

AT90CAN32, hoofdstuk 8

E. Gernaat (ISBN 978-90-79302-06-2)

1 CAN-controller

1.1 Introductie

De AT90CAN32 is intern uitgevoerd met een CAN-controller. Dit betekent dat we vanuit de AT90CAN een CAN-boodschap kunnen verzenden en ontvangen. CAN (Controller Area Network) is een real-time, seriëel netwerk met een zeer grote betrouwbaarheid dat werkt volgens het broadcast protocol, d.w.z. een boodschap kan worden verzonden en alle CAN-controllers die op de bus zijn aangesloten (met uitzondering van de zender) kunnen deze boodschap ontvangen. Elk CAN-station kan zowel zenden als ontvangen. De AT90CAN CAN-controller voldoet aan de specificaties van CAN 2.0 A en 2.0 B, dat wil zeggen dat er zowel met 11 bits of 29 bits identifiers kan worden gewerkt. De maximum baudrate bij een 8 MHz processor clock bedraagt 1 Mbit/s. De AT90CAN CAN-controller kan voor maximaal 15 unieke boodschappen worden geprogrammeerd. Deze worden Message Objects (MOB) genoemd.

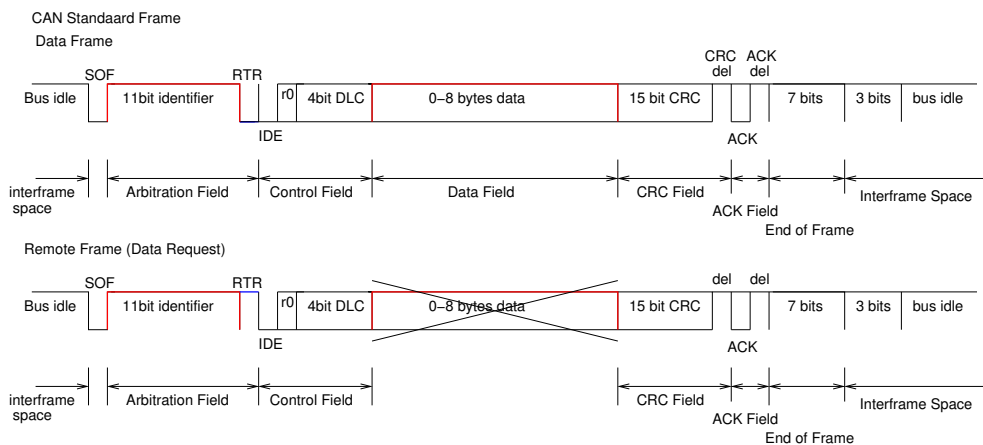
Zoals reeds gezegd is CAN gebaseerd op het broadcast communicatie principe. De communicatie berust op een boodschap (message) georiënteerd transmissie protocol. De boodschap wordt gekenmerkt door een boodschap-identificer. De boodschap-identificer moet uniek zijn binnen het gehele netwerk en geeft niet alleen aan wat de betekenis is van de boodschap maar bepaalt ook de prioriteit van de boodschap. De prioriteit (voorrang) wordt vastgelegd tijdens het programmeren en kan dan niet meer dynamisch worden gewijzigd. De identificer met het laagste binaire nummer heeft de hoogste prioriteit. Wanneer een bus-conflict dreigt te ontstaan in geval twee modules tegelijk willen gaan zenden dan trekt de module met het hogere identificer-nummer zich terug (de nul domineert) en gaat dan over op ontvangen. Pas wanneer de bus weer toegankelijk is zal de module het opnieuw gaan proberen.

