

Documentação de Projeto – Parte 2

Design, Estudo da Plataforma

Projeto: Space Invaders

Autor: Luan Carlos Klein

Versão: 13-Dezembro-2021

Parte 2a – Design

1 Introdução

O presente documento tem por objetivo apresentar de maneira detalhada a forma da construção do sistema a ser desenvolvido, o qual foi apresentado anteriormente nos documentos de CONOPS, especificação e domínio do problema. Neste documento, será apresentado detalhes de como a solução será construída, trazendo informações técnicas para o desenvolvimento do projeto, sendo ele uma parte fundamental da documentação e desenvolvimento do projeto.

2 Arquitetura Funcional

A Figura 1 apresenta o diagrama da arquitetura funcional do sistema a ser desenvolvido. Este diagrama é o mesmo apresentado nos documentos anteriores, pois trata de maneira suficiente o funcionamento do sistema.

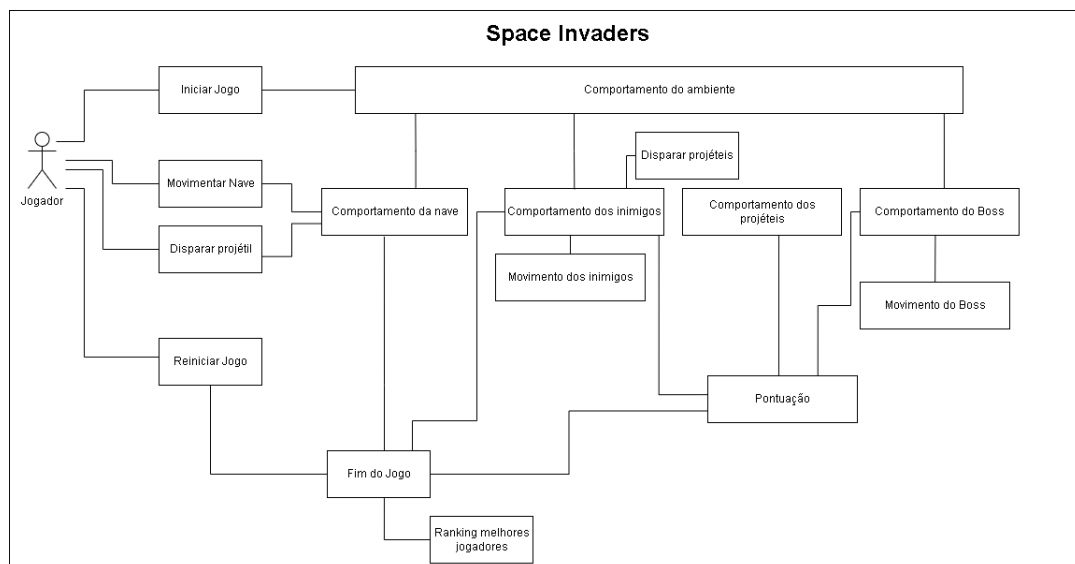


Figura 1: Diagrama da arquitetura funcional do sistema.

3 Arquitetura Física

A Figura 2 apresenta o diagrama da arquitetura física do sistema. Observa-se que os principais componentes físicos que serão utilizados são os botões, a tela LCD e seu *touchscreen* (e todas as respectivas interrupções relacionadas a eles). Além disso, será utilizada uma *event_flags* para controlar os disparos do jogador: apenas um projétil pode estar ativo por vez. A Figura 3 apresenta um diagrama complementar a Figura 2, que auxilia no entendimento da construção da solução.

