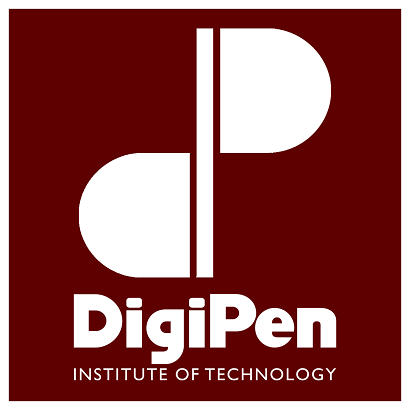


|  |
| --- |
| Snake!  Game Implementation on FreeRTOS and STM32F746ZG |
|  |
| November 10  Neil Cabrera  Capstone Report  CET341 - Real-Time Operating Systems for Embedded Microcontrollers |



Contents

[I. Introduction 3](#_Toc24320233)

[II. Hardware List 4](#_Toc24320234)

[Nucleo-F746ZG Specification 4](#_Toc24320235)

[Hardware Configuration 6](#_Toc24320236)

[Hardware Configuration Details 7](#_Toc24320237)

[Cortex M7 7](#_Toc24320238)

[Connectivity 7](#_Toc24320239)

[Security 7](#_Toc24320240)

[Extras 7](#_Toc24320241)

[III. Software Design 7](#_Toc24320242)

[FreeRTOS Tasks 7](#_Toc24320243)

[static void SystemInitTask( void\* pvParameters ) 7](#_Toc24320244)

[static void Led1Task( void\* pvParameters ); 7](#_Toc24320245)

[static void Led2Task( void\* pvParameters ); 7](#_Toc24320246)

[static void SnakeTask( void\* pvParameters ); 8](#_Toc24320247)

[Semaphores and Mutexes 8](#_Toc24320248)

[IV. Testing 9](#_Toc24320249)

[Idle CPU Utilisation 9](#_Toc24320250)

[V. References 10](#_Toc24320251)

# Introduction

This document describes the implementation of the classic Nokia game “Snake” on STM32F746ZG with the following requirements:

* The primary computing platform must be the provided STM32F746 board set;
* The final demo must work in the Release Mode;
* Users should be able to give input signals, such as by joystick, or camera, or bytes from a PC via UART;
* The LCD must be used to display either text, static photo or dynamic video/animation;
* FreeRTOS must be used;
* The Idle task utilization is less than 20%;
* No timing deadline violation;
* All proposed projects must be different;

# Hardware List

## Nucleo-F746ZG Specification

This project implementation was based on the STM32 Nucleo-144 development board (Nucleo-F746ZG) with STM32F746ZGT6 MCU which provides an affordable and flexible way for users to try out new concepts and build prototypes with the STM32 microcontroller.

This board does not require any separate probe, as it integrates the ST-LINK/V2-1 debugger/programmer and it comes with the STM32 comprehensive software HAL library, together with various packaged software examples.

Microcontroller Features

• STM32F746ZGT6 in LQFP144 package

• ARM 32-bit Cortex-M7 + FPU + Chrom-ART (Adaptive Real-Time) Accelerator

• 216 MHz max CPU frequency

• VDD from 1.7 V to 3.6 V

• 1 MB Flash

• 320 KB SRAM

• GPIOs (168, LQFP208 package?) with external interrupt capability

• 12-bit ADCs with 24 channels (3)

• 12-bit DAC channels (2)

• USART/UART (8)

• I2C (4)

• SPI (6)

• Advanced-control Timer (2)

• Low-power Timer (1)

• General Purpose Timers (12)

• Watchdog Timers (2)

• CAN (Controller Area Network, bxCAN) 2.0B active (2)

• SAI (2)

• RNG (Random Number Generator)

• USB 2.0 OTG FS (Full Speed, 12 Mb/s)

• USB 2.0 OTG HS (High Speed, up to 480 Mb/s)

• Ethernet MAC (Media Access Controller) interface with dedicated DMA and IEEE 1588 support

