



ENGENHARIA REVERSA EM EQUIPAMENTOS ELETRÔNICOS COM UART E STM32: PRÁTICAS DE ENSINO-APRENDIZAGEM NA EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

Davi de Siqueira Freitas

<http://lattes.cnpq.br/6952185987503358>

João Carlos Sena Rodrigues

<http://lattes.cnpq.br/0973741964791708>

John Percival Rodrigues Linhares

<http://lattes.cnpq.br/3102952581225038>

Érica Nascimento de Castro

<http://lattes.cnpq.br/5533828129290557>

RESUMO

O presente estudo resulta de um projeto de pesquisa desenvolvido no IFPA Campus Óbidos, articulado às práticas de educação profissional e tecnológica, tendo como objetivo investigar o uso do protocolo de comunicação UART no processo de engenharia reversa de equipamentos eletrônicos com a participação de discentes em atividades de laboratório. Para isso, utilizou-se o microcontrolador STM32 integrado ao ecossistema Arduino, possibilitando a análise, captura e futura replicação de comandos internos, tomando como referência o CDJ-200 da Pioneer. A metodologia adotada foi de caráter exploratório, envolvendo a abertura do equipamento, a identificação dos pontos RX, TX e GND correspondentes ao protocolo UART, além de testes com analisador lógico para determinar a velocidade de comunicação não documentada pelo fabricante. Também foi desenvolvido, em ambiente didático, um código para o STM32 capaz de interceptar e exibir os bytes trafegados, favorecendo a aprendizagem de conceitos de eletrônica digital e protocolos seriais. Os resultados preliminares demonstram comunicação estável e a captura de bytes válidos, permitindo a formação posterior de pacotes e, conseqüentemente, a manipulação das funções do aparelho via software, ao mesmo tempo em que contribuem para a formação técnica dos estudantes em temas como reuso tecnológico e engenharia reversa.

Palavras-chave: Engenharia reversa. Protocolo UART. STM32. Educação profissional e tecnológica. Comunicação serial.

1 INTRODUÇÃO

O avanço tecnológico na área da eletrônica e dos sistemas embarcados constitui um longo caminho de prática, estudo e inovação científica contínua, refletindo-se na diversificação das formas de produzir soluções baseadas em hardware e software, como dispositivos eletrônicos voltados à reprodução e mixagem de sons. Por serem aparelhos específicos, possuem alto valor agregado e costumam sofrer com a obsolescência programada, devido às intensas mudanças de mercado e ao alto custo de manutenção por serviços especializados. Nesse cenário, abre-se também um campo fértil para a formação de estudantes em cursos da área de informática e eletrônica, uma vez que tais equipamentos podem ser utilizados como recursos didáticos em atividades de ensino, pesquisa e extensão.

Assim, muitos profissionais optam por fazer modificações internas nos dispositivos, que exigem certo grau de conhecimento em eletrônica, na tentativa de atualizá-los e proporcionar sobrevida a estes. Frente a esse cenário, Santos, Arantes e Trajano (2023) destacam que se faz necessário empregar mecanismos e metodologias que assegurem condições socioambientais sustentáveis e favoráveis a todos no desenvolvimento tecnológico em que o mundo se encontra. Do ponto de vista educacional, trabalhar com a atualização e o reaproveitamento de equipamentos eletrônicos contribui para a formação de sujeitos críticos, capazes de compreender o impacto ambiental do descarte de lixo eletrônico e de propor soluções tecnológicas socialmente responsáveis.

Neste contexto, o presente estudo integra ações de educação profissional e tecnológica, ao articular conteúdos de eletrônica, programação e sistemas embarcados em atividades práticas de laboratório. O objetivo geral é investigar a utilização do protocolo **UART** na engenharia reversa de equipamentos eletrônicos, possibilitando sua reprogramação e atualização por meio do uso do microcontrolador STM32 e da IDE Arduino. Busca-se apresentar uma forma menos invasiva e totalmente atualizável de se realizar esse procedimento, ao mesmo tempo em que se proporciona aos discentes a vivência de um processo real de investigação aplicada. **Para isso, é realizado o estudo do protocolo UART, visto que este está empregado em diversos dispositivos, desde modelos mais antigos até equipamentos atuais.**

Este artigo é resultado do projeto de pesquisa intitulado “Uso do protocolo UART para engenharia reversa em equipamentos utilizando STM32 e IDE Arduino”, desenvolvido com estudantes do Instituto Federal de Educação, Ciência e Tecnologia do Pará – Campus Óbidos, no âmbito do Laboratório de Inovação Tecnológica do Baixo Amazonas (LITBA). O projeto conta com apoio do Edital nº 03/2025/PROPPG/IFPA – Auxílio Assistência Pesquisa, o que viabiliza a participação de discentes bolsistas nas atividades. O projeto teve como objetivo geral realizar a comunicação entre o microcontrolador STM32 e o dispositivo de mixagem CDJ-200. Como objetivos específicos: a sincronização da velocidade de comunicação (baud rate); a captura de bytes do aparelho de som pelo microcontrolador; e a impressão desses valores, configurando um ambiente de aprendizagem em que os estudantes experimentam, registram e analisam dados reais de um sistema embarcado.

