

Revisão do Projeto

Disciplina: Trabalho de Conclusão de Curso I – BCC

Caro, orientando,

segue abaixo a tabela de cálculo da média das notas obtidas no Pré-Projeto e Projeto, as DUAS revisões do seu projeto contendo a avaliação do professor “avaliador” e professor “TCC1”. Lembro que os ajustes indicados nestas revisões não precisam ser feitos no projeto, mas sim quando levarem o conteúdo do projeto para o artigo (se for o caso). Este material contendo todo o histórico das revisões é encaminhado para o professor de TCC2.

Tabela de cálculo da média das notas:

	Nome	Pessoa	PreProjeto										Projeto													
			TCC1				Avaliador				Banca		TCC1				Avaliador									
			A	P	N	Nota	A	P	N	Nota	Ori.	Esp.	Nota	A	P	N	Nota	A	P	N	Nota	Média				
	Gustavo Felipe Soares	204074	12	3	1	16	8,75	10	5	0	15	8,89	9,50	9,50	9,50	16	0	0	16	10,00	15	0	0	15	10,00	9,6

Atenciosamente,

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC		
() PRÉ-PROJETO	(X) PROJETO	ANO/SEMESTRE: 2023/1

IMPLEMENTAÇÃO DA M+++ NO ESP-32 UTILIZANDO O SISTEMA OPERACIONAL NUTTX

Gustavo Felipe Soares

Prof. Danton Cavalcanti Franco Junior – Orientador

Prof. Miguel Alexandre Wisintainer – Coorientador

1 INTRODUÇÃO

Sistemas embarcados são importantes na atualidade e podem ser encontrados em uma ampla gama de dispositivos utilizados no dia a dia. Esses sistemas são compostos por hardware e software e possuem periféricos de entrada e saída de dados, além de um microcontrolador que é o principal componente de um sistema embarcado. O microcontrolador é um computador com capacidade de processamento reduzida, responsável por executar e gerenciar uma aplicação. Tanto o ESP-32 quanto a M+++ são exemplos de microcontroladores utilizados em sistemas embarcados, sendo que a M+++ foi desenvolvida na Fundação Universidade Regional de Blumenau (FURB) em 2003 e vem sendo aperfeiçoada desde então.

A M+++ é um microcontrolador virtual utilizado nas aulas de Arquitetura I na FURB que interpreta algumas das instruções de código assembly. Desenvolvida em 2003, vem sendo aperfeiçoada pelos alunos desde então em trabalhos de conclusão de curso e algumas de suas versões serão apresentadas ao longo do trabalho.

Para auxiliar no gerenciamento e na eficiência dos sistemas embarcados, é possível instalar um Sistema Operacional (S.O.) compatível com o microcontrolador. Porém, é importante que esses sistemas operacionais sejam leves e otimizados para o processamento, a fim de executar a aplicação de forma eficiente. Um dos S.O.s que vem ganhando destaque no mercado é o NuttX, um sistema operacional recente e voltado para microcontroladores, que será abordado nesse estudo.

O NuttX é um sistema operacional de tempo real, ou seja, executa uma aplicação por vez. Pode ser usado em microcontroladores de 8 a 64 bits e é compatível com os padrões Portable Operating System Interface (POSIX) e American National Standards Institute (ANSI) e diversos hardwares. Lançado em 2007 por Gregory Nutt e passado para a Apache em 2019, possui uma grande comunidade de desenvolvedores e se tornou uma ótima opção para uso em microcontroladores.

No entanto, apesar da sua crescente popularidade, o NuttX possui pouca documentação unificada que detalhe o passo a passo para o desenvolvimento e disponibilização de aplicações a serem usadas no próprio NuttX. Diante desse contexto, esse trabalho propõe trazer a M+++ para o mundo físico em um ESP-32 utilizando o NuttX como sistema operacional, visando gerar documentação para ele. E assim, facilitar e auxiliar os futuros projetos e aplicações de sistemas embarcados que utilizarem o NuttX.

1.1 OBJETIVOS

O objetivo principal é tornar a M+++ um microcontrolador físico, utilizando o ESP-32 como hardware e o NuttX como seu sistema operacional, visando a geração de documentação de todo o processo.

Os objetivos específicos são:

- disponibilizar uma interface web para a entrada de código assembly no ESP-32;
- adicionar a aplicação da M+++ ao NuttX;
- gerar documentação da configuração do sistema operacional NuttX;
- testar o interpretador em sala de aula com alunos.

2 TRABALHOS CORRELATOS

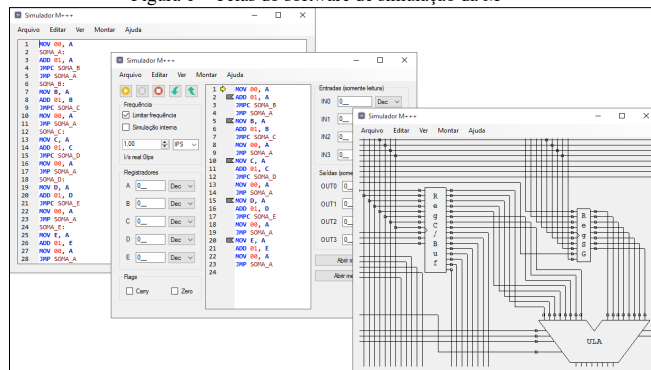
Nessa seção serão apresentados trabalhos semelhantes aos principais objetivos do estudo. Na subseção 2.1 é apresentado o trabalho de Klann (2017) que desenvolveu uma aplicação desktop para simular o microcontrolador M+++ . Na subseção 2.2 é apresentado o trabalho de Biegging (2018) que implementou a M++ em um Field-Programmable Gate Array (FPGA) e na subseção 2.3 é apresentado o artigo de Assis e Jerpelea (2023) que apresentou o NuttX, um sistema operacional POSIX voltado para sistemas embarcados.

