ST6-Realizer für ST6-Mikroprozessoren

Der Realizer, die einfache Art zu programmieren, mit einem grafischen Programmierwerkzeug, ohne daß die Kenntnis einer Programmiersprache, wie z. B. Assembler oder C, erforderlich ist.

Allgemeines

Mikrocontroller sind vielseitig und für unterschiedlichste Aufgaben im gesamten Bereich der Elektronik einzusetzen. Mit zu den Marktführern auf dem Gebiet der 8Bit-Mikrocontroller gehört SGS-Thomson mit der ST6-Familie, die zur Zeit mehr als 25 Controller für unterschiedlichste Aufgaben umfaßt.

Der Mikrocontroller kann jedoch die ihm zugedachte Aufgabe erst dann erfüllen, wenn ein entsprechendes Programm erstellt wurde. Ohne umfangreiche Programmierkenntnisse eine schier unlösbare Aufgabe.

Mit dem Realizer geht SGS-Thomson nun neue Wege. Hier handelt es sich um ein grafisches Programmierwerkzeug zum einfachen Programmieren der ST6-Mikrocontroller, ohne daß die Kenntnis einer Programmiersprache erforderlich ist.

Die Software generiert aus der grafischen Symboldarstellung den Programmcode für den entsprechenden Mikrocontroller. Bei der Programmerstellung unterstützt eine umfangreiche Symbolbibliothek die komfortable grafische Eingabe, und am PC-Bildschirm können die erstellten Programme simuliert, analysiert und verbessert werden, bevor die Programmierung des Prozessors erfolgt.

Es wird also mit dem Realizer zunächst nichts anderes als ein Schaltplan erstellt, der dann mit dem Analyzer analysiert wird. Dieser erzeugt daraufhin den Assembler-Code und den endgültigen HEX-Code.

Über den im Paket enthaltenen Simulator kann die erstellte Applikation auf grafischer Basis auf Funktion getestet werden.

Vor der Programmierung ist der für die jeweilige Aufgabe passende Mikrocontroller aus der großen ST6-Familie auszuwählen, wobei nicht zuletzt ökonomische Gründe für die Auswahl entscheidend sind. Ein für die gewünschte Aufgabe überdimensionierter Prozessor verursacht in den meisten Fällen nur höhere Kosten, ohne daß die zusätzlichen Funktionen nutzbar sind.

Bevor wir nun mit der grafischen Programmierung beginnen, zuerst ein kurzer Überblick über die ST6-Mikrocontroller-Familie, die Architektur des Prozessors und die weiteren zur Verfügung stehenden Entwicklungs-Tools.

Einführung in die ST6-Mikrocontroller-Familie

Die ST6-Familie von SGS-Thomson bietet im Bereich der unteren und mittleren Ebene eine Vielzahl von interessanten 8Bit-

