

THE BASIC TIGER™



88,-

The Superlative Control Computer

- **128 KB static RAM**
128 KB FLASH memory
(program and data memory)
- **Lightning fast: performance**
up to 100,000 instructions / s
- **Multitasking:**
up to 32 BASIC tasks
simultaneously
- **38 I/Os integrated in the**
module (analog, digital,
serial, universal)
- **Easy I/O expansion**
- **Up to 1920 digital I/Os with**
standard 74-series ICs
- **Simple connection of external**
devices, LC-displays,
keyboards, printers etc.
- **Modules with up to 16 MB**

Never before has so much BASIC power been packed into such a small, inexpensive control computer package. But that's not all. BASIC Tiger™ applications are also:

- ready faster,
- more powerful, and
- more cost-effective

The system has ample reserve power, giving you the confidence of knowing that you can upgrade and expand your applications as the need arises. To install a new software version you simply download it to the BASIC Tiger™, where it is stored permanently in the unit's non-volatile memory. No EPROM programming or chip replacement required!

Despite its tiny dimensions the BASIC Tiger™ is a real number-crunching powerhouse, giving you user-friendly handling, straightforward programming:

- The BASIC CPU executes Tiger BASIC™ programs at impressive speed, many times faster than the standard in this controller class.
- The BASIC Tiger™ is highly eco-

nomical solution, both for small tasks for large-scale projects with many thousands of lines of program code.

■ Multitasking.

The BASIC Tiger™ can execute up to 32 tasks simultaneously, performance that is probably unique in this class.

Multitasking offers many advantages,

even in smaller applications. Dynamic prioritization enables you to allocate CPU capacity precisely to the individual tasks, as needed. When a task's requirements in-

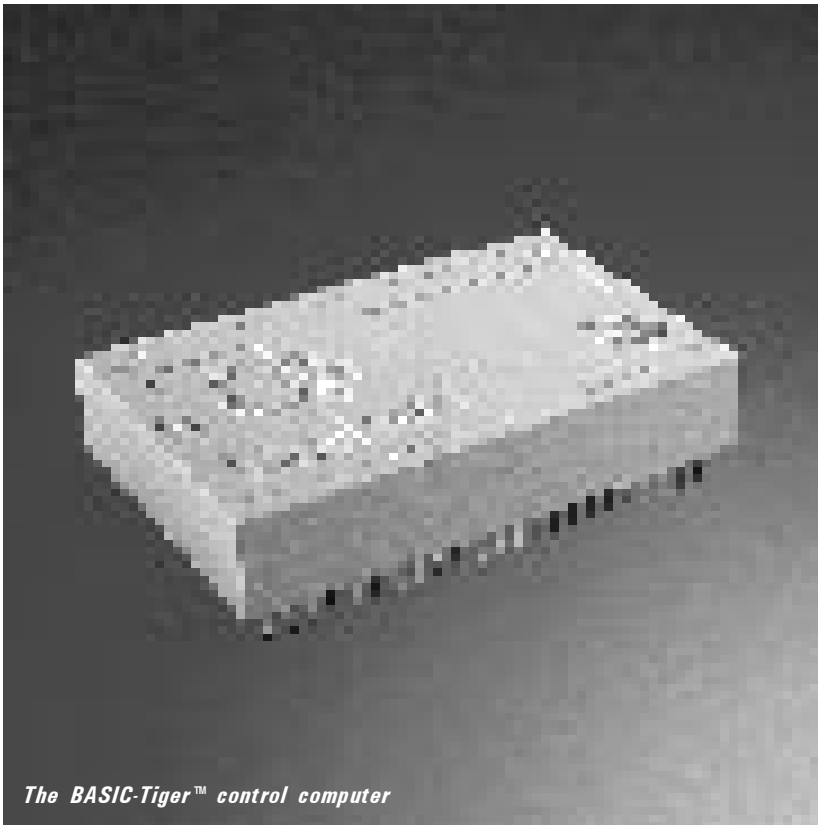
crease or decrease you simply change its CPU resources allocation accordingly. This can be done either externally or directly by the task itself. The BASIC Tiger™ multitasking system is so simple to use that you can have your first sample programs up and running in a matter of minutes.

■ Powerful I/O-functions.

Communication with the 'real' world is handled via the I/O channels. This another very powerful BASIC Tiger™ feature – even the smallest BASIC Tiger™ module has 38 software-configurable I/O lines. Connection and use of LC-displays, keyboards, I/O expansions



BASIC TIGER



The BASIC-Tiger™ control computer

etc. couldn't be easier. In your BASIC programs these peripherals are simply addressed as logical devices.

■ **Everything you need to get started right away.**

The controller is supported by an intuitive program development system that runs under Windows™ 3.1 and Windows 95™. Ergonomical program design helps to optimize the speed and efficiency of the repetitive edit/compile/download/debug cycle. For example, you can set the compiler to branch directly to the source code error in the editor when a compile error is encountered. Then you simply correct the error and start the compile sequence again. Multiple windows are supported to maintain a clear overview of all aspects of your program development project, e.g. for source code, variables, system information, editor window etc.

The powerful source-level debugger is also an invaluable program development aid, both when you are learning the ropes and for isolating complex errors in big programming projects.

■ **Fast updates and modifications.**

You can change your BASIC programs and transfer the new version to the BASIC Tiger™ quickly and easily whenever necessary. Non-volatile flash memory eliminates time-consuming erasing, programming and replacement of EPROMs.

When the new version has been compiled you simply transfer it to the BASIC Tiger™ and it's ready to run immediately.

Short development cycles and fast programs make the BASIC Tiger™ the ideal choice for countless applications. And it has enough reserve power to ensure that you can add new functions later to respond to new situations and changing user requirements.

A few examples of BASIC-Tiger™ applications:

- Machine control systems
- Intelligent measuring instruments
- Monitoring applications
- Work time logging
- Fire protection
- Radio equipment control
- Alarm systems
- Heating control (energy conservation)
- Lighting control (energy conservation)
- Process data logging
- Automated modem links to central host computers
- Weather stations
- Animal feeding systems
- Sound and speech recording
- Programmable announcement systems
- Universal protocol converter
- data encryption and decryption

- Postscript generator
- Communications tester
- Data logging
- Hand-held RS-232 terminal
- Print server
- Building automation systems
- 24h ECG monitoring unit
- Automated long-term test system
- Camera control for photo studio automation
- IC and component tester
- High-precision temperature control
- PID controller
- Electronic 4-channel data recorder
- Intercom system with mailbox
- In-house paging systems
- Large-scale displays
- special functions for PBX systems

BASIC Tiger™ programs execute so fast that it's frequently possible to dispense with time-consuming C and assembler programming entirely. The programs are also very readable, which makes maintenance easier and faster. Both real pros and occasional programmers use Tiger BASIC™ to create powerful applications.

