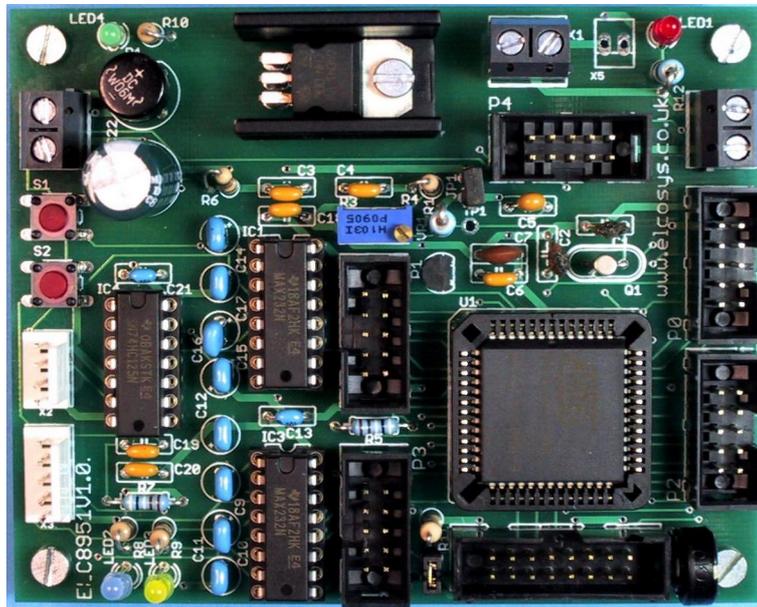


ELC8951

Single Board Computer

HARDWARE REFERENCE MANUAL



For Revision 1.0 Boards
Manual Revision 1.0

ELCOSYS Ltd
ELECTRONIC CONTROL SYSTEMS
145-157 ST JOHN ST LONDON
ENGLAND EC1V 4PW
www.elcosys.co.uk

DISCLAIMER

ELCOSYS has made every attempt to ensure that the information in this document is accurate and complete. However, ELCOSYS assumes no liability for any damages that result from use of this manual or the equipment that it documents. ELCOSYS reserves the right to make changes at any time.

© 2011-2014 by ELCOSYS Ltd. All rights reserved.

ELC8951 Single Board Computer Technical Manual. No part of this work may be reproduced in any manner without written permission of ELCOSYS Ltd. All trademarked names in this manual are the property of respective owners.

WARRANTY

ELCOSYS warrants ELC8951 for a period of 2 years. If the board becomes defective during this period, ELCOSYS will at its option, replace or repair the board. This warranty is voided if the product is subjected to physical abuse or operated outside stated electrical limits. ELCOSYS will not be responsible for damage to any external devices connected to ELC8951. ELCOSYS disclaims all warranties express or implied warranties of merchantability and fitness for a particular purpose. In no event shall ELCOSYS be liable for any indirect, special, incidental or consequential damages in connection with or arising from the use of this product.

1. Overview

ELC8951 is a general purpose, low-cost and highly-expandable micro-controller system. It is based on the ATMEL AT89C51AC3 single-chip Flash micro-controller. The board features:

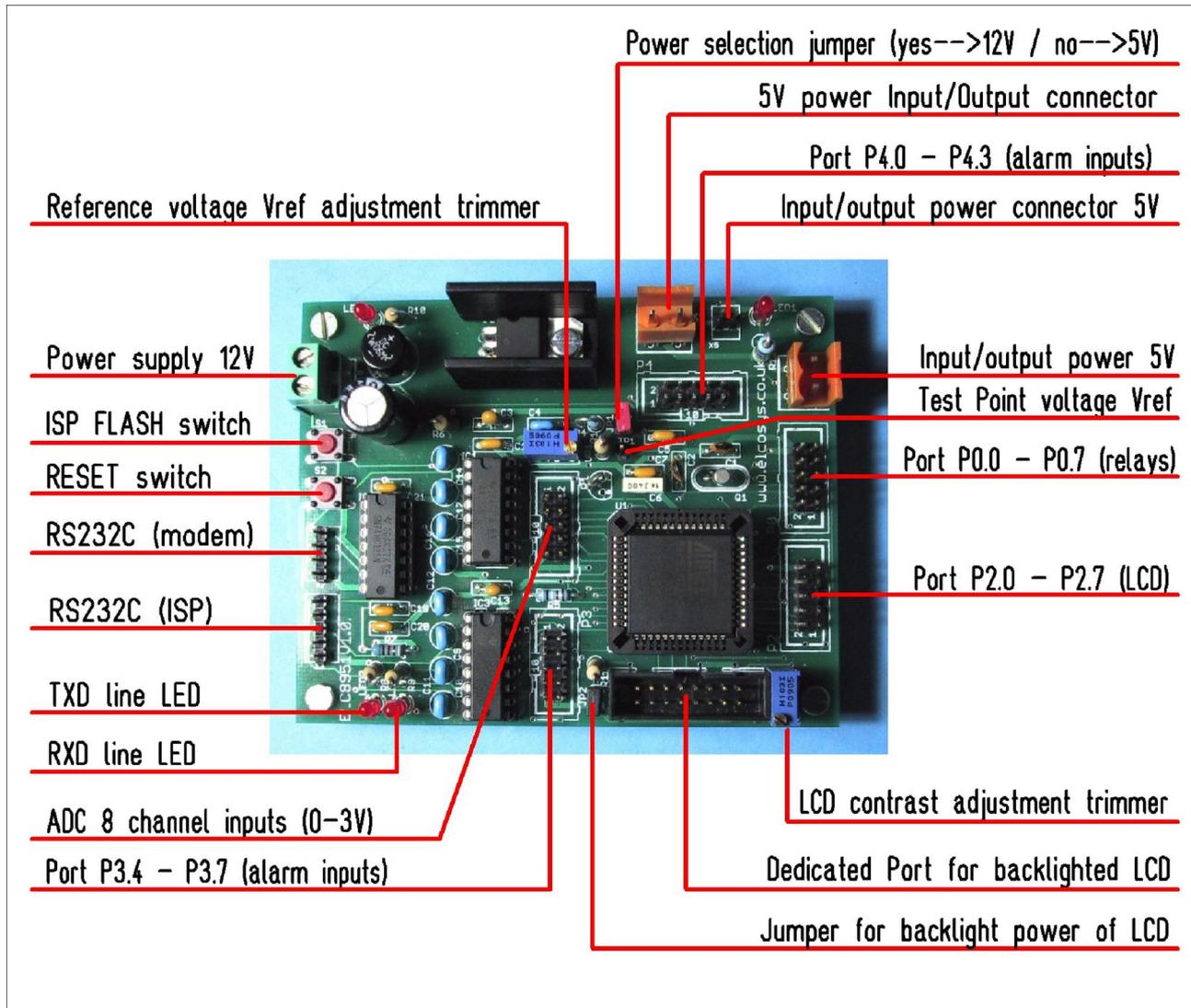
- Main clock frequency 29.4912 MHz
- 64 Kilobytes of In-System Re-programmable Downloadable Flash Memory
- 256 bytes of IRAM
- 2048 bytes of ERAM
- 2048 bytes Serial EEPROM
- 9-sources 4-level Interrupts
- Programmable Counter Array with
 - High Speed Output
 - Compare/Capture
 - Pulse Width Modulator
- Three 16 bit Timer/Counters
- Programmable Enhanced UART Serial Channel
- SPI Serial Interface
- Programmable Watchdog Timer
- 5 I/O ports 8 bit
- 40 general purpose I/O pins
- RS232 Serial Port connector for In-circuit Programming and for data communications
- 8-channel 10-bit ADC with 3.000 V on-board voltage reference source
- Dual-row 14-pin LCD connector (with trimmer contrast adjustment for LCD)
- 5 x 10-pin IDC connectors for peripheral boards and devices
- In-circuit Programming of the main Flash micro-controller through the RS232 Serial Port
- On-board 5 Volt linear regulator
- 12 V presence LED, 5V presence LED, RS232 TXD and RXD activity LEDs
- 12 VDC power supply (6 to 12VDC power sources can be used) or 5VDC power supply
- Dimensions of 8.0 X 10.0 centimeters.
- Mounting holes of 3.5 millimeters are on four corners.
- 0° - 70° C operating, -40° - +85° C storage temperature range.

The Flash micro-controller can be serially programmed while in the target application circuit. Customers can program the micro-controller with the most recent firmware or custom firmware. This function of the FLASH micro-controller simplifies new program development and debugging. Downloading of a program to the micro-controller typically takes few seconds.

[BASCOS51](#), SDCC development systems based on [Micro-IDE](#) Integrated Development Environment from ELCOSYS, fully support in-system programming and debugging on the ELC8951 board using the serial port. There is also the [Reads51 IDE](#) available on line.

A Windows-based program [ATMEL FLIP](#) is also provided to easily download programs to the board.

