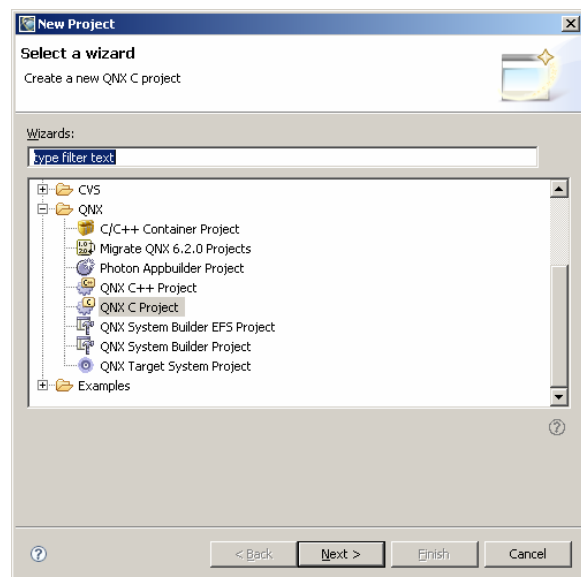


## Entwicklung von C Bibliotheken unter QNX

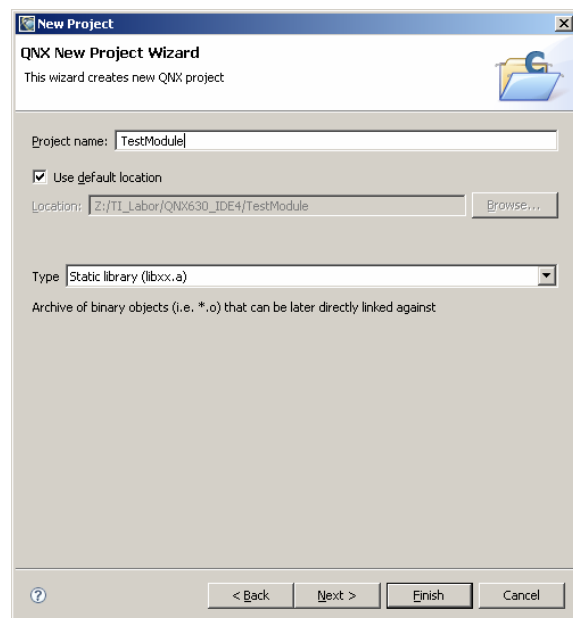
Diese Kurzanleitung beschreibt die Entwicklung einer C Bibliothek unter QNX und QNX Momentics IDE. Die Funktionen einer C Bibliothek werden in einem C Modul implementiert.

### Erzeugen und Übersetzung eines C Moduls

Für das C Modul wird ein eigenes QNX C Projekt (File→New→Project...) angelegt.



Der Typ des Projekts wird auf „Static library (libxx.a)“ gestellt.



Legen Sie in diesem Projekt die \*.h Datei und die \*.c Dateien des C Moduls an.

Zur Erinnerung ein Beispiel für ein Modul, das aus einer \*.h Datei und zwei \*.c Dateien besteht.

```
#ifndef _M_h_ // Prevent multiple inclusions
#define _M_h_

/*
 * M.h
 *
 * Our Test module. This module ...
 * Describe the purpose of the module and briefly how it works.
 *
 * Changes: -Change history, in reverse chronological order-
 * 17 Sep 05 -- print_text function added -- FK
 * 16 Sep 05 -- Created -- FK
 */

// Include other header files by this interface.

// Macros used by module clients

// Types and structures defined by this module

// "Extern" declarations of global variables defined in this module

// Prototypes for functions implemented by this module

void print_text(char *s);
// print_text prints ...

void print_hallo_world();
// print_hallo_world prints ...

#endif /* M.h */
```

```
/*
 * M1.c
 *
 * Implementation of part 1 of our test module.
 *
 * Note that this file header is quite short! There is no need to repeat
 * information given in the header file.
 */
#include "M.h"
#include <stdio.h>

// Global variables, if any-

// Types needed only by the implementation

// Define local (static) variables-

// Prototypes for local functions-

// Local and global definitions

void print_text(char *s)
{
    printf(s);
    printf("\n");
}
```

```
/*
 * M2.c
 *
 * Implementation of part 2 of our test module.
 *
 * Note that this file header is quite short! There is no need to repeat
 * information given in the header file.
 */
#include "M.h"
#include <stdio.h>

// Global variables, if any-

// Types needed only by the implementation

// Define local (static) variables-

// Prototypes for local functions-

// Local and global definitions

void print_hallo_world()
{
    printf("Hallo World\n");
}
```

