UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
BACHARELADO EM ENGENHARIA ELÉTRICA – BEE

HENRIQUE TREVISAN

ESTUDO DO TPM E RTOS PARA SEGURANÇA CIBERNÉTICA EM HARDWARE DE CONVERSORES ESTÁTICOS

JOINVILLE
2025

HENRIQUE TREVISAN

ESTUDO DO TPM E RTOS PARA SEGURANÇA CIBERNÉTICA EM HARDWARE DE CONVERSORES ESTÁTICOS

Trabalho de Conclusão de Curso apresentado ao departamento de Engenharia Elétrica do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, como requisito parcial para a obtenção do grau de Bacharel em Engenharia Elétrica.
Orientador: Prof. Dr. Sérgio Vidal Garcia Oliveira

JOINVILLE
2025

HENRIQUE TREVISAN

ESTUDO DO TPM E RTOS PARA SEGURANÇA CIBERNÉTICA EM HARDWARE DE CONVERSORES ESTÁTICOS

Trabalho de Conclusão de Curso apresentado ao departamento de Engenharia Elétrica do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, como requisito parcial para a obtenção do grau de Bacharel em Engenharia Elétrica.

BANCA EXAMINADORA:

Sérgio Vidal Garcia de Oliveira, Dr.
Universidade do Estado de Santa Catarina

Membros:

Joselito Anastacio Heerdt, Dr.
Universidade do Estado de Santa Catarina

Tiago Martins, Me.
WEG Drives & Controls Automação Ltda

Joinville, 10 de julho de 2025

RESUMO

O trabalho apresenta um estudo sobre cibersegurança em conversores estáticos, abordando incidentes antigos e recentes a infraestruturas críticas, fragilidades de protocolos de comunicação e a adoção do Módulo de Plataforma Confiável (TPM) em microcontroladores que executam o FreeRTOS. Inicialmente discute-se os tipos de ataques e o aumento da percepção de risco no setor industrial. Em seguida, detalham-se as vulnerabilidades de conversores conectados, como falhas de atualização de *firmware*, exposição física e gestão de chaves. A parte experimental utilizou um microcontrolador STM32 configurado no STM32CubeIDE para integrar as bibliotecas wolfSSL e wolfTPM e implementar o TPM em conjunto com um microcontrolador. Realizaram-se testes de inicialização segura e de comunicação protegida; contudo, a compilação do sistema não foi concluída, sempre indicando cabeçalhos de bibliotecas e variáveis ausentes, evidenciando a complexidade de provisionamento exigida pelo TPM 2.0. Conclui-se que o TPM pode aumentar a proteção dos conversores ao fornecer raiz de confiança, atestado de *firmware* e gestão de chaves devido aos requisitos fortes definidos em sua concepção. Entretanto, sua adoção direta requer adaptações consideráveis em plataformas embarcadas, especialmente no que diz respeito à integração com o *firmware* existente. Enquanto os protocolos e bibliotecas para as chamadas de funções do TPM não forem incorporadas nos *drivers* nativos dos microcontroladores, sua implementação será desafiadora, exigindo bibliotecas elaboradas por terceiros, como é o caso da wolfTPM e wolfSSL.

Palavras-chave:Conversores estáticos; FreeRTOS; Microcontroladores; Segurança cibernética; TPM;

ABSTRACT

This paper presents a study on cybersecurity in static converters, addressing historical and recent incidents targeting critical infrastructures, vulnerabilities in communication protocols, and the adoption of the Trusted Platform Module (TPM) in microcontrollers running FreeRTOS. Initially, the types of attacks and the growing risk awareness in the industrial sector are discussed. Subsequently, the vulnerabilities of connected converters are detailed, including *firmware* update failures, physical exposure, and key management. The experimental section employed an STM32 microcontroller configured in STM32CubeIDE to integrate the wolfSSL and wolfTPM libraries and to implement the TPM in conjunction with the microcontroller. Secure boot and protected communication tests were conducted; nevertheless, system compilation was not completed, consistently reporting missing library headers and variables, thus evidencing the provisioning complexity required by TPM 2.0. It is concluded that the TPM can enhance converter protection by providing a root of trust, *firmware* attestation, and key management due to the stringent requirements defined in its design. However, its direct adoption necessitates considerable adaptations in embedded platforms, particularly concerning integration with existing *firmware*. Until the protocols and libraries for TPM function calls are incorporated into native microcontroller *drivers*, its implementation will remain challenging, requiring third-party libraries such as wolfTPM and wolfSSL.

Keywords: Cybersecurity; FreeRTOS; Microcontrollers; Static converters; TPM;

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

API	Interface de Programação de Aplicações
BIOS	Sistema Básico de Entrada/Saída
CA	Autoridades Certificadas
CAN	Controller Area Network
C&C	Command and Control
CISA	Cybersecurity & Infrastructure Security Agency
CLP	Controladores Lógicos Programáveis
CPU	Central Processing Unit
DDoS	Distributed Denial-of-Service
DMA	Direct Memory Access
DoS	Denial-of-Service
DSP	Digital Signal Processor
DTCM	Data Tightly Coupled Memory
EA	Enhanced Authorization
ECC	Elliptic Curve Cryptography
FIPS	Federal Information Processing Standards
FreeRTOS	Sistema Operacional em Tempo Real de código aberto
GPIO	General Purpose Input/Output
HAL	Hardware Abstraction Layer
HMAC	Hash-based Message Authentication Code
HSM	Hardware Security Module
ICS	Industrial Control System
IDE	Integrated Development Environment
IHM	Interface Homem-Máquina
IIoT	Industrial Internet of Things
JTAG	Joint Test Action Group
KB	Kilobytes
MITM	Man-in-the-Middle
NV	Non-Volatile
OPC	OLE for Process Control

PCR	Platform Configuration Register
PID	Proportional-Integral-Derivative
PLC	Programmable Logic Controller
PWM	Pulse Width Modulation
RBAC	Role Based Access Control
RISC	Reduced Instruction Set Computer
RoT	Root of Trust
RSA	Rivest-Shamir-Adleman
RTOS	Real-Time Operating System
SaaS	Software as a Service
SCADA	Supervisory Control and Data Acquisition
SD	Secure Digital
SEGGER	SEGGER SystemView - ferramenta para depuração e análise em sistemas embarcados
SMP	Symmetric Multiprocessing
SPI	Serial Peripheral Interface
SQL	Structured Query Language
SSL	Secure Sockets Layer
ST	STMicroelectronics
STM32CubeIDE	Ambiente de Desenvolvimento Integrado para STM32
STM32H7	Família de microcontroladores ST
TCG	Trusted Computing Group
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TI	Tecnologia da Informação
TLS	Transport Layer Security
TO	Tecnologia Operacional
TPM	Trusted Platform Module
TrustZone	Extensão de segurança da Arm
TSS2	Trusted Software Stack 2
UART	Universal Asynchronous Receiver-Transmitter

USB	Universal Serial Bus
WEF	World Economic Forum
wolfCrypt	Biblioteca de algoritmos criptográficos
wolfSSL	Biblioteca leve de criptografia e TLS
wolfTPM	Biblioteca que fornece interface de software para o TPM

SUMÁRIO

1 INTRODUÇÃO

A digitalização crescente de processos industriais amplia a exposição de sistemas de potência às ameaças cibernéticas. Conversores estáticos, fundamentais em equipamentos eletrônicos e em infraestruturas críticas, passam a operar conectados a redes de comunicação e podem sofrer intervenções maliciosas.

1.1 JUSTIFICATIVA

O tema apresenta relevância acadêmica e industrial para o desenvolvimento de soluções em eletrônica de potência aliadas à segurança cibernética, permitindo avanços em dispositivos que controlam sistemas críticos. No âmbito industrial, a proteção de chaves e a verificação de integridade do ***firmware***[1] são fundamentais para a confiabilidade dos processos de conversão de energia. A adoção de práticas de segurança cibernética em conversores estáticos é essencial para garantir a disponibilidade e a integridade dos sistemas, prevenindo interrupções, danos físicos e financeiros para as empresas e usuários.

Dessa maneira, surge o seguinte problema de pesquisa: como aumentar a segurança de conversores estáticos empregando o TPM?

1.2 ORGANIZAÇÃO DO TRABALHO

O Capítulo 2 apresenta os conceitos essenciais de segurança cibernética e, na Seção 2.4, são descritos episódios que evidenciam a necessidade de maior proteção nos sistemas de potência.

O Capítulo 3 discute as vulnerabilidades de conversores estáticos e sua importância para a confiabilidade de equipamentos industriais.

O Capítulo 4 descreve ferramentas de comando e controle, abordando recursos de ***hardware*** e ***software*** empregados na proteção desses dispositivos.

O Capítulo 5 detalha a integração do TPM ao microcontrolador escolhido, estruturando as práticas de segurança propostas.

A Seção 5.2 explica a programação do microcontrolador e apresenta os resultados obtidos nos testes realizados.

No Capítulo 6 são discutidas as conclusões do estudo e apresentadas perspectivas para trabalhos futuros.

Por fim, apresenta-se um glossário que reúne as definições dos principais termos técnicos utilizados ao longo do texto.

[1] *Software* embarcado diretamente no *hardware*, responsável por controlar funções específicas do dispositivo.

1.3 OBJETIVO GERAL

O objetivo geral deste trabalho é avaliar a complexidade da aplicação do TPM em conversores estáticos, executando práticas de proteção de chaves aos sistemas de controle em eletrônica de potência.

1.4 OBJETIVOS ESPECÍFICOS

Os objetivos específicos incluem:

- revisar a literatura sobre segurança cibernética e TPM;
- analisar incidentes de segurança cibernética relevantes para conversores estáticos;
- identificar vulnerabilidades comuns em conversores estáticos e propor medidas de mitigação;
- estudar o funcionamento do TPM 2.0 e suas aplicações em sistemas embarcados;
- implementar o FreeRTOS para controlar tarefas no microcontrolador STM32H7;

2 FUNDAMENTOS DA SEGURANÇA CIBERNÉTICA

Com a ampla interconexão de dispositivos e sistemas, a segurança cibernética emerge como uma preocupação alarmante em escala global. O Relatório de Riscos Globais do Fórum Econômico Mundial identifica instituições financeiras e infraestruturas críticas como os ambientes com a maior percepção de risco a curto prazo, destacando o potencial perigo associado à crescente disseminação de crimes cibernéticos (MUNDIAL, 2023).

Com isso, o aprimoramento dos sistemas de segurança cibernética deve ser um processo contínuo, sujeito a avaliações e aperfeiçoamentos regulares (Martins, 2021), uma vez que nenhum sistema se encontra completamente imune a todas as ameaças cibernéticas.

Os principais elementos a serem considerados no contexto da segurança cibernética incluem o agente, a ameaça, o alvo, as vulnerabilidades e as contramedidas (Branquinho; Branquinho, 2021).

2.1 PRIORIDADES NA TECNOLOGIA OPERACIONAL (TO) E NA TECNOLOGIA DA INFORMAÇÃO (TI)

Na Tecnologia Operacional (TO), que envolve o controle de processos físicos por meio de *hardware* e *software*, três elementos são essenciais para assegurar a segurança cibernética do processo ou sistema: a disponibilidade (considerada de importância primordial), juntamente com a integridade e a confidencialidade (Branquinho; Branquinho, 2021).

2.1.1 Disponibilidade

Assegura a operação contínua dos sistemas, permitindo que um usuário autorizado acesse as informações necessárias quando requerido. Desafios significativos nesse contexto envolvem ataques de negação de serviço[1] (DoS, *Denial-of-Service*) e a redução ou perda de capacidade de processamento dos dados, que pode comprometer severamente essa propriedade (Martins, 2021);

2.1.2 Integridade

Garante a proteção dos dados contra destruição ou alteração, seja acidental ou intencional, garantindo que os sistemas operem de acordo com suas finalidades. Ferramentas de controle como *hash*[2], *checksum*[3], antivírus, listas de permissões e assinatura digital são amplamente utilizadas para assegurar a integridade de um sistema (Martins, 2021);

[1] Ataque que visa tornar um serviço indisponível por sobrecarga, impedindo o acesso legítimo a sistemas ou redes.

[2] Função matemática que transforma dados de entrada em uma sequência fixa de bits, usada para verificar integridade.

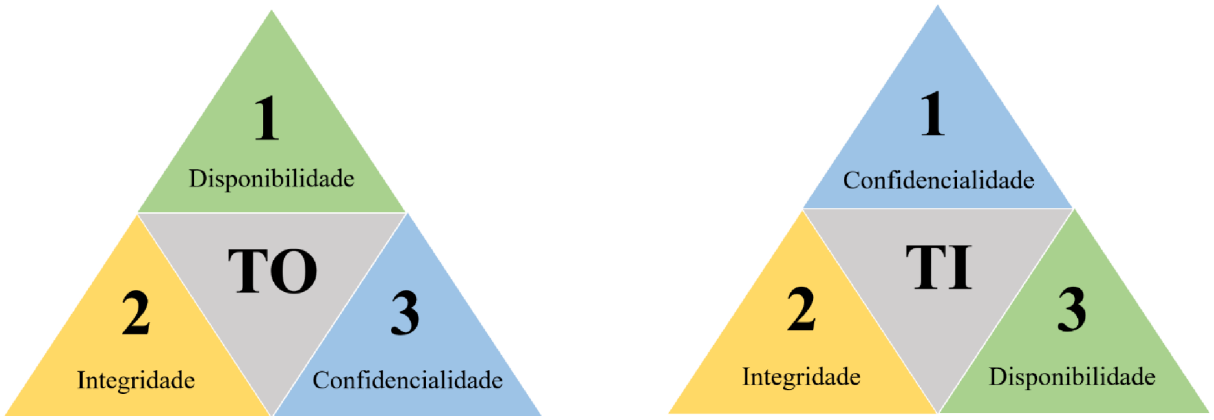
[3] Valor calculado a partir de um conjunto de dados para detectar erros de integridade na transmissão ou armazenamento.

2.1.3 Confidencialidade

Impede o acesso não autorizado de indivíduos ou programas a sistemas e informações. Normalmente está associada à transmissão e ao armazenamento de dados e é comprometida quando algum agente contorna os mecanismos de controle de acesso ou explora outras vulnerabilidades do sistema, comprometendo a confidencialidade (Martins, 2021).

Esse enfoque difere da abordagem adotada na Tecnologia da Informação (TI), na qual a confidencialidade é a principal prioridade, seguida pela integridade e, por fim, a disponibilidade. Essa diferença reflete as distintas prioridades e necessidades de operação entre os dois domínios, conforme ilustrado na Figura 1.

Figura 1 – Diferentes prioridades de segurança cibernética na Tecnologia Operacional (TO) e na Tecnologia da Informação (TI).



(a) Prioridades da Tecnologia Operacional.

(b) Prioridades da Tecnologia da Informação.

Fonte: (Martins, 2021).

2.2 ATORES E AMEAÇAS NO CONTEXTO INDUSTRIAL

Com utilização de dispositivos de sensoriamento e atuadores conectados à rede, torna-se viável realizar remotamente modificações substanciais no processo de produção (Balda et al., 2017). Tal abordagem proporciona novas perspectivas para a atuação de determinados agentes na alteração indevida do processo, sendo esses agentes referidos como "atores"(Branquinho; Branquinho, 2021). No âmbito cibernético, diversos perfis de atores podem desempenhar papéis distintos, dentre eles:

2.2.1 Criminosos

Dedicam-se ao sequestro de infraestruturas, praticando espionagem industrial e extorquindo suas vítimas;

2.2.2 Vândalos virtuais

Motivam-se a demonstrar suas habilidades perante um público, geralmente em fóruns e não possuem foco em ganhos financeiros;

2.2.3 Pesquisadores/Estudantes

Conduzem ataques sem a intenção de causar danos. Normalmente estão em processo de aprendizado de novas ferramentas ou tecnologias, informando frequentemente ao alvo acerca das vulnerabilidades identificadas no sistema em questão (Branquinho; Branquinho, 2021);

2.2.4 Colaboradores insatisfeitos

Representam uma ameaça grave, motivados por questões relacionadas à função, remuneração, aspectos sociais ou políticos (Branquinho; Branquinho, 2021). Os colaboradores descontentes representam um risco elevado, visto que, em muitos casos, detêm privilégios no acesso a ferramentas e ao sistema todo, o que lhes possibilita alterar integralmente um processo ou conceder acesso a agentes externos, como *black-hat hackers*[4], para a execução de um ataque.

2.2.5 Colaboradores descuidados

Embora não tenham a intenção de causar danos, podem comprometer a segurança do sistema ao não seguir as diretrizes de segurança estabelecidas. Por exemplo, ao utilizar senhas fracas ou reutilizar senhas em diferentes sistemas, esses colaboradores inadvertidamente abrem brechas para invasões cibernéticas. Outros exemplos incluem o compartilhamento de senhas com colegas, o vazamento de informações sensíveis em redes sociais ou a instalação de *softwares* não autorizados nos sistemas corporativos.

2.3 TIPOS DE ATAQUES

Diversos métodos podem ser empregados para invadir um sistema, sendo a utilização de *malwares* a mais comum (Martins, 2021). Os *malwares* consistem em *softwares* maliciosos desenvolvidos com o propósito de prejudicar um sistema, seja sequestrando dados, desativando servidores, corrompendo arquivos, ou realizando outras ações danosas. Outros métodos envolvem o uso de ferramentas específicas, ou a personificação de um agente legítimo. Entre os principais métodos de ataque, destacam-se:

[4] **Black-hat** hackers invadem redes de forma ilegal visando ganhos pessoais ou maliciosos (roubo de credenciais, venda de dados, etc.); **white-hat** hackers são profissionais autorizados que identificam e corrigem vulnerabilidades para fortalecer a segurança dos sistemas; **gray-hat** hackers ficam no meio-termo, invadindo sistemas sem permissão mas geralmente sem intenções puramente maliciosas e às vezes oferecendo conserto das falhas encontradas (Moore Michelle, 2025).

2.3.1 Cavalos de Tróia (*Trojans*)

São *softwares* que muitas vezes aparentam cumprir sua função declarada, mas também abrem uma porta de acesso remoto ao dispositivo em que são instalados, conhecida como "*backdoor*"[5].

2.3.2 Kits de exploração (*Exploit Kits*)

São ferramentas prontas encontradas na *internet* para realizar ataques, que podem ser adquiridas como serviço (SaaS - *Software as a Service*). Esses kits proporcionam interfaces simplificadas, tornando acessível o seu uso até mesmo por indivíduos com pouco conhecimento técnico e são projetados para serem executados em uma ampla gama de dispositivos.

