



I-jet connect macros

IAR Embedded Workbench® for ARM

This document describes the connect macros for initializing device-specific debug access.

The specific problem addressed here is the unlocking debug access in password-protected devices.

This document is valid for I-jet and JTAGjet debugging probes.

Connect macros

These device-specific macros are called before the debugger accesses the target core debug resources:

`_ExecDeviceProbeConnect`

`_ExecDeviceProbeReset`

`_ExecDeviceCoreConnect`

Each macro call is followed by its application-specific counterpart `exexUser...` setup macro, making it possible to override or extend the function of the device-specific macro.

Because those connections are called at very early stages of the debugger connection, the command set is limited:

- Any C-SPY system macro that can access core resources must be avoided. The restriction list includes, but is not limited, to these macros: all reset macros, macros that set a breakpoint, `__jtagCP15*`, and macros for memory access.
- Only DAP commands are allowed in the I-jet probe macro `__probeCmd`, for example `__probeCmd("dap.RAPw", ...)`.

The connect macros should be placed in a `dmac` file referred from the device `i79` file, or in a `mac` file with a path referred from the **Project>Options>Debugger>Setup>Setup macros** option.

`_ExecDeviceProbeConnect`

This function is called when the probe detects a power-up event, in other words, a rising edge on the Vref/Vcc lines of the connector. The event is registered by the probe even if the debugger is not running, and reported later when the debugger starts.

Powering the target from the probe also generates the power-up event.

Starting the debugger does not call `_ExecDeviceProbeConnect` if there is no power-up edge on the JTAG connector.

`_ExecDeviceProbeReset`

This function is called when the probe detects a rising edge at the nSRST/nRESET pin of the JTAG

I-jet connect macros

header. The event is registered by the probe even if the debugger is not running, and reported later when the debugger starts.

Depending on the way the nSRST/nRESET line is wired on the board, the reset event might be reported in these cases:

- On power up, following the call to the `_ExecDeviceProbeConnect` function.
- On hardware or custom reset from the debugger.
- On system reset (Cortex-M), watchdog and other in-chip *software* resets, if they assert an external reset line.
- After pressing the reset button on the board.

If both power-up and reset events are latched by the probe, only `_ExecDeviceProbeConnect` is called.

Starting the debugger does not call `_ExecDeviceProbeReset` if there is no rising reset edge on the JTAG.

`_ExecDeviceCoreConnect`

This function is called just before the debugger connects to a core, allowing the device-specific code to enable debug access to the core.

In multicore systems, this function is called for each core the debugger connects to.

Unlock macro example

This is an example of a C-SPY `dmac` macro that unlocks debug access with a four-word password.

```
// Default password
__param passwd0 = 0xFFFFFFFF;
__param passwd1 = 0xFFFFFFFF;
__param passwd2 = 0xFFFFFFFF;
__param passwd3 = 0xFFFFFFFF;

UnlockDAP(w0, w1, w2, w3)
{
    __var port; // DAP AP port
    __var baseAddr; //Unlock register base address
    __var unlockKey; //Unlock key

    Port = 3;
    baseAddr = 0xC000;
    unlockKey = 0xCCAA5533;

    __probeCmd("dap.APw", baseAddr + 0x0, unlockKey, port);
    __probeCmd("dap.APw", baseAddr + 0x04, w0, port);
```

I-jet connect macros

```
__probeCmd("dap.APw", baseAddr + 0x08, w1, port);
__probeCmd("dap.APw", baseAddr + 0x0C, w2, port);
__probeCmd("dap.APw", baseAddr + 0x10, w3, port);

If (__probeCmd("dap.Apr", baseAddr + 0x20, port) & 0x20)
{
    __message "Device is unlocked\n";
}
else
{
    __message "Device remains locked\n";
}
}

__ExecDeviceCoreConnect()
{
    __message "Unlock DAP with password ", passwd0, passwd1, passwd2, passwd3,
    "\n";
    UnlockDAP(passwd0, passwd1, passwd2, passwd3);
}

__ExecDeviceExit()
{
    __message "Lock up DAP on exit\n";
    UnlockDAP(0, 0, 0, 0);
}
```

To set the actual password, use **Project>Options>Debugger>Extra Options**:

```
--macro_param passwd0=0x12345678
--macro_param passwd1=0x22223333
--macro_param passwd2=0x44445555
--macro_param passwd3=0x66667777
```

CoreSight DAP Access Commands

This section lists the DAP commands that can be used with the `__probeCmd` function.

DAP command syntax

? help	Displays a list of commands.
----------	------------------------------

I-jet connect macros

<code>p [port]</code>	Sets/displays a DAP port number 0..255 (the default is 1). Specifies only the APSEL part of the SELECT register.
-----------------------	--

Memory Access MEM-AP:

<code>w addr v [port]</code>	Writes to the specified address at the selected memory access AP port.
<code>r addr v [port]</code>	Reads from the specified address at the selected memory access AP port.
	These commands always set address bank zero.

AP register access:

<code>APw reg v [port]</code>	Writes to the specified address at the selected AP port.
<code>APr reg [port]</code>	Reads from the specified address at the selected AP port.
	These commands use <code>reg[7:4]</code> bits as the bank selector APBANKSEL.

DP and AP elementary register operations:

<code>RDPr reg</code>	Elementary DAP-DP read.
<code>RDPw reg v</code>	Elementary DAP-DP write.
<code>RAPr reg</code>	Elementary DAP-AP read (result of previous operation).
<code>RAPw reg v</code>	Elementary DAP-AP write.
	These commands do not use the port set by the <code>p</code> command. Use <code>dap.RDPw</code> to set the SELECT register.

Note: The port parameter only specifies the APSEL part of the SELECT register.

The `__probe` function allows for some flexibility in listing the command parameters. These syntax variants are all equivalent:

```
__probeCmd("APw 0x100 20 1");
__probeCmd("APw 0x100 20", 1);
__probeCmd("APw 0x100", 20, 1);
__probeCmd("APw", 0x100, 20, 1);
```

I-jet connect macros

IAR, IAR Systems, IAR Embedded Workbench, C-SPY, visualState, The Code to Success, IAR KickStart Kit, IAR, I-jet, I-scope, and the logotype of IAR Systems are trademarks or registered trademarks owned by IAR Systems.

All information is subject to change without notice. IAR Systems assumes no responsibility for errors and shall not be liable for any damage or expenses.

© 2014 IAR Systems AB.