2 DESENVOLVIMENTO

Para esta pesquisa foi utilizado o aparelho de discotecagem CDJ-200 da fabricante Pioneer, lançado no ano de 2005. Este faz uso do protocolo UART e possui, internamente em sua placa, pontos de conexão RX e TX (protocolo UART) usados para diagnósticos, atualização de firmware e comunicação interna. No contexto deste estudo, esses elementos foram explorados em atividades práticas de laboratório do IFPA Campus Óbidos, onde os discentes participaram ativamente do processo de identificação dos sinais, abertura do equipamento e registro dos testes, consolidando conhecimentos de eletrônica básica, instrumentação e programação.

O protocolo UART (Universal Asynchronous Receiver/Transmitter) é um dos protocolos de comunicação assíncrona mais utilizados para a troca de dados em série, devido à sua facilidade de conexão e comunicação (Silva, 2025). Utilizado para comunicar dois dispositivos entre si, este necessita de apenas três ligações, sendo um fio de transmissão de dados (TX), um de recepção (RX) e um fio comum (GND) (Santos, 2024). A exploração desses conceitos em situações reais de laboratório contribui para que os

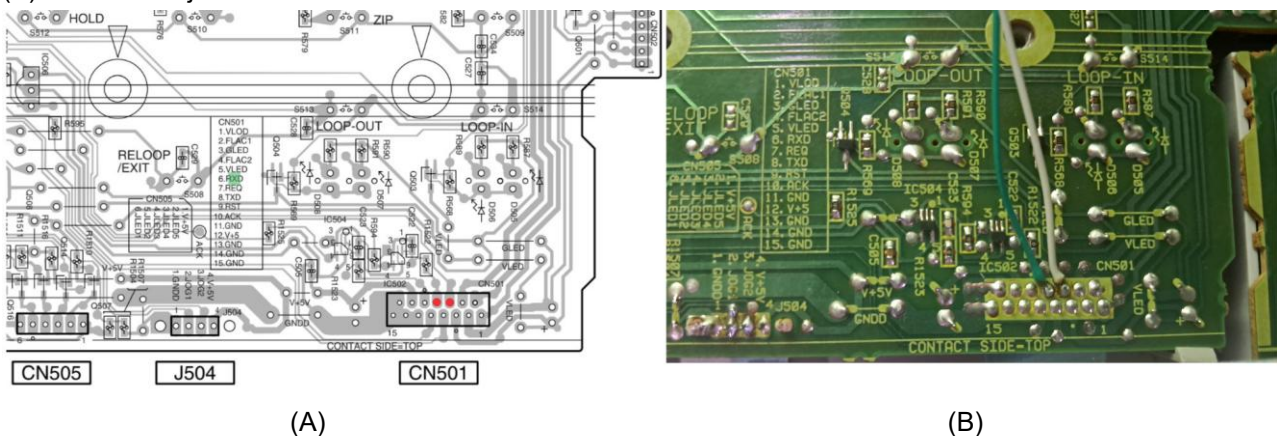


estudantes relacionem conteúdos teóricos estudados em sala de aula com a prática de análise de sinais e de protocolos de comunicação.

Facilmente adaptável, esse protocolo apresenta três modos de transmissão de dados, sendo: o primeiro, chamado de simplex, determina que os dados trafeguem em apenas uma única direção; o segundo, half-duplex, permite que os dados transitem de ambos os lados, porém um dispositivo de cada vez; e o terceiro (mais utilizado), denominado full-duplex, admite que os dados sejam enviados e recebidos de forma simultânea entre os dispositivos conectados. A discussão desses modos, aliada à observação dos sinais no analisador lógico, foi incorporada às atividades didáticas, fortalecendo o entendimento dos discentes sobre arquitetura de comunicação serial.

Ao abrir o CDJ-200, em seu circuito encontram-se pontos de conexão entre a placa principal e a secundária. Na placa secundária (localizada na parte superior do CDJ) há uma lista gravada indicando o tipo de conexão de cada um desses pontos. Nesses, devem ser soldados dois fios, um no ponto de conexão 6, referente ao RX, e o segundo no ponto 8, correspondendo ao TX, conforme a imagem abaixo:

Figura 1 - Pontos de conexão na placa CDJ-200: (A) Desenho técnico (pontos de conexão em vermelho), (B) Placa física já com os fios soldados.



Fonte: Elaborado pelos autores (2025).