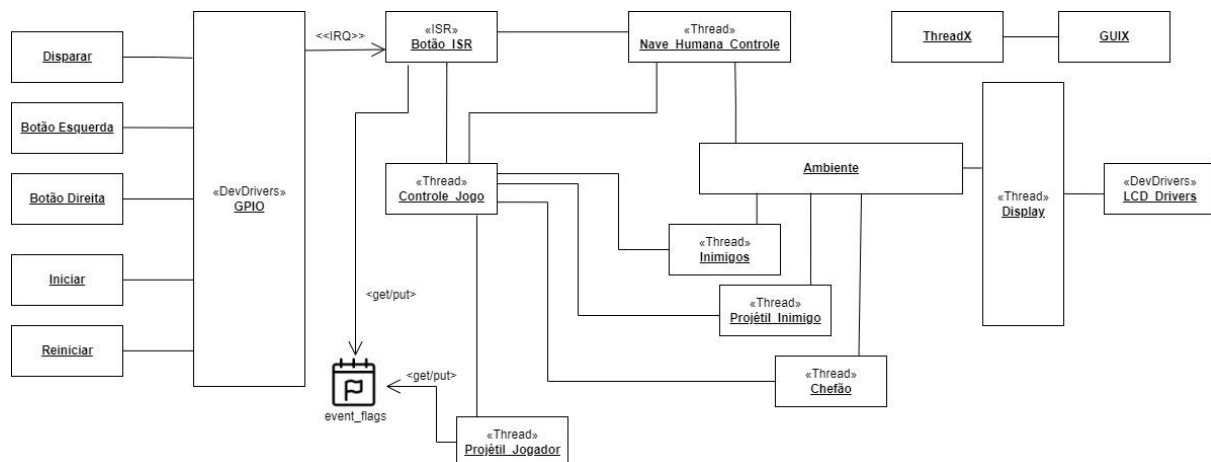


Figura 2: Diagrama da arquitetura física do sistema.

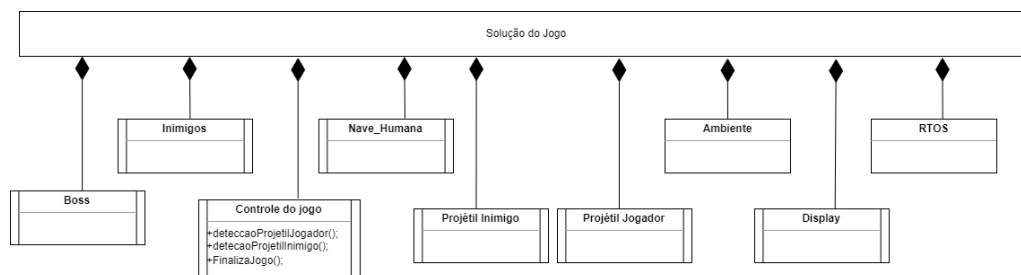


Figura 3: Diagrama complementar a arquitetura física do sistema.

4 Interface com o Usuário

A Figura 4 apresenta a solução da interface com o usuário. A interface consiste em 3 telas distintas:

- Tela Inicial: É a primeira tela que aparece para o jogador. Contém o botão para ir para o jogo;
- Tela do Jogo: Essa tela aparece quando o usuário clica no botão 'Start' na 'Tela Inicial' ou no botão 'Restart' na 'Tela Final'. Contém todas as imagens da nave do jogador, dos inimigos, do chefe e dos projéteis. Apresenta também a pontuação e a quantidade de vidas atuais do jogador. Essa é a tela do jogo propriamente dito;

- Tela Final: Essa tela aparece quando o jogador perde todas as suas vidas ou um dos inimigos chega até o fim da tela. Contém o ranking das melhores pontuações e um botão de 'Restart'.

Na Figura 4 são apresentadas as 3 telas, suas possíveis transições e um exemplo de como essas telas tendem a ser (em termos gráficos).

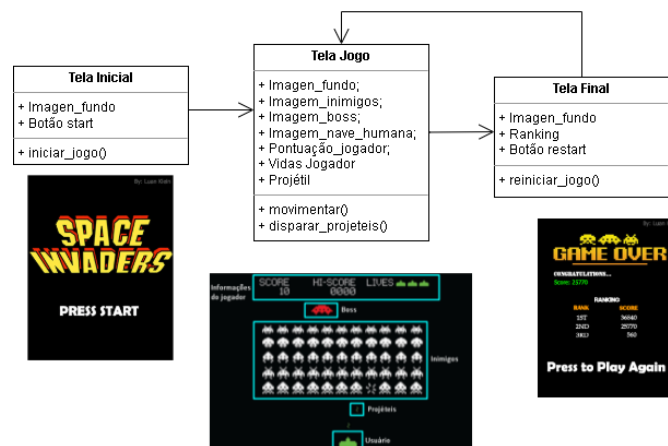


Figura 4: Interfaces com o usuário.

