

C LANGUAGE DEVELOPMENT SYSTEM

Introl-C/6809 is a powerful C language compiler system that is designed to facilitate the development of high-efficiency software for the 6809. The Introl-C package includes a C Compiler, 6809 Relocating Assembler, Linker, Loader, Library Manager, and Standard Library. It has been in the field since early 1982 and has gained widespread acceptance among users for its reliable and comprehensive support of the C language as well as its ease of use. Its ability to generate exceptionally compact, fast executing code has long distinguished the Introl-C implementation as being the most efficient high level language that is available for the 6809 and has resulted in Introl-C's widespread use in the industrial community for development of process control software. Programs developed using Introl-C are typically within 15 to 20% or less of the size and speed of programs written entirely in 6809 assembler. For the particular case of the Eratosthenes Sieve Benchmark, the p/n UC6809 resident Introl-C Compiler, for example, produces a 176 byte compiled module, a total program size of 2007 bytes, and a program execute time of 8 seconds on a 2 Mhz 6809.

Code produced under Introl-C is re-entrant, relocatable, and ROMable and may be installed on any 6809 target, including ROM-based systems. No fees or royalties of any type are imposed on object code programs developed using the compiler. Introl-C/6809 is available as resident software for 6809-based hosts running UniFlex, Flex, or OS9. Cross-software versions of Introl-C are available for PDP-11 based hosts running UNIX (or any of the UNIX look-alikes such as TNIX, VENIX, etc), PDP-11 based hosts running RSX11M, and also for IBM PC hosts running PC DOS or MDOS.

Introl-C is designed to the standard C language specification defined by Kernighan & Ritchie and supports all features of the language except fields, doubles, and the #if and the #line preprocessor directives (all other preprocessor directives, including #ifdef and #ifndef, are fully supported). Extensions to the standard language include provisions to permit nesting of comments, use of separate name spaces for all union and structure member names, and provision to allow symbol names up to 90 significant characters in length. Most C source programs developed using Introl-C are directly usable as input to standard UNIX C compilers.

User interface is designed for ease of use permitting C source files to be converted into executable outputs with a minimum of effort on the part of the programmer. For example, a single command line entry of the form:

```
icc filename (options)
```

will cause a C source file to be fully compiled and assembled to produce a relocatable object module. Similarly, relocatable modules may be linked and automatically loaded to produce an executable output via a simple command line entry of the form:

```
ilink file1 [file2 file3 ...] [options]
```

Numerous options are supported to permit versatile, user-controlled alteration of the standard compilation, assembly, linking and loading processes. Option specifications, however, are generally required only for specialized circumstances since the defaults for unused options are designed to select standard conditions that "make sense" for the great majority of program development situations.

The C Compiler is a 4-pass program that generates an optimized assembly language file as its output. In normal use, the 4 sequential compilation passes execute automatically and are followed by automatic execution of the included Assembler, thus resulting in a fully compiled, fully assembled relocatable object module as the result of a typical compiler call. The intermediate assembly language file generated by the C Compiler is available to the user, however. Compile time options include capability for the user to selectively place data of a given type under any of 16 different location counters (ie "segments"), ability to generate either position-dependent or position-independent code and/or data, and capabilities for specifying #define pre-processor directives directly on the compiler call line. The 16 location counters provided by Introl-C, and the features for generating either position-dependent or position-independent code and data, allow significant flexibility when finished programs are to be placed in ROM.

The included R09 Assembler is a full-featured 6809 Relocating Assembler. Although nominally furnished to provide automatic assembly of the C Compiler's output, the Assembler may also be called directly by the user for converting user-written assembler text files into relocatable object modules. The Assembler supports all addressing modes of the 6809, recognizes all standard opcodes, and will accept arbitrarily complex assembly-type input expressions. A unique feature of the Introl Assembler is its assignment of a user-definable "class" identifier to each module it produces which is used (in combination with the module's name) for identification purposes by the Linker. This feature allows the user to create (and the Linker to distinguish between) multiple versions of identically-named C support functions within the Standard Library, for example, and provides the basis for a powerful and convenient link-time capability for tailoring the link process to minimize runtime overhead in the final program.

The Linker accepts any number of relocatable modules as input and produces a single relocatable module as its output. Multiple-pass linking is supported. The Linker also supports "partial linking", wherein several component modules of an overall program may be linked together to form a single resultant "partially linked" module which may then be reused as input in subsequent linking operations. Linked programs up to 64K in size may be produced.