Both the development package and the BASIC Tiger™ single-board computer are attractively priced to help keep your costs low.

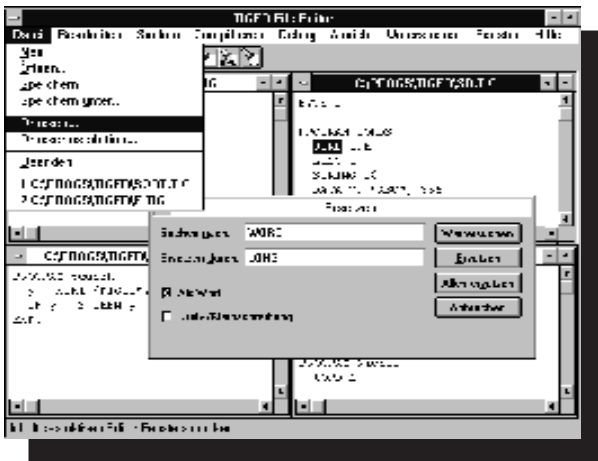
Program development with BASIC-Tiger™

The BASIC Tiger™ development system contains all the software and hardware components you need to start developing your BASIC Tiger™ computer programs right away:

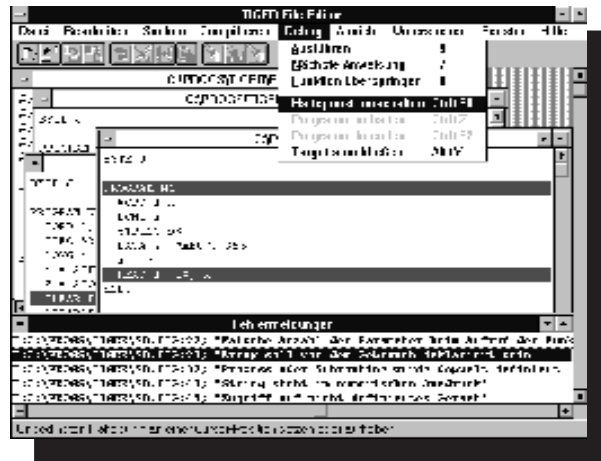
- The BASIC-Tiger™ program development environment for WINDOWS™ 3.1/ WINDOWS™ 95 with editor, compiler, downloader and debugger
- The BASIC-Tiger™ hardware toolkit for performing fast tests, including tests on your own hardware. With LC-display (4 x 20), alpha-keyboard, 64 expansion lines, relays, power outputs, speaker, ...
- Power supply, cables, adapters, manual, reference card and accessories

You can develop BASIC Tiger™ programs on a standard PC running Windows™ 3.1 or Windows 95™.

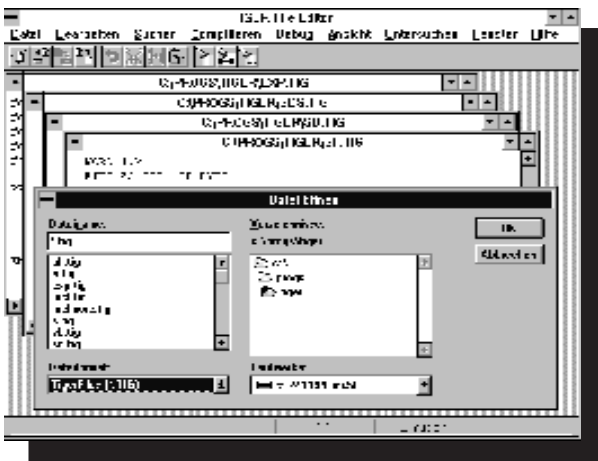
The BASIC-Tiger™ program development environment includes everything you need for program development and testing:



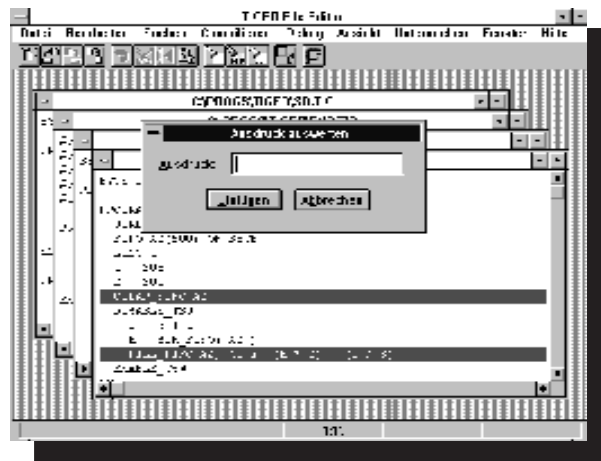
The editor supports multiple windows that can be arranged on the screen as required for a clear overview of all aspects of your project.



Any errors found during compilation are displayed in the Errors window. When you select an error message the editor cursor automatically moves to the line where the error originated.



The main editor functions can be selected with both menu options and the icons in the toolbar.



When you debug your programs with the BREAKPOINTS, SINGLE-STEP or TRACE functions the values of the selected variables are always displayed.

□ **The Editor**

This is the tool for writing and editing your program code, for applications consisting of anything from just a few lines to many thousands of lines in length. To speed up development even more you can import the selection of ready-to-run program modules supplied with the system directly into your source code in the editor. You can also import ASCII texts into the editor from other sources and use them as-is or edit them as necessary.

□ **The Compiler**

The compiler checks the syntax of both finished and unfinished programs. You can set the compiler to activate the editor automatically when it finds an error; it then places the cursor in the

corresponding line so that you can correct the mistake immediately. Alternatively, you can choose to have all errors displayed in a list in a separate window after compilation. If no errors are found the compiler processes the entire program and generates the executable code and/or the documentation output.

□ **The Downloader**

This module transfers the executable program code to the BASIC Tiger™ single-board computer connected to the PC. In addition to downloading directly after compiling you can also download compiled programs stored in disk files. The BASIC Tiger™ is then immediately ready to execute the program or switch to debug mode.

