

**Desenvolvimento de plataforma embarcada para ensino**  
**Development of an embedded platform for teaching**  
**Desarrollo de plataforma embarcada para la enseñanza**

**Fábio Rocha da Silva**

ORCID: <https://orcid.org/0000-0002-2227-0189>

Universidade Federal de Itajubá, Brasil

E-mail: [fabio.eco15@gmail.com](mailto:fabio.eco15@gmail.com)

**Augusto Cesar Rossi de Mattos**

ORCID: <https://orcid.org/0000-0001-9718-3074>

Universidade Federal de Itajubá, Brasil

E-mail: [augustocrmattos@gmail.com](mailto:augustocrmattos@gmail.com)

**Gabriel Silva Marcatto**

ORCID: <https://orcid.org/0000-0003-0253-886X>

Universidade Federal de Itajubá, Brasil

Email: [gabriel.marcatto7@gmail.com](mailto:gabriel.marcatto7@gmail.com)

**Rafael de Moura Moreira**

ORCID: <https://orcid.org/0000-0001-7473-5737>

Universidade Federal de Itajubá, Brasil

Email: [rafaelmmoreira@gmail.com](mailto:rafaelmmoreira@gmail.com)

**Rodrigo Maximiano Antunes de Almeida**

ORCID: <https://orcid.org/0000-0002-2663-5464>

Universidade Federal de Itajubá, Brasil

Email: [rmaalmeida@gmail.com](mailto:rmaalmeida@gmail.com)

Recebido: 12/11/2018 | Revisado: 17/12/2018 | Aceito: 14/02/2019 | Publicado: 22/02/2019

**Resumo**

Este trabalho tem como objetivo o desenvolvimento de uma plataforma embarcada que possa guiar os primeiros passos de um aluno na área de programação embarcada. É descrito o desenvolvimento de um pacote de software para suporte ao aprendizado de programação embarcada, principalmente fazendo o uso de sistemas operacionais de tempo real, bem como documenta as atividades iniciais da plataforma PQDB (Pão de Queijo Development Board). As técnicas de desenvolvimento para sistemas embarcados não são as mesmas das aplicadas

em desktops e mainframes. Existem restrições de memória, capacidade limitada de processamento, gerenciamento dos dispositivos de entrada e saída, protocolos de comunicação diversos, isto porque cada sistema embarcado é um projeto diferente que pode não seguir padrões, dessa maneira o conhecimento do hardware é essencial para o projeto do software que será incorporado no dispositivo. As implementações práticas foram realizadas com a ferramenta Kinetis Design Studio 3 ® IDE, o sistema operacional FreeRTOS®, a placa de desenvolvimento FRDM-K64F da NXP e a PQDB projetada e fabricada na Universidade Federal de Itajubá. Neste trabalho apresentamos como esta placa auxilia no aprendizado do aluno em software embarcado, apresentando um guia rápido de como portar o FreeRTOS além de desenvolver bibliotecas e drivers para comunicação com periféricos. Conclui-se que a utilização de uma placa pré-montada simplifica o entendimento do aluno na medida que remove camadas de complexidade da atividade de aprendizagem. O mesmo acontece com a utilização de um sistema operacional de tempo real, que abstrai as questões de restrições de tempo para sistemas críticos. Também se notou que a placa PQDB oferece diversos recursos que, se bem explorados, garantem um ótimo início na área de desenvolvimento de softwares embarcados.

**Palavras-chave:** Embarcados; Firmware; Sistemas Operacionais de Tempo real.

### **Abstract**

This work has as objective the development of an embedded platform that can guide the first steps of a student in the area of embedded programming. Describes the development of a package of software to support the learning of embedded programming, mainly using real time operating systems, as well as documents the initial activities of the PQDB platform (Pão de Queijo Development Board). The development techniques to embedded are not the same applied to desktops and mainframes. There are memory restrictions, limited processing capacity, input and output device management, various communication protocols, this because each embedded system is a different project that may not follow patterns, in this way the knowledge of the hardware is essential to the software design that will be embedded in the device. Thus, we studied the fundamentals of microprocessed and microcontrolled systems, general characteristics of operating systems, real-time systems, and some relevant features of software engineering. The implementations were performed with the Kinetis Design Studio 3 ® IDE, the FreeRTOS ® operating system, the NXP FRDM-K64F development board and the Pão de Queijo Development Board (PQDB) that was designed and developed in the Federal University of Itajubá. In this work we present how this board assists student learning

in embedded software, presenting a quick guide to how to port the FreeRTOS in addition to developing libraries and drivers to communicate with peripherals. It is concluded that the use of a pre-assembled board simplifies the understanding of the student as it removes layers of complexity from the learning activity. The same happens with the use of a real time operating system, that abstract the time constraints questions to critical systems. It was also noted that the PQDB board offers several features that, if well exploited, guarantee a great start in the area of development of embedded software.

