



Application Notice Title

AP180 Development Kit User's Guide

Revision History

Rev.	History	Issue Date	Remark
1.0	Initial issue	April 27, 2004	Preliminary
1.1	Added JTAG ICE example description	July 27, 2004	
1.2	Added C language support	November 10, 2004	

PRELIMINARY

Kit Contents

The AP180 Development Kit contains the following items:

- AP180 Target Board
- AP180 Serial Adaptor (RS-232 to Target Board Protocol Converter)
- AP180 IDE and Product Information CD-ROM. CD content includes:
 - Installation utility to install the IDE (AP180IDE.EXE)
 - Documentation
- Ribbon Cable for Kit and Target Board interface

The following 2 items are not supported, the SPEC. are as follow:

- RS232 Serial Cable (Male to Male, COM Port on the Serial Adaptor, PIN 2: TXD to PC, PIN 3: RXD from PC, PIN 5: GND)
- AC to DC Power Adapter (Voltage range: 6.0V~4.5V; the inner polarity of Power jack is positive)

Hardware Setup

The target board is connected to a PC running the AP180 IDE via the Serial Adapter as shown in Figure1.

1. Connect one end of the RS232 serial cable to a serial (COM) port on the PC.
2. Connect the other end of the RS232 serial cable to the port on the Serial Adapter.
3. Connect the Serial Adapter to the JTAG port on the target board using the 10-pin ribbon cable.
4. Connect the AC/DC power adapter to power jack JP1 on the target board.

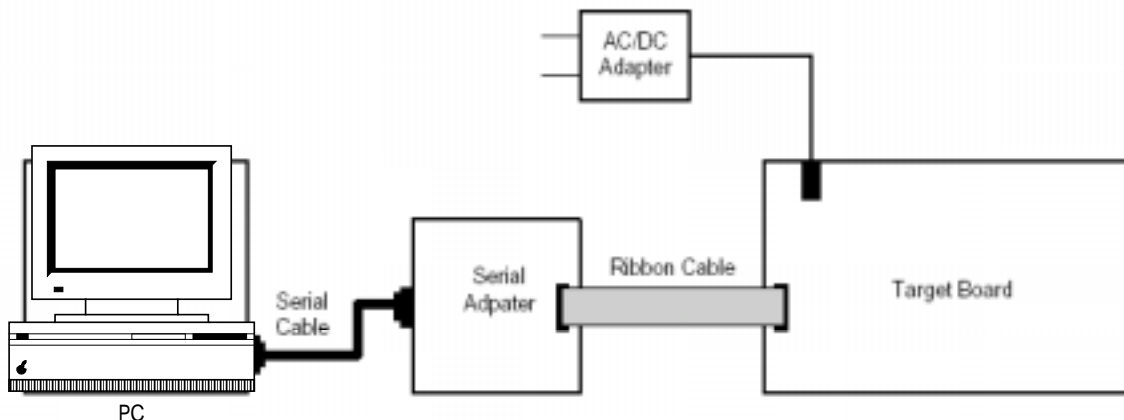


Figure1. Hardware Setup

Software Setup

The included CD-ROM contains the AP180 Integrated Development Environment (IDE) and additional documentation. To install the DK software is to copy the file "AP180IDE.EXE" found of the CD-ROM to the PC.

The AP180 IDE requires:

- Pentium-class host PC running Microsoft Windows 95/98, Windows NT/2000/XP
- One available RS232 COM port

The AP180 IDE restrictions:

- Can not support Power Saving Mode and Watch Dog Timer

Target Board

The target board provides access to all AP180 signals (except the four JTAG signals: TCK, TMS, TDO and TDI are used to connect the Serial Adapter interface through JP2)

Refer to Figure2 for the locations of the following connections:

JP1 : Power connector

JP2 : JTAG port used to connect the Target and Serial board via a 10-pin ribbon cable.