2.3.3 Negação de serviço (DoS - *Denial-of-Service*)

Consiste em sobrecarregar uma rede com a circulação massiva de dados, o que dificulta a entrega eficiente de pacotes de dados legítimos ao seu destino. Frequentemente, esse tipo de ataque provoca lentidão nos sistemas, o que pode inviabilizar a prestação de alguns serviços, como a transmissão de vídeo sob demanda, podendo, em casos mais graves, levar à interrupção completa do serviço. Uma segunda forma é a negação de serviço distribuída (DDoS - *Distributed Denial-of-Service*), que utiliza múltiplos dispositivos comprometidos para amplificar o ataque;

2.3.4 *Bots* e Zumbis

Dispositivos podem ser infectados e transformados em "robôs"(*Bots*) ou "zumbis", passando a ser controlados remotamente. Esse tipo de ataque passa geralmente despercebido pela vítima, uma vez que seu objetivo é alocar parte do poder computacional disponível para ser utilizado em outros tipos de ataques, como o DDoS, ou para atividades como a mineração de criptomoedas (Branquinho; Branquinho, 2021; Martins; Oliveira, 2019);

2.3.5 Vírus e Vermes (*Worms*)

Constituem um tipo específico de *malware* caracterizado por sua notável capacidade de propagação. Além de causarem danos, eles são frequentemente disseminados por meio de *e-mails*, vulnerabilidades no sistema operacional ou vulnerabilidades em aplicativos já instalados.

2.3.6 *Phishing*

O *hacker* utiliza artifícios psicológicos para atrair a vítima, como a oferta de um aplicativo que aparenta resolver um problema específico ou a apresentação de uma ferramenta que desperte a ganância da vítima, prometendo ganhos substanciais. Quando a vítima executa o *malware*

[5] Mecanismo oculto que permite acesso não autorizado a um sistema, contornando os métodos normais de autenticação.

com privilégios de administrador, o criminoso obtém acesso aos seus dados e assume o controle parcial ou total do dispositivo (Rastenis et al., 2020). Entretanto, nem sempre o *phishing* tem intenção de comprometer o dispositivo da vítima, mas sim de obter informações confidenciais, como credenciais de acesso e senhas.

2.3.7 Ransomware

O *ransomware* é uma forma de *malware* em constante evolução, projetada para criptografar[6] arquivos em um dispositivo e tornar inutilizáveis tanto os dados quanto os sistemas que deles dependem (CISA, 2025). Criminosos cibernéticos exigem resgate em troca da chave de descriptografia[7] e frequentemente ameaçam vender ou vazar dados confidenciais ou credenciais roubadas caso o pagamento não ocorra (CISA, 2025). Segundo o FBI (Departamento Federal de Investigação - *Federal Bureau of Investigation*), esse tipo de ataque impede o acesso a arquivos, sistemas ou redes, gerando interrupções e risco de perda de informações críticas (Investigation, 2025). Além disso, o *ransomware* se propaga comumente por *e-mails* de *phishing*, *downloads* maliciosos e falhas em *softwares* desatualizados, representando uma ameaça grave para indivíduos e organizações.

2.3.8 MITM - *Man-in-the-Middle*

O invasor não adentra diretamente o dispositivo da vítima, mas intercepta a via de comunicação entre dois dispositivos (Martins; Oliveira, 2019). Esse método é particularmente recorrente em redes sem fio, como as redes Wi-Fi, em que o ator intercepta pacotes de dados transmitidos pelo dispositivo, realiza cópias e/ou modificações e, em seguida, retransmite esses pacotes para o roteador, que os reconhece como legítimos.

A crescente interconexão de dispositivos expõe instituições críticas a riscos elevados, exigindo avaliação e monitoramento contínuos dos cinco elementos da segurança cibernética (agente, ameaça, alvo, vulnerabilidades e contramedidas). Em ambientes de Tecnologia Operacional, prioriza-se disponibilidade > integridade > confidencialidade (na TI, o inverso), refletindo necessidades distintas de operação e proteção. Atores diversos - de criminosos a *insiders* insatisfeitos - exploram *malwares*, DoS/DDoS, *botnets*, *phishing* e MITM, o que demanda a adoção de defesas multicamadas.

Defesa multicamadas *(*defense in depth) é uma abordagem de segurança que empilha produtos e práticas de controle em camadas físicas, técnicas e administrativas para proteger redes e recursos críticos contra diversos vetores de ataque de forma redundante (Cloudflare, 2025). Cada camada funciona como um ponto de verificação independente, de modo que, se uma falhar ou for comprometida, as camadas subsequentes ainda oferecem barreiras que retardam e

[6] Processo de codificar informações para torná-las inacessíveis a terceiros não autorizados.
[7] Processo de reverter a criptografia, tornando os dados novamente legíveis para usuários autorizados.

identificam o invasor (Defense. . . , 2025). Além disso, essa estratégia fortalece a resiliência do ambiente, permitindo detectar e neutralizar ameaças em diferentes níveis antes que alcancem o alvo principal (Cloudflare, 2025).

2.4 INCIDENTES DE SEGURANÇA CIBERNÉTICA

Nas últimas décadas, o avanço tecnológico, a digitalização de processos industriais e a interconectividade das infraestruturas tornaram os sistemas ciberfísicos[8] alvos preferenciais de ataques cibernéticos. A crescente sofisticação das ameaças, somada ao uso malicioso de ferramentas automatizadas, expôs vulnerabilidades críticas em setores antes considerados isolados ou protegidos. Relatórios recentes, como o *Global Risks Report* do Fórum Econômico Mundial (World Economic Forum, 2025) e as análises da Dragos (Dragos, Inc., 2025), revelam uma intensificação dos ataques com potencial de causar disrupções significativas em operações industriais e serviços essenciais. Esta seção apresenta alguns dos principais incidentes de segurança cibernética que impactaram infraestruturas críticas ao longo dos anos, ilustrando a evolução dos métodos utilizados e a necessidade urgente de fortalecer a resiliência desses sistemas e também a percepção dos riscos por parte do setor industrial.

2.4.1 Infraestruturas críticas como alvos prioritários

Com a expansão da *internet*, *hackers* têm se especializado em obter acesso ilegal a dados confidenciais e a sistemas de alta criticidade (Branquinho; Branquinho, 2021). Seja por motivações políticas ou financeiras, as chamadas infraestruturas críticas tornaram-se alvos de grande risco, uma vez que a paralisação destas infraestruturas tem um impacto amplo sobre toda a sociedade. As infraestruturas críticas desempenham um papel fundamental na manutenção da funcionalidade da sociedade e da economia (Branquinho; Branquinho, 2021), fornecendo serviços essenciais, tais como eletricidade, telecomunicações, água, alimentos, aquecimento, assistência médica, transporte, serviços financeiros, bem como defesa e segurança pública. É fundamental ressaltar que essas infraestruturas não operam de forma isolada e a interdependência entre essas infraestruturas é intensa. Por exemplo, a falta de eletricidade afeta severamente a operação de quase todos os outros serviços, especialmente durante uma interrupção prolongada no fornecimento de energia.

Ao longo dos anos, houve diversos ataques bem-sucedidos contra empresas e infraestruturas críticas. Uma das principais iniciativas destinadas a documentar esses ataques foi o RISI (*Repository for Industrial Security Incidents*) (Branquinho; Branquinho, 2021), que registrou muitos incidentes relacionados à cibersegurança em setores industriais, abrangendo desde incidentes envolvendo sistemas de controle até uma ampla gama de automação industrial. Contudo, é importante observar que o repositório não recebe atualizações desde janeiro de 2015.

[8] Sistemas que integram componentes computacionais e físicos, com comunicação contínua entre o mundo digital e o real.

No Brasil, a empresa TI Safe lançou em 2020 o Incident Hub, um banco de dados público que centraliza informações sobre ciberataques coletadas de fontes públicas ou fornecidas pelas partes envolvidas (Branquinho; Branquinho, 2021). A seguir, serão apresentados alguns dos principais ciberataques conhecidos no contexto industrial, que também são apresentados com mais detalhes em (Branquinho; Branquinho, 2021; Martins; Oliveira, 2019).

2.4.1.1 ***Bomba lógica em oleoduto na Sibéria***

Durante a Guerra Fria, o serviço de espionagem da União Soviética realizou o roubo de projetos altamente sofisticados de um sistema de controle desenvolvido por uma empresa canadense. Contudo, a CIA, alegando ter conhecimento do plano de roubo, inseriu uma bomba lógica no sistema de controle em questão (Branquinho; Branquinho, 2021). Essa bomba lógica consiste em um trecho de código oculto dentro do código-fonte que, quando ativado por um gatilho lógico específico, "explode"e prejudica os componentes nos quais está inserida. Esse gatilho pode ser acionado por variáveis como *setpoint*[9], data ou controle, por exemplo. O oleoduto no qual os projetos roubados foram implementados era o Transiberiano. A ativação da bomba lógica resultou no aumento da pressão nos oleodutos, culminando em uma explosão de três quilotons em junho de 1982.

2.4.1.2 ***Ataque à estação de tratamento de resíduos na Austrália***

Em 2000, após a conclusão do sistema de controle e automação pela empresa Hunter Watertech para a companhia de água da cidade de Maroochy, Austrália, uma parte significativa da equipe foi dispensada. Um ex-funcionário, Vitek Boden, que havia sido demitido, reagiu de maneira indignada e optou por roubar um *laptop* contendo bibliotecas de desenvolvimento SCADA[10] (*Supervisory Control and Data Acquisition*), bem como um rádio com conexão direta à planta industrial, com a intenção de buscar vingança pela sua demissão.

Modificando variáveis de controle, Boden ligava e desativava bombas em momentos inadequados, o que resultou no despejo de esgoto em várias áreas da cidade. O esgoto inundou um terreno de um hotel, um parque e um rio, totalizando mais de um milhão de litros de esgoto bruto. Devido à falta de compreensão sobre ciberataques naquela época, inicialmente acreditou-se que os incidentes eram causados por defeitos nas bombas ou na fiação. Como resultado, houve um impacto ambiental significativo até que o problema fosse corretamente diagnosticado. Vitek Boden foi condenado a dois anos de prisão por suas ações (Engineering, 2021).

2.4.1.3 ***Stuxnet***

O Stuxnet foi um *worm* que demonstrou capacidade de ataque direcionado às instalações de uma usina nuclear no Irã, resultando na interrupção do processo de enriquecimento de urânio

[9] Valor de referência desejado para uma variável de controle em um sistema automatizado.
[10] Sistema usado para supervisão e controle de processos industriais em tempo real.

em 2010. Especula-se que tenha sido concebido por um indivíduo com profundo conhecimento do processo industrial iraniano, visto que explorou quatro vulnerabilidades específicas. O Stuxnet disseminou-se principalmente por meio de dispositivos USB, afetando computadores com o sistema operacional Microsoft Windows.

Uma vez infiltrado em um computador, o *worm* identificava o *software* Siemens Step 7[11], amplamente utilizado em computadores industriais para automação e monitoramento. O *malware* empregava uma senha padrão do *software* para acessar o banco de dados Microsoft SQL do sistema de controle. Posteriormente, acelerava gradualmente a velocidade das centrífugas ao longo de vários dias, visando evitar detecção por parte dos operadores, enquanto enviava dados adulterados para o sistema de supervisão, indicando que o processo estava em conformidade com as normas (Branquinho; Branquinho, 2021). Com a aceleração gradual, as centrífugas sofreram danos e explodiram quase simultaneamente, resultando na interrupção do processo de enriquecimento de urânio.

2.4.1.4 ***Dragonfly / Havex***

Em 2014, um grupo conhecido como Dragonfly empreendeu ataques a petroquímicas nos Estados Unidos (Poremba, 2014). Esse grupo empregava *e-mails* de *phishing* contendo o *malware* Havex, que infectava as máquinas alvo e passava a monitorar o protocolo OPC *classic*[12], carente de recursos de segurança (Martins; Oliveira, 2019), ao mesmo tempo em que proporcionava acesso remoto aos dispositivos. Além disso, foram utilizados outros métodos para a infecção, como *websites* comprometidos e *trojans*.

O órgão americano CISA (*Cybersecurity & Infrastructure Security Agency*) (CISA, 2021) relata que essa campanha foi associada à espionagem industrial de origem russa. Não foram identificadas funcionalidades adicionais que permitissem a modificação do *hardware* conectado. Contudo, é importante mencionar que o Havex estabelece conexão com um servidor de comando e controle[13] (C&C - *Command and Control*) que poderia, em teoria, disponibilizar novas funcionalidades para o *malware*, o que levanta a possibilidade de futuras adulterações no processo industrial.

2.4.1.5 ***Crashoverride***

O ataque a uma subestação elétrica na Ucrânia em 2016 resultou na interrupção do fornecimento de eletricidade para 80 mil residências. O *malware* Crashoverride foi concebido com o propósito de alvejar redes elétricas e causou danos significativos às instalações em Kiev. Especula-se que tenha sido um teste de conceito para avaliar as vulnerabilidades e o potencial do próprio *malware*. Assim como o Stuxnet, o Crashoverride explora o protocolo OPC, contudo, ele

[11] *Software* da Siemens utilizado para programação e configuração de controladores lógicos programáveis (CLPs).
[12] Padrão de comunicação industrial, usado para troca de dados entre sistemas de automação.
[13] Servidor utilizado por atacantes para enviar instruções e receber dados de dispositivos comprometidos em uma rede.

estabelece uma infraestrutura para executar ataques em diversos ambientes, não se restringindo a fornecedores específicos.

O CISA, mais uma vez, atribuiu essa invasão a atacantes russos (CISA, 2017). De acordo com a agência, o *malware* envia comandos válidos ao terminal remoto, desabilitando as portas COM[14] para bloquear a comunicação legítima com os dispositivos de campo. Além disso, o *malware* realiza a varredura e o mapeamento do ambiente do sistema de controle industrial (ICS - *Industrial Control System*) e torna os sistemas operacionais dos computadores totalmente inoperantes, exigindo, assim, a formatação ou a restauração a partir de *backups*. Os impactos listados incluem a perda temporária ou permanente de informações sensíveis, falhas na operação normal, perdas financeiras e prejuízo à reputação da organização.

2.4.1.6 ***Vulnerabilidade no Jeep Cherokee***

Através da rede celular, pesquisadores conseguiram invadir e modificar o sistema de controle de um Jeep Cherokee (Martins; Oliveira, 2019). Esse incidente ocorreu devido à presença de uma *backdoor* no sistema. Com o auxílio de uma pequena estação de rádio, os pesquisadores conseguiram controlar o veículo remotamente a uma distância de 70 milhas, desativando tanto o sistema de frenagem quanto o motor enquanto o veículo estava em trânsito na rodovia (WIRED, 2015). Esse ataque foi realizado como parte de um experimento destinado a ilustrar os riscos associados aos veículos conectados. Como resultado, a Fiat Chrysler promoveu um *recall*[15] de aproximadamente um milhão e meio de veículos para corrigir essa vulnerabilidade.

2.4.1.7 ***Operador de gás natural dos EUA fica paralisado por dois dias***

Um operador norte-americano de gás natural, cujo nome não foi divulgado pelo Departamento de Segurança Interna, enfrentou uma interrupção de dois dias em meados de fevereiro de 2020 devido a uma infecção por *ransomware* (Goodin, 2020). O incidente teve início com ataques de *phishing* por *e-mail*, possibilitando aos invasores a penetração na rede de TI e, por meio dela, o acesso à rede de TO. Apesar de não ter atingido os Controladores Lógicos Programáveis[16] (CLPs), a infecção resultou na perda de ferramentas essenciais de comunicação e monitoramento do processo físico, incluindo as Interfaces Homem Máquina[17] (IHMs) e a capacidade de leitura e operação remota do processo.

A usina de compressão de gás, afetada pelo ataque, possuía protocolos e planejamentos específicos para incidentes de natureza física, entretanto, não contemplava cenários de ciberataques. Dada a escassez de profissionais experientes e com autoridade para lidar com casos de ciberataque, o impacto foi considerado relativamente maior em comparação com as contingências

[14] Interfaces de comunicação serial usadas para troca de dados entre computadores e dispositivos periféricos.
[15] Chamada feita por um fabricante para reparar ou substituir produtos com defeito ou risco à segurança.
[16] Dispositivos eletrônicos utilizados para controlar máquinas e processos com base em lógica programada.
[17] Sistemas que permitem a interação entre operadores humanos e máquinas industriais, exibindo informações e recebendo comandos.

previstas nos planos de emergência. As medidas adotadas incluíram o reforço da segurança física e uma transição da operação para um estado inoperante, a qual demandou quatro horas. Apesar de apenas uma instalação ter sido diretamente afetada, outras unidades precisaram ser desativadas para prevenir o colapso das tubulações. As operações permaneceram suspensas por dois dias.

2.4.1.8 ***Incêndio em siderúrgica no Irã***

Em 27 de junho de 2022, um incêndio de grandes proporções foi registrado na siderúrgica de Khuzestan, no oeste do Irã, às margens do Golfo Pérsico (Darande, 2022). Segundo o próprio grupo responsável pelo ataque, identificado como *Predatory Sparrow* (Gonjeshke Darande), o incidente foi cuidadosamente orquestrado via acesso remoto ao sistema de controle da usina, de modo a não ferir operários presentes no local (Darande, 2022). De acordo com a cobertura da BBC News Brasil, os invasores exploraram vulnerabilidades nas IHMs da planta para desabilitar etapas críticas do processo metalúrgico, provocando o vazamento de aço fundido que culminou no incêndio nos tanques de armazenamento (Tidy, 2022). O episódio trouxe à tona a vulnerabilidade de infraestruturas industriais críticas a ataques cibernéticos sofisticados e levantou debates sobre a urgência de reforçar a cibersegurança no setor de energia e manufatura (Tidy, 2022).

2.4.1.9 ***Telecomunicações: ataque ao Kyivstar***

Em 12 de dezembro de 2023, o provedor ucraniano Kyivstar sofreu um ataque que interrompeu serviços móveis e de *internet* de milhões de assinantes, afetando também chamadas de emergência. O incidente foi atribuído ao grupo Sandworm/Solntsepek vinculado ao GRU (Diretoria Principal de Inteligência) russo (WIKIPEDIA, 2023; WIRED, 2024).

2.4.1.10 ***Fuxnet***

Em abril de 2024, o hacktivista[18] BlackJack comprometeu a empresa Moscollector, responsável por monitorar infraestrutura de gás, água e esgoto em Moscou. O *malware* Fuxnet realizou *fuzzing*[19] no barramento Modbus[20] e desativou milhares de *gateways*[21] de sensores, interrompendo a coleta de dados por até 48 horas (Team82, 2024; Thompson, 2025).

[18] Pessoa que utiliza técnicas de invasão digital como forma de protesto político ou ativismo social.

