

Formula Blue

Digitala Projekt 8p

Jesper Ferm E02

Carl Hakenäs E04



LUNDS TEKNISKA HÖGSKOLA
Lunds universitet

Abstract

The purpose with this project was to get some understanding how Bluetooth works and how to use it in practical applications. A suitable and fun project is to control a small car with a cell phone using Bluetooth.

This report describes the result and development of our project. Hardware and software explanation are also included.

2. Inledning	4
3. Specifikation	5
4. Komponenter.....	6
4.1 Processor	6
4.2 Minne	6
4.3 Logik	6
4.4 Drivkrets	6
4.5 Spänningsomvandlare	7
4.6 Multichipp.....	7
4.7 Kristalloscillator.....	7
4.8 Spänningsregulatorer	8
4.9 Bluetoothmodulen.....	8
5. Konstruktion	9
5.1 Mjukvara	9
5.1.1 Mobiltelefon.....	9
5.1.2 Bilen.....	10
6. Resultat / Problem.....	11
7. Sammanfattning	11

2. Inledning

Vi valde att styra en liten egenbyggd bil med en mobiltelefon via Bluetooth. Detta gav oss lite inblick i hur Bluetooth fungerar samt att vi lärde oss en del om hur man konstruerar datorsystem. Mobiltelefonen pratar med ett Bluetooth chip som i sin tur är anslutet till en mikroprocessor av typen Motorola 68008. Det roliga med projektet är att det innefattar hela kedjan från högnivå programmering till lågnivå hårdvara.

3. Specifikation

Projektet går ut på att vi vill få en fungerande uppkoppling mellan en mobil av märket Nokia 6600 som innehåller bluetooth och en bluetoothmodul som är placerad på en liten egentillverkad bil. Genom att kommunicera genom det trådlösa gränssnittet bluetooth vill vi kunna styra bilen från mobilen. Båda använder sig av en uppkoppling som heter UART Universal Asynchronous Receiver/Transmitter. Kommunikationen som sker genom standarden UART består av seriell kommunikation och det går att använda sig av fullduplex, men vi var enbart intresserade av att kunna skicka data från mobilen till bluetoothmodulen. För att kunna få en fungerande uppkoppling med bluetooth från mobilen utvecklade vi en programvara i språket J2ME som är snarlikt java fast en mer nerbantad version och som innehåller mer anpassade funktioner för den mobila marknaden.

4. Komponenter

4.1 Processor

Motorola 68008 microprocessor

För att få ut så mycket som möjligt av projektet använde vi oss av en processor som inte innehåller några extra delar.

Alla övriga komponenter blir man då tvungen att adressera. Själva adressbussen innehåller i sig 20 bitar vilket innebär att man kan adressera totalt 1 Mbyte. Datan som skickas mellan processorn och övriga komponenter sker via databussen, den har en storlek på 8 databitar. För att kommunikation överhuvudtaget ska kunna fungera på data bussen måste komponenter kunna gå i tri-state vilket betyder att enheterna blir höghögiga när det inte är adresserade.

4.2 Minne

- EPROM 27C64 8K x 8 Bit Parallell EPROM
- RAM 6264 8K x 8 Parallell Bit CMOS SRAM.

Programmet som processorn körs på är lagrat i ett EPROM på 8 kbyte.

EPROMet är återskrivningsbart, genom att man belyser det så raderar man datan. Vi byggde även in ett ram-minne på kortet för att kunna lagra data ifall vi ville utöka projektet med ytterligare funktioner.

4.3 Logik

PALCE22V10

För att få komponenterna att gå i tri-state använder man sig av programmerbara logik kretsar. Genom att man matchar rätt adressering med rätt komponent i logikkresten så bestämmer man vilken komponent man vill ha kontakt med.

Vi använde även en logikkrets som enbart hade hand om avbrotten som genereras från multichippet.

4.4 Drivkrets

L298 Dual full-bridge driver

För att kunna driva motorerna använde vi oss av en drivkrets som klarar av att leverera tillräckligt med ström. Denna styrs i sin tur från utgångarna på multichippet.

4.5 Spänningsomvandlare

DM7407N

Eftersom multichippet använder sig av en spänning på 5 V och Bluetoothmodulen arbetar vid 3.3 V behöver vi denna krets för att få upp spänningen från 3.3 V till 5 V vid data överföring från Bluetoothmodulen till multichippet.

4.6 Multichipp

Multi-Function Peripheral, MK68901

Istället för att använda sig av flera olika chip som enbart innehåller en specifik funktion föll valet på denna enhet som innehåller många olika funktioner som lämpade sig väl för oss. Funktionerna vi använde oss av var i första hand "Singel channel full duplex USART för att kommunicera med bluetoothmodulen.

Multichippet lämnade i sin tur ett vektoriserat avbrott varje gång vi får indata från bluetoothmodulen som fyller databuffern i multichippet. För att få lite indikation av vad som försegår när vi var uppkopplade mot mobilen använde vi oss av två stycken lysdioder som var kopplade till Output på multichippet. Ena lysdioden indikerar att programmet är igång och flyter på utan problem och den andra lysdioden indikerar ifall vi läser in ett paket som inte är korrekt. För att få rätt baudrate (kommunikationshastighet) mellan multichippet och bluetoothmodulen använde vi oss av en timer på multichippet. Bluetoothmodulen är inställd på att kommunicera i 9.6 KHz och vi ställde in timern därefter. Vi använde även samma timer för att kunna köra bilen i olika hastigheter och för att få olika drivning på hjulen vid svängning. Timern levererar då ett vektoriserat avbrott som vi tog hand om i programmet och delade ner till lämpligt värde för att leverera rätt pulser till drivkretsen. Drivkretsen styr i sin tur motorerna.

