seriellt protokoll som har en data bus och en klocka (öppen kollektor). Protokollet består av 11-12 bitar.

- 1 start bit. Denna är alltid 0.
- 8 data bitar, minst signifikant bit först.
- 1 paritets bit (udda paritet).
- 1 stop bit. Alltid 1.
- 1 acknowledge bit (dator-till-mus/tangentbords kommunikation)

Det finns tre tillstånd på busarna

1. Data = hög, Clock = hög: Redo.
2. Data = hög, Clock = låg: Kommunikation förbjuden.
3. Data = låg, Clock = hög: Datorn, förfrågan-att-sända.

Kortfattat så går kommunikationen till så att musen genererar klockpulser. Datorn har absolut kontroll över kommunikationen och kan när som helst dra klockan låg och förbjuda kommunikation. När datorn vill sända till musen drar datorn klockan låg sedan dras data låg. Efter detta släpper datorn klockan, detta signalerar till musen att börja genera klockpulser.

Datorn läser data-busen på fallande klockflank och musen läser data-busen på stigande flank.

Klockfrekvensen måste ligga i intervallet 10-16.7 kHz detta medför att klockan ska ligga låg respektive hög i 30-50 μ s. Data busen ska samplas cirka 15-25 μ s efter varje tillståndsbyte på klockan. Detta medför att timingen är mycket kritiskt.

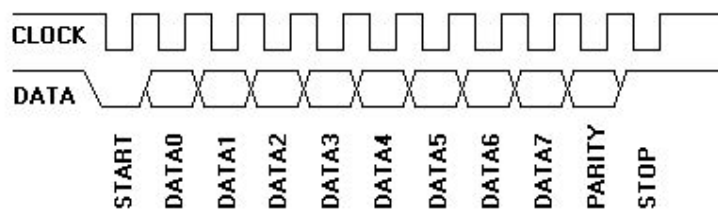


Fig 1. Mus till datorn kommunikation. Data ändrar tillstånd när klockan är hög och är tillgänglig när klockan går låg.

Eftersom tiden var begränsad för projektet valde vi en färdig krets för att implementera musgränssnittet. Kretsen som valdes kommer från Holtek och heter HT82M398A.

För att styra musen använder kretsen sig av fyra ingångar. Två för x-led och två för y-led. Beroende hur de två ingångarna tilldelas logiska ettor rör sig musen upp/ner respektive vänster/höger. För att detektera knapptryck finns dedikerade portar till detta.

2.3 Relä

För att kunna ringa upp datorn och styra densamme behövs att vår krets ”svarar” när det ringer. Att lyfta luren när det ringer motsvaras av ett motstånd på cirka 600 ohm kopplas in mellan linjerna. När microcontrollern har fått det sista tecknet från DTMF-receivern (alltid ett ”C”) skickas en styrsignal till reläet som slår om och kopplar in motståndet.

2.4 Display

Som nämdes i inledning skulle en display visa vilket nummer som ringt upp datorn. Valet föll på en alfanumerisk display med två rader från Sharp <http://www.it.lth.se/datablad/>. Displayen styrs helt och hållet av microcontrollern. När microcontrollern har fått ett giltigt tonpar från DTMF-receiver skickas det direkt ut till displayen. Någon mellanlagring av telefonnumret i microcontrollern behövs inte göras för att kommunikationen mellan controllern och displayen går relativt snabbt. Displayen visar också vilket kommando som skickats. Skickas samma kommando uppdateras inte displayen. När programmet startar visas en välkoms text följt av en text som indikerar att kretsen är redo.

2.5 ATmega16

Processorn är en s.k. enchips datorn med diverse inbyggda funktioner såsom A/D-D/A, SPI, EEPROM och avbrotts hantering etc. I detta projekt används främst timerstyrt avbrott för att periodiskt kontrollera om ny DTMF-ton mottagits, d.v.s. ett knapptryck. ATmega16 har fyra block med vardera 8 portar som är individuellt konfigurerbara.