[19] Técnica de teste que envia dados malformados ou aleatórios a um sistema para identificar falhas de segurança ou erros.

[20] Padrão de comunicação serial amplamente usado na automação industrial para troca de dados entre dispositivos mestres e escravos.

[21] Dispositivos que fazem a interligação e conversão de protocolos entre redes diferentes, permitindo a comunicação entre sistemas distintos.

2.4.1.11 ***FrostyGoop***

Também em abril de 2024, a Dragos identificou o *malware* FrostyGoop, o primeiro a usar Modbus TCP[22] para manipular dispositivos de controle industrial. Relatos indicam que controladores de aquecimento em Lviv, Ucrânia, ficaram sem funcionamento por dois dias (Dragos, 2024; Nather, 2024).

2.4.1.12 ***Campanha estatal: Volt Typhoon***

De acordo com a CISA, o grupo Volt Typhoon, associado à inteligência chinesa, obteve acessos persistentes a redes de energia, telecomunicações, transporte e sistemas de água nos Estados Unidos, visando preposicionar *malwares* para futuras disrupções (CISA, 2024; Nextgov, 2025).

De acordo com (Branquinho; Branquinho, 2021), está previsto o desenvolvimento contínuo de ataques cibernéticos cada vez mais sofisticados e poderosos, com um foco na paralisação de infraestruturas críticas. Esses ataques podem ser orquestrados por atores independentes, grupos ideológicos ou até mesmo financiados por forças militares. Os *hackers* buscam explorar vulnerabilidades que ainda não têm correção disponível, de modo que os ataques não possam ser parados. Portanto, é indispensável que empresas e governos intensifiquem seus investimentos em segurança nas infraestruturas críticas, com o objetivo de minimizar vulnerabilidades e corrigi-las prontamente assim que forem identificadas. Além disso, deve-se fornecer treinamento adequado e orientações para os colaboradores, uma vez que eles também podem representar um grande risco em relação à cibersegurança.

2.5 PANORAMA DE PERCEPÇÃO DE RISCO DE CIBERSEGURANÇA

O ano de 2024 marcou um crescimento significativo nas ameaças cibernéticas contra ambientes industriais. Segundo o relatório anual da Dragos, os ataques de *ransomware* direcionados a organizações com sistemas de controle industrial (ICS) aumentaram 87 % em relação ao ano anterior, e o número de grupos de ameaça ativos cresceu 60 % nesse mesmo intervalo (Dragos, Inc., 2025). Entre os grupos mais ativos, destacam-se KAMACITE, ELECTRUM, BAUXITE e GRAPHITE, com campanhas voltadas a setores como energia, transporte e manufatura.

A Dragos define *malware* direcionado a ICS como aquele que atende a três critérios: (i) capacidade de afetar componentes de controle industrial (ICS-*capable*), (ii) intencionalidade maliciosa e (iii) objetivo explícito de impactar fisicamente processos industriais. O relatório também evidencia falhas graves na comunicação de vulnerabilidades: 22 % dos avisos publicados por fornecedores ou fontes públicas continham erros técnicos ou de criticidade, e em quase metade desses casos (47 %), a Dragos precisou desenvolver mitigações alternativas (Dragos, 2024).

[22] Versão do protocolo Modbus adaptada para redes Ethernet.

2.5.1 Tendências globais de riscos cibernéticos

No *Global Risks Report 2025*, publicado pelo WEF (*World Economic Forum* - Fórum Econômico Mundial), a insegurança cibernética aparece como um dos riscos globais mais relevantes no longo prazo. A categoria "espionagem e guerra cibernética" é listada como o quarto maior risco para a próxima década, perdendo apenas para desinformação generalizada, eventos climáticos extremos e polarização social (MUNDIAL, 2023).

O relatório ressalta que os riscos cibernéticos estão fortemente interligados com riscos geopolíticos e sociais, como o aumento da propaganda digital, ataques a infraestruturas críticas e manipulação da informação em larga escala. O avanço da inteligência artificial gerativa e a proliferação de conteúdos sintéticos (*deepfakes*[23]) agravam o cenário de desinformação, aumentando a superfície de ataque e dificultando a detecção de ameaças em tempo hábil (World Economic Forum, 2025).

2.5.2 Comparativo de riscos: WEF 2023 vs 2024 vs 2025

A Tabela 1 apresenta uma comparação entre os principais riscos globais identificados nos relatórios de 2023, 2024 e 2025 do WEF, considerando o horizonte de curto prazo (próximos dois anos). Nota-se a persistência dos riscos relacionados à desinformação, polarização social e insegurança cibernética como tópicos recorrentes no topo da lista.

Tabela 1 – Comparativo de riscos globais — curto prazo (WEF 2023–2025)

Risco	**WEF 2023**	**WEF 2024**	**WEF 2025**
Desinformação e *fake news*	16º	1º	1º
Eventos climáticos extremos	2º	2º	2º
Polarização social	5º	3º	3º
Insegurança cibernética / espionagem	**8º**	**5º**	**4º**
Escassez de recursos naturais	9º	4º	5º

Fonte: (World Economic Forum, 2023; World Economic Forum, 2024; World Economic Forum, 2025).

Este capítulo apresentou os principais elementos da segurança cibernética - dos pilares de disponibilidade, integridade e confidencialidade nas camadas de TO e TI, aos agentes de ameaça, tipos de ataque e incidentes emblemáticos que evidenciam a urgência de uma defesa multicamadas - e, por fim, analisou a evolução da percepção de risco global por meio dos relatórios do WEF. O próximo capítulo se dedica a aplicar esses conceitos ao domínio dos conversores estáticos, explorando suas superfícies de ataque específicas e propondo estratégias de mitigação alinhadas às exigências da indústria 4.0.

[23] Conteúdos falsos gerados com inteligência artificial que simulam rostos, vozes ou vídeos de pessoas reais com alta fidelidade.

3 SEGURANÇA CIBERNÉTICA EM CONVERSORES ESTÁTICOS

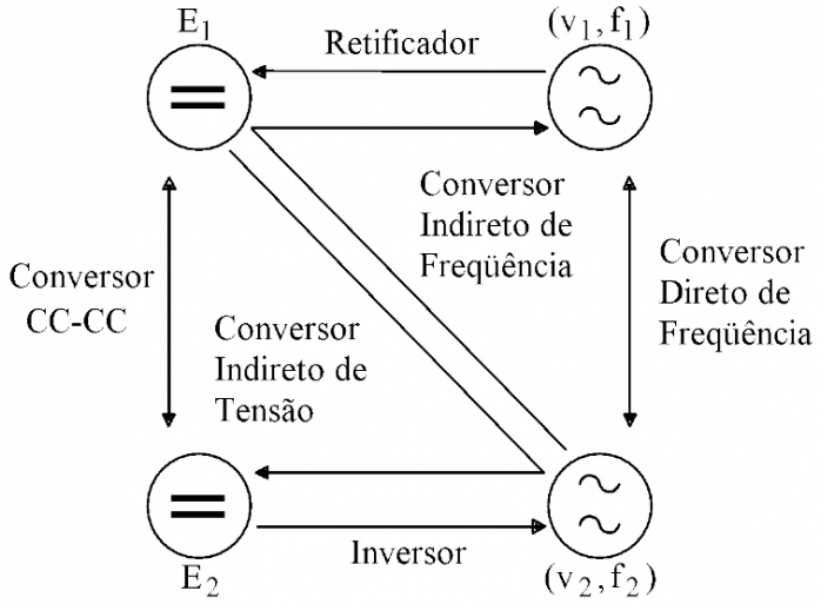
Este capítulo explora a interface entre conversores estáticos e segurança cibernética. Como os sistemas modernos dependem de microcontroladores, esses dispositivos podem se tornar pontos de entrada para diferentes tipos de ataque. Primeiro, serão apresentados os princípios de funcionamento e a importância dos conversores estáticos; em seguida, será detalhada sua natureza e as vulnerabilidades mais comuns. A revisão bibliográfica abrange as linhas de pesquisa mais citadas em eletrônica de potência e cibersegurança, destacando estudos como o de Balda et al. (Balda et al., 2017) e as recomendações consolidadas em *Segurança Cibernética Industrial* (Branquinho; Branquinho, 2021).

3.1 CONVERSORES ESTÁTICOS

Conversores são dispositivos eletrônicos responsáveis por adequar e controlar o fluxo de energia entre a rede de distribuição e os produtos que utilizam energia elétrica (Barbi, 2006). São compostos por duas partes, sendo uma delas de potência (que faz a adequação da energia) e a outra de controle (responsável por acionar os componentes da parte de potência corretamente).

A Figura 2 mostra os principais métodos de operação de um conversor, que pode ser de maneira direta ou de maneira indireta, passando por um estágio intermediário (Martins; Oliveira, 2019). A parte de potência deve ser projetada para suportar os níveis de tensão e corrente para o funcionamento do produto que ele está fornecendo energia.

Figura 2 – Principais funções dos conversores estáticos.



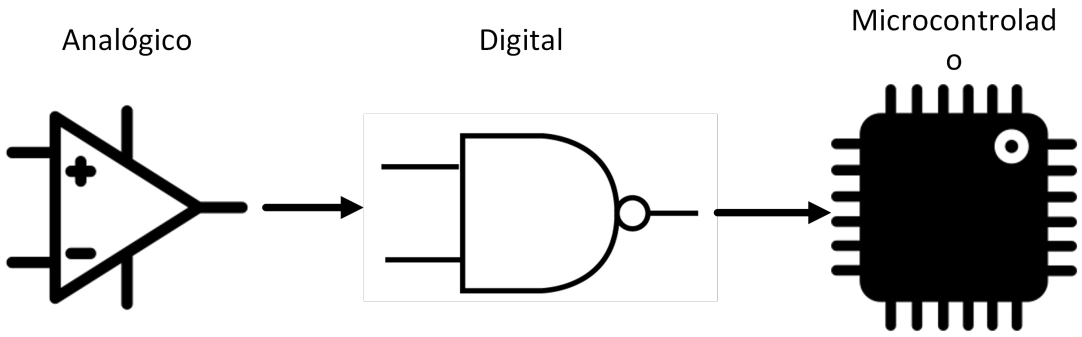
Fonte: Adaptado de (Barbi, 2006).

Apesar de, ao longo dos anos, ter sido observado a crescente complexidade estrutural

dos conversores de potência, seus fundamentos mantêm-se firmes: um elemento acumulador de energia e semicondutores para modular o fluxo, garantindo elevada robustez. Entretanto, justamente nesse ponto de estabilidade da parte de potência, a parte de controle viveu uma trajetória oposta.

Partindo da eletrônica analógica, mudando para a eletrônica digital e, por fim, incorporando o uso de microcontroladores (conforme ilustrado na Figura 3), o controle dos conversores passou por transformações profundas. Essa evolução drástica possibilitou a implementação de algoritmos de controle cada vez mais sofisticados em intervalos de tempo cada vez menores, permitindo que mesmo pequenas alterações na eletrônica de potência resultassem em ganhos significativos de eficiência por meio de estratégias de controle mais avançadas.

Figura 3 – Mudança dos circuitos de comando e controle dos conversores.



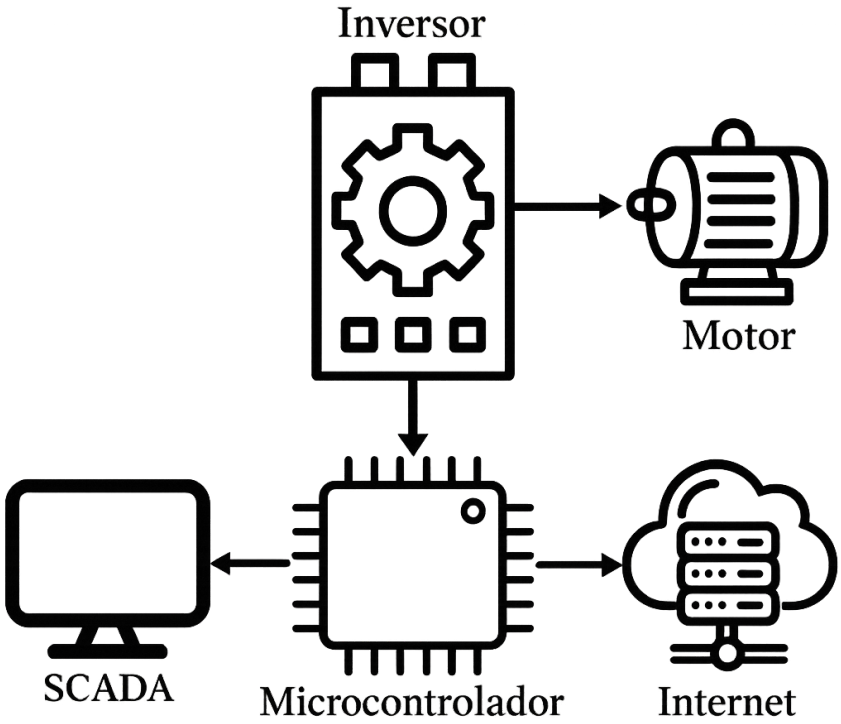
Fonte: do Autor.

Com as demandas para conectar os dispositivos que a indústria 4.0 trouxe, os microcontroladores passaram a racionar seu processamento com novas tarefas, como, por exemplo, a comunicação dos dispositivos por meio de diversas redes. Essa conexão dos conversores trouxe várias vantagens, como o monitoramento, atualização de parâmetros e até o controle remoto da operação deles. A Figura 4 ilustra como os microcontroladores passaram a executar diversas tarefas simultaneamente, como o controle do conversor, a comunicação com outros dispositivos e com a *internet*, bem como o monitoramento de eventos por sistemas SCADA.

Essas implementações de conversores usando microcontroladores foram realizadas por muito tempo pelos engenheiros que projetaram a parte de potência, sem muita preocupação com o tratamento dos dados recebidos e enviados pelo conversor. Na maioria das vezes isso não foi um problema, tendo em vista que as redes nas quais estavam conectados eram isoladas tanto logicamente, como fisicamente.

A interconexão entre as redes de TO e de TI torna esses dispositivos, que estavam operando de maneira isolada, visíveis para toda a *internet*. Com essa exposição, os conversores que não estão preparados, passam a exibir sérios riscos para uma organização devido a inúmeras vulnerabilidades (muitas vezes desconhecidas ou simplesmente ignoradas), que vão sendo descobertas e ativamente exploradas.

Figura 4 – Multitarefas em microcontroladores.



Fonte: do Autor.

3.2 VULNERABILIDADES

Os conversores estáticos, componentes essenciais em acionamentos elétricos e sistemas de eletrônica de potência, quando conectados à rede, apresentam múltiplas superfícies de ataque que podem ser exploradas. Nesta seção, analisam-se as principais vulnerabilidades desses dispositivos, considerando conectividade IIoT[1] (*Industrial Internet of Things - Internet* Industrial das Coisas), protocolos de comunicação, *firmware* e *hardware*, bem como questões de segurança física e da cadeia de suprimentos.

3.2.1 Conectividade e protocolos de comunicação

A conectividade de conversores estáticos requer a adoção de interfaces de rede que viabilizem funções de monitoramento, atualização de parâmetros e controle remoto. Entre os protocolos mais comuns, destaca-se o Modbus/TCP, cuja versão original carece de mecanismos robustos de autenticação, confidencialidade e integridade de dados. Assim, ataques DoS, interceptação de pacotes (MITM) e injeção de comandos maliciosos tornam-se viáveis, sobretudo quando as comunicações não são criptografadas ou devidamente autenticadas (Martins; Oliveira, 2022).

Protocolos como Modbus/TCP Security (versão com TLS) oferecem mais segurança ao prover confidencialidade e integridade, mas ainda precisam de atenção na gestão de certificados digitais e na implementação de RBAC (*Role Based Access Control* - Controle de Acesso por

[1] Extensão da Internet das Coisas (*IoT*) voltada para aplicações industriais, conectando máquinas, sensores e sistemas para monitoramento e automação em tempo real.

Função). A adoção incorreta desses mecanismos, ou a falta de suporte por parte de dispositivos antigos, pode levar à exposição de vulnerabilidades críticas. Observa-se, também, que vulnerabilidades em bibliotecas de TCP/IP[2] e SSL/TLS[3] em suas versões mais antigas, caso utilizadas, podem comprometer toda a comunicação do conversor (Balda et al., 2017; Martins; Oliveira, 2022).

3.2.2 Atualização de *firmware*

O *firmware* de conversores estáticos é responsável por implementar funções de controle, protocolos de comunicação e outras funções internas. Métodos de atualização de *firmware* inseguros, tais como ausência de verificação de assinatura digital[4] ou processos manuais de atualização sem garantia de integridade, expõem o dispositivo a ataques de modificação de *firmware*. Não utilizar técnicas de inicialização segura e de raiz de confiança[5] (RoT - *Root of Trust*) permite que versões adulteradas de *firmware* sejam inicializadas, comprometendo não só o conversor, mas podendo criar uma nova superfície de ataque pela qual um *hacker* pode invadir a rede industrial (Arthur; Challener; Goldman, 2015). Para reduzir esse risco, algumas medidas podem ser tomadas, como somente instalar *firmwares* assinados digitalmente por Autoridades Certificadas[6] (CA - *Certified Authorities*).

3.2.3 Vulnerabilidades físicas e de *hardware*

Conversores estáticos podem sofrer ataques físicos, dada a natureza crítica de componentes de eletrônica de potência no ramo industrial. Tentativas de acesso não autorizado ao *hardware*, como inserção de dispositivos de injeção de sinal ou sondas capacitivas sobre os conversores, permitem que invasores alterem parâmetros de operação ou induzam falhas de maneira imperceptível ao operador. Além disso, o acesso às portas seriais e interfaces de programação (tais como JTAG ou SPI) possibilita leitura e modificação de *firmware*, extração de informações sensíveis (por exemplo, chaves criptográficas) ou instalação de kits de exploração de vulnerabilidades de baixo nível (Balda et al., 2017).

O projeto do enclausuramento deve considerar implementar técnicas de proteção contra adulteração e de detecção de aberturas não autorizadas. Quando cabível, a utilização de sensores de intrusão e mecanismos de bloqueio criptográfico dos módulos de potência e controle aumenta a resiliência contra ataques físicos. Os ataques físicos podem chegar ao ponto de danificar o *hardware* do conversor, como por exemplo, a remoção de um cartão SD (*Secure Digital*) que

[2] Conjunto de protocolos de comunicação que define como os dados são transmitidos na internet e em redes locais.

[3] Protocolos criptográficos usados para proteger a comunicação de dados em rede, garantindo confidencialidade, integridade e autenticação.