4.7 Kristalloscillator

EXO-3

För att leverera rätt klockpuls till processorn använde vi oss av en kristalloscillator. Processorn klarar av en max hastighet på 10 MHz och det valde vi att köra på. Även Multichippet är behov av en klockpuls men den klarar av en klockpuls på max 4 MHz. För att få en fungerande puls i denna enhet använde vi oss av en pinne på processorn som delar klockpulsen 10 ggr, alltså körde vi multichippet på 1 MHz.

4.8 Spänningsregulatorer

LP 3855 Ultra Low Dropout Regulator 3.3V /5V

För att få rätt matningsspänning till komponenterna används två regulatorer. En ”delar” ner spänningen från batteriet som levererar 6 V till övriga komponenter som drivs med 5 V. Bluetoothmodulen drivs i sin tur med 3.3 V och där använde vi oss av en regulator som ”delar” ner spänningen från 5 V.

4.9 Bluetoothmodulen

ARF32

Bluetoothmodulen ARF32 klarar av standarden V1.1 Bluetooth standard och den kan hantera en data hastighet på upp till 723 kbps. Data överföringen mellan bluetoothmodulen och multichippet sker genom UART data porten, vilken använder sig av seriell kommunikation.

Mellan mobilen och bluetoothmodulen använder vi trådlös UART under kommunikations tjänsten SPP (Serial Port Profile).

5. Konstruktion

5.1 Mjukvara

5.1.1 Mobiltelefon

Mjukvaran i mobiltelefonen skrevs i JAVA. Vi skrev en midlet vilket påminner om en vanlig java applet. Vi kunde emulera vår applikation i PCn genom att använda ett program som heter KtoolBar.

Programmet hade följande uppgifter:

- Leta efter Bluetooth enheter (vår bil)
- Leta efter tjänster på enheten (SPP)
- Ansluta till enheten
- Läsa av data från styrspeaken på telefonen och skicka data över Bluetooth

Vi använde oss av JSR82 vilket är en JAVA standard för att hantera Bluetooth i JAVA. Serial Port Protocol (SPP) är en Bluetooth tjänst som är tänkt att emulera en seriekabel trådlöst – man behöver alltså inte bry sig om att informationen skickas i luften och inte i en sladd.

När man väl har anslutit till enheten så kommunicerar man genom helt vanliga java-strömmar.

Vi valde att lägga in bilder i mobilapplikationen så att den blev lite snyggare.



Bilden visar emulatoren som vi körde på PCn. Vi sparade mycket tid på att använda denna då vi slapp att flytta överprogrammet till mobiltelefonen varje gång vi skulle testa.

5.1.2 Bilen

Vi skrev all mjukvara i C. Mjukvaran i bilen hade följande uppgifter:

- Initiera multichippet för avbrott och UART
- Styra el motorerna med hjälp av ett timer interrupt
- Läs in och tolka data från multichippet genom UART interrupt.

Initierings processen går ut på att initiera interuptionerna samt möjliggöra UART kommunikation. Efter initiering går programmet in i en oändlig loop, programmet ligger och väntar på att ett interrupt skall inträffa. När ett timer interrupt inträffar så kan motorernas status ändras beroende på hur snabbt motorerna skall gå.

När bilen svänger vänster så drivs vänster hjulet med 100% och höger hjulet med 40%, omvänt vid höger sväng.

Processorn får ett avbrott för varje byte som Bluetooth modulen skickar.

Kommunikationen till Bluetooth chippet sker via ett protokoll som tillverkaren har skapat. Detta protokoll måste tolkas så att rätt data kan läsas ut. Förutom information som användaren skickar från mobiltelefonen så skickas även annan information som t.ex Connect/Disconnect, Power Save osv.

Start delimiter	Packet type	Operation code	Data length	Checksum	Data	End delimiter
1 byte	1 byte	1 byte	2 bytes	1 byte	<data length> bytes	1 byte

Bilden visar paketstrukturen mellan Bluetooth chippet och CPU.

6. Resultat / Problem

Vi hade flera instabilitets problem med vårt multichip. När vi skrev till register i multichippet så sparades dem inte. När vi läste från registrerna så stämde inte det med det skrivna värdet. Detta problem stod för den mesta tiden under projektets gång. Den pinsamma lösningen med att vi glömt ett pull-up mostånd kändes besvärlig då vi hade lagt ner mycket tid på detta problem, tackar Bertil som upptäckte detta på 5 minuter. Även Bus Master pinnen på processorn hade vi glömt, detta märktes tydligt då vi inte använde IT68 utvecklingskittet. Även detta problemet löste vi i slut fasen. Slutresultatet blev en bra konstruktion förutom att drivningen på hjulen var för dålig för att få en bra svängradie.

7. Sammanfattning

När vi väl började med projektet så var det svårt att få grepp om den totala arbetsmängden man behövde lägga ner. Detta är något som man kommer att kunna uppskatta bättre i framtiden när det vankas ännu större uppgifter som t.ex. examensarbete. Eftersom mycket var nytt för oss båda när det gäller hårdvaru konstruktion, gick det åt mycket tid för att inhämta rätt information från datablad. Detta är något som man har fått värdefull träning i under kursen gång och kommer att ha stor nytta av i framtiden. Under kursen har man även stött på en hel del problem och finna lösningen till dessa har gett värdefull erfarenhet i problemlösning. Att få utveckla en idé och sen verkligen få sätta sina kunskaper i praktiken och även lära sig mycket nytt under resans gång är nåt som har uppskattats stort. Denna kurs rekommenderas varmt till alla dom som vill få goda kunskaper i hur man går till väga för att bygga en egen digitalkonstruktion. Vi vill avsluta med att tacka vår handledare Bertil Larsson för all hjälp vi har fått och ”pushning” i rätt riktning under kursens gång.