2.1 SOFTWARE SIMULADOR DO MICROCONTROLADOR M+++

O software desenvolvido em 2017 por Klann, ex-aluno do curso de Ciências da Computação da FURB, é uma ferramenta didática de simulação virtual da M++, um microcontrolador criado na FURB em 2003 usado nas aulas de Arquitetura de Computadores I. Nele é possível codificar e depurar assembly, salvar e carregar projetos,

construir circuitos eletrônicos e visualizar sinais internos do microcontrolador a partir de uma interface gráfica. Além disso, o software também possui um montador assembly que valida o código criado ao rodá-lo. A Figura 1 mostra uma das versões das telas de codificação, de depuração e do circuito.

Figura 1 – Telas do software de simulação da M+++



Fonte: Klann (2017, p.44).

Por se tratar de uma evolução, traz uma série de melhorias em relação à versão anterior como a performance e a possibilidade de adicionar pontos de parada, embora também possua desvantagens, pois há uma quantidade menor de componentes disponíveis. Segundo o autor, essa solução apresentou resultados positivos nas pesquisas realizadas por ele, auxiliando na compreensão do funcionamento de sistemas embarcados, sendo bastante utilizado nas aulas.

2.2 IMPLEMENTAÇÃO DA M++ EM FPGA

O projeto desenvolvido em 2018 por Biegging, também ex-aluno do curso de Ciências da Computação da FURB, tem como objetivo trazer a M++ para um FPGA funcionando em um *clock* superior a 10KHz. Para isso, é possível carregar programas para a memória do microprocessador para serem lidos e interpretados. Com a finalidade de reproduzir a M++, seis módulos foram criados:

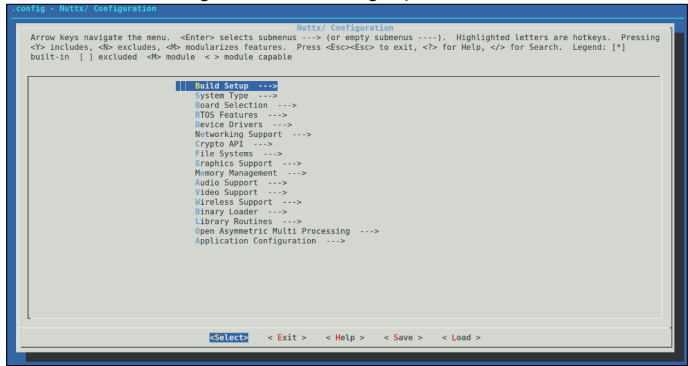
- módulo de controle: conecta todos os módulos;
- Unidade Lógica e Aritmética (ULA): implementa a ULA da M++;
- endereçador de memória de programa;
- banco de registradores: vetor para armazenar conteúdo dos registradores;
- endereçador da memória RAM externa: contador de incremento e decremento;
- memórias ROM.

Segundo o autor, a frequência máxima obtida foi de 90MHz e utilizou apenas 129 das mais de 41.000 portas lógicas disponíveis. Biegging ainda comenta que seu projeto obteve um avanço significativo, uma vez que com o simulador da M++ da época só era possível alcançar 50Hz. Seu trabalho atingiu todos os objetivos esperados, além de ter gerado documentação sobre a arquitetura.

2.3 (APACHE) NUTTX A LINUX-LIKE RTOS FOR MICROCONTROLLERS

Criado em 2007 por Gregory Nutt e doado para o Apache em 2019, o NuttX é um sistema operacional POSIX voltado para sistemas embarcados por ser bastante pequeno. “Pensando em ser uma alternativa de ser o que o Linux deveria ser para microcontroladores” (Assis e Jerpelea, 2023, p.2, tradução nossa), ele possui uma série de características fundamentais para esses dispositivos: é altamente configurável, roda em tempo real, é determinístico, possui suporte para hierarquia de prioridade entre outros. Além de ter diversas funcionalidades e drivers como sistema de arquivos, protocolos de internet, suporte gráfico, suporte à USB e áudio, também possui compatibilidade com vários outros microcontroladores como o ESP-32. Na Figura 2 é mostrado o menu de configuração do NuttX.

Figura 2 – Tela de configuração do NuttX



Fonte: Apache Software Foundation, 2020.

O NuttX foi desenvolvido desde o começo para ser compatível com POSIX, facilitando a transição para o NuttX para os desenvolvedores, uma vez que soluções Linux poderão ser levadas para microcontroladores rodando NuttX sem a necessidade de mudanças significativas (Assis e Jerpelea, 2023, tradução nossa).

3 PROPOSTA DO SOFTWARE

A seguir serão apresentadas a proposta e a justificativa, assim como as principais características dos trabalhos correlatos, os requisitos e a metodologia utilizada junto ao cronograma.

