Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 49(63), Fascicola 2, 2004

Development System Equipped with AT89S8252 Microcontroller

Petruţ Duma[1], Luminiţa Scripcariu[2]

Abstract – The work describes the hardware structure of a development system based on AT89S8252 microcontroller. This system is used for testing and checking various user applications based on systems equipped with ATMEL family microcontrollers. The monitor program of the system and its commands are also presented.
Keywords: development system, ATMEL microcontroller, monitor program commands.

I. INTRODUCTION

Systems equipped with microcontrollers are classified depending on their purpose as dedicated systems and development systems.
Dedicated microcontroller-based systems are used only in the specific applications for which they were designed, have a reduced volume of hardware and the command software is included in the internal program memory of the microcontroller.
Development microcontroller-based systems are used for educational purposes, i.e. initiating the users in the hardware and software technique of microcontroller-based applications, but also for checking and testing both programs and hardware structures designed for various applications.
The hardware volume of these systems is larger, while the monitor program must allow viewing and/or editing resources contents, writing user programs, executing them on segments, etc.

II. DEVELOPMENT SYSTEM EQUIPPED WITH AT89S8252 MICROCONTROLLER

AT89S8252 microcontroller includes on a single chip the following resources:
- a microprocessor optimized for command and control applications;
- an 8 Kbytes in-system reprogrammable downloadable FLASH program memory, which can be used up to 1,000 writing/erasure cycles;
- an interface for serial program loading;
- a three-level internal program memory access blocking system;
- a 256 bytes RAM data memory;
- a 2 Kbytes EEPROM data memory, which can be used up to 100,000 writing/erasure cycles;
- a memory space for special function register (SFR);
- four 8-bit parallel ports (P0, P1, P2, P3);
- three 16-bit counters (T0, T1, T2);
- programmable Watchdog timer (WDT);
- a unit for the serial asynchronous data communications (UART);
- a serial peripheral interface (SPI);
- maskable interrupt system, with six sources on two priority levels;
- low-power idle and power-down modes;
- clock oscillator operating on frequencies up to 24 MHz (33 MHz).

The development system equipped with AT89S8252 microcontroller has the structure shown in fig.1.
The clock generator is internal and requires connecting at pins $XTAL_1$, $XTAL_2$ a quartz crystal of 11.0592 MHz and two 30 pF capacitors.
The automatic power-on initialization of the microcontroller is performed by the external RC group (10 kΩ, 10 μF).
The manual initialization of the microcontroller can be made using the K switch.
The monitor program of the development system is placed in the internal FLASH program memory of the microcontroller which requires /EA pin to be connected at +5V.
In order to run applications with the development system, they must be stored in an external non-volatile memory. The author chose to use a 32 Kb EPROM memory chip (27256), connected as in fig.2 and addressed in 0000 – 7FFFH memory space, by connecting BA_{15} to /CE and signals /PSEN to /OE.
In this case (/EA= + 5V) the program memory of the development system consists of the 8 Kb of microcontroller internal FLASH memory, addressed in the memory area of 0000 – 1FFFH, and of 24 Kb of external EPROM memory, in the memory area between 2000H – 7FFFH. The solution is useful for development systems with ATMEL microcontrollers that don't have internal program memory, case that requires pin /EA to be connected to Gnd.

Technical University "Gh. Asachi" Iaşi, Faculty of Electronic and Telecommunications, Bd. Copou 11, Iaşi, 6600, Romania, E-mail: pduma@etc.tuiasi.ro

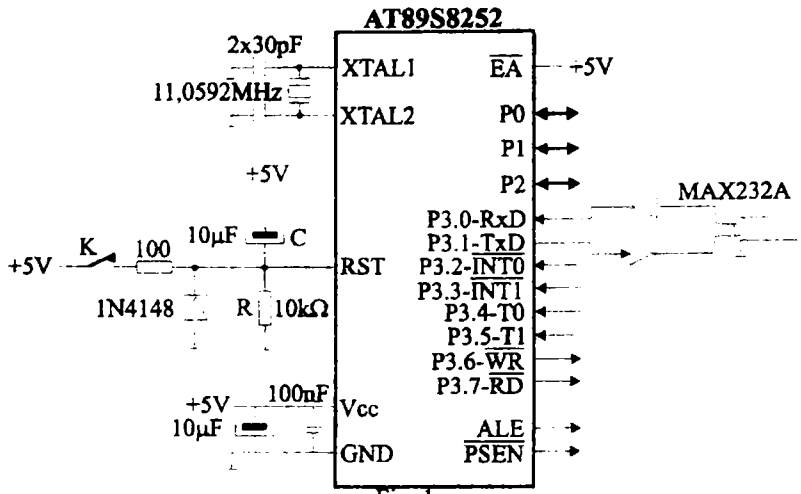


Fig. 1.

