The P89LPC900

Part 1: the architecture

By M. Stottelaar

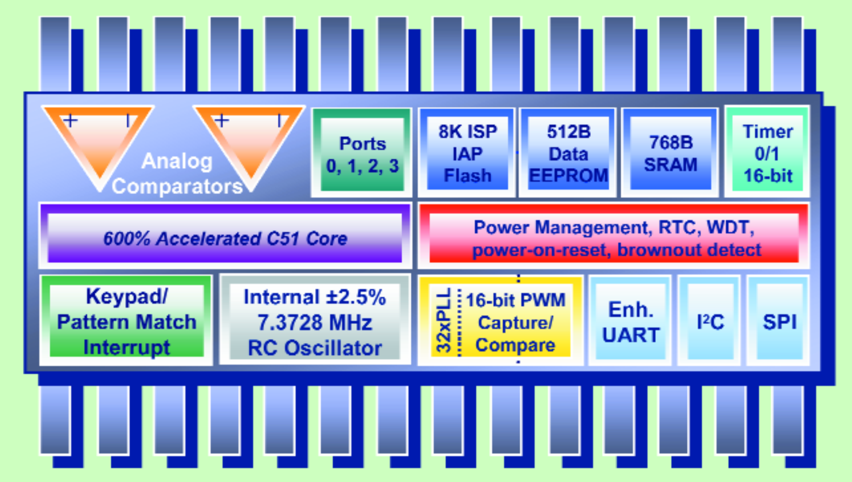
Eurodis Texim

Under the moniker of P89LPC900, Philips have introduced an entirely new family of 80C51 successors that have a number of pins ranging from 8 to 80 and are all based on an (in-system-programmable) flash memory. An important characteristic of the controller is that despite a large number of new features it is still fully compatible with the 80C51.

That the 80C51-based controllers are extremely popular is nothing new, certainly when considering the large number of designs that have appeared in this magazine over the years. The reason may well be the fact that the tools (both hardware and software) that are available for this controller are very affordable and there is an enormous amount of information readily available. In addition, a very active forum provides answers to many questions.

Philips, one of the biggest players in the 80C51 field has, with the part number P89LPC900, recently breathed new life into this architecture and improved the controller family to the point that its lifespan has been extended for many more years.

In this article we will describe the architecture of the P89LPC900. In a follow-up instalment we will publish a simple programmer/I/O-board.


Analog Comparators
Ports 0, 1, 2, 3
8K ISP IAP Flash
512B Data EEPROM
768B SRAM
Timer 0/1 16-bit
600% Accelerated C51 Core
Power Management, RTC, WDT, power-on-reset, brownout detect
Keypad/ Pattern Match Interrupt
Internal ±2.5% 7.3728 MHz RC Oscillator
32xPLL
16-bit PWM Capture/ Compare
Enh. UART
I²C
SPI

The heart of the controller

One of the most significant features of the P89LPC900 is that the core now requires only **two clock cycles per instruction**. 8051 experts will already know that this used to be 12 or 6 cycles until now. But to be fair, it needs to be mentioned that the maximum clock frequency has been reduced to 12 MHz (167 nsec per instruction). The net improvement is still more than a factor of two, compared with previous solutions that used to run at 33 MHz. The improved performance at lower crystal frequencies is beneficial to power consumption and EMI. The P89LPC900 is therefore very appropriate for low-energy designs such as battery-powered devices. On average, the current consumption per MHz is not much lower than before, but you do get a six times improvement in performance. In practice, this means that the crystal frequency can be drastically lowered. The lowest possible current consumption, by the way, is only 1 μA.

The clock

Most of the existing 80C51 controllers have a relatively simple clock distribution circuit. A crystal, resonator or external clock signal is connected directly to the CPU and also to the peripherals. In contrast, the P89LPC900 has a much more elaborate mechanism, which allows the user to determine where the clock source comes from and how it is to be presented to the CPU and peripherals. A standard, and in the context of microcontrollers, very accurate internal RC oscillator (2.5% over the entire temperature range) will in many cases suffice as the clock source. Even a UART will generally operate very well with this kind of accuracy. In the event that the requirements are more stringent, the processor offers the option of mea-

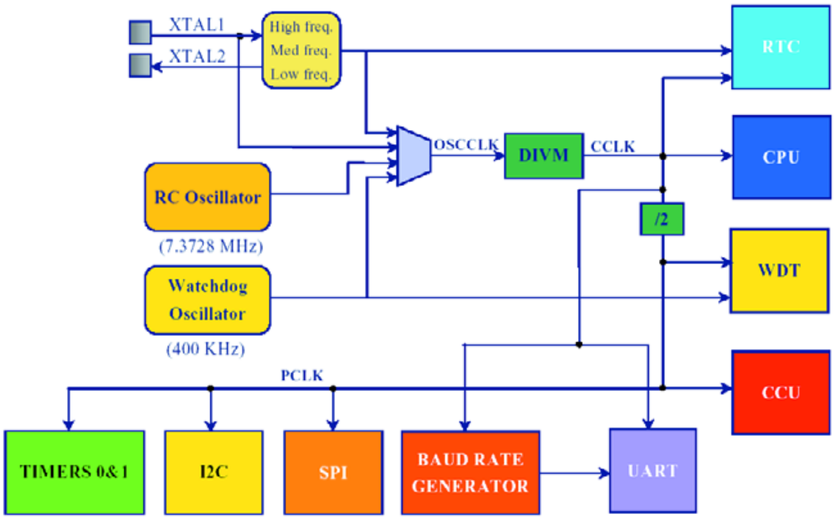


Figure 1. Complete overview of the clock distribution of the P89LPC900.

suring the frequency at the clock output and fine-tune it with the aid of the TRIM register. Note that a nominal frequency of 7.3728 MHz was selected, which is a very appropriate value in order to generate all the known baud rates.

To be able to trade off performance and current consumption, the user can switch 'on the fly' to another frequency by changing the prescaler. Even the separate oscillator for the watchdog can serve as the clock source during moments when performance is not that important, but current consumption plays a more significant role. **Figure 1** gives a complete overview of the clock circuitry.

The memory

The P89LPC900 is a complete embedded controller, which means that no external memory can be addressed (unless this is done via SPI or I^{2}C, of course).

The data memory (RAM) of the P89LPC900 partly consists of a section that can be **directly or indirectly** addressed (256 bytes) and, depending on the type, can be supplemented with a section of **external data**. The name may imply that this is external to the device, but in the case of the P89LPC900 it is definitely inside. The implications for the code are that the MOVX-instructions are required for access. A recent type, such as the P89LPC932, has, for example, 758 bytes, of which 256 bytes are direct/indirect and 512 are 'external'.

The code memory, based on flash technology, is not only repeatedly programmable, but can also be programmed **In System** (ISP) and even **In Application** (IAP). For this purpose, Philips provides a bootloader which can be easily activated and communicates via the UART by default. An RS232 level shifter is all that is required to download the code from a PC. Esacademy provide the necessary software (Flash-Magic), which uses a simple user interface or command line (for production purposes, for example) to send the hex file to the processor.