3. Board Layout and connectors



Board connectors and resources:

- 1 screw connector for 7 to 12 V power supply input (X4).
- 1 screw connector for 5VDC power supply input/output (X1)
- 1 screw connector for 5VDC power supply input/output (X6)
- 1 2-pin connector for 5VDC power supply input (X5)
- 1 power supply selection jumper, closed=7.....12VDC, opened=5V (JP1)
- 1 RESET switch (S2) for general hardware Micro-controller reset
- 1 PSEN switch (S1) for manual enabling of Flash programming mode.
- 1 RS232C 4 pin Molex connector for general communication purpose (X2).
- 1 RS232C ISP 5 pin Molex connector for downloading software auto-mode (X3).
- 1 blue LED for monitoring ELC8951 TXD line of RS232 port (LED2).
- 1 yellow LED for monitoring ELC8951 RXD line of RS232 port (LED3).
- 1 trimmer for reference voltage regulation to 3.00 V (R3).
- 1 Test Point for reference voltage measurement (TP1).
- 1 trimmer for LCD contrast adjustment (R2).
- 5 10-pin IDC connectors reported to Micro-controller I/O ports (P0,P1,P2,P3,P4)
- 1 16-pin IDC connector reserved for LED back-lighted LCD module.

4. I/O Parallel Ports pin-out and functional description

In the following notes are described the functions performed by each line of the five 8-bit parallel ports of the Micro-controller referred to the IDC connectors pin-out. When a line is multifunction, any different feature is separated by comma.

Port 0 – IDC connector P0

Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in this state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pull-ups when emitting 1's. Port 0 also outputs the code Bytes during program validation. External pull-ups are required during program verification.

Pin 1 = P0.0 open drain bi-directional I/O line
Pin 2 = P0.1 open drain bi-directional I/O line
Pin 3 = P0.2 open drain bi-directional I/O line
Pin 4 = P0.3 open drain bi-directional I/O line
Pin 5 = P0.4 open drain bi-directional I/O line
Pin 6 = P0.5 open drain bi-directional I/O line
Pin 7 = P0.6 open drain bi-directional I/O line
Pin 8 = P0.7 open drain bi-directional I/O line
Pin 9 = +5V reference line
Pin 10 = GND reference line

Port 1 – IDC connector P1

Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. Port 1 pins can be used for digital input/output or as analog inputs for the Analog Digital Converter (ADC) in the range 0-3V with on board voltage reference. Port 1 pins that have 1's written to them are pulled high by the internal pull-up transistors and can be used as inputs in this state. As inputs, Port 1 pins that are being pulled low externally will be the source of current(IIL, see section "Electrical Characteristic") because of the internal pull-ups. Port 1 pins are assigned to be used as analog inputs via the ADCCF register (in this case the internal pull-ups are disconnected). As a secondary digital function, port 1 contains the Timer 2 external trigger and clock input; the PCA external clock input and the PCA module I/O. Port 1 receives the low-order address byte during EPROM programming and program verification. It can drive CMOS inputs without external pull-ups.

Pin 1 = P1.0 AN0: Analog input channel 0, External clock input for Timer/counter2.
Pin 2 = P1.1 AN1: Analog input channel 1, Trigger input for Timer/counter2.
Pin 3 = P1.2 AN2: Analog input channel 2, PCA external clock input.
Pin 4 = P1.3 AN3: Analog input channel 3, PCA module 0 Entry of input/PWM output.
Pin 5 = P1.4 AN4: Analog input channel 4, PCA module 1 Entry of input/PWM output.
Pin 6 = P1.5 AN5: Analog input channel 5, PCA module 2 Entry of input/PWM output.
Pin 7 = P1.6 AN6: Analog input channel 6, PCA module 3 Entry of input/PWM output.
Pin 8 = P1.7 AN7: Analog input channel 7, PCA module 4 Entry of input/PWM output.
Pin 9 = +5V reference line
Pin 10 = GND reference line

Port 2 – IDC connector P2

Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. Port 2 pins that have 1's written to them are pulled high by the internal pull-ups and can be used as inputs in this state. As inputs, Port 2 pins that are being pulled low externally will be a source of current (IIL, see section "Electrical Characteristic") because of the internal pull-ups. Port 2 emits the high-order address byte during accesses to the external Program Memory and during accesses to external Data Memory that uses 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1's. During accesses to external Data Memory that use 8 bit addresses (MOVX @Ri), Port 2 transmits the contents of the P2 special function register. It also receives high-order addresses and control signals during program validation. It can drive CMOS inputs without external pull-ups.

Pin 1 = P2.0 bi-directional I/O line with internal pull-up
Pin 2 = P2.1 bi-directional I/O line with internal pull-up (shared with LCD port)
Pin 3 = P2.2 bi-directional I/O line with internal pull-up (shared with LCD port)
Pin 4 = P2.3 bi-directional I/O line with internal pull-up (shared with LCD port)
Pin 5 = P2.4 bi-directional I/O line with internal pull-up (shared with LCD port)
Pin 6 = P2.5 bi-directional I/O line with internal pull-up (shared with LCD port)
Pin 7 = P2.6 bi-directional I/O line with internal pull-up (shared with LCD port)
Pin 8 = P2.7 bi-directional I/O line with internal pull-up (shared with LCD port)
Pin 9 = +5V reference line
Pin 10 = GND reference line

Port 3 – IDC connector P3

Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. Port 3 pins that have 1's written to them are pulled high by the internal pull-up transistors and can be used as inputs in this state. As inputs, Port 3 pins that are being pulled low externally will be a source of current (IIL, see section "Electrical Characteristic") because of the internal pull-ups. The output latch corresponding to a secondary function must be programmed to one for that function to operate (except for TxD and WR).

Pin 1 = P3.0 RxD: Receiver data input, data input/output (synchronous) of the serial interface.
Pin 2 = P3.1 TxD: Transmitter data output, clock output (synchronous) of the serial interface.
Pin 3 = P3.2 INT0: External interrupt 0 input, timer 0 gate control input
Pin 4 = P3.3 INT1: External interrupt 1 input, timer 1 gate control input
Pin 5 = P3.4 T0: Timer 0 counter input
Pin 6 = P3.5 T1: Timer 1 counter input, SPI Slave Select
Pin 7 = P3.6 WR: Ext. Data Mem. write strobe; latches data byte from port 0 into the ext. memory
Pin 8 = P3.7 RD: External Data Memory read strobe; Enables the external data memory.
Pin 9 = +5V reference line
Pin 10 = GND reference line

Port 4 – IDC connector P4

Port 4 is an 2-bit bi-directional I/O port with internal pull-ups. Port 4 pins that have 1's written to them are pulled high by the internal pull-ups and can be used as inputs in this state. As inputs, Port 4 pins that are being pulled low externally will be a source of current (IIL, on the datasheet) because of the internal pull-up transistor. It can drive CMOS inputs without external pull-ups.

Pin 1 = P4.0 bi-directional I/O line with internal pull-up
Pin 2 = P4.1 bi-directional I/O line with internal pull-up
Pin 3 = P4.2 MISO: Master Input Slave Output of SPI controller
Pin 4 = P4.3 SCK: Serial Clock of SPI controller
Pin 5 = P4.4 MOSI: Master Output Slave Input of SPI controller
Pin 6 = Not used
Pin 7 = Not used
Pin 8 = Not used
Pin 9 = +5V reference line
Pin 10 = GND reference line

LCD Port – IDC connector SV2

The dedicated LCD Port is derived from Port P2 and is connected as industry standard Hitachi HD44780 controller or equivalent controller, in 4-bit mode, connected as follows (the symbols refer to LCD data sheet):

Pin 1 = GND
Pin 2 = +5V
Pin 3 = VO (contrast adjustment, connected to on-board trimmer))
Pin 4 = P2.1 => RS
Pin 5 = P2.2 => RW
Pin 6 = P2.3 => EN
Pin 7 = D0 => GND (4-bit mode)
Pin 8 = D1 => GND (4-bit mode)
Pin 9 = D2 => GND (4-bit mode)
Pin 10 = D3 => GND (4-bit mode)
Pin 11 = P2.4 => D4
Pin 12 = P2.5 => D5
Pin 13 = P2.6 => D6
Pin 14 = P2.7 => D7
Pin 15 = A (backlight LED supply)
Pin 16 = GND (backlight LED supply)

RS232C Port – Molex connector X2

The 4-pin Molex connector is connected to on-chip UART peripheral through an industry standard RS232 driver/receiver. Two LEDs are provided to indicate activity on Rx and Tx lines of Micro-controller UART. When a PC is connected to this serial port, to enter in ISP mode, press both the RESET (S2) and ISP (S1) buttons simultaneously; first release the RESET button and then the ISP button and the device enter in ISP mode and is possible to download HEX files from PC to board.

Pin 1 = +5V
Pin 2 = TXD (referring to PC RS232 port acting as DTE with ELC8951 as DCE)
Pin 3 = RXD (referring to PC RS232 port acting as DTE with ELC8951 as DCE)
Pin 4 = GND

ISP Port – Molex connector X3

The AT89C51AC3 Micro-controller features UART boot-loader and Auto ISP capability and it allows the Host PC application (Example [ATMEL Flip Software](#)) to control the hardware conditions from serial lines RTS and DTR and to enter in ISP mode automatically,so, using the ISP Port, the user does not need to push the ISP and RESET buttons.