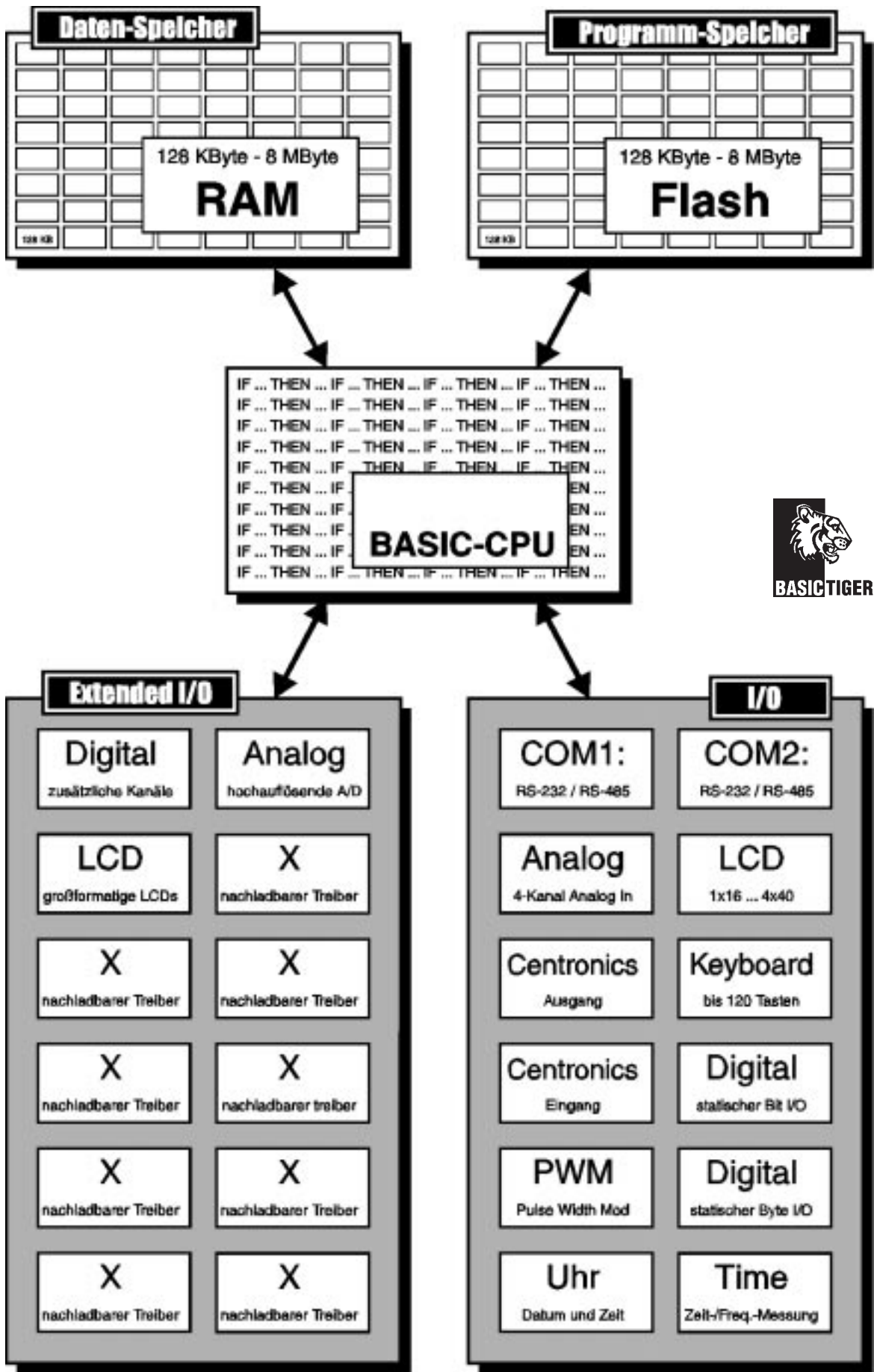
□ **The Source-Level Debugger**

The BASIC Tiger™ debugger is a powerful tool for testing and debugging your BASIC programs. It includes all the features you could possibly want for locating errors and analyzing runtime performance:

- Display variables
- Edit variables
- Task information
- Program execution data
- Breakpoints
- Single-step mode

You can activate multiple windows for simultaneously viewing of source code, the contents of relevant variables and system informations.

When development is complete you



BASIC-Tiger® System-Übersicht

can then protect your programs with the 'No Debug' attribute, which prevents the BASIC Tiger™ from outputting any debug information. You can also use other protection functions, including signatures and copyright entries.

The block diagram on page 4 shows the main components of the BASIC-Tiger™. Even the smallest version of the BASIC-Tiger™, the ANN-1/1, is a powerful, full-functioned control computer with:

- 128 KB RAM
- 128 KB FLASH memory
- Tiger-BASIC™ CPU
- 4 analog channels (8 / 10 / 12 bit), up to 50.000 samples / sec.
- Plus 34 other universal I/Os (including serial I/O, PWM-outputs, timer, expansion bus and digital I/Os)

Larger modules can contain additional resources to match the needs of your applications:

- Up to 8 MB RAM
- Up to 8 MB program memory (FLASH memory)
- Real-time clock / calendar (non-volatile)
- Up to 128 additional I/Os
- XSpeed: double speed execution
- Extended I/O-devices

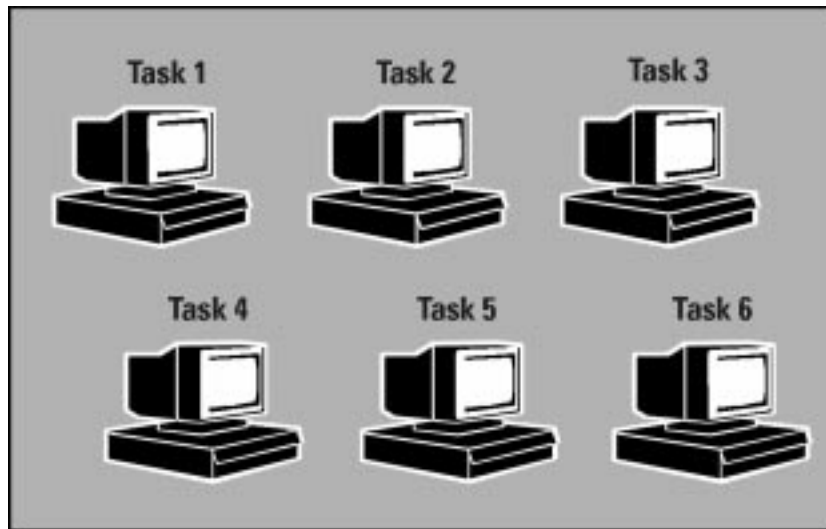
Multitasking:

You will be amazed at how easy multitasking is with the BASIC Tiger™, and delighted by the great advantages of parallel tasks, even in small, simple applications. Of course, the benefits are even greater in large projects, which you can split up into individual sub-jobs assigned to different tasks. This makes your programs shorter, more clearly structured and easier to handle. BASIC Tiger™ multitasking gives you better results faster, making your applications:

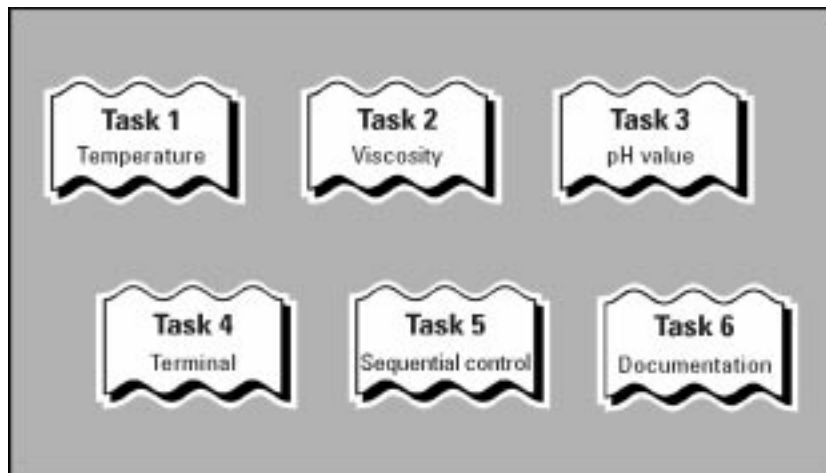
- Clearer
- Faster
- Easier to program
- More structured
- Cheaper to develop
- Easier to maintain

The multitasking capability of the BASIC-Tiger™ is a powerful function that greatly simplifies the BASIC programmer's work.

To begin with, it allows you to handle individual tasks as autonomous programs, which can perform different and



You can use the multitasking system as if it was a group of separate computer systems. Each task is completely independent and does not have to have any logical relationship to the other tasks (see Example 1).



Here in Example 2 a complex lab automation application consisting of a series of different tasks is running on a single BASIC-Tiger™ computer. All the tasks are part of the same automation process and they are all executed simultaneously.

entirely independent jobs within the same application (Example 1). In addition to this, you can also use multitasking as a structured programming tool - this enables you to split up large programming projects into several individual units that can be edited separately (Example 2).

Example 1:

Here several completely separate programs are executed simultaneously by a single BASIC Tiger™ computer, which thus replacing several computers:

Task 0: Power-Up sequence, initialization of the system and the vari-

ous tasks.

Task 1: A strategy game programmed in BASIC, using a 14x2 numeric display (7-segment) and 12 keys. You can start, play or quit the game at any time.

Task 2: Heating control system for a private home, supporting separate temperature profiles for each room for specific times of day/weekdays.

Task 3: Lighting control system. Activation of the house lights is controlled on the basis of inputs from

movement sensors, ambient light sensors, light switches and the time of day/weekday. Power conservation program.

Task 4: Home security. The program registers and analyzes input from windows, doors, movement sensors, body noise etc., with activation and deactivation functions.

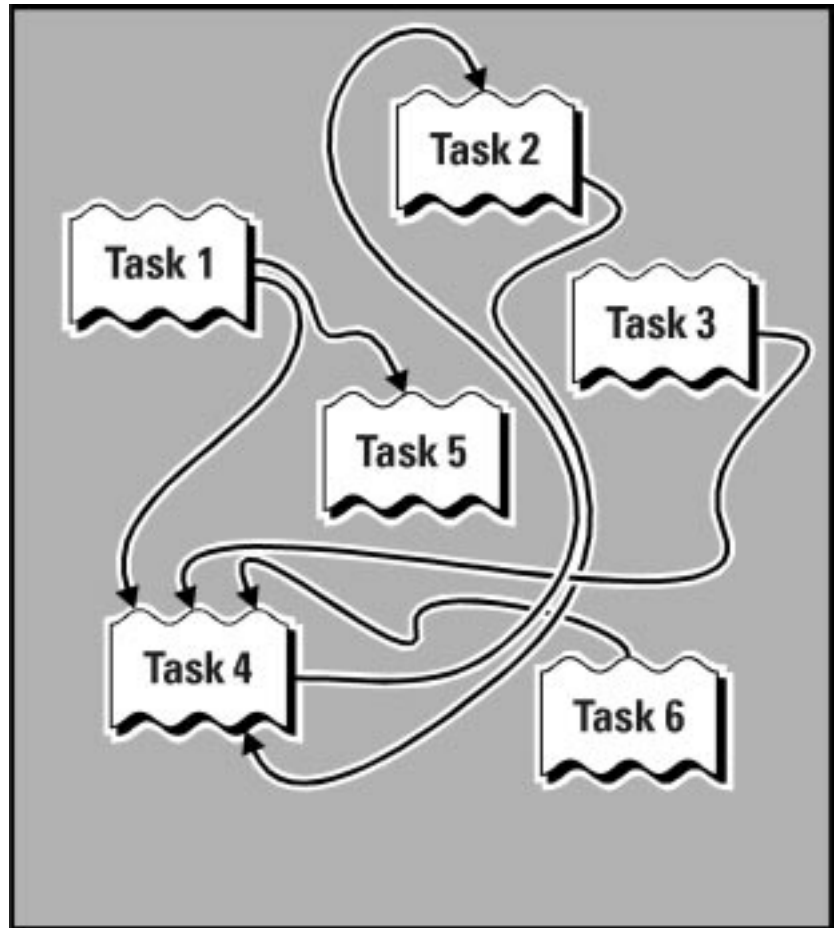
Task 5: A compact ASCII terminal is connected to the BASIC Tiger™ via an RS-232 interface. From the terminal the user can:

- Control the heating system
- Control the house's lighting
- Control the alarm system
- Enter personal appointments with alarms as reminders
- Generate printouts of logs/reports

Task 6: Printer output. The system can print out a variety of logs and listings that are generated by other tasks, either automatically or in response to user requests.

Task 7: A powerful time switch program for 12 separate circuits.

Task 8: ... other tasks (max 32) ...



Each task can start, stop, interrupt, suspend, resume or change the priority of any other task. Tasks can also control their own execution in the same way.

Example 2:

A lab automation process. The various different sub-tasks that make up the process are assigned to tasks that are executed simultaneously by the BASIC Tiger™ control computer:

Task 0: Power-up sequence, initialization of the system and the individual tasks.

Task 1: High-precision temperature control

Task 2: Sample-taking and viscosity measurement

Task 3: pH value measurement

Task 4: User terminal for monitoring and controlling the process

Task 5: Sequential control of the entire process

Task 6: Documentation

Task 7: Network connection with the lab computer

In this example a single BASIC Tiger™ computer controls an entire complex process consisting of a series of individual sub-tasks. Each sub-task is handled separately, and the individual tasks are controlled by control tasks.

All the tasks execute simultaneously and none of the tasks impedes the execution of the other tasks. There are no delays (e.g. the system is not held up if the printer isn't ready when a report printout is requested). Independent jobs are executed independently. Tasks can start and stop other tasks, pass parameters to them, increase or reduce their priority and exchange semaphore messages.