Tabelle 1: ST62-Familienübersicht

Device	Program Memory	RAM x 8	EEPROM x 8	A/D Input	WD Timer	Timers	Serial Interface	I/Os (High Current)	Package
ST6200	lK	64		4x8-Bit	Yes	1x8-Bit		9 (3)	DIP16/SO16
ST6201	2K	64		4x8-Bit	Yes	1x8-Bit		9 (3)	DIP16/SO16
ST6203	lK	64			Yes	1x8-Bit		9 (3)	DIP16/SO16
ST6208	lK	64			Yes	1x8-Bit		12 (4)	DIP20/SO20
ST6209	lK	64		4x8-Bit	Yes	1x8-Bit		12 (4)	DIP20/SO20
ST6210	2K	64		8x8-Bit	Yes	1x8-Bit		12 (4)	DIP20/SO20
ST6215	2K	64		16x8-Bit	Yes	1x8-Bit		20 (4)	DIP28/SO28
ST6220	4K	64		8x8-Bit	Yes	1x8-Bit		12 (4)	DIP20/SO20
ST6225	4K	64		16x8-Bit	Yes	1x8-Bit		20 (4)	DIP28/SO28
ST6230	8K	192	128	16x8-Bit 1x16-Bit	Yes	1x8-Bit	SPI UART	20 (4) PWM	DIP28/SO28
ST6232	8K	192	128	21x8-Bit 1xl6-Bit	Yes	1x8-Bit	SPI UART	30 (9) PWM	SDIP42/QFP52
ST6235	8K	192	128	24x8-Bit 1xl6-Bit	Yes	1x8-Bit	SPI UART	36 (12) PWM	QFP52
ST6240B	8K	216	128	12x8-Bit	Yes	2x8-Bit	SPI	24 (4)	QFP80 45x4 LCD
ST6242B	8K	216	128	6x8-Bit	Yes	2x8-Bit	SPI	18 (4)	QFP64 40x4 LCD
ST6245B	4K	140	128	7x8-Bit	Yes	2x8-Bit	SPI	19 (4)	QFP52 24x4 LCD
ST6246B	4K	140	128	8x8-Bit	Yes	2x8-Bit	SPI	20 (4)	SDIP56 24x4 LCD
ST6252	2K	128		4x8-Bit 1x8-Bit AR	Yes	1x8-Bit		9 (5)	Dip16/SO16
ST6253	2K	128		7x8-Bit 1x8-Bit AR	Yes	1x8-Bit		13 (6)	DIP20/SO20
ST6255	4K	128		13x8-Bit 1x8-Bit AR	Yes	1x8-Bit		21 (8)	DIP281S028
ST6260	4K	128	128	7x8-Bit 1x8-Bit AR	Yes	1x8-Bit	SPI	13 (6)	DIP20/SO20
ST6262	2K	128	64	4x8-Bit 1x8-Bit AR	Yes	lx8-Bit		9 (5)	DIP16/SO16
ST6263	2K	128	64	7x8-Bit 1x8-Bit AR	Yes	1x8-Bit		13 (6)	DIP20/SO20
ST6265	4K	128	128	13x8-Bit 1x8-Bit AR	Yes	1x8-Bit	SPI	21 (8)	DIP28/SO28
ST6280B	8K	320	128	12x8-Bit 1x8-Bit AR	Yes	1x8-Bit	SPI UART	22 (10)	QFP100
ST6285B	8K	288	128	8x8-Bit 1x8-Bit AR	Yes	1x8-Bit	SPI UART	12 (4)	QFP80

Abbreviations and Notes

ADC = Analog to Digital Converter UART = Universal Asynchronous Receiver / Transmitter AR = Auto -Reload
SPI = Serial Peripheral Interface WDG = Watchdog H /S PWM = Pulse Width Modulation

Controllern an. Sie zeichnet sich durch hohe Störfestigkeit und geringe Kosten aus.

Basierend auf ein und demselben CPU-Kern gibt es eine große Auswahl an unterschiedlichsten Varianten von DIP/SO16 bis zum 80poligen QFP-Gehäuse. Für jedes Familienmitglied ist ein kostengünstiges Starterkit, worauf wir im weiteren Verlauf des Artikels noch näher eingehen, erhältlich. Das wohl interessanteste Tool zur Software-Entwicklung ist neben einem neuen „C"-Compiler sicherlich der hier vorgestellte Realizer.

Sämtliche Prozessoren sind sowohl in EPROM-Version (mit UV-Licht löschbar) für Prototypen und Kleinserien mit der Möglichkeit zum einfachen Software-Update und als OTP-Variante (one time programmable) lieferbar. Für große Serien sind besoners kostengünstige, maskenprogrammierte ROM-Versionen lieferbar.

Tabelle 1 zeigt die zur Zeit zur Verfügung stehenden Varianten der ST6-Familie im Überblick. Bei den Prozessoren der ST6-Familie wurde besonders viel Wert auf die elektromagnetische Verträglichkeit gelegt, da häufig gerade die Prozessoren im unteren Preisbereich in der Haushalts- und Konsumerelektronik eingesetzt werden. Aufwendige Abschirmmaßnahmen sind dann häufig aus Kostengründen nicht mehr möglich. So muß z. B. in Elektrowerkzeugen und auch in Haushaltsgeräten oft die Elektronik direkt ohne Abschirmung auf einem Motor positioniert werden.