A Figura 6 apresenta o *hardware* que estará disponível para o jogador. De maneira simples, o usuário terá a disposição 2 botões (que servirão para mover a nave horizontalmente), um display LCD (que irá apresentar as telas citadas anteriormente) e nesse mesmo display, também será utilizado seu *touchscreen* (possibilitando que o usuário interaja com o jogo, por exemplo clicando em start e disparando projéteis).

5 Mapeamento da Arquitetura Funcional à Arquitetura Física

A Figura 5 apresenta a tabela com o mapeamento entre a arquitetura Física com a Arquitetura Funcional. Nela, as linhas são os elementos físicos enquanto que as colunas são os elementos funcionais. Na interseção de uma linha e uma coluna, quando marcado um X, indica que aquele elemento físico faz parte da implementação daquela funcionalidade.

Funcionalidades: →	Iniciar Jogo	Movimentar Nave	Disparar Projétil (jogador)	Reiniciar Jogo	Comportamento da nave	Comportamento do ambiente	Comportamento dos inimigos	Disparar projéteis (inimigos)	Comportamento do Boss	Pontuação	Comportamento dos projéteis	Fim de Jogo	Ranking dos melhores jogadores
Elementos Arquitetura Física													
GPIO	x	x	x	x									
Botão_ISR	x	x	x	x									
Nave_Humana_Control		x	x		x	x				x			
Projétil jogador			x			x	x		x	x	x		
Ambiente					x	x	x		x		x		
Inimigos						x	x	x		x		x	
Projétil Inimigo						x		x			x		
Chefão						x			x	x		x	
Controle_Jogo	x		x	x		x	x	x		x	x	x	x
Display	x		x	x								x	x
LCD_Drivers	x		x	x								x	x
event_Flags			x								x		
ThreadX													
GUIX													

Figura 5: Mapeamento da arquitetura funcional a Arquitetura Física.

6 Arquitetura do Hardware

O sistema a ser construído deve ser desenvolvido utilizando o Kit SK S7-G2 da Renesas. Este é um kit com diversos componentes e que apresentam diversas funcionalidades distintas. Entretanto, uma vez que o sistema consiste no jogo *Space Invaders*, serão utilizadas apenas alguns desses componentes, que serão basicamente voltados para a interação física e lógica com o usuário. A Figura 6 apresenta uma imagem da placa em questão, com os elementos que serão utilizados em destaque.

Serão utilizados os dois botões na parte superior da Figura 6 (que terão seu comportamento tratados através de interrupções) e também a tela *touchscreen*, que além de apresentar as imagens para o usuário (como um display comum), também permitirá que ele interaja com o jogo através de toques na tela (como por exemplo para iniciar o jogo, disparar projéteis etc).

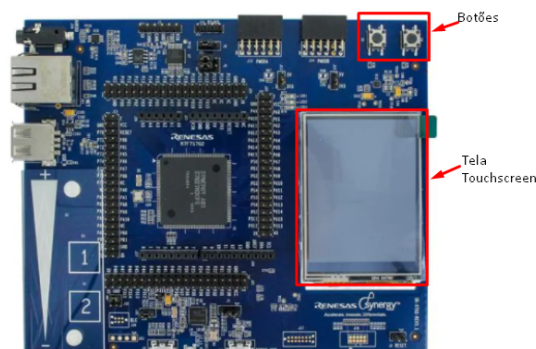


Figura 6: Kit Renesas SK S7-G2.

7 Design Detalhado

A Figura 7 apresenta o diagrama do *design* da solução a ser construída.

