Ausgewählte Kapitel eingebetteter Systeme

Die Motorola HC12 Mikrocontrollerfamilie

Bearbeitet von Lukasz Fedorowicz

Betreuer: Prof. Dr.-Ing. Wolfgang Schröder-Preikschat

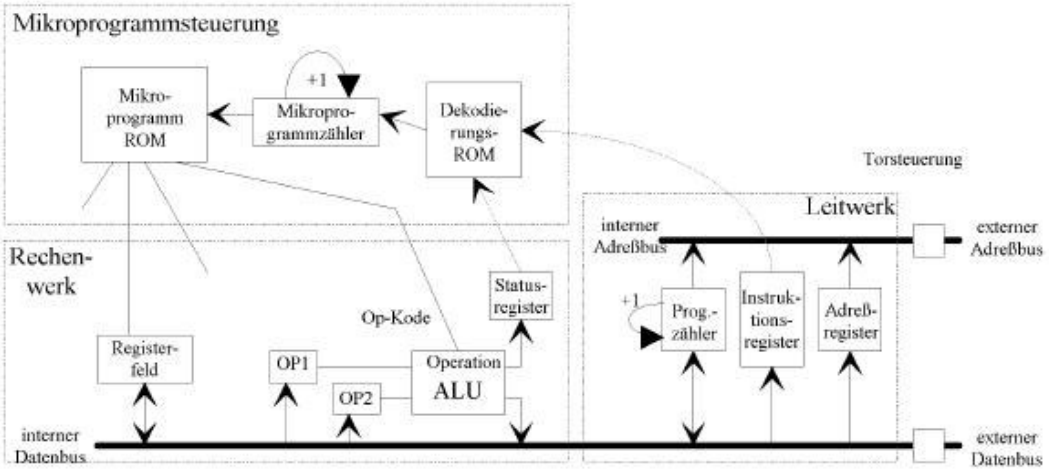
Inhaltsverzeichnis

1. Einleitung

1.1 Geschichtlicher Überblick

Motorola wurde 1928 als „Galvin Manufacturing Corporation“ (GMC) gegründet und 1947 in Motorola - eine Wortschöpfung aus Motor (motorcar, motion) und ola (Schall, Welle, la ola) - umbenannt. Ihr erster Erfolg kam 1930 mit dem ersten kommerziellen Autoradio. Anfang der 40er Jahre verlagerte sich das Interesse auf die Funktechnik.
Schlagzeilen machte das Unternehmen, als die Raumsonde Mariner II, die unterwegs zur Venus war, mit Motorolas Funktechnik ausgestattet wurde und 1969 als Neil Armstrongs legendäre Worte vom Mond zur Erde durch einen Radiotransponder dergleichen Firma übertragen wurden. In den folgenden Jahren konzentrierte sich das Unternehmen auf Mobilkommunikation, Halbleiter, Netzwerke und integrierte Elektroniklösungen.
Bis 1985 war das Unternehmen deutlich vor Intel der Vorreiter auf dem Markt der Prozessoren. Die Prozessoren wurden verbaut zum Beispiel im Apple (Lisa und Macintosh), im Commodore Amiga, Atari ST und Sinclair QL (68008) sowie in Spielkonsolen wie dem Sega Mega Drive oder dem NeoGeo.

1.2 Aufbau eines Mikroprozessors



Ein Mikroprozessor kann im Grunde genommen in drei Teile gegliedert werden

a) Steuerwerk aus Mikroprogrammsteuerung und Leitwerk

b) Rechenwerk

Das Steuerwerk - der Kern des Mikroprozessors - besteht aus dem Befehlsregister und dem Befehlsdecoder. Beide sind für die Ausführung der Befehle und die Koordination der Funktionseinheiten zuständig. Hier wird jeder Befehl nach dem **Von-Neumann-Zyklus** verarbeitet (laden, entschlüsseln, ausführen)
Ein Programmzähler gibt die Adresse im Hauptspeicher der nächsten Instruktion an und ein Instruktionsregister enthält den Code, der aktuell ausgeführt werden soll.

