Versioner/ändringar	
1.0	2003-07
1.01	BUGFIX: FLASH unsecure algoritm
1.02	Fördröjning vid flash-prog. Statusutskrift under programmeringen. Programmeringsalgoritmen hanterar nu även 'bankade' minnet. Översättning samma som ETERM/XCC
1.1	Hanterar kristallfrekvenser 4 resp 8 MHz. Automatisk utfyllnad vid försök att programmera udda antal bytes.
1.2	'Mass Erase' raderar nu även EEPROM (DG256). Efter 'Mass Erase' programmeras nu 'security' bitarna till 'unsecured' automatiskt.

Introduktion

BDM12 är ett enkelt användargränssnitt till POD10. Tillsammans ger det möjlighet att initiera och testa program i en HCS12-baserad mikrodator. Version 1.2 har testats för microlf mc12 och microlf mc12s men såväl POD10 som BDM12 är av generell natur och fungerar med flertalet mikrodatorer baserade på Freescale HCS12.

Filer

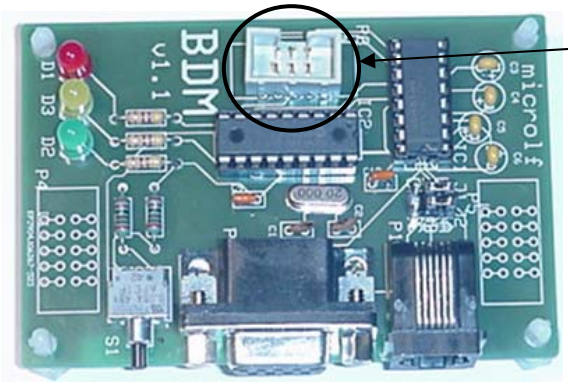
Följande filer används av POD10:

- BDM12.EXE, Windows-applikation
- POD10.DLL, drivrutiner för gränssnittet RS232 (COMport)/POD10
- FLERASE-MC12.S19, laddfil, används av BDM12 för att radera FLASH-minne
- FLPROG-MC12.S19, laddfil, används av BDM12 för att programmera FLASH-minne.

Innan Du börjar...

För att BDM12 ska kunna upptäcka POD10 måste du
Ansluta POD10 via en seriekabel till din PC
Koppla en standard RS232/MCxx kabel mellan din PC (9-polig DIN) och POD10 (RS45).

Koppla POD10 till din måldator
Anslut den 6-poliga flatkabeln mellan POD10 och måldator. Tänk på att POD10 strömförsörjs via denna kabel, spänningssätt måldatorn *efter* inkoppling till POD10.



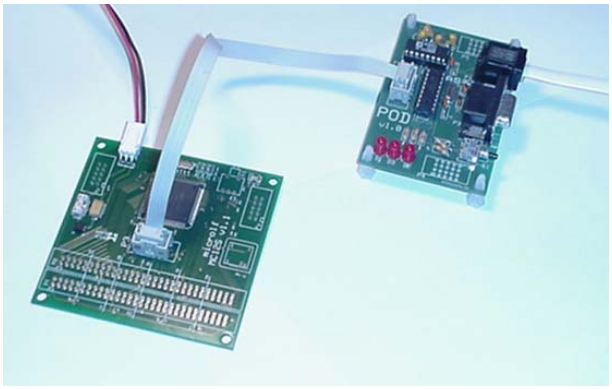
Anslutning (BDM) till måldator

Anslutning till måldator kan göras *antingen* via DSUB9 kontakten *eller* RJ45 med en vanlig terminalkabel för exvis. MC12.

Därefter kan du ansluta matningsspänningen till måldatorn.

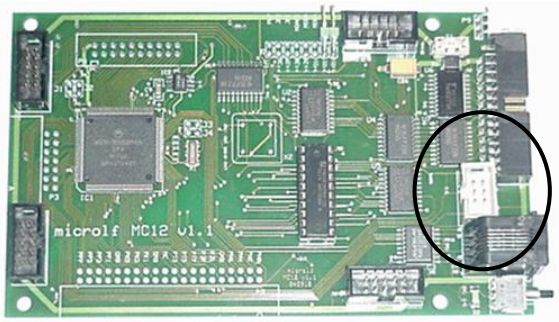
© Copyright, 2003-2007. GMV. Får kopieras fritt.

Anslutning till MC12s



Bilden visar en komplett koppling, MC12s, POD10 med strömförsörjning och kommunikation med värddator.