The key benefits of BASIC Tiger™ multitasking:

- Simpler, structured programming: In traditional programs, loops with countless conditional statements, a corresponding number of program branches and complex status handling resulting in convoluted code that is difficult to follow. In contrast, multitasking gives you:
 - Clearly-structured programs
 - Faster program development
 - Easier maintenance

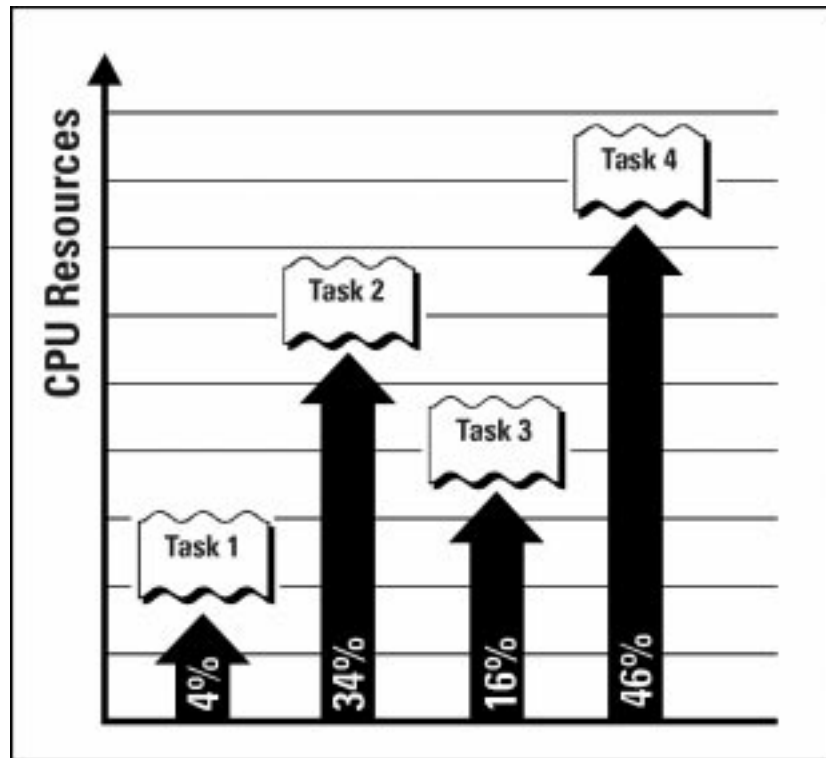
- Lower debugging overheads
- More reliable programs
- Programs that are easier to follow
- Lower costs

- Your programs are more reliable:
 - An endless loop doesn't bring your entire application to a screeching halt, it only stops the affected task. Instead of total program failure you just have a single, isolated problem.
 - Additional security: If a task contains highly-critical procedures you can program a separate 'supervisor task' whose only job is to check that the important operations are performed correctly. If problems are registered this task can then activate an alarm or start an emergency routine.
- Additional options:
 - Self-optimizing programs with learning capability: When one task is handling a control loop (e.g. PID

or similar) a second task can act as a supervisor, checking the quality of the control performance. From time to time the supervisor task can change specific control parameters, check the results and continuously adapt to the current situation in the control circuit.

□ Multitasking makes sense:

Most applications do not consist of a single huge job, but rather of a whole series of sub-jobs that can be executed and handled independently. Complex multiway selection loops are a standard feature of conventional programs. This results in unclear program structure that costs a lot of time and space, making the programmer's job more difficult instead of easier. Multitasking makes things simpler, giving you clear, legible structures and fast program execution. Additional jobs can simply be programmed in their own tasks, as needed.



Example of the allocation of the BASIC-Tigers™ CPU resources at a specific point in time: Task 1 is currently only getting 4% of the CPU resources, while Task 2 is using 34%, or around 8 times as much.

Multitasking programming

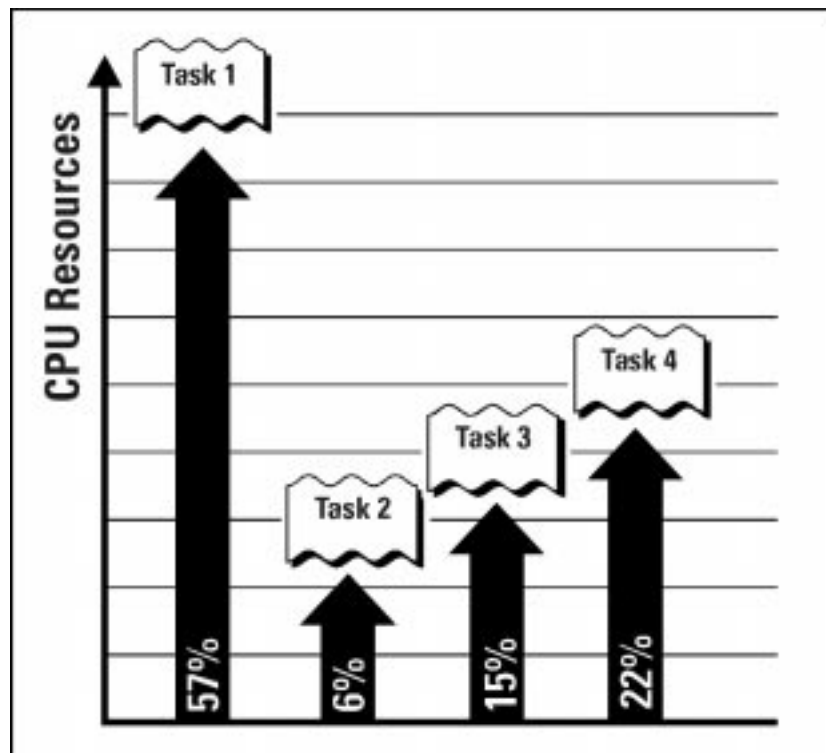
Each task normally has a clearly-defined job, which is often related to other jobs performed by the application. Tiger® BASIC provides full support for these relationships; it is very easy for tasks to communicate with other tasks and influence their performance. Any task can

- start
- stop and
- change the priority of

any other task. Tasks can also:

- communicate with semaphore messages and
- use common variables.

In addition to this, tasks can also share the same subroutines without getting in each other's way. Each task behaves like a completely independent program with its own variables, its own runtime error handling routines and its own CPU resources.



At another point the application could require a totally different distribution of the CPU resources. Here, for example, the lion's share (57%) is allocated to Task 1. The resources allocation can be changed in response to special requirements such as the need to handle large volumes of data or to provide faster response times.

When the increased load on Task 1 is over its priority can be reduced again and the freed resources assigned to the other tasks.

