## Freescale Semiconductor

# Creating a New MQX Project Using GCC and C++

1) Add the following macros in ***C:\Freescale\Freescale_MQX_4_1\config\<board>\user_config.h***
   - `#define` MQX_SUPPRESS_FILE_DEF 1
   - `#define` BSPCFG_ENABLE_CPP 1
   - `#define` MQX_SUPPRESS_STDIO_MACROS 1

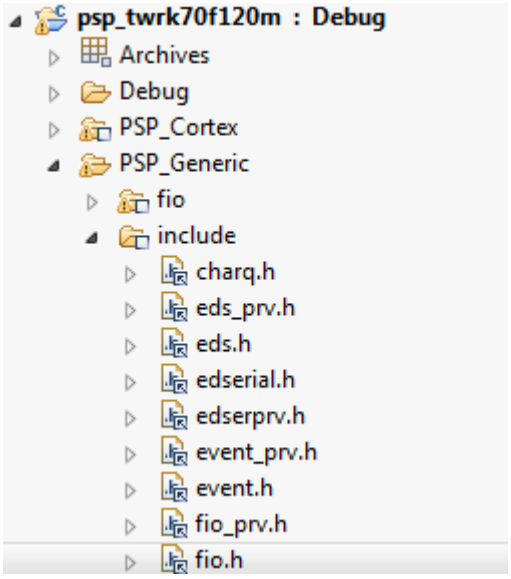
**MQX_SUPRESS_FILE_DEF** disables definition of FILE. EWL has its own FILE with different meaning, thus, EWL should link with EWL's FILE, MQX should link with MQX_FILE, but the linking order is only one – you either take FILE from MQX or from EWL. Therefore we disable, with this macro, MQX to define FILE. Then MQX has to use MQX_FILE and EWL can continue using EWL FILE.

**BSPCFG_ENABLE_CPP** enables EWL functions required when using c++.

**MQX_SUPRESS_STDIO_MACROS** Similar as MQX_SUPRESS_FILE_DEF, stdio macros are different files (MQX_FILE) than EWL definitions, different pointer in memory and different structure members. Then, MQX must use its own definitions.

2) Now, stdin, stdout and stderr definitions must be redefined. To do this there are 2 options.

   A) The correct way would be to add the following macros in ***C:\Freescale\Freescale_MQX_4_1\mqx\source\include\fio.h.***




```
/*--------------------------------------------------------------------------*/
/*
 *                          MACRO DECLARATIONS
 */
#if !defined(MQX_SUPPRESS_STDIO_MACROS) || MQX_SUPPRESS_STDIO_MACROS == 0

#define stdin     (MQX_FILE_PTR)_io_get_handle(IO_STDIN)
#define stdout    (MQX_FILE_PTR)_io_get_handle(IO_STDOUT)
#define stderr    (MQX_FILE_PTR)_io_get_handle(IO_STDERR)

#define getchar()     _io_fgetc(stdin)
#define getline(x,y) _io_fgetline(stdin, (x), (y))
#define gets(x)       _io_fgets((x), 0, stdin)
#define putchar(c)    _io_fputc((c), stdout)
#define puts(s)       _io_fputs((s), stdout)
#define status()      _io_fstatus(stdin)
#define ungetc(c)     _io_fungetc(c, stdin)
#endif /* MQX_SUPPRESS_STDIO_MACROS */

#define _mqxio_stdin     (MQX_FILE_PTR)_io_get_handle(IO_STDIN)
#define _mqxio_stdout    (MQX_FILE_PTR)_io_get_handle(IO_STDOUT)
#define _mqxio_stderr    (MQX_FILE_PTR)_io_get_handle(IO_STDERR)
```

Rebuild BSP and PSP:
***C:\Freescale\Freescale_MQX_4_1\mqx\build\cw10gcc\bsp_<board>***
***C:\Freescale\Freescale_MQX_4_1\mqx\build\cw10gcc\psp_<board>***