Although a specific host-related Loader is supplied with each Introl-C software package, any of four different types of Introl Loaders are potentially usable with any of the Introl-C compiler systems: a FLD Loader, which generates a file of output load records in hex format (Motorola S Records, Intel Hex, Tektronix Hex, or Extended Tektronix Hex formats); a ULD Loader, which generates outputs that are executable under UniFlex; a FLD Loader, which generates outputs executable under Flex; and an OLD Loader, which generates outputs executable under OS9. The loader supplied with P/N XC6809, RC6809, and PC6809 cross-software packages is the HLD Loader; the ULD Loader is supplied with the P/N UC6809 package; the FLD Loader is supplied with the P/N FC6809 package; and the OLD Loader is supplied with the P/N OC6809 package. Any of these several Loader types, however, is also optionally available, at extra cost, as a "cross-loader" for use with any resident or cross-software version of Introl-C. The HLD loader is particularly useful when developing software for standalone, ROM-based applications.

The Standard Library contains an extensive collection of C programming support functions, including I/O and arithmetic functions not directly performed by the 6809. Only those library functions actually required for program execution are extracted by the Linker, resulting in minimum runtime overhead in developed programs. The library furnished with each of the several resident versions of Introl-C is host-O.S.-specific in nature whereas the library supplied with cross-software versions is operating-system-independent and is tailored for use in standalone-target applications. Standard Library source code is available as an extra cost option.

The Library Manager provides convenient and versatile capabilities for adding, deleting, and modifying Standard Library functions, thus permitting unique libraries of C support functions of any type to be created by the user. All library functions are stored in linkable format, thus avoiding any need for recompiling them before each use. The Library Manager, in combination with the Linker, significantly reduces development times for large programs by allowing individual parts of the program to be independently developed and compiled with a minimum of effort.

TRADEMARKS: Introl-C is a registered trademark of Introl Corporation; Flex and UniFlex are trademarks of Technical Systems Consultants; OS9 is a trademark of Microware Systems; UNIX is a trademark of Bell Laboratories; TNIX is a trademark of Tektronix Inc; VENIX is a trademark of VenturCom; PDP-11 and RSX11 are trademarks of Digital Equipment Corp; IBM PC is a trademark of International Business Machines.

6809 RESIDENT AND CROSS MACRO ASSEMBLERS

Introl's M09 resident and cross macro assemblers are designed to translate MC6809 assembly language source programs into 6809 machine code. All M09 assembler packages include the Introl M09 Relocating Macro Assembler, the ILINK Linker, a Loader, and the LIBMAN Library Manager. Resident versions of the M09 software package are available for use on 6809-based microcomputers running under UniFlex, Flex, or OS9. Cross-software versions are available for use on PDP-11 based development systems running UNIX or RSX-11M, and also for IBM PC hosts running PCDOS.

The Assembler is a full-featured relocating macro assembler that accepts a 6809 assembly language text file as input and produces a relocatable object file as its output. The included Linker and Loader permit any number of assembled modules and/or library modules to be linked together and then loaded to produce a single resultant output file in an executable format. The Library Manager provides convenient and versatile features for the user to create, and maintain, libraries of assembled modules. Resident assembler packages incorporate a host-specific Loader (ULD, FLD, or OLD, as applicable) that produces output files which are executable under the assembler's host operating system. M09 cross-assembler packages incorporate a hex-type Loader (HLD) which generates a file of output load records in any of several hex formats: Motorola S Records, Intel Hex, Tektronix Hex, or Extended Tektronix Hex format, at user option. Introl's HLD, ULD, FLD, and OLD Cross-Loader packages are optionally usable with any version of M09, and are available from Introl at extra cost. The M09 Macro Assembler is fully compatible, both in source format and object output, with the R09 Relocating Assembler that is supplied with the Introl-C/6809 C Compiler system.

M09 supports macros and conditional assembly, recognizes the standard opcodes recognized by Motorola assemblers, and supports the complete instruction set and all addressing modes of the 6809. The Assembler accepts assembly type input expressions that are arbitrarily complex. Parentheses are allowed in expressions to modify the evaluation order of operators. Symbols may be of any length, with the first 100 characters being significant and retained by the assembler. Assembly time expressions may be used in the operand of any assembler opcode or directive. Symbols and constants may be used interchangeably in an expression. All results of expressions at assembly time are 32-bit truncated integers.

The following assembly time options are supported by the M09 Assembler:

- a - Place all symbols except those beginning with a "?" character in the object file.
- c - Send Assembler's output listing to console.
- f - Force listing of all conditionally excluded code.
- i - Include all included files in output listing.
- j - Include symbols beginning with a "?" character in the output listing.
- l=(filename) - Place output listing in specified file.
- m - Include all macro-expansion-generated code in the output listing.
- n - Don't produce an output listing.
- O=(filename) - Explicitly assign name to the relocatable output file.
- p(digit)=(value) - Pass parameters to macro program being assembled.
- q=(class) - Assign numeric class identifier to the relocatable output module.
- s - Suppress listing of the symbol table.
- U - Force all undefined symbols to default to imported symbols.
- X - Don't generate an object module.
- y=(pathname) - Search "pathname" for macro files after searching current working area.
- z - Delete input file after Assembler has finished using it.