The microcontroller has separate command signals for accessing the program memory (/PSEN- addresses 64 Kbytes of program memory) and the data memory (/RD, /WR – also address 64 Kbytes of data memory). In order to execute user programs stored in the data memory, the program memory space and data memory space must be joined.

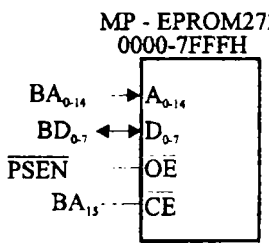


Fig. 2.

The SRAM memory chip (55257), connected as in fig.3, has also a 32 Kbytes capacity and it is addressed in 8000H – FFFFH memory space, by connecting line BA_{15} to /CE and the signal /PSEN./RD./WR to /OE.

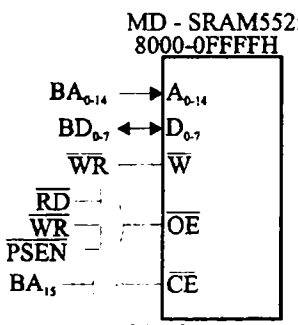
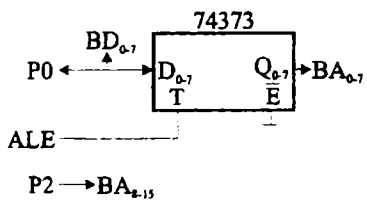


Fig. 3.

When addressing an external memory, P2 port generates the high part of the address bus, while P0 port generates the low part of the address bus multiplexed with the data bus. In the first stage of any access cycle to the external memory, the low part of the address bus is delivered on the P0 port lines, while during the other stages, these lines are used by data.
Using an external latch (74373) on the falling edge of the signal ALE, the lower part of the address bus is demultiplexed by the data bus (fig.4).



g. .

Using external memory in the structure of the microcontroller-based system determines the loss of P0, P2 and partly P3, which become data, addresses and commands busses.
The communication of the development system with a serial console / personal computer requires the use of the serial asynchronous interface (UART) and of a counter (T1 or T2) for setting the serial communication rate. These resources of the microcontroller are indispensable for numerous applications and therefore, several peripheral circuits must be connected to the system's structure in order to compensate.
The peripherals used are: INTEL 8255, INTEL 8253 and INTEL 8251, connected as in fig.5.
The INTEL 8255 is a parallel programmable interface with 3 parallel ports of 8 bits each and compensate ports P0, P2 and P3.
The INTEL 8253 is a programmable counter/timer interface with three 16 bits counters and compensates the counter used for setting the serial communication rate.
The INTEL 8251 is a programmable serial interface, used for transmitting and receiving serial synchronous and asynchronous data (UART) and compensates the microcontroller's serial interface.
These peripherals are assumed as data memory locations and are addressed using 8205 decoder.

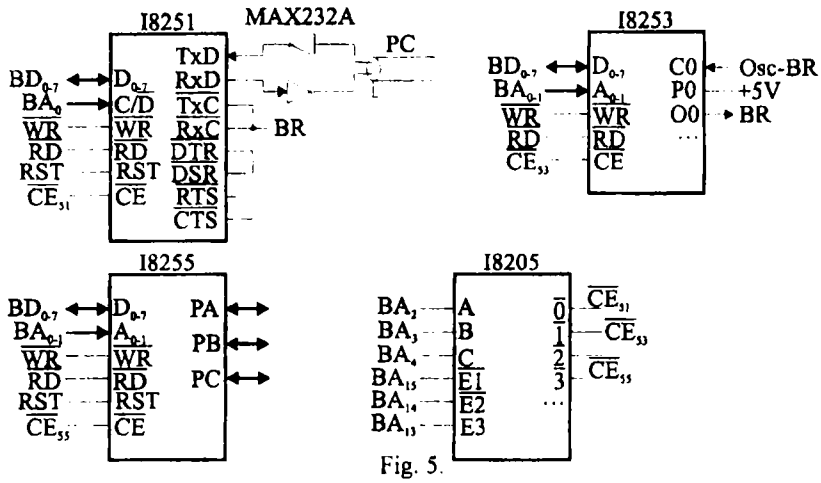


Fig. 5.