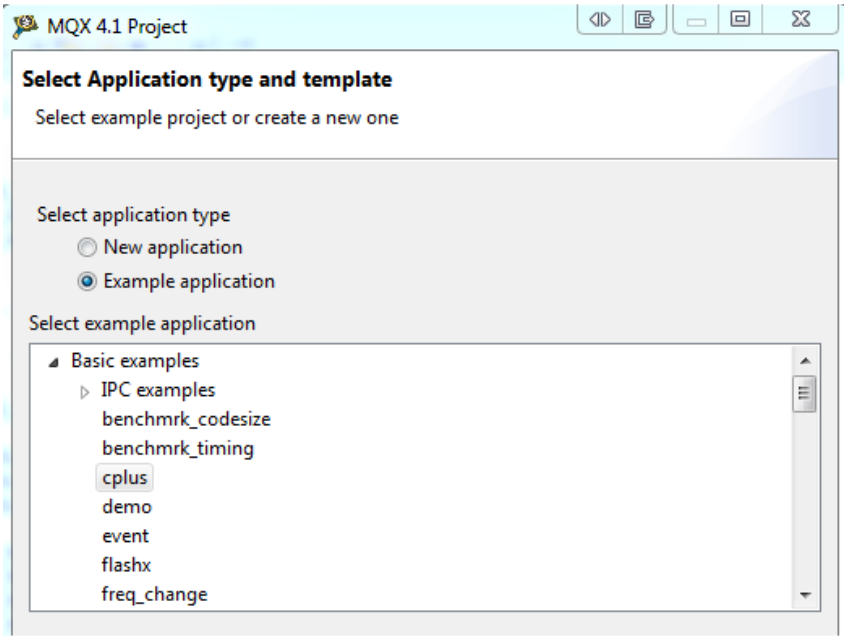
Errors will appear, go to each of them and replace `stdin` for `_mqxio_stdin` and `stdout` for `_mqxio_stdout`. Recompile again, no errors should appear this time.

- `#define _mqxio_stdin (MQX_FILE_PTR)_io_get_handle(IO_STDIN)`
- `#define _mqxio_stdout (MQX_FILE_PTR)_io_get_handle(IO_STDOUT)`
- `#define _mqxio_stderr (MQX_FILE_PTR)_io_get_handle(IO_STDERR)`

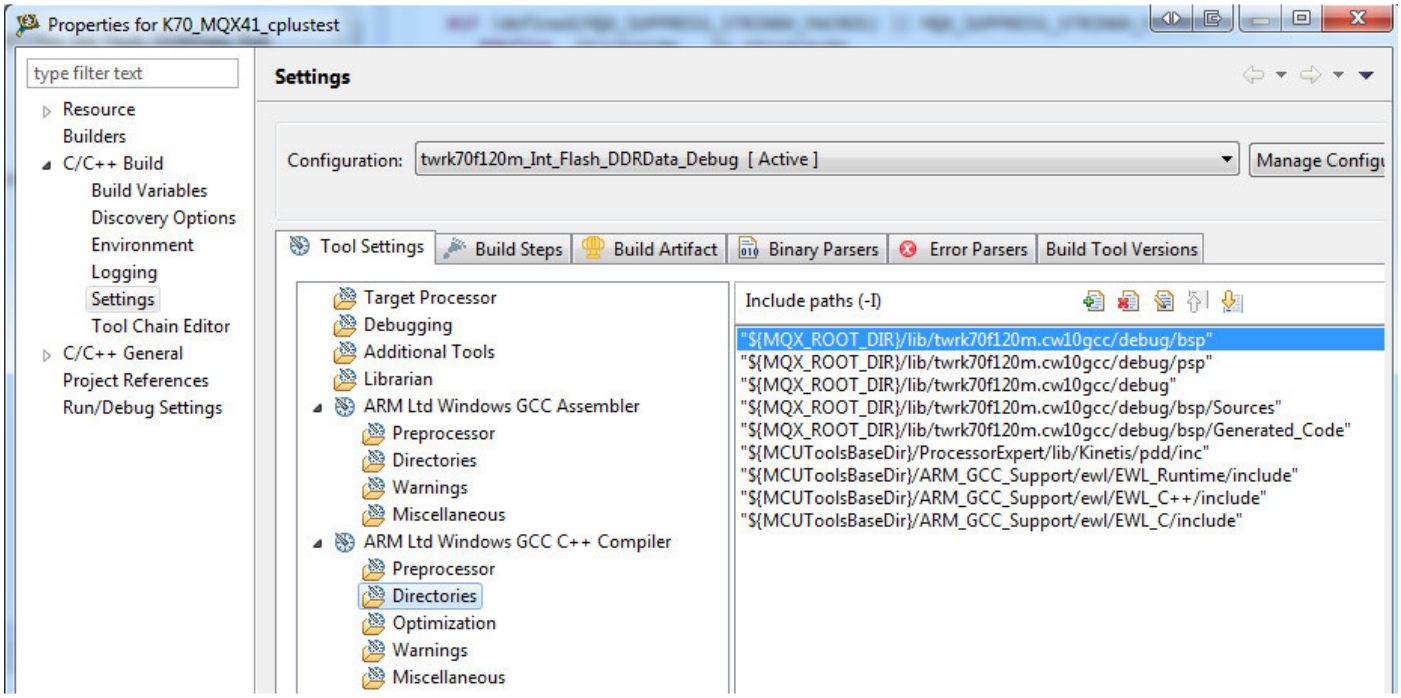
B) A work around would be to force the project to use `stdin`, `stdout` and `stderr`. With the default build settings the linker will use the MQX definitions, the side effect is that warnings will appear alerting that these macros are redefined.

```
/*--------------------------------------------------------------------------*/
/*
 *                          MACRO DECLARATIONS
 */
#if !defined(MQX_SUPPRESS_STDIO_MACROS) || MQX_SUPPRESS_STDIO_MACROS == 0

#define getchar()     _io_fgetc(stdin)
#define getline(x,y) _io_fgetline(stdin, (x), (y))
#define gets(x)       _io_fgets((x), 0, stdin)
#define putchar(c)    _io_fputc((c), stdout)
#define puts(s)       _io_fputs((s), stdout)
#define status()      _io_fstatus(stdin)
#define ungetc(c)     _io_fungetc(c, stdin)
#endif /* MQX_SUPPRESS_STDIO_MACROS */

#define stdin      (MQX_FILE_PTR)_io_get_handle(IO_STDIN)
#define stdout     (MQX_FILE_PTR)_io_get_handle(IO_STDOUT)
#define stderr     (MQX_FILE_PTR)_io_get_handle(IO_STDERR)
```