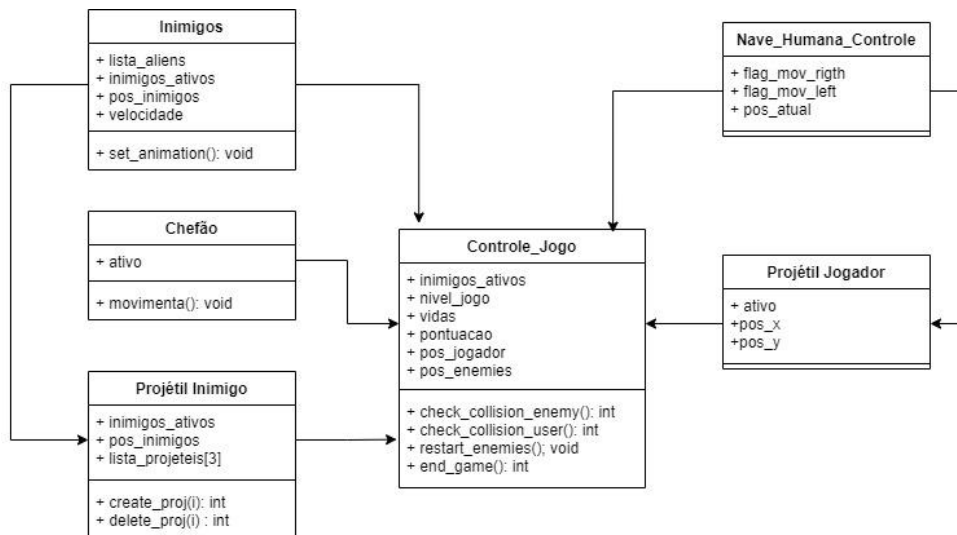


Figura 7: Design da solução do sistema.

Já a Figura 8 expõe, de maneira genérica, o diagrama de estados do sistema.