Das Rechenwerk wird auch als Operationswerk bezeichnet und besteht aus einer ALU sowie diversen Hilfs- und Statusregistern. Die ALU ist für arithmetische und logische Funktionen zuständig. Sie verknüpft die Werte der Operandenregister nach den Vorgaben des Steuerwerkes und legt das Ergebnis auf den internen Datenbus in das Statusregister ab. Dieses bildet einen kleinen „Spezialspeicher“ für Zwischenergebnisse von logischen und arithmetischen Operationen auf die besonders schnell zugegriffen werden kann.

Der Registerblock hat als Aufgaben die Steuerung des Ablaufs der Befehlszyklen in der CPU (Steuerung des Programmablaufs), Befehle holen und decodieren
Operanden holen/speichern, Steuerung der Befehlsausführung, Koordination und Steuerung der einzelnen Werke der CPU.

Zur Erhöhung der Rechengeschwindigkeit können Prozessoren mit Caches oder besonderen bzw. zusätzlichen Rechenwerken ausgestattet sein (z.B. Fließkommarechenwerk). Zur effizienteren Bearbeitung von Befehlen werden Pipelines verwendet. Alle komplexeren Mikroprozessoren sind interruptfähig, d.h. eine Unterbrechung des Programmablaufes wird durch ein externes Signal bewirkt.

2. Vorstellung der HC Motorola Familie

µcontroller 68HC11

Vorgänger des 68HC12 Mikroprozessors war der 68HC11 in einem PLCC-Gehäuse mit bis zu 52 Anschlüssen. Dieser ist relativ weit verbreitet, verfügt über eine 8-Bit Architektur und eine breite Auswahl an Fachliteratur. Um den Umstieg zu erleichtern und die 68HC11 Fangemeinde zu behalten, wurde der 68HC12 zu seinem Vorgänger softwarekompatibel entworfen. Die Struktur der CPU-Register, die Assemblerbefehle und die Adressierungsarten sind teilweise um neue Features erweitert worden – bleiben aber zum Vorgängermodell kompatibel. Der Umstieg ist dadurch wesentlich erleichtert worden. Der Umsteiger - Entwickler fühlt sich in der neuen Umgebung sofort vertraut und der Einsteiger wird mit einer beherrschbaren Komplexität konfrontiert.

Die HCS12 Mikrokontroller locken mit einem handlich verpackten 16 Bit Chip, einer BDM[1] Schnittstelle mit der Möglichkeit Breakpoints zu setzen, großer Auswahl an Softwaretools, geringen Kosten, seriellen Schnittstellen (SCI, SPI, IIC, CAN), Timersystem mit Capture und Compare Einheiten, Pulsakkumulator, sowie div. Überwachungsfunktionen wie Clock Monitor, Watchdog und Key Wakeup. Weiterhin verfügt er über einen 10 Bit A/D Wandler welcher Spannungspotential in digitale Spannungspegeln wandelt, bis zu 1 KB RAM, 4 KB EEPROM und eine interne Taktfrequenz von max. 25 MHz. Das ganze verpackt in einem Flat-Pack Gehäuse in QFP-Bauform, dessen Größe die der HCS11 Familie nicht übersteigt.