Assembly time expression operators supported:

- unary minus (twos complement)	& bitwise and
~ not (ones complement)	^ bitwise exclusive or
* multiplication	bitwise inclusive or
/ division	> greater than
% mod (remainder)	< less than
+ addition	>= greater than or equal to
- subtraction	<= less than or equal to
<< shift left	== equal to
>> shift right	!= not equal to

Assembler directives supported:

comm - Common Area	if - Numeric Conditional Assembly
dc - Define Data Constant	ifn - Numeric Conditional Assembly
ds - Define Data Storage	ifc - String Compare Conditional Assembly
else - Conditional Assembly Else	ifnc - String Non-Compare Conditional Assembly
endif - End Of Conditional Assembly	import - External Symbol Reference
endm - End Of Macro Text	lib - Load A Disk File
end - End Of Assembly	list - Terminate Previous Nolist Directive
equ - Equate Symbol With A Value	loc - Select Location Counter
err - Programmer-Generated Error	macro - Define A Macro
exitm - Exit From Macro	nolist - Turn Off The Listing
export - External Symbol Definition	offset - Absolute Offset From An Origin
fcb - Form Constant Byte	repeat - Repeat The Next line
fcc - Form Constant Character	rmb - Reserve Memory Bytes
fdb - Form Double Byte Constant	set - Set Symbol To A Value
ident - Identify Module	syn - Equate Labels

.....

Trademarks: INTROL-C is a registered trademark of Introl Corporation; UniFlex-and Flex are trademarks of Technical Systems Consultants; OS9 is a trademark of Microware Systems; UNIX is a trademark of Bell Laboratories; PDP-11 and RSX-11M are trademarks of Digital Equipment; IBM PC is a trademark of International Business Machines.

INTROL-C PRICE LIST

PART NO.	PRODUCT DESCRIPTION	1-YEAR MAINTENANCE	
		PRICE	DOMESTIC/FOREIGN

INTROL-C/6809 COMPILER PACKAGES

(All packages include ICC Compiler, R09 Assembler, ILINK Linker, LIBMAN Library Manager; resident compilers include applicable host-compatible Loader and Standard Library; cross-compilers include HLD Loader and STA09 Standard Library.

UC6809-UFIX09	Resident compiler package for 6809/UniFlex host	\$425	\$100/\$135
FC6809-FIX09	Resident compiler package for 6809/Flex host	\$425	\$100/\$135
OC6809-05909	Resident compiler package for 6809/059 host	\$425	\$100/\$135
XC6809-UNIX11	Cross-compiler package for POP-11/UNIX host	\$2500	\$500/\$550
RC6809-RSX11	Cross-compiler package for POP-11/RXS11M host	\$2500	\$500/\$550
PC6809-PCDOS	Cross-compiler package for IBM PC/PCDOS host	\$750	\$200/\$235
MAN-C6809	Manual only (Specify compiler type)	\$75	NA
MANEX-C6809	Additional Manual (Specify compiler type)	\$35	NA

.....

LIBRARY SOURCE CODE PACKAGES

UFIX09LS-(*)	Source code for UC6809 Standard library	\$400	\$100/\$135
FLX09LS-(*)	Source code for FC6809 Standard Library	\$400	\$100/\$135
OS909LS-(*)	Source code for OC6809 Standard library	\$400	\$100/\$135
STA09LS-(*)	Source code for XC6809/PC6809/RC6809 (Standalone) Library	\$400	\$100/\$135

(\*Note: Specify host O.S. format desired; ie whether UniFlex, Flex, OS9, UNIX, RSX11M, or PC DOS)

.....

CROSS LOADER PACKAGES

(HLD Loaders include STA09 Standalone Library; others include targeted-host-compatible Library)

HLD-UFIX09	HLD (hex format output) Loader for UC6809 compiler	\$150	\$30/\$40
HLD-FLX09	HLD (hex format output) Loader for FC6809 compiler	\$150	\$30/\$40
HLD-OS909	HLD (hex format output) Loader for OC6809 compiler	\$150	\$30/\$40
ULD-FLX09	ULD (UniFlex format output) Cross-loader for FC6809 compiler	\$150	\$30/\$40
ULD-OS909	ULD (UniFlex format output) Cross-Loader for OC6809 compiler	\$150	\$30/\$40
ULD-UNIX11	ULD (UniFlex format output) Cross-Loader for XC6809 compiler	\$300	\$75/\$90
ULD-PCDOS	ULD (UniFlex format output) Cross-Loader for PC6809 compiler	\$225	\$50/\$65
ULD-RSX11	ULD (UniFlex format output) Cross-Loader for RC6809 compiler	\$300	\$75/\$90
FLD-UFIX09	FLD (Flex format output) Cross-Loader for UC6809 compiler	\$150	\$30/\$40
FLD-OS909	FLD (Flex format output) Cross-Loader for OC6809 compiler	\$150	\$30/\$40
FLD-UNIX11	FLD (Flex format output) Cross-Loader for XC6809 compiler	\$300	\$75/\$90
FLD-PCDOS	FLO (Flex format output) Cross-Loader for PC6809 compiler	\$225	\$50/\$65
FLD-RSX11	FLO (Flex format output) Cross-Loader for RC6809 compiler	\$300	\$75/\$90
OLD-UFIX09	OLD (OS9 format output) Cross-Loader for UC6809 compiler	\$150	\$30/\$40
OLD-FLX09	OLD (OS9 format output) Cross-Loader for FC6809 compiler	\$150	\$30/\$40
OLD-UNIX11	OLD (OS9 format output) Cross-Loader for XC6809 compiler	\$300	\$75/\$90
OLD-PCDOS	OLD (OS9-format output) Cross-Loader for PC6809 compiler	\$225	\$50/\$65
OLD-RSX11	OLD (OS9 format output) Cross-Loader for RC6809 compiler	\$300	\$75/\$90

