This feature is only applicable to microcontrollers which are based on ARM Cortex M3 plus processors. M3, or M4, M7, or higher. So, this discussion will not be applicable to ARM Cortex M0 or M0 plus processors. So, in that case, you will not be able to use the ARM Cortex processors trace functionality. In ARM Cortex processor, we can make printf work by using the SWO pin of the SWD interface. So, SWO stands for serial wire output.

Now, our board also has one more circuitry so which is at the front end of the board and that we call as ST link V2 or V1 debug circuitry. So, that is a ST link onboard debug circulatory. So, by using that debug circuitry your PC communicates with the board. Through that debug circuitry you actually write your program to the internal flash of the microcontroller, you read various memory locations of the microcontroller, you make processor run you make processor stop, so all those debug related activity you do by taking help of this debug circuitry which is present on the board. Debug circuitry will talk to your PC over a USB connection. So, there is a pin called SWO pin which is coming all the way from ARM Cortex M processor and it is connected to the debug circuitry. So, the printf actually works over this SWO pin.

Inside the ARM Cortex M4 processor, there is a unit or a peripheral called ITM unit. ITM stands for Instrumentation Trace Macrocell Unit. So, this is inside the processor. So the ITM is an optional application driven trace source that supports printf style debugging to trace operating system and application events, and it can also be used to generate diagnostic system information.

This unit is only available in ARM Cortex M3 or above processors. So, it is not available in ARM Cortex M0 processor. And to debug the processor, so debug means if you want to read the memory location, if you want to read the processor related register, if you want to make the processor halt or if you want it to run. So if you want to do all these activities then we do that using the debug interface. The debug interface what we are using here is SWD. So, SWD stands for Serial Wire Debug which is a two wire protocol for accessing the ARM debug interface. SWD works over SWD connector, and that SWD connector has three pins. In which 2 pins are used for debug and one pin is used for trace. So, trace means in order to get the trace related information from the processor.

Let's explore some more points about SWD debug interface. Now the serial wire debug it is a two-wire protocol for accessing the ARM debug interface. So, it is part of the ARM debug interface specification v5 and it is an alternative to JTAG. The physical layer of SWD consist of only two lines . One is called SWDIO, which is a bidirectional data line which carries debug related data and SWD clock . A clock which is driven by the host. So, in our board the host is actually the ST link circuitry. So, SWDIO a which is a data line which actually carries the debug related commands. So, like for example, when you insert a breakpoint in your IDE. So, that information will be sent over SWDIO data line to the processor. So, if you want to stop the processor from the IDE, then that information is actually carried over the SWDIO line with the help of SWCLK to the processor.

So, in order to talk to the processor you can use these 2 pins of the SWD interface. SWDIO and SWCLK. So, both are managed by the ST Link circuitry which is present on your boards. So, by using SWD interface you should be able to program MCUs internal flash, you can access memory regions, add breakpoints, stop or run the CPU.So, other good thing about SWD is you can use the serial wire viewer for your printf statements for debugging.

So, as I said SWD comes with only two pins which are used for debug, but there is one optional pin that is what we call as SWO, which we can use for printf functionality. So, now there is also another debug interface which is called as JTAG. The difference between JTAG and SWD is, JTAG actually needs more pins than SWD. JTAG was the traditional mechanism for debug connections for ARM7 or ARM 9 family, but with the Cortex-M family, ARM introduced the Serial Wire Debug(SWD) interface. SWD is designed to reduce the pin count required for debug from the 4 used by JTAG down to 2. So, in addition, SWD interface provides one more pin called SWO which is used for Single Wire Viewing, which is a low cost tracing technology. Let's move forward.

If I zoom this ITM unit further. So, what you see is a FIFO, or you can call it as a buffer, or a register. It's a hardware buffer which is there inside the ITM unit. So, now all you need to do is write the printf data into this FIFO.So, that FIFO is actually connected to the SWO pin which is coming out of the processor, and it is coming all the way to your debug circuitry which is present on the board, right? So, the moment your printf writes into this FIFO that messages will actually come over the SWO pin and you then capture it. That's it. So, in the IDE there is a provision to capture this SWO pin. So, remember that not all IDE support this feature of capturing SWO pin.

Fortunately STM32 Cube IDE and true studios, so those IDEs actually support these functionalities. So, SWO pin is connected to ST link circuitry of the board and can be captured using our debug software. So, let's see that in a moment how to do that. But, this is a idea behind how printf works in ARM Cortex Mx processor. So, there is a ITM unit, and it has a FIFO, and your printf somehow should write into that FIFO, and that FIFO is connected to the SWO pin, and through that you actually get that message back to the IDE.

Refer <https://interrupt.memfault.com/blog/printf-on-embedded#redirecting-stdio> also