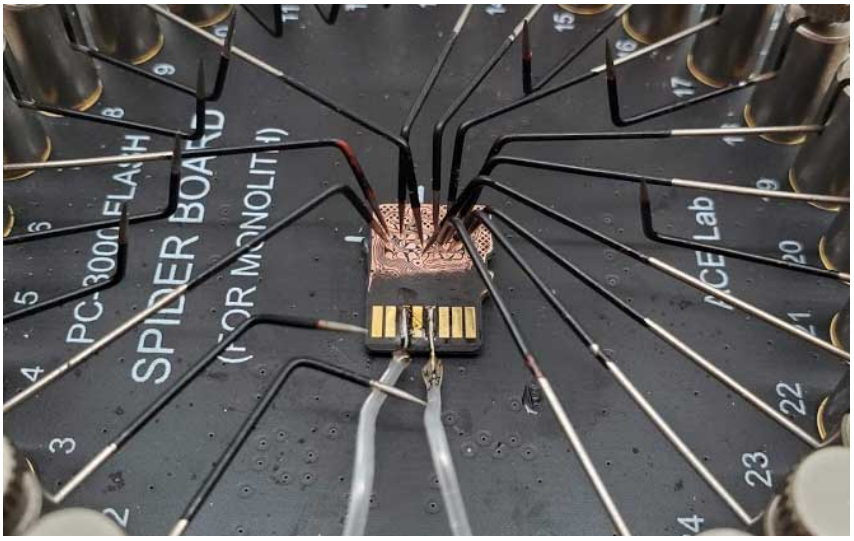
[4] Mecanismo criptográfico que garante a autenticidade e a integridade de dados eletrônicos, associando-os de forma única ao signatário.

[5] Componente fundamental em um sistema de segurança que serve como base confiável para validar identidades e integridade de dados.

[6] Entidades confiáveis responsáveis por emitir e gerenciar certificados digitais usados para autenticação e criptografia.

armazena dados críticos. A Figura 5 mostra a recuperação de dados de um cartão SD que foi formatado. Isso mostra como vulnerabilidades físicas podem levar à perda de dados críticos, que poderiam ser explorados por invasores para comprometer a segurança do sistema.

Figura 5 – Processo de recuperação de dados de cartão SD.



Fonte: (Ferrick, 2024).

3.2.4 Vulnerabilidades de camada operacional (TO)

Em TO, em que conversores estáticos interagem com sistemas SCADA e PLCs (*Programmable Logic Controllers* - Controladores Lógico Programáveis), a alta criticidade dos processos exige que vulnerabilidades no conversor não provoquem impactos em cascata sobre toda a planta, aumentando assim a disponibilidade do recurso. A inserção de tráfego forjado, alteração de valores de referência ou desvio de rotinas de proteção podem resultar em sobrecargas elétricas, danificando máquinas conectadas e interrompendo a produção de uma fábrica. Cabe destacar que equipamentos de proteção integrados ao conversor podem ser desabilitados por meio de manipulação de comandos (Martins; Oliveira, 2022). O monitoramento de integridade em tempo real e a segmentação por zonas de confiança[7] contribuem para detectar e conter comportamentos anômalos.

3.2.5 Cadeia de suprimentos e segurança do ciclo de vida

A confiabilidade de conversores estáticos depende também da procedência dos componentes eletrônicos, mas principalmente de *software* de terceiros integrados ao produto. O uso de *hardwares* comprometidos ou de bibliotecas de código com vulnerabilidades conhecidas (por exemplo, camadas de TCP/IP, bibliotecas criptográficas) durante o processo de fabricação amplia a superfície de ataque. A elaboração de lista de materiais indicando sua procedência e código do

[7] Segmentações lógicas ou físicas dentro de uma rede industrial com diferentes níveis de segurança e acesso, usadas para limitar o impacto de ataques.

fabricante (tanto para *hardware* como para *software*), bem como, a exigência de evidências de testes de segurança por fornecedores, contribuem para equilibrar o risco de ataques à cadeia de suprimentos (Michael et al., 2022).

Além disso, a realização de testes de penetração[8] periódicos e de análises de vulnerabilidades em componentes críticos durante as fases de compra, produção e manutenção minimiza a probabilidade de introdução de *backdoors* ou códigos maliciosos (Michael et al., 2022).

3.2.6 Vulnerabilidades de criptografia e gerenciamento de chaves

Algumas implementações de TLS/SSL em canais de comunicação entre o conversor e o servidor remoto podem adotar configurações criptográficas fracas ou obsoletas, tais como:

- algoritmos ultrapassados;
- chaves RSA com tamanho inferior a 2048 bits;
- suporte a versões descontinuadas do protocolo (por exemplo, TLS 1.0).

Adicionalmente, o armazenamento de chaves diretamente em *firmware*, sem recorrer a um mecanismo de segurança dedicado, aumenta o risco de exposição das credenciais (Communications, 2023).

A adoção de um TPM (*Trusted Platform Module* - Módulo de Plataforma Confiável) 2.0 — que oferece uma Raiz de Confiança, hierarquias de chaves e políticas de validade e atualização periódica — eleva consideravelmente o nível de segurança da criptografia embarcada (Arthur; Challener; Goldman, 2015).

Os cenários de ataque utilizados nos testes priorizam vetores recorrentes na literatura, como negação de serviço em Modbus/TCP, adulteração de *firmware* e manipulação física do *hardware*. Essas abordagens refletem os eventos mais comuns observados em infraestruturas críticas e, portanto, servem como base representativa para a avaliação experimental.

Em síntese, os conversores estáticos, embora essenciais para o controle e a eficiência dos sistemas de eletrônica de potência, tornam-se vetores críticos de ataque assim que são conectados às redes industriais e à *internet*. Ao longo deste capítulo, demonstrou-se que suas múltiplas superfícies de vulnerabilidade — desde protocolos legados[9] como Modbus/TCP sem criptografia até falhas em processos de atualização de *firmware* e exposições físicas em interfaces JTAG ou portas seriais — podem ser exploradas para comprometer a disponibilidade, a integridade e a confidencialidade dos sistemas de automação. A adoção de boas práticas, como autenticação forte, uso de TLS em comunicações, validação de assinaturas digitais em *firmware*, segmentação

[8] Simulações controladas de ataques cibernéticos realizadas para identificar vulnerabilidades exploráveis em sistemas ou redes.

[9] Protocolos antigos ainda em uso, mas que geralmente não atendem aos requisitos modernos de segurança e desempenho.

de redes OT/IT, implementação de TPMs e auditorias de cadeia de suprimentos, é altamente recomendável para mitigar esses riscos. Somente um projeto que integre desde o nível físico até o ciclo de vida completo do conversor, aliando robustez de *hardware*, segurança de *software* e governança de processos, garantirá que essas peças-chave da indústria 4.0 operem sem expor as organizações a ameaças cibernéticas cada vez mais sofisticadas.

4 FERRAMENTAS DE SEGURANÇA CIBERNÉTICA PARA CONVERSORES ESTÁTICOS

Nesta seção são apresentadas as principais ferramentas de *hardware* e *software* empregadas na operação e segurança de conversores estáticos, com ênfase em dois elementos-chave: o microcontrolador responsável pela aquisição de dados, geração de sinais de controle e comunicação com redes externas e o módulo TPM, que provê a Raiz de Confiança para as funções críticas de segurança.

4.1 FUNDAMENTOS DE UM MICROCONTROLADOR

Um microcontrolador é um dispositivo de computação embarcado em um único Circuito Integrado (CI), projetado para controlar dispositivos eletroeletrônicos. Sua arquitetura típica inclui:

4.1.1 Unidade Central de Processamento (CPU)

Geralmente baseada em um núcleo RISC (Computador com Conjunto de Instruções Reduzidas - Reduced Instruction Set Computer), executando cálculos para equações de controle.

4.1.2 Memória Flash e RAM

A Flash armazena o *firmware*; a RAM armazena variáveis e dados temporários.

4.1.3 Periféricos Integrados

*4.1.3.1 **Temporizadores/Contadores***

Para geração de PWM e medições de tempo.

*4.1.3.2 **Conversores AD/DA***

Para interface analógica com sensores e atuadores.

*4.1.3.3 **Módulos de Comunicação***

Vários protocolos como SPI, I²C, UART, CAN, Ethernet para troca de dados com outros dispositivos.

4.1.4 Barramento de Sistema

Interconecta CPU, memórias e periféricos.

4.1.5 Sistema de Interrupções

Permite tratamento em tempo real de eventos externos, crítico em controle de conversores.

Para o controle de um conversor estático, o microcontrolador deve adquirir medições de tensão e corrente em alta velocidade, executar algoritmos de controle (PID, vetorial, preditivo, etc.) e gerar sinais PWM com sincronismo preciso.

4.2 COMPARAÇÃO DE MICROCONTROLADORES

A escolha do microcontrolador influencia diretamente o desempenho do controle e a segurança da comunicação em conversores estáticos. Entre os fatores mais importantes estão a capacidade de processamento, a presença de periféricos analógicos de alta resolução e o suporte a módulos de criptografia por *hardware*. Recursos de memória também são relevantes para acomodar pilhas de comunicação e bibliotecas de segurança, especialmente quando se utiliza um RTOS (discutido mais a frente).

Para este trabalho, uma placa Nucleo com o microcontrolador STM32H753ZI foi utilizada por ter sido doada por um terceiro já acompanhada do módulo de TPM SC-KTPM-RASPIKG9. Outra opção considerada foi o F28388D Control Card, da Texas Instruments, cuja capacidade é superior para controle digital de potência (considerando principalmente os periféricos). No entanto, devido a dificuldades na configuração dessa plataforma optou-se por não utilizá-la.

A Figura 6 resume características de diferentes microcontroladores e DSPs consultados durante a pesquisa. Modelos com maior frequência de operação e periféricos avançados, como conversores AD[1] de alta velocidade e PWMs de alta resolução, são preferidos para aplicações de potência. Para comunicação segura, destacam-se os que oferecem aceleração criptográfica ou módulos dedicados como TPM.

[1] Dispositivos que realizam a conversão de sinais analógicos em digitais, permitindo que sistemas digitais processem dados do mundo real.

Figura 6 – Comparação de microcontroladores e DSPs para controle de conversores estáticos.

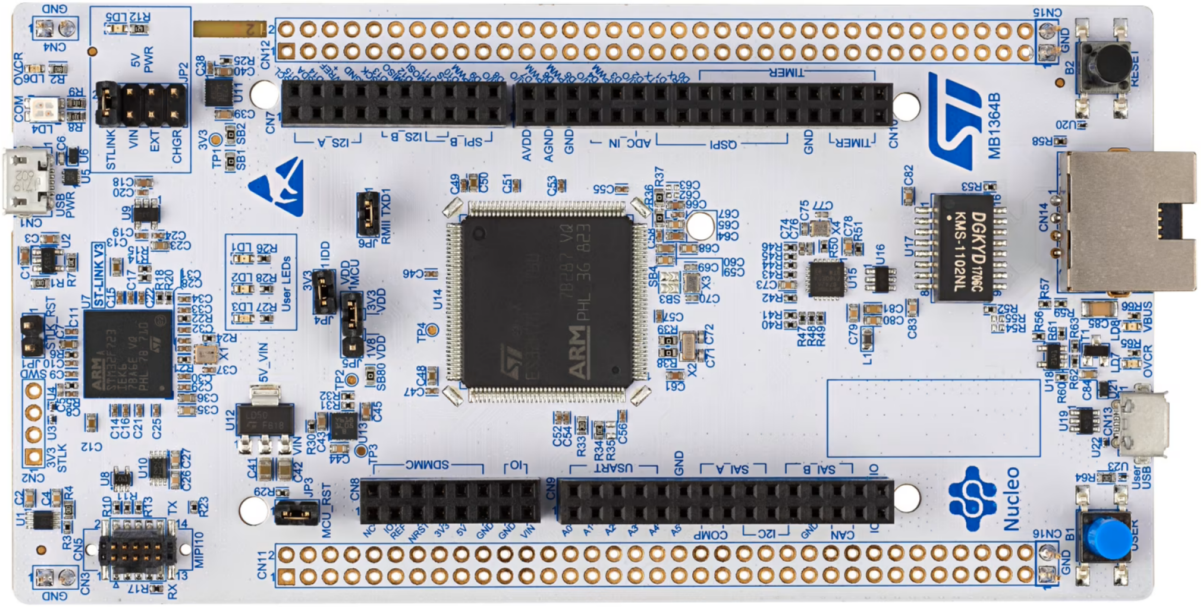
Fabricante	STM	STM	STM	TI	STM	TI	TI	Analog	Analog
Modelo	STM32MP157	STM32H730	STM32F756ZG	TMS320F28379	STM32H753ZI	TMS320F28388	TMS320C6678	ADSP-21992	ADSP-SC594
Categoria	MPU	MCU	MCU	MCU	MCU	MCU	DSP	DSP	DSP
Processador(es)	2x ARM Cortex A7	ARM Cortex M7	ARM Cortex M7	2x TMS320C28x	ARM Cortex M7	2x TMS320C28x	8x TMS320C66x	ADSP-2199x DSP core	ARM Cortex A5
Arquitetura	32-bit	32-bit	32-bit	32-bit	32-bit	32-bit	32-bit	16-bit	32-bit
Precisão (FPU)	double-precision	double-precision	single-precision	single-precision	double-precision	double-precision	double-precision	NO (fixed point)	double-precision
Clock	800MHz	550MHz	216MHz	200MHz	480MHz	200MHz	1,25GHz	160MHz	1GHz
GPU	SIM	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO
RAM	708 KB	564 kB	320 kB	128 kB	1 MB	128 KB + (44KB / nucleo)	576kB / núcleo	96 kB	32kB L1 + 256 kB L2
ADC	2	2	3	4	3	4	0	2	1
Resolução ADC	16-bit (3.6 Msps)	16-bit (3.6 Msps)	12-bit (2.4 Msps)	16-bit (1.1 Msps)	16-bit (3.6 Msps)	16-bit (1.1 Msps)	-	14-bit (20 Msps) (Será?)	12-bit (1 Msps)
Worst Conversion time	41,18us	131,84us		915ns (16-bit)	131.84us	915ns (16-bit)	-	725ns (all channels converted)	
Best Conversion time	22,88us	16,48us	16,20us	280ns (12-bit)	16.20us	280ns (12-bit)	-	375ns	
DAC	2	2	2	3	2	3	0	NÃO	NÃO
Resolução DAC	12-bit	12-bit	12-bit	12-bit	12-bit	12-bit	-	NÃO	NÃO
Timers	2 x 32-bit	4 x 32-bit	2 x 32-bit	6 x 32-bit	6 x 32-bit + 1 HRTIM	6 x 32-bit	16 x 64-bit	3 x 32-bit	16 x ?-bit
PWM	4 PWM por timer	4 PWM por timer	4 PWM por timer	8 PWM + 16 (High-resolution PWM)	4 PWM por timer	16 PWM + 16 (High resolution PWM)	0	3-phase PWM + 2 aux PWM	16
I/O	176	128	168	169	168	169	16	16	135
DMA	3	4	4	2	4	2	3	1	1
I^2C	6	5	4	2	4	2	1	NÃO	6
UART	4	5	4	4	4	4	1	Software emulation	4
SPI	6	6	6	3	6	4	1	1	4
CAN	2	3	2	2	2	2	0	1	1
USB	2	1	2	1	2	1	NÃO	NÃO	1
Ethernet	SIM	SIM	SIM	NÃO	SIM	SIM	SIM	NÃO	SIM
Criptografia acelerada por hardware	HASH (MD5, SHA-1, SHA224, SHA256), HMAC	AES 128, 192, 256, TDES, HASH (MD5, SHA-1, SHA-2), HMAC	AES 128, 192, 256, triple DES, HASH (MD5, SHA-1, SHA-2), HMAC	NÃO	AES 128, 192, 256, SHA1, SHA2, HMAC, TRNG	AES 128, 192, 256	ECB, CBC, CTR, F8, A5/3, CCM, GCM, HMAC, CMAC, GMAC, AES, DES, 3DES, Kasumi, SNOW 3G, SHA-1,	NÃO	AES 128, 192, 256, ECB, CBC, ICM, CTC, DES, ARC4, 3DES, SHA-1, SHA-2, HMAC, MD5
Hardware CRC	2	1	1	1	1	1	0	Talvez (Pág 9 diz que o protocolo tem, mas não explicita se no DSP tem)	2
True Random Number Generator	2	1	1	NÃO	1	NÃO	NÃO	NÃO	SIM

Fonte: do Autor.

4.2.1 Estrutura do projeto

O projeto foi criado utilizando o *STM32CubeIDE* para a placa Nucleo-H753ZI, que é uma placa de desenvolvimento mostrada na Figura 7 conectada ao módulo de TPM SC-KTPM-RASPIKG9, exibido na Figura 8 por meio da interface de comunicação SPI. Ao criar o projeto na IDE[2] (Ambiente de Desenvolvimento Integrado - *Integrated Development Environment*), já são incluídos os arquivos de inicialização da placa, drivers HAL[3] (camada de abstração de *hardware* – *Hardware Abstraction Layer*) e o código-fonte principal.

Figura 7 – Placa de desenvolvimento Nucleo-H753ZI.



Fonte: (STMicroelectronics, 2025a).

O ambiente de desenvolvimento *STM32CubeIDE* organiza a estrutura do projeto em diferentes painéis, como ilustrado na Figura 9. No conjunto de janelas abertas na figura, ficam visíveis o código fonte, a árvore de arquivos e as opções de compilação, o que facilita a navegação entre os diretórios gerados automaticamente pela IDE.

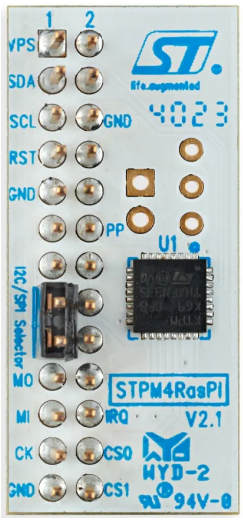
A Figura 10 apresenta a interface de configuração *.ioc* do microcontrolador, na qual podem ser habilitados periféricos, definidos pinos e ajustada a frequência do *clock* de forma gráfica. Sempre que esse arquivo é alterado, o *STM32CubeIDE* atualiza os códigos de inicialização, reduzindo o trabalho manual ao prototipar. É através da interface *.ioc* que é feita a inclusão do FreeRTOS no projeto, permitindo a configuração de tarefas, filas e temporizadores, assim como a inclusão de bibliotecas adicionais como wolfSSL e wolfTPM.

Dentro da pasta `Core` localizam-se os arquivos `main.c`, funções de inicialização e configuração do *clock*. Os `Drivers` contêm o HAL da ST para o microcontrolador STM32H7.

