

Mikro LAN till mikro budget Digitala Projekt EDI021, VT-03

Niklas Brogden (d99nb)
Magnus Hölén (d99mhl)
Jonas Hörström (d99jho)

Abstract

Smallscale control networks are generally thought to need ethernet or other complex communication standards. These protocols are fast but often need expensive chips and cabling, making their use prohibitive for simple control networks. In home electronics networking, the bitrates provided by ethernet are unnecessarily high for the intended transmission requirements. This report describes an alternative communications project designed to demonstrate how simpler protocols can be used to implement a miniature LAN. The resultant network would enable transmission of small addressable data packets from a central computer to and from a variable number of input/output nodes such as temperature sensors, regulators and displays. The project involves one PC compatible computer, one Motorola HC11 microcontroller as well as a couple of digital input/output nodes connected to the HC11 via a dynamic tree structure.

Innehåll

1 Inledning	4
2 Design	4
2.1 Systemdesign	4
2.2 Kommunikationscentral	4
2.3 Funktionsnod	6
2.4 Dator	6
2.5 Kommunikation	6
3 Genomförande	7
4 Resultat	8
4.1 PC-program	8
4.2 Kommunikationscentral	8
4.3 Nätverksnoder	8
5 Utvärdering	9
5.1 Utvecklingsmiljön för HC-11	9
5.2 Nätverksnoderna	9
6 Bilagor	10
6.1 Kommunikationscentral	10
6.1.1 Version 1.0	10
6.1.2 Version 1.4	10
6.1.3 Program kod	10
6.2 Nätverksnoderna	10
6.2.1 Kommunikationsnod Version 1.1	10
6.2.2 Kommunikationsnod Version 1.7	10
6.2.3 LED-display Version 1.0	11
6.2.4 DIL-koppling Version 1.0	11
6.2.5 DIOD-display Version 1.0	11
6.2.6 Temperaturmätarnod Version 1.1	11
6.2.7 GAL programkod	11

1 Inledning

Vi har utvecklat ett system för att styra små komponenter i det s k “intelligenta hemmet”. Vår idé består i att på ett enkelt, modulärt och kabelsnålt sätt koppla ihop funktioner i ett vanligt hem. Exempel på dessa funktioner är:

- Adaptiv styrning av belysning (ljusstyrkan styrs beroende av bakgrunds-belysning)
- Rökdetektion med uppföljning (ex samtal till brandkår)
- Kontroll av dörrar och kodlås
- Klimatkontroll och reglering

Eftersom modulerna är så pass enkla behövs inte ett avancerat och dyrt nätverk såsom exempelvis ett vanligt ethernet-nätverk. Vi har därför utvecklat ett eget nätverk anpassat för små moduler som inriktar sig på att vara enkelt och billigt. De olika funktionerna ovan, d v s sensorer och givare, byggs som separata moduler och antingen antingen tar emot eller sänder data. Funktionerna kopplas sedan till var sin nätverksmodul som hanterar kommunikation med kommunikationscentralen. Eftersom nätverksdelen och funktionsdelen konstrueras separat torde lansering av nya funktioner bli både snabb och billig. Kostnaden minskas dessutom ytterligare genom att alla komponenterna byggs med enkla och billiga komponenter. Installationskostnaden kan även den minskas då komponenterna kopplas samman med en vanlig korskopplad nätverkskabel innehållande både signal och strömförsörjning.

