How to configure Eclipse for semihosting

1 What is semihosting

The following explanation is an excerpt from a much more comprehensive description which can be found here: https://www.keil.com/support/man/docs/armcc/armcc_pge1358787046598.htm.

“Semihosting is a mechanism that enables code running on an ARM target to communicate and use the Input/Output facilities on a host computer that is running a debugger.

Examples of these facilities include keyboard input, screen output, and disk I/O. For example, you can use this mechanism to enable functions in the C library, such as printf() and scanf(), to use the screen and keyboard of the host instead of having a screen and keyboard on the target system.

This is useful because development hardware often does not have all the input and output facilities of the final system. Semihosting enables the host computer to provide these facilities. “

E.g. with Eclipse configured for semihosting the console of the Eclipse IDE can be used to display output sent by printf() to the host system.

There are three small steps to be followed to enable semihosting:

1. Modify the GNU linker settings

2. Enable semihosting in the debugger settings

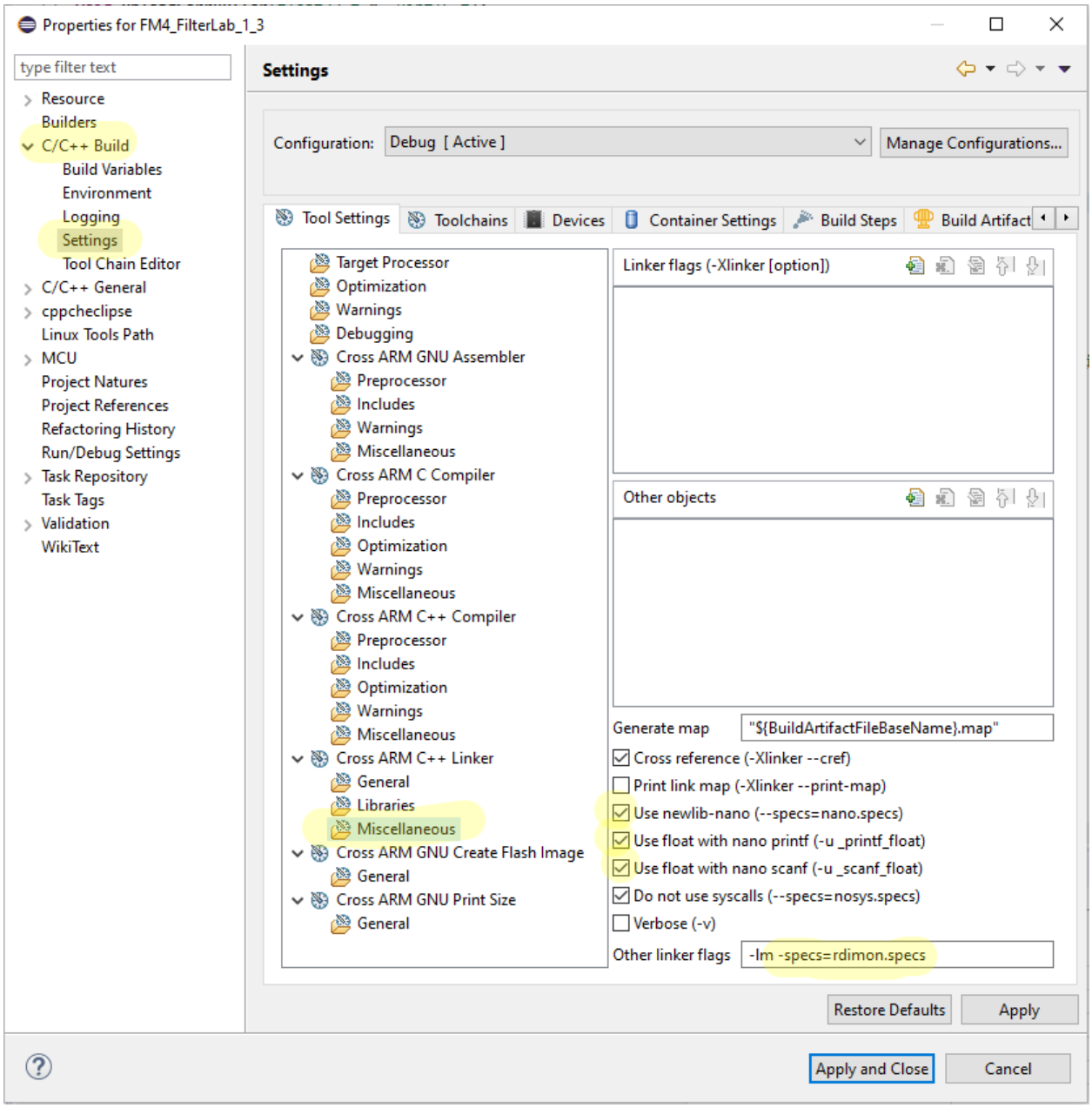
3. Enable semihosting in the application

