Proposal for GSoC 2020 Arduino: Port FreeRTOS to Portenta

About Me

Name: Zhihong Niu

Email: a6813140@hotmail.com

GitHub Account: MRNIU

Timezone: UTC+8:00

EDUCATION: Xi'an Shiyou University, Xi'an China, Computer science , expected graduation July 2021

PULL REQUESTS: arduino-libraries/Keyboard#35, rewrite the official keyboard library.

Abstract

Porting FreeRTOS for Portenta.

Different from the general situation, we have two cores and belong to the AMP(asymmetric multiprocessor architecture). So we have to consider the cooperation between the two cores. There are three main aspects: task scheduling, interrupt management, and memory sharing. Fortunately, FreeRTOS provides a reference example. Our main job is to make small changes by referring to existing code.

Technical Details

We can use the Example given by FreeRTOS to achieve our goal. Referring to this example, combined with the official porting guide, we can roughly determine our working steps:

1. Use STM32CubeMX to generate basic code, and test the hardware status. The directory structure is roughly as follows:

```
├── CM4
│   ├── Inc
│   │   ├── main.h
│   │   ├── stm32h7xx_hal_conf.h
│   │   └── stm32h7xx_it.h
│   └── Src
│       ├── main.c
│       ├── stm32h7xx_hal_msp.c
```

```
│       └── stm32h7xx_it.c
├── CM7
│   ├── Inc
│   │   ├── main.h
│   │   ├── stm32h7xx_hal_conf.h
│   │   └── stm32h7xx_it.h
│   └── Src
│       ├── main.c
│       ├── stm32h7xx_hal_msp.c
│       └── stm32h7xx_it.c
├── Common
├── Drivers
└── EWARM
```

2. Add FreeRTOS files to the project directory, including configuration files, FreeRTOS header files, interface files, etc.

```
├── CM4
│   ├── Inc
│   │   ├── FreeRTOSConfig.h
│   │   ├── main.h
│   │   ├── portmacro.h
│   │   ├── stm32h7xx_hal_conf.h
│   │   └── stm32h7xx_it.h
│   └── Src
│       ├── main.c
│       ├── port.c
│       ├── stm32h7xx_hal_msp.c
│       └── stm32h7xx_it.c
├── CM7
│   ├── Inc
│   │   ├── FreeRTOSConfig.h
│   │   ├── main.h
│   │   ├── portmacro.h
│   │   ├── stm32h7xx_hal_conf.h
│   │   └── stm32h7xx_it.h
│   └── Src
│       ├── main.c
│       ├── port.c
│       ├── stm32h7xx_hal_msp.c
│       └── stm32h7xx_it.c
├── FreeRTOS-Kernel
│   ├── include
│   │   ├── FreeRTOS.h
│   │   ├── StackMacros.h
│   │   ├── atomic.h
│   │   ├── croutine.h
│   │   ├── deprecated_definitions.h
│   │   ├── event_groups.h
```

```
│   │   ├── list.h
│   │   ├── message_buffer.h
│   │   ├── mpu_prototypes.h
│   │   ├── mpu_wrappers.h
│   │   ├── portable.h
│   │   ├── projdefs.h
│   │   ├── queue.h
│   │   ├── semphr.h
│   │   ├── stack_macros.h
│   │   ├── stdint.readme
│   │   ├── stream_buffer.h
│   │   ├── task.h
│   │   └── timers.h
│   └── src
│       ├── croutine.c
│       ├── event_groups.c
│       ├── heap_4.c
│       ├── list.c
│       ├── queue.c
│       ├── stream_buffer.c
│       ├── tasks.c
│       └── timers.c
├── Common
├── Drivers
└── EWARM
```