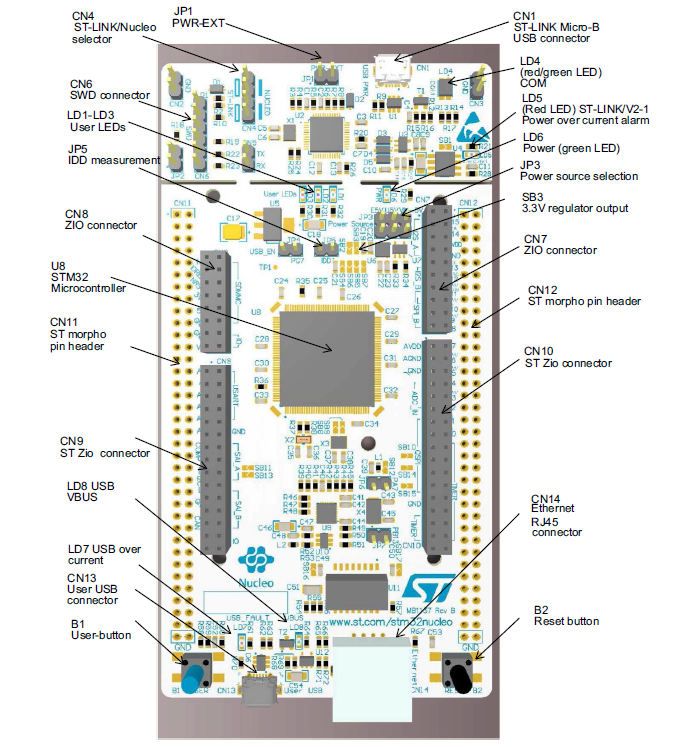
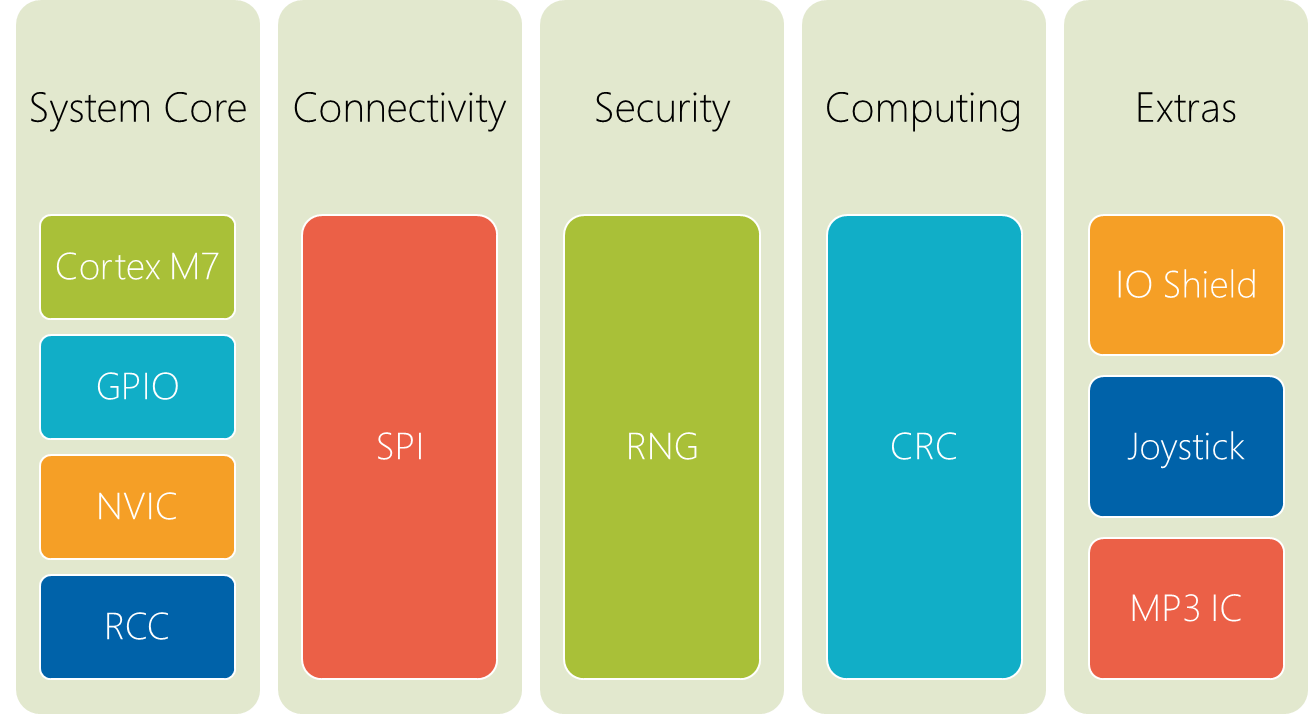


Figure 2.1 Layout of the Nucleo-F746ZG

## Hardware Configuration



## Hardware Configuration Details

### Cortex M7

* Set to run at maximum frequency of 216 Mhz
* ART, D-Cache and I-Cache were all enabled

### Connectivity

* SPI was used to interface with ST7735R TFT-LCD

### Security

* Used RNG to generate random coordinates for the snake food to appear

### Extras

* Used joystick to provide input for the Snake Movement as well as to Start the game
* A separate board with built-in MP3 player was used to provide sound effects to the game

# Software Design

## FreeRTOS Tasks

### static void SystemInitTask( void\* pvParameters )

* This task is the Snake Game Initialisation. It sets up the LCD, UI and opening music. Once the user presses start using the Navigation button, it will create the other tasks listed below.

### static void Led1Task( void\* pvParameters );

* This task is the Game Running indicator task. It is useful for debugging purposes if the LCD and IO shield is not attached. It indicates whether the game was successfully initialized and running.

### static void Led2Task( void\* pvParameters );

* This task is the Game Over indicator task. It is useful for debugging purposes if the LCD and IO shield is not attached. It indicates whether the game was successfully initialized and running.