In the following only the steps to enable semihosting are described.

2 GNU linker settings

Select the project in the Project Explorer window.

Navigate to *Project -> Properties -> C/C++ Build -> Settings*.



In tab *Tool Settings* select *Cross ARM C++ Linker / Miscellaneous.*

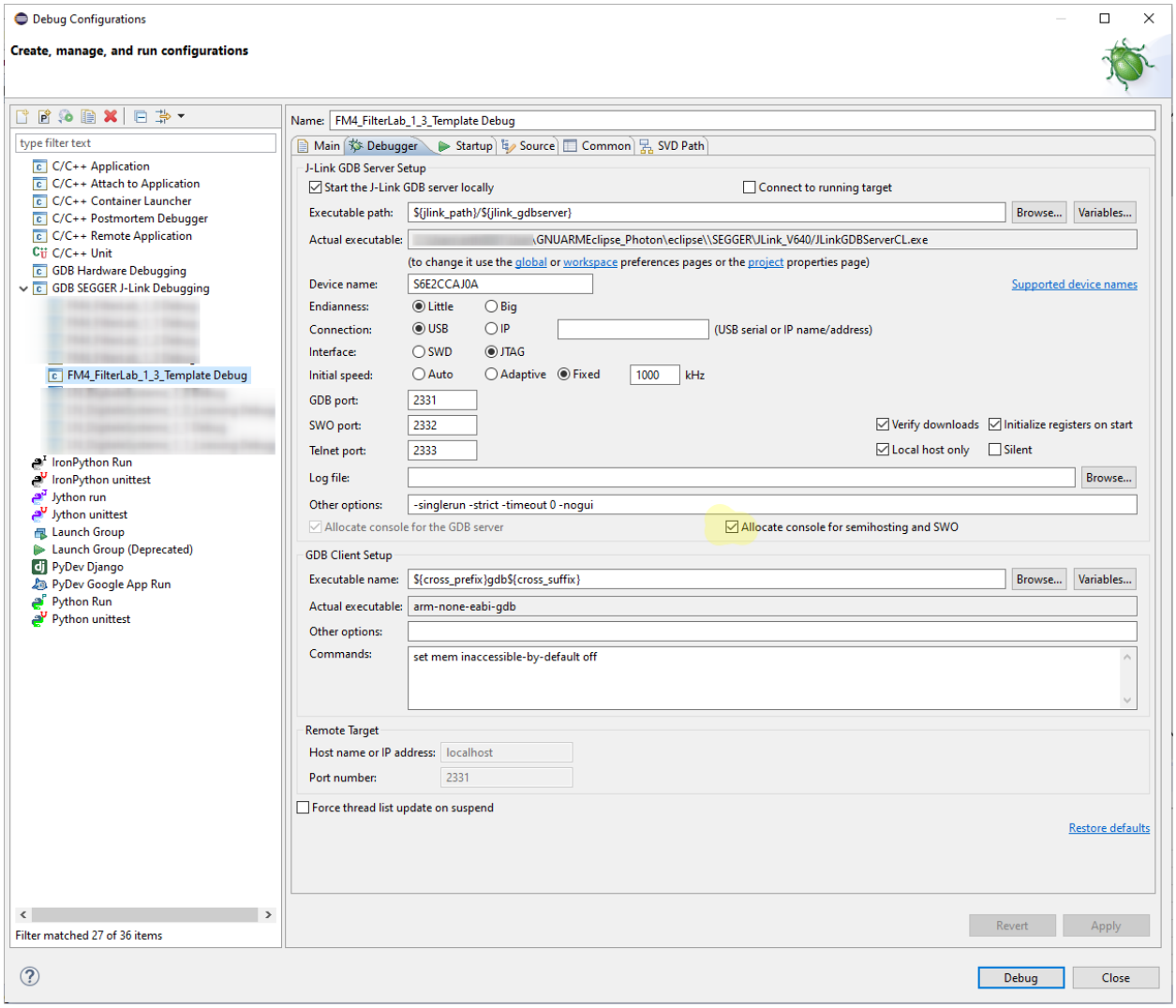
- Click checkboxes
  - Use newlib-nano
  - Use float with nano printf (check only if printf with floating point support is needed)
  - Use float with nano scanf (check only if scanff with floating point support is needed)
- In the field *Other linker flags* add *-specs=rdimon.specs*