Subroutines

Subroutines can pass up to 16 input and output parameters and can use both local and global variables. Recursive subroutine calls are supported. All subroutines can be accessed by all tasks and all other subroutines. Fundamentally, there are no restrictions here at all. Every task can call any subroutine at any time. The

only natural limit for the nesting depth and number of parameters is the size of the stack, which is also adjustable.

There are no limits on the size of tasks and subroutines. Provided there is free program memory you can go on adding subroutines indefinitely. The local variables of subroutines are strictly local; they cannot be read or changed by any other programs or tasks.

Variables

Tiger BASIC™ supports an extensive range of variable types for handling and manipulating large data structures and individual bits and bytes. Supported data types include:

BYTE, WORD, LONG, REAL, STRING and ARRAY. Support is included for arrays of different data types with up to 8 dimensions and virtually no limit on the number of individual elements contained.

Multitasking / Multi-user

The powerful multitasking capabilities of the BASIC Tiger™ also make it very easy to implement multitasking/multi-user applications.

In the lab application in Example 2 you could program a second user terminal for User No. 2 (Terminal 1 = User 1).

Both users could then access the system simultaneously. For example, User 1 could prepare the next test while User 2 calibrates the viscosity meter with computer support.

This type of multi-user application brings benefits when systems need to be operated from different locations, and when different users need to perform different tasks simultaneously.

Development Systems

The comprehensive BASIC Tiger™ program development systems give you everything you need to get started quickly and develop powerful applications in record time. The packages include a large selection of sample programs and applications that you can download to the target system and run directly:

The BASIC-Tiger™ Standard package:

- BASIC-Tiger™ software for Windows™ 3.1 / Windows95™: Development environment with Tiger BASIC™ compiler, editor, source-level debugger and downloader.
- BASIC-Tiger™ Prototyping-Board
- Manual, connection cable set, power supply unit, accessories.



..... \$ 299,-

The BASIC-Tiger™ Professional package:

The Professional package contains all items of the Standard package plus:

- Hardware tool kit for program development and immediate tests. Large selection of ready-to-run programming examples and applications. Plug connections (no soldering), PC interface, LED status indicators, supertwist 4-line, 20 character LCD-display, alphanumeric keyboard, 64 expansion outputs, power relays, power transistors, speaker, analog inputs, expansion connectors etc.
- BASIC-Tiger™ control computer type ANN-1/1 in 46-pin DIP housing. Dimensions approx. 41 x 63 x 12 mm (1.6 x 2.5 x 0.5"), 128 KB low-power static RAM, 128 KB FLASH memory, PC-channel.



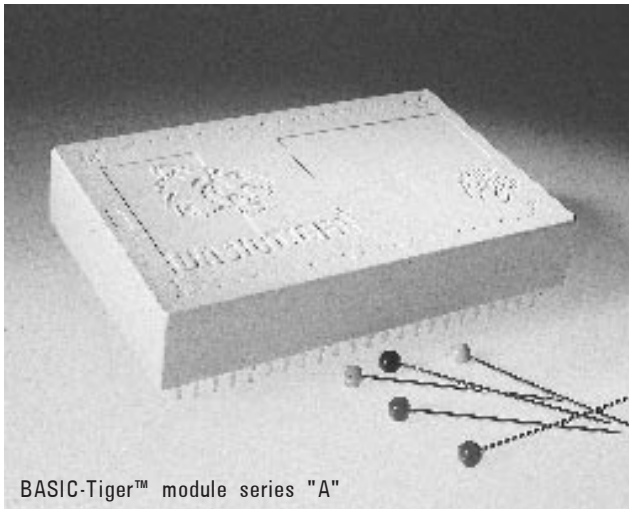
..... \$ 795,-

The TINY-Tiger™ Development Kit:

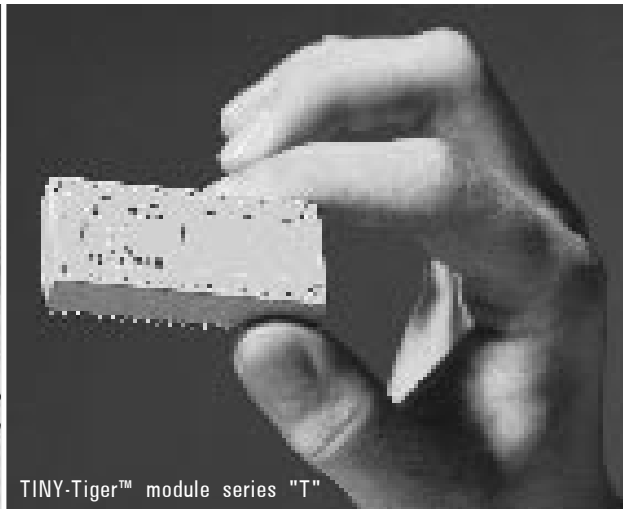
- TINY-Tiger™ software development system for Windows95™: Development environment containing Tiger BASIC™ compiler, editor, source-level debugger and downloader.
- TINY-Tiger™ Prototyping-Board
- Manual, connection cable set, power supply unit, accessories.



..... \$ 99,-



BASIC-Tiger™ module series "A"



TINY-Tiger™ module series "T"

BASIC-Tiger™ module series "A"

module type	Low-power static RAM				FLASH memory		Clock w. date	V24 channels	Industrial version	Price US\$			
	128K	256K	512K	1 MB	128K	512K				≥	>0	>100	>1000
ANN-1/1	●				●		–	–	–	88,-	83,-	78,-	73,-
ACN-1/1	●				●		●	–	–	95,-	90,-	85,-	79,-
AXN-1/1	●				●		●	●	–	102,-	96,-	91,-	85,-
ACN-2/4		●				●	●	–	–	108,-	102,-	96,-	90,-
AXN-2/4		●				●	●	●	–	115,-	109,-	102,-	95,-
ACI-4/4			●			●	●	–	●	199,-	188,-	177,-	165,-
AXI-4/4			●			●	●	●	●	206,-	195,-	183,-	171,-
ACI-8/4				●		●	●	–	●	229,-	216,-	204,-	190,-
AXI-8/4				●		●	●	●	●	236,-	223,-	210,-	196,-

TINY-Tiger™ module series "T"

module type	Low-power static RAM				FLASH memory		Clock w. date	V24 channels	Industrial Version	Preis US\$			
	32K	128K	256K	512K	128K	512K				≥	>0	>100	>1000
TNN-R/1	●				●		–	–	–	59,-	56,-	53,-	49,-
TCN-1/1		●			●		●	–	–	79,-	75,-	70,-	65,-

Other modules are available in larger quantities on request.

BASIC-Tiger™ and TINY-Tiger™ are Trademarks of Wilke Technology, Germany.
 WINDOWS™ and WINDOWS 95™ are Trademarks of Microsoft Corp. USA.

