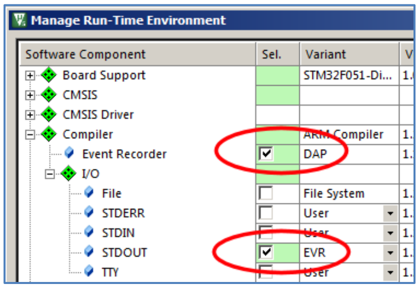
23) printf for STM32F0 Cortex-M0: Uses Event Recorder and CoreSight DAP:

Event Recorder is a new Vision feature that can be used to instrument your code. Keil RTX5 and Middleware is already instrumented with Event Recorder. Event Recorder can provide a printf utility using the DAP read/write abilities of the Debug Access Port. A UART is not used. This can be used with STM32F0 Cortex-M0 parts as well as any STM32F processor. This is the same technology used in Watch, Memory and Peripheral windows. This method can also be used with any Cortex-M processor. It does not use SWV as does the method on page 14.

1. Stop the program if it is running and exit Debug mode.

Configure Event Recorder:

2. Open the Manage Run-Time Environment utility. This window opens:
3. Expand Compiler and I/O as shown.
4. Select Event Recorder and STDOUT and EVR as shown:
5. All the blocks should be green. If not, click on the Resolve button.
6. Click OK to close this window.
7. retarget_io.c and EventRecorder.c will be added to your project under the Compiler group in the Project window.
8. Right click near the top of Blinky.c, and select Insert "#include" and select #include "EventRecorder.h".
9. At the beginning of the main() function, add this line: EventRecorderInitialize (EventRecordAll, 1);



Manage Run-Time Environment

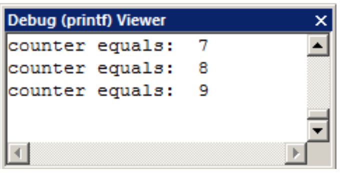
Software Component	Sel.	Variant	V
Board Support		STM32F051-Di...	1.
CMSIS			
CMSIS Driver			
Compiler		ARM Compiler	1.
Event Recorder	☑	DAP	1.
I/O			
File	☐	File System	1.
STDERR	☐	User	1.
STDIN	☐	User	1.
STDOUT	☑	EVR	1.
TTY	☐	User	1.

Add a printf statement to Blinky.c:

1. In Blinky.c add #include "stdio.h" near the top of the file.
2. In Blinky.c, near line 91 just after the if (counter>…. Line, add this line: printf("counter equals: %d\n", counter);
3. Select File/Save All or click .

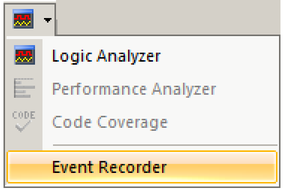
Build and RUN the Blinky program and view printf:

1. Rebuild the source files .
2. Enter Debug mode . Click on RUN .
3. Select View/Serial Windows and select Debug (printf) Viewer.
4. The values of counter is displayed as seen here:
5. Open the Event Recorder window:
6. Information about the printf statements are displayed as shown below:
7. You can annotate your own sources and display events. See www.keil.com/support/man/docs/uv4/uv4_db_dbg_evr.htm

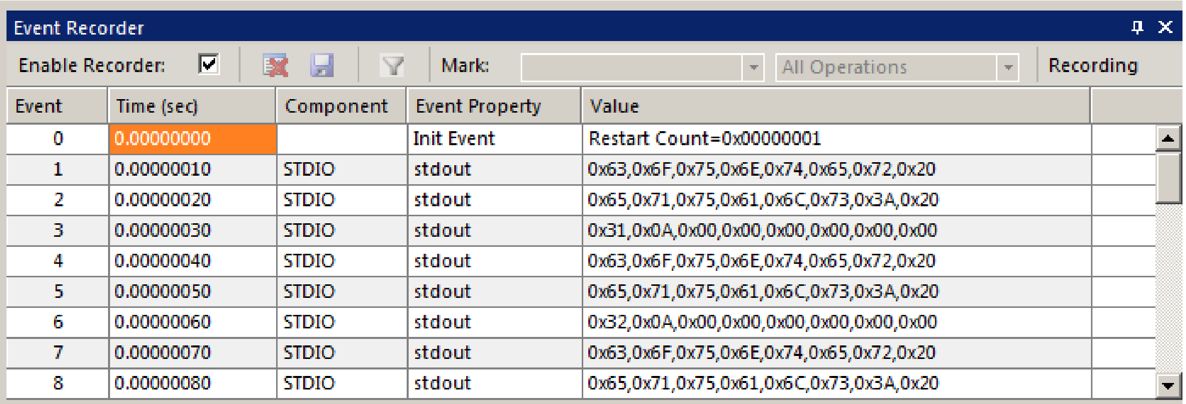


Debug (printf) Viewer

```
counter equals:  7
counter equals:  8
counter equals:  9
```



TIP: Keil Middleware and RTX5 are annotated using Event Recorder.



Event Recorder

Enable Recorder: ☑ Mark: All Operations Recording

Event	Time (sec)	Component	Event Property	Value
0	0.00000000		Init Event	Restart Count=0x00000001
1	0.00000010	STDIO	stdout	0x63,0x6F,0x75,0x6E,0x74,0x65,0x72,0x20
2	0.00000020	STDIO	stdout	0x65,0x71,0x75,0x61,0x6C,0x73,0x3A,0x20
3	0.00000030	STDIO	stdout	0x31,0x0A,0x00,0x00,0x00,0x00,0x00,0x00
4	0.00000040	STDIO	stdout	0x63,0x6F,0x75,0x6E,0x74,0x65,0x72,0x20
5	0.00000050	STDIO	stdout	0x65,0x71,0x75,0x61,0x6C,0x73,0x3A,0x20
6	0.00000060	STDIO	stdout	0x32,0x0A,0x00,0x00,0x00,0x00,0x00,0x00
7	0.00000070	STDIO	stdout	0x63,0x6F,0x75,0x6E,0x74,0x65,0x72,0x20
8	0.00000080	STDIO	stdout	0x65,0x71,0x75,0x61,0x6C,0x73,0x3A,0x20