			1-YEAR	
			MAINTENANCE	
<u>PART NO</u>	<u>PRODUCT DESCRIPTION</u>	<u>PRICE</u>	<u>DOMESTIC/FOREIGN</u>	
6809 MACRO RELOCATING ASSEMBLER PACKAGES				
(All packages include M09 Macro Relocating Assembler, ILINK Linker and LIBMAN Library Manager; resident assemblers include host-compatible Loader; cross-assemblers include HLD Loader.)				
M09-UFX09	6809 Macro Relocating Assembler (UniFlex-09 host)	\$250	\$65/\$80	
M09-FLX09	6809 Macro Relocating Assembler (Flex-09 host)	\$250	\$65/\$80	
M09-OS909	6809 Macro Relocating Assembler (OS9-09 host)	\$250	\$65/\$80	
M09-UNX11	6809 Macro Relocating Cross-Assembler (PDP-11/UNIX host)	\$1200	\$250/\$300	
M09-RSX11	6809 Macro Relocating Cross-Assembler (PDP-11/RSX11M host)	\$1200	\$250/\$300	
M09-PCDOS	6809 Macro Relocating Cross-Assembler (IBM PC/PC DOS host)	\$375	\$100/\$135	
MAN-M09	Manual only (Specify Macro Assembler type)	\$35	NA	
MANEX-M09	Additional Manual (Specify Macro Assembler type)	\$20	NA	

#### ORDERING INFORMATION

Introl software is available on the following floppy disk formats:

UniFlex formats: 8" SSSD 77 track  
 Flex formats: 8" SSSD 77 track; 5" DSDD 40 track; 5" SSSD 35 track  
 OS9 formats: 8" SSSD 77 track; 5" DSDD 40 track; 5" SSSD 40 track  
 PDP-11/UNIX formats: 8" RX01 Tar; 8" RX02 Tar; 8" Tektronix Tar  
 PDP-11/RXS11M formats: 8" RT-11  
 IBM PC/PC DOS formats: 5" DSDD

All prices are F.O.B. Milwaukee, Wisconsin, U.S.A. Prices and product specifications are subject to change without notice. All orders must be prepaid in U.S. funds drawn on a U.S. bank or shipped C.O.D. VISA and Master Card accepted. End users in Wisconsin, please add applicable Wisconsin State Sales Taxes. All domestic orders should include \$10.00 shipping and handling, \$25.00 for all overseas orders

Prices shown are for single-CPU use licenses. Site licensing and OEM licensing is also available.

An Introl Binary Software license Agreement must be completed and returned to Introl Corporation prior to software delivery.

Trademarks: Introl-C is a registered trademark of Introl Corporation; UniFlex and Flex are trademarks of Technical Systems Consultants; OS9 is a trademark of Microware Systems; UNIX is a trademark of Bell Laboratories; IBM PC is a trademark of International Business Machines; PDP-11, RSX11, and RT-11 are trademarks of Digital Equipment Corp.

INTROL CORPORATION  
 647 West Virginia Street \*\*\* Milwaukee, Wisconsin 53204  
 Telephone (414) 276-2937

# INTROL-C/6809 STANDARD LIBRARIES

(Representative Support Functions Provided)