Também foi realizada a soldagem de um fio no ponto de escrita GND, já que este se refere ao negativo do circuito, utilizado para equiparar os potenciais elétricos nas comunicações. Todo esse procedimento foi realizado em ambiente supervisionado, servindo como situação de aprendizagem sobre segurança em eletrônica, técnicas de dessoldagem/soldagem e leitura de serigrafia de placas, aspectos importantes na formação de técnicos e tecnólogos.

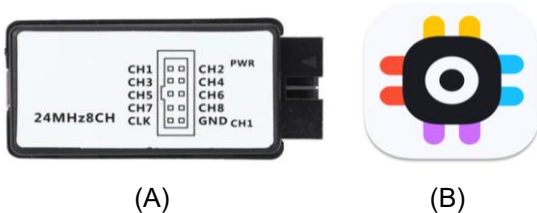
Os passos da pesquisa a seguir estão divididos em subtópicos, nos quais são apresentados a estrutura lógica e o encadeamento metodológico utilizados para o desenvolvimento deste artigo. O primeiro passo consiste no estabelecimento da comunicação serial entre o equipamento e o microcontrolador, por meio do alinhamento da velocidade de comunicação (baud rate). A segunda etapa refere-se à leitura dos pacotes de dados que circulam no funcionamento do CDJ-200 e a terceira está relacionada com os resultados alcançados e esperados desta pesquisa, tanto na perspectiva técnica quanto formativa. Ressalta-se que essa pesquisa é de caráter exploratório e que muitos dos dados foram obtidos por meio de tentativa e erro, devido à baixíssima quantidade de informações sobre o aparelho em específico, o que exigiu dos estudantes postura investigativa, registro sistemático e análise crítica das evidências.

2.1 Método de encontro do Baud Rate

O termo baud rate refere-se à taxa de transmissão de dados por segundo e, quanto maior o valor, maior será a quantidade de dados que trafegam nesse intervalo (Bernatzki, Souza e Silva, 2024). Para estabelecermos uma conexão clara entre o CDJ-200 e o microcontrolador STM32, é necessário encontrar a velocidade de comunicação; essa informação não se encontra presente no manual de serviço desse equipamento nem em artigos, fóruns ou semelhantes. Para isso, foi desenvolvido um método de descoberta por meio do uso de um analisador lógico, explorado com os discentes como atividade investigativa orientada.

Foi utilizado um analisador lógico de 24 MHz e 8 canais (Figura 2), no qual o pino transmissor (TX) da placa é conectado ao pino 1 do analisador e o GND da placa é conectado ao GND do analisador. Junto a isso, também foi utilizado o software Logic 2 (versão 2.4.36). Os estudantes foram responsáveis por configurar o instrumento, observar os sinais coletados e registrar os resultados, exercitando a leitura de waveforms e a interpretação de dados digitais.

Figura 2 - Imagens dos itens utilizados: (A) Analisador Lógico de 24 MHz e 8 canais, (B) Ícone do programa Logic2.



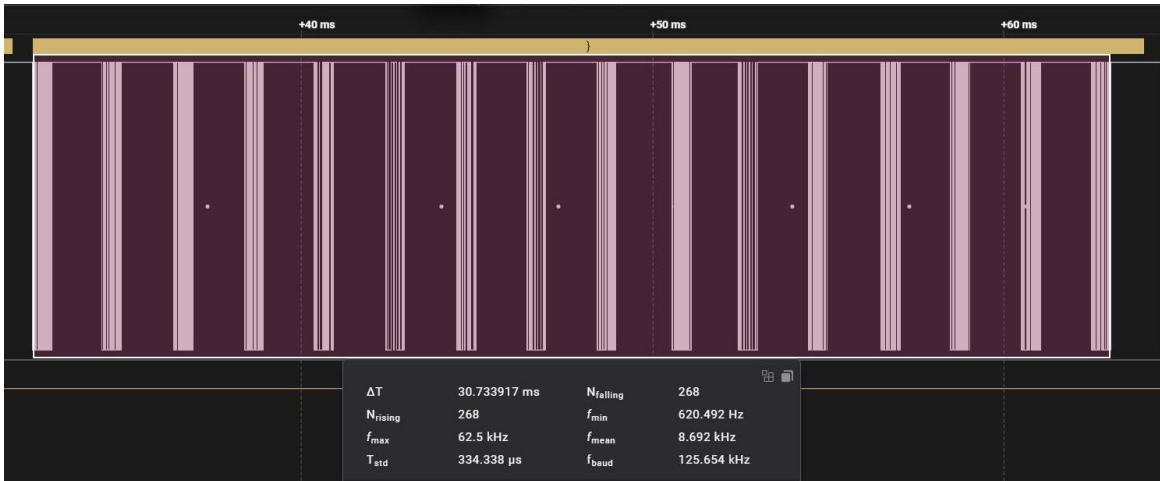
Fonte: Elaborado pelos autores (2025).

