

A New Work Item Proposal: C++ for Embedded Systems (Embedded C++)  
The Embedded C++ Technical Committee

Proposal:

The Embedded C++ Technical Committee will propose to set a new sub-working group in the WG21 working group in order to establish a new authorized subset of the C++ which applies to embedded systems programming. The Embedded C++ Technical Committee will submit its Embedded C++ specification as a base document for the discussions. Further more, the committee will be willing to prepare Technical Report based on our specification if desired by the ISO.

1. Background

1-1 Shifting from C to C++ is obvious even for Embedded Programming

Very Popular Visual C++

Object oriented methodology is widely introduced.

US embedded programmers are already using C++

(According to the Nikkei BP research in 1996, 15% of embedded system engineers are using C++ )

1-2 Some problems in C++

"Unexpected" code may be generated.

For programmers, it's difficult to avoid dangerous (untested) code.

For programmers, the programming may be restricted as for embedded.

1-3 Requirement from embedded system programmers

Necessary and sufficient enough functions for embedded programming.

Keeping the spirit of C++ as much as possible.

Compilers with good quality.

No or very few "unexpected" code.

2. Scope of the Embedded Systems

All systems other than native applications on PCs and workstations.

Basic industrial equipment such as telephone switchboards.

Industrial control equipment such as robots.

Plant process control systems and building control systems.

Electric equipment for office automation and home-use.

Table 1 shows typical memory size used in relatively small systems.

3. Design Philosophy of the Embedded C++

3-1 Complete subset of ISO/ANSI C++

Not a dialect and no derivative should be added. The goal is a pure subset of the ISO-C++.

3-2 Avoid complexity

For the average C programmer to understand when switches to C++.

3-3 Embedded Systems

Need to fulfil the basic restrictions of the embedded systems:

Real-time systems

Poor resource systems

Cost sensitive systems

Avoid practically unpredictable overhead in both time and memory.

4. Toward the real world wide standard

4-1 Some problems with document

As the EC++ spec is described as amendments to the final CD-2, it is not a kind document . We feel it necessary to describe along with the main strategy of the ISO document.

4-2 Some Dialect problems

As we stated in this document, because of the wide range of the embedded systems, based on the EC++ idea, it is possible to define another EC++. It will cause confusions among engineers who wish to use the EC++.

The Embedded C++ Technical Committee wishes to submit the idea of setting a subset from the ISO C++ in order to wide use among embedded systems engineers with proper document and guidelines. We do not wish to compete nor challenge to the ISO standard activity but wish to cooperate. The Embedded C++ Technical Committee does insist that the activity for setting a subset for embedded programming should be designed by both the language experts and the experts from the embedded systems industry. Within this basic mutual understanding we can create a real standard namely de fact and ISO. For this purpose, the Embedded C++ Technical Committee willingly to submit our specification to the ISO as a base document for the standardization discussions.

Table 1

Application	Current	Next	Future	
CAMERA	48K-64K	64-128K	96K-256K	Optical
Rice Cooker	16K-48K	16K-64K	64K	4-8bit MCU
PHS	128K-	192K-256K	256K-	
Cellular Phone	384K-	512K-	768K-	
Printers	32K-64K	32K-96K	64K-128K	Ink-Jet
Internet-TV	2M-4M	4M-	8M-	RAM=1/2-1/4 of ROM
Regular TV	16K-48K	16K-64K	32K-96K	
Handy VCR	192-256K	256-294K	320K-	
HDD	32K-64K	64K-96K	64K-128K	

Typical RAM size of each application is 1/4 to 1/9 of ROM size.  
Resource : NEC Semiconductor Application Engineering Division

- +All features must be cleanly and elegantly integrated into the language.
- +It must be possible to use features in combination to achieve solutions that would otherwise have required extra, separate features.
- +There should be as few spurious and "special-purpose" features as possible.
- +A feature should be such that its implementation does not impose significant overheads on programs that do not require it.
- +A user need only know about the subset of the language explicitly used to write a program.

From the C++ Programming Language 2nd edition

## Appendix A The Embedded C++ Technical Committee

### A-1 Basic Policy forming the committee

Promoted as a voluntary work of members and advisors.  
Open a window widely to anyone.

### A-2 Short History

Began December 1995  
1st international meeting at Tokyo, March 1996  
1st Draft June 1996  
WWW August 1996 (<http://www.caravan.net/ec2plus/>)  
3rd International Meeting at San Jose, September 1997

### A-3 The Committee

The members (alphabetical order)

FUJITSU LIMITED  
Hitachi, Ltd.  
Japan Novel Corporation  
MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.  
MITSUBISHI ELECTRIC SEMICONDUCTOR SOFTWARE CORPORATION  
NEC Corporation  
Nihon Cygnus Solutions  
Nihon Motorola Ltd.  
TOSHIBA CORPORATION  
Yokogawa Digital Computer Corporation

The advisors

Mr. Craig Franklin (Green Hills Software Inc.)  
Dr. P.J. Plauger (Dinkumware, Ltd.)  
Dr. Thomas Plum (Plum Hall Inc.)  
Dr. H. Takada (Toyohashi University of Technology)

Secretariat

ADVANCED DATA CONTROLS, CORP. .

## Appendix B Current Results

### B-1 Available Commercial Products

Edison Design Group, Inc.	Compiler Front End product with EC++ option
Green Hills Software, Inc.	Compilers, Debuggers
Cygnus Solutions	GNU based Compilers, Debuggers
Dinkumware, Ltd.	Libraries

### B-2 Why accepted?

Reasonably small specification  
Bridge from C programmer to C++ programmer  
Low compiler implementation hurdle  
Based on the requirement from the embedded industry  
Effective even for those large systems such as exchange machines or  
factory management systems.  
Voluntary and Non profitable activity  
No charge is required to use the spec.