

Initial Z  
Digitala Projekt 2005

Håkan Karlsson – e03hk  
Zoltan Michis – e02zm

Handledare: Bertil Lindvall

May 17, 2005

## **Abstract**

One of the greatest problems in drag racing is the loss of traction in high horsepower racecars. That's why we decided to design a traction control for the Nissan Z32 Fairlady. We have implemented three major areas of function, Launch-Control, Traction-Control and Full-Throttle-Shift. These functions are combined to seamlessly improve the safety and handling of high-powered racecars. All of the functions communicate with the car-ECU via the built-in Consult-interface. Data from the cars stock ABS-sensors is processed by the microcontroller unit (Atmel AVR-Mega16) and used to calculate wheel slip. It then sends a command to the car to limit the number of cylinders firing, thus limiting the power of the VG30DETT engine.

# Contents

<b>1</b>	<b>Inledning</b>	<b>1</b>
<b>2</b>	<b>Teori</b>	<b>2</b>
2.1	Launch control . . . . .	2
2.2	Traction control . . . . .	2
2.3	Full throttle shift . . . . .	2
2.4	ABS-sensorer . . . . .	2
2.5	Diagnostikuttaget hos Nissan 300ZX (Consult) . . . . .	3
2.6	PID-regulatorn . . . . .	3
<b>3</b>	<b>Genomförande</b>	<b>4</b>
3.1	Kravspecifikation . . . . .	4
3.2	Design av krets . . . . .	4
3.2.1	ABS-kretsen . . . . .	4
3.2.2	Consult-Interfacet . . . . .	5
3.2.3	Display . . . . .	5
3.2.4	Strömförsörjning . . . . .	5
3.3	Komponentlista . . . . .	5
3.3.1	Processor . . . . .	5
3.3.2	Display . . . . .	5
3.3.3	Komparatorer . . . . .	5
3.3.4	Spänningskällor/regulatorer . . . . .	5
3.3.5	Räknare/Divider . . . . .	6
3.3.6	Transistorer . . . . .	6
3.3.7	Kristaller . . . . .	6
3.4	Design av mjukvara . . . . .	7
3.4.1	Insamling och bearbetning av data . . . . .	7
3.4.2	Beräkningar och reglering . . . . .	7
3.4.3	Exekvering . . . . .	7
<b>4</b>	<b>Resultat och Diskussion</b>	<b>8</b>
<b>A</b>	<b>Kretsschema</b>	<b>10</b>
<b>B</b>	<b>Källkod</b>	<b>11</b>

# 1 Inledning

Trimmade bilar idag har allt högre effekt vilket har som biverkning att oerfarna förare har svårigheter att tygla denna effekt. Denna kombination innebär en större risk både för allmänheten och förarna. Vi har valt att åtgärda en del av problemet genom att utveckla, designa samt producera ett väghållningskontrollsystem till en Nissan Z32 fairlady (300 ZX TT). Systemet består huvudsakligen av tre delar:

- Launch control
- Full throttle shift
- Traction control

Dessa samverkar för att få större kontroll över bilens väghållningsegenskaper. I teoridelen beskriver vi hur vi går tillväga för att bestämma hjulhastigheterna utifrån bilens original ABS-system. Vi går även igenom hur vi reglerar bilens effekt m.h.a. en PID-regulator för att återfå greppet via bilens egen diagnostikport (Consult). I genomförandedelen beskriver vi hur vi går tillväga för att uppfylla kravspecifikationerna, t.ex. hur vi med hjälp av komparatorer konverterar ABS-signalerna till fyrkantspulser så att processorn kan behandla datan. I resultatdelen presenterar vi den färdiga kretsen och de kompromisser vi tvingas göra. Vi har begränsat oss till att endast framställa en prototyp och ej föra projektet till en färdig produkt.

## 2 Teori

### 2.1 Launch control

En launchcontrols funktion är att vid stillastående starter maximera initial acceleration t.ex. vid dragracing. Detta görs genom att hålla motorn vid optimalt startvarvtal och reglera effekten så att optimalt väggrepp erhålls. I praktiken sätts ett varvstopp in när kopplingen trycks ner så att motorn håller sig på rätt varvtal. När kopplingen sedan släpps, kopplas varvstoppet ur och traction controllern tar över.