3.1 JUSTIFICATIVA

No Quadro 1 são comparados os três trabalhos correlatos acima em relação às principais características buscadas.

Quadro 1 - Comparativo dos trabalhos correlatos			
Trabalhos Correlatos	Klann (2017)	Bieging (2018)	Assis e Jerpelea (2023)
Interpreta código assembly	X		
Simula a M++	X	X	
Possui sistema operacional NuttX			X
Código gerado é executável em microcontroladores			X
É executado em microcontroladores			X

Fonte: elaborado pelo autor.

Conforme mostrado no Quadro 2, tanto a aplicação desenvolvida por Klann (2017) quanto a desenvolvida por Bieging (2018) simulam a M++, porém não rodam em microcontroladores e no NuttX. O software desktop implementado por Klann (2017) tem como propósito simular virtualmente, possuindo uma série de componentes para acoplar ao microcontrolador, podendo interpretar código assembly e visualizar o comportamento dos componentes de forma virtual.

Já o projeto do Bieging (2018) tem como objetivo rodar a simulação em FPGA que, diferentemente de um microcontrolador que se utiliza de software e já possui alguns componentes, possui diversas portas lógicas programáveis e precisa de periféricos externos.

O NuttX, apresentado por Assis e Jerpelea (2023) tem a proposta de rodar como um Linux em microcontroladores e permite a criação de aplicações novas nele.

Para a implementação do estudo, se faz necessário o entendimento sobre o NuttX, a M++ e compiladores. No primeiro trabalho correlato, os principais aspectos técnicos da M++ e como ela funciona são mostrados assim como a definição de um compilador e seus principais analisadores: léxico, sintático e semântico.

Esse trabalho tem a intensão de mostrar e documentar todos os passos para disponibilizar uma aplicação compatível com o NuttX visando a geração de um guia de desenvolvimento para o sistema operacional. Além disso, o produto gerado pelo estudo permitirá visualizar a execução de sistemas embarcados em meio físico em que, aplicado de forma didática, possibilitará uma maior imersão por parte dos alunos.

3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Os principais Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) são:

- permitir a inserção de código assembly (RF);
- salvar e carregar código assembly (RF);
- exibir erros retornados pelo compilador (RF);
- interpretar uma linha de código assembly por vez (RF);
- ser compatível com o sistema operacional NuttX (RNF);
- ser desenvolvido em C (RNF).

3.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- levantamento bibliográfico: buscar por trabalhos relacionados à M++ e ao NuttX, assim como artigos e outras fontes voltadas ao desenvolvimento em NuttX;
- levantamento de requisitos: detalhar os requisitos da aplicação com base nas bibliografias e definições dadas pelo orientador e coorientador;
- fundamentação teórica: estudo sobre o NuttX e a linguagem assembly;
- levantamento de ferramentas a serem usadas: definir quais ferramentas (software e hardware) são necessárias para a execução e desenvolvimento da aplicação;
- documentação: detalhar todos os passos necessários para a configuração do sistema operacional NuttX, desenvolvimento da aplicação e disponibilização dela;
- desenvolvimento: implementação da aplicação a partir dos requisitos levantados, sendo ela desenvolvida em C utilizando o Visual Studio Code;
- testes: validar as principais funcionalidades do software tais como o montador assembly e o servidor web;
- testes com alunos: testar a aplicação com alunos em sala para avaliar o funcionamento.

As etapas serão realizadas nos períodos relacionados no Quadro 2 abaixo.

Quadro 2 - Cronograma

etapas / quinzenas	2023																							
	fev.		mar.		abr.		maio		jun.		jul.		ago.		set.		out.		nov.		dez.			
	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2		
levantamento bibliográfico																								
levantamento de requisitos																								
fundamentação teórica																								
levantamento de ferramentas a serem usadas																								
documentação																								
desenvolvimento																								
testes																								
testes com alunos																								

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

Essa seção descreve brevemente os assuntos que fundamentarão o estudo a ser realizado: o NuttX, sistemas embarcados, a M++ e a linguagem assembly.

4.1 SISTEMAS EMBARCADOS

Sistemas embarcados são úteis para “executar uma tarefa específica em um sistema maior. [...]. Eles também precisam ser capazes de lidar com restrições de espaço e recursos como memória e processamento limitados.” (SOUZA, 2023). Por isso, sistemas operacionais como Windows e Linux não são boas opções para serem utilizados nesses dispositivos.

Para solucionar isso, existem sistemas operacionais modelados para essas circunstâncias como é o caso do NuttX, que começou a ter visibilidade nos últimos anos, podendo rodar com apenas 32K de memória. Esse sistema operacional já possui uma série de aplicações prontas para o uso e a cada dia novas funcionalidades são adicionadas a ele pela comunidade de desenvolvedores.

4.2 NUTTX

O NuttX é um sistema operacional de código aberto fortemente inspirado no Linux desenvolvido para sistemas embarcados iniciado por Gregory Nutt em 2007 e doado para a Apache em 2019 que, junto com uma

comunidade de desenvolvedores, mantém o S.O. atualizado. Por se tratar de um S.O. para sistemas embarcados, algumas de suas características são:

- a) compatível com diversos microcontroladores de 8 a 64 bits;
- b) compatível com os padrões ANSI e POSIX;
- c) sistema operacional de tempo real;
- d) possui suporte a sistema de arquivos (VFS, NFS, etc);
- e) é altamente escalável e configurável;
- f) possui suporte a protocolos de internet (IP, SOCKET, TCP, UDP, etc);
- g) possui suporte gráfico, a áudio e a USB.

