HIGH POWER DEVELOPMENT TOOLS FOR USE WITH THE CMS EMBEDDED CONTROLLER RANGE

CAMBRIDGE MICROPROCESSOR SYSTEMS LIMITED

# TOOLS *for* DEVELOPMENT

'C'
Assembler
Debug


*The picture shows the contents of the C Development Pack (PC not included). The chosen controller will be supplied in place of the FM-200 shown.*

## FEATURES

- Full Library Support
- Expandable
- Extensive Demonstration Programs
- Mains Power Supply / Adapter
- Full Documentation
- PC Utilities
- UK Based Technical Support
- Minos Operating System
- Fully Multitasking
- Very easy to use
- Powerful Debug tools

### 'C' Development

- Optimising 'C' compiler
- 68000 Cross Assembler
- Relocating Linker
- Flexible Text Editor
- Make Utility
- Project Based
- Library Manager
- C Source Level Debug
- Symbolic Debug

## DESCRIPTION

**The 'C' Development Pack provides the user with very powerful development tools suitable for use with any of our embedded control product ranges at a very competitive price. The user has a choice of programming in 'C', Assembler or a mixture of the two. All Development Packs are supplied with Minos, the real time multi tasking Royalty FREE operating system. Please see separate leaflet for full details. The same development tools can be used to develop applications for all our controllers, although the libraries and Minos modules will vary from one controller to another. In many cases the user can then choose the controller most suitable to a particular application. Please refer to the controller data sheets for details on each controller.**

### 'C' or Assembler Development

The 'C' development tools include a fully optimising 'C' Cross Compiler, relocating linker, cross assembler, library manager, make utility, flexible text editor, terminal emulator, comprehensive documentation, extensive examples, Minos, power supply and interface cable. Along with the user's P.C. this is all that is required to develop programs to run on our controller product range in 'C' or 68000 assembler.

The programming environment is very easy to use. It is project based which allows each application to be made up of a number source files and header files, each project having its own directory to keep the project files together. Each source file can be 'C' or assembler depending on the application. The text editor provided in the Development Tools allows additional files to be added to the project automatically, however if you prefer to use your own text editor you can easily edit the project makefile to include your additional source files.

Once the source files have been written the project should be built. In the text editor provided this is done simply by selecting an option from the menu. The build operation runs through the project makefile compiling and assembling the source files. It will then link the source files using the included libraries to produce a Minos program. Any errors that are detected during the compile or linking phase will be displayed in a window in the text editor. The errors can then be corrected and the application rebuilt. The program is then downloaded to the battery backed RAM on the controller card where the code can be tested and debugged. Once the program is working correctly it is a very simple matter to recompile the code for an address in the Flash memory and then simply download the program into the Flash for permanent storage using the same serial port connection. All programs can be run from power on, whether they are located in battery backed RAM or Flash. The programming language is very easy to learn, simple to use and quickly enables the user to develop their own programs using the libraries and hardware drivers provided.

The 'C' compiler used in the Development Packs is the GNU C compiler. This software is provided on a royalty free basis allowing multiple copies of the Development environment to be used free of charge, ideal for the larger projects where more than one engineer may be working on the project. Full support is provided for floating point calculations as well as symbolic debug and in-line assembler code. There is also full support for writing interrupt service routines in 'C'.

# TOOLS *for* DEVELOPMENT

CAMBRIDGE MICROPROCESSOR SYSTEMS LIMITED

A new addition to the 'C' Development tools is the Source Level Debug. This allows 'C' source code to be debugged at source level rather than the symbolic level. The source level debugging is performed using the remote serial protocol in GDB to debug the application using one of the serial ports on the controller card. Source level debug is included in the QuickFire Development Packs and can be purchased as an addition for Flash-Module and Micro-Module Development Packs. Source Level Debug is only available for use with our GNU based compilers.

```
/* This 'C' program will take each channel low then high, writing  */
/* the state of each channel to the Liquid Crystal Display and     */
/* sending the same text down the RS-232 serial to the PC Screen   */
/* This text is also sent to a modem via serial S2 for remote viewing*/

#include <stdio.h>
#include <adio.h>

void main(void)
{
    int i,n;                    /* Declare Integer Variables */
    FILE *lcd;                  /* Declare Pointer Variable */
    FILE *modem;                /* Declare Pointer Variable */
    lcd = fopen(":LCD","w");    /* Open path to LCD for output */
    modem = fopen(":S2","w");   /* Open path to modem for output */
    adioinit();                 /* Initialise digital I/O */
    for (i=0;i<4;i++)
    {
        n = outport(i);         /* set first four ports as outputs */
    }
                                /* repeat until a key is pressed */
    while (!ready(fileno(stdin)))
    {
        for (i=0;i<32;i++)
        {
            writech(i,1);       /*Pull channel low */
            fprintf(lcd,"Channel %d = Low",i); /* Write to LCD */
            fprintf(modem,"Channel %d = Low",i); /* Write to modem */
            printf("Channel %d = Low",i); /* Write to PC Screen */
            delay(10);          /* delay 100 ms */
            writech(i,0);       /* Take channel high */
            fprintf(lcd,"Channel %d = High",i); /* Write to LCD */
            fprintf(modem,"Channel %d = High",i); /* Write to modem*/
            printf("Channel %d = High",i); /* Write to PC Screen */
        }
    }
    fclose(modem);              /* Close the path to the Modem */
    fclose(lcd);                /* Lastly Close Path to LCD */
}
```