The internal bootloader consists partly of built-in, low-level code that takes care of the interaction with the memory, and partly of a program in flash that deals with the communications aspect and the translation of the hex file. The programmer can quite easily use the low-level subset of the bootloader by writing to a few registers and executing a 'call' to a specific address. In this way you can write your own implementation of a bootloader, which, for example, does not communicate with the UART, but uses some other method chosen by the programmer. This new bootloader can replace the standard one. **Figure 2** shows how a call to the IAP subset is realised in practice.

The memory consists of pages of 64 bytes each. The pages assemble into 1k sectors. Pages and sectors can be individually erased, with an impressive erase time of only 2 ms. Also note that when programming the flash memory, no additional programming voltages are required. The second article will treat the subject of programming the flash memory in more depth. It may not be necessary to mention this, but the code memory can be programmed with a parallel programmer as well.

In addition to the usual data and code memories, the P89LPC900 also has an EEPROM memory (64 to 512 bytes). The EEPROM is accessed via a number of registers (pointer, data and status), so that it is not necessary to write specific communications code, as required for some controllers when the EEPROM is accessed via the I^{2}C bus.

The P89LPC900 also has a number of special memory locations: the bootvector, the sta-

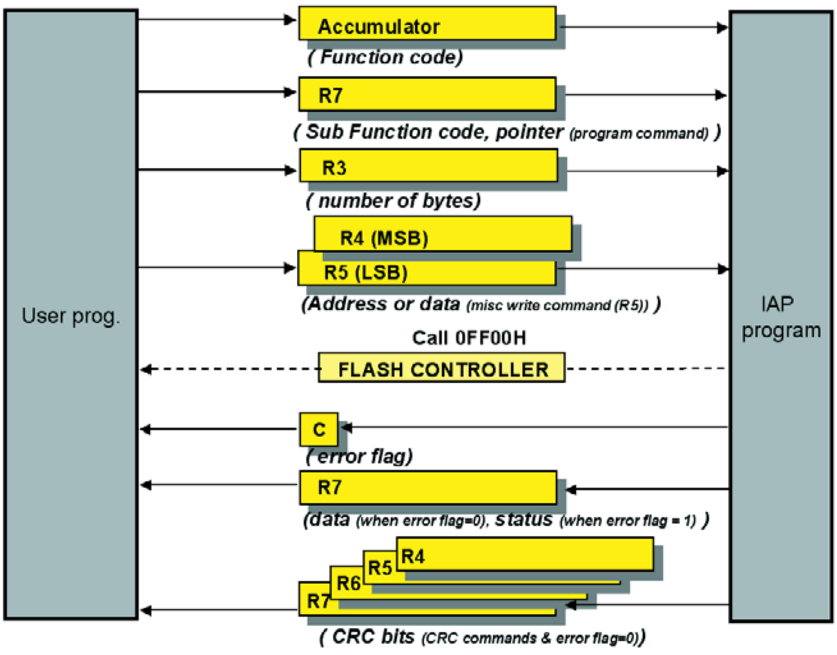

User prog.
Accumulator
(Function code)
R7
(Sub Function code, pointer (program command))
R3
(number of bytes)
R4 (MSB)
R5 (LSB)
(Address or data (misc write command (R5)))
Call 0FF00H
FLASH CONTROLLER
C
(error flag)
R7
(data (when error flag=0), status (when error flag = 1))
R4
R5
R6
R7
(CRC bits (CRC commands & error flag=0))
IAP program

Figure 2. Here you can see how a 'call' has been realised to the IAP subset.

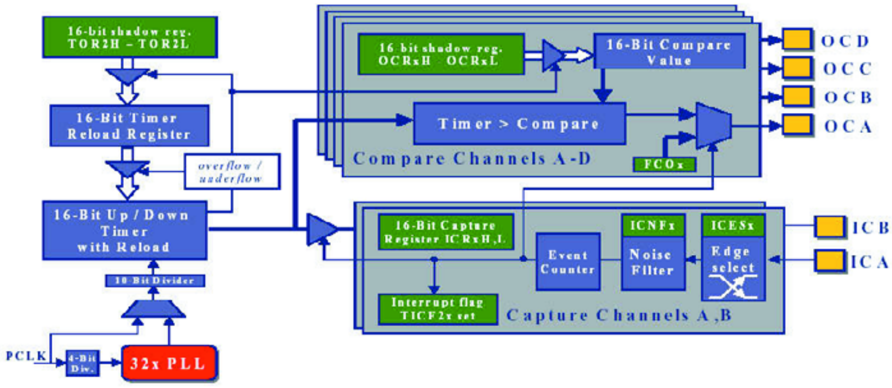


Figure 3. The user of the P89LPC932 can make use of 4 compare outputs and 2 capture inputs.

tus byte and the configuration byte. The bootvector determines the starting location when the chip is in boot mode. This vector is initially factory programmed to point to the internal bootloader. The status byte determines whether after reset the user code or the boot code is activated. In this case the initial value is again such that the boot code will be executed.

The last byte in this category is the configuration byte. It is used to select the desired **clock source**, the function of the **reset pin** and the system **supervisory settings**. The locations can also be accessed via ISP and IAP.

In- and outputs

Because the P89LPC900 has an internal **RC-oscillator** and a **Power-On-Reset** circuit, the basic rule is that the number of pins available for I/O is equal to the number pins on the package minus 2 for the power supply. In almost every case all the port pins can be programmed as **Input, Open-drain, Push-pull** or **Quasi-bidirectional**. Exceptions are the pins for the I^{2}C bus and reset input. These have a different configuration because of their dual functionality. Once the processor has started, all pins are set to **input**. Two registers for each port (PxM1 and PxM2) together define the desired configuration. For the analogue functions there is a third register to ensure that the digital circuitry is completely decoupled from the relevant pin.

Naturally, a number of pins have more than one function (for communications, among others) and a few are suitable for interrupts. A group of eight port pins is available that can be used as a **keyboard-interrupt**. This mechanism is also provided with a **Pattern Match**, which ensures that an interrupt (or a wake-up) is generated only when a specific pattern appears.

Five pins have, in addition to the digital I/O, an analogue comparator function. Two comparators with a selectable reference (internal or external) and switchable inputs can be used to compare signals and use the results to generate an interrupt. There are also versions with an 8- or 10-bit ADC.

When using the reset pin, take note of the alternative function of this pin. Even though the pin is defined as an input after power up, this pin is still sampled during reset. In the event that this pin is permanently ‘low’ the processor will not start up properly. In contrast to previous 80C51 controllers, the P89LPC900 has an active *low* reset!

System supervisory

The P89LPC900 is provided with all the features with regards to ‘System Supervisory’ functions. In the past a significant additional expense had to be made for external components that perform the same functions. Fortunately, they are now built in. The P89LPC900 has a **Power-On-Reset/Detect**, a **Low-Voltage Detect** and a **Watchdog**. The latter can, because of its separate oscillator, also function as an **Oscillator Fail Detect**. The activation of any of these functions does not necessarily result in a reset, but can generate an interrupt instead. A software-reset can then perform this task, if appropriate.

Power management

We mentioned before that the speed of the processor can be changed dynamically. For circuits that require even lower power consumption, there is the option of placing the processor in either **Idle** or **Power-down mode**. In both modes, the core itself is halted. The difference is that in Idle mode the peripherals continue to run, while in Power-down mode everything is stopped. The sources that can re-activate the processor differ for each mode. In Idle mode a timer can re-activate the processor, but in Power-down mode only an external/keypad or comparator interrupt can do this. Note that the watchdog can continue to run in Power-down mode, although this will increase the power consumption somewhat.