Sämtliche Prozessor-IO-Ports haben eine Schmitt-Trigger-Funktion und sind mit Schutzmaßnahmen gegen eingeprägte Ströme ausgestattet.

Tabelle 2: Software-Tools

Device	Name	Beschreibung
ST 62	ST6-FUZZY/PC	Fuzzy Logic Compiler
	ST6-REALIZER	Grapisches Entwicklungstool
	ST6-SW/PC	Macro-Assembler,Linker & Simulator

Tabelle 3: Hardware-Tools

Device	EPROM Programmierer 1)			Emulatoren 3)		Starter Kit
	Singel Eprom	Komplett Gang	Gang Adapter 4)	Komplett	Dedication Board	
ST620X ST521X ST622X	ST62E2X-EPB 1)2)			ST626X-EMU2	ST626X-DBE	ST622X-KIT
ST623X	ST62E3X-EPB 1)2)			ST623X-EMU2	ST623X-DBE	ST623X-KIT
ST624X	ST62E4X-EPB 1)2)			ST6240-EMU2 ST6242-EMU2 ST6245-EMU2 ST6246-EMU2	ST624X-DBE	ST624X-KIT
ST626X	ST62E6X-EPB 1)2)	ST62E60-GP/SO ST62E60-GP/DIP ST62E65-GP/SO ST62E65-GP/DIP	ST62E60-GPA/SO ST62E60-GPA/DIP ST62E65-GPA/SO ST62E65-GPA/DIP	ST626X-EMU2	ST626X-DBE	ST626X-KIT
ST628X	ST62E8X-EPB 1)2)	ST62E80-GP/QFP ST62E85-GP/QFP	ST62E80-GPA/QFP ST62E85-GPA/QFP	ST6280-EMU2 ST6285-EMU2	ST628X-DBE	

Notes:
1. Jeder EPROM-Programmer unterstützt alle Bauteilgehäuse der Bausteine.
2. Alle Emulatoren enthalten Probes außer denen für die ST624x, ST628x und ST623x (ST6240, ST6242, ST6245-P/QFP, ST6280, ST6285-P/QFP). Alle Emulatoren & Kits enthalten Software (z. B. Macro-Assembler, Linker, Debugger, Simulator).
3. Der Gang-Adapter ermöglicht es, das Bauteilgehäuse zu wechseln, ohne den kompletten Gang-Programmer zu wechseln.

Für zusätzliche Sicherheit sorgt ein Watchdog-Timer, der im Störungsfall einen gezielten Reset auslöst.

Um die Störabstrahlung zu verringern, wurde die interne Architektur so ausgelegt, daß möglichst wenig Transistoren gleichzeitig schalten. Des weiteren wurde zur Verringerung der Störabstrahlung die Flankensteilheit der internen Schaltvorgänge kontrolliert reduziert.

Umfangreiche Entwicklungs-Tools unterstützen die Arbeit mit den ST6-Prozessoren. Während Tabelle 2 die zur Verfügung stehenden Software-Tools zeigt, sind die aus Starterkits bestehenden Hardware-Tools in Tabelle 3 aufgelistet.

Den CPU-Kern des 8Bit-Mikrocontrollers zeigt das Blockdiagramm in Abbildung 1, dessen einzelne Funktionsblöcke wir nachfolgend näher betrachten.

Multifunktionale I/O

Die 9 bis maximal 36 Ein/Ausgabe-Pins (je nach Prozessortyp) können softwaremäßig in verschiedenen Modes betrieben werden. Hierbei sind prozessortypabhängig 3 bis max. 12 Ausgänge als 20mA-Hochstrom-Treiberausgänge zum direkten Treiben von z. B. LEDs oder Triacs zu verwenden.