JP3 : Analog port for DAC0/DAC1 output and ADC input

JP4 : AP180 signals for reset, ALE and Port 0 ~ 3

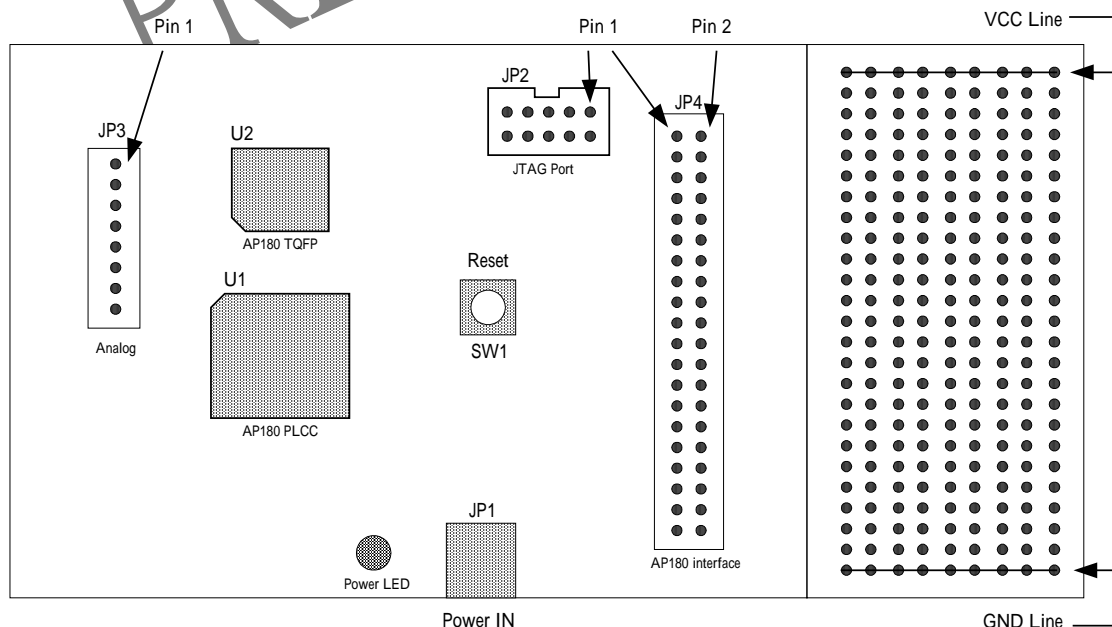


Figure2. AP180 Target Board

**AP180 Target Board JTAG Connector Pin (JP2) & Analog Pin (JP3) Descriptions**

JP2 Pin	JP2 Description	JP3 Pin	JP3 Description
1	3.3V DC Output	1	DAC0 output
2, 3, 9	DGND	2	DAC1 output
4	TCK	3,8	AGND
5	TMS	4	ADC AIN0
6	TDO	5	ADC AIN1
7	TDI	6	ADC AIN2
8,10	Not Connected (NC)	7	ADC AIN3

AP180 Target Board I/O Connector Descriptions (JP4)

Pin	Description	Pin	Description
1,2	3.3V (voltage supply)	37,38,39,40	DGND
3	P1.0	4	P0.0
5	P1.1	6	P0.1
7	P1.2	8	P0.2
9	P1.3	10	P0.3
11	P1.4	12	P0.4
13	P1.5	14	P0.5
15	P1.6	16	P0.6
17	P1.7	18	P0.7
19	P3.0	20	P2.7
21	P3.1	22	P2.6
23	P3.2	24	P2.5
25	P3.3	26	P2.4
27	P3.4	28	P2.3
29	P3.5	30	P2.2
31	P3.6	32	P2.1
33	P3.7	34	P2.0
35	RESET	36	ALE

Serial Adapter

The Serial Adapter provides the interface between the PC's RS232 serial port and the AP180 JTAG interface, supporting in-system debug/programming circuitry. The Serial Adapter may be powered from the target board via the Serial Adapter's 10-pin JTAG connector. (The target board can not be powered from the Serial Adapter.)

Notes: When powering the Serial Adapter via the JTAG connector, the input voltage to the JTAG connector's power pin must be 3.0V to 3.6V DC.