- PortA[0]-[4] är anslutna till DTMF-avkodare
- PortA[7] går till enable signalen på displayer
- PortB kommunicerar med displayen
- PortD[1]-[6] styr muskretsen
- PortC[0]-[1] går också till displayen
- PortC[2]-[7] är kopplat till Jtag Ice

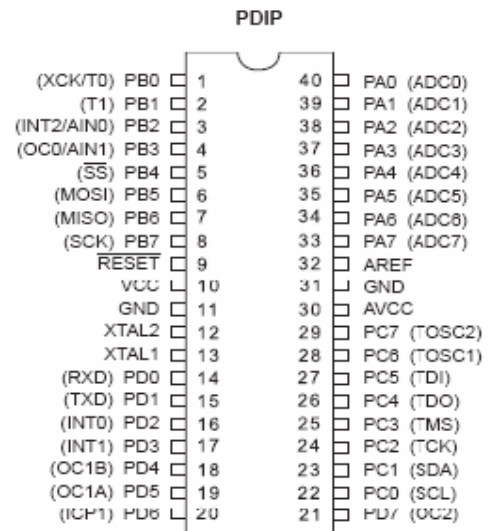


Fig 2. ATmega16

3 Mjukvaran

3.1 Utvecklings miljön

För utveckling av programmet valdes AVR JTAG ICE. JTAG är ett programmerings och debuggnings verktyg som styrs via AVR studio 4. För att skriva själva C koden valdes ”Programmers Notepad” med en AVR GCC kompilator.

3.2 Källkoden

Källkoden är skriven i C, och består av två delar: Main.c och Display.c. I Main.c initieras portar och avbrott, dessutom körs main-loopen som håller igång programmet. I Display.c finns funktioner för att initiera, återställa och att skriva tecken på displayen. Det finns också två header filer importerade: delay.h och bool.h, den första används vid tidskritiska program delar där man måste vänta på att hårdvara skall hinna göra färdig ett visst arbete innan man kan fortsätta, den senare krävdes då GCC kompilatorn inte kände till booleska variabler.

Nedan följer en beskrivning av de funktioner som implementerats samt hur portar och avbrott är initierade på AVR processorn.

3.3 Main.c

3.3.1 Init_interrupt()

Denna funktion initierar ett timer styrt avbrott (OCR1A), med ”normal” mode på counter och prescaler satt till noll. Funktionen aktiverar även biten ENABLE INTERRUPTS i registret SREG, vilken är ett måste för att aktivera avbrott.

3.3.2 Interrupt(sig_output_compare1a)

Denna funktion fungerar som kärnan i programmet, här kontrolleras om nya knapptryck inträffat och vad som i så fall skall utföras.

Varje gång som räknaren i AVR och det inställda värdet i outputcompare registret överensstämmer, aktiveras denna funktion. Här läses ”STD” (steering delayed) värdet från DTMF-avkodaren, vilket blir sant i några millisekunder om en ny DTMF-ton mottagits. För att undvika ”knappstuds” används en variable ”bounce” som förhindrar displayen att skriva ut ”studstecken”, dock är knappstudsar en önskad effekt då det

gäller att kontrollera musen i x,y-led (musen rör sig mjukare), varav inget studsskydd används här.

Om processorn inte mottagit avslutnings tonen ”C” på nummer presentationen så körs en ”writeChar” funktion som skriver ut varje ny ton på displayen. Då ”C” tonen mottagits exekveras istället mus flyttning/klickning samt en utskrift till displayen. Kommandona visas i tabell 1.

Knapp	Kommando
0	Dubbel klick
1	1:a tecken för avslutning
2	Mus upp
3	Ej implementerad
4	Mus Vänster
5	Ej implementerad
6	Mus Höger
7	Ej implementerad
8	Mus Ned
9	Ej implementerad
*	Vänster klick
	Höger klick
#	eller 2:a tecknet för avslutning

Tabell 1.