Weiter können die I/Os als Pull-up, analog gemultiplext und dem automatischen Umschalten zwischen I/O und Peripherie-Funktionen programmiert werden.

Durch Umschaltung von Data, Datadirection und der Pull-up-Register ist es möglich, die für die Applikation nötige Konfiguration zu wählen.

Alle Inputs sind über Software zum internen Interruptsystem zu führen, wobei die Interrupts auf fallende und steigende Flanken sowie auf aktiv low reagieren können.

Multifunktionale Timer/Counter

Timerblöcke mit 1 x 8Bit, 2 x 8 Bit, 1 x 16 Bit oder auch 1 x 8 Bit mit Auto-Reload sind in der ST6-Familie enthalten. Die Timer können Prescaler, externe Clock Input, eine Input Capture und Output Compare-Einheit enthalten. Zusätzlich zu Timer und Echtzeit-Clock-Aufgaben ist es möglich, die Timer zum Generieren und Analysieren von Frequenzen, z. B. PWMs, zu benutzen. Der hierzu benötigte Takt wird über einen externen Pin oder den durch 12 geteilten Systemtakt zugeführt.

Bei einem Systemtakt von z. B. 4 MHz heißt das 333,33 kHz, also 3 µs oder bei 8 MHz 1,5 µs, wobei das 8 Bit breite Zählregister einen Zählbereich von 0 bis 255 zuläßt. Durch einen 7Bit-Vorteiler kann zusätzlich durch 1, 2, 4, 8, 16, 32, 64 oder 128 geteilt werden. Der Zähler selbst zählt immer von einem vorgegebenen Wert auf 0. Die Ausgabe geschieht dann über Setzen eines Zustands-Bits, eines Interrupts an die CPU oder über einen Timer-Pin.

Digitaler Watchdog

Beim digitalen Watchdog handelt es sich um einen digitalen 6Bit-ladbaren Abwärtszähler, der mit einem Eingangs-Taktvorteiler versehen ist. Mit diesem Watchdog ist es möglich, beim Zählerstand 0 einen System-Reset zu erzeugen. Diese Eigenschaft wird genutzt, wenn ein Software-Fehler aufgetreten ist, d. h. das System wird durch einen Neustart in einen definierten Zustand versetzt.

Analog/Digital-Wandler (ADC)

Der 8Bit-Analog/Digital-Wandler der ST6-Familie arbeitet nach dem Verfahren der sukzessiven Approximation. Die analogen Inputsignale müssen dabei im Versorgungsspannungsbereich liegen, die gleichzeitig als analoge Referenz benutzt wird. Aus der ST6-Familie sind Bausteine mit 4 bis 24 analogen Eingängen lieferbar. Die Wandlungszeit beträgt bei einem mit 8 MHz getaketen Mikrocontroller 70 µs, wobei der Takt aus dem durch 12 geteilten Systemtakt gewonnen wird. Der AD-Wandler besitzt keine Sample- und Hold-Stufe und wird am Eingang gemultiplext.

Serielles Peripherie Interface (SPI)

Um mit Baugruppen, die sich in der Peripherie des ST6-Controllers befinden, zu kommunizieren, besitzen einige Bausteine zusätzlich eine serielle Schnittstelle (SPI). Hierbei handelt es sich um ein synchrones serielles Interface mit programmierbaren Übertragungs-Modes, die im Master- oder Slave- Mode betrieben werden können. Im wesentlichen besteht das