4.3 M++

A M++ é um microcontrolador criado na FURB e desenvolvido no software Logisim. De acordo com Jung (2014), ela possui:

- a) memória RAM de 8 bits;
- b) memória de pilha de 8 bits;
- c) memória ROM de 16 bits;
- d) 4 registradores de entrada;
- e) 4 registradores de saída;
- i) operações da Unidade de Lógica e Aritmética (ULA);
- f) 5 operações de salto;
- g) linguagem semelhante ao assembly;
- h) as flags End Of Instruction (EOI), Carry e Zero;
- i) 4 registradores de 8 bits (B – E) + acumulador de 8 bits (A).

Para se adequar à M++, o componente PCA9555 deverá ser adicionado ao ESP-32, uma vez que a M++ possui 32 entradas/saídas e o ESP-32 utilizado tem apenas 16. Esse componente é um expansor de 16 portas e será integrado ao microcontrolador a partir do protocolo I2C, que possui dois canais de comunicação: *serial data*, para transmissão de dados e o *serial clock*, para manter a sincronia entre os dois dispositivos conectados.

4.4 ASSEMBLY

O assembly é uma linguagem de programação de baixo nível que funciona como uma abstração do código de máquina, tornando mais fácil seu entendimento e bastante útil quando se quer trabalhar byte a byte e/ou com menos memória, além de ser mais rápido. Porém, pode ser demorado escrever o código por ser uma linguagem mais complexa.

Algumas das instruções encontradas no assembly x86, conforme o *website* GitBook (2022), estão listadas abaixo:

- a) matemáticas: ADD, SUB, INC, DEC, MUL e DIV;
- b) lógicas: AND, OR, XOR, CMP, NEG e NOT;
- c) saída/atribuição: MOV, POP e PUSH.

Também possui os seguintes registradores:

- a) AX – Acumulador;
- b) BX – Endereço base;
- c) CX – Contador;
- d) DX – Dado;
- e) SP – Ponteiro para o topo da pilha;
- f) BP – Ponteiro para o início da pilha;
- g) SI – Endereço de origem dos dados;
- h) DI – Endereço de destino dos dados.

Já o assembly da M++ possui algumas diferenças. Abaixo estão listadas as instruções disponibilizadas por Jung (2014):

- a) matemáticas: ADD, SUB e INC;
- b) lógicas: AND, OR, XOR e NOT;
- c) saída/atribuição: MOV, POP e PUSH.

Os seguintes registradores são encontrados na M++: B, C, D e E.

REFERÊNCIAS

APACHE SOFTWARE FOUNDATION. **Configuring**. 2020. Disponível em: <https://nuttx.apache.org/docs/latest/quickstart/configuring.html>. Acesso em: 13 mar. 2023.

ASSIS, Alan Carvalho de; JERPELEA, Alin. **(Apache) NuttX**: a linux-like rtos for microcontrollers. 2023. 6 f. Apache, 2023. Disponível em: <https://pt.scribd.com/document/628557282/NuttX-RTOS>. Acesso em: 13 mar. 2023.

BIEGING, André Leonardo. **IMPLEMENTAÇÃO DA M++ EM FPGA**. 2018. 75 f. TCC (Graduação) - Curso de Ciência da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2018. Disponível em: <https://www.furb.br/dsc/tcc/index.php?cd=6&tcc=1937>. Acesso em: 20 mar. 2023.

GITBOOK. **Registradores de propósito geral**. 2022. Disponível em: <https://mentebinaria.gitbook.io/assembly/a-base/registradores-de-proposito-geral>. Acesso em: 10 abr. 2023.

JUNG, Jean. **M+++**. 2014. Disponível em: <https://github.com/jejung/maquina-plus-plus/blob/master/README.md>. Acesso em: 13 mar. 2023.

KLANN, Jean Carlos. **SOFTWARE SIMULADOR DO MICROCONTROLADOR M+++**. 2017. 61 f. TCC (Graduação) - Curso de Ciência da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2017. Disponível em: <https://www.furb.br/dsc/tcc/index.php?cd=6&tcc=1837>. Acesso em: 13 mar. 2023.

SOUZA, Fábio. **O que são sistemas embarcados?** 2023. Disponível em: <https://embarcados.com.br/o-que-sao-sistemas-embarcados/>. Acesso em: 21 mar. 2023.

FORMULÁRIO DE AVALIAÇÃO BCC – PROFESSOR AVALIADOR – PROJETO

Avaliador(a): **Francisco Adell Péricas**

Atenção: quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.

ASPECTOS AVALIADOS		Atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?	X		
	O problema está claramente formulado?	X		
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	X		
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?	X		
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?	X		
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?	X		
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?	X		
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?	X		
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?	X		
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?	X		
	7. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?	X		
ASPECTOS METODOLÓGICOS	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?	X		
	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?	X		
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	X		

O projeto de TCC será reprovado se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos **4 (quatro)** itens dos **ASPECTOS TÉCNICOS** tiverem resposta ATENDE PARCIALMENTE; ou
- pelo menos **4 (quatro)** itens dos **ASPECTOS METODOLÓGICOS** tiverem resposta ATENDE PARCIALMENTE.