Pin 1 = RTS

Pin 2 = RXD (referring to PC RS232 port acting as DTE with ELC8951 as DCE)

Pin 3 = TXD (referring to PC RS232 port acting as DTE with ELC8951 as DCE)

Pin 4 = DTR

Pin 5 = GND

Power Supply – screw connectors X4 and X1

X4 = 7 to 12 V DC/AC (with jumper JP1 closed)

X1 = 5VDC (with jumper JP1 open)

Jumper and test point table

Reference	function	comments	default
JP1	Supply voltage selection	ON: 7 – 12 V AC/DC supply OFF: 5VDC supply	closed (ON)
JP2	LCD backlight switch	ON: LCD backlight activated OFF: LCD backlight disabled	Closed (ON)
TP1	Reference Voltage test	Adjustable by trimmer R3	3.00 V

DOWNLOAD HEX Files to ELC8951 board

We use program "FLIP" from ATMEL to download Hex file into Flash Memory of MCU on Board, it connects with MCU through Serial Port of computer PC. The User can download Program FLIP from [ATMEL Flip Software](#) free without any charge and in case of user purchases board from ELCOSYS Ltd, Program FLIP is provided in CD-ROM. **Program FLIP (Flexible In-system Programmer)** which is a program for development Micro-controller system of ATMEL can support ISP development of Micro-controller MCS51 including **AT89C51AC3**. It runs under Windows9X/Me/NT/2000 and Windows XP and supports the connection with Hardware system as RS232 type or CAN type or USB type.

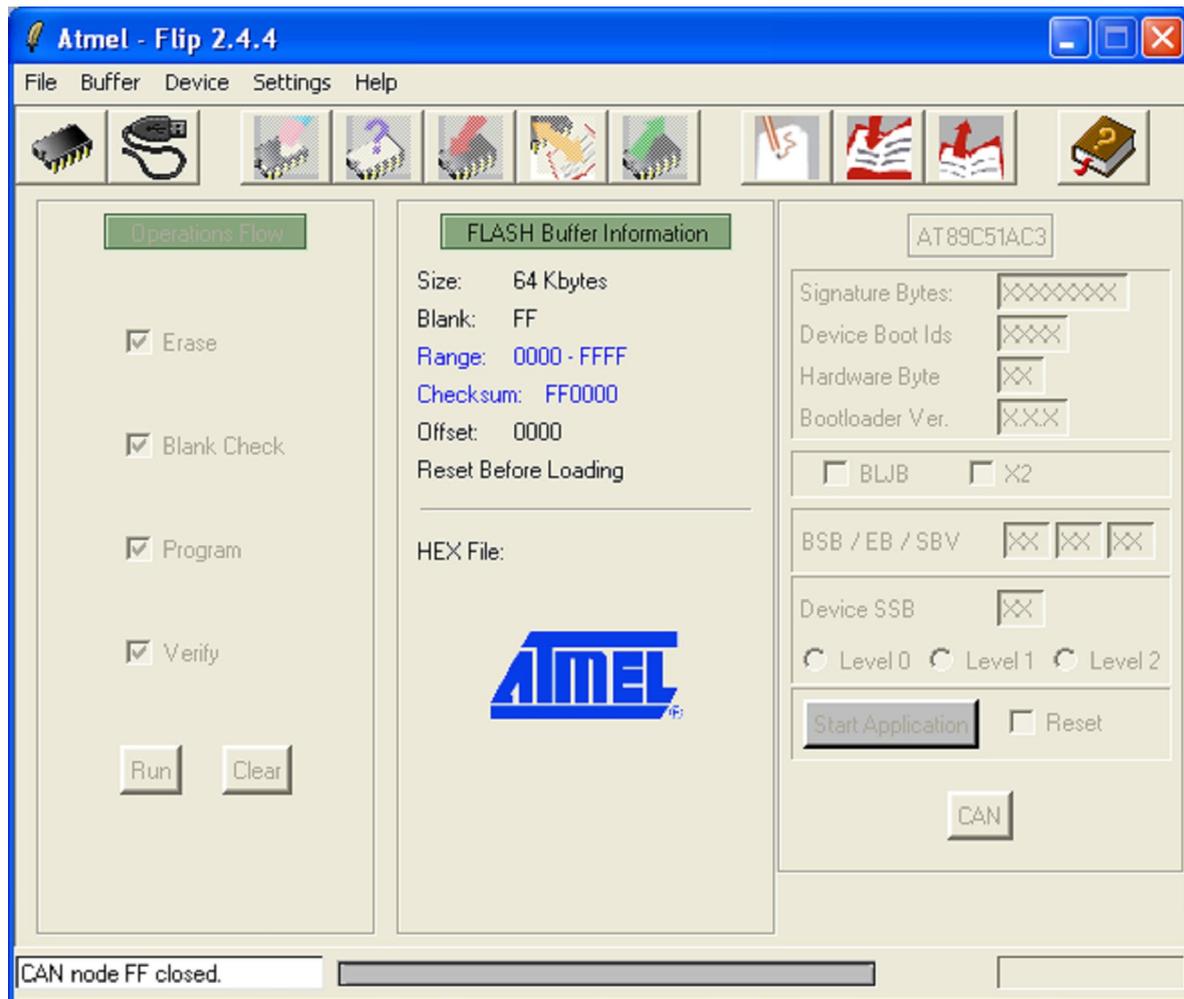
The method to connect Program FLIP and Hardware system of Micro-controller depends on the ability of the communication of Micro-controller. In this case, Micro-controller **AT89C51AC3** can be connected through RS232 Serial Port only. Program FLIP downloads data into memory of Micro-controller that runs in Monitor Mode, so we can Erase, Blank Check, Program, Verify and Read memory internal CPU.

When we want Program FLIP connect with CPU in Monitor Mode, we must RESET CPU to run in Monitor mode first and then we can use CPU through Program FLIP. We must configure Pin PSEN to be Logic "0", when CPU is not in status RESET. Generally, CPU will check status of Pin PSEN to be Logic "0" or not, if it is not, it will run in Normal Mode; if it is Logic "0", it will check status of other signals that are related to operation in Monitor Mode and if there's no any error, it will run in Monitor Mode instantly. There are 2 methods to RESET CPU ATMEL of Board ELC8951:

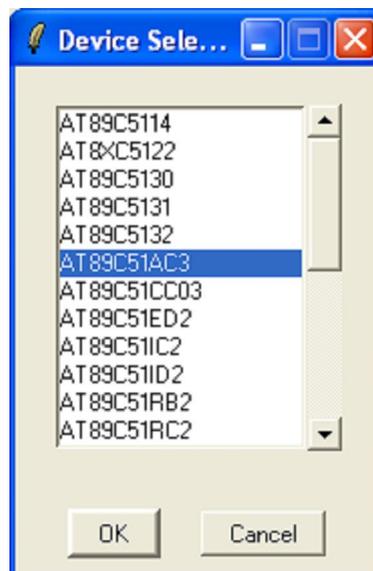
- **Download as Manual Type:** We must use cable RS232 4 Pin with Switch PSEN and Switch RESET for downloading.
- **Download as Auto Type:** We must use cable ET-DOWNLOAD 5 Pin for downloading.

DOWNLOAD HEX Files in manual Mode

1. Interface RS232 cable from Com Port of Computer PC with Connector RS232 4 Pin of Board.
2. Supply power into circuit of Board and we can see red LED of PWR is in status ON.
3. Run Program FLIP V2.4.4 and it will display windows shown the picture below



4. Select truly MCU number that is installed in Board, click **Device** → **Select**, for example, we select MCU number as (AT89C51AC3).



Picture displays **the selection CPU No. for Board ELC8951 (AT89C51AC3)**.

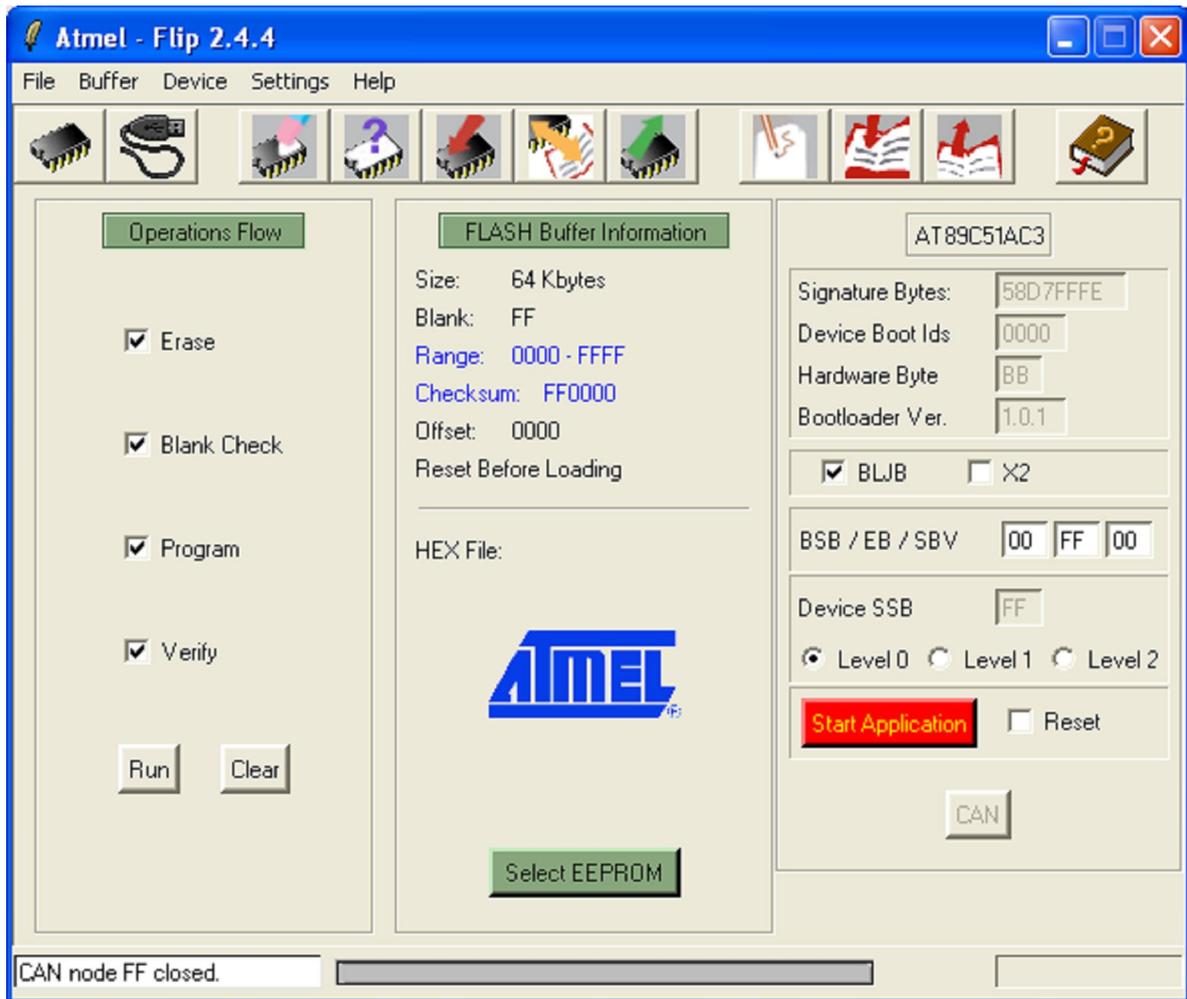
5. Click **Setting** → **Communication** → **RS232**, and then configure truly COM PORT that is connected with cable, in the sample is Com1.