Timers and counters

The P89LPC900 has the well-known timer/counter 0 and timer/counter 1, which are more or less standard on every 80C51. New is the fact that the timers have an additional operating mode, which can be used as a ‘poor man’s PWM’. Registers THx and TLx together determine the frequency and duty cycle. There are, incidentally, versions (the P89LPC932 for example) that have a very comprehensive capture/compare unit with 16-bit resolution and a 32xPLL (refer **Figure 3**). This provides the user with 4 compare outputs, which permit all kinds of wave shapes, and two capture inputs. This module can be used to drive a stepper motor, for example, or control an H-bridge. A (noise filtered!) **Brake input** allows the option of hardware intervention in the event there is a need to immediately stop a stepper motor, for instance.

In addition to the timer/counters, the P89LPC900 is also provided with an RTC. This is a convenient way to keep track of elapsed time, without sacrificing another timer. **Figure 4** illustrates its operation.

Communications

The P89LPC900 has also acquired a range of new features in the area of communications. One of the more notable new features is the presence of a separate **baud-rate generator** for the UART. In the past it was necessary to sacrifice a timer for this purpose, which limited further use of this timer to whatever frequency it had to run at.

The UART also has a break-detect mechanism. This makes it possible to, for example, activate the bootloader in a very simple manner. The user can now, without any effort, program the processor from within the programming environment. The Flash-Magic software has a facility to make this happen.

The I^{2}C-interface complies with the 400-kHz standard and is byte-wide implemented, which greatly simplifies the code for the various I^{2}C-modes and makes it less critical. A **state machine** can make use of the codes that are returned by the I^{2}C hardware. This is an easy way to determine the progress of the I^{2}C-transfer and decide what the next step should be. Philips provides a detailed description in the user manual as to which states belong with which mode (Master transmit/receive, Slave transmit/receive). By the way, the I^{2}C-interface is compatible with earlier devices such as the P8xC554.

Even the SPI interface hasn't been forgotten. Communications at a speed of 3 Mbit is possible with a wide range of SPI devices. The SPI interface can be configured for either master or slave operation.

Interrupts

A large number of interrupts (15 on the P89LPC932) can be utilised to facilitate the smooth operation of the program. The interrupts have, by default, a pre-determined priority, but the user can divide them into four different levels. The interrupts can be activated or deactivated individually as well as globally.

Development tools

Because the P89LPC900 still has an 80C51 core, an assembler or compiler from the early days can still be used to write software for this family. The only thing that has changed is the required register file. This may be part of the assembler/compiler package or, alternatively, you can easily write one yourself.

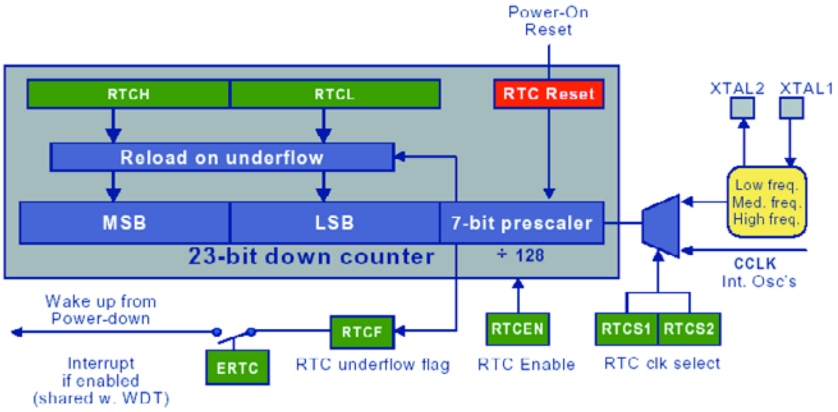


Figure 4. The standard RTC peripheral keeps track of elapsed time.

On the Esacademy web site you should be able to find 'Code-Architect' by the time this issue of Elektor Electronics is published. This tool can generate online code for various peripherals, without having to consult the data sheet. The code can then be processed further in a C compiler.

Although the entire P89LPC900 family is available with flash memory and code is readily reprogrammed, an 'in-system-emulator' can drastically reduce the development time and generally make life much easier. There are a number of manufacturers that offer emulators for these controllers (Philips, ESS and Keil). the Keil emulator is remarkable. For the amount of only 199.00 euro (approx. £140), Keil supplies a fully-fledged emulator, including a C compiler with a 4 K limit. These tools all have a USB connection, so that, up to a point, an external power supply can be omitted. For hobbyists who do not want to spend this much money, the design to be described in Part 2 of this article will provide an alternative solution.

The family

The number of P89LPC900-derivatives is growing at a great pace. At the time of writing, it appears that Philips has chosen to fill the category of up to 28 pins first with the P89LPC900, (there are, after all, several flash-controllers in the higher category already) before producing the larger devices. Even controllers with unusual peripherals such as CAN and USB have been planned.

Noteworthy are the 8-pin versions. Philips has decided to release a range of 8-pin devices into the market. It is simply the case that with 8 pins you don't get a great deal of choice regarding the peripherals. The various devices will therefore feature different combinations of peripherals.

Philips has done well to introduce a range of products on the market that have the same or greater functionality and are pin-compatible with the popular P87LPC76x, for example, which are only available in OTP versions.

The P89LPC900 are available in several different packages, but the emphasis will be on smaller packages such as the TSSOP (only 9.8 × 6.6 mm for the 28-pin version). The smaller technology (0.35 micron) and the fact that they are in-circuit programmable contribute to this. Fortunately, there are more practical package options for experimental work. In the project to be described in Part 2, the PLCC28 version has been used.

(030161-1)

Useful websites

www.philipsmcu.com
Philips 80C51-controllers and additional information such as application-notes and entry to the forum.

www.esacademy.com
Tools for the 80C51 such as Flash-Magic and Code-Architect.

www.altium.com
C-compiler, demo software limited to 2 K code can be downloaded.

www.keil.com
C-compiler, demo software limited to 2 K code can be downloaded.

www.pds51.com
Hardware tools for 80C51.