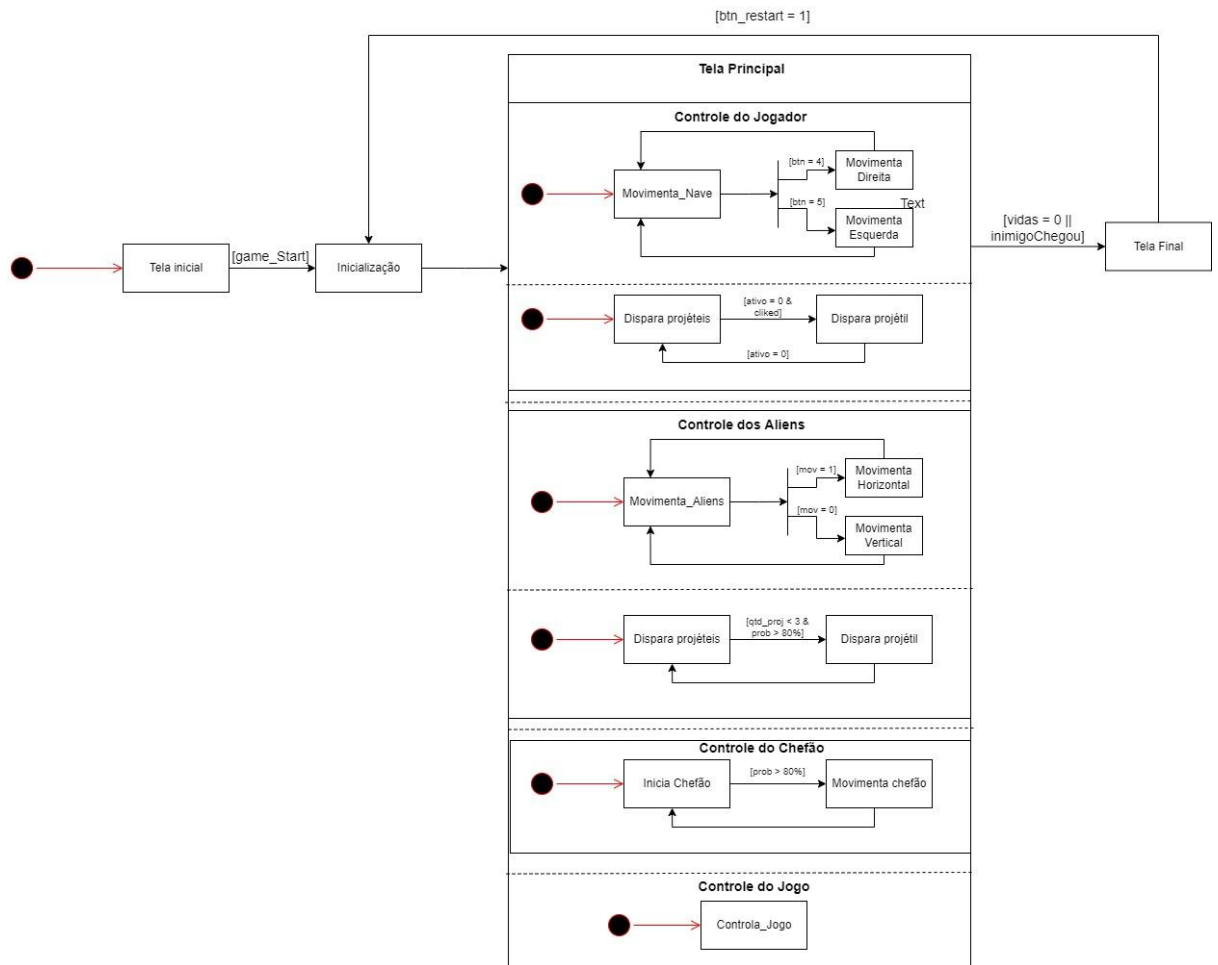


Figura 8: Diagrama de estados do sistema.

Parte 2b – Estudo da Plataforma

1 Estudo da Plataforma de Hardware

Para o estudo do hardware, será utilizado o manual de instruções disponibilizado pela própria Renesas¹. É possível obter uma visão geral da placa através de seu diagrama de blocos, que é representado na Figura 9 (disponível na página 2 do manual). Nele podemos observar de maneira simplificada os diversos elementos que compõem o kit. No atual projeto, as partes mais relevantes serão os *user buttons* e o LCD (e seu *touchscreen*).

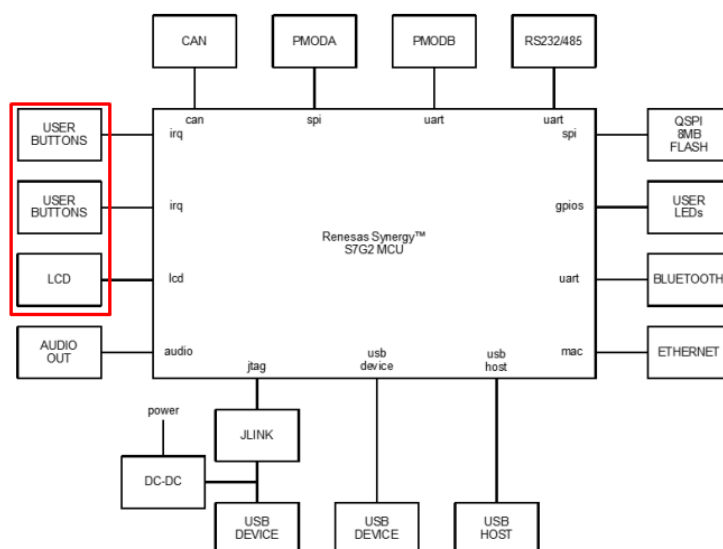


Figura 9: Diagrama de Blocos SK S7G2

As informações sobre os botões podem ser encontradas na página 10 do manual e seu esquemático está presente na Figura 10. Observa-se que há dois botões disponíveis para o usuário, chamados de S4 e S5 e que estão ligados diretamente aos pinos de GPIO P0.6 e P0.5. É possível gerar interrupções através desses botões, que serão geradas nos pinos P0.6 (P006) e P0.5 (P005) nos IRQ11 e IRQ10, respectivamente. Vale ressaltar que não há nenhum circuito voltado para o *debounce* (i.e. retirar o efeito de Bouncing dos botões mecânicos) nessas chaves, e portanto ao utilizar esses botões, se necessário realizar o *debounce*, ele deve ser feito via *software*.