3. Modify FreeRTOSConfig.h

   FreeRTOSConfig.h is used to configure the kernel. We need to set the same configuration for the two MCU. The configuration information obtained from the official is as follows:

```
/*
 * FreeRTOS Kernel V10.0.1
 * Copyright (C) 2017 Amazon.com, Inc. or its affiliates.  All Rights Reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining a copy of
 * this software and associated documentation files (the "Software"), to deal in
 * the Software without restriction, including without limitation the rights to
 * use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
 * the Software, and to permit persons to whom the Software is furnished to do so,
 * subject to the following conditions:
 *
```

```
 * The above copyright notice and this permission notice shall be included in all
 * copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
 * FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
 * COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
 * IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
 * CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
 *
 * http://www.FreeRTOS.org
 * http://aws.amazon.com/freertos
 *
 * 1 tab == 4 spaces!
 */

#ifndef FREERTOS_CONFIG_H
#define FREERTOS_CONFIG_H

#if defined(__ICCARM__) || defined(__CC_ARM) || defined(__GNUC__)
    #include <stdint.h>
    extern uint32_t SystemD2Clock;
    void vGenerateM4ToM7Interrupt( void * xUpdatedMessageBuffer );
#endif

#define configUSE_PREEMPTION                    1
#define configUSE_IDLE_HOOK                     0
#define configUSE_TICK_HOOK                     0
#define configCPU_CLOCK_HZ                      ( SystemD2Clock )
#define configTICK_RATE_HZ                      ( ( TickType_t ) 1000 )
#define configMAX_PRIORITIES                    ( 7 )
#define configMINIMAL_STACK_SIZE                ( ( uint16_t ) 128 )
#define configTOTAL_HEAP_SIZE                   ( ( size_t ) ( 20 * 1024 ) )
#define configMAX_TASK_NAME_LEN                 ( 16 )
#define configUSE_TRACE_FACILITY                1
#define configUSE_16_BIT_TICKS                  0
#define configIDLE_SHOULD_YIELD                 1
#define configUSE_MUTEXES                       1
#define configQUEUE_REGISTRY_SIZE               8
#define configCHECK_FOR_STACK_OVERFLOW          0
```

```
#define configUSE_RECURSIVE_MUTEXES             1
#define configUSE_MALLOC_FAILED_HOOK            0
#define configUSE_APPLICATION_TASK_TAG          0
#define configUSE_COUNTING_SEMAPHORES           1
#define configGENERATE_RUN_TIME_STATS           0

#define configUSE_CO_ROUTINES                   0
#define configMAX_CO_ROUTINE_PRIORITIES         ( 2 )

#define configUSE_TIMERS                        0
#define configTIMER_TASK_PRIORITY               ( 2 )
#define configTIMER_QUEUE_LENGTH                10
#define configTIMER_TASK_STACK_DEPTH            ( configMINIMAL_STACK_SIZE * 2 )

#define INCLUDE_vTaskPrioritySet                1
#define INCLUDE_uxTaskPriorityGet               1
#define INCLUDE_vTaskDelete                     1
#define INCLUDE_vTaskCleanUpResources           1
#define INCLUDE_vTaskSuspend                    1
#define INCLUDE_vTaskDelayUntil                 1
#define INCLUDE_vTaskDelay                      1
#define INCLUDE_xQueueGetMutexHolder            1
#define INCLUDE_xTaskGetSchedulerState          1
#define INCLUDE_eTaskGetState                   1

#ifdef __NVIC_PRIO_BITS
 #define configPRIO_BITS                        __NVIC_PRIO_BITS
#else

#define configPRIO_BITS                         4
#endif

#define configLIBRARY_LOWEST_INTERRUPT_PRIORITY     0xf

#define configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY  5

#define configKERNEL_INTERRUPT_PRIORITY     ( configLIBRARY_LOWEST_INTERRUPT_PRIORITY << (8 - configPRIO_BITS) )

#define configMAX_SYSCALL_INTERRUPT_PRIORITY  ( configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY << (8 - configPRIO_BITS) )

#ifndef __IASMARM__
    void vAssertCalled( const char *pcFile, const uint32_t ulLine );
#endif /* __IASMARM__ */
#define configASSERT( x ) if( ( x ) == 0 ) { vAssertCalled( __FILE__, __LINE__ ); }
```

```
96
97  #define vPortSVCHandler    SVC_Handler
98  #define xPortPendSVHandler PendSV_Handler
99  #define xPortSysTickHandler SysTick_Handler
100
101 #define sbRECEIVE_COMPLETED( pxStreamBuffer )
    vGenerateM4ToM7Interrupt( pxStreamBuffer )
102
103 #endif /* FREERTOS_CONFIG_H */
```

4. Modify stm32h7xx_it.c

   FreeRTOS has already implemented the three functions `SVC_Handler`, `PendSV_Handler`, and `SysTick_Handler` . We can delete these three functions in stm32h7xx_it.c.

5. Modify main.c

   Perform related tests.

6. Debug

References：

portenta-h7

FreeRTOS-porting-guide

STM32H7_Dual_Core_AMP_RTOS_demo

simple-multicore-core-to-core-communication-using-freertos-message-buffers

mcu-mpu-embedded-software/stm32-embedded-software/stm32cube-mcu-mpu-packages/stm32cubeh7

Schedule of Deliverables

Community Bonding Period

Set up a development environment and wait for the portenta to arrive.