PARECER: (X) APROVADO () REPROVADO

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC		
() PRÉ-PROJETO	(X) PROJETO	ANO/SEMESTRE: 2023/1

IMPLEMENTAÇÃO DA M+++ NO ESP-32 UTILIZANDO O SISTEMA OPERACIONAL NUTTX

Gustavo Felipe Soares

Prof. Danton Cavalcanti Franco Junior – Orientador

Prof. Miguel Alexandre Wisintainer – Coorientador

5 INTRODUÇÃO

Sistemas embarcados são importantes na atualidade e podem ser encontrados em uma ampla gama de dispositivos utilizados no dia a dia. Esses sistemas são compostos por hardware e software e possuem periféricos de entrada e saída de dados, além de um microcontrolador que é o principal componente de um sistema embarcado. O microcontrolador é um computador com capacidade de processamento reduzida, responsável por executar e gerenciar uma aplicação. Tanto o ESP-32 quanto a M+++ são exemplos de microcontroladores utilizados em sistemas embarcados, sendo que a M+++ foi desenvolvida na Fundação Universidade Regional de Blumenau (FURB) em 2003 e vem sendo aperfeiçoada desde então.

A M+++ é um microcontrolador virtual utilizado nas aulas de Arquitetura I na FURB que interpreta algumas das instruções de código assembly. Desenvolvida em 2003, vem sendo aperfeiçoada pelos alunos desde então em trabalhos de conclusão de curso e algumas de suas versões serão apresentadas ao longo do trabalho.

Para auxiliar no gerenciamento e na eficiência dos sistemas embarcados, é possível instalar um Sistema Operacional (S.O.) compatível com o microcontrolador. Porém, é importante que esses sistemas operacionais sejam leves e otimizados para o processamento, a fim de executar a aplicação de forma eficiente. Um dos S.O.s que vem ganhando destaque no mercado é o NuttX, um sistema operacional recente e voltado para microcontroladores, que será abordado nesse estudo.

O NuttX é um sistema operacional de tempo real, ou seja, executa uma aplicação por vez. Pode ser usado em microcontroladores de 8 a 64 bits e é compatível com os padrões Portable Operating System Interface (POSIX) e American National Standards Institute (ANSI) e diversos hardwares. Lançado em 2007 por Gregory Nutt e passado para a Apache em 2019, possui uma grande comunidade de desenvolvedores e se tornou uma ótima opção para uso em microcontroladores.

No entanto, apesar da sua crescente popularidade, o NuttX possui pouca documentação unificada que detalhe o passo a passo para o desenvolvimento e disponibilização de aplicações a serem usadas no próprio NuttX. Diante desse contexto, esse trabalho propõe trazer a M+++ para o mundo físico em um ESP-32 utilizando o NuttX como sistema operacional, visando gerar documentação para ele. E assim, facilitar e auxiliar os futuros projetos e aplicações de sistemas embarcados que utilizarem o NuttX.

5.1 OBJETIVOS

O objetivo principal é tornar a M+++ um microcontrolador físico, utilizando o ESP-32 como hardware e o NuttX como seu sistema operacional, visando a geração de documentação de todo o processo.

Os objetivos específicos são:

- d) disponibilizar uma interface web para a entrada de código assembly no ESP-32;
- e) adicionar a aplicação da M+++ ao NuttX;
- f) gerar documentação da configuração do sistema operacional NuttX;
- g) testar o interpretador em sala de aula com alunos.

6 TRABALHOS CORRELATOS

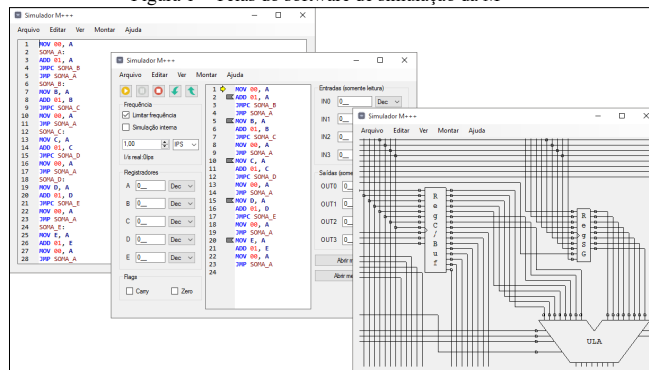
Nessa seção serão apresentados trabalhos semelhantes aos principais objetivos do estudo. Na subseção 2.1 é apresentado o trabalho de Klann (2017) que desenvolveu uma aplicação desktop para simular o microcontrolador M+++ . Na subseção 2.2 é apresentado o trabalho de Biegging (2018) que implementou a M++ em um Field-Programmable Gate Array (FPGA) e na subseção 2.3 é apresentado o artigo de Assis e Jerpelea (2023) que apresentou o NuttX, um sistema operacional POSIX voltado para sistemas embarcados.

6.1 SOFTWARE SIMULADOR DO MICROCONTROLADOR M+++

O software desenvolvido em 2017 por Klann, ex-aluno do curso de Ciências da Computação da FURB, é uma ferramenta didática de simulação virtual da M++, um microcontrolador criado na FURB em 2003 usado nas aulas de Arquitetura de Computadores I. Nele é possível codificar e depurar assembly, salvar e carregar projetos,

construir circuitos eletrônicos e visualizar sinais internos do microcontrolador a partir de uma interface gráfica. Além disso, o software também possui um montador assembly que valida o código criado ao rodá-lo. A Figura 1 mostra uma das versões das telas de codificação, de depuração e do circuito.