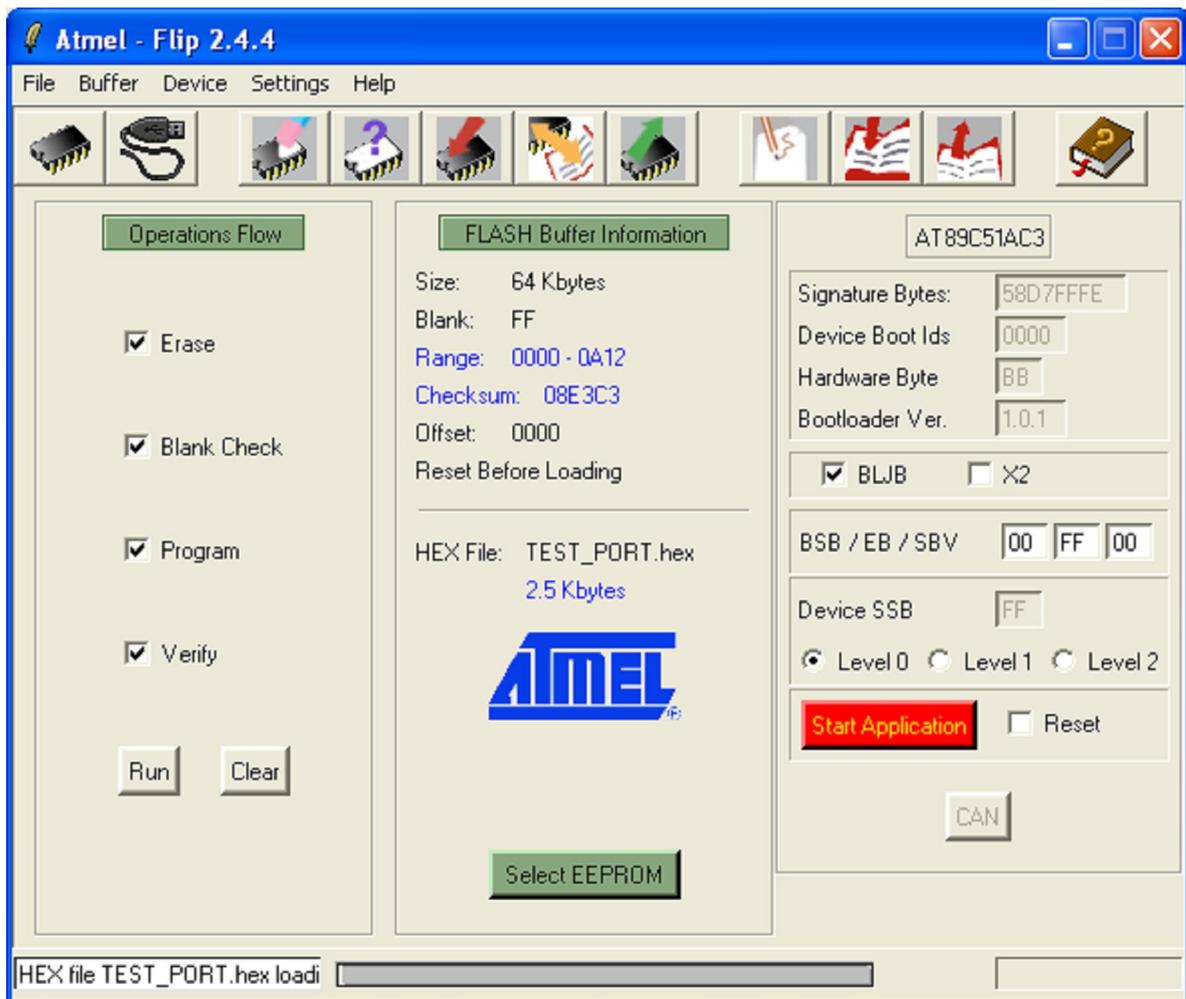
6. Reset MCU for running in Monitor as following;

- a) Press Switch PSEN for a while until status of Pin PSEN is Logic "0".
- b) Press Switch RESET to send signal RESET into CPU but Switch PSEN remains pressing.
- c) Let Switch RESET, it makes CPU is not in status RESET but Switch PSEN remains pressing.
- d) Finally, let Switch PSEN.

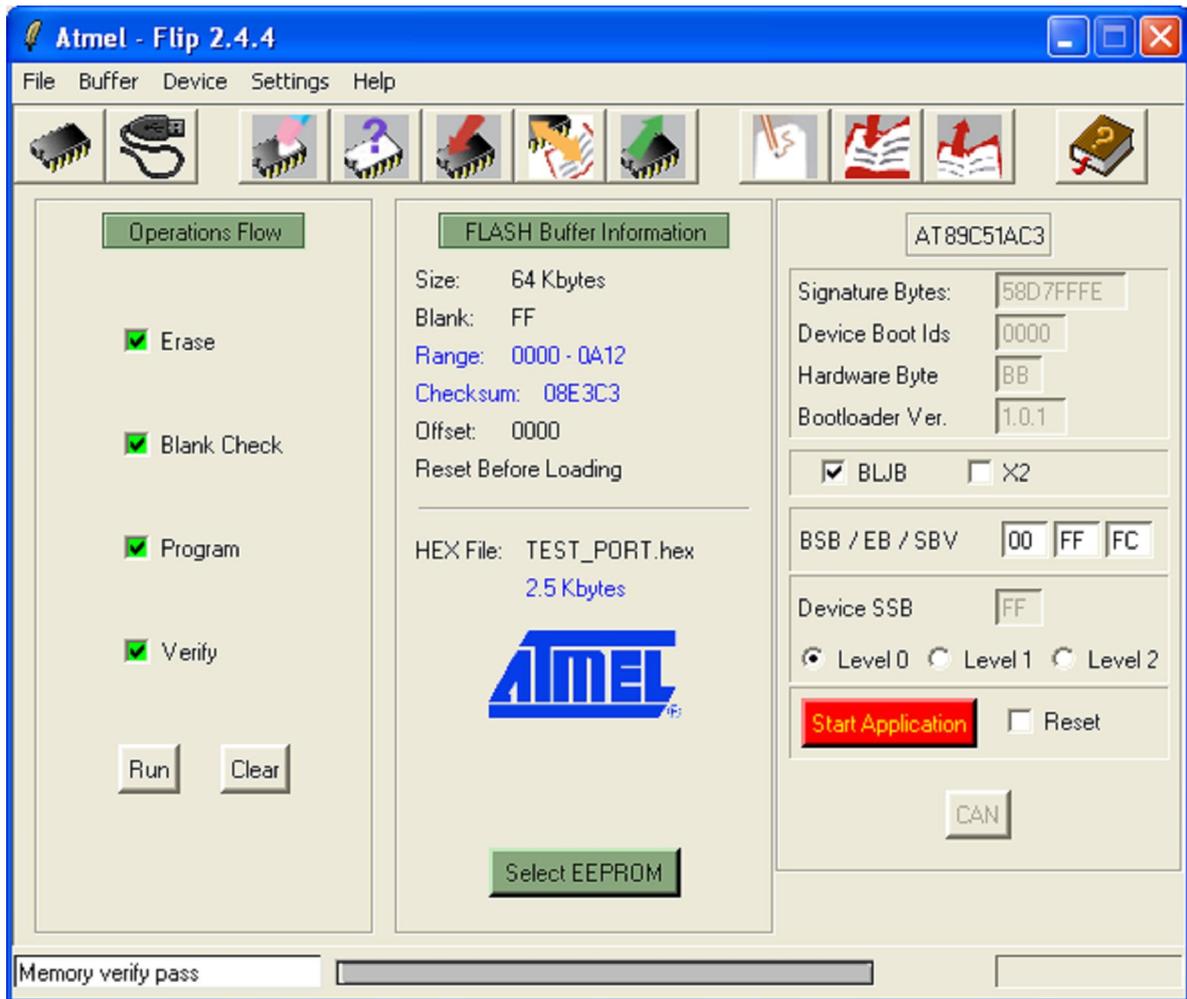
7. Click **Connect** button to communicate with MCU in Monitor Mode and it will display result as shown the picture below.