Realizadas essas conexões, configura-se no software a porta 1 como Async Serial (Serial Assíncrono); em seguida, realiza-se uma leitura em um intervalo de tempo de 5 segundos. A faixa obtida (Figura 3) apresenta diversas leituras de dados. Para chegar ao valor aproximado do baud rate, isola-se um campo de informação; nesse exemplo, selecionou-se o espaço de tempo referente ao caractere “}”.

O protocolo UART, ao iniciar o envio de sinais, baixa o nível lógico de sua borda (colocando-o em nível baixo) e, em seguida, envia os blocos de dados, finalizando as instruções ao elevar o nível lógico (colocando o valor em nível alto). Desse modo, em nosso exemplo, selecionamos as informações referentes ao caractere “}” a partir de sua primeira borda de descida até sua última borda de subida.



Figura 3 - Leitura UART do equipamento CDJ-200.

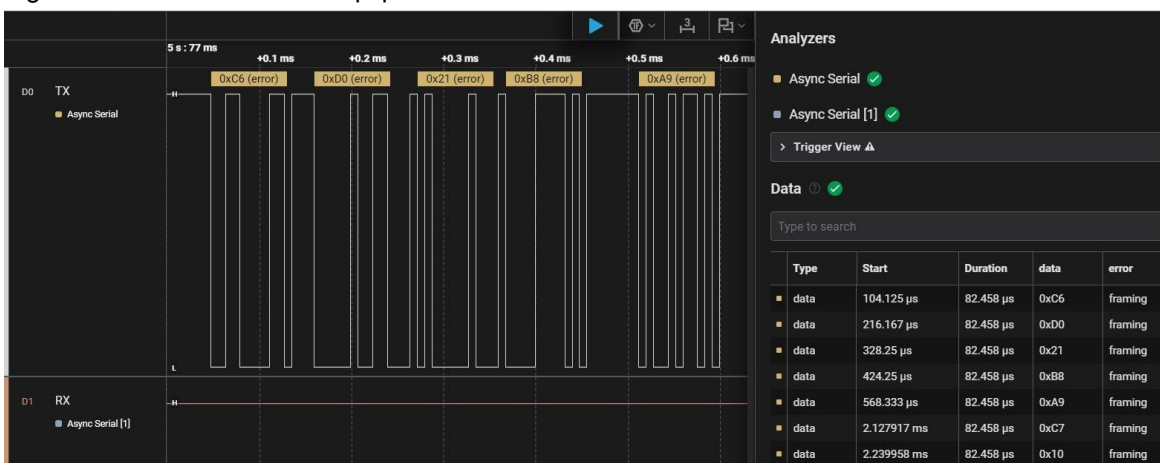


Fonte: Elaborado pelos autores (2025).

Com isso, obtém-se o valor de fbaud; essa informação refere-se ao provável valor de comunicação, sendo este de 125.000 MHz. Nota-se que este valor não é padrão, o que levanta a hipótese de estar errado ou de a fabricante ter adotado essa frequência exclusiva. Para a confirmação de uma dessas hipóteses, a etapa seguinte se dá na leitura de informações já utilizando o provável valor de baud rate; também foi testado um valor padrão próximo de 115.200 MHz. No primeiro teste, o analisador lógico estava na seleção de baud rate da porta em modo automático, o que fazia com que, em determinado momento, não conseguisse capturar a informação de forma completa.

No segundo teste, iniciou-se com a configuração da frequência de comunicação da porta 1 para 115.200 MHz e realizou-se uma leitura. Nota-se que os pacotes obtidos apresentam o estado “error framing”; esse erro ocorre devido ao analisador lógico não conseguir encontrar o byte de parada da informação, em razão de a frequência de comunicação estar incorreta

Figura 4 - Leitura UART do equipamento CDJ-200 com o baud rate de 115.200 MHz.

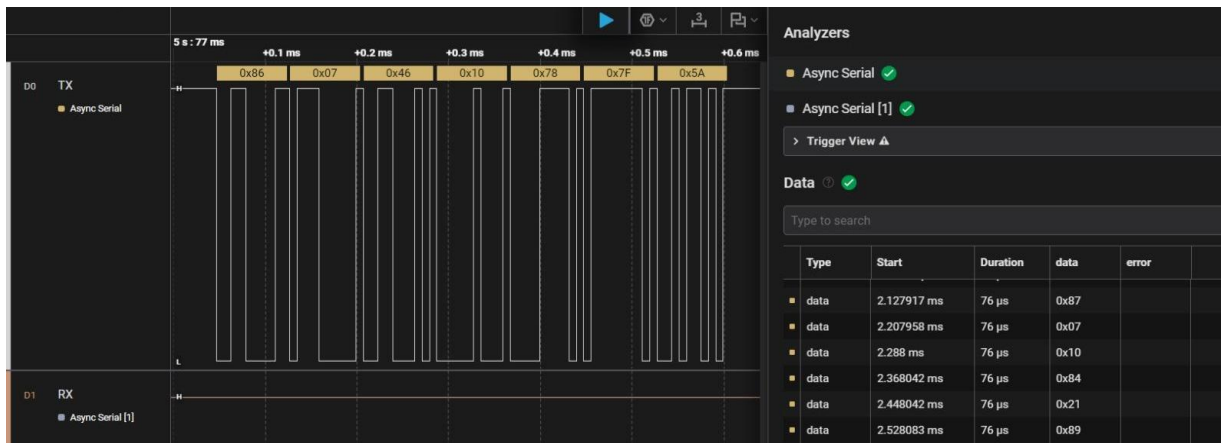


Fonte: Elaborado pelos autores (2025).