Serial Adapter JTAG Connector

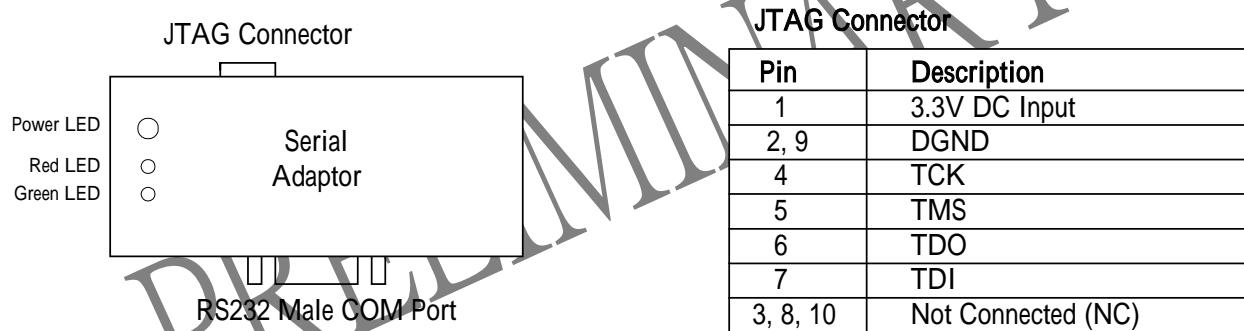


Figure3. Serial Adaptor JTAG Connector

Red LED	Green LED	Description
OFF	OFF	Target MCU is in Normal Run
ON	ON	JTAG ISP download or JTAG ICE access
ON	OFF	ISP download Fail or Something wrong for ICE
OFF	ON	ISP download Pass or ICE Pass

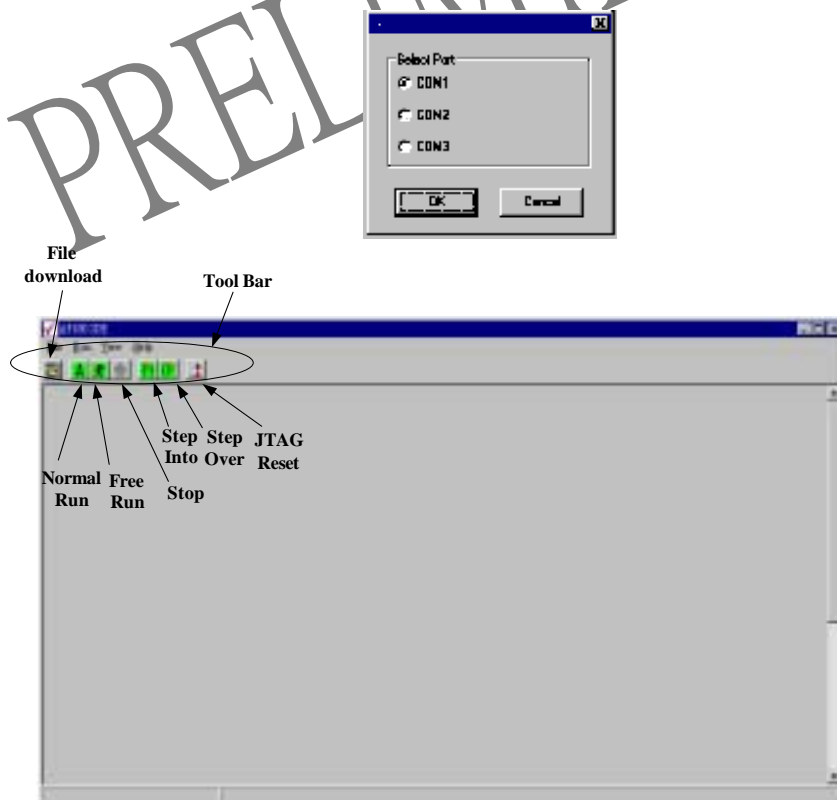
AP180 IDE

AP180 integrated development environment (IDE) provides all the tools necessary to test your projects. The IDE interfaces with the AP180 MCU's on-chip JTAG and emulation logic provide in-system programming and non-intrusive, full speed, in-circuit emulation. The source code can be assembler code or C language, and its download file for FLASH ROM code is Intel Hex format or Keil C Build Target file. Assembly language or C language source-level debug is supported depending on download file .

AP180 emulation system supports assembler or C language source-level debugging with single-step execution (including step into and step over), run-to-breakpoint, and the inspection/modification of registers, data memory and Flash ROM memory.

RUN AP180 IDE

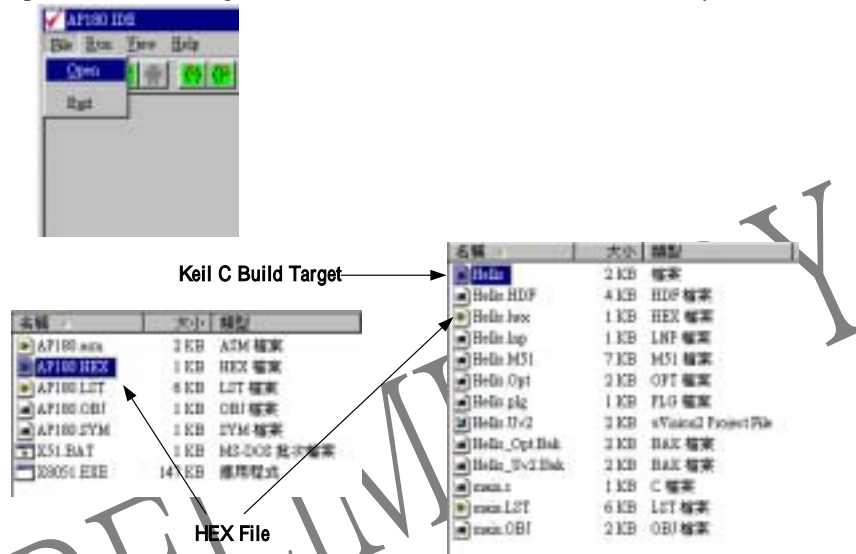
After the hardware setup, user needs to run the IDE tool (double click AP180IDE.exe) and select the com port. The examples are as follow:



Flash Programming

The JTAG ISP allows code to be downloaded to the MCU's on-chip Flash memory immediately after a build without having to leave the IDE, minimizing the time between source-code modification and in-system debug.

File Open: Select "File/Open" to load the target file and download it to Flash ROM memory.

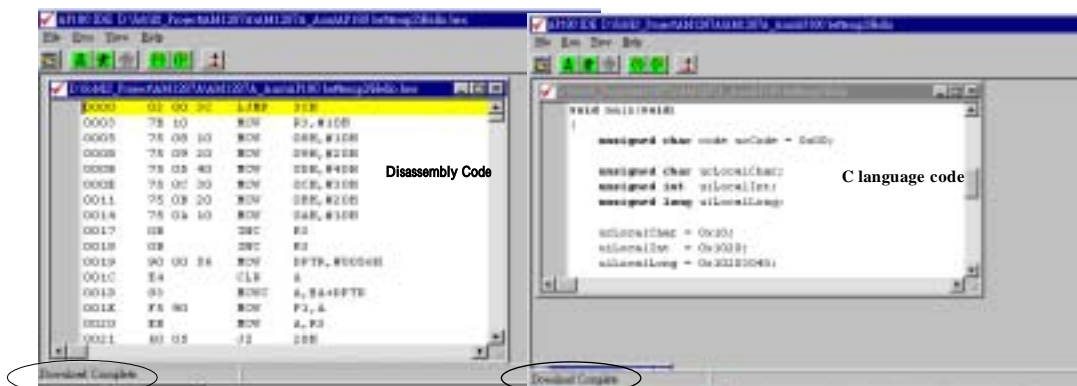


Source Level Debug

The source window is also the working debug window. You can view the current program counter location in the source code window, set and clear breakpoints, and perform single-step execution at the source-code level while monitoring register and memory contents.

Download HEX File and Disassembly code window

Download Keil C build target and C language code window



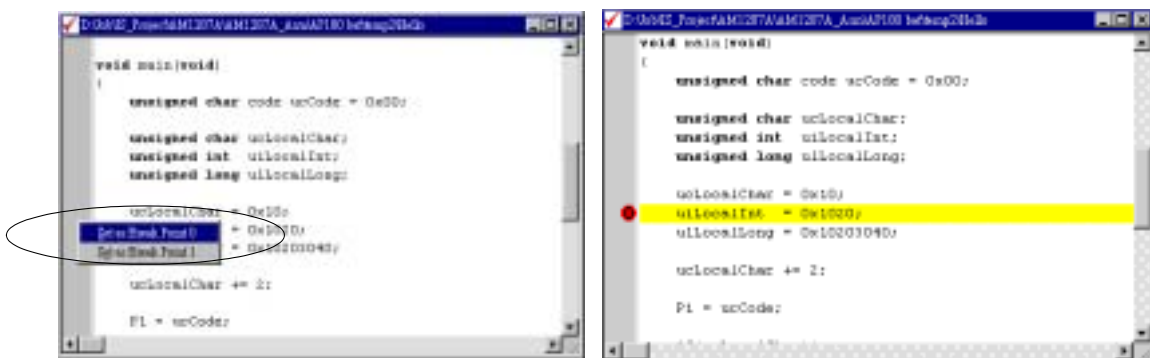
Register and Data Memory Windows

Register and memory windows display SFR and data memory contents. The windows are updated every time program execution stops and values that have changed since the last stop are highlighted. SFR Register and memory contents can be modified by editing the window displays.



Breakpoints

There are 2 breakpoints used for AP180. Breakpoints can be set in source lines to stop execution immediately before the first instruction of the specified source line is executed. It is supported by the MCU's on-chip emulation circuitry and do not affect the real-time execution of the program. The IC will stop at break point after executing "Normal Run".



[illegible]