Figura 1 – Telas do software de simulação da M+++



Fonte: Klann (2017, p.44).

Por se tratar de uma evolução, traz uma série de melhorias em relação à versão anterior como a performance e a possibilidade de adicionar pontos de parada, embora também possua desvantagens, pois há uma quantidade menor de componentes disponíveis. Segundo o autor, essa solução apresentou resultados positivos nas pesquisas realizadas por ele, auxiliando na compreensão do funcionamento de sistemas embarcados, sendo bastante utilizado nas aulas.

6.2 IMPLEMENTAÇÃO DA M++ EM FPGA

O projeto desenvolvido em 2018 por Biegging, também ex-aluno do curso de Ciências da Computação da FURB, tem como objetivo trazer a M++ para um FPGA funcionando em um *clock* superior a 10KHz. Para isso, é possível carregar programas para a memória do microprocessador para serem lidos e interpretados. Com a finalidade de reproduzir a M++, seis módulos foram criados:

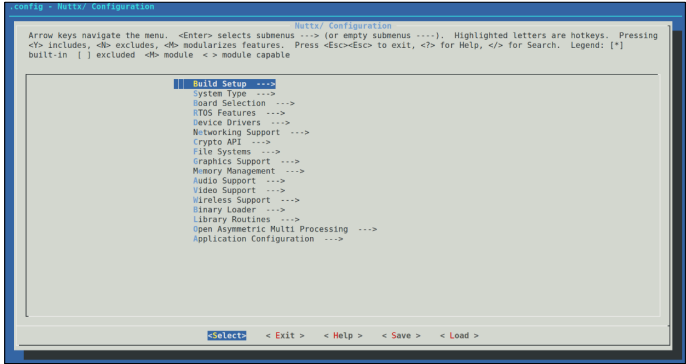
- g) módulo de controle: conecta todos os módulos;
- h) Unidade Lógica e Aritmética (ULA): implementa a ULA da M++;
- i) endereçador de memória de programa;
- j) banco de registradores: vetor para armazenar conteúdo dos registradores;
- k) endereçador da memória RAM externa: contador de incremento e decremento;
- l) memórias ROM.

Segundo o autor, a frequência máxima obtida foi de 90MHz e utilizou apenas 129 das mais de 41.000 portas lógicas disponíveis. Biegging ainda comenta que seu projeto obteve um avanço significativo, uma vez que com o simulador da M++ da época só era possível alcançar 50Hz. Seu trabalho atingiu todos os objetivos esperados, além de ter gerado documentação sobre a arquitetura.

6.3 (APACHE) NUTTX A LINUX-LIKE RTOS FOR MICROCONTROLLERS

Criado em 2007 por Gregory Nutt e doado para o Apache em 2019, o NuttX é um sistema operacional POSIX voltado para sistemas embarcados por ser bastante pequeno. “Pensando em ser uma alternativa de ser o que o Linux deveria ser para microcontroladores” (Assis e Jerpelea, 2023, p.2, tradução nossa), ele possui uma série de características fundamentais para esses dispositivos: é altamente configurável, roda em tempo real, é determinístico, possui suporte para hierarquia de prioridade entre outros. Além de ter diversas funcionalidades e drivers como sistema de arquivos, protocolos de internet, suporte gráfico, suporte à USB e áudio, também possui compatibilidade com vários outros microcontroladores como o ESP-32. Na Figura 2 é mostrado o menu de configuração do NuttX.

Figura 2 – Tela de configuração do NuttX



Fonte: Apache Software Foundation, 2020.

O NuttX foi desenvolvido desde o começo para ser compatível com POSIX, facilitando a transição para o NuttX para os desenvolvedores, uma vez que soluções Linux poderão ser levadas para microcontroladores rodando NuttX sem a necessidade de mudanças significativas (Assis e Jerpelea, 2023, tradução nossa).

7 PROPOSTA DO SOFTWARE

A seguir serão apresentadas a proposta e a justificativa, assim como as principais características dos trabalhos correlatos, os requisitos e a metodologia utilizada junto ao cronograma.

7.1 JUSTIFICATIVA

No Quadro 1 são comparados os três trabalhos correlatos acima em relação às principais características buscadas.

Quadro 1 - Comparativo dos trabalhos correlatos			
Trabalhos Correlatos Características	Klann (2017)	Bieging (2018)	Assis e Jerpelea (2023)
Interpreta código assembly	X		
Simula a M++	X	X	
Possui sistema operacional NuttX			X
Código gerado é executável em microcontroladores			X
É executado em microcontroladores			X

Fonte: elaborado pelo autor.

Conforme mostrado no Quadro 2, tanto a aplicação desenvolvida por Klann (2017) quanto a desenvolvida por Bieging (2018) simulam a M++, porém não rodam em microcontroladores e no NuttX. O software desktop implementado por Klann (2017) tem como propósito simular virtualmente, possuindo uma série de componentes para acoplar ao microcontrolador, podendo interpretar código assembly e visualizar o comportamento dos componentes de forma virtual.