STA09	UFIX09	FLX09	OS909	FUNCTION	DESCRIPTION
LIB.	LIB.	LIB.	LIB.		
.....	X	.....	.....	abs	integer absolute value
.....	X	.....	.....	access	determine accessibility of file
.....	X	.....	.....	acct	turn accounting on/off
.....	X	.....	.....	alarm	send alarm signal after specified time
X	....	X	....	alloc	allocate memory
X	....	X	....	atof	convert string to float
X	....	X	....	atoi	convert string to integer
X	....	X	....	atol	convert string to long
.....	X	.....	.....	brk	change core allocation
.....	.....	X	.....	cforkf	fork off a program
.....	.....	X	.....	chain	chain a new executable module from a C program
.....	X	.....	.....	chdir	change default directory
.....	X	.....	.....	chmod	change file access permission
.....	X	.....	.....	chown	change the owner of a file
.....	X	.....	X	close	close a file
X	....	X	....	cprep	prepare environment for a C program
.....	.....	.....	X	crc	cycle redundancy count
X	....	X	....	creat	create a file
.....	X	....	X	cstart	runtime preparation routine
.....	X	.....	.....	dup	duplicate an open file descriptor
.....	X	.....	.....	dup2	duplicate an open file descriptor
X	....	X	....	ecvt	float to string conversion
.....	X	....	X	execl	execute a program
.....	X	.....	.....	execv	execute a program
.....	X	....	X	exit	exit a program with file cleanup
.....	X	....	X	_exit	exit a program without file cleanup
X	....	X	....	_extend	extend float
.....	X	....	X	fclose	close file
X	....	X	....	fcvt	float to string conversion
.....	X	.....	.....	fdopen	open a file
.....	X	.....	X	fflush	flush file buffer
X	....	X	....	fgets	read file into string
.....	X	.....	.....	fdopen	open a file
.....	X	.....	X	fork	spawn a new process
X	....	X	....	fprintf	formatted output Conversion
X	....	X	....	fputs	write a string to a file
X	....	X	....	free	free memory
X	....	X	....	fscanf	formatted input conversion
.....	X	.....	.....	fseek	seek to position in a stream
.....	X	.....	.....	fstat	get file status of open file
.....	X	.....	.....	ftell	tell the current position in a file
.....	X	.....	.....	ftime	current time
X	....	X	....	getc	get next character from a file
X	....	X	....	getchar	get a character from the standard input
.....	X	.....	.....	getegid	get effective group user id
.....	X	.....	.....	geteuid	get effective tiser id
.....	X	.....	.....	getgid	get group user id
.....	X	.....	X	getpid	get process id
X	....	X	....	gets	read input into string
.....	.....	.....	X	getstat	get status of file or device id
.....	X	.....	X	getuid	get user
.....	X	.....	.....	gtty	get status of file or device
X	....	X	....	index	find first occurrence of character
.....	.....	X	.....	intercept	intercept signals
.....	X	.....	.....	ioctl	control device
X	....	X	....	isalpha	test for alpha character
.....	X	.....	.....	isatty	test for terminal
X	....	X	....	isdigit	test for digit
X	....	X	....	islower	test for lower case
X	....	X	....	isspace	test for white space
X	....	X	....	isupper	test for upper case
X	....	X	....	itoa	convert integer to ASCII string
.....	X	.....	.....	kill	send a signal to a process
.....	X	.....	.....	link	link to a file
X	....	X	....	longjmp	non-local goto
.....	X	.....	X	lseek	seek to a position in a file
X	....	X	....	malloc	allocate memory
X	....	X	....	max	return the maximum of two values
X	....	X	....	min	return the minimum of two values

INTROL-C/6809 STANDARD LIBRARIES (cont'd)

STA09 LIB.	UFX09 LIB.	FLX09 LIB.	OS909 LIB.	FUNCTION	DESCRIPTION
.....	X	.....	.....	mknod	make special file or directory
X	....	X	....	modf	return fractional part of a float
.....	X	.....	.....	mount	mount a file sub-system
X	....	X	....	movmem	copy a block of memory from one location to another
.....	X	.....	.....	nice	change program priority
.....	X	.....	X	open	open a file
.....	X	.....	.....	pause	stop until signal
.....	X	.....	.....	perror	print error message
.....	X	.....	.....	pipe	create an inter-process channel
X	....	X	....	printf	formatted output conversion
.....	X	.....	.....	profil	profile a process
.....	X	.....	.....	ptrace	process trace
X	....	X	....	putc	write a character to a file
X	....	X	....	putchar	write a character to the standard input
X	....	X	....	puts	write a string to standard output
.....	X	.....	X	read	read from a file
.....	.....	.....	X	readln	read a line from a file
X	....	X	....	reverse	reverse a string in place
.....	X	.....	X	rewind	reset specified file to beginning
X	....	X	....	rindex	find last occurrence of character
X	....	X	....	sbrk	allocate memory
X	....	X	....	scanf	formatted input conversion
.....	.....	X	....	send	send a signal to a process
.....	X	.....	.....	setgid	set group user id
X	....	X	....	setjmp	non-local goto
.....	.....	.....	X	setstat	set status of file or device
.....	X	.....	.....	setuid	set the effective user id
.....	X	.....	.....	signal	catch or ignore signals
.....	X	.....	X	sleep	suspend execution of process
X	....	X	....	sprintf	formatted output conversion
X	....	X	....	sscanf	formatted string conversion
.....	X	.....	.....	stat	get file status
.....	X	.....	.....	stime	set the system time and date
X	....	X	....	strcat	copy string
X	....	X	....	strcmp	compare strings lexicographically
X	....	X	....	strcpy	copy string
X	....	X	....	strlen	return string length
X	....	X	....	strncat	copy string
X	....	X	....	strncmp	compare strings lexicographically
X	....	X	....	strncpy	copy string
X	....	X	....	strsave	save string in memory
.....	X	.....	.....	stty	get status of file or device
.....	X	.....	.....	sync	update all disks
.....	X	.....	.....	syscall	C system call interface
.....	X	.....	.....	system	run a command string
.....	X	.....	X	tell	return the current file position
.....	X	.....	X	time	return the system time
.....	X	.....	.....	times	get process times
X	....	X	....	tolower	convert to lower case
X	....	X	....	toupper	convert to upper case
X	....	X	....	uldiv	unsigned long integer divide
X	....	X	....	ulmod	unsigned long modulo operation
X	....	X	....	ulmul	unsigned long multiply
.....	X	.....	.....	umask	set the default file access bits
.....	X	.....	.....	umount	unmount a file sub-system
X	....	X	....	_unext	unextend float
.....	X	.....	X	ungetc	push character back on input stream
.....	X	.....	X	unlink	delete file
.....	X	.....	.....	wait	wait for child process to terminate
.....	X	.....	X	write	write to a file
.....	.....	.....	X	writeln	write a line to a file