Het CAN-protocol ondersteunt twee boodschap formaten. Het verschil zit in de lengte van de indentificer. De CAN-standaard, bekend onder CAN 2.0 A ondersteunt een 11 bits indentificer en CAN extended bekend onder CAN 2.0 B ondersteunt een 29 bits identificer. CAN 2.0 A kan dus $2^{11} = 2048$ boodschappen aan. Zou elke controller 15 unieke boodschappen kunnen versturen dan kunnen er

volgens deze redenering 136 CAN-controllers op de bus worden aangesloten. Het CAN standaard data frame (het gehele informatiepakket) bestaat uit:

- Start Of Frame (SOF);
- 11 bit identifier (arbitrage veld);
- Remote Transmission Request (RTR);
- Identifier Extension (IDE);
- Data Length Code (DLC);
- Data field (max. 8 bytes);
- Cyclic Redundant Check (CRC);
- ACKnowledge field (ACK);
- ACK delimiter;
- End of frame.

De (voor het begrip) belangrijkste delen uit het pakket zijn ongetwijfeld de identifier en het dataveld. Behalve informatie zenden kan de controller ook om informatie vragen. We spreken dan over het ontvangst of remote frame. Het data- en het remote frame kunnen wel met dezelfde identifier werken. In geval van gelijktijdigheid gaat het data-frame voor. Het data en remote-frame zijn vrijwel identiek alleen het dataveld ontbreekt bij het remote frame. Fig. 1 geeft het data-frame grafisch weer. Omdat we de zaken niet nodeloos gecompliceerd willen maken laten we het extended frame achterwege. De twee



Figuur 1: Het CAN-standaard frame. Boven het 'Data Frame', d.w.z. data wordt verzonden en onder het 'data-verzoek frame' of Remote Frame. Hier wordt in het zendbericht om data gevraagd.

formaten (CAN 2.0 A en CAN 2.0 B) kunnen, als de controllers de mogelijkheid hebben, wel samen op de bus. Als dat het geval is dan spreken we van 2,0 B Active. De boodschappen in CAN-standaard hebben dan wel altijd voorrang. De CAN controller in de AT90CAN32 kan op 2.0 B active worden ingesteld.

CANbit-timing moet worden toegepast om de bits te synchroniseren gedurende het zenden van het frame. Synchronisatie vindt plaats bij het begin van elke

CAN-boodschap en op elke dalende flank. De baudrate van de CAN-controller kan worden ingesteld. De meest bekende CAN-baudrates zijn: 1000, 500, 250, 200, 125 en 100 kbits/s.

Het CAN-protocol signaleert fouten (errors) op het moment dat ze optreden. Er worden fouten onderscheiden op bit-niveau en op boodschap-niveau. Als er een fout optreedt dan wordt het zenden gestopt door een 'foutvlag' uit te zenden. De zendende controller probeert het dan opnieuw.

2 Het programmeren van de AT90CAN32 CAN Controller

2.0.1 Inleiding

Gedurende de initialisatie bepaalt de programmeur voor elke controller welke boodschappen verzonden moeten worden en welke moeten worden ontvangen. Alleen wanneer de CAN-controller een boodschap ontvangt die correspondeert met de opgeslagen identifier wordt de boodschap opgeslagen en wordt de AT90CAN32 geïnformeerd door een interrupt. Voor elke boodschap die ontvangen of verzonden moet worden beschikt de CAN-controller over Message Objects buffer registers (MOB's). MOB's zijn data-buffers. Men onderscheidt dan de registerbuffer van MOB0, MOB1 etc. Alle MOB's vormen tezamen 120 Bytes geheugen ruimte. De programmeur dient eerst de Mob's te vullen met de gegevens van het CAN-frame.

2.0.2 De baudrate

Dit is een tamelijk ingewikkelde materie. Het meest eenvoudige is om de baudrate voorbeelden uit de data-gegevens van de AVR-handleiding te halen. We moeten dan rekening houden met de kloksnelheid van de controller. De baudrate wordt weggeschreven in drie registers (CANBT1 t/m CANBT3). Bijvoorbeeld: wanneer we de hexadecimale getallen 16, 04 resp. 13 wegschrijven naar het CANBT1 t/m CANBT3 register dat wordt intern de baudrate op 125 kbits/s gezet.

