

THE C INTERFACE MANUAL
FOR THE MOTOROLA MC68000

VERSION 2.2

WHITESMITHS, LTD.

MARCH 1983

Copyright (C) 1983 by Whitesmiths, Ltd.

All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of Whitesmiths, Ltd.

DISCLAIMER

Whitesmiths, Ltd. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose.

Further, Whitesmiths, Ltd. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Whitesmiths, Ltd. to notify any person of such revision or changes.

Whitesmiths is a registered trademark of Whitesmiths, Ltd.

UNIX is a trademark of Bell Laboratories.

MC68000 is a trademark of Motorola, Inc.

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 Overview of the C Interface	1
1.2 System Requirements	2
2. THE MACHINE MODEL	5
2.1 Data Representations	5
2.2 Register Usage and Conventions	7
2.3 The Stack Frame	9
3. RUN-TIME LIBRARY INTERFACE	12
3.1 Calling Sequence Conventions	12
3.2 Argument Passing and Returning Values	15
3.3 Built-in Functions and Utilities	18

1. INTRODUCTION

This manual describes the interface between programs written in the C language and the target Motorola MC68000 hardware environment, using the Whitesmiths, Ltd. C Native Compiler or Cross Compiler systems.

The machine interface layout enforces strict conventions regarding byte ordering, function entry/exit, stack alignment, and register management to maintain uniform execution structures across complex cross-compilation boundaries.

1.1 DATA SIZES AND ALIGNMENTS

The C compiler mappings onto the MC68000 microprocessor follow specific alignment grids dictated by the 16-bit external bus architecture. Basic data configurations include the following properties:

Type	Size (Bytes)	Alignment (Bytes)

char	1	1 (Byte aligned)
short	2	2 (Word aligned)
int	2 or 4	2 (Word aligned)
long	4	2 (Word aligned)
pointer	4	2 (Word aligned)
float	4	2 (Word aligned)
double	8	2 (Word aligned)