[2] *Software* que integra em uma única interface editor de código, compilador e depurador, facilitando o desenvolvimento de *software*.

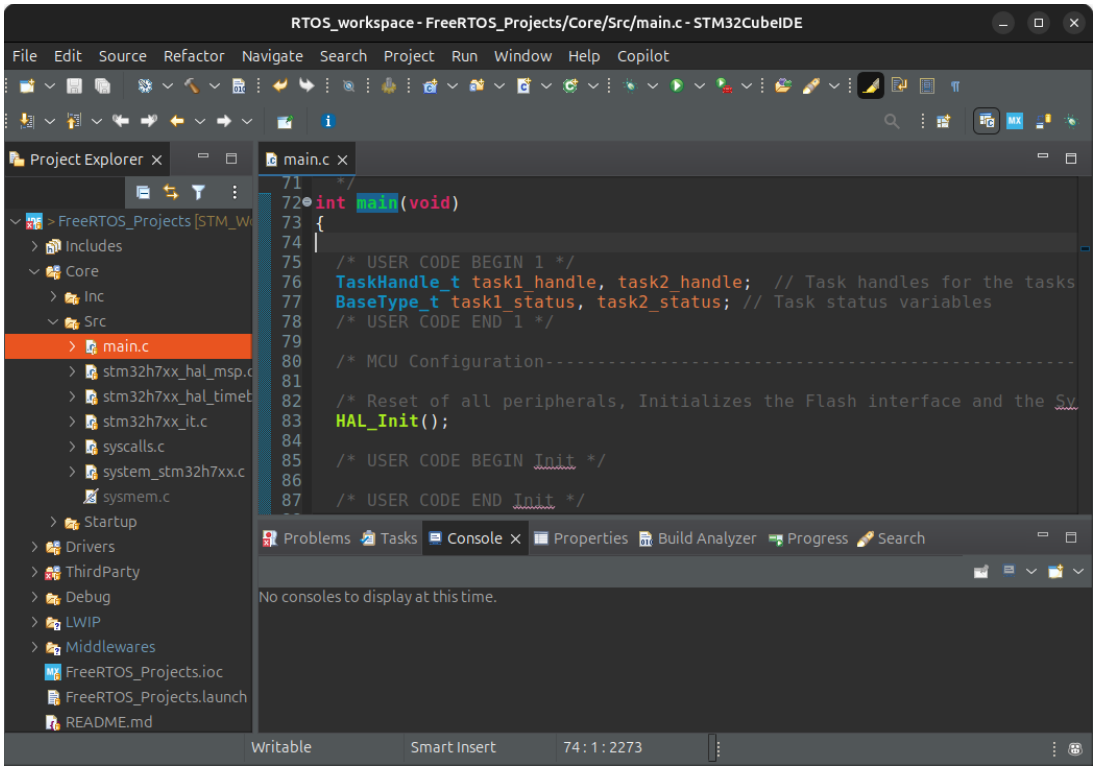
[3] Interface de *software* que oculta os detalhes do *hardware*, permitindo que o código seja desenvolvido de forma mais portátil e modular.

Figura 8 – Módulo de TPM SC-KTPM-RASPIKG9.



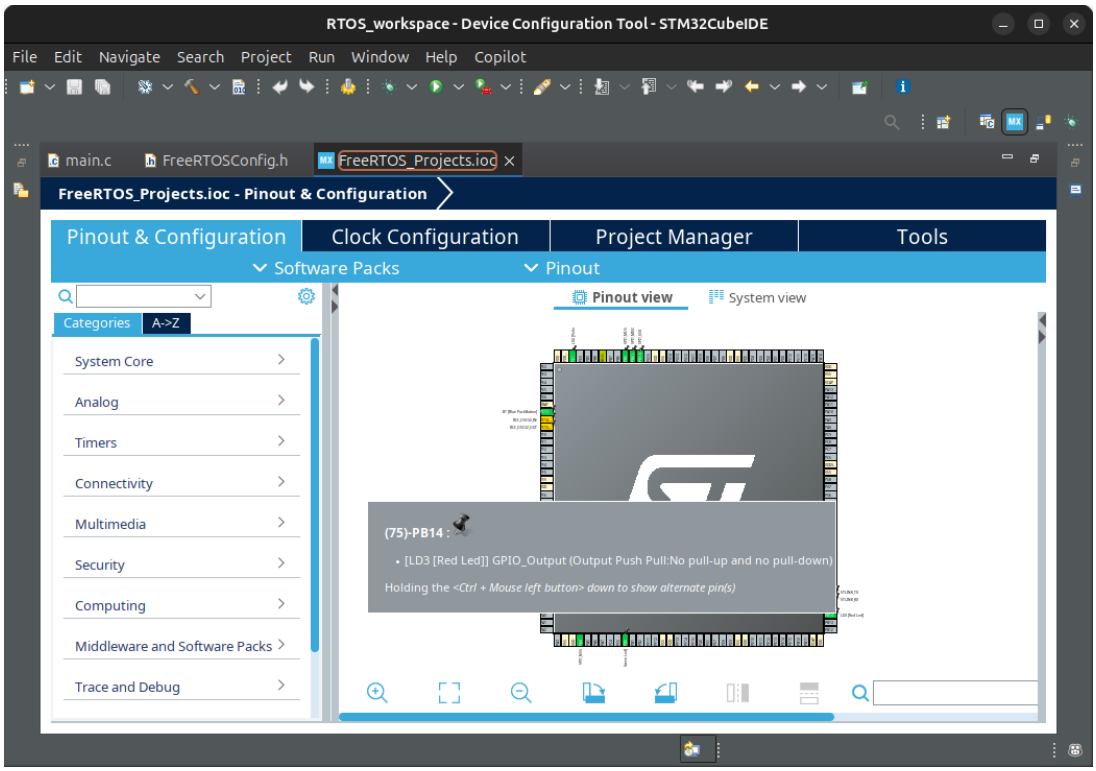
Fonte: (STMicroelectronics, 2025b).

Figura 9 – Janela principal do STM32CubeIDE.



Fonte: do Autor.

Figura 10 – Configuração do microcontrolador no arquivo .ioc.



Fonte: do Autor.

Na pasta `ThirdParty` encontram-se o código do FreeRTOS propriamente dito e as bibliotecas adicionais.

4.3 TRUSTED PLATFORM MODULE - TPM

O Trusted Platform Module (TPM) é um coprocessador criptográfico padronizado que fornece Raiz de Confiança para operações sensíveis, como armazenamento de chaves, assinatura digital e medição de integridade de *software* (Arthur; Challener; Goldman, 2015).

4.3.1 Histórico e uso

O TPM foi concebido para proteger Sistemas Operacionais contra modificações não autorizadas e ataques de *software*, oferecendo:

- Geração e armazenamento seguro de chaves (RSA, ECC).
- Registros de configuração em PCRs[4] (Registradores de Configuração de Plataforma - *Platform Configuration Registers*) para atestação do estado de inicialização.

[4] Registros internos do TPM usados para armazenar valores de integridade calculados durante o processo de inicialização do sistema.

- Proteção contra ataques de dicionário[5] a senhas de autenticação (Arthur; Challener; Goldman, 2015).

Com o advento do TPM 2.0, possibilitou-se o uso de múltiplos algoritmos criptográficos e políticas de autorização mais avançadas, como o EA (Autorização Avançada - *Enhanced Authorization*) (Arthur; Challener; Goldman, 2015).

4.3.2 Origem e foco inicial

O TPM foi descrito no início dos anos 2000 por quem se tornou o Trusted Computing Group (TCG) como uma Raiz de Confiança de baixo custo para computadores pessoais e servidores comerciais. Inspirado em coprocessadores de segurança, usados em servidores, o TPM 1.1b — lançado em 2003 — foi projetado para ser soldado diretamente à placa-mãe de PCs (Computador Pessoal - *Personal Computer*), utilizando interfaces de barramento padrão (como SPI) já presentes em *desktops*, de modo a não elevar substancialmente o custo das plataformas existentes (Arthur; Challener; Goldman, 2015).

A ênfase original estava na proteção de chaves criptográficas, na medição de integridade do *firmware* (via PCRs) e em operações de assinatura digital, todas visando o ambiente de computadores pessoais e industriais, em que surgia o maior volume de ameaças relacionadas a código malicioso e roubo de dados. A decisão de deslocar ao máximo a lógica não essencial para o *software* e manter o *chip* o mais simples possível veio da necessidade de manter o custo baixo (Arthur; Challener; Goldman, 2015).

Em novembro de 2013, o "Relatório para o Presidente: Oportunidades Imediatas para o Fortalecimento da Cibersegurança Nacional"recomendou a adoção universal do TPM, como um *microchip* desenvolvido para prover funções relacionadas à segurança básica, incluindo em telefones e tablets (Arthur; Challener; Goldman, 2015). Embora haja iniciativas para uso em IIoT e automação industrial, até hoje o TPM permanece quase onipresente em *desktops* e servidores, mas pouco incorporado em sistemas embarcados, reforçando seu foco inicial nesses mercados.

4.3.3 Arquitetura do TPM 2.0

A especificação do TPM 2.0 organiza seus recursos em vários componentes principais, conforme descrito por (Arthur; Challener; Goldman, 2015):

*4.3.3.1 **Hierarquias e Entidades***

- ***Endorsement Hierarchy***: contém a chave de *endorsement*[6], usada para identificar o TPM.

[5] Um ataque de dicionário é um tipo de ataque de força bruta onde hackers utilizam uma lista predefinida de palavras, frases e combinações numéricas (o “dicionário”) para adivinhar senhas de contas online (Kaspersky,).

[6] Chave única gerada pelo fabricante e armazenada no TPM, usada para provar a autenticidade do módulo a terceiros.

- **Hierarquia de Armazenamento - *Storage Hierarchy***: gerencia chaves de armazenamento e dados persistentes.
- **Hierarquia de Plataforma**: controla PCRs associados ao *firmware* e BIOS (Sistema Básico de Entrada e Saída - *Basic Input/Output System*).

4.3.3.2 Objetos Criptográficos

- Chaves RSA/ECC, Índices NV (áreas de memória não volátil).
- Objetos protegidos por políticas de acesso.

4.3.3.3 Autorização e Sessões

- Múltiplas técnicas de autorização (senha, HMAC, políticas compostas).
- Sessões de criptografia e auditoria de comandos

4.3.3.4 Operações de Criptografia

- Geração de chave (TPM2_Create).
- Assinatura e verificação (TPM2_Sign, TPM2_VerifySignature).
- Codificação e Decodificação (*Sealing* e *Unsealing*) de dados sensíveis com PCRs.

4.3.4 TPM em sistemas de controle de conversores

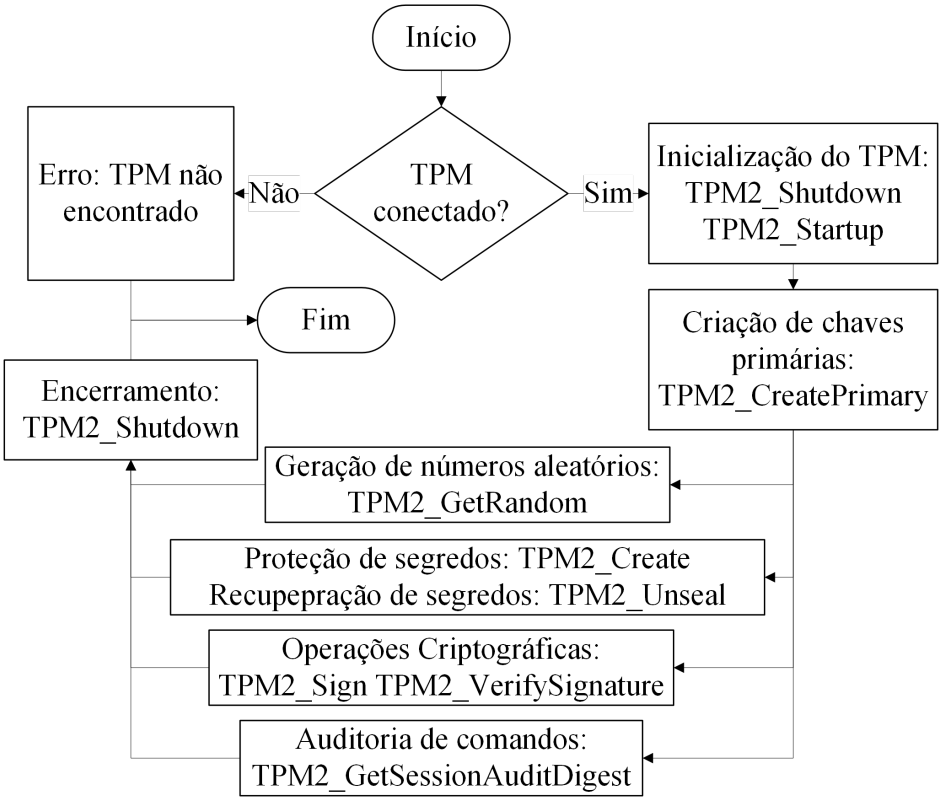
No contexto de conversores estáticos, o TPM pode ser usado para:

- Autenticar o *firmware* na inicialização, garantindo que apenas código assinado pelo fabricante seja executado.
- Proteger protocolos de comunicação entre HMI (Interface Homem Máquina - *Human Machine Interface*), PC de supervisão e o conversor, armazenando chaves para TLS.
- Registrar eventos críticos (atualizações de *firmware*, tentativas de acesso) em registros de auditoria dentro do TPM.
- Gerenciar políticas de acesso (por exemplo, RBAC) ao sistema de controle, evitando que comandos não autorizados sejam aceitos pelo conversor.

Com esses mecanismos, o TPM reduz a superfície de ataque de todo o sistema de comando e controle, elevando significativamente a confiabilidade operacional (Arthur; Challener; Goldman, 2015). A Figura 11 ilustra o fluxo de trabalho simplificado do TPM em qualquer sistema. É feita a checagem inicial de disponibilidade, a preparação e configuração do ambiente,

a geração e proteção de credenciais e dados criptográficos, e finalmente o encerramento ordenado de todas as operações.

Figura 11 – Fluxo de trabalho simplificado do TPM.



Fonte: do Autor.

Pesquisas anteriores investigaram a aplicação de *TrustZone*[7] (Arm Ltd., 2022) e HSMs[8] (módulos de segurança de hardware - *Hardware Security Modules*) certificados conforme o FIPS 140-3 (National Institute of Standards and Technology, 2019) para proteger sistemas embarcados, porém tais abordagens costumam aumentar o custo final ou dependem de *firmwares* proprietários. O uso do TPM, por sua vez, permite a adoção de bibliotecas abertas e pode ser integrado a microcontroladores convencionais via SPI, diferenciando-se dessas alternativas e oferecendo flexibilidade para conversores estáticos de baixo custo.

[7] Extensão de segurança da arquitetura Arm que cria dois ambientes de execução isolados: um seguro e outro não seguro.

[8] Dispositivos físicos dedicados à proteção de chaves criptográficas e à execução segura de operações criptográficas.

5 IMPLEMENTAÇÃO DE UM TPM EM UM MICROCONTROLADOR

Este capítulo apresenta a implantação do TPM em um microcontrolador e descreve o uso de um sistema operacional em tempo real, recomendado para a implementação do TPM.

5.1 SISTEMA OPERACIONAL EM TEMPO REAL (RTOS - *REAL-TIME OPERATING SYSTEM*)

Um Sistema Operacional em Tempo Real (RTOS) é um *software* projetado para atender a requisitos de tempo críticos, processando eventos e dados dentro de limites de tempo estritos (FreeRTOS.org, n.d.). Ao contrário de sistemas de tempo compartilhado (sistemas operacionais tradicionais como Windows, Linux e macOS), que visam alta taxa de processamento de dados e resposta interativa, um RTOS foca na previsibilidade de execução, garantindo que tarefas de alta prioridade sejam realizadas dentro de prazos determinísticos (FreeRTOS.org, n.d.).

A característica fundamental de um RTOS é a variação mínima no tempo de resposta a interrupções e na troca de contexto entre tarefas, elemento determinante para aplicações de controle industrial e de sistemas embarcados (FreeRTOS.org, n.d.). Dependendo do grau de rigor temporal, RTOSs podem ser classificados como *soft* (em que atrasos ocasionais são toleráveis) ou *hard* (em que toda interrupção perdida é considerada falha) (FreeRTOS.org, n.d.). Em sistemas embarcados, um RTOS deve oferecer latência de interrupção e troca de tarefas mínima, tipicamente na faixa de microssegundos, para responder efetivamente a sinais externos (FreeRTOS.org, n.d.).

A Figura 12 apresenta a arquitetura em camadas de um sistema embarcado executando o FreeRTOS. Na camada superior, as tarefas da aplicação (Tarefa 1 ... Tarefa n) implementam a lógica de aplicação e fazem uso das APIs[1] de criação, temporização e comunicação do RTOS. A seguir, o *kernel*[2] do FreeRTOS provê serviços fundamentais de escalonamento preemptivo[3] (*preemption*), sincronização (semáforos, filas), temporizadores de *software*, gerenciamento de memória e modos de economia de energia, garantindo resposta determinística a eventos externos. A camada de *driver* de dispositivo torna a programação do microcontrolador mais amigável, abstraindo detalhes de *hardware* (periféricos como SPI/I²C/GPIO, interrupções, DMA) e expõe uma API uniforme, permitindo que o *kernel* e as tarefas interajam com registros de periféricos de forma portátil. Na base, o *hardware* (microcontrolador, módulos periféricos e barramentos) executa as operações de entrada e saída reais, sendo acessado indiretamente pelo *kernel* via *drivers* ou diretamente pelas tarefas quando necessário. Essa separação em camadas assegura modularidade, portabilidade e previsibilidade temporal, facilitando a migração entre diferentes arquiteturas de *hardware* sem alterações na lógica de aplicação.