Isso indica que, de fato, o CDJ-200 não utiliza uma faixa de frequência convencional. Já no terceiro teste, ao alterar a frequência de comunicação para 125.000 MHz, obtém-se uma leitura de pacotes limpa, na qual nenhum dos dados apresenta erro. Desse modo, podemos afirmar que essa é a frequência de comunicação utilizada pelo equipamento. A

realização comparativa desses testes, acompanhada de discussões em grupo, contribuiu para o desenvolvimento do raciocínio lógico, da capacidade de formular hipóteses e de validar resultados entre os estudantes.

Figura 5 - Leitura UART do equipamento CDJ-200 com o baud rate de 125.000 MHz.



Fonte: Elaborado pelos autores (2025).

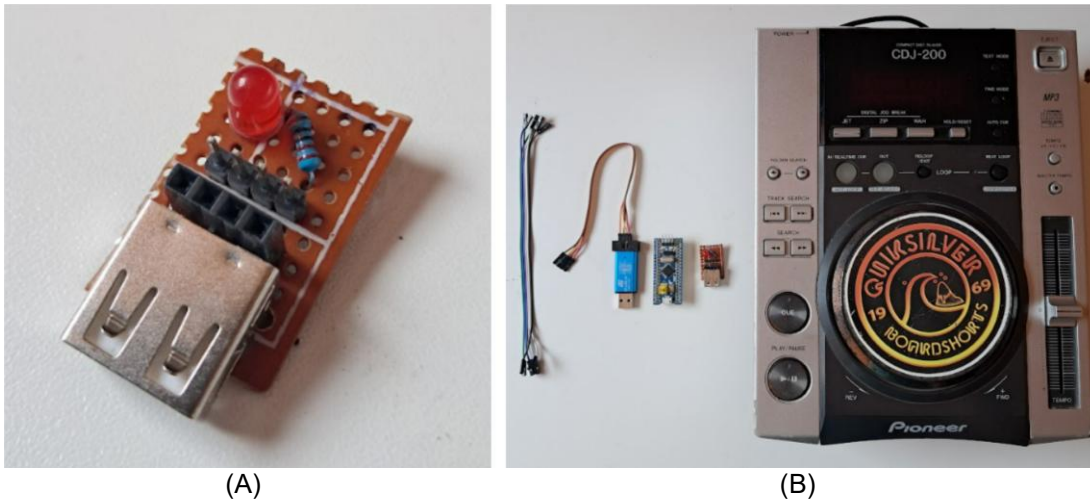
2.2 Captura de Pacotes Utilizando o STM32

Lançado no ano de 2007 pela empresa STMicroelectronics, o STM32 é baseado no núcleo ARM Cortex-M3; por padrão, apresenta portas nativas para comunicação em diversos protocolos, incluindo o protocolo UART (Brown, 2015, p. 33). Outra característica que chama a atenção nesse microcontrolador é sua frequência de operação, que pode chegar até 72 MHz, e uma memória Flash de 64 KB, o que representa um diferencial em comparação à maioria dos microcontroladores tradicionais, como, por exemplo, o Arduino Uno.

Pelo fato de se necessitar de uma comunicação rápida e de um tempo de resposta ágil, o STM32 torna-se um candidato adequado para os estudos de testes. Além disso, foi utilizado o conjunto de bibliotecas Arduino_Core_STM32, composto de arquivos e ferramentas que permitem à plataforma Arduino IDE adicionar suporte a microcontroladores da família STM32, possibilitando programar essas unidades usando a infraestrutura do Arduino (a partir da versão 2.x), incluindo esboços e bibliotecas. Na perspectiva educacional, esse ecossistema reduz a curva de aprendizagem para estudantes que já têm familiaridade com a IDE Arduino, facilitando a migração para um hardware mais robusto e estimulando a autonomia na exploração de novas placas.

Com as ligações elétricas feitas no CDJ-200 e com sua velocidade de comunicação encontrada, pode-se estabelecer uma comunicação entre este e o STM32. Para isso, utilizou-se o ambiente de programação Arduino IDE para a escrita do código, um módulo ST-Link (utilizado para transferir o código para dentro da memória do STM32), fios jumpers para a conexão e um módulo de fabricação própria (Figura 6) para facilitar as ligações entre cada um dos componentes supracitados.