Nachdem das Modul fehlerfrei übersetzt wurde, liegen unterschiedliche Versionen der Bibliothek (z.B. Debug Version, Release Version) unter

TestModule/x86/a	(Release Version)
TestModule/x86/a-g	(Debug Version)

## Einbinden einer Bibliothek

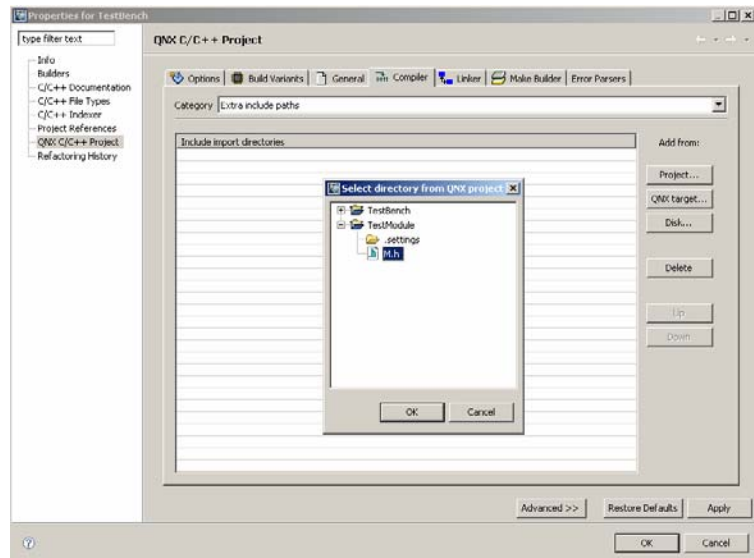
Das folgende Testprogramm verwendet die oben erstellte C Bibliothek.

```
#include <stdlib.h>
#include <M.h>

int main(int argc, char *argv[])
{
    print_text("TEST print_text");
    print_hallo_world();
    return EXIT_SUCCESS;
}
```

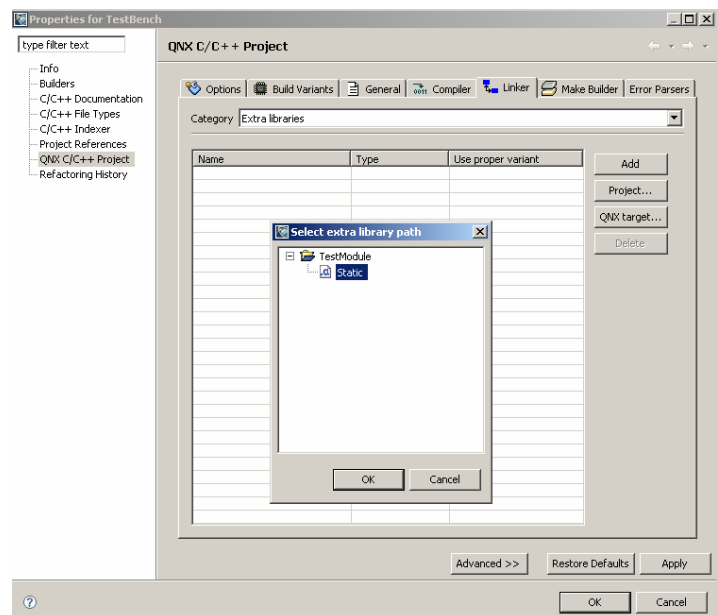
Da die Datei M.h das Interface der Bibliotheksfunktionen beschreibt, benötigt der C-Compiler die Information, wo die Datei M.h liegt. Dies wird über die Properties des Projektes, das die C Bibliothek einbindet, wie folgt eingestellt:

- Wählen Sie auf der linken Seite des Fenster die Kategorie „QNX C/C++ Project“.
- Wählen Sie den Punkt „Compiler“ aus.
- Wählen Sie die Category „Extra include paths“ aus.
- Wählen Sie auf der rechten Seite über „Project...“ die notwendigen \*.h Dateien aus den entsprechenden Projekten aus.



Der Linker benötigt die Information, wo die C Bibliothek liegt. Über die Properties des Projekts, das die C Bibliothek benötigt, wird dies wie folgt eingestellt:

- Wählen Sie auf der linken Seite des Fenster die Kategorie „QNX C/C++ Project“.
- Wählen Sie den Punkt „Linker“ aus.
- Wählen Sie die Category „Extra libraries“ aus.
- Wählen Sie auf der rechten Seite über „Project...“ die notwendigen Bibliotheken aus.
- Setzen Sie „User proper variant“ auf YES.



Nun können Sie das Programm wie üblich ausführen bzw. im Debugger analysieren.