2.0.3 Message Objects

De MOB beschrijft het CAN-frame. De gehele CAN-message wordt beschreven als een object. De MOB's vormen de mailbox waaruit de messages worden ontvangen en verzonden. Vanuit het CANPAGE register selecteren we met bit 4 t/m 7 de gewenste MOB's. De MOB's zijn genummerd van 0 t/m 14. Schrijven we bijv. het hexadecimale getal 00 naar het CANPAGE-register dat selecteren we MOB0. De MOB's hebben werkmoden. Deze zijn:

- Disable mode (00);
- Enable Transmit mode (01);

- Enable Reception mode (10);
- Enable frame buffer reception mode (11).

De gewenste werkmode kan worden gekozen met behulp van bit 6 en 7 het CANCDMOB register. In hetzelfde register kan ook worden gekozen voor het aantal bytes van de data en de grootte van de indentifier (11 of 29 bits). Schrijven we bijv. 0b01000001 naar het CANCDMOB register dan betekent dat:

- transmissie geactiveerd;
- CAN-standaard 2.0 A geselecteerd;
- Message 1 byte groot (0001).

Voordat de CAN-bus geactiveerd wordt (Enable) moet elke MOB worden geconfigureerd. Niet gebruikte MOB's moeten in de disable mode (00) worden gezet. Voorbeeld:

```
LDI R16,0b00000000
STS CANPAGE,R16 ; selecteer MOB0
STA CANCDMOB,R16 ; disable MOB0
```

De CAN-controller in de AT90CAN onderscheidt de general CAN-Registers en de CAN Message Object buffer (MOB) registers. De 'general registers' hebben na het opstarten een default (standaard) waarde. De MOB registers niet. Zodra de CAN-bus geactiveerd wordt door bit 1 in het CANGCON(trol)-general register op logisch 1 te zetten worden de configuratiebits van alle 15 MOB's onmiddellijk actief. Het zal duidelijk zijn dat dus eerst de MOB's geconfigureerd moeten worden voordat de CAN-bus geactiveerd wordt.

3 Het zenden van een dataframe

De volgende handelingen moeten worden verricht wanneer we gaan zenden:

- Zet alle MOB's in disable mode;
- Stel de baudrate in m.b.v. de registers CANBT1 t/m CANBT3;
- Selecteer de MOB's die je wilt gebruiken en stel dan in:
- Gewenste mode, data lengte en 11 of 29 bits indentifier in het CANCDMOB register;
- Zet de indentifier in de vier CANIDT registers;
- Zet de data byte(s) in het CANMSG register;

Wanneer we nu de CAN-bus op 'enable' zetten (02h naar het CANGCON register) dan wordt het CAN-frame continu verzonden. In het nu volgende programma CAN2.asm wordt de stand van de schakelaars op de CAN-bus gezet. De betrekkelijk uitgebreide initialisatie vindt plaats in timlotoCAN.inc file. Verder betreft het de instelling van de baudrate, de selectie van de MOB en het instellen van de indentifier. Hier volgt eerst het programma. Lees eerst de toelichting voordat de opgaven worden uitgevoerd.

*/*Programma naam: CAN2.ASM
Programma voor de AT90CAN32 Elektor-Timloto print.
Het programma leest de schakelaars in en zend de stand van de schakelaars
via de CAN-bus uit met de identifier 000 0000 0000.
Programma en data in het Flash geheugen (8 MHz controller)
Baudrate op 125 kb/s CANBT1 0x16; CANBT2 0x04 en CANBT3 0x13
'Message' te controleren met oscilloscoop en/of USB-CAN adapter van Elektor
Aansluiting op pin 1 en 3 van K7. */*

.DEVICE AT90CAN32

*.INCLUDE "can32def.inc" ; definitie poorten in aparte file
.INCLUDE "timlotoCAN.inc" ; eigen initialisaties*

;Nu volgt de eigenlijke instelling

*/*INITIALISATIE BAUDRATE 125 kb/s */*

*LDI R16,0x16 ;baudrate instelling
STS CANBT1,R16 ;prescaler afwijkend van tabel*