# ORDER CODES

| Order Number | Product Name | 1 off | 50 off |
|---|---|---|---|
| QFD-200 | QuickFire C Development | £599 | |
| *Multi License Minos 'C' programming package ( including QF-200 )* | | | |
| QFA-200 | QuickFire Additional Packs | £195 | £146.25 |
| *Additional Development Packages for Educational and Commercial Users* | | | |
| GC-100FM | FlashModule C Development | £595 | |
| *Multi License Minos 'C' programming package ( including FM-200 )* | | | |
| GC-200FMS | FlashModule C Starter Pack | £295 | |
| *Single license Minos 'C' programming package ( including FM-200 )* | | | |
| GH-030 | MicroModule C Development | £595 | |
| *Multi License Minos 'C' programming package ( including Micro-Module)* | | | |
| GH-035 | Micro-Module C Starter Pack | £295 | |
| *Single license Minos 'C' programming package ( including Micro-Module )* | | | |
| FM-SLDB | Source Level Debug for FM | £95 | |
| *GNU Source level debug for use with the FlashModule C Development or Starter pack* | | | |
| UM-SLDB | Source Level Debug for UM | £95 | |
| *GNU Source level debug for use with the Micro-Module C Development or Starter pack* | | | |

DevelopmentPack02 020719

To see if your compiler can be upgraded to the GNU compiler please contact our sales department.

The run time environment on the CMS controller cards utilises the Minos real time multi tasking operating system. This allows many programs and tasks to run concurrently. A 'shell' environment is provided which allows the user to download, debug and run programs. A number of utilities are also provided in the 'shell' environment to allow the user to view the module directory listing, examine all running functions and redirect input or output to other devices.

The tools supplied with our GNU based 'C' Development systems are suitable for use on the following operating systems

Windows 95

Windows 98

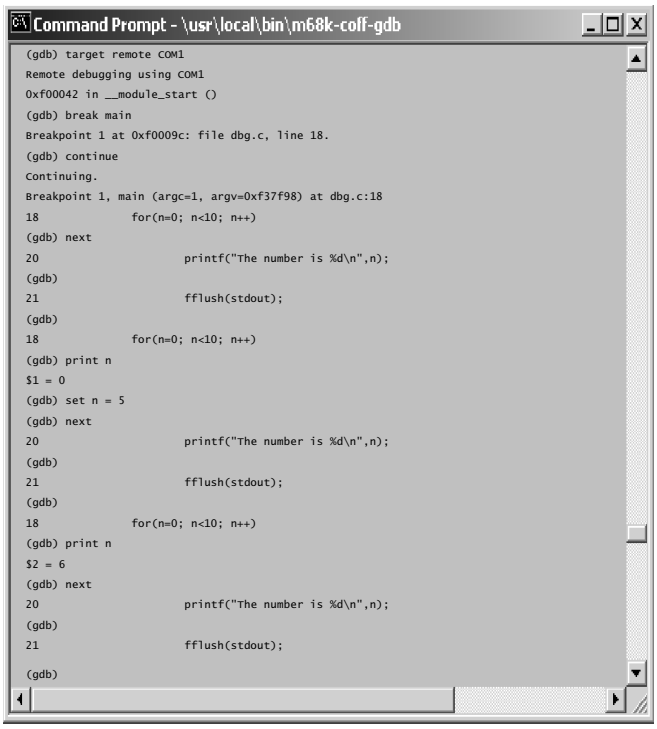
Windows Me

Windows NT4

Windows 2000

Windows XP

They are not compatible with earlier versions of Windows or DOS based systems.




The above shows a typical Source Level Debug screen. It allows the user to examine and modify program variables as the code is being traced while the application is running.


CAMBRIDGE MICROPROCESSOR SYSTEMS LIMITED

Units 17 - 18 Zone 'D', Chelmsford Road Industrial Estate, Great Dunmow, Essex UK CM6 1XG

Telephone +44 (0) 1371 875 644

Fax +44 (0) 1371 876 077

Email sales@cms.uk.com

Web Site http://www.cms.uk.com