All prices in US\$ without taxes or shipping charges ex works Aachen, Germany. Prices are valid starting October 1, 1997. All older price lists cease to be valid. Price list is subject to change, errors excepted.

Specifications:

BASIC-Tiger™ module series "A":

- Dimensions
Size approx. 41 x 63 x 12 mm (1.61 x 2.48 x 0.47"), 46-pin unit, weight approx. 48g
- Operating temperature
Standard: 0°C to +70°C (32°F to 158°F)
Industrial: -20°C to +85°C (-4°F to 185°F)
- Power Supply
4.7V-5.5V / approx. 45-100mA (sleep mode approx. 50-100µA)
- I/O pins
38 universal I/O-lines
- I/O pin driving capacity
1.6mA / 3.5mA at max. 8 pins simultaneously
- Memory capacity
128 KB ... 2 MB static low power RAM
128 KB ... 4 MB FLASH memory
- Reset
Power-on reset in the module and RESET input
- Real-time clock
With calendar and alarm function (alarm active low), battery for clock and RAM backup, connection for external backup battery.
- 2 serial channels
With either TTL levels or on-board V24 drivers (Pin 24-29). Pins 24-29 can also be software configured for use as digital I/O lines
- Impulse durations / frequency measurement
Resolution: 3 ranges (0.4 / 1.6 / 6.4µs)
- A/D inputs
4 channels
Resolution 8, 10, 12 bits (12 bits with oversampling)
A/D converter rate up to 50,000 samples/s
A/D reference voltage (Vref) 3.5-5 V
A/D input voltage 0V-Vref
A/D buffer size 16 bytes - 4 MB
A/D modes single / repeat
- D/A outputs
2 PWM channels
PWM resolution 6, 7, 8 bits
PWM repeat frequency 0.6-39kHz

TINY-Tiger™ module series "T":

- Dimensions
approx. 28 x 60 x 11 mm (1.10 x 2.36 x 0.45"), 44-pin unit, weight approx. 28g
- Operating temperature: 0°C to +70°C (32°F to 158°F)
- I/O pins: 36 universal I/O-lines
- Memory capacity
32 KB or 128 KB static low power RAM, 128 KB FLASH memory (TINY-Tiger™)
- 2 serial channels
With TTL levels (pin 23-28). Pins 23-28 can also be software configured for use as digital I/O lines

all other specifications are identical with those of the BASIC-Tiger™ module series "A".

Tiger-BASIC™:

- Multitasking
Up to 32 BASIC tasks simultaneously
- Program execution speed
Approx. 10,000 - 100,000 BASIC instructions per second. High-density and I/O instructions are executed more slowly, depending on their complexity and the data volume
- BASIC memory capacity
Approx. 2,000 - 5,000 BASIC instructions (128 KB FLASH); approx. 10,000 - 30,000 BASIC instructions (512 KB FLASH); more instructions with larger FLASH. BASIC programs can read, write and clear the FLASH memory directly.

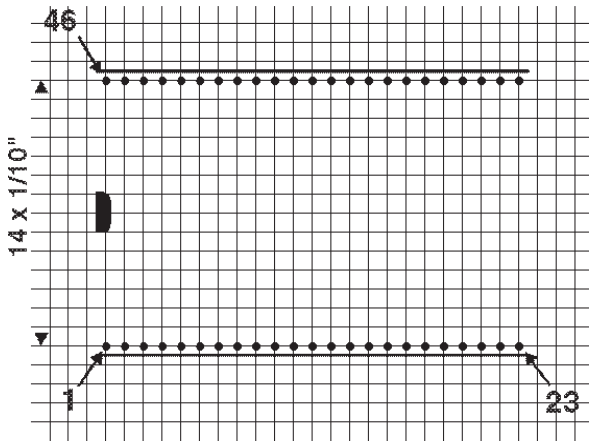
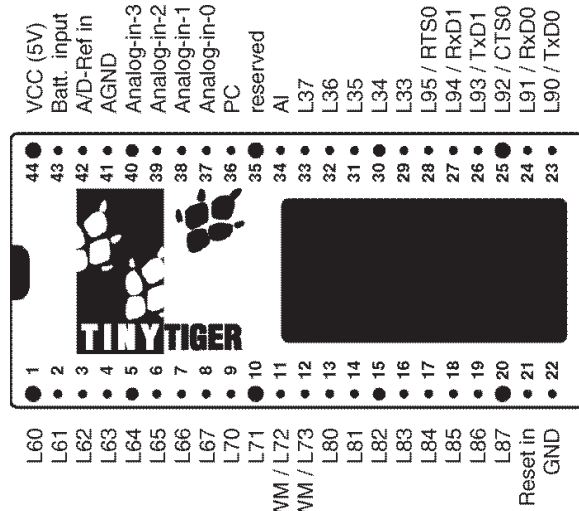
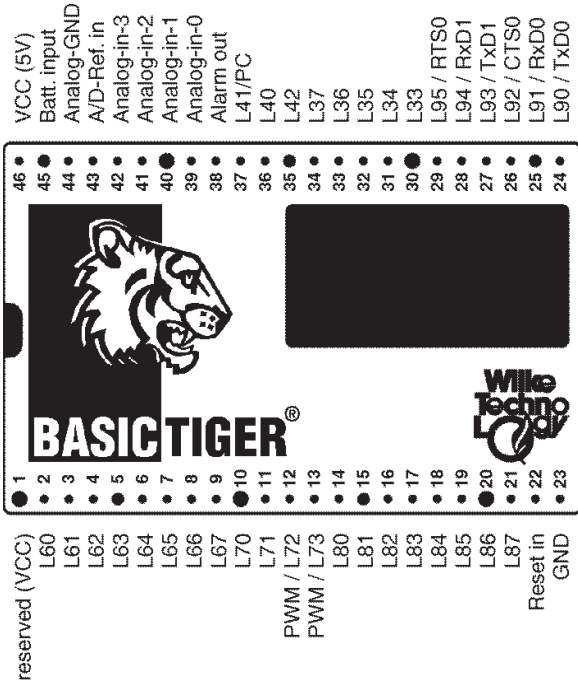
Peripheral Connections:

- Standard I/O pins
The I/O architecture of the BASIC Tiger™ is designed for flexibility and easy expansion. The module has standard I/O pins that can be used in a wide variety of ways. You can also assign other I/O functions to the pins with appropriate instructions in your BASIC program.
- Serial channels
Channel 0: Rx/D, Tx/D, RTS, CTS. Channel 1: Rx/D, Tx/D
- Communication parameters
Baud rates 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 76800, 153600; 7 or 8 data bits; odd or even parity; RS-485 bus.
- Byte I/O expansions
You can add up to 240 additional 8-bit ports (=1920 digital I/O pins) very easily by external connection of standard 74-series ICs ('138, '245, '377). No multiplex signal generation or address decoding needs to be performed by the BASIC program. Programming is very simple. For example, all it takes to output 8 bits to port 78 is the statement **OUT 78, DATA**. The statement for data input is similar.
- A/D conversion
8, 10 and 12 bit analog inputs supporting rates of up to 50,000 samples per second.
- LC displays
Simple 1:1 connection of text displays with the standard HD44780 controller. Fifteen definable special character sets with 8 characters each. Cursor control, ESC sequences, menu selection, graphical characters, icons, moving symbols, moving bars, tachometers etc.
- Keyboard
Up to 128 keys / DIP switches. User-programmable key codes, definable key attributes: debounce; auto-repeat; key-click; Shift, Ctrl and special functions; beep, alarm.
- I²C bus
For the connection of I²C devices.
- Clock
Real-time clock with alarm function. Can switch both devices and the Tiger module itself on and off. With battery input.
- Specific I/O channels
You can implement any other types of I/O channels required with BASIC routines or device drivers.

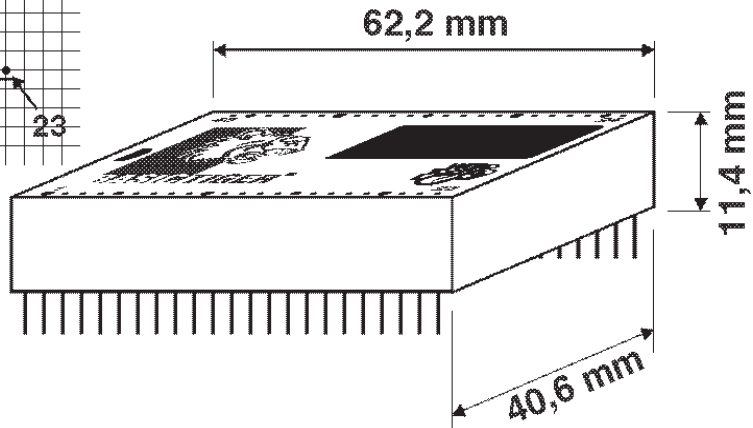
Pin-Layout

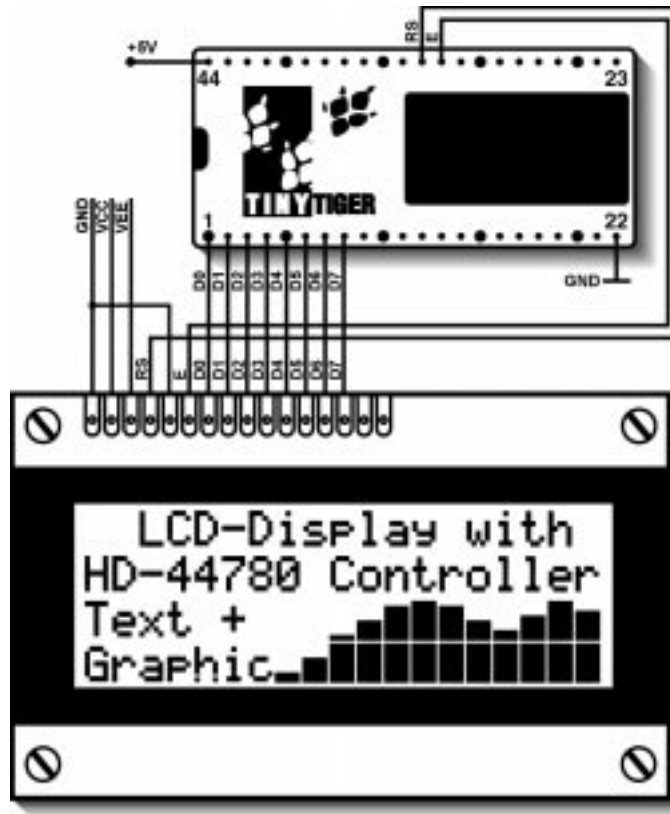
BASIC-Tiger™

TINY-Tiger™

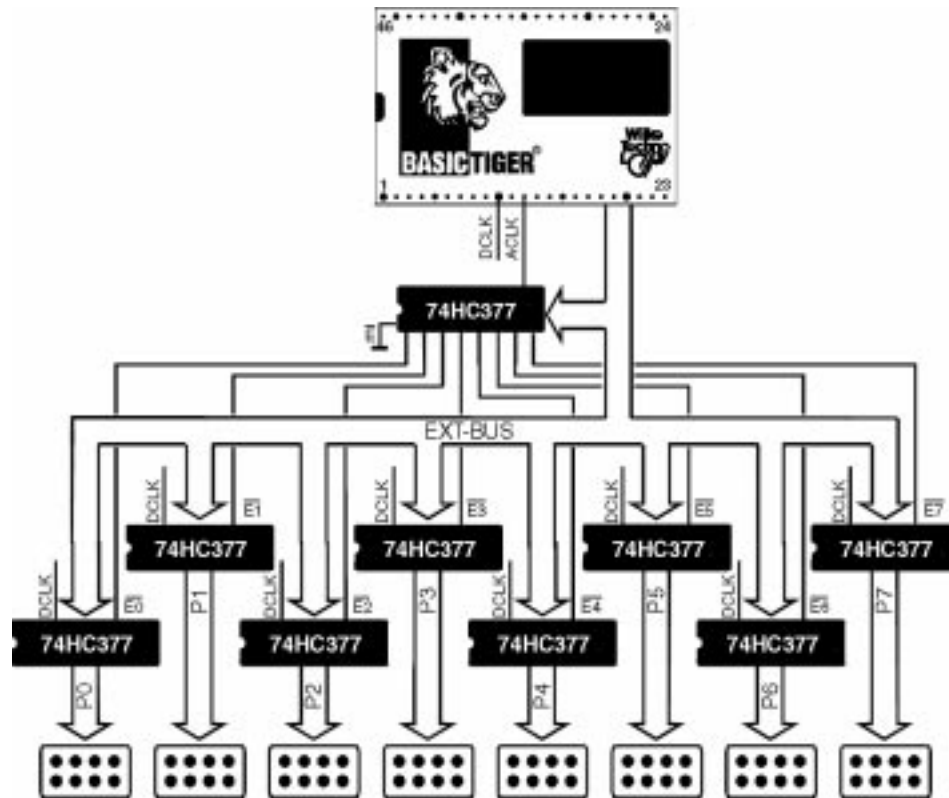


BASIC-Tiger™
Illustration approximately original size





One possible connection option for an LC display



Eight extended output ports with a total of 64 outputs