Já o projeto do Bieging (2018) tem como objetivo rodar a simulação em FPGA que, diferentemente de um microcontrolador que se utiliza de software e já possui alguns componentes, possui diversas portas lógicas programáveis e precisa de periféricos externos.

O NuttX, apresentado por Assis e Jerpelea (2023) tem a proposta de rodar como um Linux em microcontroladores e permite a criação de aplicações novas nele.

Para a implementação do estudo, se faz necessário o entendimento sobre o NuttX, a M++ e compiladores. No primeiro trabalho correlato, os principais aspectos técnicos da M++ e como ela funciona são mostrados assim como a definição de um compilador e seus principais analisadores: léxico, sintático e semântico.

Esse trabalho tem a intensão de mostrar e documentar todos os passos para disponibilizar uma aplicação compatível com o NuttX visando a geração de um guia de desenvolvimento para o sistema operacional. Além disso, o produto gerado pelo estudo permitirá visualizar a execução de sistemas embarcados em meio físico em que, aplicado de forma didática, possibilitará uma maior imersão por parte dos alunos.

7.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Os principais Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) são:

- g) permitir a inserção de código assembly (RF);
- h) salvar e carregar código assembly (RF);
- i) exibir erros retornados pelo compilador (RF);
- j) interpretar uma linha de código assembly por vez (RF);
- k) ser compatível com o sistema operacional NuttX (RNF);
- l) ser desenvolvido em C (RNF).

7.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- h) levantamento bibliográfico: buscar por trabalhos relacionados à M+++ e ao NuttX, assim como artigos e outras fontes voltadas ao desenvolvimento em NuttX;
- i) levantamento de requisitos: detalhar os requisitos da aplicação com base nas bibliografias e definições dadas pelo orientador e coorientador;
- j) fundamentação teórica: estudo sobre o NuttX e a linguagem assembly;
- k) levantamento de ferramentas a serem usadas: definir quais ferramentas (software e hardware) são necessárias para a execução e desenvolvimento da aplicação;
- l) documentação: detalhar todos os passos necessários para a configuração do sistema operacional NuttX, desenvolvimento da aplicação e disponibilização dela;
- m) desenvolvimento: implementação da aplicação a partir dos requisitos levantados, sendo ela desenvolvida em C utilizando o Visual Studio Code;
- n) testes: validar as principais funcionalidades do software tais como o montador assembly e o servidor web;
- o) testes com alunos: testar a aplicação com alunos em sala para avaliar o funcionamento.

As etapas serão realizadas nos períodos relacionados no Quadro 2 abaixo.

Quadro 2 - Cronograma

etapas / quinzenas	2023																							
	fev.		mar.		abr.		maio		jun.		jul.		ago.		set.		out.		nov.		dez.			
	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2		
levantamento bibliográfico																								
levantamento de requisitos																								
fundamentação teórica																								
levantamento de ferramentas a serem usadas																								
documentação																								
desenvolvimento																								
testes																								
testes com alunos																								

Fonte: elaborado pelo autor.

8 REVISÃO BIBLIOGRÁFICA

Essa seção descreve brevemente os assuntos que fundamentarão o estudo a ser realizado: o NuttX, sistemas embarcados, a M+++ e a linguagem assembly.

8.1 SISTEMAS EMBARCADOS

Sistemas embarcados são úteis para “executar uma tarefa específica em um sistema maior. [...]. Eles também precisam ser capazes de lidar com restrições de espaço e recursos como memória e processamento limitados.” (SOUZA, 2023). Por isso, sistemas operacionais como Windows e Linux não são boas opções para serem utilizados nesses dispositivos.

Para solucionar isso, existem sistemas operacionais modelados para essas circunstâncias como é o caso do NuttX, que começou a ter visibilidade nos últimos anos, podendo rodar com apenas 32K de memória. Esse sistema operacional já possui uma série de aplicações prontas para o uso e a cada dia novas funcionalidades são adicionadas a ele pela comunidade de desenvolvedores.

8.2 NUTTX

O NuttX é um sistema operacional de código aberto fortemente inspirado no Linux desenvolvido para sistemas embarcados iniciado por Gregory Nutt em 2007 e doado para a Apache em 2019 que, junto com uma

comunidade de desenvolvedores, mantém o S.O. atualizado. Por se tratar de um S.O. para sistemas embarcados, algumas de suas características são:

- h) compatível com diversos microcontroladores de 8 a 64 bits;
- i) compatível com os padrões ANSI e POSIX;
- j) sistema operacional de tempo real;
- k) possui suporte a sistema de arquivos (VFS, NFS, etc);
- l) é altamente escalável e configurável;
- m) possui suporte a protocolos de internet (IP, SOCKET, TCP, UDP, etc);
- n) possui suporte gráfico, a áudio e a USB.

8.3 M++

A M++ é um microcontrolador criado na FURB e desenvolvido no software Logisim. De acordo com Jung (2014), ela possui:

- j) memória RAM de 8 bits;
- k) memória de pilha de 8 bits;
- l) memória ROM de 16 bits;
- m) 4 registradores de entrada;
- n) 4 registradores de saída;
- p) operações da Unidade de Lógica e Aritmética (ULA);
- o) 5 operações de salto;
- p) linguagem semelhante ao assembly;
- q) as flags End Of Instruction (EOI), Carry e Zero;
- r) 4 registradores de 8 bits (B – E) + acumulador de 8 bits (A).