Anslutning till MC12



BDM-kontaktens placering på MC12.

POD10

POD10 är försedd med kontakt för anslutning till seriekommunikationsport (RS232/RJ45). Ytterligare en kontakt (6-polig stiftlist) är en standard BDM-anslutning till Freescale HCS12-baserade mikrodatorer. POD10 har tre olika ljusdioder för att indikera:

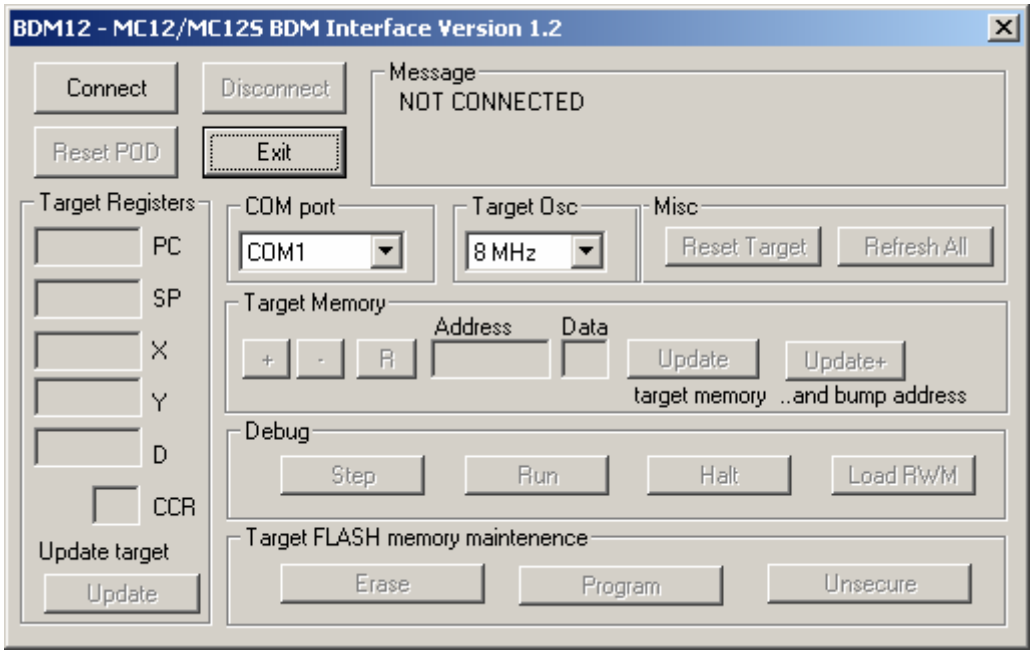
- L1 (grön) POD10 är spänningssatt
- L2 (gul) POD10 utför kommando och är 'upptagen', normalt är detta enbart snabba blinkningar
- L3 (röd) POD10 är i något feltillstånd, detta varar i c:a 1 sekund varefter POD10 återstartar.

Dessutom finns en RESET-knapp på POD10, det är ovanligt att denna behöver användas men i förekommande fall kan den användas för att återstarta POD10.

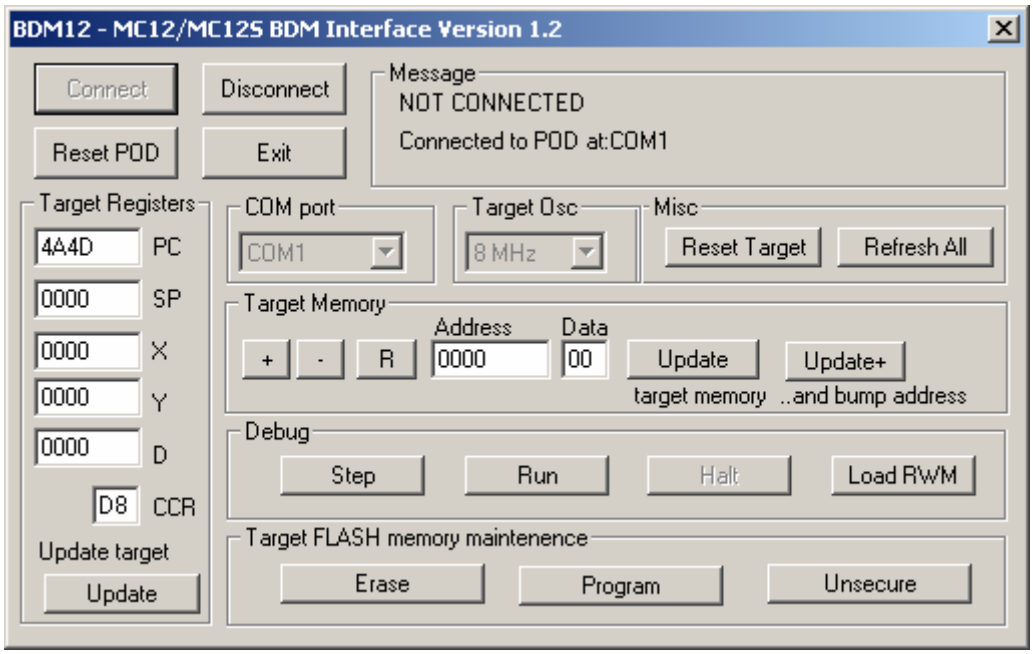
© Copyright, 2003-2007. GMV. Får kopieras fritt.