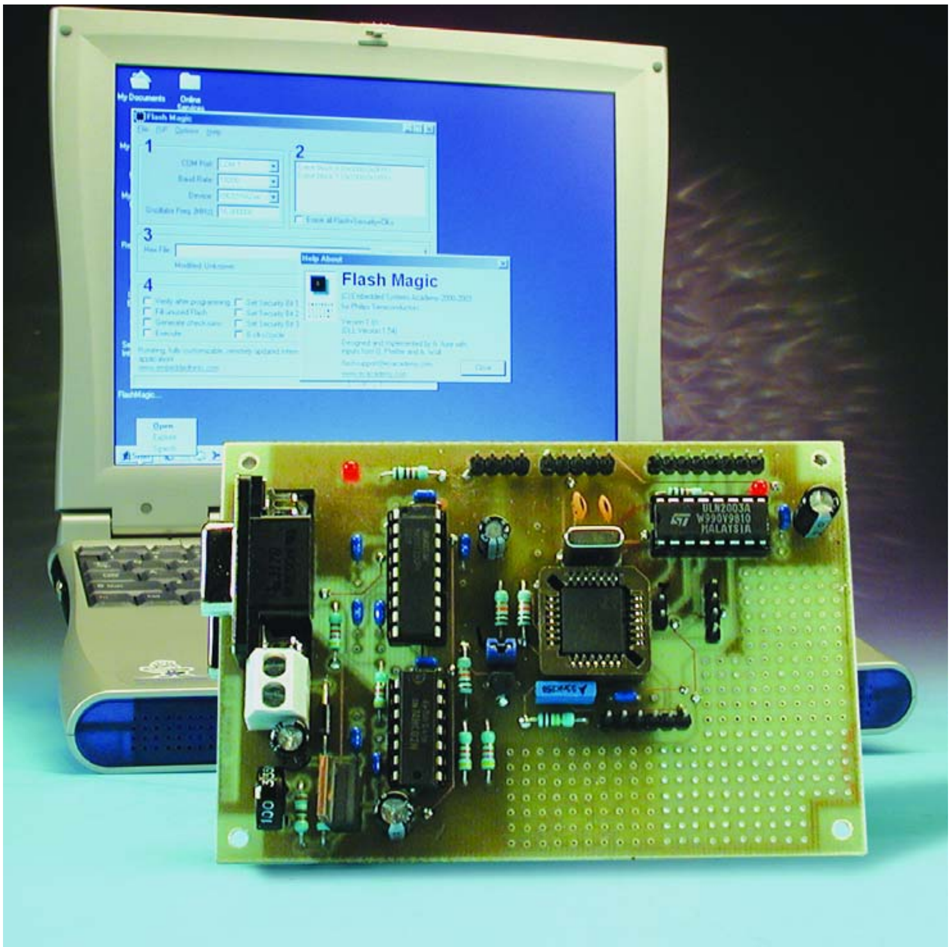
The P89LPC900 (2)

Part 2 (final): practical application

M. Stottelaar

Eurodis Texim Electronics

In this second instalment we will look at the actual application of this microcontroller in greater detail. Notably the 'in system programming' functionality is described in detail. In addition, a practical design is presented that not only functions as a programmer for the controller, but as a prototyping board as well. As an example project, guided by an application note from Philips, a very good ADC can be built without requiring any additional peripherals.



Addition to Part 1 (October 2003)

Hitex are the sole distributor of the Keil/Philips LPC kits the UK. The kits are available from www.hitex.co.uk/shop. We thank Mr. Chris Hills of Hitex UK for bringing this to our attention.

Flash memory

Many of you will recognise the ceramic microcontroller with a glass window. These are the UV erasable versions that were (and sometimes still are) used during the debugging stage of the application software. An expensive solution indeed, but the alternative, the burning of one-time programmable (OTP) devices was often more costly and far from flexible. The arrival of multi-time programmable flash (MTP) has made the UV-erasable type mostly obsolete. With that, the relatively long UV erasure time is now also a thing of the past.

Despite this large step forward, there is another highly desirable feature: it would be very nice if the microcontroller could be programmed and erased while it

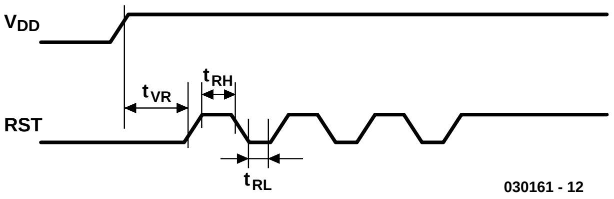


Figure 1. Three specific pulses at the Reset pin put the chip in boot-mode.

remains in the application circuit. In particular with SMD packages this very desirable of course. And if it were also possible to replace the code 'in the field', then all of our wishes would have come true.

Two pieces of code

All the recent versions of microcontrollers from Philips Semiconductors have the option of being (re)programmed while fitted in the circuit, and this includes the P89LPC900 series. Because nearly all versions have at least one UART, this is an eminently suitable method to provide the microcontroller with code. The only other things that are necessary are a PC running a small application and a level shifter to convert the RS232 signal levels to TTL logic levels. All the programming functionality is controlled by two pieces of code that reside in the microcontroller. The *low-level* part takes care of the actual reading and writing of the memory, while the *high-level* part (boot-loader) provides for the communications via the UART. The low level functions are available to a programmer and can be used to write your own boot loader, which would be able, for example, to reprogram the microcontroller via a CAN-network.
The P89LPC900 is incidentally quite happy with only one power supply voltage during programming, so there is no need for an additional programming voltage. The next few paragraphs provide more detail about the inner workings of the various elements.

In System Programming (ISP)

When it leaves the factory, the P89LPC900 has been pre-programmed with an enabled boot loader. When power is applied the microcontroller is ready to receive the application code. After the code has been sent, the microcontroller is switched to *user-mode*, so that the application code will be executed following the next reset.

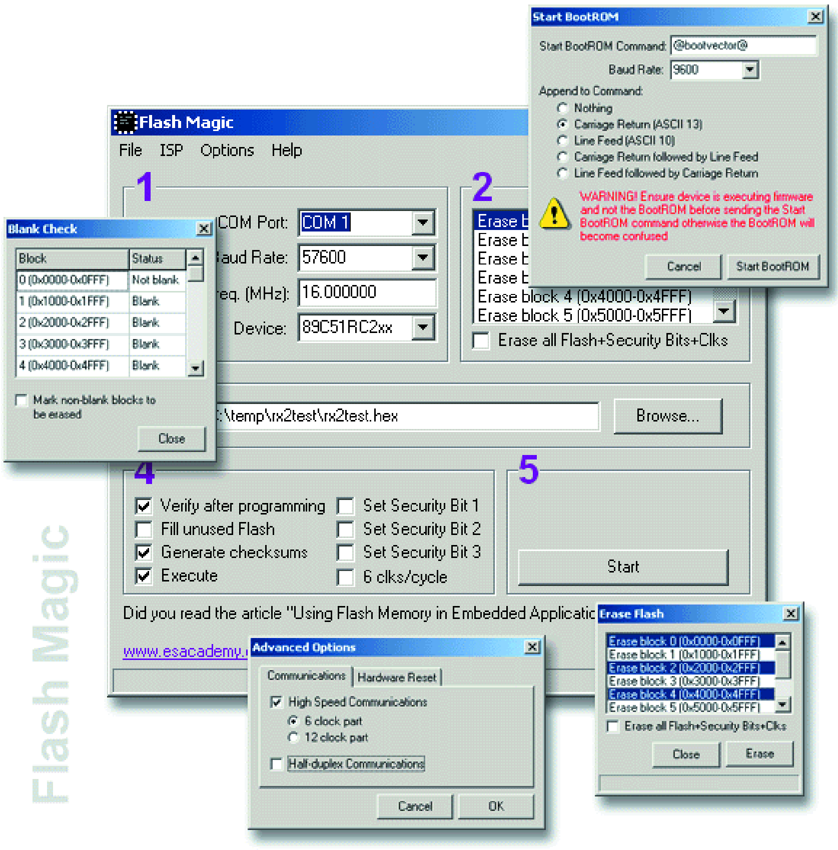
There is, of course, a method to return the microcontroller to boot mode again. The P89LPC900 actually has multiple methods to do this, two of which will be described here. A third method is outside our scope; it is also possible to put the processor in boot mode using a parallel programmer.