Figura 6 - Componentes utilizados no estudo: (A) Módulo para auxiliar as conexões elétricas, (B) Dispositivos e equipamentos do estudo.



Fonte: Elaborado pelos autores (2025).

Para facilitar o acoplamento entre o CDJ e o STM32, as ligações internas deste foram realizadas utilizando um cabo USB; desse modo, as conexões entre eles foram feitas da seguinte forma: o fio de transmissão (TX) do CDJ-200 foi conectado ao pino de recepção (RX) do STM32, sendo a porta PA10. Também é necessária a conexão dos GNDs dos dois dispositivos, para se obter a equiparação dos potenciais elétricos. Esses procedimentos foram executados pelos estudantes, sob orientação docente, reforçando noções de mapeamento de pinos, organização de cabos e boas práticas de montagem de protótipos.

Com as devidas conexões estabelecidas, o próximo passo se dá no desenvolvimento do código, que irá interceptar os bytes que trafegam no equipamento de som e os exibirá no monitor serial. Com essa lógica de funcionamento, o seguinte código foi desenvolvido:

Figura 7 - Código de captura dos bytes de comunicação.

```

leitura_pacotes_simples.ino
1 // CONEXÕES:
2 // GND(CDJ) -> GND(STM32)
3 // TX(CDJ) -> RX(PA10)
4
5 void setup() {
6     Serial.begin(115200); // UART via USB (debug)
7     Serial1.begin(125000); // UART física conectada ao dispositivo externo
8
9     Serial.println("STM32 UART RX Iniciado...");
10 }
11
12 void loop() {
13     // Verifica se chegou um byte
14     if (Serial1.available()) {
15         byte b = Serial1.read();
16
17         // Exibe o valor no Serial Monitor
18         Serial.print("0x");
19         if (b < 0x10) Serial.print("0"); // formatação bonitinha
20         Serial.println(b, HEX);
21         Serial.print(" ");
22     }
23 }
    
```

Fonte: Elaborado pelos autores (2025).

Nesse código, na função void setup(), inicia-se com o alinhamento das velocidades de comunicação, no qual a primeira linha representa a velocidade de baud rate do monitor serial. Em trechos anteriores deste artigo citou-se que o CDJ está se comunicando em uma velocidade não convencional de 125.000 MHz; essa taxa de transmissão não está presente no monitor serial e, desse modo, utiliza-se a velocidade mais próxima, sendo 115.200 MHz. A segunda linha define justamente a velocidade de comunicação do CDJ encontrada em testes anteriores.

As linhas de código 14 e 15 são responsáveis por armazenar cada byte capturado; já as linhas 18 a 21 realizam a formatação dos bytes para hexadecimal e os exibem no monitor serial, para que se tenha uma resposta visual de quais dados estão trafegando no aparelho, conforme demonstra a Figura 8. Na prática, essa etapa foi utilizada como atividade de leitura e interpretação de código, em que os discentes precisaram compreender cada instrução, propor modificações e observar os efeitos no fluxo de dados.

Figura 8 - Resultado de captura de bytes do CDJ-200 utilizando o STM32.

```
06:23:41.243 -> 0x00
06:23:41.243 -> 0x00
06:23:41.243 -> 0x00
06:23:41.243 -> 0x04
06:23:41.243 -> 0x85
06:23:41.243 -> 0x07
06:23:41.243 -> 0x46
06:23:41.243 -> 0x75
```

Fonte: Elaborado pelos autores (2025).

Neste recorte, percebe-se o funcionamento integral do código e das conexões: cada uma dessas linhas carrega um byte; a junção desses bytes gera o que se denomina protocolo de comunicação, e cada um desses protocolos representa uma função do aparelho, como, por exemplo, exibir no display o tempo de cada música, verificar o estado de um botão e demais funções do dispositivo. A análise desses pacotes, realizada em conjunto com os estudantes, favorece a compreensão de como comandos de alto nível (apertar um botão, girar um jog wheel) se traduzem em mensagens binárias específicas, conectando teoria e prática.

2.3 Resultados alcançados e esperados

As primeiras etapas deste estudo foram alcançadas, uma vez que a integração entre o CDJ-200 e o microcontrolador STM32 foi realizada, somada à interceptação dos bytes de comunicação do dispositivo de som. O próximo passo dessa pesquisa será a organização dos pacotes a partir da identificação do byte de início (chamado de sinal de borda de descida) e do byte de fim (nomeado de sinal de borda de subida). A partir desses valores será possível definir com precisão a função de cada pacote do CDJ-200 e, em seguida, replicá-los, de modo a realizar o acionamento dessas funções no aparelho por meio de comandos enviados do microcontrolador.

Como destacam Clements et al. (2020), fornecedores de microcontroladores tendem a investir em determinadas bibliotecas e a aplicar licenças mais flexíveis a estas, o que pode significar um aumento no uso de componentes licenciados. A respeito da possibilidade



de o presente estudo propiciar trabalhos futuros, seja com intenção de executar planos no âmbito educacional ou mesmo comercial, mostra-se fundamental registrar informações sobre as principais licenças atreladas ao uso dos arquivos e bibliotecas presentes no repositório STM32duino.