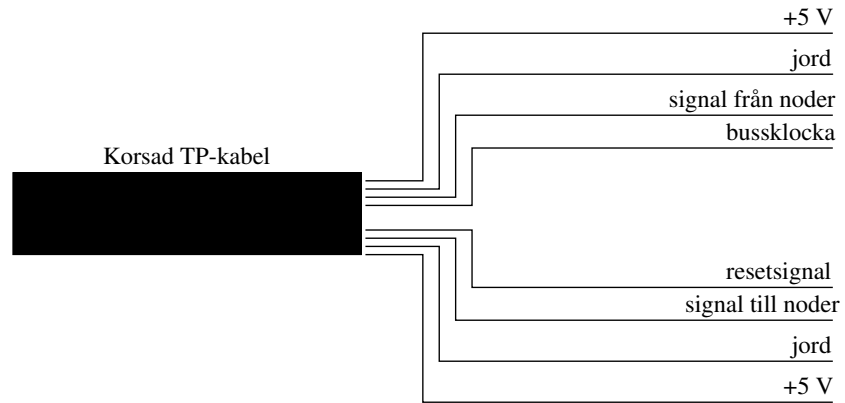
2 Design

2.1 Systemdesign

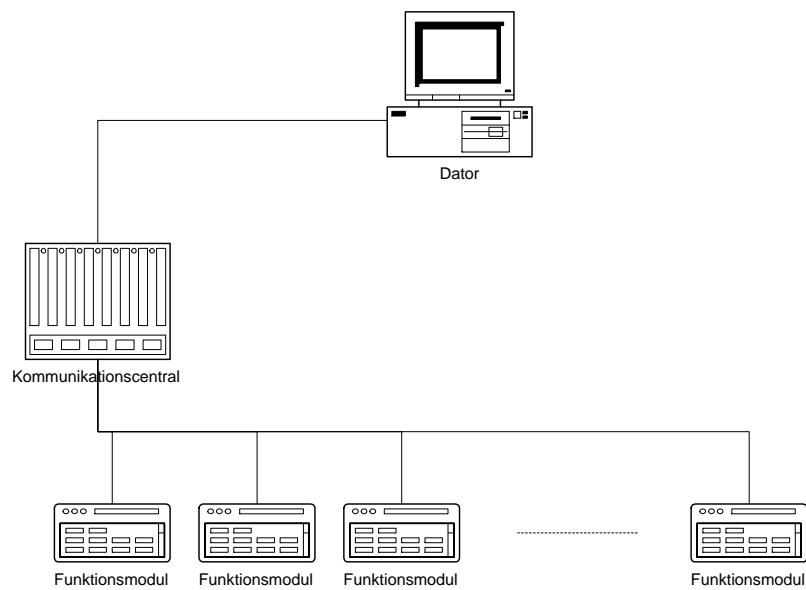
Systemet är uppbyggt av 0 till 127 funktionsmoduler som alla är kopplade i serie mot en kommunikationscentral. Kopplingen mellan funktionsmodul och kommunikationscentral består av korskopplad TP-nätverkskabel som innehåller både signalledningar och strömförsörjning, se figur 1. Kommunikationscentralen är i sin tur kopplad till en dator där alla signaler analyseras och beslut grundade på dessa tas, se figur 2. Dator och kommunikationscentral är sammankopplade med en seriekabel. Kommunikation sker enligt protokollet RS232.

2.2 Kommunikationscentral

Kommunikationscentralen består av en HC-11 som är kopplad delvis till en PC via RS-232C och även till de olika nätverksnoderna via en 'Serial Peripheral Interface' SPI. Separata protokoll används för dessa två kommunikationstyperna. För att initiera en överföring eller reset, skickar PC:n ett till tre bytes bestående av en kommando-byte följd av en adress-byte och eventuellt även en data-byte. HC-11 an



Figur 1: Nätverkskabel med signal- och strömledningar



Figur 2: Datorn kopplad kommunikationscentralen som i sin tur kan ha mellan 0 och 127 funktionsmoduler.

skickar alltid tre konfirmationsbyte tillbaka¹. HC-11an körs med en extern hastighet av 8 MHz som ger en intern clockning på 2 MHz. RS-232C förbindelsen är konfigurerad för 9600 baud och SPI överför data i 62,5 kBps. Överföringshastigheterna kan drastiskt förbättras med enbart mjukvaruförändringar, men det är tänkt att systemet skulle vara så kompatibel som möjligt med 'legacy' system så som gamla datorer mm. Den ursprungliga designen för kommunikationscentral finns bifogad tillsammans med den slutliga implementeringen i slutet av detta dokument som bilaga 6.1.1 respektive 6.1.2. Där syns tydligare hur både RS-232C kopplingen samt SPI är implementerade i hårdvaran.