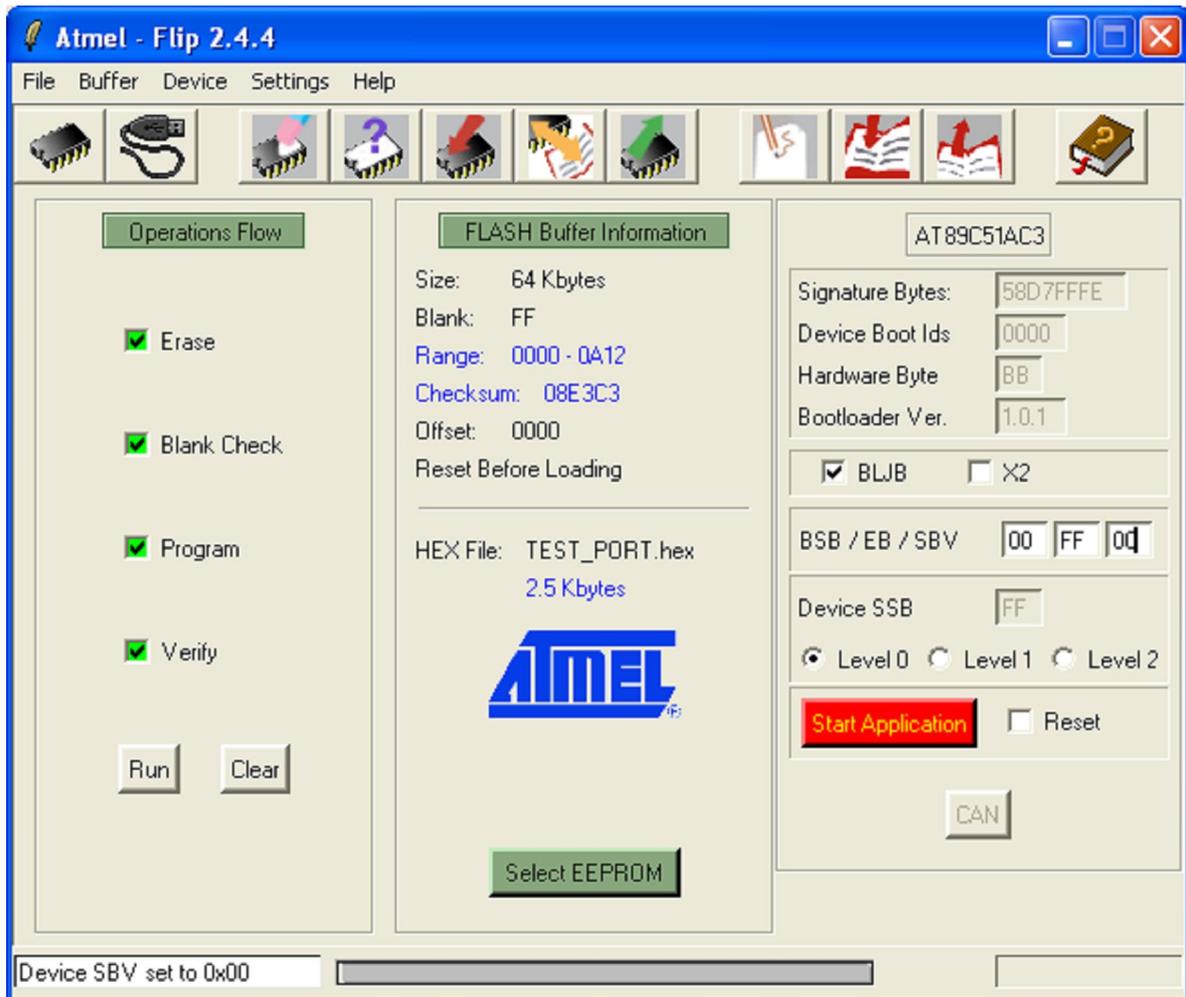
8. Open Hex File that we want to download into MCU and configure it to wait for in Buffer of Program FLIP, click “**File** → **Load Hex File...**”.



9. Click all commands in Tab of Operation Flow such as **Erase**, **Bank Check**, **Program**, and **Verify**. Then click **Run** button and wait for operation of program is finished as in the picture.



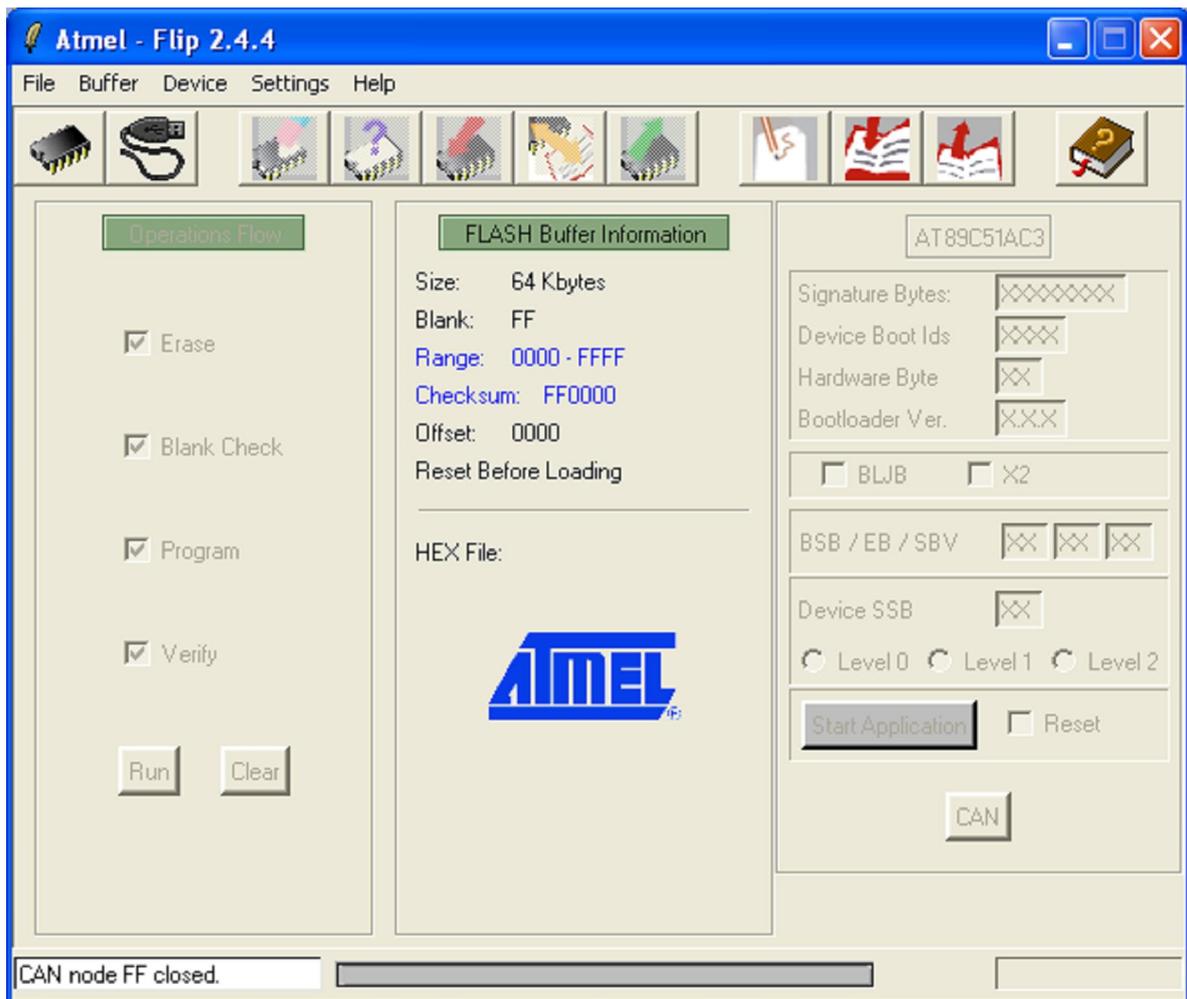
10. Check default value of Device BSB and SBV are '00' or not. If it is not, click in both blank numbers and then type 00 as in the picture.



