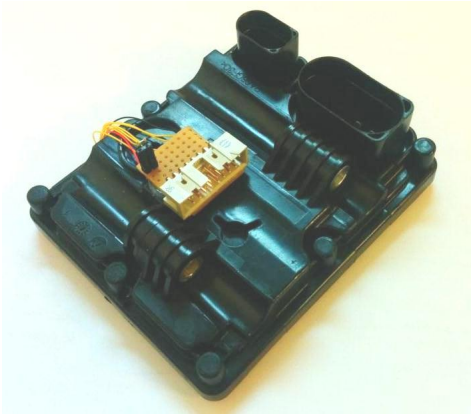
JTAG-optimering för CANoe

Anton Karlsson, *BorgWarner TorqTransfer AB, Lunds Universitet, Institutionen för Reglerteknik*

May 28, 2014

I dag genomförs mjukvaru-tester på en - för testet - specialanpassad mjukvara för att evaluera intern data i ECU:n (Electric Control Unit). Detta är inte önskvärt och istället har en metod där denna data läses från ECU:n med en JTAG-debugger för att sedan integreras i testmiljön tagits fram. Med denna metod kan *vissa* tester genomföras på en mjukvara som är den absolut sista kund-versionen. Men på grund av asynkron läsning av den data som evalueras uppstår viss problematik och metoden kan i nuläget inte fullständigt ersätta konventionell testmetod.



Figur 1: *En ECU med JTAG-kontakt monterad.*

Introduktion

På BorgWarner TorqTransfer utvecklas och tillverkas kopplingar för styrning av fyrhjulsdrift till personbilar. Grundidén bygger på att bilen under normala omständigheter är tvåhjulsdriven till dess att något i omgivningen - körsituation, väglag, etc - kräver att moment förses till den normalt icke drivna axeln.

Hur mycket moment som ska överföras styrs av en ECU som hela tiden övervakar bilens attribut så som hjulhastigheter, motorlast, etc. Mjukvaran i ECU:n genomgår - innan leverans - genomgående tester för att verifiera att funktionen är sådan som kunden efterfrågat. Dessa tester genomförs genom att koppla upp ECU:n mot en test-rigg som innehåller den väsentliga hårdvara som behövs och som simulerar den bil som ECU:n senare ska monteras i. Testaren använder ett program kallat CANoe [1] för att implementera de olika testfallen.

Många testfall kräver åtkomst till intern data i ECU:n, detta åstadkoms konventionellt genom en specialanpassad mjukvara som skickar denna data (tillsammans med all icke-testspecifik data) över den kommunikationslänk som finns mellan ECU:n och test-riggen. Detta medför att det inte är kundmjukvaran som testas utan en test-anpassad sådan, denna artikel beskriver en metod som gör att samma tester ska kunna genomföras på ”ren” kundmjukvara.

JTAG

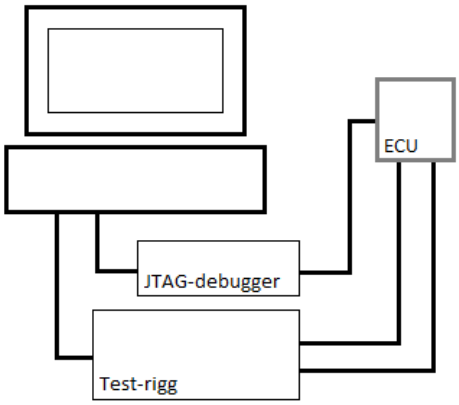
JTAG [2] är ett sätt att felsöka och komma åt kod och register i bland annat mikrochip. Verktyget lämpar sig för att läsa data i minnet på det mikrochip som sitter i ECU:n. Det förutsätter en enskild kontakt för att koppla upp en JTAG-debugger mot ECU:n, se Figur 1.

För att komma åt JTAG-debuggerns funktionalitet körs ett program kallat Trace32 [3] där man kan läsa och skriva till chipets minne, stoppa exekveringen av kod på specifika ställen osv.