[1] Background Debug Mode

2.1 68HC12 Typen im Überblick

	812A4	912B32	912BC32	912D60	912DA128	912DG128	912DT128
Prozessorkern	CPU12	CPU12	CPU12	CPU12	CPU12	CPU12	CPU12
Betriebsspannung	4,5-5,5V	4,5-5,5V	4,5-5,5V	4,5-5,5V	4,5-5,5V	4,5-5,5V	4,5-5,5V
Flash	-	32K	32K	60K	128K	128K	128K
EEPROM	4K	768	768	1K	2K	2K	2K
RAM	1K	1K	1K	2K	8K	8K	8K
Businterface	Non-MUX (a)	MUX	MUX	MUX	MUX	MUX	MUX
Bustakt ECLK max.	8 MHz	8 MHz	8 MHz	8 MHz	8 MHz	8 MHz	8 MHz
PLL	x	-	-	x	x	x	x
Timermodul	TIM	TIM	TIM	ECT	ECT	ECT	ECT
PWM (Kan. x Bit)	-	4 x 8 (b)	4 x 8 (b)	4 x 8 (b)	4 x 8 (b)	4 x 8 (b)	4 x 8 (b)
SCI	2	1	1	2 (c)	2	2	2
SPI	1	1	1	1	1	1	1
IIC	-	-	-	-	1	1	1
CAN	-	-	1	1	1	2	3
BDLC	-	1	-	-	-	-	-
ADC (Kan. x Bit)	8 x 8	8 x 10	8 x 10	16 x 10 / 8 x 10	16 x 10 / 8 x 10	16 x 10 / 8 x 10	16 x 10 / 8 x 10
BDM	x (d)	x	x	x	x	x	x
I/O Leitungen (max. ohne PORTE)	85	55	55	80 / 52			
Key-Wakeup Lines	24	-	-	15 / 2			
Package	TQFP112	QFP80	QFP80	TQFP112 / QFP80	TQFP112 / QFP80	TQFP112 / QFP80	TQFP112 / QFP80

(a) unterstützt Memory Pageing (max. 5 MB), 7 Chip Selects
(b) oder 2 x 16
(c) optional: 1x SCI + 1x MI-BUS
(d) jedoch ohne Hardware-Breakpoints

TIM = Standard Timer Module
ECT = Enhanced Capture Timer
MUX = Multiplexed External Bus Interface
Non-MUX = Non-Multiplexed Ext. Bus Interface

2.2 68HC12A Familie

Die Familie der 68HC12A Mikrocontroller besteht aus den beiden Derivaten MC68HC812A0 und MC68HC812A4, wobei HC812A0 nicht mehr lieferbar ist.

Somit stellt der HC12A4 mit seinem 1K integrierten Speicher RAM und 4K EEPROM den Grundtyp dar, der bereits unter 10,- EUR zu haben ist. Die CPU hat – wie alle HC12 Derivate – eine 16 Bit-Zentraleinheit, Timermodul mit Capture/Compare zur Feststellung bzw. Erzeugung von zeitlich definierten Funktionen, ein synchrones serielles Interface (SPI) (von Motorola entwickeltes Bus-System mit dem digitale Schaltkreise miteinander verbunden werden können, besonders für Mess- und Audioanwendungen interessant) und einen 8-Bit A/D-Wandler mit 8 Kanälen. Außerdem findet man auch noch zwei asynchrone serielle Schnittstellen SCI0 und SCI1 über die Programme geladen werden können, sowie einen Timer- und Pulsakkumulator – Modul mit 8 Kanälen und 16 Bit Zählern, mit dem Ereignisse gezählt werden können. Die Betriebsspannung beträgt 5 V und muss gleichmäßig auf alle Pins verteilt werden.

2.3 68HC12B

Wie die komplette 68HC12 Familie, ist auch dieser ein CISC – Mikrocontroller und verfügt über fast 200 verschiedene Assemblerbefehle mit teilweise komplexer Wirkungsweise. Diese Derivate erhielten als erstes integrierten Flash-Memory auf dem Chip. Dieser war für 100 Lösch- und Schreib-Zyklen mit 10 Jahren Datenerhaltszeit spezifiziert. Zusätzlich wurde das Background Debug Mode (BDM) Interface um die Möglichkeit erweitert, Hardware Breakpoints zu setzen und die Auflösung des A/D-Wandler auf 10 Bit erhöht. Mit einer Umgebungstemperatur zwischen -50 und 150 °C wächst das Anwendungsspektrum über den Innenbetrieb hinaus. Das Gehäuse ist kleiner als das des Vorgängers und verfügt über 80 Pin im QFP Design. Auch das Businterface hat sich geändert – dieses verfügt über ein multiplexes Businterface was soviel bedeutet, dass Adress- und Datenleitungen geteilt werden. Auch ein BDLC[2] Interface kann man vorfinden, dieses
Implementiert den SAE Standard J1850, ein Diagnose - Übertragungsprotokoll in der amerikanischen Automobilindustrie.