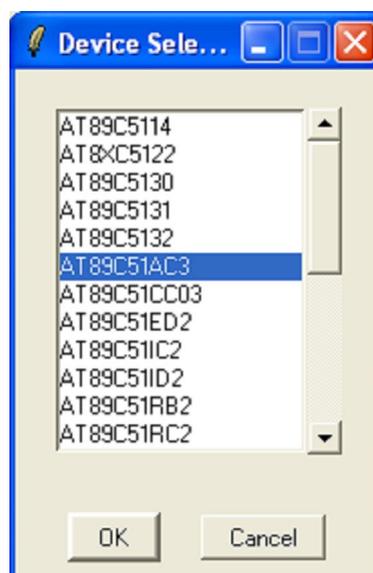
11. Press Switch RESET into Board and Board starts running followed the downloaded program. In case of there's no any mistake, it will start running instantly.

DOWNLOAD HEX Files in Auto ISP Mode

1. Interface RS232 cable from Com Port of Computer PC with Connector ET-DOWNLOAD 5 Pin of Board.
2. Supply power into circuit of Board and we can see red LED of PWR is in status ON.
3. Run Program FLIP V2.4.6 and it will display window as shown in the picture below.

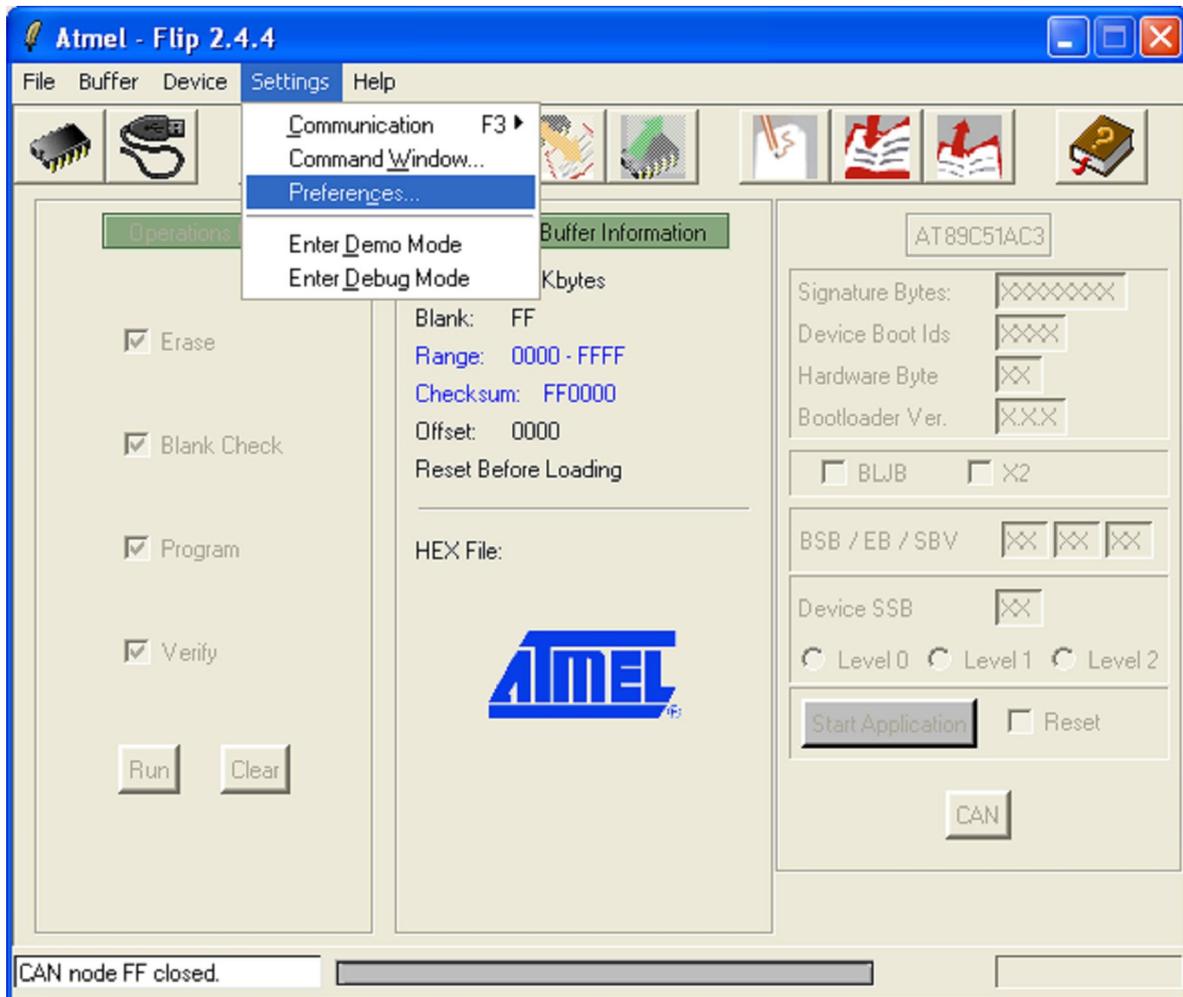


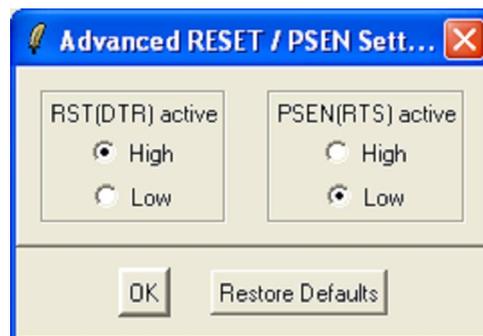
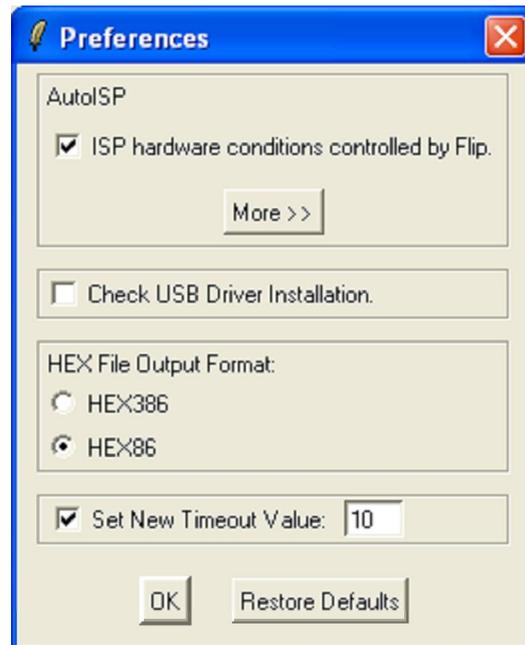
4. Select MCU chip that is installed in Board, click **Device** → **Select**, for example, we select MCU number as (AT89C51AC3)



Picture displays the selection CPU No. for Board ELC8951: ATMEL AT89C51AC3.

5. Configure Option of RS232 Communication for using as Auto Download type. Click **Setting** → **Preferences...** and then configure default values as shown in the picture below.