III. THE MONITOR PROGRAM COMMANDS

After connecting the development system to the DC power supply, the microcontroller starts to execute the monitor program, which consists, at the beginning, of a sequence of initializations, then the console's display is cleared and a monitor program launching message is displayed. Afterwards, a loop is started, during which the console keyboard, used for the transmission of the commands towards the microcontroller-based system, is tested.

When a user command is issued, the monitor program receives the command and its parameters, tests the syntax then allows its execution. After performing the user's command, the loop is resumed, awaiting a new command.

Once the microcontroller-based system powered-on, its specific process is working continuously, executing either the monitor program or commands or user programs issued by the user through the monitor program.

The general syntax of a command is:

£ P_1 _ P_2 _ P_3 _ ..... _ P_k .

where: **£** is the name of the command starting with a letter; P_1, P_2, P_3, ..., P_k are the command's parameters, consisting of data or addresses.

Between the parameters, a separator must be used, consisting of *BLANK* (_) or *COMMA* (,), any command being concluded with terminator *RETURN* (.). Any command of the monitor program can be abandoned at any time if "O" key is pressed.

The syntaxes and the effects of the commands accepted by the monitor program are described below, in alphabetical order.

A .

Displays on the console the list of the commands accepted by the monitor program.

B .

Sets the serial asynchronous communication rate for the UART interface (19,200 bits/s, 9,600 bits/s, 4800 bits/s 2,400 bits/s, 1,200 bits/s, 600 bits/s, 300 bits/s, 150 bits/s).

CA P_1 _ P_2 .

Performs arithmetic operations: addition (P_1+P_2), subtraction (P_1-P_2), multiplication (P_1*P_2), and division (P_1:P_2), between parameters P_1 and P_2.

CL P_1 _ P_2 .

Performs logic operations: AND, OR, XOR between parameters P_1 and P_2.

DC P_1 _ P_2 .

Displays on the console the user program in machine code, stored in the program memory from address P_1 to address P_2. Each console row displays the address and the 1 up to 3 instruction byte. The user program is written in machine code using substitute command or is loaded through the serial asynchronous interface.

DF P_1 .

Displays on the console the user program listing (maximum 26 – 28 Kb), assembled on a personal computer. This program is stored in the external RAM memory starting from address P_1, using the serial asynchronous interface.

DH .

Displays on the console the internal data memory area of the microcontroller, indirectly addressable between addresses 80H and FFH.

DL .

Displays on the console the internal microcontroller data memory area, directly and indirectly addressable between addresses 00H and 7FH; in the memory space between 00H-1FH are placed the four memory banks of eight general registers each, while the memory area 20H-2FH is also addressable on bit level.

DP P_1 .

DD P_1 .

DE P_1 .

Displays on the console the contents of a program memory area (DP command), external data memory area (DD command) and internal EEPROM data memory area, respectively, (DE command) from address P_1 until address P_1+80H. Each row displays the address and the

contents of 16 memory bytes in hexadecimal.

DP P_1 _ P_2 .

DD P_1 _ P_2 .

DE P_1 _ P_2 .

Displays on the console the contents of a program memory area (DP command), external data memory area (DD command), and internal EEPROM data memory area, respectively (DE command) from address P_1 until address P_2.

DP command displays any internal program memory area within the memory space between 0000 – 1FFFH and any external program memory area within the memory space between 2000H – FFFFH, if /EA= + 5V, or even any external program memory area whatsoever if /EA=Gnd.

DE command validates the access to the internal EEPROM data memory, then displays the specified area and, in the end, re-blocks the access to this area.

DS .

Displays on the console the internal memory zone of the microcontroller, addressable directly between the addresses 80H and FFH; this memory zone consists of the special function registers (SFR) area.

DT P_1 _ P_2 .

Displays on the console the characters corresponding to ASCII codes stored in the external data memory from address P_1 to address P_2.

The console display can be suspended if key "A" is pressed, resumed if key "C" is pressed, or aborted if key "O" is pressed.

E P_1 _ P_2 _ P_3 _... .

Extracts break-points in the user program from addresses P_1, P_2, P_3, ...

FD P_1 _ P_2 _ P_3 .

FE P_1 _ P_2 _ P_3 .