Tabela 1 - Comparativo de licenças do STM32duino/Arduino_Core_STM32

Licença	Uso educacional	Uso comercial	Liberação de código	Observações
LGPLv2.1	Permitido	Permitido	Necessária somente em partes alteradas	Liberar modificações da biblioteca, permitir substituição
BDS - 3 Cláusulas	Permitido	Permitido	Não necessária	Preservação de copyright e não usar nome original para promoção
Ultimate Liberty License	Permitido	Permitido	Não necessária, porém o uso deve ser em hardware ST, sem sujeição a termos que obriguem código aberto	Certos componentes de software são incompatíveis com códigos sujeitos à licença GPL
Apache 2.0	Permitido	Permitido	Não necessária, porém deve-se indicar em caso de alteração e incluir aviso de patente	Inclui cláusulas de patente
MIT	Permitido	Permitido	Não necessária	Manter o aviso de copyright conforme exigido

Fonte: Elaborado pelos autores (2025)

Na Tabela 1 estão dispostas informações sobre a permissão para o uso educacional e comercial, a liberação de código — referente ao ato de tornar público o código-fonte do produto final — e observações consideradas válidas até o atual momento da pesquisa. Muitas das licenças observadas são consideradas permissivas, tais como a licença BSD de três cláusulas, que diz respeito à permissão para redistribuição de código-fonte contendo aviso de direitos autorais, isenção de responsabilidade por parte da fabricante e proibição do uso do nome do detentor dos direitos e de seus colaboradores para promoção de quaisquer produtos derivados (Open Source Initiative, 2025). A Ultimate Liberty License (ULL) é fundamentalmente permissiva; entretanto, possui maiores restrições quanto ao seu escopo de uso e redistribuição. O acordo permite a redistribuição e o uso do código-fonte em binário, com ou sem modificações, desde que o software ou firmware seja executado somente ou em combinação com um dispositivo de unidade de processamento da fabricante (STMicroelectronics, 2025).

Utilizar o STM32duino permite aproveitar o ecossistema Arduino, a facilidade de esboços e a enorme biblioteca de exemplos, juntamente com os benefícios da potência dos microcontroladores STM32, com frequências mais altas, maior quantidade de memória e periféricos avançados. Isso reduz a curva de aprendizado entre utilizadores de Arduino que pretendem migrar para exemplos de hardware mais capacitados, o que é particularmente relevante em contextos educacionais, pois viabiliza que estudantes avancem gradativamente de projetos introdutórios para desafios mais complexos, como a engenharia reversa de equipamentos comerciais.

Em termos formativos, o desenvolvimento das etapas descritas permitiu que os estudantes vivenciassem um ciclo completo de investigação em engenharia reversa: formulação do problema, levantamento de hipóteses sobre o protocolo de comunicação,

planejamento de testes, uso de ferramentas de medição, implementação de código e validação dos resultados. Essa trajetória contribuiu para aproximar os conteúdos trabalhados em sala de aula das demandas reais de um projeto de pesquisa aplicada, favorecendo o protagonismo discente, o trabalho colaborativo e a compreensão do papel social da tecnologia na ampliação da vida útil de equipamentos eletrônicos.

3 CONCLUSÃO

Neste estudo sobre a engenharia reversa em equipamentos por meio do uso do protocolo UART e do microcontrolador STM32, foi possível compreender e aplicar diversos conceitos atrelados à engenharia de protocolos de comunicação. Um achado importante foi a descoberta do baud rate do dispositivo em 125.000 MHz, um valor não padrão que não estava documentado em manuais ou fóruns, exigindo o desenvolvimento de um método de descoberta por meio do uso de um analisador lógico e de testes sucessivos realizados em contexto de laboratório didático.

Essa falta de informações específicas pode estar relacionada a segredos comerciais somados à baixa quantidade de estudos sobre a manipulação de aparelhos de discotecagem por meio de seu próprio protocolo de comunicação interna, visto que a maioria dessas manipulações exige modificações no próprio hardware. O resultado desse cenário é um trabalho mais oneroso e, por consequência, mais lento para melhor acurácia dos dados, o que, por outro lado, se converteu em oportunidade formativa para os estudantes envolvidos, que precisaram formular hipóteses, registrar experimentos e interpretar resultados.

Ressalta-se que este estudo exploratório alcançou as etapas iniciais de interceptação dos bytes que compõem os pacotes e, na presente data, continua em andamento como parte de um projeto de pesquisa realizado no Instituto Federal de Educação, Ciência e Tecnologia do Pará – Campus Óbidos, com apoio do Edital nº 03/2025/PROPPG/IFPA – Auxílio Assistência Pesquisa. Demais testes continuarão a ser executados para o cumprimento das demais etapas, envolvendo a identificação e a análise dos pacotes referentes a cada função do dispositivo, processo que segue sendo desenvolvido com participação ativa de discentes em atividades de iniciação científica.