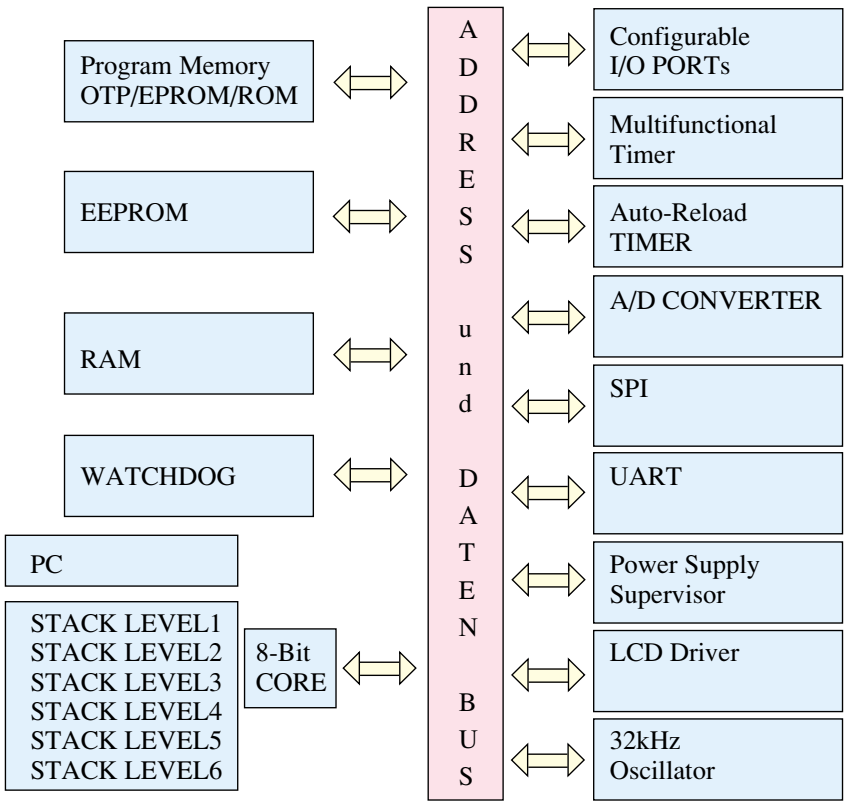


Bild 1: CPU- Kern als Blockdiagramm

Interface aus einem Schieberegister, aus dem die Daten übertragen oder in das die Daten beim Empfang geschrieben werden.

Universaler asynchroner Receiver/ Transmitter UART

Auf einigen Varianten aus der ST6-Familie befindet sich ein UART, der bei 8 MHz eine Baudrate von 38.400 Baud zuläßt. Der UART arbeitet im Half-Duplex-Verfahren mit 11 Bit, davon 1 Start-Bit, 9 Daten-Bits und 1 Stopp-Bit. Die Sendedaten werden direkt gesendet, während die Empfangsdaten in einem Empfangsregister zwischengespeichert werden. Daten senden hat Priorität vor dem Empfangen.

LCD-Treiber

Der bei einigen Prozessorvarianten auf dem Chip integrierte LCD-Treiber besitzt neben dem Multiplexer für die commonplates auch ein internes LCD-RAM, um die Anzeigemuster zu speichern, sowie die von der CPU unabhängige Spannungsversorgung für das LCD. Der benötigte 32kHz-Takt wird vom internen CPU-Takt abgeleitet, so daß kein eigener Oszillator erforderlich ist. Alternativ kann der Takt auch extern zugeführt werden, was wiederum im Stopp- oder Standby-Mode wichtig ist.

EEPROM

In der ST6-Familie gibt es Bausteine mit 64 oder 128 Byte EEPROM, die entweder im Byte- oder Parallel-Mode programmiert werden. Byte-Mode heißt hierbei 8 Bit pro Zeile und Parallel-Mode 8 Bit mal 8 Zeilen. Der Parallel-Mode wird benutzt, um Zeit und Energie zu sparen.

ST6-Realizer Software und Arbeitsweise

Der ST6-Realizer benötigt zum Starten MS-Windows 3.x oder MS-Windows 95. Der Rechner sollte mindestens mit einem 80386-Prozessor, 4MB-RAM, 14MB-Festplattenplatz, einer Maus und einer VGA-Grafikkarte ausgestattet sein.