*LDI R16,0x04
STS CANBT2,R16*

*LDI R16,0x13
STS CANBT3,R16*

;MOB0 selecteren

*LDI R16,0b00000000 ;selecteer MOB nummer 0000
STS CANPAGE,R16*

*LDI R17,0b01000001 ;enable transmissie 0100
STS CANCDMOB,R17 ;data lengte 1 byte 0001
;11 bits identifier*

;stel de identifier (000 0000 0000) in:

*LDI R19,0b00000000
STS CANIDT1,R19
STS CANIDT2,R19*

```

STS CANIDT3,R19
STS CANIDT4,R19

;hoofdprogramma

IN R18,PINC           ;lees schakelstand in
OUT PORTA,R18        ;schakelstand op poort B
STS CANMSG,R18       ;schakelstand als message

LDI R16,0b00000010   ;CAN enable bit op 1
STS CANGCON,R16     ;message wordt nu verzonden

```

*/*VRAGEN en OPGAVEN*

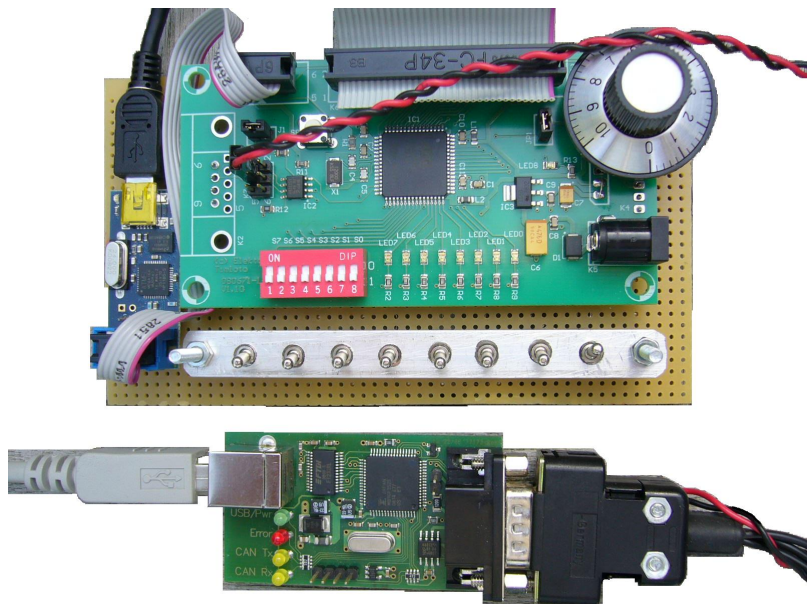
1. *Assembleer en download het programma en controleer of het geheel werkt door de oscilloscoop aan te sluiten op pin 1 (min) en pin 3 (plus) van de K7 aansluiting op de print.*
2. *Sluit nu de USB-CAN adapter aan en controleer of de verschillende standen van de schakelaars worden uitgezonden. Gebruik het filter zodat alleen de geselecteerde identifier zichtbaar wordt.*
3. *Tel nu 01h op bij de schakelstand en verklaar de uitgezonden data.*/**