3 Debugger settings

Navigate to *Run -> Debug Configurations… -> GDB SEGGER J-Link Debugging.*

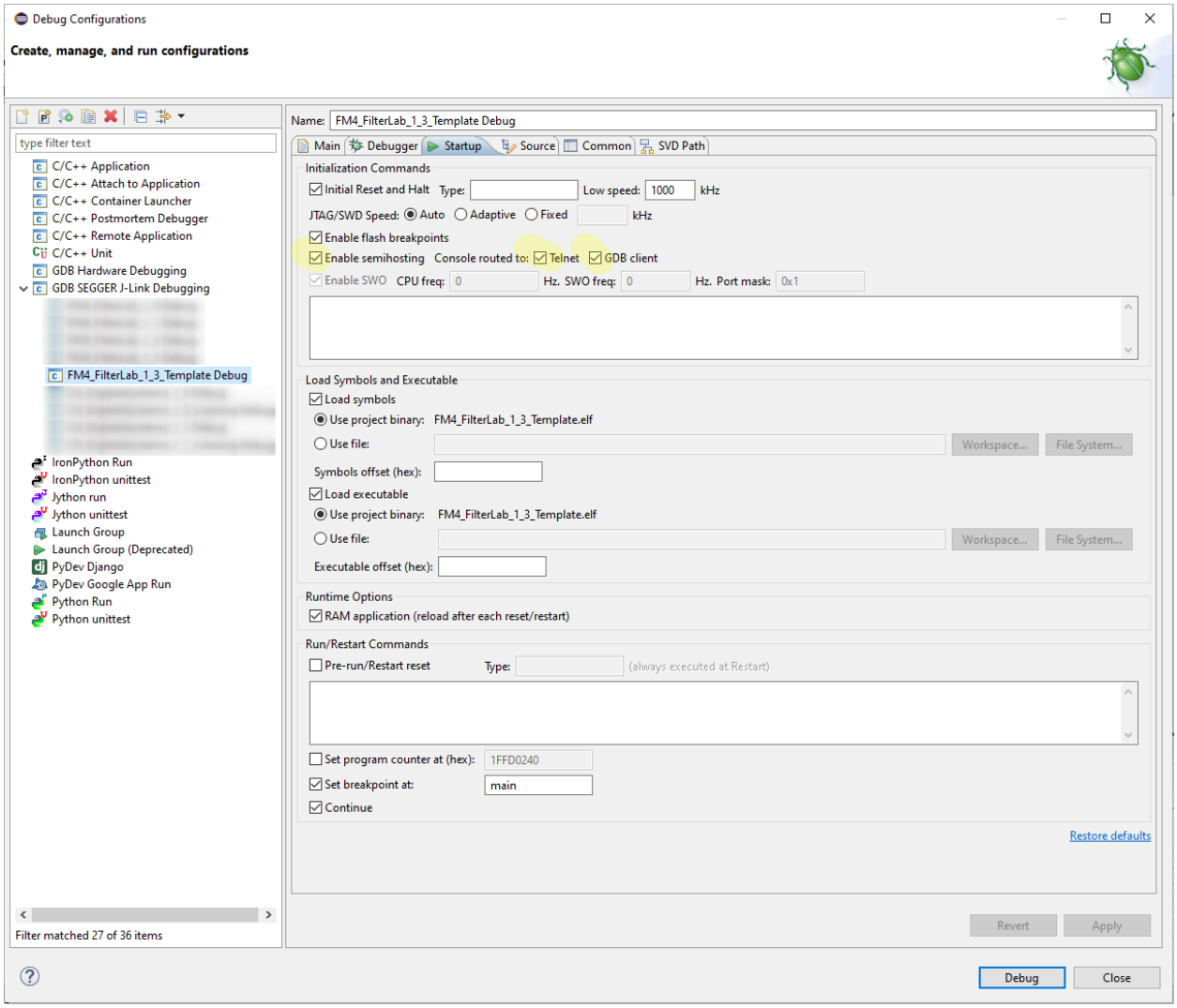
Select the debug configuration for your project.