FI P_1 _ P_2 _ P_3 .

Fills the external RAM data memory (FD command), the internal EEPROM data memory (FE command) and the internal RAM data memory, respectively (FI command) from address P_1 to P_2 with P_3 data value.

G P_1 _ P_2 _ P_3 _... .

Determines the execution of a user program from address P_1 until one of the addresses of the break-points P_2, P_3, ... After executing the user program segment, the break-points assigned by this command are removed.

I P_1 _ P_2 _ P_3 ... .

Introduces break-points in the user program at the addresses P_1, P_2, P_3...

LC .

Loads from a personal computer through the serial interface, a machine code user program into the external RAM program memory and runs the program. The address used for loading the user program, the length of the program, the execution address and the break-point addresses are transmitted using the header of the file that includes the user program.

LP P_1 .

Loads from a personal computer through the serial interface, the user program listing file (max. 26-28 Kb) into the external RAM memory starting from address P_1. The machine code user program is extracted from this file.

MD P_1 _ P_2 _ P_3 .

ME P_1 _ P_2 _ P_3 .

MI P_1 _ P_2 _ P_3 .

Transfers the contents of the external data memory area (MD command), internal EEPROM data memory area (ME command), and internal data memory area, respectively (MI command) from address P_1 to address P_2 into the memory area starting at address P_3 .

N .

Clears the console display.

P .

Authorizes the password access of the user to the SFR peripheral registers.

R P_1 _ P_2 .

Receives a hexadecimal data block through the serial interface, from a computer or another development system and stores it in the external RAM data memory from address P_1 to address P_2 .

SD P_1 _... .

Displays and/or substitutes the contents of an external data memory area that starts at address P_1. The command displays every byte of any external data memory area, while the substitution is performed only in RAM external data memory locations.

For every substitution commands, the use of *BLANK* separator displays and/or substitutes the contents of the following memory location, while the use of *COMMA* separator displays and/or substitutes the contents of the previous memory location. The command is closed with (.) terminator.

SE P_1 _... .

Displays and/or substitutes the contents of an internal microcontroller EEPROM data memory area that starts at address P_1. When the command is issued, the access to the internal EEPROM data memory is validated, then the memory locations are displayed and/or substituted, and in the end, the access to this memory is blocked.

SI P_1 _... .

Displays and/or substitutes the contents of an internal microcontroller RAM data memory area that starts at address P_1.

SP P_1 _... .

Displays and/or substitutes the contents of a program memory area that starts at address P_1. The command displays every byte of any internal program memory area within the space between 0000H and 1FFFH and of any external program memory area within the space between 2000H and FFFFH if /EA= + 5V and, respectively, any external program memory area if /EA=Gnd. The substitution is performed only in the

RAM external program memory locations.

SS P_1 _ ... .

Displays and/or substitutes the contents of an internal microcontroller data memory area destined to the special function registers (SFR) that starts at address P_1. Certain microcontroller peripheral circuits provide password-protected access.

ST P_1 _ ... .

Substitutes the contents of an external RAM data memory area with the ASCII codes of the characters issued by the user, starting from address P_1. The command is closed when CTRL+P key is pressed.

T P_1 _ P_2 .

The command executes the user program from address P_1 and performs P_2 instructions.

W P_1 _ P_2 .

Transmits a hexadecimal data bloc from the external data memory from address P_1 to address P_2 towards a computer or another development system through a serial interface.

X .

Displays the names and the contents of the user registers of the AT89S8252 microcontroller. The system's console displays the following registers:

R0, R1, R2, R3, R4, R5, R6, R7 (bank 0)
R0, R1, R2, R3, R4, R5, R6, R7 (bank 1)
R0, R1, R2, R3, R4, R5, R6, R7 (bank 2)
R0, R1, R2, R3, R4, R5, R6, R7 (bank 3)
ACC, B, PSW, SP, DPTR0, DPTR1, PC
P0, P1, P2, P3
TCON, TMOD, TH0, TL0, TH1, TL1
T2CON, T2MOD, RCAP2H, RCAP2L, TH2, TL2
WMCON
IE, IP
PCON
SCON, SBUF
SPCR, SPSR, SPDR

XA .

Displays the names and the contents of the data and control registers of the serial asynchronous interface used for UART data transmission and reception, then the names of the control flags and their binary values.

SCON, SBUF
SM0, SM1, SM2, REN, TB8, RB8, TI, RI, SMOD

XB .