2.4 68HCBC32

Die beiden Derivate B32 und BC32 sind pinkompatibel, der BC32 hebt sich nur mit seinem msCAN12 – Modul ab. Statt des amerikanischen BDLC Interfaces ist hier ein CAN[3]-Modul integriert, eine Reaktion auf die Anforderungen des europäischen Marktes.

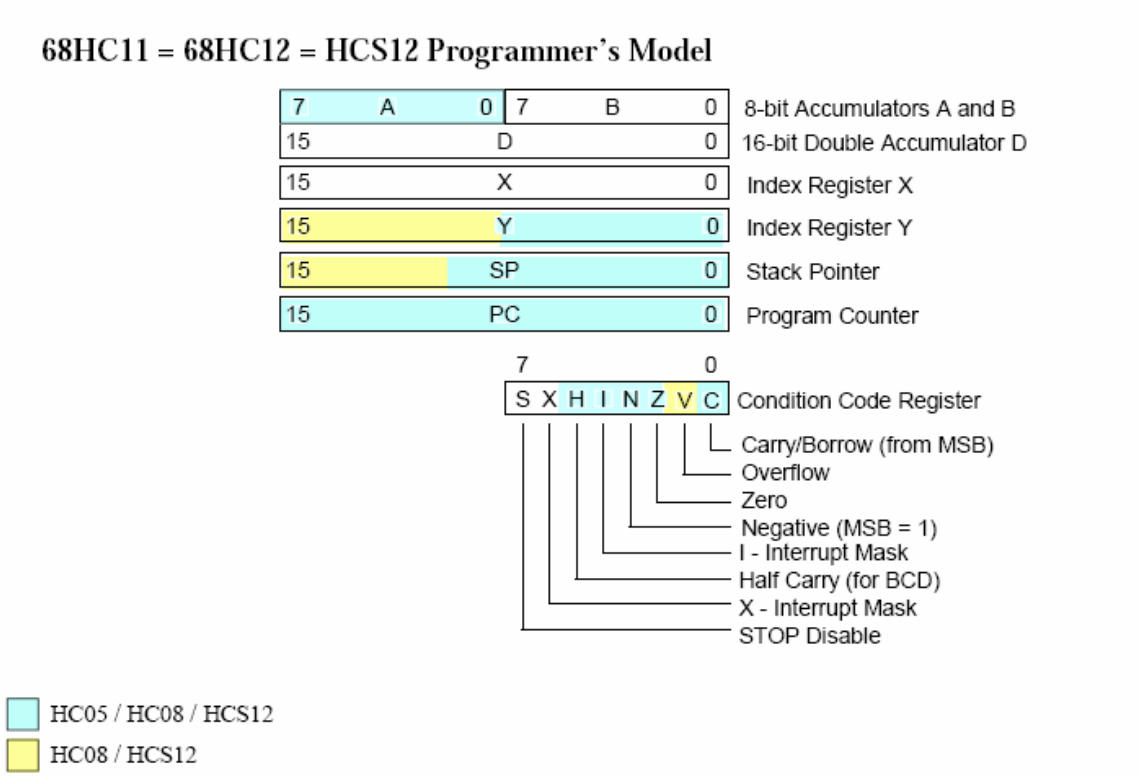
[2] Byte Data Link Communication
[3] Controller Area Network – „Netzwerk aus mehreren Mikrocontrollern“

2.5 68HC12D-Familie

Die neueste D-Familie kam nicht nur mit den heiß ersehnten CAN–Modulen, zwei davon beim 912DG128 und sogar drei beim 912DT128, sondern erweiterte das Angebot an Flashspeicher auf respektable 60 KB bzw. 128 KB. Zudem verfügt der Baustein über eine zweite SCI Schnittstelle und die PLL[4] zur Takterzeugung. Auch in RAM und EEPROM unterscheiden sich die Derivate der D–Familie in ihrer Größe, die Auswahl reicht von 2K bzw. 1K bei der 912D60 Version bis hin zu 8K, bzw. 2K in der 912DT128 Version. Zwei Der A/D Wandler wurden integriert die insgesamt 16 Analogkanäle zur Verfügung stellen (im TQFP112 Gehäuse).