BDM12 - Användargränssnitt

Starta BDM12 från program-menyn.



'Target Osc' är kristallfrekvensen hos måldatorn (HCS12). Du kan välja mellan 4 och 8 MHz. MC12 och MC12s är bestyckade med 8 MHz kristaller. Du måste också välja den seriekommunikationsport du kopplat till POD10. Välj den porten och klicka på 'Connect'.



Message:
Här visas de tre senaste meddelandena från BDM12. Varje val du gör resulterar i någon typ av meddelande och du kan här se resultatet av ditt val.

Disconnect:
Välj detta om du valt fel kommunikationsport och vill frigöra den valda porten.

© Copyright, 2003-2007. GMV. Får kopieras fritt.

Exit:
Avslutar programmet.

Reset POD:
Frikopplar och återansluter porten, skickar därefter en ny initieringssekvens till POD10. Använd detta om den gula dioden lyser på POD10. Det kan då också vara nödvändigt att trycka ned RESET-knappen på POD10.

Misc:
Reset Target
Detta behöver du säkert göra ofta. Kommandot gör att POD10 utför RESET på måldatorn och samtidigt håller MODC låg, vilket innebär (om måldatorn byglats med MODA=LÅG, MODB=LÅG) måldatorn startas om i SPECIAL SINGLE CHIP-mod med aktiv BDM-debugger, dvs POD10.

Refresh All
Uppdaterar BDM12's fönster med måldatorns registervärden och minneinnehåll. Normalt behöver du inte använda detta (fönster uppdateras efter varje kommando) men är du osäker så är detta en metod att tvinga fram data från måldatorn och uppdatera fönstren i BDM12.

Target Registers:
Visar måldatorns registerinnehåll. Om 'FFFF' visas här betyder det ofta att POD10 ännu inte lyckats hämta de verkliga registrens innehåll, utför då *ResetTarget* och se om innehållen ändras. Om inte, ska du kontrollera anslutningen mellan POD10 och måldatorn.
Du kan sätta registerinnehållen hos måldatorn genom att skriva in nya värden i dessa fönster. Observera dock att dessa värden måste uppdateras, dvs. skickas till måldatorn (se nedan) för att ha effekt.

Update:
Uppdaterar måldatorns registerinnehåll. Samtliga registervärden uppdateras samtidigt.

Target Memory:
Här kan du visa och ändra minnesinnehåll i måldatorn.

Address-Edit:
Anger adress till måldatorns minne. Du kan skriva in en godtycklig adress här (hexadecimal form) och kan sedan använda någon av nedanstående funktioner.

Data-Edit:
Anger data i måldatorns minne (hexadecimal form). Om du ändrat adressen kan du utföra 'R' (Refresh) för att läsa data från måldatorn. Detta fönster uppdateras då. Om du vill ändra data i måldatorn, skriv nytt data i detta fönster, kontrollera därefter att adressen är korrekt (ändra om du vill) och klicka sedan på 'Update' (se nedan)
R:
Refresh - Uppdaterar data-fönstret med innehåll enligt adress-fönstret. Påverkar inte måldatorns minne.
+
Ökar adressen i adressfönstret (med 1), uppdaterar därefter data-fönstret. Påverkar inte måldatorns minne.
-
Minskar adressen i adressfönstret (med 1), uppdaterar därefter data-fönstret. Påverkar inte måldatorns minne.

© Copyright, 2003-2007. GMV. Får kopieras fritt.

Update
Uppdaterar måldatorns minne på den adress som anges i adress-fönstret med data som finns i data-fönstret.