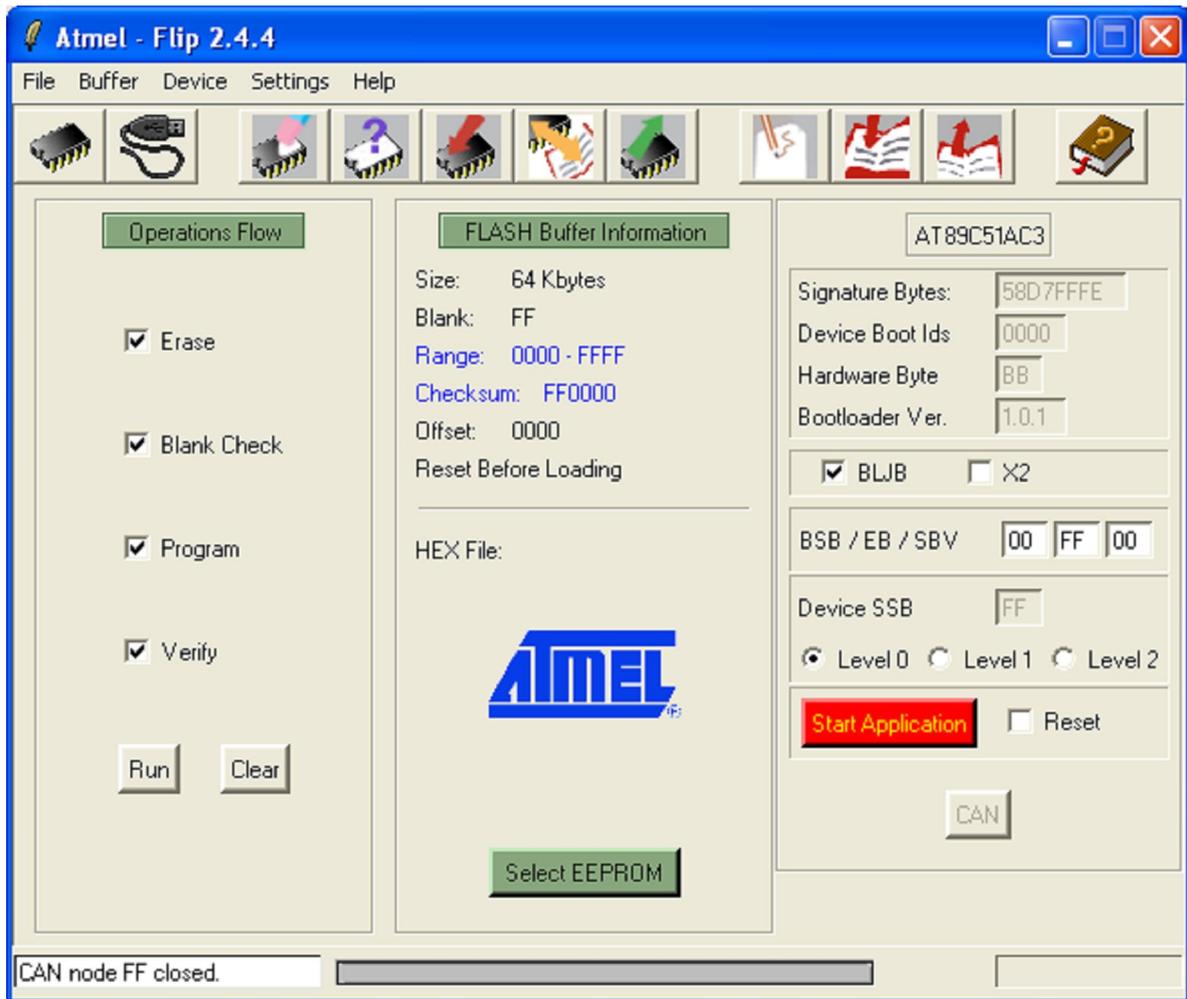




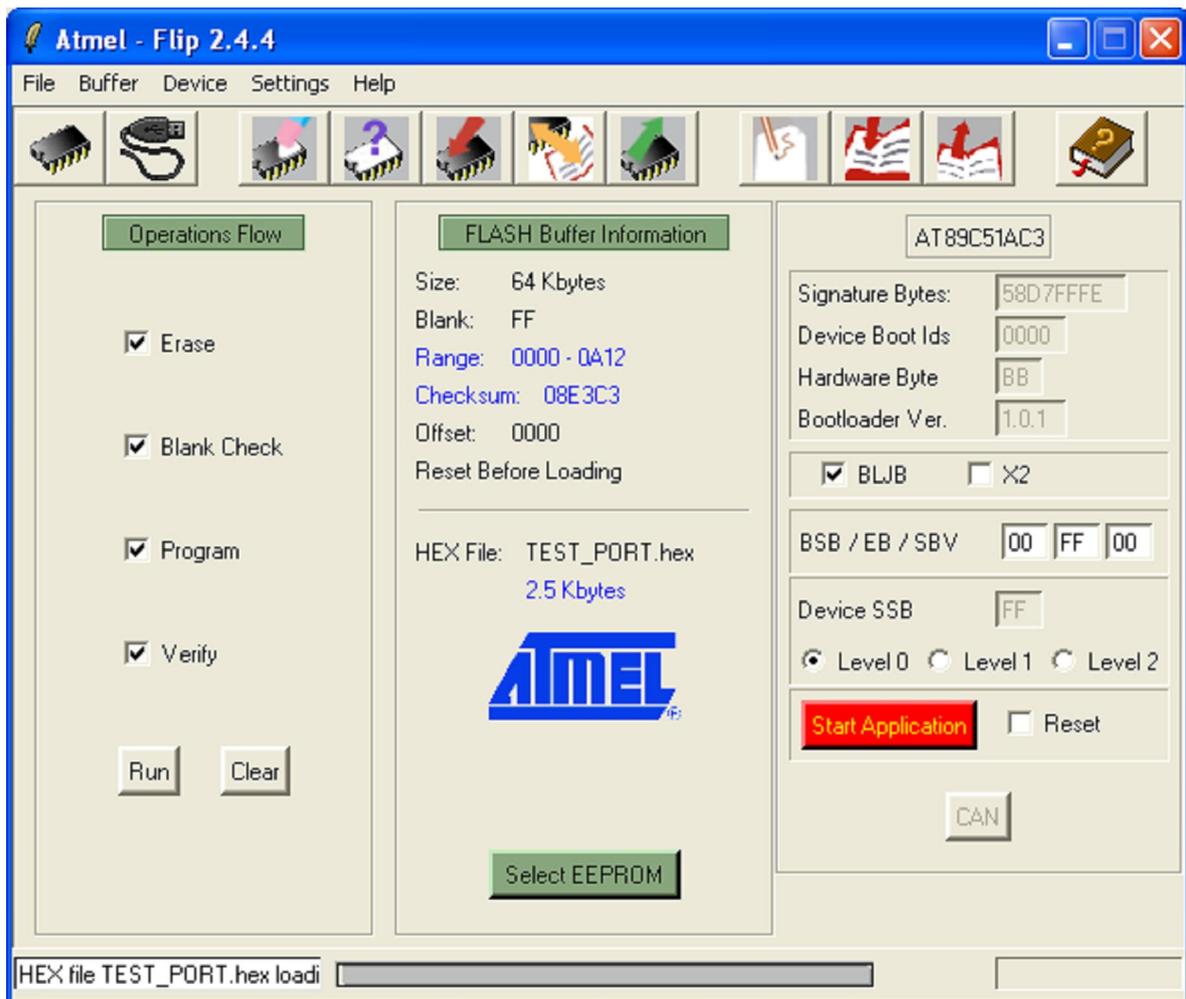
6. Click **Setting** → **Communication** → **RS232**, and then configure truly Comport that is connected with cable, in the sample is COM1.



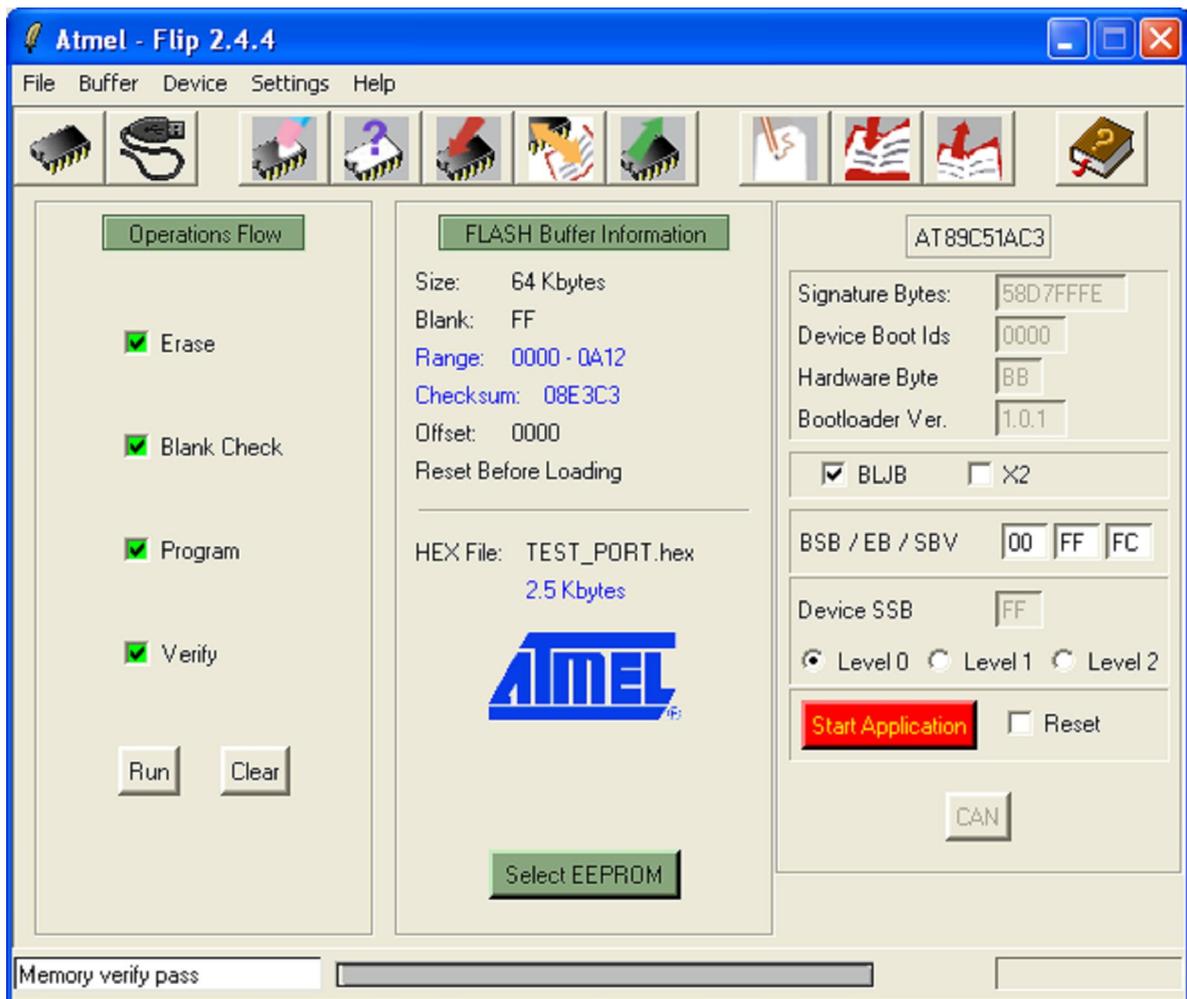
7. Click **Connect** button to communicate with MCU in Monitor Mode and it will display result as the picture below.