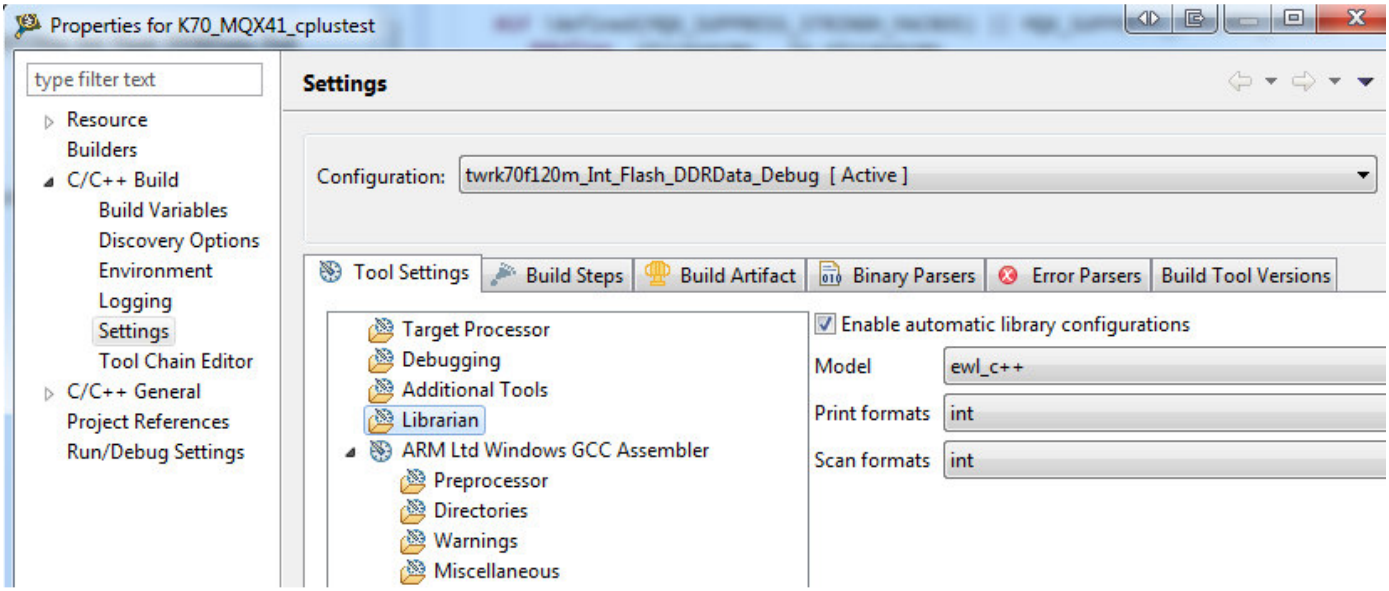
3) Now you can create a new C++ project. Go to ***menu > File > New > MQX4.1 Project*** and in the Wizard "***Select Application type and template***" window select ***Examples Application > Basic examples > cplus***.



4) In menu ***Project > Properties > C/C++ Build > Settings > ARM Ltd Windows GCC C++ Compiler > Directories*** add "${MCUToolsBaseDir}/ARM_EABI_Support/ewl/EWL_C++/include" and put the below paths at the end of the Compiler Directories list.
   - "${MCUToolsBaseDir}/ARM_EABI_Support/ewl/EWL_C++/include"
   - "${MCUToolsBaseDir}/ARM_EABI_Support/ewl/EWL_C/include"




5) Then in ***menu Project > Properties > C/C++ Build > Librarian*** check "***Enable automatic library configurations***" and choose "***Model***" = ewl_c++ / int / int.



6) Finally, if you replaced `stdout` for `_mqxio_stdout` (in step 2) you need to do the same in cplus.cpp and build the project.