3.0.4 Toelichting bij het CAN-programma

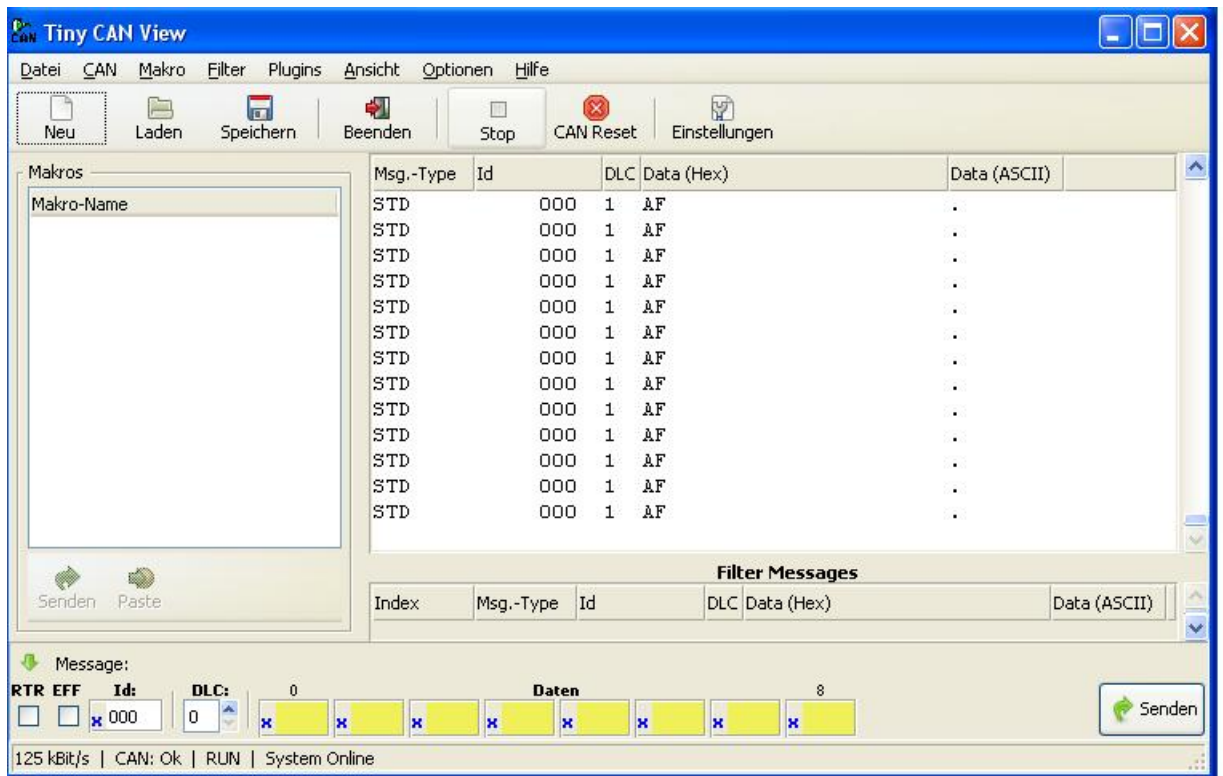
CAN-berichten zijn eigenlijk niet met een oscilloscoop te analyseren. Met de oscilloscoop kan eigenlijk alleen maar de kwaliteit van het elektrische signaal worden bekeken en kan de baudrate van het signaal worden vastgesteld. Om de identifier en de boodschap te bekijken kan gebruik worden gemaakt van een PC en een CAN-interface. Elektor brengt een USB-CAN adapter op de markt voor relatief weinig geld. De bijbehorende software Tiny CAN View is open source. Fig. 2 laat ons de Elektor USB-CAN interface zien terwijl fig. 3 in een de schermafbeelding de werking van het CAN-programma toont.

4 Het ontvangen van een dataframe

Het ontvangen en vervolgens verwerken van een dataframe is wat ingewikkelder en gebeurt met behulp van interrupten. Op dit moment valt het buiten het kader van dit boekje. Er wordt aan een voorbeeldprogramma gewerkt. Zodra het voorbeeld programma operationeel is zal het op de Timloto site worden gezet (onderwijsmatrix, AT90CAN32).



Figuur 2: De USB-CAN interface aangesloten op de AT90CAN32.



Figuur 3: Schermafdruck van het Tiny Can View programma.

5 Vragen en opgaven

1. Wat is het verschil tussen CAN 2.0 A en 2.0 B?
2. Wat verstaat men onder het broadcast principe?
3. Wat verstaat men onder de identifier van de CAN-message?
4. In feite bepaalt een identifier twee zaken. Welke worden bedoeld?
5. Hoeveel identifiers kunnen maximaal door de AT90CAN32 worden uitgezonden?
6. Wat zijn -voor het begrip de twee belangrijkste delen van de totale CAN-boodschap?
7. Kunnen CAN 2.0 A en CAN 2.0 B controllers op dezelfde CAN-bus actief zijn?
8. Wat verstaat men onder CANbit timing?
9. Noem een veel voorkomende baudrate van een CAN-bus.