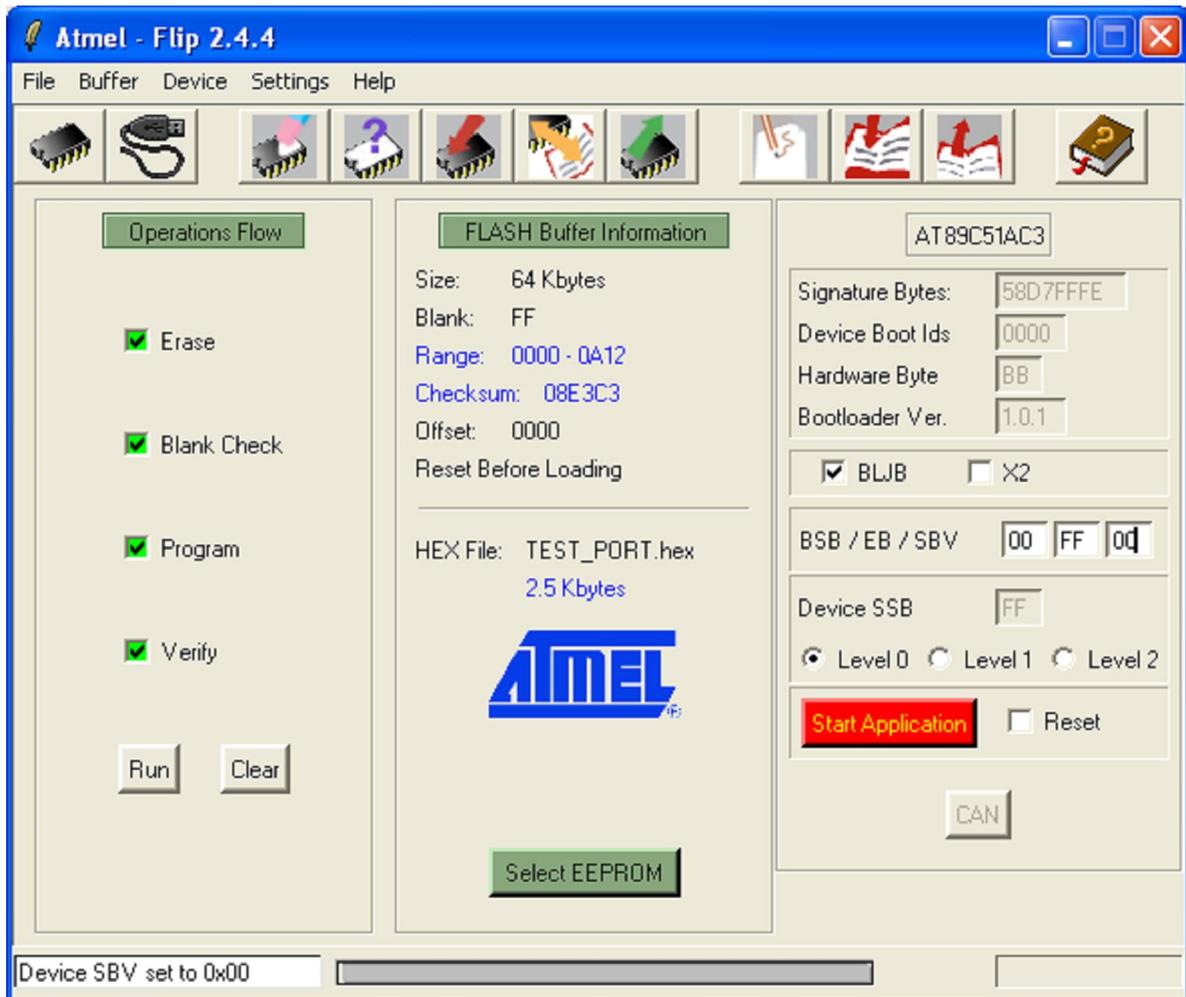
8. Open Hex File that we want to download into MCU and configure it to wait for in Buffer of Program FLIP, click "File → Load Hex File...".



9. Click all commands in Tab of Operation Flow such as **Erase**, **Blank Check**, **Program**, and **Verify**. Then click **Run** button and wait for operation of program is finished as in the picture.



10. Check default value of Device BSB and SBV are '00' or not. If it is not, click in both blank numbers and then type 00 as in the picture.



11. Click **“Start Application”** or Press Switch RESET into Board and Board starts running followed by the downloaded program. In case of there’s no any mistake, it will start running instantly

Problems and Solutions

When we use commands of Program FLIP, some error maybe occur. It maybe occurs because of Hardware system, it maybe occurs because of some configuration of Parameter in program incorrect. So, when Program FLIP can not run, it will display error as following;

1. **Time Out Error:** It is error because program FLIP cannot communicate with CPU in Monitor Mode. This problem may be occurring because of these causes below;

- Interface cable between Connector RS232 of Board with Connector RS232 Serial Port Communication of computer incomplete; or not interface cable as configured in program; or configure type and default values incorrect. So, we must click **“Setting → Preferences.. and Setting → Communication → RS232”**

- In case of downloading as Manual type, user does not RESET CPU to run in Monitor Mode and not configures it to wait for in Buffer but user commands it to run program instantly; or board not ready to run such as not supply power into board yet.

- Configure Baud Rate too quickly. In case of using Board with high speed computer, should configure Baud rate is slower. It may be 19200 or 9600 because program FLIP sends data into CPU continuously but CPU can not collect data all.

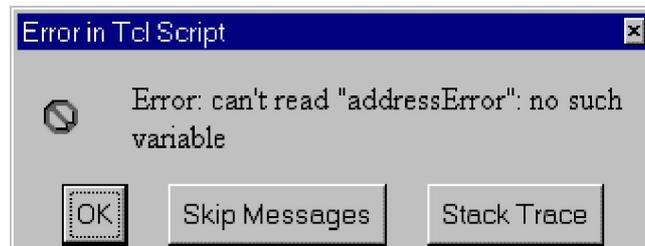
2. **Software Security Bit Set. Cannot access device Data:** The error occurs because of using CPU that is in status Lock Bit of Security Bit without modify or erase the old data. So, before we use CPU to Program, Verify or Read, must erase the old data first and then we can rewrite the new one.

3. **The board reply is not correct:** This error occurs because of communication between Program FLIP and Micro-controller incorrect as Framing Error. User maybe configures Baud Rate for using with Crystal Frequency on board incorrect.

4. **The RS232 port could not be opened:** This error occurs because of Program FLIP can not open operation of RS232 Serial Port Communication of computer PC. User select Comport No. incorrect; or other program remains using the Comport No. So, close program FLIP and open the program again. If the error remains occur, must check other related causes and modify it.

5. **Check sum error:** This error occurs because of CPU receives data from computer PC incompletely. User may be configure Baud Rate too quickly; or configure Baud Rate to be unsuitable for Crystal Frequency, so user can configure Baud Rate to be slower such as 9600, 19200, and 38400. If speed of computer is not higher, maybe configure as 57600 or 115200.

6. **Can not Load HEX:** This error occurs because of Program FLIP can not read data in Hex File. Remember, Program FLIP can use with Intel Hex File only, so other Files can not be used with this Program. This problem may be occurs because of Program FLIP can not read Hex File although it is Intel Hex File. It is translated by Program Assembler of SXA51.EXE. There's 1 blank line at the beginning line of program but the first line of Hex File must begin with colon sign (:) and follows by data. It will display error as shown in the picture below;



So, we must use program Text Editor to open Hex File that is translated by SXA51.EXE then delete the 1 blank line at the beginning line and save the new one. After that we can use the Hex File without any error

```
:1000000758921759850D2BCC2ACC2A9C2AB74FB31
:10001000F58DF58BD28CD28ED2AF1200880D0A0AE4
:1000200044454D4F20544553542043502D4A52359A
:100030003141433220563120262056322028313299
:100040002D436C6F636B290D0A50726573732041E9
:00000001FF
```

Picture displays feature of HEX File is interpreted by SXA51 and there's 1 blank line.

```
:1000000758921759850D2BCC2ACC2A9C2AB74FB31  
:10001000F58DF58BD28CD28ED2AF1200880D0A0AE4  
:1000200044454D4F20544553542043502D4A52359A  
:100030003141433220563120262056322028313299  
:100040002D436C6F636B290D0A50726573732041E9  
:00000001FF
```

Picture displays **feature of Hex File that can be used with Program FLIP after deleted the blank line.**

7. When we commanded to program data into CPU completely, and RESET Board to run but it does not. This problem maybe occurs as the situation below;

- The written program is not correct, so user must check and correct them by self.
- User does not Load Hex File and not configures it to wait for in Buffer but user runs Program Device instantly. Remember, Program FLIP uses data in Buffer to download into memory of program.
- Switch PSEN remains pressing. When we reset board, CPU always run in Monitor mode. So, we must check and configure Logic of Pin 45 (PLCC-52) must be "1", in case of not press Switch PSEN and it must be Logic "0", in case of Switch PSEN remains pressing.
- Switch RESET may be remain pressing. It makes CPU is in status RESET. So, we must check Logic Pin 1 (PLCC 52) Pin must be "0", in case of not press Switch RESET and it must be Logic "1", in case of Switch RESET remains pressing.
- We do not configure both Device BSB and SBV to be 00H, it makes program jump to run in other position that is configured by Device SBV. If default value of Device SBV is not '00', program is not run. So, after programming data into CPU completely, should configure default value of Device BSB and Device SBV to be 00H.