The first method of putting the chip in boot mode is to apply three specific pulses to the reset pin of the processor. In practice, the serial port of a PC can be used to generate these pulses. In addition to the usual RxD and TxD signals a third line is required to generate these pulses (refer to **Figure 1**).

The second, possibly even simpler, method uses the *Break Detect* function of the microcontroller. If this function has been enabled, the TxD line from the PC can put the microcontroller in boot mode. This means that the *Break Command* has to be sent to the microcontroller. This is certainly a very convenient method during the debugging stage of the software development cycle, particularly if the UART is accessible to the world via a sub-D connector. However this method also has its risks. The wave shape for the Break signal is rather simple and there is a significant risk that the controller will inadvertently enter boot mode. It is therefore recommended to disable this feature when the project has been completed. Software examples for enabling the break detect feature can be found in the application notes from Philips.

In part 1 we mentioned that Flash Magic (refer to **Figure 2**) is the name of the PC software used to program the microcontroller. This software is maintained by ESACADEMY (www.esacademy.com) and is periodically updated with a new list of microcontrollers to be programmed. The design is very simple and the environment has the look and feel of a typical programmer (note that Flash Magic is also suitable for other derivatives from the 80C51 family from Philips).

The circuit described below is suitable for use with Flash Magic and includes all the fea-



Start BootROM Command: @bootvector@
Baud Rate: 9600
Append to Command:
○ Nothing
● Carriage Return (ASCII 13)
○ Line Feed (ASCII 10)
○ Carriage Return followed by Line Feed
○ Line Feed followed by Carriage Return
WARNING! Ensure device is executing firmware and not the BootROM before sending the Start BootROM command otherwise the BootROM will become confused

COM Port: COM 1
Baud Rate: 57600
Freq. (MHz): 16.000000
Device: 89C51RC2xx

Block	Status
0 (0x0000-0x0FFF)	Not blank
1 (0x1000-0x1FFF)	Blank
2 (0x2000-0x2FFF)	Blank
3 (0x3000-0x3FFF)	Blank
4 (0x4000-0x4FFF)	Blank

□ Mark non-blank blocks to be erased

:\temp\rx2test\rx2test.hex

☑ Verify after programming
□ Fill unused Flash
☑ Generate checksums
☑ Execute
□ Set Security Bit 1
□ Set Security Bit 2
□ Set Security Bit 3
□ 6 clks/cycle

Did you read the article "Using Flash Memory in Embedded Applicatic
www.esacademy.c

Erase block 0 (0x0000-0x0FFF)
Erase block 1 (0x1000-0x1FFF)
Erase block 2 (0x2000-0x2FFF)
Erase block 3 (0x3000-0x3FFF)
Erase block 4 (0x4000-0x4FFF)
Erase block 5 (0x5000-0x5FFF)
□ Erase all Flash+Security Bits+Clks

☑ High Speed Communications
● 6 clock part
○ 12 clock part
□ Half-duplex Communications

Figure 2. Menu structure of Flash Magic.

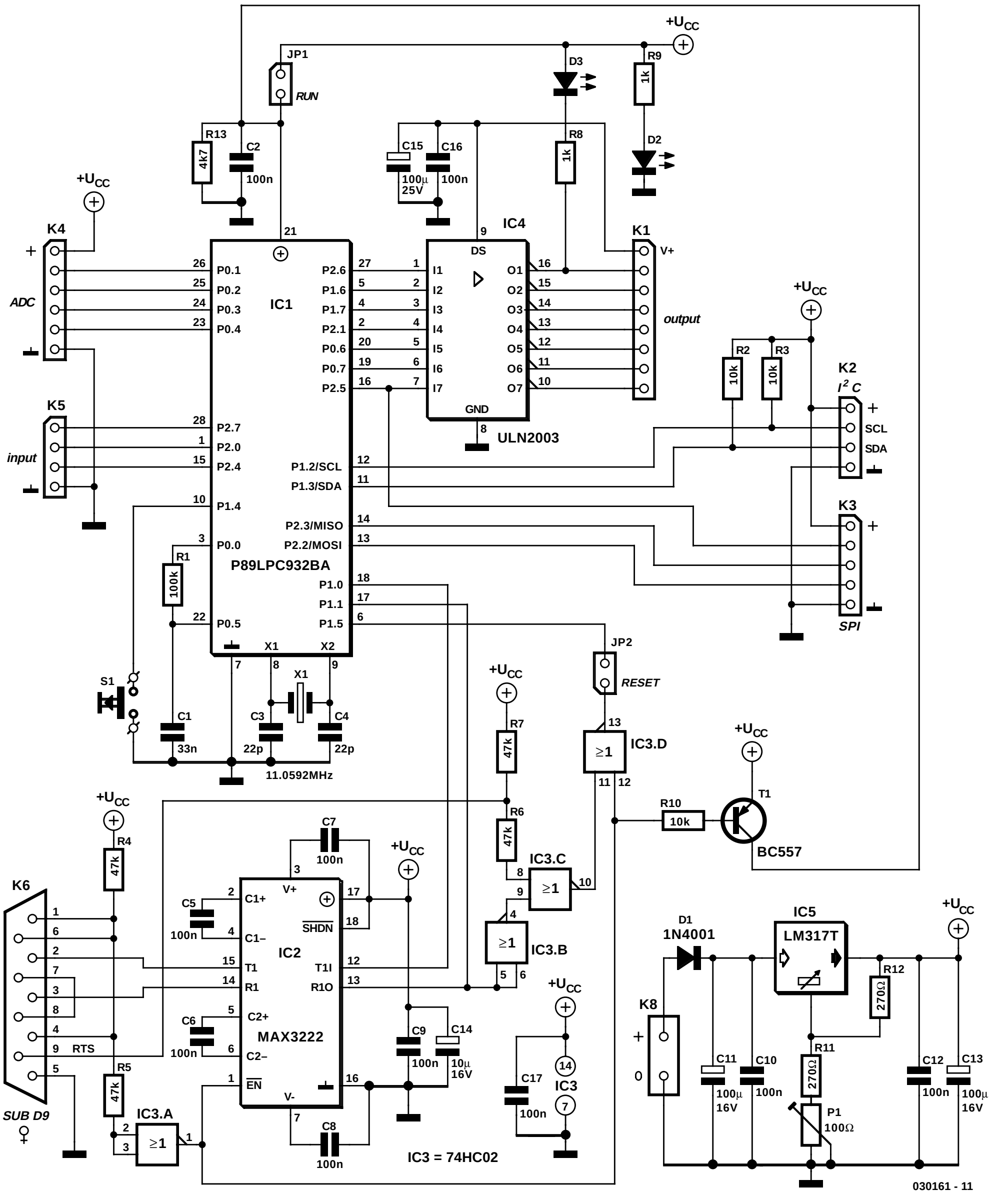


Figure 3. The schematic for the programmer and prototyping board.

tures for putting the microcontroller in bootmode. Flash Magic can also be operated from the command line. This makes enables you to use this tool as part of an automated production process.

It is also possible, of course, to write your own programming environment to suit your own unique requirements. For this purpose, the Intel Hex strings are supplemented with additional instructions. It is beyond the scope of this article to describe all these instructions in detail, but the P89LPC932 user manual provides all the necessary information.

In Application Programming (IAP)

In some high-end applications, there may be a need to replace the code in the microcontroller without replacing the IC itself. Of course, the ISP method is a possible solution in these situations. But what about an environment where there is no technical know-how or nobody is actually present?