NOTE: The STA09 Library is included with cross-compiler packages and with HLD cross-loader packages; the UNX09 Library with UC6809 compilers and with ULD cross-loader packages; the FLX09 Library with FC6809 compilers and with FLD cross-loader packages; the OS909 Library with OC6809 compilers and with OLD cross-loader packages.

Introl Corporation \* 647 W. Virginia St. \* Milwaukee, Wisconsin 53204 U.S.A.  
Telephone (414) 276-2937



INTROL-C BINARY SOFTWARE LICENSE AGREEMENT  
(Without Maintenance Option)

Introl Corp, (hereinafter called Licensor), for and in consideration of the terms and conditions set forth herein, and for a one-time license fee, hereby grants to Licensee, and Licensee accepts a personal, non-exclusive, non-transferrable license to use the binary software programs named below (hereinafter referred to as Licensed Programs) subject to the following terms and conditions:

DEFINITIONS: "Developed Programs" means any compiled or assembled program created by Licensee through use of the Licensed Programs, including the object code generated by the Runtime Library which is supplied as part of the Licensed Programs.

LICENSE: The Licensed Programs are supplied by Licensor solely for Licensee's internal business use on a single Designated CPU, identified below. This use includes the right for Licensee to construct Developed Programs using the Licensed Programs, and to sell, give away, or otherwise distribute the object code generated by the Runtime Library in creating these Developed Programs. Except as provided in the preceding sentence, all right, title, and interest in and to the Licensed Programs and all related materials, including all source code furnished by Licensor with the Licensed Programs, remains the sole and exclusive property of Licensor. Neither this Agreement, the Licensed Programs, or any portions thereof, may be sold, leased, assigned, sub-licensed, or otherwise transferred by Licensee, except as expressly provided herein, without prior written consent of Licensor.

TERM: This License shall begin on the date hereof and shall remain in effect only as long and during such period as Licensee complies with the terms and conditions specified in this Agreement. This License Agreement may be terminated by Licensor if Licensee fails to comply with any terms or conditions specified herein. This License Agreement shall automatically terminate upon any act of bankruptcy by or against Licensee, upon any assignment for the benefit, of creditors of the Licensee, upon any attachment, execution of judgement or process against Licensee or its assets, or upon dissolution of Licensee.

LIMITED PERMISSION TO COPY LICENSED PROGRAMS: Licensee shall not copy, in whole or in part, any Licensed Programs which are provided by Licensor in machine readable form except for use by Licensee on the Designated CPU or for backup or archival purposes. This applies to copies in any form and generated by any means. Licensee shall maintain appropriate records of the number and location of all copies of the Licensed Programs, or portions thereof, and shall make these records available to Licensor upon request thereof. The original and any copy of the Licensed Programs, in whole or in part, shall at all times be the sole and exclusive property of Licensor. Licensee shall reproduce the following copyright notice on all copies of the Licensed Programs, in whole or in part, in any form: "Copyright 1983 by Introl Corp. Reproduction or publication in any form prohibited. Property of Introl Corp.". Use of the copyright notice is not to imply that the Licensed Programs have been published.

PROTECTION AND SECURITY: Licensee shall not cause or permit disclosure of any Licensed Programs, in whole or in part, in any form, to any person other than Licensee's or Licensor's employees without prior written consent of Licensor. Licensee shall take all reasonable steps to safeguard the Licensed Programs so as to ensure that no unauthorized person has access to them, and that no unauthorized copies, in whole or in part, in any form, shall be made. Licensee expressly acknowledges that the Licensed Programs are confidential and proprietary property of Licensor and agrees to receive and maintain same as a confidential disclosure. Licensee further expressly acknowledges that unauthorized copying, use, or disclosure of the Licensed programs, in whole or in part, in any form, does great damage to Licensor and seriously impairs Licensor's ability to do business.