Continue to optimize the solution and go to the Arduino and FreeRTOS communities to find possible help.

Discuss the scheme details and summarize the issues that cannot be identified before starting work.

Phase 1

- Determine the directory structure, run without MCU collaboration, and write preliminary usage documents.

	Content	Note
Week 1	Familiar with workflow and complete basic testing	
Week 2	Add FreeRTOS code to the project and passing the compile	
Week 3	Write functions that may cause problems, such as clocks and interrupts + Unit test	
Week 4	Debug for M4+Unit test	

Phase 2

- FreeRTOS have a good running on Portenta

	Content	Note
Week 1	Debug for M7 + Unit test	
Week 2	Debug two MCUs in collaboration	
Week 3	Debug two MCUs in collaboration + Integration test	
Week 4	Test + Write documentation	

Final Week

Check the commit information, code style, comments, etc. again, and submit the PR according to Related Requirements

Development Experience

On GitHub, my personal project, SimpleKernel, an x86 operating system project, received 700+ stars and 70+ forks.

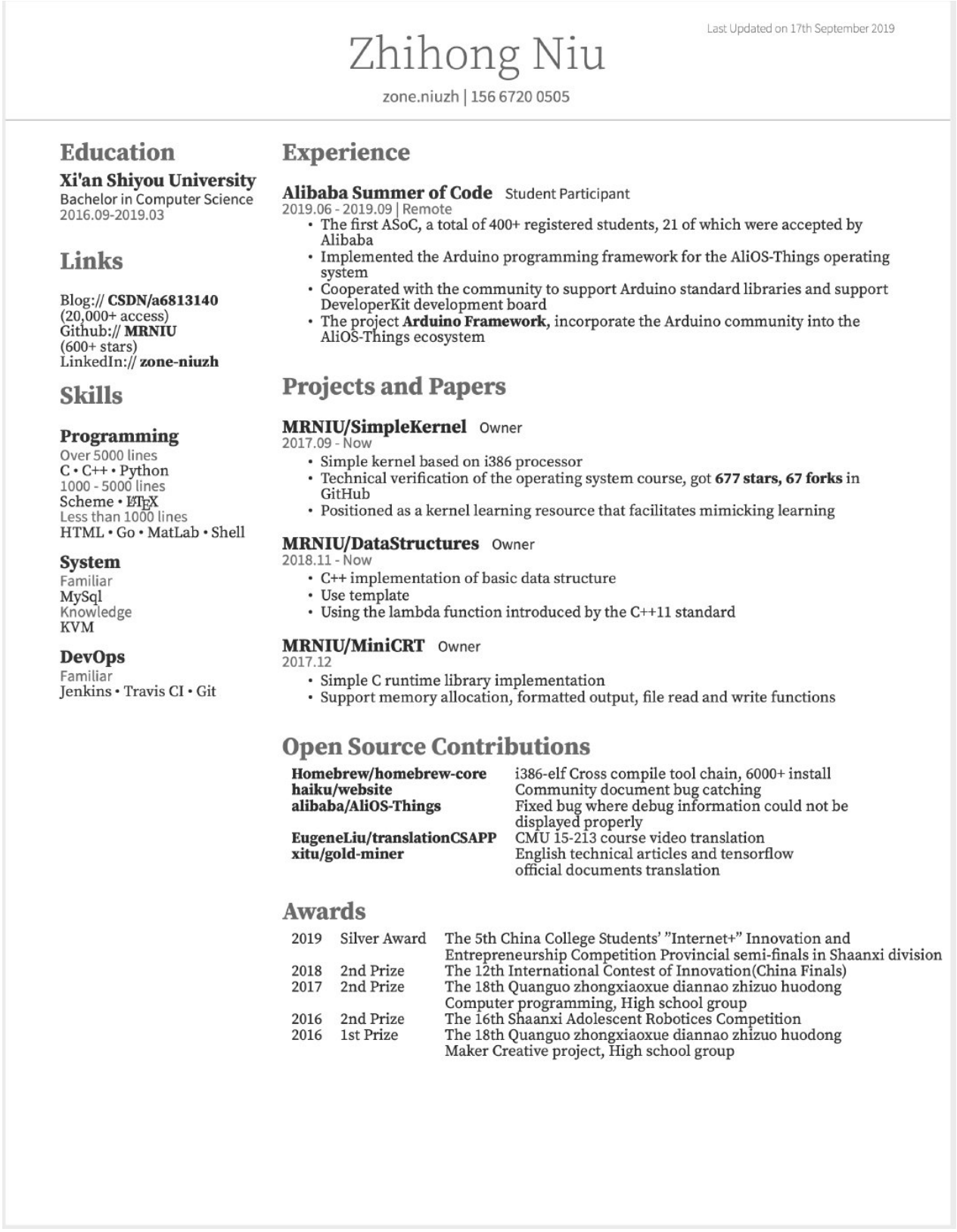
In addition, I have contributed in many communities, such as Homebrew, AliOS-Things.