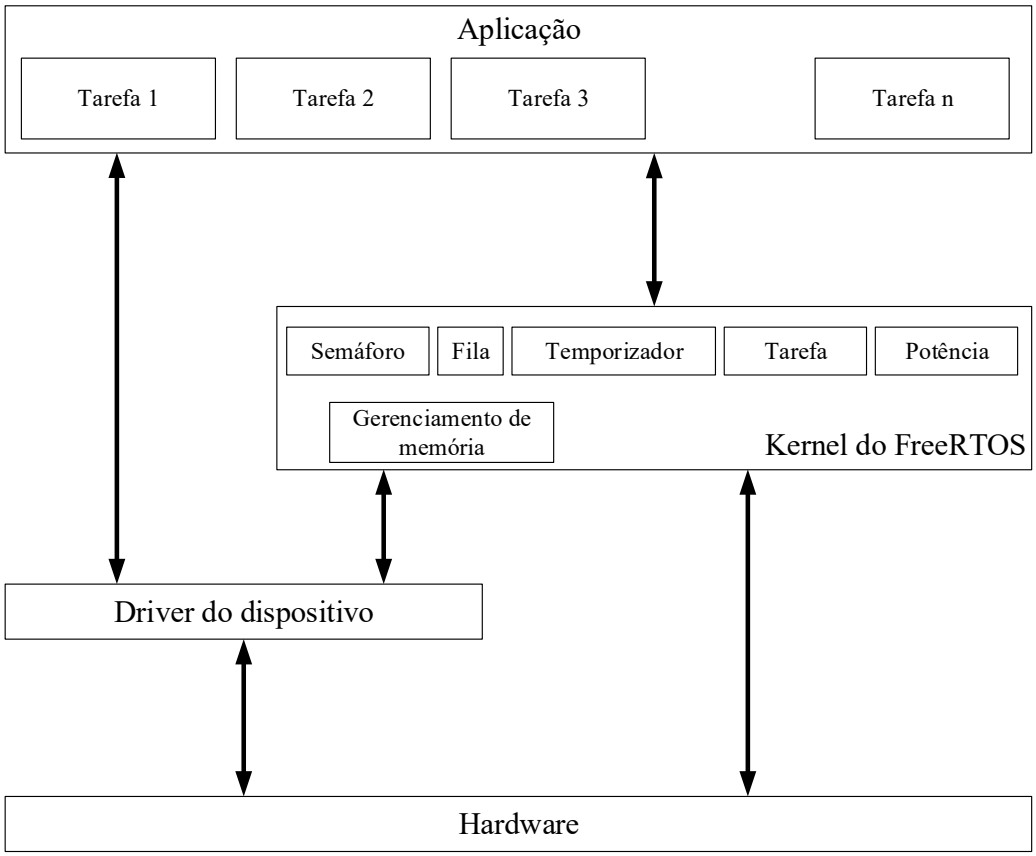
---

[1] Sigla para *Application Programming Interfaces*, conjuntos de rotinas e padrões que permitem a comunicação entre diferentes *softwares*.

[2] Núcleo do sistema operacional responsável pelo gerenciamento de recursos.

[3] Capacidade do RTOS de interromper a execução de uma tarefa para dar lugar a outra de maior prioridade.

Figura 12 – Estrutura típica de implementação de um RTOS (usando o FreeRTOS como exemplo).



Fonte: adaptado de (FastBit Embedded Brain Academy; Kiran Nayak, 2025).

5.1.1 FreeRTOS

Para microcontroladores com recursos limitados, o FreeRTOS destaca-se como um dos RTOSs mais populares e amplamente adotados. O FreeRTOS é um sistema operacional de código aberto distribuído, suportando mais de 40 arquiteturas de processadores, o que facilita sua portabilidade e integração em diversas placas de desenvolvimento (FreeRTOS.org, n.d.). Ele fornece funções essenciais de sistemas multitarefa, incluindo criação de tarefas, filas, ferramentas de sincronismo de exclusão mútua (*mutexes*[4]), semáforos, notificações diretas de tarefas e *buffers* de fluxo de mensagens (FreeRTOS.org, n.d.).

O FreeRTOS tem um tamanho binário típico entre 4 e 9 KB, tornando-o adequado para sistemas com memória restrita e permitindo tanto alocação estática quanto dinâmica para a memória de cada tarefa, conforme o projeto exige. Seu escalonamento padrão é preemptivo, priorizando imediatamente a tarefa de maior prioridade pronta para execução. O FreeRTOS

[4] Mecanismo de sincronização usado para evitar que múltiplas tarefas acessem simultaneamente um recurso compartilhado.

também suporta escalonamento cooperativo e tarefas sequenciais com período de execução configurável.

A documentação "*RTOS Fundamentals*" do FreeRTOS oferece um guia de introdução aos conceitos básicos e exemplos de código para várias famílias de microcontroladores, acelerando o aprendizado inicial. Em versões recentes, foram adicionados recursos como Multiprocessamento Simétrico (SMP *Symmetric Multiprocessing*), protocolo TCP/IP com suporte a IPv6 e integração nativa com serviços em nuvem para aplicações IoT (*Internet* das Coisas - *Internet of Things*). Graças à sua ampla comunidade, manutenção ativa e documentação abrangente, o FreeRTOS é uma excelente opção para poder usar o TPM com suas APIs.

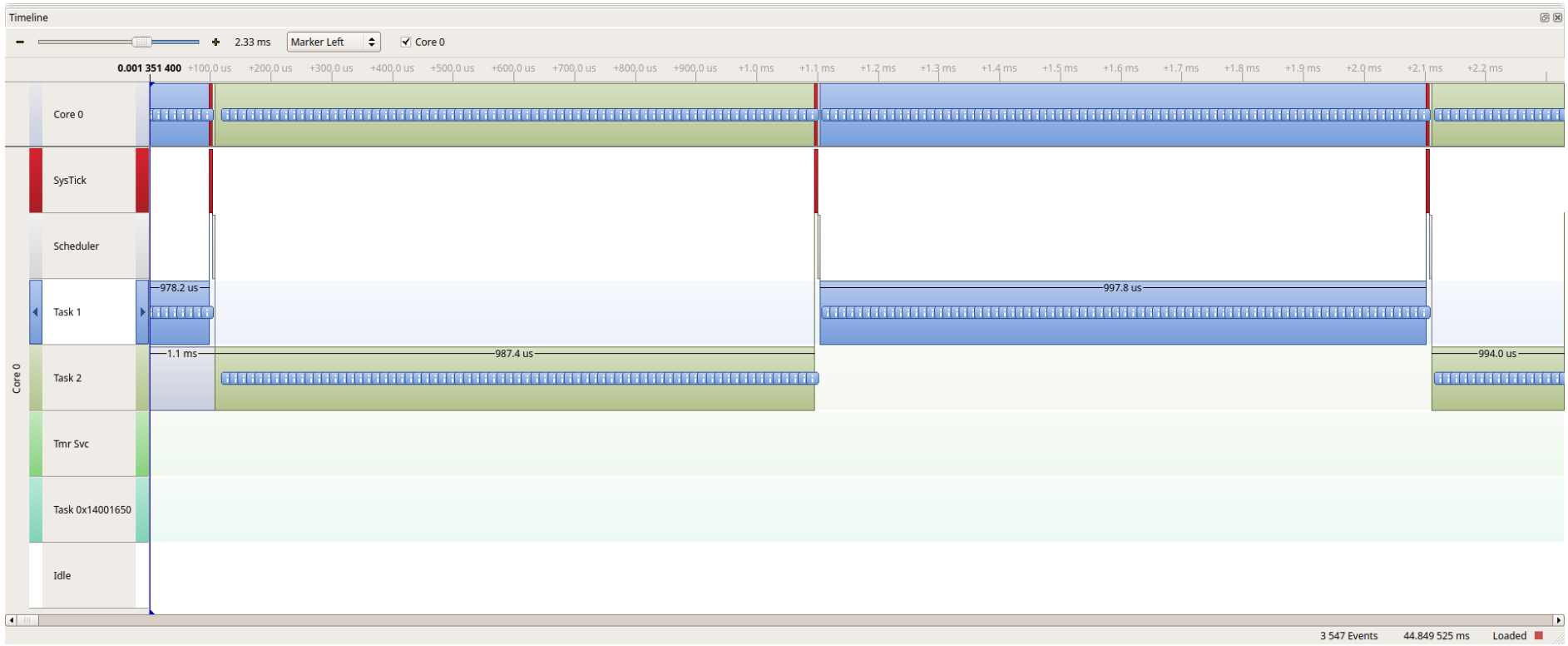
5.1.2 Compreensão da execução de tarefas no FreeRTOS

A execução de tarefas no FreeRTOS é, por padrão, baseada em um modelo de escalonamento preemptivo, onde o *kernel* decide qual tarefa deve ser executada com base em suas prioridades. As tarefas são executadas em *loop*, e o *kernel* garante que a tarefa de maior prioridade seja executada primeiro. Quando uma tarefa de maior prioridade se torna pronta para execução, o *kernel* interrompe a tarefa atual e troca o contexto para a nova tarefa.

A Figura 13 mostra a execução de duas tarefas em loop no modo preemptivo, onde o *kernel* alterna entre as tarefas com base em suas prioridades. A troca de contexto ocorre a cada 200 ms, conforme configurado no *FreeRTOSConfig.h*. Essa abordagem permite que o FreeRTOS responda rapidamente a eventos críticos, garantindo que as tarefas de alta prioridade sejam executadas em tempo hábil.

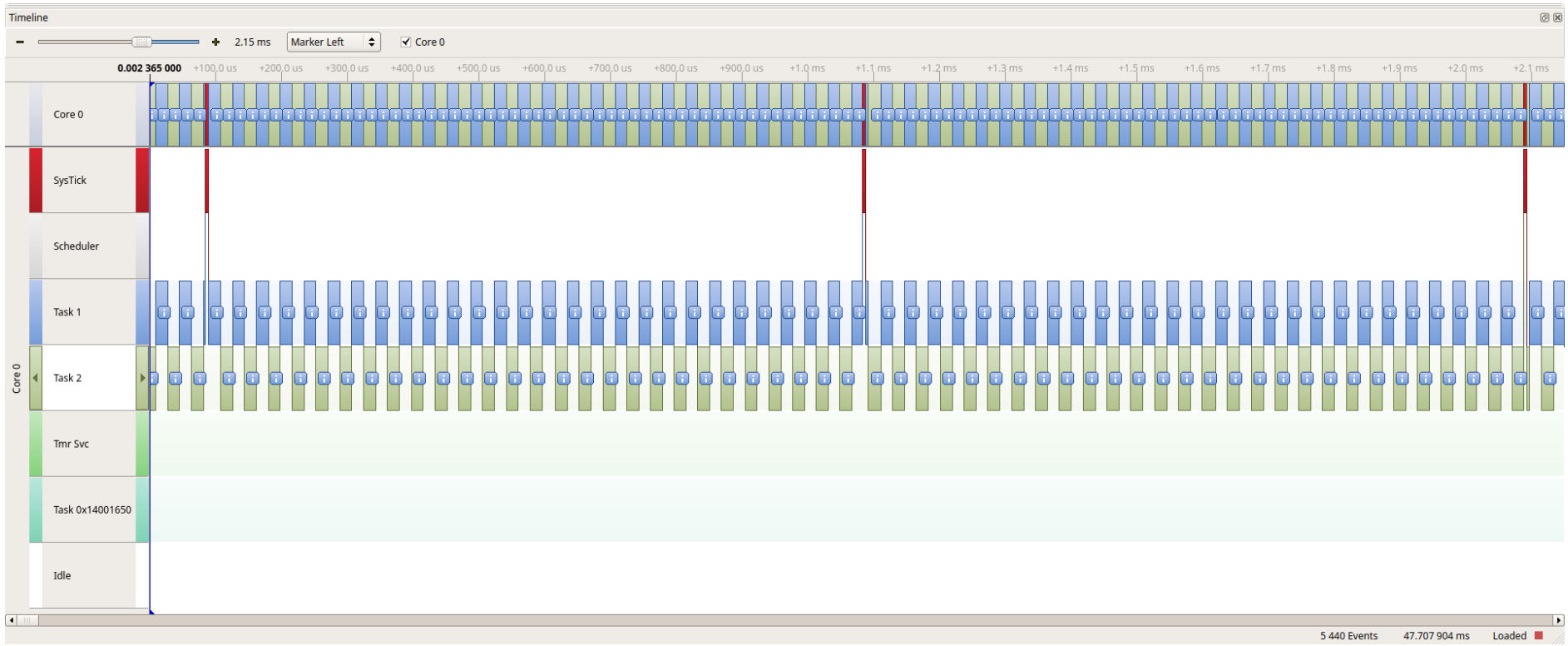
Já a Figura 14 ilustra a execução de duas tarefas em loop no modo cooperativo, onde as tarefas devem ceder explicitamente o controle ao *kernel* para que outra tarefa possa ser executada. Nesse modo, a troca de contexto ocorre apenas quando uma tarefa chama uma função de bloqueio e cede seu tempo de CPU para outra tarefa. Essa abordagem é menos eficiente em termos de resposta a eventos críticos, mas pode ser útil em cenários onde as tarefas não precisam ser interrompidas com frequência.

Figura 13 – Depuração da execução de duas tarefas em loop no modo preemptivo no FreeRTOS usando o SEGGER SystemView.



Fonte: do Autor.

Figura 14 – Depuração da execução de duas tarefas em loop no modo cooperativo no FreeRTOS usando o SEGGER SystemView.



Fonte: do Autor.

5.1.3 Configuração inicial e integração do TPM

Antes de iniciar o fluxo de trabalho, é necessário realizar os seguintes passos para integração e configuração do TPM:

1. **Habilitação no Firmware:** Ativar o TPM, garantindo que o módulo esteja desbloqueado e em modo operacional correto (Arthur; Challener; Goldman, 2015).

2. **Conexão Física:** Verificar a interface de comunicação (I²C ou SPI) e assegurar a soldagem adequada do chip TPM à placa-mãe ou ao conector designado (Arthur; Challener; Goldman, 2015).

3. **Instalação de Drivers e Bibliotecas:** Implantar o Trusted Software Stack (TSS2) ou equivalente no sistema operacional, disponibilizando a API de alto nível para os comandos TPM (Arthur; Challener; Goldman, 2015).

4. **Verificação de Capacidades:** Executar comandos básicos, como TPM2_GetCapability, para confirmar a presença do módulo e identificar a versão e os algoritmos suportados (Arthur; Challener; Goldman, 2015).

5. **Ações Iniciais:** Gerar as chaves principais (*Endorsement Key* e *Storage Root Key*) e estabelecer políticas de hierarquia (Arthur; Challener; Goldman, 2015).

5.1.4 Fluxo de trabalho típico do TPM

O fluxo de trabalho de um TPM embarcado em um sistema de controle de conversor estático segue, em linhas gerais, as etapas descritas a seguir (Arthur; Challener; Goldman, 2015):

*5.1.4.1 **Inicialização e provisionamento (Startup & Provisioning):***

Ao ligar o sistema, o TPM executa seu processo de inicialização, carregando hierarquias internas e restaurando registros de plataforma (PCRs) com base em valores armazenados na memória não volátil. Em seguida, são carregadas chaves de atestado e credenciais iniciais do fabricante, definindo a Raiz de Confiança local (Arthur; Challener; Goldman, 2015).

*5.1.4.2 **Medição de integridade (Secure Boot & Measurement):***

Antes de iniciar o *firmware* do conversor, cada componente de *software* (*bootloader*[5], *kernel*, aplicativo de controle) é medido (*hash* calculado) e os valores são armazenados nos PCRs correspondentes. Essa cadeia de medições — o chamado *secure boot* — garante que apenas código de controle autenticado seja executado, prevenindo injeção de *firmware* malicioso (Arthur; Challener; Goldman, 2015).

[5] Programa inicial executado após o acionamento do sistema, responsável por carregar e iniciar o *firmware* principal do dispositivo.

*5.1.4.3 **Provisão de chaves criptográficas (Key Provisioning):***

Para operações criptográficas no controle em tempo real (assinaturas digitais de comandos, estabelecimento de canais TLS, criptografia de parâmetros), o TPM gera internamente chaves simétricas e assimétricas, ou insere chaves provisionadas externamente em um armazenamento não volátil. As chaves podem ser associadas a políticas de autorização (por exemplo, uso somente se os PCRs estiverem em estados aprovados) (Arthur; Challener; Goldman, 2015).

*5.1.4.4 **Sessões e autorizações (Sessions & Authorization):***

Quando o sistema de supervisão (IHM/SCADA) solicita operações sensíveis, é estabelecida uma sessão autorizada com o TPM. Utiliza-se o mecanismo de Autorização Avançada para avaliar políticas compostas (verificação de PCRs, uso de PIN/padrão, limites de tempo), assegurando que comandos críticos sejam executados apenas sob condições devidamente autenticadas (Arthur; Challener; Goldman, 2015).

*5.1.4.5 **Atestado e verificação remota (Attestation):***

Para validação externa da integridade do conversor, o TPM gera uma citação (*quote*) dos PCRs assinada com uma chave de atestado. O sistema de gerenciamento remoto verifica a assinatura e os valores dos PCRs contra referências conhecidas, confirmando que o *firmware* embarcado não foi alterado (Arthur; Challener; Goldman, 2015).

*5.1.4.6 **Operação contínua e atualizações (Runtime & Updates):***

Durante a operação, o TPM é capaz de gerar números verdadeiramente aleatórios, criptografia, descriptografia e assinatura para rotinas de controle. Para atualizações de *firmware*, o TPM verifica assinaturas antes de aplicar novos binários, registra as operações em logs[6] auditáveis e, se necessário, atualiza chaves ou redefine PCRs conforme novas políticas (Arthur; Challener; Goldman, 2015).

Este fluxo garante a construção de um subsistema de segurança robusto, no qual a integridade do *firmware*, a confidencialidade das chaves e a autenticidade dos comandos de controle de inversores estáticos são mantidas durante todo o ciclo de vida do dispositivo.