TERMINATION: Within thirty (30) days of termination of this Agreement for any reason, Licensee shall, at Licensee's option, either (a) return to Licensor all existing copies, in whole or in part, and their related materials, or (b) furnish to Licensor evidence satisfactory to Licensor that the original and all copies of the Licensed Programs, in whole or in part and in any form, have been destroyed.

DISCLAIMER OF WARRANTY: Licensor makes no warranties with respect to the licensed Programs. The licensed Programs are licensed 'as is' by Licensor, without warranty, and Licensor shall have no liability or responsibility to Licensee or any other person or entity with respect to any liability, loss, or damage caused or alleged to be caused directly or indirectly by the Licensed Programs.

LIMITATION OF LIABILITY: THE FOREGOING WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. LICENSEE FURTHER AGREES THAT LICENSOR SHALL NOT BE LIABLE FOR ANY LOST PROFITS, OR FOR ANY CLAIM OR DEMAND AGAINST LICENSEE BY ANY OTHER PARTY, EXCEPT AS PROVIDED HEREIN. IN NO EVENT SHALL LICENSOR BE LIABLE FOR CONSEQUENTIAL DAMAGES, EVEN IF LICENSOR HAS BEEN ADVISED OF, THE POSSIBILITY OF SUCH DAMAGES.

MISCELLANEOUS: This Agreement constitutes the entire agreement between Licensor and Licensee and supersedes all prior agreements and representations. Licensee agrees to hold Licensor harmless on all liability associated with Licensee's breach of this Agreement including, but not limited to, all reasonable attorney's fees. This Agreement shall be governed by the laws of the State of Wisconsin in the United States of America and Licensee expressly submits to jurisdiction therein by process served by mail on Licensee at its below business address. Licensee agrees to advise Licensor of all changes in Licensee's address. Licensor's main address is given below. If any provisions of this Agreement, or portions thereof, are invalid under any applicable statute or rule of law, they are to that extent deemed to be omitted. The signing of this Agreement constitutes acceptance of the terms of this Agreement. No provision in correspondence or on Purchase Orders shall in any way modify this Agreement. Licensor represents that it has sufficient right, title, and interest in and to the Licensed Programs to make this Agreement with Licensee.

Licensee Name: \_\_\_\_\_

Normal Business Address of Licensee:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Country: \_\_\_\_\_

Phone: \_\_\_\_\_

Licensed Program Name: \_\_\_\_\_ S/N \_\_\_\_\_

Purchased From: \_\_\_\_\_

DESIGNATED CPU:

Mfgr: \_\_\_\_\_ Model # \_\_\_\_\_ S/N \_\_\_\_\_

LICENSEE SIGNATURE: \_\_\_\_\_

Name And Title: \_\_\_\_\_

Date: \_\_\_\_\_

\*\*\*\*\*

\*\*\*\*\* ICC COMPILER \*\*\*\*\*

-a[t|d|b|s)=(loc) - Place 'text', 'data', 'bss', or 'string' type data under (loc) location counter  
-b=(directory) - Find current and subsequent C Compiler passes in (directory) location.  
-c - Override default condition with respect to generation of position dependent/independent code.  
-d - Override default condition with respect to generation of position dependent/independent data.  
-g(c) - use alternate version of preprocessor pass for compilation (RC6809 and OC6809 compilers only).  
-i=(directory) - Search (directory) location for #include files.  
-k - Display progress of compilation/assembly sequence on console.  
-m(name)[=(string)] - Define (name) in preprocessor, with value (string) optionally assigned to (name).  
-n - Inhibit execution of next compiler pass in the compilation sequence.  
-r - Save C Compiler's intermediate assembly language output file.  
-s - Disallow nested comments.  
-s=(size) - Set maximum size of triple buffer.  
-t=(directory) - Place C Compiler's temporary files in (directory) location.  
-y=[=(n)] - Strip all identifiers to a maximum length of (n) characters.  
-z Interpret "\n" (newline) characters as being carriage returns.

\*\*\*\*\* R09 RELOCATING ASSEMBLER \*\*\*\*\*

-a - Place all symbols except those beginning with a "?" character in the object file.  
-c - Send Assembler's output listing to console.  
-i - Include all included files in output listing.  
-j - Include symbols beginning with a "?" character in the symbol table listing.  
-l=(filename) - Place output listing in specified file.  
-n - Do not produce an output listing.  
-o=(filename) - Assign name to Assembler's relocatable output module.  
-q=(class) - Assign numeric class identifier (0 through 255) to relocatable output module.  
-s - Suppress listing of the symbol table.  
-u - Force all undefined symbols to default to imported symbols.  
-x - Don't generate an object module.  
-z - Delete input file when Assembler has finished using it.

