

ICS 35.060
CCS L 74



**NATIONAL STANDARD OF THE
PEOPLE'S REPUBLIC OF CHINA**

GB/T 30997-2014

**Programming languages-C -- Extensions to support embedded
processors**

编程语言 C 支持嵌入式处理器的扩展

(English Translation)

Issued on: September 3, 2014

Implemented on: February 1, 2015

Issued by: Standardization Administration of China

Table of Contents

Introduction

Introduction

1 Scope

2 Normative references

1 3 Compliance

4 Fixed-point arithmetic

4.1 Overview of fixed-point data types and principles

4.2 Pairs of ISO/IEC 9899:1999 change Detailed

5 Name and address space named register storage class

5.1 Name and Address Space Overview Principle

Named register storage class 5.2 Summary and Principle

5.3 Pairs of ISO/IEC 9899:1999 30 detailed changes

6 Basic I/O hardware addressing

6.1 Principle

6.2 Terminology

6.3 Basic I/O hardware addressing header \u003ciohw.h\u003e

6.4 Of the I/O register

6.5 Pairs of ISO/IEC 9899:1999 40 detailed changes

Appendix A (informative) fixed-point arithmetic

A.1 fixed-point data types

Foreword

This standard was drafted in accordance with GB/T 1.1-2009 given rules.

This standard uses the translation method identical with ISO/IEC Technical Report ISO/IEC TR18037:2008 "support embedded programming language C the processor expansion ", made the following editorial changes.

--- To increase the scope of the standard description;

--- Meets the requirements of the original made the following editorial changes. text statements "Because this is a technical report, so there is no Compliance requirements, implementors are free to choose those specifications they need.

(AsthisisaTechnicalReportthere

are no conformance requirements and implementers are free to select those specifications that they need) "into our national standard after, should not have this expression, it will be deleted;

--- Include the original text of ISO/IEC 9899:1999 editorial changes to the programming language C section entry for this standard itself Article Head to distinguish the ISO/IEC 9899:1999 C programming language entry plus a solid frame to distinguish;

--- Remove the informative Appendix G of the Technical Report 18037 second edition updates and changes.

This standard by the National Information Technology Standardization Committee (SAC/TC28) and focal points.

This standard was drafted. China Electronics Standardization Institute, Fudan University, Shanghai Key Laboratory of Computer Software Testing.

Introduction

In the fast-growing market for embedded systems, such as the use of writing applications demand high-level language like C language is increasing.

Basically, the reasons for this trend, there are two. embedded systems become more complex procedures (assembly language would be difficult to maintain) and insert Lifecycle-in system processor models in shorter (which implies more frequent application to re-adapt to the new set of instructions). C language the level of programming that can be obtained on code reuse is considered an important step forward to solve the above problems.

Many areas of technology defines a function provided by the processor (such as DSP), the use of these functions in embedded systems is not easy it is written in C applications. For example, fixed-point operations, the use of different memory space and the underlying I/O operation. Current mention We can only solve these cases only the technical field a small part of the problem.

Embedded processor is often used to analyze analog signals, and the data received via the docking using filtering algorithms to process these signals.

Typical applications can be found in all of the wireless devices. Common data types used in the filtering algorithm is fixed-point data type, and Moreover, in order to achieve the necessary speed, embedded processors often with special fixed-point devices. Since the C language (according to ISO/IEC 9899:1999 in the definition) does not provide support for fixed-point arithmetic operations, leading to the present can not be selected because the programmer had to make Write most of its algorithms in assembly language manual. This standard specifies a definition for the C language within a certain range of accuracy and saturation Fixed-point data types. Optimizing C compiler can be like for integer and floating

point data as easily generate efficient fixed-point data types Code.

Many embedded processor has a plurality of different memory bank and requires the data packets according to different libraries in order to achieve maximum performance. example for example, to ensure for FIR filter design concurrent data streams and collaborative data into the processor multiplier/accumulator, the operation of the processor Essential. In order to allow the programmer must be removed from the statement that a particular data object's memory space, this standard provides for more than Basic support address space. Thus, optimizing compiler that can take advantage of the ability to support multiple processor address space, for example, in a Cycle to read data from two separate memory for maximum execution speed.

With the C language these years matured, language was added to the expanded access to a variety of basic I/O hardware (iohw) register to overcome Defects language. Today, almost all of the stand-alone environment and embedded systems C compiler supports a number directly from the C source level Access iohw registers. However, these extensions between different C language "dialect" is still inconsistent.

This standard provides a method for the codification of common practices and provides a single, unified syntax for basic iohw register addressing.

The standard recommendation with ISO/IEC 9899:1999 used in combination.

C programming language support the expansion of embedded processor

1 Scope

This standard specifies the ISO/IEC 9899:1999 series of extension of the C programming language. These extensions to support embedded processors.

The standard of each chapter dealing with a specific topic. 4,5,6 The first chapter contains the technical characteristics of the subject matter described Said. These provisions provide an overview and does not contain all the details. The last chapter contains for ISO/IEC 9899:1999 series Editorial changes to the programming language C, the provisions of these modifications fully ISO/IEC 9899:1999 related topic is necessary, and to provide a Complete definition. Additional explanation and principles listed in the Appendix.

This standard applies to the use of C language for embedded processor application development process.

2 Normative references

The following documents for the application of this document is essential. For dated references, only the dated version suitable for use herein Member. For undated references, the latest edition (including any amendments) applies to this document.

ISO/IEC 9899:1999 Programming Language C (ISO/IEC 9899:1999-Programminglanguages-C)

3 Compliance

This standard is divided into three chapters expressed the basic norms of three independent set of functions (fixed-point arithmetic in Chapter 4, Chapter 5, named address spaces and life Name register storage class, and Chapter 6, the basic input/output hardware addressing). If you implement the function of a chapter, the chapter recommends that implementors implement Full functionality.

4.1.1 Data Types

The standard, fixed-point data value refers to small value (a value between -1.0 to 1.0) or by the integer part and a fractional part Value. Because of the decimal point position is known, the value of these types of operations, able to operate with the same efficiency integer real Now. Its value to the typical use of fixed-point data operations, can convert analog values into digital representation of some filtering algorithms After the application in Preface found. For more information about fixed-point data types, see A.1.1.

This standard for the C language adds two fixed-point data types. fract type and accum type. No data values fract type Integer part, so the value fract type between -1.0 and 1.0. accum type range depending on the data type integer the number of digits.

Fixed-point data type specified by the appropriate keywords and new type specifier `_Fract` and `_Accum`. These type specifier and has Some type specifier short, long, signed, unsigned combination can be used to specify the following 12 point type.

`unsignedshort_Fract unsignedshort_Accum unsigned_Fract unsigned_Accum
unsignedlong_Fract unsignedlong_Accum`