2.3 Funktionsnod

En funktionsmodul består av en nätverksdel med åtta pinnar in och åtta ut, samt utrustning som styrs av dessa pinnar. Dessa kopplas med fördel ihop på samma krets. Varje nod har en ställbar adress och kan adresseras för både skrivning och läsning. Sändning sker på en separat linje och interfererar därmed inte med data som skickas från kommunikationscentralen. Varje nätverksdel har en så kallad "Open collector koppling" till sändarlinjen och endast den modul som blivit adresserad kommer att skicka ut data på linjen. Detta förhindrar modulerna att störa varandra vid återsändning till kommunikationsnoden. Alla sensorer och givare levererar eller hämtar åtta bitar parallellt till nätverksdelen. Det är möjligt att ordna en interruptlinje när ny data anlänt, dock implementerades detta inte. Nätverksdelen använder en gal² som styrkrets.

2.4 Dator

Vi har utvecklat ett program för kommunikation med nätverket. Även detta är uppbyggt modulärt, dvs med nätverksdelen separerad från funktionen, och kan enkelt anpassas till alla möjliga applikationer såsom exempelvis grafiska eller funktions-specifika system. I projektet använder vi ett textbaserat användargränssnitt där det är möjligt att skicka och ta emot en byte från olika adresser. Eftersom vi utvecklar vår produkt i Windows NT miljö var vi på grund av NT:s säkerhetsrutiner tvungna att använda oss av fördefinierade rutiner i Windows API istället för att skicka data direkt till hårdvaran.

2.5 Kommunikation

All kommunikation initieras av styrsystemet i datorn enligt protokollet i tabel 1. När data ska sändas till en funktionsmodul sker följande:

1. Datorn skickar adress, data och kommandot "put" till kommunikationscentralen

¹Se gärna Bilaga 6.1.3 för kodutskrift.

²En liten programmerbar logik krets, se datablad för gal22V10D.

Dator -> kommunikationscentral			
Kommando	Adress	Data	Kommentar
reset (0x0)	-	-	Nollställer noder och HC11
get (0x1)	adress	-	Hämta data från nod
put (0x2)	adress	data till nod	Skicka data till nod
Kommunikationscentral -> dator			
Svar	Adress	Data	Kommentar
reset (0x0)	-	-	Fel i synkronisering av nätverket gör en reset
get (0x1)	adress	data från nod	Lämna data från nod
put (0x2)	adress	data till nod	Bekräfta kommando

Tabell 1: Kommunikation mellan dator och kommunikationscentral. Kommando- och datafältet är 8 bitar långa och adressfältet är 7 bitar.

2. Kommunikationscentralen skickar vidare adress, data och "put" till alla funktionsmoduler i nätverket och skickar en bekräftelse på att den mottagit kommandot till datorn.
3. Alla funktionsmoduler tar emot adressen och jämför med sin egen. Om modulen finner sin egen adress sätter den databitarna som följer till sin utgång annars ignoreras resten av överföringen.

När data hämtas från en funktionsmodul sker följande likartade händelseförlopp:

1. Datorn skickar adress, data och kommandot "get" till kommunikationscentralen
2. Kommunikationscentralen skickar vidare adress, data och "get" till alla funktionsmoduler i nätverket och skickar en bekräftelse på att den mottagit kommandot.
3. Alla funktionsmoduler tar emot adressen och jämför med sin egen. Om modulen finner sin egen adress skickas den data som för tillfället finns på dess sensor annars ignoreras resten av överföringen.
4. Kommunikationscentralen tar emot ett eventuellt svar från nätverket och skickar tillbaka detta till datorn