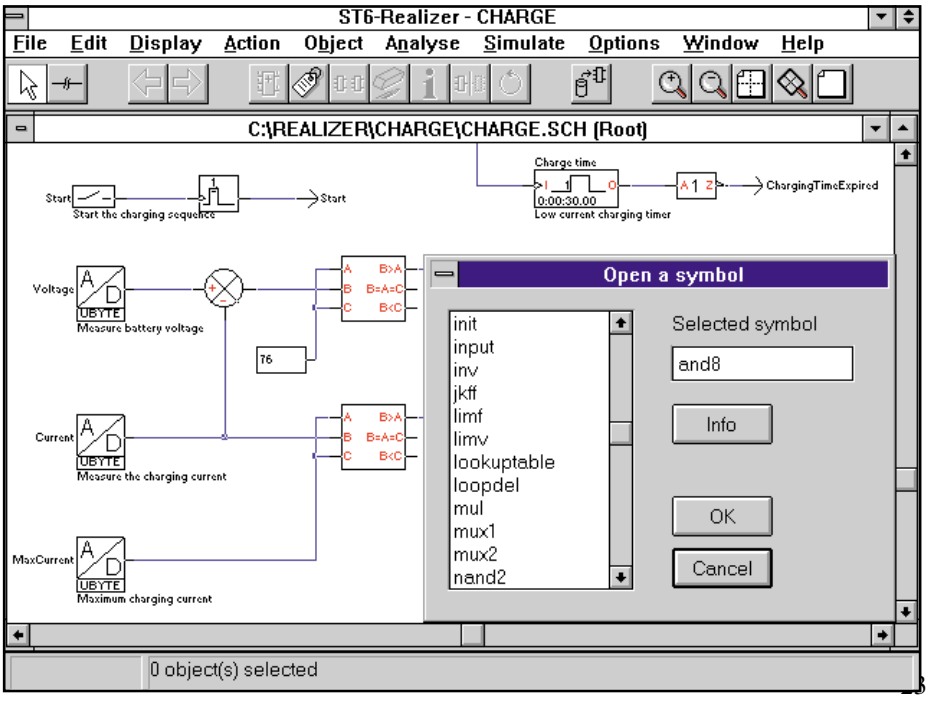
Der ST6-Realizer ist ein CASE-Tool, daß qualitativ hochwertige Designs mit der ST6-Mikrocontroller-Familie ermöglicht.

Bei Benutzung des ST6-Realizers wird die Applikation durch Zeichnen und Verdrahten von Symbolen erstellt. Jedes Symbol entspricht einem Prozeß wie z. B. dem Addieren von zwei Werten, welche dann zu einem Assembler-Code-Macro zusammengefügt werden. DieVerdrahtung (Verbindung) entspricht dem Datenfluß, der mit Variablen und Konstanten verbunden ist. Jede Applikation wird in einer Hauptzeichnung realisiert. Die auf dieser Zeichnung plazierten Symbole können auch aus zusammengesetzten Schaltungssymbolen bestehen, also aus Subzeichnungssymbolen, die dann zu einem Symbol zusammengefaßt werden. Das spart Platz auf der Hauptzeichnung und vereinfacht die Lesbarkeit.

Der ST6-Realizer besteht aus drei Hauptkomponenten:

- Dem Realizer zum Erstellen der Zeichnung, aus der dann am Ende der Assembler-Code erzeugt wird.
- Dem Analyzer, der aus der Zeichnung eine Netzliste und Cross-Referenzen erstellt. Diese werden benutzt, um aus der fehlerfreien, noch nicht compilierten Assemblercode(.asm)-Datei, eine compilierte binäre ST62-Datei zu erstellen. Des weiteren wird eine Report-Datei, welche Informationen über die Variablen, I/O Pins und den benutzten Speicher beinhaltet, erstellt.
- Der Simulator wird benutzt, um das Verhalten der Schaltung, durch Anlegen und Anzeigen von Eingangssignalen zu testen. Die dann von der Applikation generierten Ausgangssignale können mit dem Simulator zur Anzeige gebracht werden.

Im zweiten Teil des Artikels erfolgt die Realisierung eines Software-Projektes. ELV



Erstellen der Applikation durch Zeichnen und Verdrahten von Symbolen