3.3.3 Mus aktivitet

Här ingår alla funktioner som påverkar musen.

- doubleClick()
- leftClick()
- rightClick()
- moveDown()
- moveUp()
- moveLeft()
- moveRight()

Dessa funktioner skickar styrsignaler till ”mus-kontrollern” HT82M398A vilken i sin tur skickar styrsignaler till PC med PS/2 gränssnitt.

För djupare förklaring av styrsignaler se datablad för HT82M398A, eller <http://www.holtek.com/>

3.3.4 Main()

I Main initieras B-blocket, D-blocket, PA7, PC0 och PC1 som utportar, och övriga som inportar. Avbrottet initieras och aktivers med `init_interrupt()` och `sei()`; Då DTMF-tonen "C" mottagits sluts reläet och samtalet är nu uppkopplat, och all kontroll av nya DTMF-toner sker i avbrottet.

3.4 Display.c

Denna programdel är främst skriven för att få en bättre översikt av programmet både för programmerare och för granskare. Då det krävs en väldig massa upprepningar för att ställa displayen i korrekt läge, och för att skriva ut en text sträng. För att skicka kommando/tecken till displayen måste en viss ordning och tidskrav följas:

1. Vid kommando
 - Lägg ut kommandot på porten
 - RS = R/W = 0
 - Enable = 1
 - Enable = 0
 - Vänta minst 40us innan nästa kommando
2. Vid teckenutskrift
 - Lägg ut Koden för tecknet på bussen
 - RS = 1, R/W = 0
 - Enable = 1
 - Enable = 0
 - RS = 0
 - Vänta minst 40us innan nästa tecken skrivs

3.4.1 Init_Display()

Denna funktion initierar displayen, samt ställer den i standardläge. Detta görs enligt:

- Function Set (0x38) vilket ställer in datalängden till 8 bitar och 2 radig display.
- Display ON/OFF (0x0E): rensar minnet.
- Entry Mode Set (0x06): ställer in addr för cursorn till standard (0)
- Display clear (0x01): rensar displayen
- Display cursor home (0x02): återställer cursorn från bl.a. "shift-mode"

3.4.2 setScrollMode()

Här ändras positionen på cursorn till sist på första raden, ”shift enable” medför att texten kommer att skiftas in från höger till vänster. Detta utförs enligt:

- Display Cursor Shift (0x07): Aktiverar shift mode
- DD Ram Addr Set (0x8F): positionerar cursor längst till höger

3.4.3 setNormalMode()

Återställer displayen i standard läge, dvs. cursorn placeras längst till vänster på första raden, och ”shift enable” avaktiveras. Detta utförs enligt:

- Display Clear (0x01)
- Entry Mode Set (0x06)
- DD Ram Addr Set (0x80)

3.4.4 resetDisplay()

Rensar både display och minnet. Återställer också displayen till standardläge.

- Display Clear (0x01)
- Display ON/OFF (0x0E)
- Entry Mode Set (0x06)

3.4.5 writeChar()

Skriver ut tecken till displayen enligt appendix 3.

3.4.6 changeRow()

Byter från första raden på displayen till den andra.

- DD Ram addr Set (0xC0)

3.4.7 getCode()

Returnerar display kod för siffror 0-9. (se datablad).