In all these situations, the IAP-method can be used to replace the code using a method chosen by the application programmer. The communications method can be selected to suit the environment (telephone line, Internet, RF, etc.). The principle is very simple. The programmer defines the communication method and makes use of the so-called *IAP calls* in the microcontroller to carry out the actual erasing and writing operations. Obviously, part of the code needs to remain active during this process in order to maintain the communications channel. However, using segmentation, all the code can be replaced in two or more steps (in many cases the system software can be left unaltered because it is generally subjected to fewer changes than the actual application code).

A practical description of the IAP operation may be found on the *Elektor Electronics* website under Free Download file number **030161-2** (November 2003) on www.elektor-electronics.co.uk.

The hardware

That was enough theory for now. The circuit shown in **Figure 3** makes it possible to get started with the P89LPC900 family in a practical way. The design supports the previously mentioned ISP and IAP facilities, so that the testing of your own programs can be realised very easily. A C compiler, for example, can be used for the development of the software. If you don't have a compiler, one can easily be downloaded from the Internet (for example Tasking or Keil). In general these are 'full featured' demo versions, but limited with respect to the maximum generated code size. However, a 2k limited version is often adequate to write a realistic application. Also, the 'Code Architect' from Esacademy can be very helpful when first treading in 80C51 territory. Using this tool, you can interactively write code for the P89LPC932 that has been used in this design. This member of the family is part of a whole range of controllers with 8 to 28 pins and also available in PLCC packages.

To maximise the functionality of the board, all the I/O from the microcontroller has been brought to the outside and is available for use. The combination of the serial interface and analogue and/or digital I/O can easily be put to practical use to collect measurements with a PC or control some outputs.

In order to program the microcontroller, the simplest connection between the PC and P89LPC932 could consist of a single RS232 level shifter that we have used on many previous occasions in *Elektor Electronics* designs. The RXD and TXD lines from the PC and P89LPC932 are simply connected via the level shifter. In this design we have chosen for a slightly more elaborate setup. Not only are the RXD and TXD lines connected, but some additional signals as well, to provide a safe method of switching the programmer on and off, and for generating the clock pulses at the reset pin (for reboot).

COMPONENTS LIST

Resistors:

R1 = 100kΩ
R2,R3,R10 = 10kΩ
R4...R7 = 47kΩ
R8,R9 = 1kΩ
R11 = 390Ω
R12 = 270Ω
R13 = 4kΩ7
P1 = 100Ω preset V

Capacitors:

C1 = 33nF
C2,C5-C10,C12,C16,C17 = 100nF
C3,C4 = 22pF
C11,C13,C15 = 100µF 25V radial
C14 = 10µF 16V radial

Semiconductors:

D1 = 1N4001
D2,D3 = low-current LED
T1 = BC557
IC1 = P89LPC932BA
IC2 = MAX3222CPN
IC3 = 74HC02
IC4 = ULN2003
IC5 = LM317T

Miscellaneous:

JP1,JP2 = jumper
K1 = 8-way SIL pinheader
K2,K5 = 4-way SIL pinheader
K3 = 5-way SIL pinheader
K4 = 6-way SIL pinheader
K6 = 9-way sub-D socket (female), PCB mount
K8 = 2-way PCB terminal block, lead pitch 5mm
S1 = pushbutton, one make contact
X1 = 11.0592MHz quartz crystal
PCB, available from The PCBShop

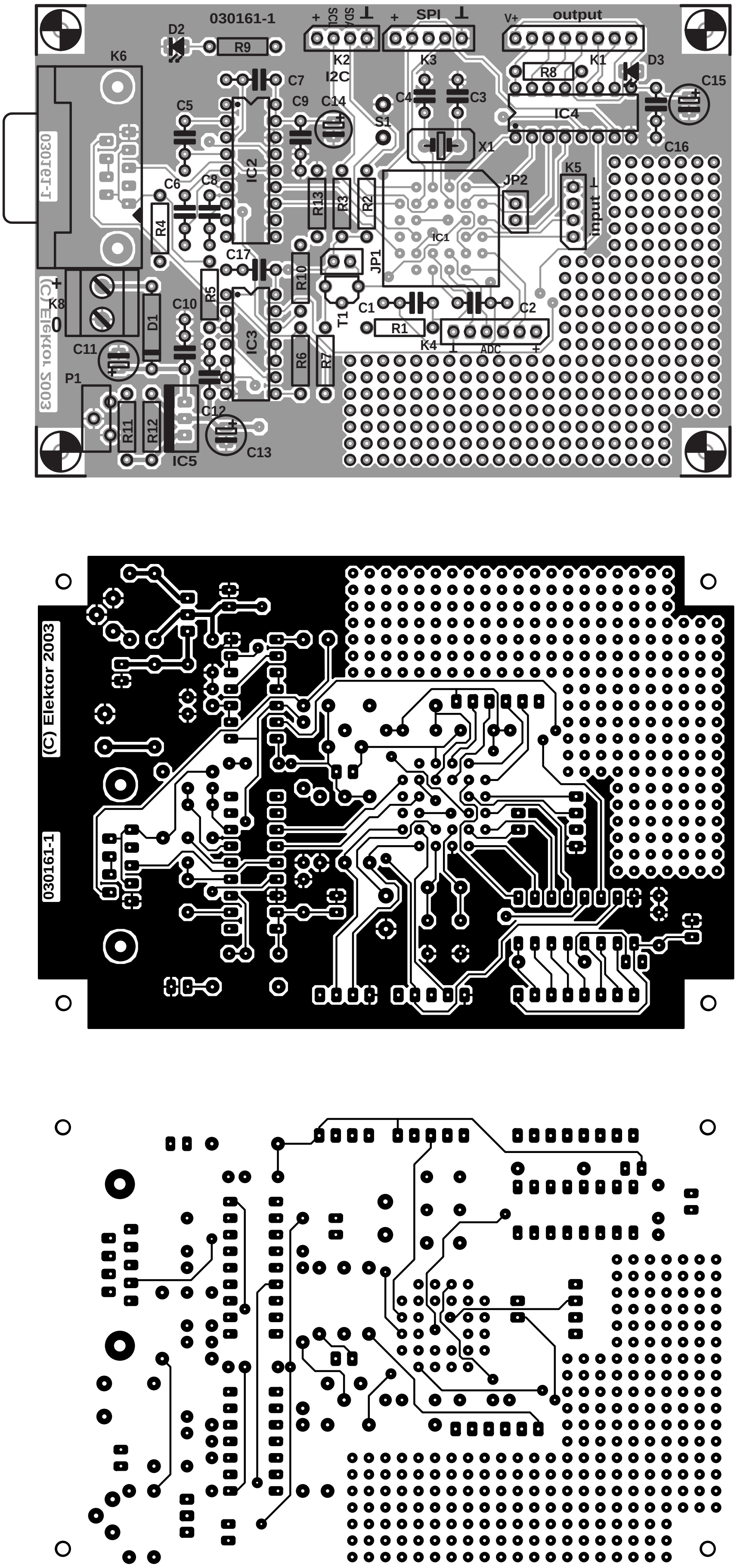


Figure 4. There is plenty of room on the PCB for your own applications.