[4] Phase Locked Loop

2.6 Das Programmiermodell



Unter einem Programmiermodell versteht man Architekturmerkmale und Vorstellungen bzw. Konventionen (Vereinbarungen, Vorschriften) zu deren Nutzung.
Die CPU12 ist eine Akkumulator-Architektur: Das heiß, dass sich Operanden und Ergebnisse der Operationen (z.B. Addition) in den Akkumulatoren befinden. Aus Kompatibilitätsgründen wurde das Registermodell des HC12 vom HC11 übernommen. Den Assemblerbefehlen stehen einmal die 8-Bit Register A und B, sowie das Doppelregister D, aus A und B bestehend (vgl. 80x86 Prozessor mit AX, BX, CX, DX 16 - Bit Registern und 8 Bit - Register AH, AL, etc.)

Das Indexregister X und Y werden zur indizierten Adressierung verwendet. Hier wird der Inhalt eines Indexregisters als Offset zum Inhalt eines Akkumulators oder einer Konstanten addiert. Das Ergebnis heißt effektive Adresse und zeigt z.B. auf eine Speicherzelle oder Instruktion.

Der Stackpointer dient als temporärer Zwischenspeicher für Funktionsaufrufe und kann beim Aufruf von Unterprogrammen zur Variablenübergabe genutzt werden. Wird ein Ereignis ausgelöst – so z.B. durch ein Interrupt, so werden auf den Stack die Register und die Rücksprungadresse gespeichert (vgl. 80x86 Prozessor push und pop Operationen).

Der Programmzähler enthält die Adresse der nächsten auszuführenden Instruktion und wird automatisch inkrementiert, sobald eine Instruktion ausgeführt wurde.

Das Statusregister wird bei bestimmten Operationen automatisch geändert und dient der weiteren Auswertung. So kann z.B. nach einer Arithmetischen Operation ein bedingter Sprung erfolgen, falls das Zero Flag eine Null anzeigt.
Beim Stop Flag handelt es sich um eine Anweisung, die den stromsparenden Betrieb indem große Teile des Mikrocontrollers stillgelegt werden, deaktiviert.
Das X und I Interrupt Maskenbit sperrt alle maskierten Interrupts. Tritt ein Interrupt in dieser Phase auf, so wird es gespeichert und kommt erst dann zur Ausführung, wenn das Flag wieder freigegeben wird. Das X Bit steht dabei für den externen Interrupteingang.

2.7 Interruptverarbeitung

Interrupts sind Ereignisse, die die CPU veranlassen die aktuelle Codeausführung zu unterbrechen, den Inhalt sämtlicher CPU-Register zu sichern und zur Interruptserviceroutine zu springen. Ihre Adresse trägt man in eine Interrupt-Vektortabelle ein, die am Ende des HC12 Programmspeichers beginnt und für jeden HC12 Interrupt einen 16 Bit langen Eintrag enthält.
Nach der Abarbeitung der Interruptserviceroutine erfolgt die Rückkehr zum Hauptprogramm mit der RTI-Anweisung. Die CPU Register werden wieder restauriert und das Hauptprogramm weiter fortgeführt – und zwar an der Stelle, an der die Unterbrechung stattfand.
Bei mehreren Interrupts gleichzeitig legt die Reihenfolge in der Vektorrabelle die Priorität fest. Dabei gilt: je höher die Adresse, desto höher ist die Priorität.
Die Interrupt Latency ist die Zeit, die angibt wie viele Taktzyklen im Worst Case vom Auftreten eines Interrupts bis zur Ausführung der Interruptserviceroutine vergehen. Dieses Wissen ist wichtig um Aussagen über Echtzeitanforderungen machen zu können und beträgt 21 Taktzyklen. Sie setzt sich aus 4 Faktoren zusammen:

- Ausführungszeit des am längsten dauernden Befehls (EMACS mit 13 Taktzyklen)
- Zeit um die CPU Register zu sichern (4 Taktzyklen)
- Dauer um den Interruptvektor zu laden (1 Taktzyklus)
- Füllen der Instruktionspipeline (3 Taktzyklen)

Bei einer Busfrequenz von 8 MHz beträgt die Interruptsverzögerungszeit 0,840µs.