Update+
Uppdaterar måldatorns minne på den adress som anges i adress-fönstret med data som finns i data-fönstret. Ökar därefter adressen i adressfönstret (med 1) och visar måldatorns minnesinnehåll på den nya adressen. Det nya minnesinnehållet ändras inte.

Debug
Dessa kommandon ger dig möjlighet att testa och 'avlusa' dina program med hjälp av BDM12/POD10. Du börjar vanligtvis med att ladda ditt program till RWM-minnet (Load RWM). Kontrollera först att du använt en giltig minnesarea. MC12/MC12S har 8 kByte RWM som relokerats till adress 0 vid RESET. Samtidigt har HCS12 en registerarea som har prioritet före RWM-minnet, mellan 0-3FF. Det finns också ett EEPROM-minne (4 kByte) som relokeras till adress 0 vid RESET. Det finns flera möjligheter att hantera detta men om du inte, just nu, vill veta alla detaljer kan det vara bra att veta att minnesområdet 1000-1FFF är tillgängligt för dina enkla applikationer. Om du använder BDM12 för att programmera FLASH-minne måste du dock veta att detta minne används för programmeringen (av BDM12) och detta kommer då eventuellt att skriva över den kod du laddat till RWM (se 'embedded algoritms').

Load RWM
Ladda till RWM, du anger en *.s19 fil, med giltiga adresser (ingen kontroll utförs)

Step
POD10 skickar ett 'TRACE'-kommando till måldatorn. Instruktionen som finns på nästa adress utförs. Du måste ha initierat PC korrekt först (Target registers), efter utförd instruktion uppdateras alla fönster.

Run
POD10 skickar ett 'GO'-kommando till måldatorn. Om programmet inte avslutats med en 'BGND'-instruktion kommer BDM12 att vara ganska grå, du kan dock klicka 'Halt' (se nedan).

Halt
Avbryt pågående exekvering, POD10 avbryter måldatorns programexekvering, alla fönster uppdateras med aktuella registerinnehåll såväl som minnesinnehåll.

Target flash memory maintenance
För att dessa kommandon ska lyckas krävs att måldatorn är i SPECIAL SINGLE CHIP-mode. Flera kommandon kräver NORMAL-SINGLE-CHIP mode och BDM12 sätter automatiskt om måldatorn då sådana kommandon utförs. Innan du utför någon av följande kommandon måste du därför försäkra dig om att måldatorn verkligen är i SPECIAL-mode, det enklaste sättet är att först utföra Reset Target.Därefter kan följande kommandon utföras:
Erase
Kommandot utför 'mass' ('bulk') erase, dvs försöker radera måldatorns hela FLASH-minne. Detta gäller såväl obankade områden (4000-7FFF, C000-FFFF) som alla bankade områden (8000-BFFF i alla bankar, resten av FLASH-minnet). Om du vill göra 'Unsecure' måste detta kommando ha utförts först eftersom HCS12 inte accepterar 'Unsecure' om något FLASH eller EEPROM är programmerat (icke-raderat).

© Copyright, 2003-2007. GMV. Får kopieras fritt.

Program
Kommandot används för att programmera FLASH-minnet. Du måste ange en ".s19"-fil, exempelvis skapad under ETERM eller XCC för HC12. För att initiera 'boot'-sektorn korrekt (FFF0-FFFF) bör du konsultera färdiga exempel på så kallade 'boot-loaders'. Du finner sådana på GMV's hemsida.
Anm: Utförandet använder den 'inbäddade' algoritmen " FLERASE-MC12"
Anm: Utförandet använder den 'inbäddade' algoritmen " FLPROG-MC12"

Unsecure
Kommandot utför 'Unsecure'-algoritmen, dvs försöker programmera om FLASH-minnet så att skyddsmekanismerna sätts ur spel. Detta lyckas inte alltid beroende på om FLASH-minne och EEPROM är raderat, respektive den MODE som måldatorns processor är i. Ett säker tecken på att måldatorns FLASH-minne är 'unsecured' är att register X innehåller 0000 efter ResetTarget (FF01 betyder att FLASH-minnet har 'security bits aktiverade).

För vidare information om MC12/MC12S och POD10 eller andra frågor om GMV/microlf's utbud av system baserade på Freescale Star-12, se GMV's hemsida.(www.gbgmv.se).

Du kan också skicka e-brev med förfrågningar till
mc12@gbgmv.se

© Copyright, 2003-2007. GMV. Får kopieras fritt.