5.2 PROGRAMAÇÃO DO MICROCONTROLADOR

Esta seção descreve de forma geral a organização do projeto para o microcontrolador, dês das primeiras funções de testes até a implementação do sistema operacional em tempo real FreeRTOS. Também são apresentados os passos tomados até o momento de integrar as bibliotecas wolfSSL ((wolfSSL Inc., 2025a)) e wolfTPM ((wolfSSL Inc., 2025b)). A opção

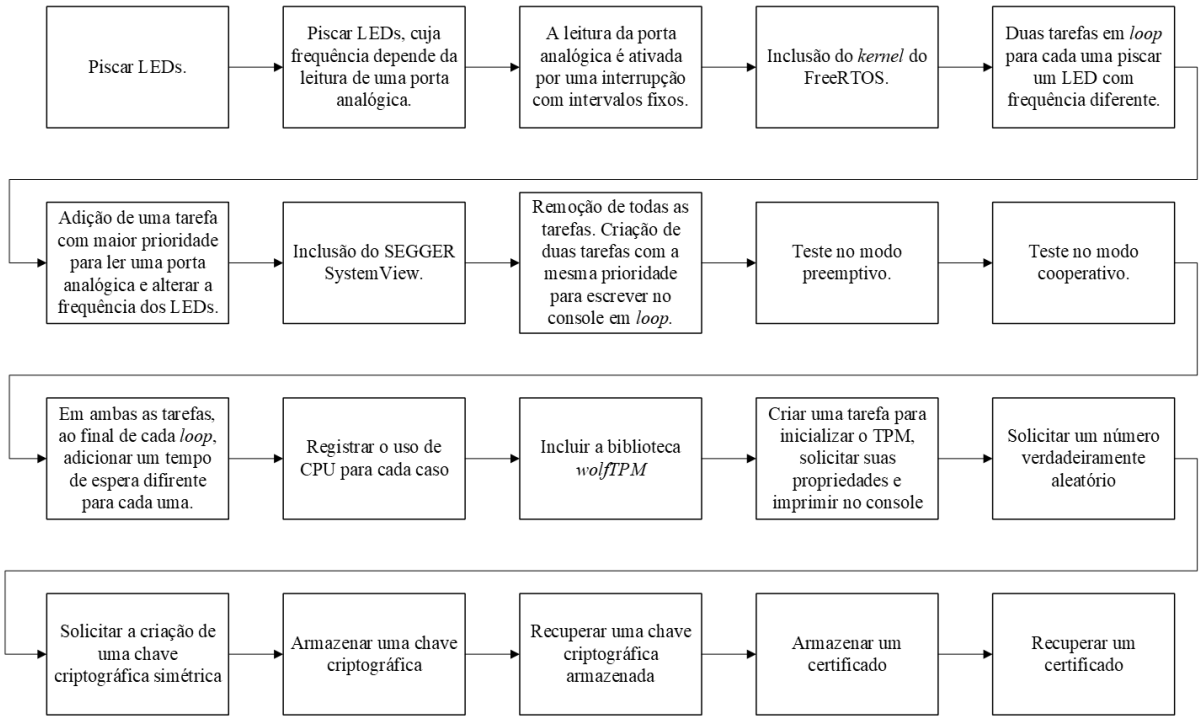
[6] Registros automáticos de eventos ou atividades de um sistema, usados para auditoria, depuração e análise de segurança.

metodológica preferenciou a implementação no *hardware* real, pois se buscava avaliar o impacto direto do FreeRTOS sobre o tempo de inicialização e o uso de memória do sistema.

5.2.1 Etapas da implementação realizadas

A organização da programação do microcontrolador foi organizada em diversas etapas menores, dês de testes iniciais até a integração completa do TPM. Essas etapas menores foram agrupadas em um roteiro de implementação, conforme ilustrado na Figura 15. O fluxo de trabalho conta com as seguintes etapas:

Figura 15 – Roteiro desejado de implementação do *firmware*.



Fonte: do Autor.

5.2.1.1 Testes dos GPIOs

Os GPIOs (General Purpose Input/Output) foram testados para garantir a funcionalidade básica de entrada e saída digital. Os testes incluíram a ativação de pinos configurados como saída comutando LEDs já embarcados na placa de desenvolvimento. A biblioteca HAL do STM32 foi utilizada para facilitar a manipulação dos GPIOs e não precisar manipular diretamente os registradores dos periféricos.

5.2.1.2 Inclusão de um periférico de ADC

Um periférico de ADC (Conversor Analógico-Digital - *Analog-to-Digital Converter*) foi adicionado para permitir a leitura do valor da tensão de uma porta analógica. O ADC foi

configurado para realizar leituras periódicas, usando o valor das leituras como parâmetro para variar a frequência de comutação dos LEDs, indicando visualmente que a leitura estava sendo feita.

5.2.1.3 Inclusão de um estouro de timer para iniciar a conversão do ADC

Um *timer* foi adicionado para iniciar a conversão do ADC em intervalos regulares. O *timer* foi configurado para gerar uma interrupção[7] a cada 100 ms, acionando a leitura do ADC e atualizando a frequência de comutação dos LEDs com base no valor lido. A atualização da frequência de comutação é feita com a interrupção disparada no fim da conversão do ADC.

5.2.1.4 Adição do kernel do FreeRTOS

A primeira etapa da implementação do FreeRTOS foi a adição do *kernel* ao projeto. Inicialmente, apenas o código fonte do FreeRTOS foi adicionado ao projeto, sem a configuração de tarefas ou filas. Correções foram feitas no *FreeRTOSConfig.h* para ajustar as configurações do *kernel*, adequando ao uso correto do microcontrolador. Algumas dessas configurações incluem a frequência do *tick*[8] do *kernel*, o tamanho mínimo da pilha de tarefas e a frequência do microcontrolador (FreeRTOS.org, 2025).

5.2.1.5 Criação de duas tarefas com a mesma prioridade

Duas tarefas foram criadas com a mesma prioridade, cada uma responsável por comutar um LED diferente. As tarefas foram configuradas para executar em loop, alternando o estado dos LEDs em uma frequência fixa. Essa etapa foi importante para validar a funcionalidade básica do FreeRTOS e garantir que as tarefas estavam sendo executadas corretamente.

5.2.1.6 Adição de uma tarefa com prioridade maior

Uma terceira tarefa foi adicionada com prioridade maior que as duas anteriores. Essa tarefa foi configurada para executar uma ação crítica, como ler o valor do ADC e atualizar a frequência de comutação dos LEDs. A prioridade mais alta garante que essa tarefa seja executada antes das outras, permitindo que a leitura do ADC ocorra em tempo hábil. A leitura do ADC é feita a cada 100 ms, e o valor lido é usado para ajustar a frequência de comutação dos LEDs, entretanto não foi preciso configurar *timers* para iniciar a conversão ou então utilizar interrupções, pois o FreeRTOS é capaz de lidar com a execução de tarefas periódicas.

---

[7] Sinal que interrompe o fluxo normal de execução de um processador para atender a eventos externos ou internos com prioridade.

[8] Pulso periódico gerado por um temporizador que define a unidade de tempo usada pelo RTOS para escalonamento e temporização de tarefas.

5.2.1.7 Inclusão do SEGGER SystemView

O SEGGER SystemView foi integrado ao projeto para monitorar o desempenho do FreeRTOS e registrar eventos de execução das tarefas. Essa ferramenta permite visualizar o tempo gasto em cada tarefa, a latência de interrupções e outros parâmetros de desempenho, facilitando a análise do comportamento do sistema em tempo real. A implementação iniciou-se com a inclusão do código fonte do SEGGER SystemView no projeto e ajustes nas configurações do FreeRTOS até sua correta integração (configuração do tamanho da pilha, por exemplo e que uma função *printf* seja capturada pelo SystemView ao invés de ser enviada para a saída padrão). A configuração do SEGGER SystemView foi feita para registrar eventos de execução das tarefas e interrupções, permitindo uma análise detalhada do desempenho do sistema.

5.2.1.8 Criação de duas tarefas com a mesma prioridade

Duas tarefas foram criadas com a mesma prioridade, ambas escrevendo uma mensagem curta (o nome da tarefa) no console do *SystemView*. Essas tarefas executam em *loop*, alternando a escrita de mensagens no console. Com as tarefas executando em *loop* e o *SystemView* capturando os eventos, foi possível observar o comportamento do sistema em tempo real, incluindo a troca de contexto entre as tarefas e a latência de interrupções.

5.2.1.9 Inclusão das bibliotecas wolfSSL e wolfTPM

As bibliotecas foram copiadas para o projeto, mas a compilação completa não ocorreu devido a dependências específicas do *hardware* e configurações de memória necessárias. A configuração incorreta desses ajustes resultou em erros de compilação, e a integração foi malsucedida. Por causa disso, também não foi possível realizar os testes planejados e comprovar as vantagens teóricas do TPM. Os itens descritos a seguir detalham como pretendia-se integrar o TPM caso as bibliotecas tivessem sido incluídas com sucesso.

5.2.1.10 Criação de uma tarefa para inicializar o TPM

Uma tarefa seria criada para inicializar o TPM, configurando-o e realizando as operações iniciais necessárias. Essa tarefa seria responsável por garantir que o TPM estivesse pronto para uso antes de qualquer outra operação crítica. Uma forma de verificar se o TPM foi inicializado corretamente seria verificar se a versão do TPM e os algoritmos suportados estão de acordo com as especificações esperadas.

5.2.1.11 Solicitar um número aleatório do TPM

Uma vez que o TPM estivesse inicializado, uma tarefa seria criada para solicitar um número aleatório do TPM. Essa tarefa utilizaria a API do TPM para gerar um número verdadeiramente aleatório, que poderia ser usado em operações criptográficas subsequentes. A

geração de números aleatórios é uma função crítica do TPM, garantindo a segurança das chaves criptográficas e outros dados sensíveis.

5.2.1.12 Criação de uma chave criptográfica simétrica

Uma vez que o TPM estivesse inicializado e um número aleatório tivesse sido gerado, uma tarefa seria criada para gerar uma chave criptográfica simétrica. Essa tarefa utilizaria as APIs do TPM para criar chaves AES. A criação de chaves criptográficas é uma etapa fundamental para garantir a confidencialidade e a integridade dos dados em sistemas embarcados.

5.2.1.13 Armazenar uma chave criptográfica no TPM

Após a criação da chave criptográfica simétrica, armazenar essa chave no TPM. O armazenamento seguro de chaves é uma das principais funções do TPM, garantindo que as chaves não sejam acessíveis diretamente pelo *firmware* ou por outras partes do sistema. Essa etapa é crucial para proteger as chaves contra acesso não autorizado e garantir a segurança do sistema

5.2.1.14 Recuperar uma chave criptográfica do TPM

Uma tarefa seria criada para recuperar a chave criptográfica simétrica armazenada no TPM. Essa tarefa utilizaria as APIs do TPM para acessar a chave e realizar operações criptográficas, como criptografia e descriptografia de dados. A recuperação de chaves é uma função essencial do TPM, permitindo que o *firmware* utilize as chaves armazenadas para operações seguras.

5.2.1.15 Fazer o armazenamento seguro de um certificado digital no TPM

Um certificado digital seria armazenado no TPM, garantindo que ele esteja protegido contra acesso não autorizado. O armazenamento seguro de certificados digitais é importante para garantir a autenticidade e a integridade das comunicações em sistemas embarcados.

5.2.1.16 Recuperar um certificado digital do TPM

O certificado digital seria recuperado do TPM e utilizado para realizar operações de autenticação e verificação de assinaturas digitais. A recuperação de certificados digitais é uma função crítica do TPM, permitindo que o *firmware* valide a autenticidade de comunicações e dados.

5.2.2 Integração com wolfSSL e wolfTPM

Houve uma tentativa de adicionar as bibliotecas wolfSSL e wolfTPM ao projeto. Essas bibliotecas são essenciais para a implementação de criptografia e integração do TPM no micro-

controlador. A documentação do wolfTPM sugere que as bibliotecas sejam adicionadas como pacotes no STM32CubeIDE, o que deveria facilitar a integração, entretanto, a complexidade de configuração e as dependências específicas do *hardware* tornaram essa tarefa desafiadora.

Embora as bibliotecas tenham sido copiadas para o projeto, a compilação completa não chegou a ocorrer. Diversas dependências do wolfTPM requerem configurações de *hardware* específicas, como a interface SPI com o TPM, e o wolfSSL exige ajustes adicionais no `FreeRTOSConfig.h` e no *linker script* para suportar o aumento de uso de memória. Algumas funções ainda exigem a inclusão de uma terceira biblioteca, a wolfCrypt (integrada na wolfSSL recentemente, ainda é uma biblioteca separada na interface de configuração do microcontrolador). A configuração incorreta desses ajustes resultou em erros de compilação, e a integração foi mal-sucedida.

5.2.3 Limitações

A integração das bibliotecas wolfSSL e wolfTPM não foi finalizada, resultando em uma compilação incompleta do projeto. Essa falha impossibilitou a execução dos testes planejados e comprometeu a comprovação prática das vantagens teóricas do TPM.

Tampouco foram obtidas métricas de desempenho, como consumo de memória, uso de CPU ou latência de inicialização. O SEGGER SystemView (SEGGER Microcontroller GmbH, 2025) foi instalado com esse objetivo, mas sua utilização não ocorreu devido à ausência de um protótipo funcional.

A metodologia prevista carece de procedimentos padronizados. Pretendia-se quantificar o uso de CPU em momentos de alta demanda de comunicação e verificar se a sobrecarga criptográfica interferiria na tarefa de controle do conversor, executada em intervalos fixos e com prioridade máxima. Como a integração das bibliotecas não foi concluída, esses ensaios permanecem em aberto.

5.2.4 Perspectivas e possíveis melhorias

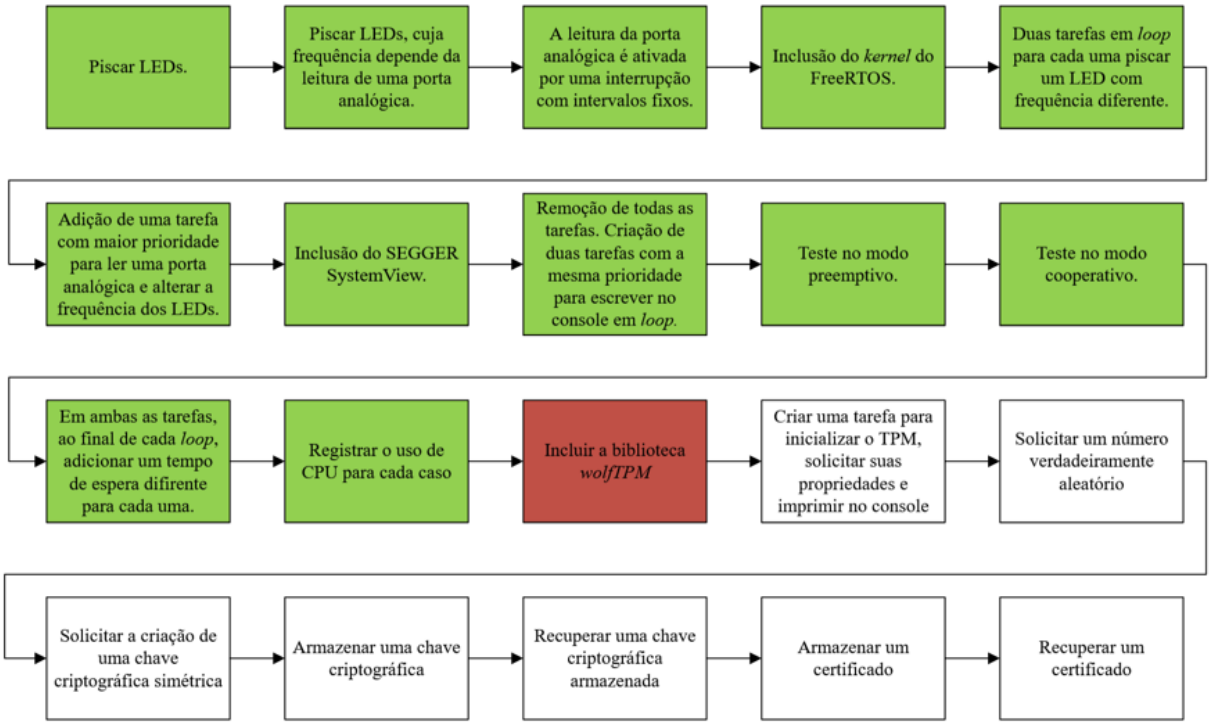
Com uma visão ampliada do projeto, algumas melhorias podem ser implementadas para facilitar a integração de bibliotecas complexas como wolfSSL e wolfTPM:

- Incluir e configurar as bibliotecas wolfSSL e wolfTPM diretamente no STM32CubeIDE, utilizando o gerenciador de pacotes do IDE para simplificar a integração e reduzir a complexidade de configuração manual.
- Compilar e executar os exemplos de código fornecidos pelo wolfTPM e wolfSSL para validar a configuração do microcontrolador e garantir que as bibliotecas estejam funcionando corretamente antes de integrá-las ao projeto principal.

- Como sequência desse trabalho, espera-se realizar testes práticos com o TPM, coletando métricas de desempenho e validando a eficácia do TPM no contexto de controle de conversores estáticos.

O roteiro de implementação executado, conforme ilustrado na Figura 16, traz uma visão sobre as perspectivas futuras do projeto. A implementação do TPM em um microcontrolador com FreeRTOS e as bibliotecas wolfSSL e wolfTPM é um passo importante para garantir a segurança e a integridade dos sistemas de controle de conversores estáticos.

Figura 16 – Roteiro executado de implementação do *firmware*.



Fonte: do Autor.

Como sequência desse trabalho, espera-se:

- Inicialização correta do TPM e extração das propriedades do dispositivo, como versão e algoritmos suportados.
- Geração de um número verdadeiramente aleatório (TRNG - *True Random Number Generator*) usando o TPM.
- Criação de chave criptográfica simétrica e assimétrica, com armazenamento seguro no TPM.
- Assinatura digital de mensagens e verificação de assinaturas usando o TPM.
- Criação de um ambiente de teste para validação das funcionalidades do TPM.

As limitações impactaram diretamente nas conclusões obtidas, pois a ausência de testes práticos impede comprovar a eficácia do TPM no contexto proposto. A coleta de métricas de desempenho é essencial para validar a viabilidade do uso em aplicações industriais.

6 CONCLUSÃO

A crescente digitalização dos sistemas industriais tornou obrigatório a adoção de mecanismos robustos de segurança cibernética em dispositivos críticos, como os conversores estáticos. Partindo do problema central - "Como aumentar a segurança de conversores estáticos empregando o TPM?- este trabalho estabeleceu quatro objetivos principais: revisar a literatura sobre segurança cibernética e TPM, identificar vulnerabilidades em conversores estáticos, estudar em profundidade o funcionamento do TPM 2.0 e implementar sua integração em um microcontrolador STM32 rodando FreeRTOS. Embora as duas primeiras etapas tenham sido plenamente cumpridas, oferecendo um panorama detalhado das ameaças e dos recursos do módulo de plataforma confiável, a implementação prática esbarrou em incompatibilidades de *hardware* e configurações de memória ao integrar wolfSSL e wolfTPM, impedindo a obtenção de um protótipo funcional.

Ao longo da pesquisa, ficou claro que o TPM exerce papel crucial como raiz de confiança, fornecendo atestado de *firmware*, gerenciamento seguro de chaves e mecanismos de detecção de adulterações. A análise das vulnerabilidades dos conversores estáticos, complementada pelo estudo de incidentes reais de ciberataques, permitiu a criação de um roteiro teórico de implementação do TPM em sistemas embarcados. Como contribuição original, este trabalho articula as exigências de configuração de *drivers*, *linker scripts* e alocação de memória necessárias para o provisionamento seguro do módulo, ao mesmo tempo em que destaca as principais dificuldades práticas enfrentadas. Apesar de não ter sido possível validar experimentalmente os benefícios de desempenho e segurança, a pesquisa oferece subsídios teóricos e metodológicos para futuras integrações em ambientes industriais.