2.8 Betriebsarten

Die 68HC12 Mikrocontroller können in acht verschiedenen Betriebsarten, sog. Modes betrieben werden. Diese werden nach dem Start oder Reset an den Pins MODA, MODB und MODC eingelesen und in das MODE-Register eingetragen. Sie können grob in Grund–Modes und Test–Modes gegliedert werden und sind auch während des Betriebes umschaltbar. Je nach der eingestellten Betriebsart sind einige Register und Kontrollbits gegen Änderungen geschützt, bzw. zu Testzwecken beschreibbar.

Table 5-1. Mode Selection

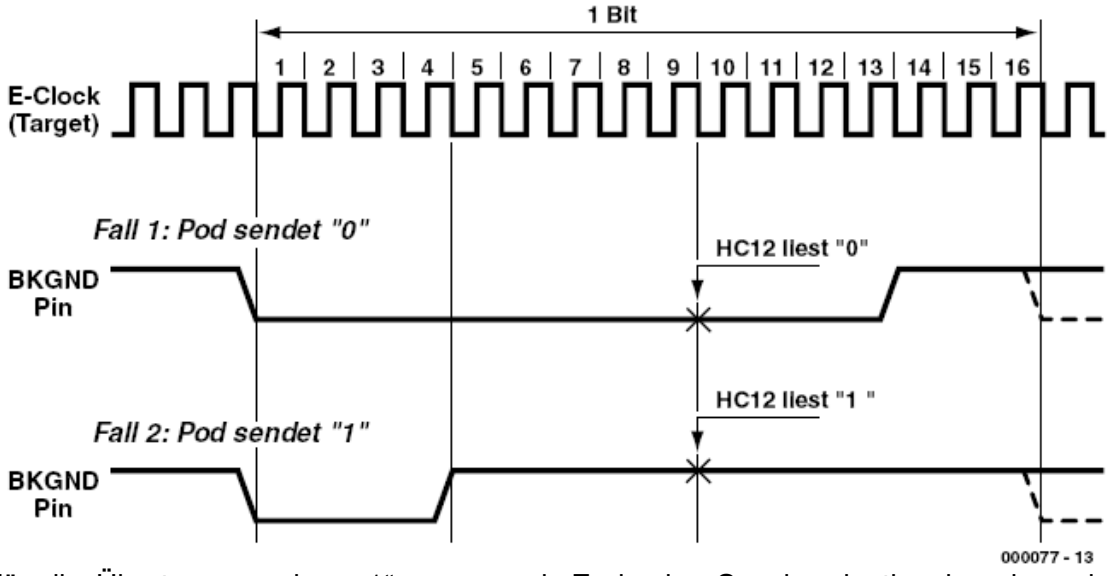
BKGD	MODB	MODA	Mode	Port A	Port B
1	0	0	Normal Single Chip	G.P. I/O	G.P. I/O
1	0	1	Normal Expanded Narrow	ADDR/DATA	ADDR
1	1	0	Reserved (Forced to Peripheral)	—	—
1	1	1	Normal Expanded Wide	ADDR/DATA	ADDR/DATA
0	0	0	Special Single Chip	G.P. I/O	G.P. I/O
0	0	1	Special Expanded Narrow	ADDR/DATA	ADDR
0	1	0	Special Peripheral	ADDR/DATA	ADDR/DATA
0	1	1	Special Expanded Wide	ADDR/DATA	ADDR/DATA

Im Single-Chip-Modus arbeitet der Controller als Einzelchiplösung, d.h. es wird nur der interne Speicher verwendet. Alle Portpins stehen als Ein- und Ausgabepins zur Verfügung. Damit ein externer Speicher angesprochen werden kann, muss der Expanded Modus eingeschaltet werden. Hier fungieren dann die A und B Ports als Daten- und Adressbusse. Diesen Mode kann man noch in narrow und wide unterteilen – je nachdem ob 8, bzw. 16 Bit breite Speicher angesprochen werden sollen. Bei den preisgünstigen 8 Bit breiten Speicherbausteinen muss der Mikrocontroller zweimal zugreifen, um 16 Bit lesen oder schreiben zu können. Bei allen drei Single-Chip-Modes steht der BDM zur Verfügung. Dieses muss allerdings durch ein BDM Kommando aktiviert werden.
Die Spezialbetriebsarten sind für Testzwecke beim Hersteller Motorola reserviert und korrespondieren mit den normalen Betriebsarten. Hier ist der BDM nach dem Reset automatisch aktiviert.
Der Peripheral – Modus schaltet die CPU aus - ein externer Master muss die Kontrolle über die Peripheriebausteine übernehmen.