### 2.2 Traction control

Traction controllerns funktion är att begränsa hjulspinn detta sker genom att bränsletillförseln till enskilda cylindrar stängs av efter behov. Därmed minskar motorns uteffekt och grepp återfås.

### 2.3 Full throttle shift

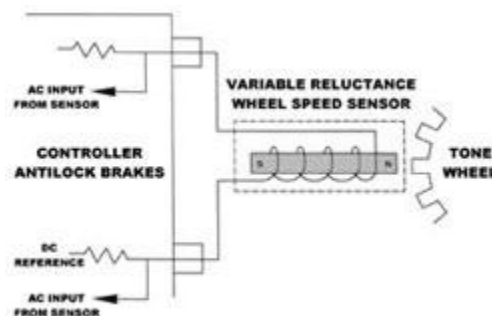
Full throttle shiftern möjliggör växlingar utan att släppa upp gaspedalen. Detta förenklar snabba växlingar genom att föraren ej själv behöver hålla rätt varvtal utan full throttle shift funktionen sköter det åt honom/henne. I praktiken är det samma funktion som launchcontrollerns startfas med den skillnaden att bilen rör sig.

### 2.4 ABS-sensorer

De flesta bilar är idag utrustade med ABS-sensorer som mäter hastigheten på individuella hjul. Det finns huvudsakligen två olika typer:

- Magneto-resistive
- Variable-reluctance

Variable-reluctance-sensorerna användes i tidiga ABS-system. De består av en spole och stavmagnet. Sensorn är fäst intill ett tonhjul, se figur 1.

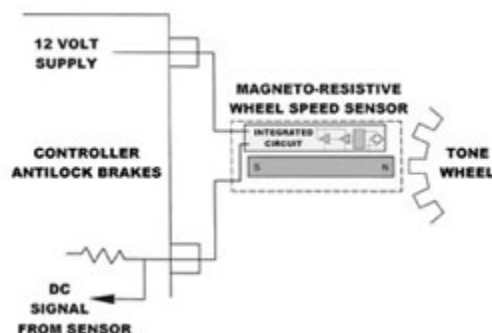


Figur. 1: Variable Reluctance

När tonhjulet snurrar kommer magnetfältet att fluktuera och en AC-spänning genereras. Problemet med dessa sensorer är att hjulet måste rotera med en viss

minsta hastighet för att en användbar signal ska erhållas.

Magneto-resistive-sensorerna har bättre precision och fungerar även vid låga hastigheter till skillnad från sina föregångare. Därför sitter det magneto-resistive-givare i de flesta moderna bilar. I likhet med variable-reluctance-sensorerna har de tonhjul och permanentmagnet.



Figur. 2: Magneto Resistive

Magneto-resistive-sensorerna mäter istället resistansen i en krets och genererar en fyrkantspuls motsvarande tonhjulets toppar och dalar med hjälp av två strömkällor.

## 2.5 Diagnostikuttaget hos Nissan 300ZX (Consult)

Alla Nissan 300ZX (1990-1996) är utrustade med ett diagnostikuttag för felsökning och service av bilen. Consult-protokollet stöder bl.a kommandon för avstängning av cylindrar och liveströmmning av data från de flesta sensorer i bilen, dock ej ABS-sensorerna. Consult kommunicerar seriellt via en 3-trådsbuss, Rx, Tx och asynkron Clk. Informationen till Consult, Rx har spänningsnivåerna 12V för nolla och 0V för etta. Tx är öppen kollektor.

Kommunikationen sker asynkront i 9600 baud, startbit låg, 8 databitar och en stoppbit hög. Consult kräver även en klocka i 16x datahastigheten vilket blir 153.6 kHz  $\pm$ 5kHz, klockan behöver inte synkroniseras med datalinjerna. Consult kräver ett handskakningsmeddelande bestående av 0xFF 0xFF 0xEF för att kunna ta emot kommandon. För komplett kommandolista se Bilaga 1.

## 2.6 PID-regulatorn

För att erhålla rätt effekt vid en given tidpunkt används en PID-regulator. PID-regulatorn består av tre delar. En P-del som är primärt en linjär del för på/av reglering. En integrationsdel (I-del) som eliminerar reglerfelet som orsakas av P- och D-delen. D-delen är en derivatadel som håller reda på lutningen hos vår reglerfunktion. Vid empiriska försök har man påvisat att optimal acceleration uppnås vid ca. 10 procent hjulspinn, d.v.s. 10 procent högre hastighet på de drivande hjulen.