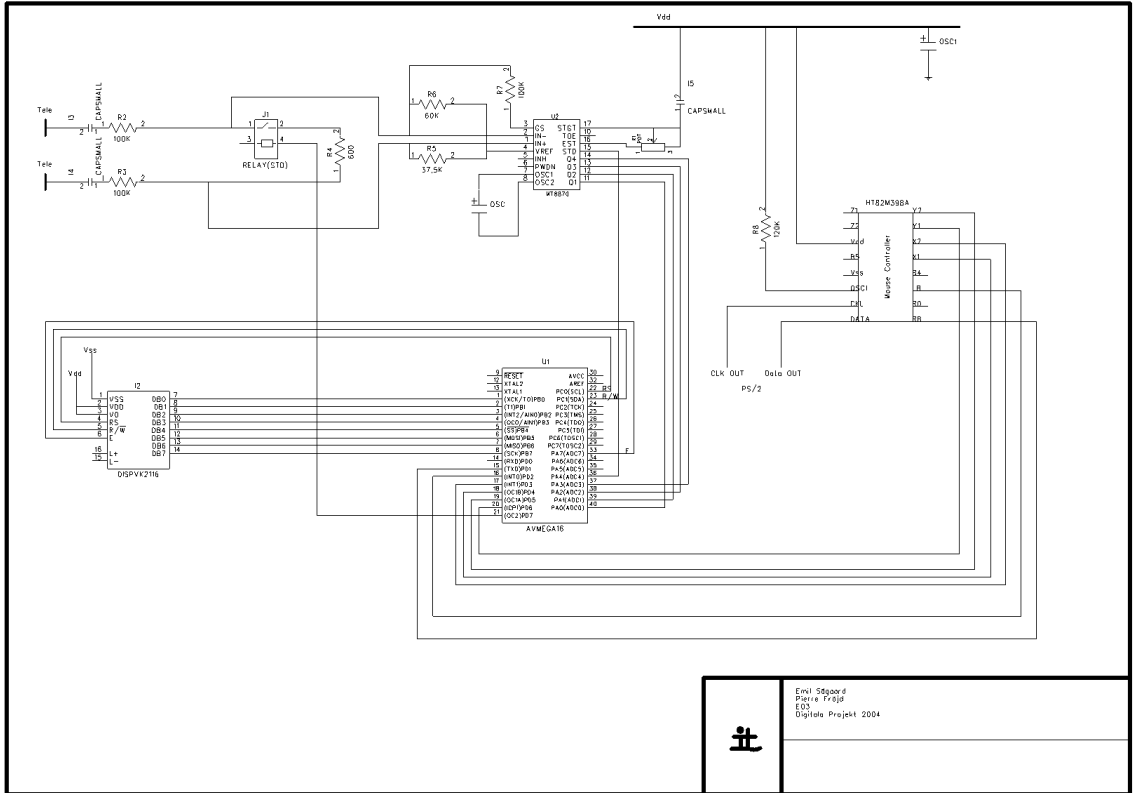
3.4.8 printWelcome() / printReady() / printLeft / Right / Up / Down

Skriver ut respektive text på displayen.

4 Resultat och slutsatser

Under projektets utveckling har en del problem uppstått. Bland annat fick vi inte keyboard kontrollern att fungera, vi kontaktade supporten via email, men vi fick inget vettigt svar. I slutet av projektet försökte vi få systemet att själv ”koppla från” då en polväxling inträffade (A-abonment lagt på), detta ledde dock till att vi någonstans jordade telelinjerna och AXE växeln bröt linjen i 20 min. På grund av tidsbrist hårdkodades en avslutnings kombination (1,#) för att på så sätt koppla från reläet och lägga på luren. I övrigt har vi konstruerat en fungerande prototyp.

Appendix 1 Kretsschema



Emil 599004
Pohle 57498
EO
Digitale Projekt 2004

Appendix 2 Specifikation av DTMF-signalering

Överföring av nummerinformation sker i enlighet med följande signaleringssekvens:
(ApS1pS2pS3p..SnP) (DpS1pS2pS3p..SnP) (DpS1pS2pS3p..SnP) (BpS1pS2p) C

Element inom parantes är inte obligatoriska.