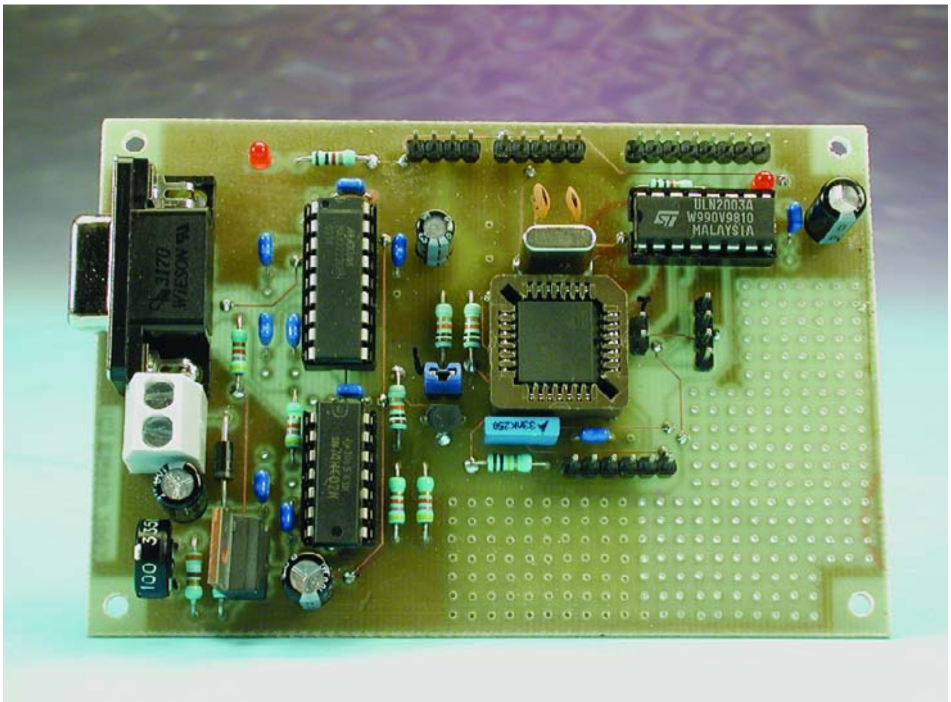


Figure 5. The construction of the prototyping board is a relatively simple job.

IC2 converts the signal levels and is provided with an enable input, which causes the chip to go completely off-line when the DTR signal is no longer active. The DTR line is also used, via IC3 and T1, to switch the power supply voltage to the P89LPC932 on and off during the programming session. The three pulses that are required to put the microcontroller in boot mode are generated using the RTS line from the PC, combined with the TXD signal.

A few more schematic details

R1 and C1 form the circuit that's used for the sigma-delta converter to be described later on. P0.1 through P0.4 are used as analogue inputs for this purpose, so that a four-channel ADC is created. Fitting C3, C4 and X1 is not strictly necessary, because the processor has a sufficiently accurate RC oscillator built in. These connections can be used as digital I/O. The signals for both I^{2}C and SPI are also brought out to pins so that many popular chips can be connected directly.

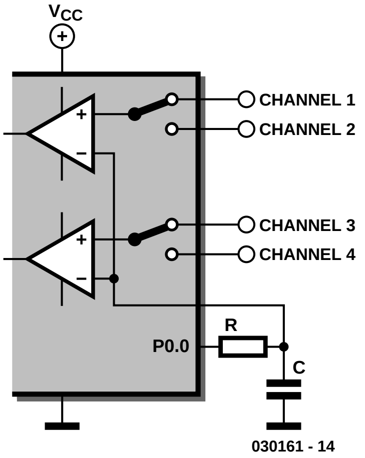


Figure 6. The switchable comparator inputs provide the option of four ADC channels.

The four compare outputs (pins 2 through 5 of K1) and another three ports are buffered and made available via IC4, allowing relays or similar to be controlled directly. Port P2.6 (pin 2 on K1) is fitted with an LED (D3) that can be useful during software development. The remaining ports, including P2.7, P2.0 and P2.4 are available as general-purpose I/O (on K5).

Note that P1.5 can never be configured as an output and in this application is only used as Reset.

A pushbutton (S1) is connected to port 1.4. This could be used, for example, to test the software example of boot-mode via IAP. Even though the processor is a 3-V type, all the inputs and outputs are tolerant of 5-V signals. It may be necessary in some cases to convert the signal levels if they do not conform to the input requirements of some optional external chips.

The power supply of the circuit is a very simple design. We chose an LM317 (IC5) with a trimpot, because it is more readily available than a fixed 3.3V-regulator. Before placing the processor in its socket check that the power supply voltage is around 3.3 V, adjustable with P1.

Construction

Figure 4 shows the (double sided) PCB designed for the prototyping board. The layout is quite spacious and conveniently arranged, so that even less experienced hobbyists will have no problem with the construction, provided you stick to the component overlay and parts list. Microcontroller IC1 is fitted in a special PLCC socket. The various connecters have been placed near the edge of the PCB as much as is possible.

A section beside the controller has been deliberately designed as prototyping board. This provides an easily accessible space for your own applications. In this way you can carry out a lot of different experiments without immediately having to make another PCB.

A standard 9-VDC mains adapter can be used to power the entire circuit. There are no specific requirements for the adapter, since IC5 provides voltage regulation and the current consumption of around 20 mA is not very large. Diode D1 prevents an incorrectly connected adapter causing damage to the circuit.

The programming

After everything has been put together and carefully checked, the PCB can be connected to the power supply and serial port. Use the following jumper settings:

	During Flash Magic	During Run-mode
JP2	Fitted	Removed
JP1	Removed	Fitted
N.B.	Reset and power controlled with Flash Magic	User-reset and uncontrolled power

Although Flash Magic is a program that's self-explanatory to a large extent, it is worthwhile to take a quick look at the settings related to *Start Bootrom* in the ISP menu. A number of things depend on whether *Break Detect* or *Pulses on Reset Pin* is used. The latter option will always work, provided the reset jumper is fitted and in the menu 'Options/Advanced Options/Hardware Config' the tickbox 'Use DTR and RTS to enter ISP-mode' has been ticked. The other method will only work when the (running) code has provision for this option, as described earlier.

Ex factory, the boot-loader is already enabled so that loading the first program will succeed even without explicitly enabling the boat loader. Consult the accompanying PDF-file for more instructions with regard to Flash Magic.

Attention: up to version C of the P89LPC932 it was not possible to activate the boot loader using the reset pin. With these devices you will always have to make use of either the *Break Detect* or *IAP* method. So be careful and convince yourself that your code is correct. The only method of escape once you're stuck is by using a parallel programmer to clear the chip after a mistake. Fortunately only a small number of these devices have been manufactured.

ADC

As already mentioned, the above design has a feature that allows an ADC to be realised even though the P89LPC932 does not have an *on-board* ADC. The principle of operation is based on the *Sigma-Delta* method. The two analogue comparators that are standard peripherals on this controller are employed for this purpose. Because each comparator has a selectable input (refer to **Figure 6**), it becomes possible to make a four-channel ADC. Application Note AN10187Ω1, available from the Philips website provides all the necessary information. But, briefly, the circuit operation is essentially as follows: C1 on the reference input is charged via R1 until the level is equal to the input of the comparator. From then on, the charge/discharge cycle is controlled by the result from the comparator. The number of charge cycles is proportional to the comparator input voltage. By counting the number of charge cycles during a predetermined number of cycles (charge and discharge) the level of te input voltage can be determined. For a more detailed description and background theory we refer you to the aforementioned application note.