## 3 Genomförande

### 3.1 Kravspecifikation

Vid projektets början skrev vi ned en kravspecifikation som innehöll det vi ville att applikationen skulle klara av efter projekttidens slut. Den skulle klara av att:

- Behandla ABS-signal från Nissan 300ZX TT så att den kan läsas av en A/D-konverter.
- Räkna ut hjulhastighet och skillnad i hjulhastighet.
- Kommunicera med Nissans konsultport.
- Reglera tändningen utifrån ABS-signal så att grepp på drivhjulen återfås.
- Launch-control.

I mån av tid skulle den:

- Låta användaren bestämma tillåten spinn på drivhjulen.
- Visa data på LCD-display:
  - Hastighet.
  - Acceleration.
  - Hjulspinn.
  - Motoreffekt.
- Timer-enhet för mätning av kvartsmile och varvtider samt 0-100 hastigheter o.s.v.
- Full throttle shift.

### 3.2 Design av krets

För vår applikation valde vi Atmels Mega16 mikroprocessor. Den innehåller många av de funktioner som vi ville ha, bl.a. en inbyggd UART för att hantera seriell kommunikation. Den innehåller även en timer som vi använder för att mäta tiden på pulserna från ABS-kretsen. Den är dessutom lagom snabb, kräver inga större yttre komponenter och har tillräckligt med minne för våra ändamål.

#### 3.2.1 ABS-kretsen

Variable-reluctance sensorerna som mäter våra hjulhastigheter ger som beskrivet i teoridelen en sinusformad utsignal som är linjärt beroende av hjulhastigheten. Amplituden för signalen varierar från några mV till flera volt beroende på hjulens hastighet. Därför används en komparator-krets som har en förinställd hysteres på  $\pm 0.05V$ . Komparatorn genererar en fyrkantspuls mycket lik signalen från de moderna magneto-resistive signalerna. Hysteresbandet gör att små störningar filtreras bort men även att små signalstyrkor mindre än 50mV försvinner. Dessa låga spänningar uppkommer vid väldigt låga hastigheter och kan därför ignoreras. För att få referensspänningen till komparatorn att ligga runt 0V krävdes att vi försåg den med negativ matningsspänning. Denna spänning tas från ett RS-233 chip som har en intern spänningsomvandlare  $\pm 10V$ .

### 3.2.2 Consult-Interfacet

Bilens Consultport jobbar med 0-12V signalnivå och vår Mega16 mikroprocessor jobbar med 0-5V. Därför behöver vi ett interface emellan som konverterar 0-12V till 0-5V och vice versa. Konverteringen mellan 0-5V och 0-12V genomförs m.h.a. en transistor som öppnar när processorn skickar en hög signal till basen och därmed jordar signalen till Consultporten som annars är 12V. (Idle high) 0-12V till 0-5V erhålls genom en enkel spänningsdelning. Consulen behöver även en frigående klocka på 153.6 kHz. (+-5 kHz) Denna åstadkommer vi genom att koppla in en oscillator krets bestående av en kristall och två kondensatorer, till en räknare som minskar frekvensen från 4.91520 MHz till 153.6 kHz samt omvandlar det till en fyrkantspuls. Klockan måste också ha amplituden 0-12V och därför använder vi en koppling liknande den för utsignalen från processorn.

### 3.2.3 Display

Displayen valdes för dess enkelhet och många tydliga tecken för att enkelt åskådliggöra mätdata. Displayen har tre styrsignaler och en 8-bitars databuss, dessutom har vi kopplat en kontraststyrkrets bestående av en potentiometer till den.

### 3.2.4 Strömförsörjning

Matningsspänningen tas från Consult-porten (12V) och regleras ned till 5V med en spänningsregulator. I kretsen har vi även en lysdiod som indikerar när kretsen är på.

## 3.3 Komponentlista

### 3.3.1 Processor

- Atmels AVR-Mega16 som arbetar i 16MHz med extern klockförsörjning. Den är en 8-bitars microcontroller med 16kB programmerbart flash-minne, 512B EEPROM och 1kB SRAM. AVR:en programmeras via ett JTAG-interface från en PC.

### 3.3.2 Display

- Vikay VK2140 40-teckens alfanumerisk dot-matrix modul med inbyggd LCD-driver.