¹ <https://www.renesas.com/us/en/document/mat/s7g2-starter-kit-sk-s7g2-users-manual?language=en>

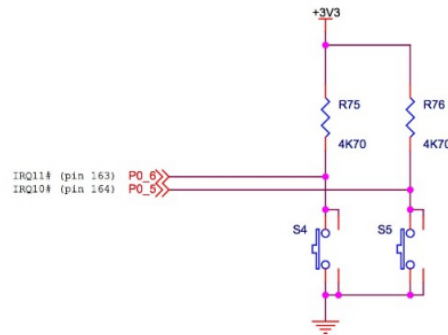


Figura 10: Circuitos dos botões disponíveis.

Já as informações sobre o LCD estão presentes na página 17 do manual. Ele consiste em um painel LCD QVGA 240x320 com interface para *touchscreen*. Este display se encontra conectado diretamente a porta para display do kit e as interações do *touchscreen* são realizadas via um controlador de tela sensível ao toque IC. A Figura 11 apresenta uma tabela com as funções disponíveis para o interação utilizando o *touchscreen*. Mais detalhes das funcionalidades do LCD, como por exemplo as suas funções, podem ser consultadas no manual.

S7G2 Pin	Function name
P609	RESET#
P004	IRQ9#
P512	SCL2
P511	SDA2

Figura 11: Tabela de funções do LCD *Touchscreen*.

Outro manual a ser estudado é o Renesas S7 Series Microcontrollers User's Manual². Nele, as informações relevantes para o projeto serão a visão geral (*overview*) do microcontrolador, a CPU, a geração de clock, o *Event Link Controller*, as portas de entrada e saída e a comunicação com as interfaces. O diagrama de blocos do microcontrolador, que fornece uma visão geral dele, está presente na página 10 do manual e na Figura 12, com os elementos citados acima em destaque.

² <https://www.renesas.com/us/en/document/dst/s7g2-microcontroller-group-datasheet>

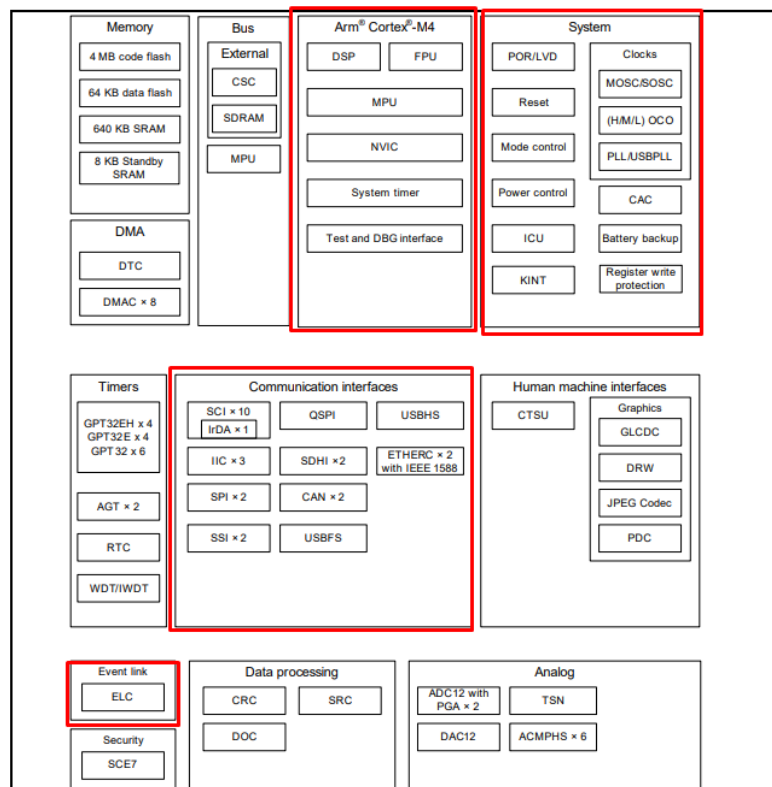


Figura 12: Diagrama de blocos do microcontrolador S7G2.

