

Contents

Förslag för implementationslösning	2
Inledning	2
Funktionsprimitiver	2
Realtidssignaler	2
Programloop	2

Förslag för implementationslösning

Följande är idéer om hur vi ska gå tillväga för att implementera optionshanteringen både på nuvarande MCU och nya OCU.

Inledning

Meddelanden mellan OCU och MCU skickas taktat med PDO-meddelanden. Anledningen till detta är att meddelanden måste komma fram deterministiskt. Att skicka SDO skulle leda till osäkerhet kring när paket kommer fram då de är lägst prioriterade.

Funktionsprimitiver

Ett antal funktionsprimitiver måste identifieras för MCU. Dessa ska tillgänglig- göras över CAN-bussen. En tidig tanke var att genom ett dynamiskt protokoll anropa dessa funktioner från OCU och sedan behandla funktionsanropen på MCU. Detta koncept får skrotas eftersom detta skulle ta en alldeles för stor bit av den tillgängliga bandbredden, samt kan vara lite odeterministisk och svår- testat.

Detta kommer innebära att vi konstruerar ett antal förbestämda register som motsvarar de PDO-paket vi ska skicka. Dessa register kapslas sedan av funktioner som sätter data i korrekt byte för att underlätta för programmeraren.

Realtidssignaler

De flesta realtidssignaler finns redan representerade på CAN-bussen - det handlar om att läsa in dessa för att minska på nödvändig bandbredd. Fyra CAN-meddelanden under 20 ms är 25 % av bandbredden.

Signalerna kommer att skickas som olika PDO-objekt. Dessa är fördefinierade och både MCU och OCU vet vad dessa betyder. Det är också implicit att dessa skickas varje loop (20:e ms). Det kanske kan te sig onödigt att skicka dessa värden hela tiden, ur bandbreddssynpunkt, eftersom de inte alltid ändras. Detta är en *kostnad* som tas för att få ett deterministiskt beteende.

PDO-objekten delas upp enligt följande:

- PDORx1 - Objekt som innehåller input från knappar och liknande. Skickas från MCU till OCU.
- PDOtx1 - Innehåller begränsning/önskat värde av hastighet, hydraulik och andra driv-egenskaper. Skickas från OCU till MCU.
- PDOtx2 - Innehåller displaydata. Skickas från OCU till MCU.

Programloop

Beskrivning över respektive enhets huvudprogramloop.