2.9 Single-Wire Background Debug Mode

Der BDM kommt mit einem einzigen Anschlusspin BKGD aus und funktioniert wie eine bidirektionale serielle Schnittstelle mit einem eigenen Protokoll. Somit werden beim Debuggen keine Ressourcen des Controllers benötigt – wie das z.B. noch beim Vorgängermodell HS11 der Fall war, bei dem das Debugging mithilfe der SCI Schnittstelle geschah. Wie der Name schon sagt, funktioniert das Debugging im Hintergrund, somit kann man z.B. die Inhalte von Speicherzellen auslesen, ohne dafür die Arbeit der CPU unterbrechen zu müssen. Die BDM Befehle bestehen in der Regel aus einem Kommandobyte und gegebenenfalls erforderlichen Parametern.
Der zugrunde liegende Takt richtet sich nach der Taktgeschwindigkeit des HC12. Wird beispielsweise ein 16-MHz-Quarz eingesetzt, ergibt sich eine Basiszeiteinheit des BDM-Systems von 125 ns. Für die Übertragung eines einzelnen Bits sind 16 Taktzyklen erforderlich.

Bei 125 ns pro Takt werden somit 2 µs (16 * 125ns) für ein Bit benötigt, für ein Byte entsprechend 16 µs. In der Praxis bedeutet das ca. 5-10 Tausend Befehle pro Sekunde.
Als erstes muss der Host (= Pod) einen Startimpuls vorgeben. Da der Ruhepegel des BKGD ein HIGH ist, wird die Synchronisation mit einem L-Pegel eingeleitet.



Für die Übertragung einer „1“ muss nach Ende des Synchronisationsimpulses der Host die BKGD Leitung wieder auf HIGH setzen. 4 Takte später tastet der Controller die BKGD Leitung ab und erkennt den HIGH Pegel.
Die Übertragung einer logischen „0“ beginnt ebenfalls mit dem Senden eines Synchronisationsimpulses. Anschließend muss der Host das Signal 13 Takte lang stabil auf LOW halten, dann für die verbleibenden drei Takte zurück auf HIGH. Wie zuvor tastet die MCU 4 Takte nach dem Erkennen der Synchronisation die Leitung ab und erkennt das Null-Bit.
Um die korrekte Arbeitsweise zu gewährleisten, darf die Pause zwischen zwei aufeinander folgenden Bits nicht länger als 512 Takte betragen. Andernfalls wird das Kommando verworfen und das BDM wartet auf ein neues Befehl.

3. Literaturhinweise

Mikrocontroller-Design, Harald Kreidl, 2003 Carl Hanser Verlag
Programming the Motorola M68HC12 Family, Gordon Doughman:

16-Bit-Mikrocontroller HC12 - Überblick für Um- und Einsteiger, 19.05.06
http://dg3aaf.no-ip.com:8080/elektor/16-Bit-Mikrocontroller%20HC12%20(I).pdf

Geschichte, 19.05.06
http://www.motorola.com/content.jsp?globalObjectId=1223-2212

Motorola, 19.05.06
http://de.wikipedia.org/wiki/Motorola

HC12 Review, 19.05.06
http://elmicro.com/hc12web/due9910/index.html

The 68HC912B32: An Overview, 19.05.06
http://www.seattlerobotics.org/encoder/apr98/68hc912intro.html

Introducing the MC68HC12, 19.05.06
http://elmicro.com/files/intro_mc68hc12.pdf

HC12 Web, 19.05.06
http://hc12web.de

HCS12, 19.05.06
http://www.physics.ubc.ca/~scho/9s12/HCS12_sjtu/HCS12_Overview.pdf