Click on tab *Debugger*.



Click checkbox: *Allocate console for semihosting and SWO.*

Click on tab *Startup*



- Click checkbox: *Enable semihosting*
- Click checkboxes: *Console routed to Telnet and GDB client*

4 Application

Before commands like printf or file I/O can be used, semihosting must be initialized in the application code. Add the following lines of code:

```
extern void initialise_monitor_handles(void); // protoype

initialise_monitor_handles(); // must be executed before printf
```

5 Examples

5.1 Hello world

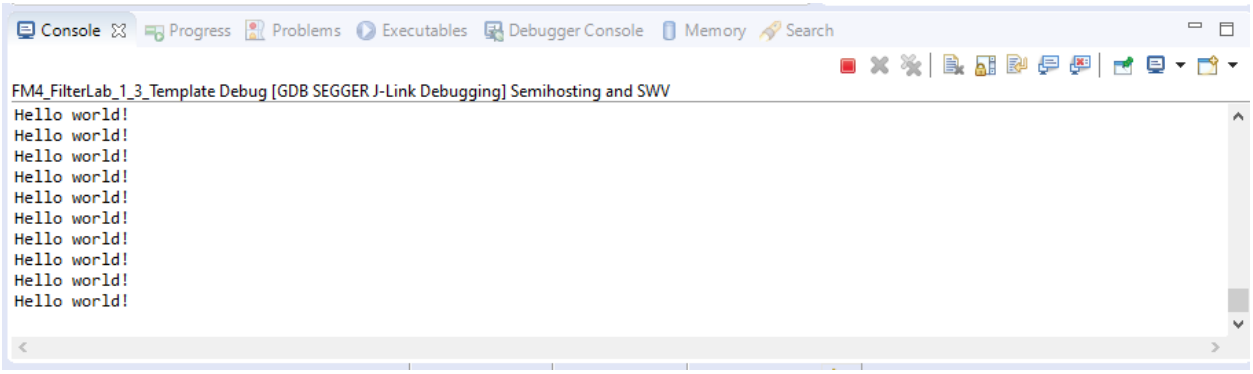
The following example prints “Hello world!” on the eclipse console

```
#include <stdio.h>


// prototypes
extern void initialise_monitor_handles(void); // needed for semihosting


int main(void)
{
	initialise_monitor_handles(); // needed for semihosting

	while(1)
	{
		// send a string to the terminal
		printf("Hello world!\n");
	}
}
```



5.2 File I/O

This example shows how the file I/O of the host system can be used to upload the content of a buffer for further examination. Don't forget to add the two lines of code mentioned before to enable semihosting.

```
void uploadBuffer(int16_t * buffer, uint16_t bufferSize)
{
	FILE * dataFile;
	uint16_t i;

	if ((dataFile = fopen ("D:/buffer.txt", "w")) == NULL)
	{
		printf("fopen for D:/buffer.txt failed \r\n");
		return;
	}

	for(i = 0; i < buffersize; i++ )
	{
		fprintf(dataFile, "%d", buffer[i]);
		fprintf(dataFile, "\n");
	}
	fclose(dataFile);
	return;
}
```