GCC-HCS12 Project Contract Proposal

CSE4080: Computer Science Project
Faculty of Science & Engineering
York University

Course Director: Dr. Uyen T Nguyen
Department of Computer Science & Engineering
Office: Computer Science and Engineering Building, CSE 2024
Phone: (416) 36-2100 x 33274
E-mail: utn@cse.yorku.ca

Project Director: Dr. Mokhtar Aboelaze
Department of Computer Science & Engineering
Office: Computer Science and Engineering Building, CSE 2026
Phone: (416) 36-2100 x 40607
E-mail: aboelaze@cse.yorku.ca

Student: Navid Mohaghegh – 206238984
E-mail: CS231381@cse.yorku.ca

Table of Contents

Background

Embedded systems

An embedded system, as oppose to general purpose computer, is a special purpose system in which the computer is completely encapsulated by or dedicated to the device or system it controls. Unlike a general-purpose computer, such as a personal computer, an embedded system performs one or a few predefined tasks, usually with very specific requirements. Since the system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product. Embedded systems are often mass-produced, benefiting from economies of scale.

Free software foundation

Proprietary software is software with restrictions on using, copying and modifying as enforced by the proprietor. Restrictions on use, modification and copying is achieved by either legal or technical means and sometimes both. Technical means include releasing machine-readable binaries to users and withholding the human-readable source code. Legal means can involve software licensing, copyright, and patent law.

Exclusive legal rights to software by a proprietor are not required for software to be proprietary, since public domain software and software under a permissive license can become proprietary software by distributing compiled versions of the program without offering the source code. Proprietary software's restrictions make it an antonym of free software.

For free software, the same laws used by proprietary software are used to preserve the freedoms to use, copy and modify the software. Proprietary software includes free-ware and shareware. It can be commercial software, but public domain and all other free software can also be sold for a price and be used for commercial purposes.

According to the Free Software Foundation (FSF), proprietary software is any software that does not meet its definitions of free software or semi-free software. The term's literal meaning covers software that has an owner who exercises control over what users can do with it. One license of the FSF's, the GNU General Public License (GPL), asserts that the restrictions of free software offer computer users freedom while the restrictions of other software benefit only the owner and are unethical.

Proponents of proprietary software, like Microsoft, argue that innovation is driven more quickly when it is lucrative. They claim that the best way to ensure this motivation is to tie revenue to innovation. The proprietor uses a temporary monopoly with copyright and sometimes software patents that makes the software more expensive. A dependency on future versions and upgrades can make the monopoly

permanent without the emergence of a competing software package, a situation termed "vendor lock-in". Proprietary software is said to create greater commercial activity over free software, especially in regard to market revenues.

A variety of activation or license management systems are emerging in proprietary software that prevent copyright infringement and determine how the software is used. If the proprietor ceases to exist or for any other reason does not provide keys for activation or to unlock discontinued products, legitimate users can be unable to re-activate existing software or use other hardware.

If the proprietor of a software package should cease to exist, or decide to cease or limit production or support for a proprietary software package, recipients and users of the package can be left at a disadvantage and have no recourse if problems are found with the software. Proprietors can fail to improve and support software because of business problems. Companies also end their support for a product for business and organizational planning purposes. The consequence is also tied to enticing more to upgrade and pay for newer versions.

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc. is an semiconductor manufacturer created by the divestiture of the Semiconductor Products Sector of Motorola in 2004. Freescale focuses on the automotive and embedded and communications markets for their semiconductor products. Freescale is among the Worldwide Top 20 Semiconductor Sales Leaders.

The 68HC12 (6812 or HC12 for short) is a 16-bit micro-controller family from Freescale Semiconductor. Originally introduced in 1994, the architecture is an enhancement of the previous Freescale 68HC11. Programs written for the HC11 are usually compatible with the HC12, which has a few extra instructions. The first 68HC12 derivatives had a maximum bus speed of 8MHz and flash memory sizes up to 128kbytes.

Like the 68HC11, the 68HC12 has two 8-bit accumulators A and B (sometimes referred to as a single 16 bit register D), also two 16-bit registers X and Y, a 16-bit program counter, a 16-bit stack pointer and an 8 bit Condition Code Register.

Motorola has launched the new HCS12 (also known as MC9S12) product line in 2000. The bus speed was improved up to 25MHz and flash sizes up to 512kbytes. The MC9S12NE64 was introduced by Freescale in September 2004, claiming to be the "industry's first single-chip fast-Ethernet Flash micro-controller." It features a 25 MHz HCS12 CPU, 64K bytes of FLASH EEPROM, 8K bytes of RAM, and an Ethernet 10/100 Mbit/s controller. Since the HCS12 was a direct upgrade to the existing HC12 family,

most of the links and information provided here were suitable for both lines.

Latest addition in 2004 was the advanced HCS12X, providing even more features, including the XGATE DMA co-processor. HCS12X is fully backward-compatible with HCS12 CPU. The S12X family utilizes the latest process technology. It boasts higher speed (40 MHz), more functionality, reduced power consumption and cost, and enhanced performance with the new XGATE on-chip memory-management and DMA module. The XGATE peripheral co-processor allows for some tasks to be offloaded from the CPU also allows several new instructions to increase performance.

Freescale announced the MC9S12XEP100 in May 2006 to further extend the S12X family to 50MHz bus speed and add a Memory protection unit (based on segmentation) and a hardware scheme to provide Emulated EEPROM.