My work at AliOS-Things is related to Arduino. AliOS-Things is an open source IoT real-time embedded operating system launched by Alibaba, which supports multiple architectures. They have a developer board that integrates Arduino pins, but does not have support for Arduino programming framework. My job is to add a set of Arduino programming framework, including Arduino basic functions, such as digitalWrite, pinMode, and several official libraries: EEPROM, SD,

Wire, etc.

As early as high school, I started using Arduino to make some interesting things. In 2017, me and my team used Arduino Mini and a six-axis acceleration sensor(MPU6050) to make a gesture control device. We got The 18th Quanguo zhongxiaoxue diannao zhizuo huodong 1st prize in Maker Creative project, High school group.

Other Experiences



Last Updated on 17th September 2019

Zhihong Niu

zone.niuzh | 156 6720 0505

Education

Xi'an Shiyou University
Bachelor in Computer Science
2016.09-2019.03

Links

Blog:// **CSDN/a6813140**
(20,000+ access)
Github:// **MRNIU**
(600+ stars)
LinkedIn:// **zone-niuzh**

Skills

Programming
Over 5000 lines
C • C++ • Python
1000 - 5000 lines
Scheme • LaTeX
Less than 1000 lines
HTML • Go • MatLab • Shell

System
Familiar
MySql
Knowledge
KVM

DevOps
Familiar
Jenkins • Travis CI • Git

Experience

Alibaba Summer of Code Student Participant
2019.06 - 2019.09 | Remote
- The first ASoC, a total of 400+ registered students, 21 of which were accepted by Alibaba
- Implemented the Arduino programming framework for the AliOS-Things operating system
- Cooperated with the community to support Arduino standard libraries and support DeveloperKit development board
- The project **Arduino Framework**, incorporate the Arduino community into the AliOS-Things ecosystem

Projects and Papers

MRNIU/SimpleKernel Owner
2017.09 - Now
- Simple kernel based on i386 processor
- Technical verification of the operating system course, got **677 stars, 67 forks** in GitHub
- Positioned as a kernel learning resource that facilitates mimicking learning

MRNIU/DataStructures Owner
2018.11 - Now
- C++ implementation of basic data structure
- Use template
- Using the lambda function introduced by the C++11 standard

MRNIU/MiniCRT Owner
2017.12
- Simple C runtime library implementation
- Support memory allocation, formatted output, file read and write functions

Open Source Contributions

Homebrew/homebrew-core	i386-elf Cross compile tool chain, 6000+ install
haiku/website	Community document bug catching
alibaba/AliOS-Things	Fixed bug where debug information could not be displayed properly
EugeneLiu/translationCSAPP	CMU 15-213 course video translation
xitu/gold-miner	English technical articles and tensorflow official documents translation

Awards

2019	Silver Award	The 5th China College Students' "Internet+" Innovation and Entrepreneurship Competition Provincial semi-finals in Shaanxi division
2018	2nd Prize	The 12th International Contest of Innovation(China Finals)
2017	2nd Prize	The 18th Quanguo zhongxiaoxue diannao zhizuo huodong Computer programming, High school group
2016	2nd Prize	The 16th Shaanxi Adolescent Robotices Competition
2016	1st Prize	The 18th Quanguo zhongxiaoxue diannao zhizuo huodong Maker Creative project, High school group

Why this project?

There are three points

1. I really like Arduino because it accompany me through three years of high school
2. The new board has strong performance and requires a mature system for management, that is FreeRTOS
3. I'm interested in operating systems. When I wrote code for AliOS-Things last year, I started to

wonder what is the difference between a real-time operating system and a traditional PC operating system. This time Arduino provided an opportunity for me to learn. I can't pass up this chance.

Do you have any other commitments during the GSoC period?

No.