\*\*\*\*\* ILINK LINKER \*\*\*\*\*

-b - Do not search the default Standard Library.  
-c=(file) - Get additional link-time parameters from command file.  
-d[(c)] - Call optional cross-loader named "(c)LD" when Linker finishes.  
-e=(symbol) - Set entry point.  
-f(string) - Search additional Standard Library named "lib(string).R"  
-l[s][x][u][=(file)] - Produce a linker output listing.  
-m=(symbol) - Define primary function naming symbol.  
-n - Inhibit Linker from automatically calling Loader.  
-o=(file) - Assign name to output file.  
-p[(c)] - Pipe Linker's output to loader.  
-r - Save Linker's output file (during automatic link-and-load operations).  
-s - Strip output file of all non-entry-defined symbols.  
-t=(classlist) - Use (classlist) classes of modules during linking process, if they are available.

\*\*\*\*\* HLD LOADER \*\*\*\*\*

-a=(seg);(placernent)[,(seg);(placement)] - Set segment memory bound (segment may begin, or end, at a specific memory location, or specified to immediately follow, or immediately precede, another segment).  
-c=(file) - Get additional parameters from command file.  
-g=(type) - Set output format (Motorola S Record, Intel Hex, Tek Hex, or Extendend Tek hex format).  
-h - Define EOL character to be carriage return (rather than newline character).  
-l[s][=(file)] - Produce a Loader output listing.  
-o=(name) - Assign name to output file.  
-u=(seg) - Place uninitialized data in specified segment.  
-v[(char)] - Modify Loader's symbol changing procedures for symbols beginning with non-alpha characters.  
-w - Produce executable output file no matter what.  
-x (type):(ext) - Set output filename extension for specified type of hex output format.  
-z - Delete loader's input file when Loader has finished using it.

\*\*\*\*\* ULD LOADER \*\*\*\*\*

-a=(sec):(seg)[,(seg)] - Assign location counter segment to UniFlex program section (text, data. or bss).  
-c=(file) - Get additional parameters from command file.  
-l[s]=[=(file)] - Generate loader output listing.  
-o=(name) - Assign name to output file.  
-v=(size) - Set stack section size.  
-w - Produce an executable output file no matter what.  
-x[=(pagesize)] - Produce output file in UniFlex segmented format.  
-y=(origin) - Set text section origin at specified location.  
-z - Delete Loader's input file when Loader has finished using it.

\*\*\*\*\* FLD LOADER \*\*\*\*\*

-a=(sec):(Seg)[,(seg)] - Assign a location counter segment to a Flex program section (text, data, or bss).  
-c=(file) - Get additional parameters from command file.  
-l[S]=[=(file)] - Produce an output listing.  
-o=(name) - Assign name to output file.  
-w - Generate executable output file no matter what.  
-y[t|d|b|]=(origin) - Set origin for text, data, or uninitialized section of output file.  
-z - Delete input file after Loader has finished using it.

\*\*\*\*\* OLD LOADER \*\*\*\*\*

-a=(sec):(seg)[,(seg)] - Assign a location counter segment to an OS9 program section (text, data, or bss)  
-c=(file) - Get additional parameters from command file.  
-l[s]=[=(file)] - Produce an output listing.  
-o=(name) - Assign name to output file.  
-V=(size) - Set stack section size.  
-w - Generate an executable output file no matter what.  
-x - Place executable program module and data initialization information module in separate files.  
-z - Delete the input file after the Loader has finished using it.

\*\*\*\*\* LIBMAN LIBRARY MANAGER COMMANDS \*\*\*\*\*

a (file),(module)[,(class)] - Add module to library; create new library.  
d (module)[,(class)] - Delete module from library.  
r (file),(module)[,(class)] - Replace module in library.  
q - Quit Library Manager 7after saving library file being edited).  
omit - Exit Library Manager (without saving edited file).  
l (module)[,(class)] - List information on named file.  
sl (module)[,(class)] - List abbreviated information on named file.  
h - Provide on-line help.  
lo (file) - Explicitly load a library file.  
ll (file) - List a loaded library.  
sll (file) - Provide abbreviated listing of a loaded library.  
s (file) - Save library using the filename indicated by (file).  
c (file) - Get additional commands from named command file.  
e (strings) - Echo specified strings to the terminal.  
f (module)[,(class)] - Find named module.  
p (module)[,(class)] - Print information for named module.  
sp (module)[,(class)] - Print abbreviated listing of information for named module.  
i (file),(module)[,(Class)] - Insert named module in library so it precedes current module.

.....

INTROL CORPORATION

647 West Virginia Street \*\* Milwaukee, Wisconsin 53204 U.S.A.

Telephone (414) 276-2937