A = Startsignal för den anropandes nummer (tonvalssignal 697/1633 Hz)

B = Startsignal för informationskod (tonvalssignal 770/1633 Hz)

C = Stoppsignal (tonvalssignal 852/1633 Hz)

D = Startsignal för omstyrt/anropandes nummer (tonvalssignal 941/1633 Hz)

S1-Sn Siffror (0-9)

p = Paus mellan varje tecken

n = Heltal mellan 1-15

Nominell sändningshastighet är 70 ms/tecken, 70 ms/paus. Mottagande utrustning bör dock kunna detektera tecken/paus ned till 48/48 ms.

De överdekadiska siffrorna ("A", "B", "C", "D") är normalt inte kända av abonnenterna och bör därför inte presenteras. Om flera nummer överförs så åtskiljs dessa av "D".

"C" är alltid den sista informationen som sänds.

De DTMF-toner som kan skickas över telenätet representeras av knapparna på en vanlig telefon samt bokstäverna A, B, C och D. Varje DTMF-ton är sammansatt av två stycken frekvenser, en radfrekvens och en kolumnfrekvens enligt tabellen nedan.

DTMF		Kolumner			
		1209 Hz	1336 Hz	1447 Hz	1633 Hz
Rader	697 Hz	1	2	3	A
	770 Hz	4	5	6	B
	852 Hz	7	8	9	C
	941 Hz	*	0	#	D

Följande tabell visar vad som kommer ut ur DTMF-avkodaren MT8870 när man matar den med olika signaler på ingångarna.

Binärkodning av DTMF-toner i MT8870				
f_low	f_high	Key	TOE	Q4-Q1
697 Hz	1209 Hz	1	1	0001
697 Hz	1336 Hz	2	1	0010
697 Hz	1477 Hz	3	1	0011
770 Hz	1209 Hz	4	1	0100
770 Hz	1336 Hz	5	1	0101
770 Hz	1477 Hz	6	1	0110
852 Hz	1209 Hz	7	1	0111
852 Hz	1336 Hz	8	1	1000
852 Hz	1477 Hz	9	1	1001
941 Hz	1209 Hz	0	1	1010
941 Hz	1336 Hz	*	1	1011
941 Hz	1477 Hz	#	1	1100
697 Hz	1633 Hz	A	1	1101
770 Hz	1633 Hz	B	1	1110
852 Hz	1633 Hz	C	1	1111
941 Hz	1633 Hz	D	1	0000
-	-	Any	0	ZZZZ

Appendix 3 Teckenkoder för display

High-Order Low- Order 4 bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)		0	@	P	\	p		-	タ	ミ	α	p
xxxx0001	(2)	!	1	A	Q	a	q	。	ア	チ	ム	ä	q
xxxx0010	(3)	"	2	B	R	b	r	「	イ	ツ	メ	β	θ
xxxx0011	(4)	#	3	C	S	c	s	」	ウ	テ	モ	ε	∞
xxxx0100	(5)	\$	4	D	T	d	t	,	エ	ト	ヤ	μ	Ω
xxxx0101	(6)	%	5	E	U	e	u	・	オ	ナ	ユ	σ	ü
xxxx0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)	、	7	G	W	g	w	ァ	キ	ヌ	ラ		π
xxxx1000	(1)	(8	H	X	h	x	イ	ク	ネ	リ	√	\bar{x}
xxxx1001	(2))	9	I	Y	i	y	。	ケ	ノ	ル	-1	y
xxxx1010	(3)	*	:	J	Z	j	z	。	コ	ハ	レ	j	
xxxx1011	(4)	+	;	K	[k	{	*	サ	ヒ	ロ	x	
xxxx1100	(5)	,	<	L	¥	l		+	シ	フ	ワ	¢	
xxxx1101	(6)	-	=	M]	m	}	*	ス	ヘ	ン	£	+
xxxx1110	(7)	.	>	N	^	n	→	。	セ	ホ	*	\bar{n}	
xxxx1111	(8)	/	?	O	_	o	←	ッ	ソ	マ	。	ö	■