Short Project Description

Since the introduction of 68HC812A4 in 1996, there has been an increasing demand for information about HC12 development tools, software, sample projects and technical resources. There are many modular hardware implementations that brings HCS12 features for engineers and educators. However the current HCS12 Integrated Development Environment offered from Freescale, known as CodeWarrior Studio, is a proprietary software which is only available on Microsoft Windows systems. In addition, due to licensing issues it is not possible to compile more than 32KB of C or 1KB of C++ code in this IDE. These limitations highly restrict the academic users to either buy the expensive commercial versions or restrict their code size and development.

Unfortunately, the current open source attempts in Linux environments are absolutely experimental and users are completely by their own. And usually users' time are being wasted by reading tremendous amount of unorganized documentation instead of focusing on the project itself.

In this project, we focus on providing GPL type development environment with various examples and guidelines for easily developing HCS12 applications. GNU compilers will be introduced and explained how they can be patched and be ready for HCS12 platform. At the end, we demonstrate a simple real-time operating system for HCS12 platform. Our goal is to focus on development process and provide step by step and quick manual for each task. Below are the hardware target platforms will be covered in this project:

- HC11 series (outdated and will not be covered)
- HC12 series (covered briefly as they are replacing by new HCS12 series)
- S12 series (covered in details)
- S12X series (covered briefly due to budget limitations)

The following training and prototyping boards are preferred:

- DRAGON12 Development Board (based on S12 series: MC9S912DP256 chip)
- Adapt9S12XDP512M2 Module (based on S12X series: MC9S12XDP512 chip)
- S12X T-Board with extra 256KB RAM Add-On-Board (based on S12X series: MC9S12XDP512 chip)

Milestones

Project Initiation

I1. Research on different project scenarios (2007-04-27)
I2. Meetings with potential supervisors (2007-05-01)

Definition

D1. Defining scope of the project (2007-05-02)
D2. Determine deliverables, constraints (2007-05-02)

Planning

P1. Brainstorming and project plans (2007-05-02)
P2. Distribute works (2007-05-05)
    Contract proposal and project scope due - (2007-05-11)
P3. Detailed work breakdown and scheduling (2007-05-11)

Execution

E1. Purchase target development platforms (2007-05-11)
E2. Make the GCC-HCS12 website (2007-05-12)
E3. Develop an IDE for GCC-HCS12 project (2007-06-16)
E4. Develop small documents and guidelines for different component of HCS12 (2007-06-13)
E5. Develop sample projects for different component of HCS12 (2007-06-23)
E6. Develop small real-time OS (2007-07-10)
E7. Introduction to FREE-RTOS for HCS12 platform (2007-07-20)

Equipments and Budget

In this project only GPL type softwares will be used. Linux is the operating system and all the other softwares which will be used are available under GPL terms. In terms of hardware, due to budget restrictions, DRAGON12 Development Board is going to be studied as it is available free of charge for this project (courtesy of York University).

Deliverables

R1. GCC-HCS12 website (2007-07-01)
R2. GPL licensed IDE for GCC-HCS12 project (2007-08-01)
R3. Complete documentation and guidelines for HCS12 component (2007-08-01)
R3. Various sample projects for HCS12 component (2007-08-01)
R4. Simple real-time OS for HCS12 (2007-08-01)
R5. Complete documentation for HCS12 flash programing (2007-08-01)
R6. Project demonstration (example application will be Modular DC to AC inverter) (2007-08-01)

Names and Signatures

Title	Name	Signature	Notes
Project Director	Dr. Mokhtar Aboelaze		
Student	Navid Mohaghegh		Student number: 206238984 E-mail: cs231381@cse.yorku.ca
Course Director	Dr. Uyen T Nguyen		

Resources

[1] MC9S12DP256 based BDM with D-bug12:

http://www.evbplus.com/Dragon_BDM/dragon_bdm.html

[2] DRAGON12 Development Board:

http://www.evbplus.com/dragon12_hc12_68hc12_9s12_hcs12.html

[3] MiniDRAGON+ Development Board:

http://www.evbplus.com/minidragonplus_hc12_68hc12_9s12_hcs12.html

[4] Oliver Thamm's HC12 Web:

http://elmicro.com/hc12web/index.html

[5] HCS12X T-Board:

http://elmicro.com/en/hcs12tb.html

[6] Technological Arts Adapt9S12XDP512M2 XGATE MCU Module:

http://www.technologicalarts.ca/catalog/product_info.php?cPath=50_154_155&products_id=367

[7] Tom Almy's Free Simulator for 68HC12:

http://hcs12text.com/freesim.html

[8] Linux Devices:

http://linuxdevices.com/

[9] GCC tool chain for 68HC12:

http://m68hc11.serveftp.org/m68hc11_inst_ptc.php

[10] GCC tool chain for 68HC12 - FAQ:

http://m68hc11.serveftp.org/wiki/index.php/FAQ:Link

[11] Developing Embedded Software in C Using ICC11/ICC12/Metrowerks:

http://users.ece.utexas.edu/~valvano/embed/toc1.htm

[12] CodeWarrior for HCS12(X) Microcontrollers:

http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=CWS-H12-STDED-CX

[13] EmbeddedGNU IDE for Windows:

http://www.ericengler.com/EmbeddedGNU.aspx

[14] Freescale S12X chip:

http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=S12XE&nodeId=0162468636bJwn

[15] More resources on 68HC12:

http://www.mgtek.com/miniide/resources/hc12.aspx

[16] Wikipedia, the free encyclopedia:

http://en.wikipedia.org/