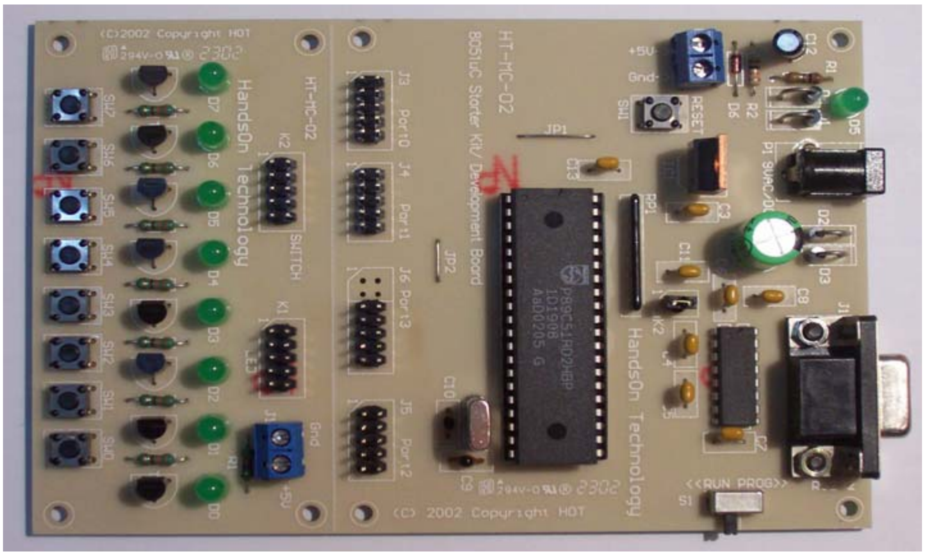
In conclusion

It would go too far to describe all the details of the P89LPC932. Fortunately, there is plenty of code available for the 80C51 (often well documented in application notes). In addition, new application notes frequently appear on the Philips website, explaining the details of specific peripherals.

(030161-2)

Low Cost 8051µC Starter Kit/ Development Board *HT-MC-02*

HT-MC-02 *is an ideal platform for small to medium scale embedded systems development and quick 8051 embedded design prototyping.* ***HT-MC-02*** *can be used as stand-alone 8051µC Flash programmer or as a development, prototyping and educational platform*



Main Features:

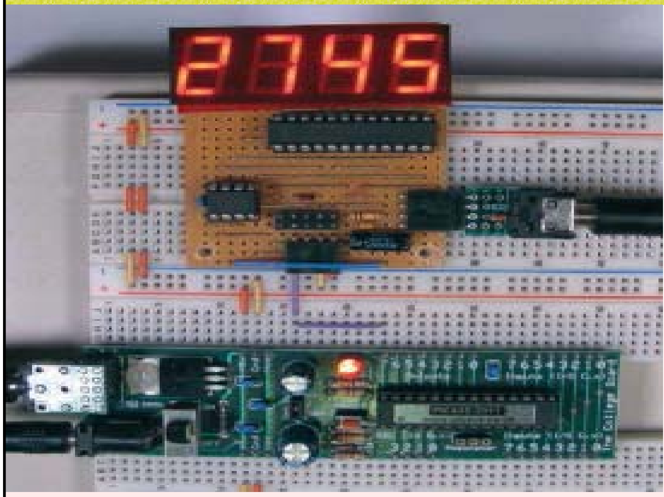
- 8051 Central Processing Unit.
- On-chip Flash Program Memory with In-System Programming (ISP) and In Application Programming (IAP) capability.
- Boot ROM contains low level Flash programming routines for downloading code via the RS232.
- Flash memory reliably stores program code even after 10,000 erase and program cycles.
- 10-year minimum data retention.
- Programmable security for the code in the Flash. The security feature protects against software piracy and prevents the contents of the Flash from being read.
- 4 level priority interrupt & 7 interrupt sources.
- 32 general purpose I/O pins connected to 10pins header connectors for easy I/O pins access.
- Full-duplex enhanced UART **–** Framing error detection Automatic address recognition.
- Programmable Counter Array (PCA) & Pulse Width Modulation (PWM).
- Three 16-bits timer/event counters.
- AC/DC (9~12V) power supply – easily available from wall socket power adapter.
- On board stabilized +5Vdc for other external interface circuit power supply.
- Included 8x LEDs and pushbuttons test board (free with ***HT-MC-02*** while stock last) for fast simple code testing.
- Industrial popular window ***Keil*** C compiler and assembler included (Eval. version).
- Free ***Flash Magic*** Windows software for easy program code down loading.

PLEASE READ ***HT-MC-02 GETTING STARTED MANUAL*** BEFORE OPERATE THIS BOARD
INSTALL ACROBAT READER (AcrobatReader705 Application) TO OPEN AND PRINT ALL DOCUMENTS

HandsOn Technology

http://www.handsontec.com

creativity for tomorrow's better living...



HandsOn Technology is a manufacturer of high quality educational and professional electronics kits and modules, uController development/evaluation boards. Inside you will find Electronic Kits and fully assembled and tested Modules for all skill levels. Please check back with us regularly as we will be adding many new kits and products to the site in the near future.

Do you want to stay up to date with electronics and computer technology? Always looking for useful hints, tips and interesting offers?

Inspiration and goals...

HandsOn Technology provides a multimedia and interactive platform for everyone interested in electronics. From beginner to diehard, from student to lecturer... Information, education, inspiration and entertainment. Analog and digital; practical and theoretical; software and hardware...

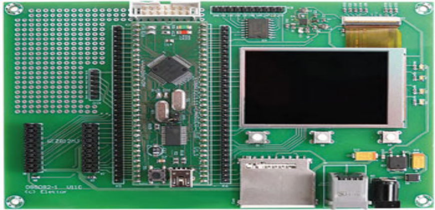
HandsOn Technology provides Designs, ideas and solutions for today's engineers and electronics hobbyists.

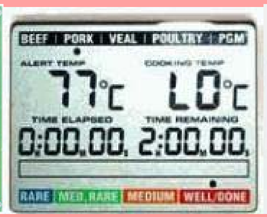
Creativity for tomorrow's better living...

HandsOn Technology believes everyone should have the tools, hardware, and resources to play with cool electronic gadgetry. HandsOn Technology's goal is to get our "hands On" current technology and information and pass it on to you! We set out to make finding the parts and information you need easier, more intuitive, and affordable so you can create your awesome projects. By getting technology in your hands, we think everyone is better off

We here at HandsOn like to think that we exist in the same group as our customers >> curious students, engineers, prototypers, and hobbyists who love to create and share. We are snowboarders and rock-climbers, painters and musicians, engineers and writers - but we all have one thing in common...we love electronics! We want to use electronics to make art projects, gadgets, and robots. We live, eat, and breathe this stuff!!

If you have more questions, go ahead and poke around the website, or send an email to sales@handsontec.com. And as always, feel free to let your geek shine - around here, we encourage it...




BEEF | PORK | VEAL | POULTRY | PGM
77°C
60°C
0:00.00
2:00.00
RARE | MED. RARE | MEDIUM | WELL DONE

http://www.handsontec.com