### static void SnakeTask( void\* pvParameters );

* This task is responsible for game play and drawing the snake in real time. The direction of the snake is dependent on the user input. There is a default direction at the beginning of the game.
* The user can change the direction of the snake using the navigation joystick. When user presses the joystick, an interrupt is triggered which makes changes to the snake direction.
* The snake is then rendered block by block and will speed up its movements the more it consumes food.
* This task also calls the random food generator which places the snake food at random places in the screen where the snake body is not present.

**static void ScoreTask( void\* pvParameters );**

* This task is responsible for monitoring the game progress in real time and has higher priority than the snake task in order to avoid the score being out of sync with the actual snake progress.

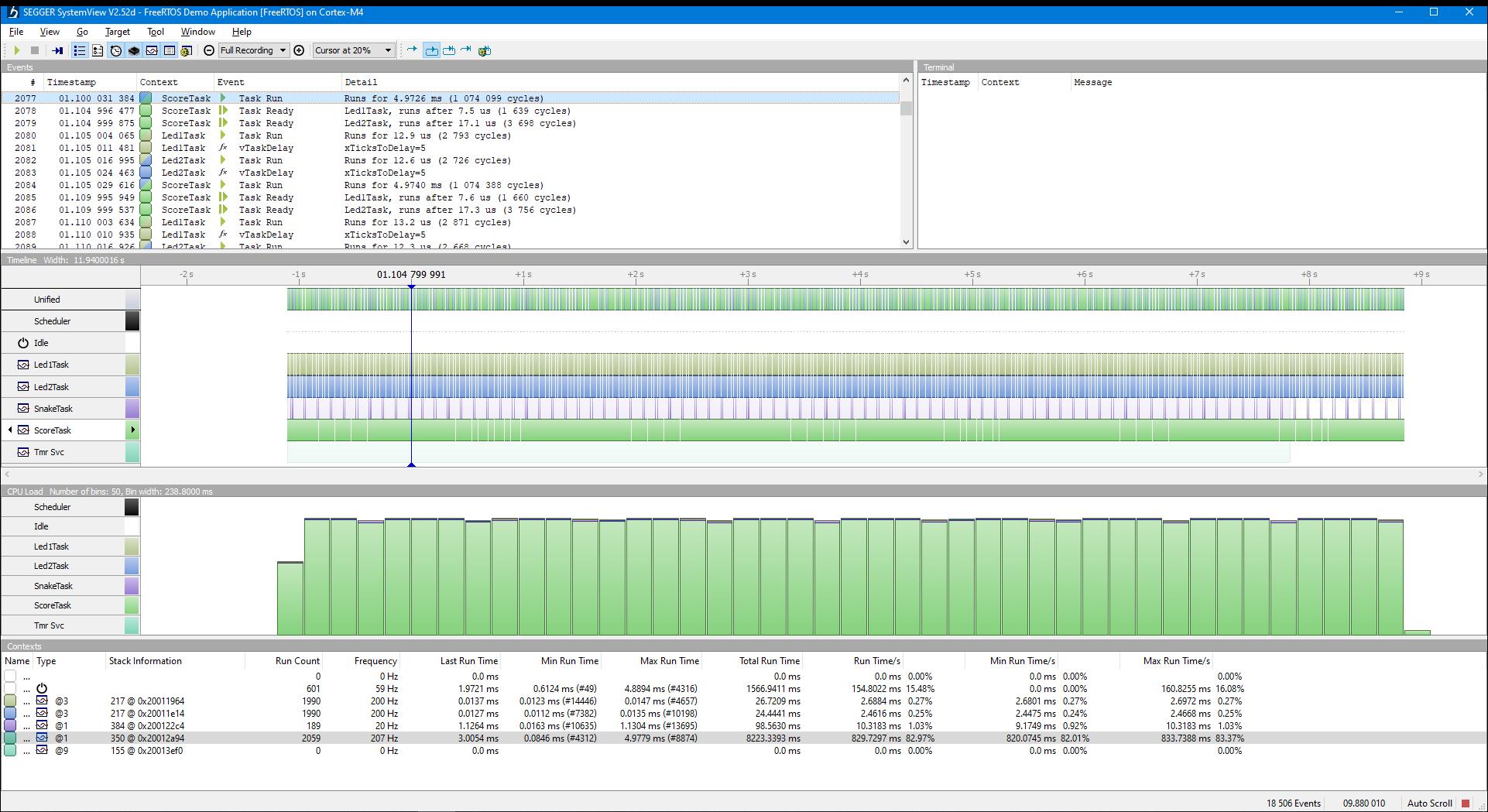
## Semaphores and Mutexes

Semaphores and Mutexes were not used for this project.

# Testing

## Idle CPU Utilisation

* One of the Design goals for this project is to make sure that the Idle Task is using less than 20% of the CPU time.
* Enabling the Overflow feature of the Game helps to do this as that feature is quite CPU intensive.
* Having a real time game monitoring task also helps to increase the CPU utilization and decrease the idle time.
* SystemView was used to check the Idle CPU time. The project must be running in debug mode for this to work.

****

# References

STM32F746ZG

<https://www.st.com/en/microcontrollers-microprocessors/stm32f746zg.html>