Para se adequar à M++, o componente PCA9555 deverá ser adicionado ao ESP-32, uma vez que a M++ possui 32 entradas/saídas e o ESP-32 utilizado tem apenas 16. Esse componente é um expansor de 16 portas e será integrado ao microcontrolador a partir do protocolo I2C, que possui dois canais de comunicação: *serial data*, para transmissão de dados e o *serial clock*, para manter a sincronia entre os dois dispositivos conectados.

8.4 ASSEMBLY

O assembly é uma linguagem de programação de baixo nível que funciona como uma abstração do código de máquina, tornando mais fácil seu entendimento e bastante útil quando se quer trabalhar byte a byte e/ou com menos memória, além de ser mais rápido. Porém, pode ser demorado escrever o código por ser uma linguagem mais complexa.

Algumas das instruções encontradas no assembly x86, conforme o *website* GitBook (2022), estão listadas abaixo:

- d) matemáticas: ADD, SUB, INC, DEC, MUL e DIV;
- e) lógicas: AND, OR, XOR, CMP, NEG e NOT;
- f) saída/atribuição: MOV, POP e PUSH.

Também possui os seguintes registradores:

- i) AX – Acumulador;
- j) BX – Endereço base;
- k) CX – Contador;
- l) DX – Dado;
- m) SP – Ponteiro para o topo da pilha;
- n) BP – Ponteiro para o início da pilha;
- o) SI – Endereço de origem dos dados;
- p) DI – Endereço de destino dos dados.

Já o assembly da M++ possui algumas diferenças. Abaixo estão listadas as instruções disponibilizadas por Jung (2014):

- q) matemáticas: ADD, SUB e INC;
- r) lógicas: AND, OR, XOR e NOT;
- s) saída/atribuição: MOV, POP e PUSH.

Os seguintes registradores são encontrados na M++: B, C, D e E.

REFERÊNCIAS

APACHE SOFTWARE FOUNDATION. **Configuring**. 2020. Disponível em: <https://nuttx.apache.org/docs/latest/quickstart/configuring.html>. Acesso em: 13 mar. 2023.

Comentado [DSdR1]: Arrumar espaçamento entre linhas.

ASSIS, Alan Carvalho de; JERPELEA, Alin. **(Apache) NuttX**: a linux-like rtos for microcontrollers. 2023. 6 f. Apache, 2023. Disponível em: <https://pt.scribd.com/document/628557282/NuttX-RTOS>. Acesso em: 13 mar. 2023.

BIEGING, André Leonardo. **IMPLEMENTAÇÃO DA M++ EM FPGA**. 2018. 75 f. TCC (Graduação) - Curso de Ciência da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2018. Disponível em: <https://www.furb.br/dsc/tcc/index.php?cd=6&tcc=1937>. Acesso em: 20 mar. 2023.

GITBOOK. **Registradores de propósito geral**. 2022. Disponível em: <https://mentebinaria.gitbook.io/assembly/a-base/registradores-de-proposito-geral>. Acesso em: 10 abr. 2023.

JUNG, Jean. **M+++**. 2014. Disponível em: <https://github.com/jejung/maquina-plus-plus/blob/master/README.md>. Acesso em: 13 mar. 2023.

KLANN, Jean Carlos. **SOFTWARE SIMULADOR DO MICROCONTROLADOR M+++**. 2017. 61 f. TCC (Graduação) - Curso de Ciência da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2017. Disponível em: <https://www.furb.br/dsc/tcc/index.php?cd=6&tcc=1837>. Acesso em: 13 mar. 2023.

SOUZA, Fábio. **O que são sistemas embarcados?** 2023. Disponível em: <https://embarcados.com.br/o-que-sao-sistemas-embarcados/>. Acesso em: 21 mar. 2023.

FORMULÁRIO DE AVALIAÇÃO BCC – PROFESSOR TCC I – PROJETO

Avaliador(a): Dalton Solano dos Reis

ASPECTOS AVALIADOS		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	9. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?	X		
	O problema está claramente formulado?	X		
	10. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	X		
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	11. JUSTIFICATIVA São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?	X		
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?	X		
	12. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?	X		
	Os métodos, recursos e o cronograma estão devidamente apresentados?	X		
ASPECTOS METODOLÓGICOS	13. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?	X		
	14. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?	X		
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	X		
	15. ORGANIZAÇÃO E APRESENTAÇÃO GRÁFICA DO TEXTO A organização e apresentação dos capítulos, seções, subseções e parágrafos estão de acordo com o modelo estabelecido?	X		
	16. ILUSTRAÇÕES (figuras, quadros, tabelas) As ilustrações são legíveis e obedecem às normas da ABNT?	X		
	17. REFERÊNCIAS E CITAÇÕES As referências obedecem às normas da ABNT?	X		
	As citações obedecem às normas da ABNT?	X		
	Todos os documentos citados foram referenciados e vice-versa, isto é, as citações e referências são consistentes?	X		

O projeto de TCC será reprovado se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos 4 (quatro) itens dos **ASPECTOS TÉCNICOS** tiverem resposta ATENDE PARCIALMENTE; ou
- pelo menos 4 (quatro) itens dos **ASPECTOS METODOLÓGICOS** tiverem resposta ATENDE PARCIALMENTE.

PARECER: (X) APROVADO () REPROVADO