Observa-se que este microcontrolador possui um circuito de geração de clock bastante flexível e pode providenciar diferentes clocks para diferentes partes da MCU. Alguns limites devem ser observados:

- Flash clock (FCLOCK) - 60 MHz
- Core clock (ICLOCK) - 240 MHz
- PCLKA - 120 MHz
- PCLKB - 60 MHz
- PCLKC - 60 MHz
- PCLKD - 120 MHz

O *Event Link Controller* (ELC) conecta os eventos gerados pelos módulos periféricos a outros módulos periféricos. Essa comunicação não necessita da intervenção da CPU. Ressalta-se que o ELC é um módulo obrigatório na configuração do SSP. Ao tratar-se dos pinos de GPIO (*General Purpose Input/Output*) encontra-se as seguintes características:

- Input/Output (Entrada ou saída);
- Habilitação de pull-up de entrada;
- Capacidade de saída: baixa, média ou alta;
- Uso do pino para função analógica (ADC ou DAC);
- Uso do pino para função periférica (SPI);
- Algumas entradas são tolerantes a 5V;
- Pinos podem gerar interrupção e bodas do sinal de entrada.

2 Estudo da Plataforma de Software - SSP e e2Studio

O e2Studio é um ambiente de desenvolvimento integrado (IDE) baseado em Eclipse para MCUs Renesas. Além do editor de código do Eclipse, ele também oferece uma gama de outras funções. O e2Studio cobre todos os processos de desenvolvimento, desde a codificação e configuração até a depuração. Ele consiste em um software que possibilita criar e configurar projetos de maneira simples devido a sua interface gráfica, no qual é possível realizar essas tarefas com apenas alguns cliques. Informações mais detalhadas do funcionamento e suporte do e2Studio podem ser encontradas na página oficial da Renesas³.

Em conjunto com o e2Studio, também é possível utilizar o Synergy Software Package (SSP), que consiste em um conjunto de software integrado, otimizado e qualificado com um RTOS comercial que simplifica serviços complexos em nível de sistema. Existem diversas versões do SSP, sendo a mais recente a 2.1.0, a qual foi utilizada nesse projeto. Toda a documentação do SSP está disponível no site da Renesas⁴

A Figura 13 apresenta uma visão desses softwares. Observa-se que o SSP e e2Studio fazem parte de um mesmo módulo e podem trabalhar conjuntamente. Nessa figura todos os elementos em azul são os componentes do SSP (observa-se que há o HAL drivers, que facilita a comunicação e uso de alguns *drivers* e dois *frameworks* voltados para as interfaces e ThreadX).

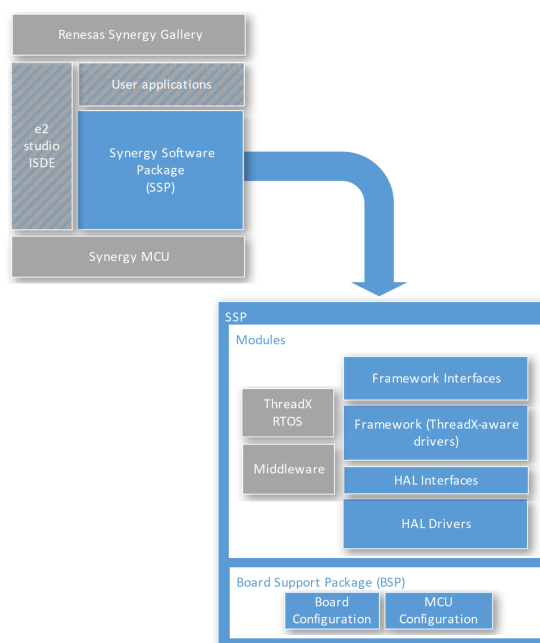


Figura 13: e2Studio e SSP. (Fonte: Manual do SSP).

³<https://www.renesas.com/us/en/software-tool/e-studio>

⁴<https://www.renesas.com/us/en/products/microcontrollers-microprocessors/renesas-synergy-platform-mcus/renesas-synergy-software-package>

3 Estudo do GUIX

O GUIX é uma implementação de tempo real de alta performance do GUI desenvolvido exclusivamente para aplicações embarcadas baseadas em ThreadX. Para o estudo dele, será utilizado um dos manuais disponibilizados pela Sinergy⁵. Diversos são os benefícios do uso do GUIX, dentre os quais estão: Uso de interface gráfica para aplicações de alta performance; disponibilidade de inúmeras funcionalidades e baixo uso de memória. Além disso, por se tratar de uma biblioteca, grande parte dos usos comuns de uma interface gráfica já estão implementados, bastando apenas chamar uma função para a execução da tarefa.