3 Genomförande

Vi började med att designa samtliga komponenter i projektet, d v s kommunikationscentralen, två nätverksnoder, två sensorer och två givare. Som sensorer använde vi en temperaturmätare och en dip-switch och som givare två hexdisplayer samt en diodvektor. För att förenkla och spara tid kopplade vi en givare och en sensor till

samma nätverksmodul. När designen var klar började vi med att bygga kommunikationscentralen och noderna. Hårdvarukonstruktionen på noderna har ändrats en hel del under arbetets gång vilket man lätt ser i bilagorna. Bl a upptäckte vi under arbetets gång att vi hade kopplat alla noders utgångar till samma ledning vilket resulterade i att vi inte kunde hämta data från noderna. Detta åtgärdades i ver 1.6 av nodstrukturen, se bilaga 6.2.2, genom att koppla in en sk "line-driver" med en "pull-up". Mjukvarudelen har varit lättare att ändra och inga större problem har uppstått. Däremot har mycket arbete lagts på att få logikkretsen i nätverksnoden att fungera.

4 Resultat

Resultaten varierade i de olika delarna av projektet. Vi som grupp är dock mycket nöjda med den höga grad av succe som de största delarna av projektet har lyckats med. Vi tycker vidare att designprincipen om små data packet överförda längs strömledande kablage till adresserbara noder har bevisats vara både möjlig och genomförbar med relativt enkla och billiga medel.

4.1 PC-program

Programmet som skapades för att kontrollera nätverkets funktionalitet skrevs i C++ för att användas i DOS-fönster läge. Denna textbaserade gränssnitt verkar fungera i alla eventualiteter och vi är nöjda med att kommunikationen mellan PCn och kommunikationscentralen sker på tillfredställande sätt. RS-232C kommunikationen har därför implicit bevisat sin användbarhet i denna situation.

4.2 Kommunikationscentral

Även kommunikationscentralen fungerar helt enligt de ursprungliga specifikationerna. Den tar emot instruktioner och data som sedan vidarebefordras ut på nätverket via SPI protokollet. Att detta verkligen var fallet kan bevisas med hjälp av en HP LogicDart signal analysator som användes för att kontrollera synkronisering av klockpulser och dataströmmen på nätverket. Tyvärr så blev dessa delar klara och färdigtestade relativt sent i utvecklingsprocessen som är en bidragande faktor till de problem som uppstod i nätverksnodernas slutliga implementeringsfas.

4.3 Nätverksnoder

Nätverksnoderna utvecklades samtidigt som de övriga systemkomponenterna. Dock var dessa till skillnad från kommunikationscentralen och PC programmet svåra att testa oberoende av andra komponenter. När de kunde testas kunde det konstateras att några alvarliga brister hade introducerats i designen. Dels hade ett missförstånd lett till att klockningen av logikkretsarna inte fungerade, detta kunde enkelt fixas.

Dels lädde en felaktigt vald D-elements krest till onödigt testande, detta åtgärdades med lätthet när det upptäcktes. Det förekom ett mindre antal felkopplingar och timing problem, speciellt i tillståndsmaskinen i gal:en som styrde övriga kretsar. Dessa kunde jagas ner med hjälp av en LogicDart som var ett ovärderligt hjälpmedel. LogicDart är en logikanalysator som kan studera 3 kanaler samtidigt. Denna används under laborationerna i digitalteknik och är som sagt ovärderlig när man ska studera timing och pulståg. En av nätverksnoderna skulle utrustas med en temperatur sensor, dock fick denna ide'n överges då den magiska vita röken läkte ut från själva sensorn. Så temperatursensorn som inputkälla fick strykas från den ursprungliga designen i projektets slutskede.

5 Utvärdering

Vårt arbete har i mångt och mycket varit inriktat på att utveckla saker i moduler. Detta anser vi oss ha lyckats mycket väl med. Arbetet har främst varit inriktat på hårdvarudelen eftersom den är mest intressant för kursen. Mycket av vår tid har gått åt till att leta felaktiga kopplingar i hårdvaran. Detta har inte bara givit oss gråa hår utan även mycket ny kunskap i både komponenterna och felsökningsmetodik. Länge leve logicDart.