### 3.3.3 Komparatorer

- 2 stycken MAX923 ytmonterade komparatorer med programmerbar hysteres och extern referens-spänning.

### 3.3.4 Spänningskällor/regulatorer

- Ett MAX RS-233CPP chip som negativ spänningskälla.
- En L7805CV för att reglera ned Consultens 12V till 5V.



### **3.3.5 Räknares/Divider**

- En Phillips 74HC4060N för att dela 4.91520 MHz sinusvåg till 153.6 kHz fyrkantsvåg.

### **3.3.6 Transistorer**

- Två stycken BC109C.

### **3.3.7 Kristaller**

- En 16.00000 MHz Kristall
- En 4.91520 MHz Kristall

## 3.4 Design av mjukvara

Huvuddelen av applikationen består av mjukvara som kan delas upp i 3 delar:

- Insamling och bearbetning av data.
- Beräkningar och reglering.
- Exekvering.

### 3.4.1 Insamling och bearbetning av data

Från komparatorerna erhålls en fyrkantsvåg vars frekvens ger hjulhastigheten. För att få fram frekvensen använder vi en samplande-algoritm som fungerar genom att ett sampel tas från den pin som ska avläsas. Samplet används sedan för att avgöra var i perioden vi befinner oss. När den slår om, startas en timer som stoppas när en hel period har gått och därigenom fås periodtiden och frekvensen som invers av periodtiden.

Motorns varvtal avläses via Consult-interfacet från Engine-ECU-register 0x00 och 0x01 och multipliceras med 12.5 för att erhålla faktiskt varvtal. Kopplingspedalswitchen avläses via Consult-interfacet från S-HICAS-ECU-register 0x04 bit 3.

### 3.4.2 Beräkningar och reglering

Vi har valt att använda en enklare P-regulator under utvecklingsfasen, dock ska en PID-regulator implementeras så snart vi är klara med alla småjusteringar. När bakhjulet (-en) roterar med en hastighet som överstiger framhjulen med mer än 10 procent stängs en cylinder av genom att skicka kommandot 0x88 till Consult-porten. För varje ökning med 10 procent stängs ytterligare en cylinder av. Dessa 10 procent är en källa för småjusteringar.

När kopplingspedalen är intryckt införs ett sekundärt varvstopp genom att i realtid övervaka varvtalet och stänga av cylindrar för att förhindra övervarvning. Varvstoppsregleringen sköts med en P-regulator. För varje 100 varv över önskat varvstopp stängs en cylinder av.

### 3.4.3 Exekvering

I exekveringsfasen av programmet utförs all dataöverföring mellan Consult och Mega16 dessutom uppdateras displayen med de mätdata och beräkningsdata som erhållits. Dataöverföringen sker via Mega16:ens inbyggda USART-protokoll.

## 4 Resultat och Diskussion

I skrivande stund är projektet ej helt klart detta beror på det icke oansenliga antal fel som förföljt oss genom hela projektet. Vår applikation kan hittills:

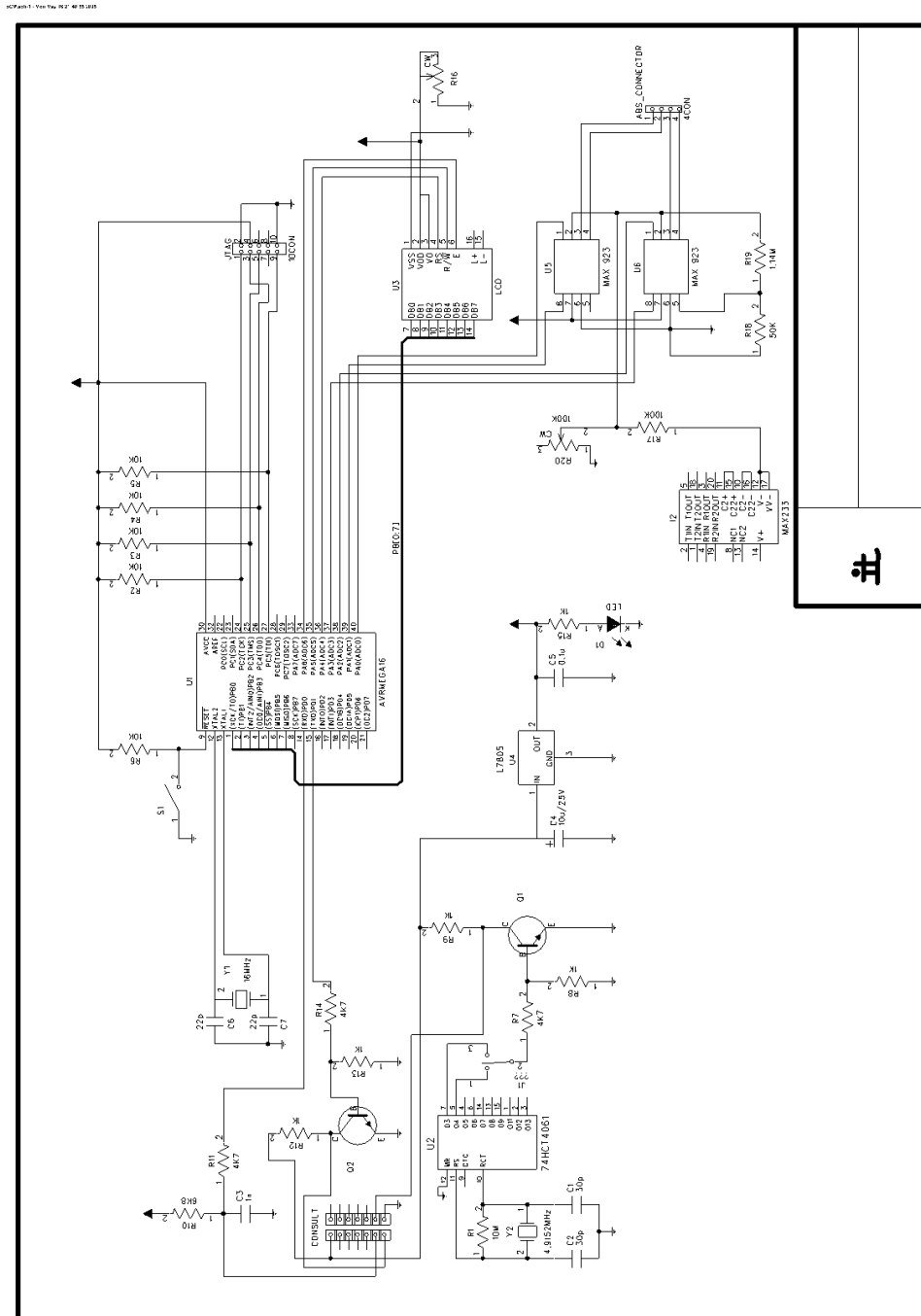
- Beräkna periodtid samt frekvens för varje enskilt hjul i realtid.
- Mata ut de mätdata och beräkningar som erhållits.
- Initiera kommunikation med Consult.
- Kommunicera med PC via RS-233-interface.

Svårigheten med projektet har hela tiden varit att vi blivit tvungna att lita på andrahandsinformation angående Consult-portens signaler. Bl.a. felaktigheten att Consulten kommunicerade med etta 12V och nolla 0V krävde flera veckors felsökning och är den största enskilda orsaken att vi inte hunnit bli klara inom utsatt tid. Andra svårigheter har mest kretsat runt labbutrustningen som i flera fall visat sig vara trasig och gett ej tillförlitliga mätvärden. Även programmeringen av AVR-chippet har bestått med svårigheter då vissa datorer fick lock-fusen att sättas på chippet. Trots alla motgångar anser vi att projektet varit lyckat då vi nu bara är några veckor ifrån en färdig prototyp. Vi kommer att fortsätta projektet under sommaren och hösten. Då kommer vi även att försöka med en separat reglering för injektorpulserna och fundera på en alternativ metod för att läsa av ABS-signalerna.

## References

- [1] PLMS Development. *<http://www.plmsdevelopments.com>*, 2005-05-17. Copyright PLMS Development, 2005.
- [2] David W. Gilbert. *<http://www.asashop.org/autoinc/aug2001/mech.cfm>*, 2005-05-17. Copyright David W. Gilbert, 2001.
- [3] ELFA. *Diverse komponent-datablad* – *<http://www.elfa.se>*, 2005-05-17.
- [4] Department of Information Technology. *Kursens hemsida* – *<http://www.it.lth.se/digp/>*, 2005-05-17.

# A Kretsschema



Figur. 3: Kretsschemat

## B Källkod