Displays the names and the contents of the basic registers of the microcontroller.

ACC, B, PSW, SP, DPTR0, DPTR1, PC

XC .

Displays the name and the contents of the low power consumption mode register, then the names of the control flags and their binary values.

PCON
PD, IDL

XE .

Displays the name and the contents of the command and control register for access to the internal data EPROM memory, then the names of the control flags and their binary values.

WMCON
EEMEN, EEMWE, DPS, RDY/BSY

XF .

Displays the names and the contents of the flag register, then the names of the flags and their binary values.

PSW
C, AC, F0, RS1, RS0, OV, PSW.1, P

XI .

Displays the names and the contents of the registers used for maskable interrupt system validation and for setting the priority level of an interrupt source, then the names of the flags and their binary values.

IE, IP
EA, ET2, ES, ET1, EX1, ET0, EX0, PT2, PS, PT1, PX1, PT0, PX0, IE1, IT1, IE0, IT0

XP .

Displays the names and the contents of the parallel input-output ports on byte and bit levels:

P0, P1, P2, P3
P0.0, P0.1, P0.2, P0.3, P0.4, P0.5, P0.6, P0.7
P1.0, P1.1, P1.2, P1.3, P1.4, P1.5, P1.6, P1.7
P2.0, P2.1, P2.2, P2.3, P2.4, P2.5, P2.6, P2.7
P3.0, P3.1, P3.2, P3.3, P3.4, P3.5, P3.6, P3.7

XS .

Displays the names and the contents of the control, status and data registers of the peripheral serial interface SPI, then the names of the control flags and their binary values.

SPCR, SPSR, SPDR
SPIE, SPE, DORD, MSTR, CPOL, CPMA, SPR1, SPR0, SPIF, WCOL

XT0 .

Displays the names and the contents of the data and control registers of counter T0, then the names of the control flags and their binary values.

TCON, TMOD, TH0, TL0, T0
GATE0, C/T0, M10, M00, TR0, TF0

XT1 .

Displays the names and the contents of the data and control registers of counter T1, then the names of the control flags and their binary values.

TCON, TMOD, TH1, TL1, T1
GATE1, C/T1, M11, M01, TR1, TF1

XT2 .

Displays the names and the contents of the data and control registers of counter T2, then the names of the control flags and their binary values.

T2CON, T2MOD, RCAP2H, RCAP2L, TH2, TL2, RCAP2, T2
TF2, EXF2, RCLK, TCLK, EXEN2, TR2, C/T2, CP/RL2, T2OE, DCEN

XTW .

Displays the name and the contents of the control register of the watchdog timer, then the names of the control flags and their binary values.

WMCON
WDTEN, WDTRST, PS2, PS1, PS0

Y .

Determinates the software initialization of AT89S8252 microcontroller.

IV. CONCLUSIONS

The development system equipped with AT89S8252 has a minimal structure, it was built practically by the author and represents a useful instrument in testing and checking user applications, designed for microcontrollers from ATMEL family.

Using microcontroller systems for these applications offers the following advantages: small volume, high reliability, low power consumption, minimal costs, etc.

The monitor program written by the author for any microcontroller in ATMEL family offers various possibilities and implements very useful features. The author implemented a command set including 52 monitor program commands, the command set being expandable with user-written commands.

The system communicates with a personal computer, which allows editing and assembling user programs. The real-time operation was tested and verified on the microcontroller. The monitor program is stored in an area of some 8 Kb of program memory.

REFERENCES

[1]. Căpăţînă O., Proiectarea cu microcalculatoare integrate, Ed. Dacia, Cluj Napoca, 1992.

[2]. Hintz J.K., Tabak D., Microcontrollers. Arhitecture, Implementation and Programming, McGaw Hill, 1993.

[3]. Lance A. Leventhal, Programmation en langage assembleur, Ed Radio, Paris, 1989.

[4]. Peatmann B.J., Design with Microcontrollers, McGraw Hill, 1998.

[5]. Somnea D., Vlăduţ T., Programarea în Assembler, Ed. Tehnică, Bucureşti, 1992

[6]. XXX ATMEL, Family Microcontroller, Data Book, 1998.

[7]. XXX INTEL, Data Book, 1988.

[8]. XXX CATALYST SEMICONDUCTOR, Data Book, 1989.

[9]. XXX Texas Instruments, Data Book, 1992.