5.1 Utvecklingsmiljön för HC-11

Utvecklingsmiljön för HC11 är utmanande men vi har ej behövt utnyttja särskilt många av dess analysverktyg. Efter några mindre problem med felaktiga interrupt header filer, och en söndrig display element, fungerade HC11 emulatoren utan anmärkning. Utvecklingen av programvaran skedde i flera etapper. Först testades emulatorns hårdvara genom implementering av en enkel räknare. Indikatoren användes sedan som status display då RS-232C protokollet infördes i systemet. När en pålitlig kommunikation hade etablerats mellan PC-datorn och HC11an, kunde arbetet med SPI protokollet koordineras med nätverksmodulerna. Det kunde konstateras att HC11 skickade protokoll-riktiga datapaket ut på bussen med hjälp av bla en HP LogicDart logikanalysator. Då återstod enbart arbetet med att felsöka nätverksmodulerna som hade några utvecklingsrelaterade problem.

5.2 Nätverksnoderna

Tyvärr har det varit svårt att få logikkretsen i nätverksnoden att fungera. Problemet låg mest i att det inte fanns någon simuleringsmiljö eller möjlighet att få ett grafiskt kopplingsschema över kompileringsresultatet. Logiken till kretsen skrevs i programspråket Able och en utskrift finns som bilaga i 6.2.7.

6 Bilagor

Bilagor för rapporten finnes på de följande sidorna.

6.1 Kommunikationscentral

Här inkluderas två kretskopplingsdiagram. Den första daterar från projektets början och den andra från projektets slut. Bifogat är även huvuddelen av programkoden som körs i HC-11an för att kommunicera med PCn och nätverksnoderna

6.1.1 Version 1.0

I denna version är största delen av designen korrekt, med RS-232 kretsen MAX232, mikrokontrollern HC-11 och reset kretsen MC34064 på plats. Dock fanns vissa fel som korrigerades i följande versioner av kommunikationscentralen.

6.1.2 Version 1.4

I denna slutgiltiga versionen ingick en display krets (TIL311) och diverse förbättringar i designen. Kondensatorerna vid MAX232 kretsen har förstärkts till 1,1 μ F, en 'på' LED är inkluderad och diagrammet speglar mycket väl den slutliga hårdvaruversionens kopplingssystem.

6.1.3 Program kod

Denna utskrift av Borg.c filen visar med tydlig 'inline' dokumentation hur protokollet har genomförts. Självaste kommunikationsrutinerna har inte skrivits ut pga storlek och att dessa i stor del var redan befintliga då projektet programmerades.

6.2 Nätverksnoderna

Här inkluderas två kretskopplingsdiagram. Den första daterar från projektets början och den andra från projektets slut. Vidare finns inkluderat utskrifter av kopplingscheman för de fyra tilläggs modulerna som används för självaste input/ouput funktionaliteten.

6.2.1 Kommunikationsnod Version 1.1

Detta var det första kompletta schemat för en kommunikations nod. Dvs, den delen av varje modul som hanterade kommunikationen med kommunikationscentralen. Som synes finns många praktiska brister som tillades efterhand i följande versioner.

6.2.2 Kommunikationsnod Version 1.7

Den slutgiltiga och fungerande versionen av nätverksnoden. Här har D-elementen bytts ut till en korrekt modell som inte har en s k "three state".

6.2.3 LED-display Version 1.0

En enkel output display enhet som konstruerades med hjälp av två st TIL311 kretsar för koppling på samma kretskort som kommunikationsnoderna.

6.2.4 DIL-koppling Version 1.0

En enkel input enhet som konstruerades med hjälp av en 8-bitars DIL kopplare och ett resistorpack.

6.2.5 DIOD-display Version 1.0

En output enhet som i stället använde en diodrad för nivå-output.

6.2.6 Temperaturmätarnod Version 1.1

En input enhet för att registrera en temperaturmätning som digitalt värde för överföring via kommunikationsnoderna. Här användes en ADC0804 omvandlare tillsammans med en LM35 temperaturgivare.

6.2.7 GAL programkod

Här finns programmet som skrevs i ABLE för att driva GAL kretsarna på kommunikationsnoderna.