Uma vez dado que o presente projeto consiste em um jogo, há a necessidade de realizar animações na tela para a interação com o usuário. Com isso, destacam-se as seguintes funções que serão bastante úteis para toda a operação visual no jogo:

- `gx_widget_find`: Encontra um elemento (*widget*);
- `gx_widget_shift`: Muda a posição de um elemento de acordo com a posição atual dele;
- `gx_system_canvas_refresh`: Força o redesenho da tela (atualização da tela);
- `gx_widget_hide`: Esconde um elemento;
- `gx_widget_show`: Mostra um elemento.

Além dessas funções, há inúmeras outras disponíveis na página 64 do manual. Além disso, outras configurações de interação com o LCD, como configurações de portas, podem ser construídas com base no exemplo ‘GUIX Hello World for SK-S7G2’⁶. Esse processo é relativamente extenso, porém se faz necessário para que todas as interações ocorram da maneira correta.

4 Estudo do GUIX-Studio

O GUIX-Studio consiste em um ambiente de desenvolvimento de interfaces gráficas baseada no Windows e que foi projetado especificamente para a biblioteca GUIX. Através desta aplicação, é possível construir interfaces gráficas (de maneira intuitiva, arrastando e soltando os elementos), definir configurações (cores, imagens, tamanhos) e setar parâmetros que serão utilizados para a utilização via linha de código (como por exemplo, IDs e funções de tratamentos). Há dois manuais que podem ser considerados como referência: O manual da Microsoft⁷, que trata de linhas gerais do software e o manual da Sinergy⁸.

A Figura 14 apresenta a interface do GUIX-Studio, onde os principais componentes estão destacados. Observa-se no centro a área de trabalho, que consiste na tela propriamente dita (que será apresentada para o usuário). Na parte superior da tela estão os menus, no lado esquerdo está a árvore de elementos do projeto e abaixo as propriedades de um elemento (que aparecem ao clicar em um elemento na área de trabalho). Já no lado direito estão presentes algumas configurações gráficas, possibilitando colocar imagens, cores fontes e tamanhos distintos.

⁵ https://synergygallery.renesas.com/media/products/8/385/en-US/GUIX_User_Guide.pdf

⁶ <https://www.renesas.com/us/en/document/apn/quix-hello-world-sk-s7g2-and-pk-s5d9-application-project?language=en>

⁷ <https://docs.microsoft.com/pt-br/azure/rtos/quix/quix-studio-1>

⁸ <https://synergygallery.renesas.com/media/products/8/340/en-US/r11um0002eu0540-synergy-quix-studio.pdf>

Projeto – Space Invaders

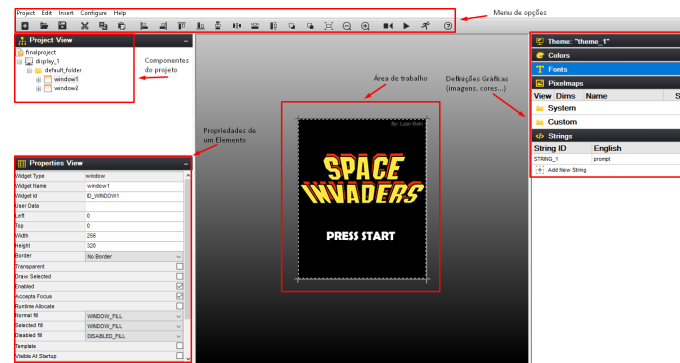


Figura 14: Exemplo de interface do GUIX-Studio.

Uma das maneiras de facilitar a construção da interface gráfica é criando o projeto do GUIX-Studio dentro do projeto que está sendo desenvolvido no e2studio. Dessa maneira, ao finalizar a construção da interface gráfica, basta gerar os códigos em C (no menu *Project*, a opção *Generate All Outputs Files*), e os arquivos serão gerados dentro do projeto do e2studio, estando disponíveis para a utilização na aplicação.