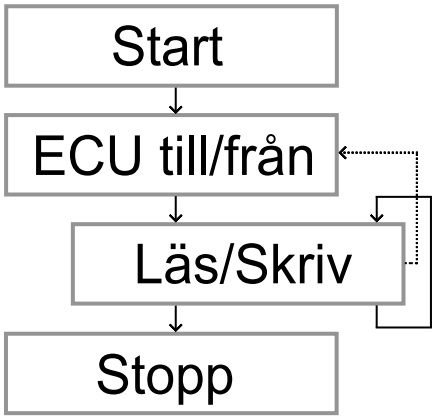
Metod

Den fysiska uppställningen visas förenklat i Figur 2 där användarens PC, JTAG-debuggern, test-riggen och ECU:n visas.

För att sammanfatta; testen genomförs i CANoe



Figur 2: *Testuppsättning för test av ECU-mjukvara.*



Figur 3: *Arbetsflöde för det program som läser data från ECU:n och integrerar densamma i test-miljön.*

och den data vi önskar ha åtkomlig där är möjlig att göra tillgänglig i Trace32. Så hur kan vi integrera den data som kan läsas i Trace32 i CANoe?

Bägge programvaror har ett så kallat API (Application Programming Interface) som gör att man kan komma åt funktionalitet som finns i programvaran från ett externt program. Genom att skriva ett sådant externt program och inkludera både CANoes och Trace32s API öppnas möjligheten för att på ett kontrollerat sätt läsa data från ECU:n med hjälp av Trace32s API och skriva den data till CANoe via dess API. Detta program är framtaget med Visual Studio 2010 C++ [4] och användandet av CANoes API dikterar att detta program körs *på* test-riggen - som i sig består av en dator. Därifrån kommunicerar detta program med Trace32 som körs på användarens PC för att läsa minnet i ECU:n och sedan skriva till CANoe, se flödesöversikt i Figur 3.

När ett test startas kallas automatiskt ”Start”-blocket där initialisering av kommunikation till Trace32, inläsning av information så som på vilken plats i minnet på ECU:n den data vi är intresserad av finns osv sker. Programmet väntar sedan i ”ECU till/från”-blocket tills ECU:n är korrekt spänningssatt varvid ”Läs/Skriv”-blocket tar över. Här ligger programmet och loopar med en period i storleksordningen 50 ms tills det att spänningen slås ifrån till ECU:n (ett test innehåller ofta flera till/från-cykler) då programmet återigen hamnar i ”ECU till/från”-blocket eller att testet avslutas där ”Stopp”-blocket stänger ner all uppstartad kommunikation osv.

Resultat och diskussion

Principiellt är konceptet med att integrera data läst via JTAG i CANoe bekräftat - det går att genomföra vissa test på en ECU med kundmjukvara på detta sätt.

Det finns emellertid tester som konsekvent inte går igenom med denna metod. Detta beror på att läsningen av data i minnet sker asynkront med exekveringen i ECU:n - viss data genomgår en behandlingsprocess varje cykel i ECU:n som är i storleksordningen 20 ms lång där den data vi vill läsa kanske är valid endast under de - exempelvis - sista 5 ms av cykeln. Problemet skulle eventuellt kunna lösas genom att lägga till en variabel i mjukvaran som säger att ”nu är det okej att läsa variabel X och Y” eller att etablera en hårdvaru-synkronisering mellan ECU och JTAG-debugger. Det förstnämnda förslaget leder till att mjukvaran måste anpassas - något som ville undvikas, det andra leder till mer hårdvaru-anpassning och högst sannolikt också ändringar i mjukvaran. Vi kan alltså konstatera att det är mycket svårt att genomföra alla tester utan att anpassa mjukvaran över huvud taget.

Referenser

[1] *ECU Development and Test with CANoe*
`http://vector.com/vi_canoe_en.html`
Besöktes: 2014-05-19

[2] *Standard Test Access Port and Boundary Scan Architecture*
IEEE 1149.1-2001
2008

[3] *Trace32 In-Circuit debugger*
`http://www.lauterbach.com/tutorial.pdf`
Besöktes: 2014-05-16

[4] *Visual Studio 2010*
`http://msdn.microsoft.com/en-us/library/dd831853(v=vs.100).aspx`
Besöktes: 2014-05-16