A possibilidade de este estudo gerar resultados não somente para fins educacionais, mas também comerciais, torna necessário o aprofundamento em documentação específica, como legislações e licenças de uso de bibliotecas e firmwares, de modo a garantir a legalidade do projeto. Do ponto de vista educacional, as ações realizadas evidenciam o potencial da engenharia reversa como estratégia pedagógica em cursos da área de eletrônica e informática, ao articular teoria e prática em situações reais de resolução de problemas, fortalecendo a integração entre ensino, pesquisa e inovação tecnológica no IFPA Campus Óbidos.

REFERÊNCIAS

BERNATZKI, Roberto Romão; SOUZA, Marcelo Silva; SILVA, Gabriel Souza.

Prototipagem de um concentrador de sinais seriais para integração de sistemas navais. Niterói, RJ: Centro de Apoio a Sistemas Operativos (CASOP), 2024. 4 f.

Disponível em: https://www.sige.ita.br/edicoes-anteriores/2024/st/243445_1.pdf. Acesso em: 14 de nov. 2025.

BROWN, Geoffrey. **Discovering the STM32 Microcontroller**. [s. l.], 2015. Disponível em: https://www.academia.edu/44443006/Descobrimdo_o_Microcontrolador_STM32. Acesso em: 13 de nov. 2025.

CLEMENTS, Abraham A. et al. **HALucinator: Firmware Re-hosting Through Abstraction Layer Emulation**. In: USENIX SECURITY SYMPOSIUM, 29., 2020. **Proceedings of the 29th USENIX Security Symposium**. [S.l.]: USENIX Association, 2020. p. 1201-1218. Disponível em: <https://www.usenix.org/conference/usenixsecurity20/presentation/clements>. Acesso em: 14 de nov. 2025.

GITHUB. **Arduino_Core_STM32: STM32 core support for Arduino**. [S.l.: s.n.], [s.d.]. Disponível em: https://github.com/stm32duino/Arduino_Core_STM32. Acesso em: 13 nov. 2025.

OPEN SOURCE INITIATIVE. **The 3-Clause BSD License** [online]. [S. l.], 2025. Disponível em: <https://opensource.org/license/BSD-3-Clause>. Acesso em 10 de out. 2025.

SANTOS, Eduardo da Silva; ARANTES, Mayara Cassiano; TRAJANO, Sílvia. **Sucatrônica para os cursos técnicos em Automação Industrial e Eletrotécnica: uma proposta de oficina para a sustentabilidade**. In: SEMINÁRIO INTERNACIONAL DE EDUCAÇÃO, EMPREENDEDORISMO E GESTÃO DE PROJETOS, 1., 2023, Engenheiro Paulo de Frontin, RJ. **Anais do Seminário Internacional de Educação, Empreendedorismo e Gestão de Projetos**. Engenheiro Paulo de Frontin, RJ: IFRJ-CEPF, 2023. Disponível em: <https://www.even3.com.br/anais/i-seminario-internacional-de-educacao-empreendedorismo-e-gestao-de-projetos-306748/783642-sucatronica-para-os-cursos-tecnicos-em-automacao-industrial-e-eleotecnica-uma-proposta-de-oficina-para-a-suste>. Acesso em: 10 nov. 2025.

SANTOS, Paulo Ricardo de Almeida. **Implementação de um datalogger utilizando DMA e UART**. 2024. 57 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Eletrônica e de Telecomunicações) – Universidade Federal de Uberlândia, Uberlândia, 2024. Disponível em: <https://repositorio.ufu.br/handle/123456789/44421>. Acesso em: 9 nov. 2025.

SILVA, Samuel Muniz. **Protocolo de comunicação microcontrolado por STM32 para o acionamento de estudos de banco de envelhecimento de IGBT's**. 2025. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Centro de Tecnologia, Universidade Federal do Ceará, Fortaleza, 2025. Disponível em: <http://repositorio.ufc.br/handle/riufc/81189>. Acesso em: 12 de nov. 2025.

STMICROELECTRONICS. **Ultimate Liberty Software License Agreement**. SLA0044 — Rev. 6. [S.l.]: STMicroelectronics, 2025. 1 p. (Software license agreement; SLA0044). Disponível em: https://www.st.com/resource/en/license_agreement/dm00216740.pdf. Acesso em: 3 de nov. 2025.

STMICROELECTRONICS. **Ultimate Liberty Software License Agreement**. SLA0044 — Rev. 6. [s. l.]: STMicroelectronics, 2025. 1 p. (Software license agreement, SLA0044). Disponível em: https://www.st.com/resource/en/license_agreement/dm00216740.pdf. Acesso em: 3 nov. 2025.