Para trabalhos subsequentes, recomenda-se priorizar a padronização do processo de integração de bibliotecas criptográficas em ambientes STM32CubeIDE, explorando pacotes nativos e exemplos oficiais, antes de migrar para o projeto principal. Deve-se também conduzir testes controlados de desempenho - incluindo métricas de uso de CPU, consumo de memória e latência de inicialização segura - e expandir o escopo para geração de TRNG, provisão de chaves simétricas e assimétricas, assinatura digital e atestado remoto.

REFERÊNCIAS

Arm Ltd. **Armv8-M Security Features**. Arm Developer, 2022. Disponível em: https://developer.arm.com/documentation/101810/0100. Citado na página 43.

ARTHUR, Will; CHALLENER, David; GOLDMAN, Kenneth. **A Practical Guide to TPM 2.0: Using the New Trusted Platform Module in the New Age of Security**. [S.l.]: Apress, 2015. Citado 7 vezes nas páginas 31, 33, 40, 41, 42, 49 e 50.

BALDA, Juan Carlos et al. Cybersecurity and power electronics: Addressing the security vulnerabilities of the internet of things. **IEEE Power Electronics Magazine**, v. 4, p. 37–43, 12 2017. Citado 3 vezes nas páginas 17, 28 e 31.

BARBI, Ivo. **Eletrônica de Potência**. 6ª. ed. [S.l.]: Edição do Autor, 2006. 328 p. Citado na página 28.

BRANQUINHO, Marcelo; BRANQUINHO, Thiago. **Segurança Cibernética Industrial**. [S.l.]: Alta Books, 2021. Citado 9 vezes nas páginas 16, 17, 18, 19, 21, 22, 23, 26 e 28.

CISA. **CrashOverride Malware**. 2017. Acessado em: 09 de novembro de 2023. Disponível em: https://www.cisa.gov/news-events/alerts/2017/06/12/crashoverride-malware. Citado na página 24.

CISA. **ICS Focused Malware | CISA**. 2021. Disponível em: https://www.cisa.gov/news-events/ics-advisories/icsa-14-178-01. Citado na página 23.

CISA. **People's Republic of China State-Sponsored Cyber Actor Living off the Land to Evade Detection**. 2024. Disponível em: https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-038a. Citado na página 26.

CISA. **Ransomware 101**. 2025. Acesso em: 25 de julho de 2025. Disponível em: https://www.cisa.gov/stopransomware/ransomware-101. Citado na página 20.

CLOUDFLARE, Inc. **What is defense in depth? | Layered security**. 2025. Acesso em: 15 de julho de 2025. Disponível em: https://www.cloudflare.com/learning/security/glossary/what-is-defense-in-depth/. Citado 2 vezes nas páginas 20 e 21.

COMMUNICATIONS, AXIS. **Gerenciamento do ciclo de vida do dispositivo**. 2023. Acesso em: 25 de julho de 2025. Disponível em: https://www.axis.com/dam/public/2a/9d/8a/flyer--cybersecurity--device-lifecycle-management-pt-BR-405095.pdf. Citado na página 33.

DARANDE, Gonjeshke. **Cyberattacks against Iran's steel industry**. 2022. Disponível em: https://www.youtube.com/watch?v=fnbCr8mgoT8. Citado na página 25.

DEFENSE in depth (computing). 2025. Acesso em: 15 de julho de 2025. Disponível em: https://en.wikipedia.org/wiki/Defense_in_depth_(computing). Citado na página 21.

DRAGOS. **Protect Against FrostyGoop ICS Malware Targeting Operational Technology**. 2024. Disponível em: https://www.dragos.com/blog/protect-against-frostygoop-ics-malware-targeting-operational-technology/. Citado na página 26.

Dragos, Inc. **2025 OT/ICS Cybersecurity Report: A Year in Review**. Hanover, MD, USA, 2025. Disponível em: https://www.dragos.com/resources/reports/2025-ot-ics-cybersecurity-report-a-year-in-review/. Citado 2 vezes nas páginas 21 e 26.

ENGINEERING, Control. **Throwback Attack: An Insider Releases 265,000 Gallons of Sewage on the Maroochy Shire**. 2021. Acesso em: 25 de julho de 2025. Disponível em: https://www.controleng.com/throwback-attack-an-insider-releases-265000-gallons-of-sewage-on-the-maroochy-shire/. Citado na página 22.

FastBit Embedded Brain Academy; Kiran Nayak. **Mastering RTOS: Hands-on FreeRTOS and STM32Fx with Debugging**. 2025. Udemy online course. Última atualização: junho de 2025. Disponível em: https://www.udemy.com/course/mastering-rtos-hands-on-with-freertos-arduino-and-stm32fx/. Citado na página 45.

FERRICK, Stephen. **DJI Mavic Pro SD Card Data Recovery After Accidental Formatting**. 2024. https://www.pitsdatarecovery.com/blog/dji-mavic-pro-sd-card-data-recovery-after-formatting/. Acesso em: 19 de junho de 2025. Citado na página 32.

FREERTOS.ORG. **Customization**. 2025. Acesso em: 16 de julho de 2025. Disponível em: https://www.freertos.org/Documentation/02-Kernel/03-Supported-devices/02-Customization. Citado na página 52.

FreeRTOS.org. **RTOS Fundamentals**. n.d. https://freertos.org/Documentation/01-FreeRTOS-quick-start/01-Beginners-guide/01-RTOS-fundamentals. Acesso em: 09 de junho de 2025. Citado 2 vezes nas páginas 44 e 45.

GOODIN, Dan. **US natural gas operator shuts down for 2 days after being infected by ransomware**. 2020. Disponível em: https://arstechnica.com/information-technology/2020/02/ransomware-infection-shuts-down-us-natural-gas-operator-for-2-days/. Citado na página 24.

INVESTIGATION, Federal Bureau of. **Ransomware**. 2025. Acesso em: 25 de julho de 2025. Disponível em: https://www.fbi.gov/how-we-can-help-you/scams-and-safety/common-frauds-and-scams/ransomware. Citado na página 20.

Kaspersky. **What is a Dictionary Attack?** Acesso em: 14 de julho de 2025. Disponível em: https://www.kaspersky.com/resource-center/definitions/what-is-a-dictionary-attack. Citado na página 41.

MARTINS, Tiago. **Estudo e aplicação de segurança cibernética para conversores de frequência conectados à IIoT**. 154 p. Dissertação (Mestrado) — Universidade do Estado de Santa Catarina (UDESC), 05 2021. Disponível em: http://sistemabu.udesc.br/pergamumweb/vinculos/0000a3/0000a3ac.pdf. Citado 3 vezes nas páginas 16, 17 e 18.

MARTINS, Tiago; OLIVEIRA, Sergio Vidal Garcia. Cybersecurity in the power electronics. **IEEE Latin America Transactions**, v. 17, p. 1300–1308, 08 2019. Citado 6 vezes nas páginas 19, 20, 22, 23, 24 e 28.

MARTINS, Tiago; OLIVEIRA, Sergio Vidal Garcia. Enhanced modbus/tcp security protocol: Authentication and authorization functions supported. **Sensors**, v. 22, p. 8024, 10 2022. Citado 3 vezes nas páginas 30, 31 e 32.

MICHAEL, James Bret et al. Can you trust zero trust? **IEEE Computer**, v. 55, n. 8, p. 103–105, Aug 2022. Disponível em: https://doi.org/10.1109/MC.2022.3178813. Citado na página 33.

MOORE MICHELLE, PhD. **Black, Gray and White-Hat Hackers: What's the Difference?** 2025. https://onlinedegrees.sandiego.edu/black-vs-gray-vs-white-hat-hackers/. Acesso em: 12 de julho de 2025. Citado na página 18.

MUNDIAL, Fórum Econômico. **Global risks report 2023**. Ginebra: Foro Económico Mundial, 2023. Citado 2 vezes nas páginas 16 e 27.

NATHER, Nicole. **Critical infrastructure malware hits Ukraine**. 2024. Disponível em: https://www.axios.com/2024/07/23/critical-infrastructure-malware-dragos-ukraine. Citado na página 26.

National Institute of Standards and Technology. **Security Requirements for Cryptographic Modules**. 2019. FIPS 140-3. Disponível em: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf. Citado na página 43.

NEXTGOV. **US agencies assessed Chinese telecom hackers likely hit data center and residential internet providers**. 2025. Disponível em: https://www.nextgov.com/cybersecurity/2025/06/us-agencies-assessed-chinese-telecom-hackers-likely-hit-data-center-and-residential-internet-providers/405920/. Citado na página 26.

POREMBA, Sue. **Russian-Based Dragonfly Group Attacks Energy Industry**. 2014. Disponível em: https://www.itbusinessedge.com/networking/russian-based-dragonfly-group-attacks-energy-industry/. Citado na página 23.

RASTENIS, Justinas et al. E-mail-based phishing attack taxonomy. **Applied Sciences**, v. 10, n. 7, 2020. ISSN 2076-3417. Disponível em: https://www.mdpi.com/2076-3417/10/7/2363. Citado na página 20.

SEGGER Microcontroller GmbH. **SystemView User Guide**. SEGGER, 2025. Disponível em: https://www.segger.com/products/development-tools/systemview/. Citado na página 55.

STMicroelectronics. **NUCLEO-H753ZI: STM32 Nucleo-144 development board with STM32H753ZI MCU**. STMicroelectronics, 2025. Disponível em: https://www.st.com/en/evaluation-tools/nucleo-h753zi.html. Citado na página 38.

STMicroelectronics. **SC-KTPM-RASPIKG9: TPM evaluation board for ST33KTPM products**. STMicroelectronics, 2025. Disponível em: https://estore.st.com/en/sc-ktpm-raspikg9-cpn.html?srsltid=AfmBOoqwczuPq2SzlAid9lVoJ6uljizrMoUXcRf1fq0tSfgYbzdYfiJV. Citado na página 39.

TEAM82. **Unpacking the Blackjack Group's Fuxnet Malware**. 2024. Disponível em: https://claroty.com/team82/research/unpacking-the-blackjack-groups-fuxnet-malware. Citado na página 25.

THOMPSON, Iain. **Hacktivism is having a resurgence**. 2025. Disponível em: https://www.theregister.com/2025/04/13/hacktivism_is_having_a_resurgence/. Citado na página 25.

TIDY, Joe. Os hackers misteriosos que dizem ter provocado incêndio em fábrica no irã. **BBC News Brasil**, 06 2022. Disponível em: https://www.bbc.com/portuguese/internacional-62124923. Citado na página 25.

WIKIPEDIA. **2023 Kyivstar cyberattack**. 2023. Disponível em: https://en.wikipedia.org/wiki/2023_Kyivstar_cyberattack. Citado na página 25.

WIRED. **Hackers Remotely Kill a Jeep on a Highway | WIRED**. 2015. Disponível em: https://www.youtube.com/watch?v=MK0SrxBC1xs. Citado na página 24.

WIRED. **Ukraine's Kyivstar outage tied to Sandworm GRU hackers**. 2024. Disponível em: https://www.wired.com/story/ukraine-kyivstar-solntsepek-sandworm-gru/. Citado na página 25.

wolfSSL Inc. **wolfSSL Embedded SSL/TLS Library**. wolfSSL Inc., 2025. Disponível em: https://www.wolfssl.com/products/wolfssl/. Citado na página 50.

wolfSSL Inc. **wolfTPM: Portable TPM 2.0 Library**. wolfSSL Inc., 2025. Disponível em: https://www.wolfssl.com/products/wolftpm/. Citado na página 50.

World Economic Forum. **The Global Risks Report 2023**. Geneva, 2023. Disponível em: https://www.weforum.org/publications/global-risks-report-2023/. Citado na página 27.

World Economic Forum. **The Global Risks Report 2024**. Geneva, 2024. Disponível em: https://www.weforum.org/publications/global-risks-report-2024/. Citado na página 27.

World Economic Forum. **Global Risks Report 2025**. Geneva, 2025. Disponível em: https://www.weforum.org/publications/global-risks-report-2025/. Citado 2 vezes nas páginas 21 e 27.

GLOSSÁRIO

API: Interface de Programação de Aplicações.
Attestation: Processo de comprovação da integridade de um sistema.
Backdoor: Mecanismo oculto que permite acesso não autorizado a um sistema.
BIOS: Sistema Básico de Entrada/Saída responsável pela inicialização básica do hardware.
Bootloader: Código inicial responsável por preparar o hardware e carregar o sistema principal.
Buffer: Área de memória reservada para armazenar dados temporariamente.
Build: Processo de compilação e geração de um programa executável.
CA: Autoridades Certificadas, responsáveis por emitir certificados digitais.
CAN: Controller Area Network, protocolo de comunicação automotivo e industrial.
Checksum: Valor calculado para verificar a integridade de dados transmitidos ou armazenados.
CISA: Cybersecurity & Infrastructure Security Agency, agência de segurança cibernética dos EUA.
CPU: Central Processing Unit, unidade central de processamento do computador.
Deepfake: Conteúdo sintético gerado por inteligência artificial para simular pessoas reais.
DDoS: Distributed Denial-of-Service, ataque distribuído de negação de serviço.
DMA: Direct Memory Access, acesso direto à memória sem intervenção da CPU.
DoS: Denial-of-Service, ataque que sobrecarrega sistemas para torná-los inacessíveis.
DTCM: Data Tightly Coupled Memory, memória de acesso rápido integrada ao microcontrolador.
Driver: Código que possibilita a comunicação do sistema operacional com periféricos.
DSP: Digital Signal Processor, processador otimizado para processamento de sinais digitais.
ECC: Elliptic Curve Cryptography, criptografia baseada em curvas elípticas.
Endorsement Key: Chave primária do TPM utilizada para identificação e atestação.
FIPS 140-3: Norma de requisitos de segurança para módulos criptográficos.
Firmware: Conjunto de instruções gravado em memória não volátil que controla o hardware.
FreeRTOS: Sistema operacional em tempo real de código aberto.
Fuzzing: Técnica de testes que envia entradas inesperadas a um programa em busca de falhas.
Gateway: Dispositivo responsável por encaminhar a comunicação entre redes distintas.
GPIO: General Purpose Input/Output, pinos de entrada e saída gerais em microcontroladores.
HAL: Hardware Abstraction Layer, camada de abstração do hardware.
Hardware: Conjunto dos componentes físicos de um sistema computacional.
Hash: Função que converte dados em um valor fixo usado para verificar integridade.
Heap: Área de memória destinada à alocação dinâmica de dados.
HMAC: Hash-based Message Authentication Code, código de autenticação baseado em hash.
HSM: Hardware Security Module, módulo de segurança dedicado ao armazenamento de chaves.
ICS: Industrial Control System, sistema de controle industrial.
IDE: Integrated Development Environment, ambiente integrado de desenvolvimento.

IHM: Interface Homem-Máquina, interface gráfica para interação humana.
IIoT: Industrial Internet of Things, internet industrial das coisas.
JTAG: Joint Test Action Group, padrão para teste e depuração de placas eletrônicas.
Kernel: Núcleo do sistema operacional responsável pelo gerenciamento de recursos.
Linker script: Arquivo que define como os módulos compilados são organizados no executável.
Malware: Programa malicioso desenvolvido para prejudicar ou explorar um sistema.
Man-in-the-Middle: Ataque em que um invasor intercepta e possivelmente altera a comunicação entre duas partes.
Mutex: Estrutura de exclusão mútua que evita acessos simultâneos a um mesmo recurso.
NV (Non-Volatile): Memória não volátil, que retém dados sem energia.
OPC: OLE for Process Control, padrão de comunicação industrial.
PCR: Platform Configuration Register, registros para armazenar medições de integridade no TPM.
Phishing: Técnica que engana o usuário para obter informações sigilosas.
PID: Proportional-Integral-Derivative, controlador usado em sistemas de controle.
PLC: Programmable Logic Controller, controlador lógico programável.
Preemption: Capacidade de interromper a execução de uma tarefa para dar lugar a outra de maior prioridade.
PWM: Pulse Width Modulation, modulação por largura de pulso.
Ransomware: Malware que bloqueia dados ou sistemas, exigindo pagamento para liberação.
RBAC: Role Based Access Control, controle de acesso baseado em funções.
RISC: Reduced Instruction Set Computer, arquitetura simplificada de processadores.
RoT (Root of Trust): Conjunto de componentes críticos em hardware ou software confiáveis para iniciar um sistema seguro.
RSA: Rivest-Shamir-Adleman, algoritmo de criptografia de chave pública.
RTOS: Real-Time Operating System, sistema operacional de tempo real.
SCADA: Supervisory Control and Data Acquisition, sistema de controle e aquisição de dados para processos industriais.
Scheduler: Componente que define a ordem de execução das tarefas no sistema.
Sealing: Processo de cifrar dados ligados ao estado de uma plataforma.
Secure Boot: Processo que valida o software de inicialização para impedir códigos não autorizados.
Secure Digital: Tecnologia de cartão de memória portátil do tipo SD.
SEGGER SystemView: Ferramenta de depuração em tempo real para analisar sistemas operacionais de tempo real.
Setpoint: Valor de referência utilizado em sistemas de controle.
Software as a Service: Modelo de fornecimento de software pela internet mediante assinatura.
SPI: Serial Peripheral Interface, protocolo de comunicação serial síncrona utilizado em eletrônica embarcada.

SQL: Structured Query Language, linguagem de consulta estruturada para bancos de dados.
SSL: Secure Sockets Layer, protocolo de criptografia para comunicação segura em redes.
Stack: Região de memória que armazena chamadas de funções e variáveis locais.
STM32CubeIDE: Ambiente de desenvolvimento integrado para microcontroladores STM32.
Storage Root Key: Chave do TPM que protege a hierarquia de armazenamento seguro.
Symmetric Multiprocessing: Arquitetura em que vários processadores compartilham a mesma memória principal.
TCG: Trusted Computing Group, organização que padroniza tecnologias seguras e confiáveis.
TCP: Transmission Control Protocol, protocolo de transporte orientado à conexão.
TCP/IP: Conjunto de protocolos de comunicação da internet.
TLS: Transport Layer Security, sucessor do SSL para comunicação segura em redes.
TPM: Trusted Platform Module, chip dedicado a operações criptográficas e segurança de plataforma.
TrustZone: Extensão de segurança da Arm que cria domínios isolados no processador.
WEF: World Economic Forum, organização internacional que publica relatórios sobre riscos globais.
wolfCrypt: Biblioteca de algoritmos criptográficos do projeto wolfSSL.
wolfSSL: Biblioteca leve para criptografia e protocolos TLS.
wolfTPM: Biblioteca que fornece interface de software para o Trusted Platform Module (TPM).