**Keywords:** Embedded; Firmware; Real Time Operating Systems.

### **Resumen**

Este trabajo tiene como objetivo el desarrollo de una plataforma embarcada que pueda guiar los primeros pasos de un alumno en el área de programación embarcada. Se describe el desarrollo de un paquete de software para apoyar el aprendizaje de programación embarcada, principalmente haciendo el uso de sistemas operativos de tiempo real, así como documenta las actividades iniciales de la plataforma PQDB (Pan de Queso Development Board). Las técnicas de desarrollo para sistemas embarcados no son las mismas que las aplicadas en escritorios y mainframes. Existen restricciones de memoria, capacidad limitada de procesamiento, administración de dispositivos de entrada y salida, protocolos de comunicación diversos, porque cada sistema embarcado es un proyecto diferente que puede no seguir patrones, de manera que el conocimiento del hardware es esencial para el proyecto del proyecto software que se incorporará en el dispositivo. Las implementaciones prácticas se realizaron con la herramienta Kinetis Design Studio 3 ® IDE, el sistema operativo FreeRTOS®, la placa de desarrollo FRDM-K64F de NXP y la PQDB proyectada y fabricada en la Universidad Federal de Itajubá. En este trabajo presentamos cómo esta placa ayuda en el aprendizaje del alumno en software embarcado, presentando una guía rápida de cómo portar el FreeRTOS además de desarrollar bibliotecas y controladores para comunicación con periféricos. Se concluye que la utilización de una placa premontada simplifica el entendimiento del alumno en la medida que quita capas de complejidad de la actividad de aprendizaje. Lo mismo ocurre con la utilización de un sistema operativo en tiempo real, que abstraen las cuestiones de restricciones de tiempo para sistemas críticos. También se notó que la placa PQDB ofrece diversos recursos que, si bien explotados, garantizan un óptimo inicio en el área de desarrollo de softwares embarcados.

**Palabras clave:** Embarcados; Firmware; Sistemas Operativos de Tiempo real.

## **1. Introdução**

Com o rápido desenvolvimento tecnológico da eletrônica, os circuitos integrados passaram a compactar um maior número de transistores, ter a capacidade de resolução de problemas mais complexos e se tornaram cada vez mais baratos. Toda essa evolução implica na disseminação de dispositivos que contenham microprocessadores “embarcados” em seu núcleo. Diversos produtos microprocessados e microcontrolados são aplicados nos mais diversos setores, desde dispositivos de entretenimento como televisores, celulares e consoles de jogos até sistemas de navegação aeroespacial, radares, satélites e computadores de bordo. Desse modo, o termo sistema embarcado é utilizado para se referir a um sistema computacional de função específica inserido em um produto maior.

À medida que o projeto do sistema cresce o número de serviços que o sistema deve atender também cresce, dessa maneira é necessário que cada tarefa utilize a CPU por um tempo determinado. A solução para este problema é implementar um sistema operacional. Através de um SO é possível garantir que as requisições de execução das tarefas sejam atendidas com uma regularidade bem definida, além de permitir que os processos se sincronizem e que a memória seja melhor gerenciada. Contudo, para um sistema embarcado é necessário que o SO gerencie mais um atributo, o tempo. Por alguns sistemas necessitarem de que o tratamento dos dados seja determinístico faz-se necessário que o sistema monitore suas I/O's com restrições de tempo bem definidas.

Como o objetivo é o desenvolvimento de uma plataforma embarcada para facilitar o aprendizado de iniciantes em programação embarcada, neste trabalho será apresentada a placa PQDB, será demonstrada a sua utilização com uma placa de desenvolvimento microcontrolada além de apresentar um estudo de caso da aplicação de sistemas operacionais de tempo real em experimentos reais. Dessa maneira, possibilitando que análises adequadas sejam feitas e que possam determinar as vantagens, desvantagens e as complexidades envolvidas em sistemas de tempo real.

## **2. Plataforma PQDB**

A plataforma PQDB (Pão de Queijo Development Board) foi desenvolvida na Universidade Federal de Itajubá visando simplificar o processo de aprendizagem em programação embarcada de maneira independente do tipo do processador utilizado, possuindo os mesmos conjuntos de periféricos. Assim, o aluno pode conectar ao kit didático a qualquer

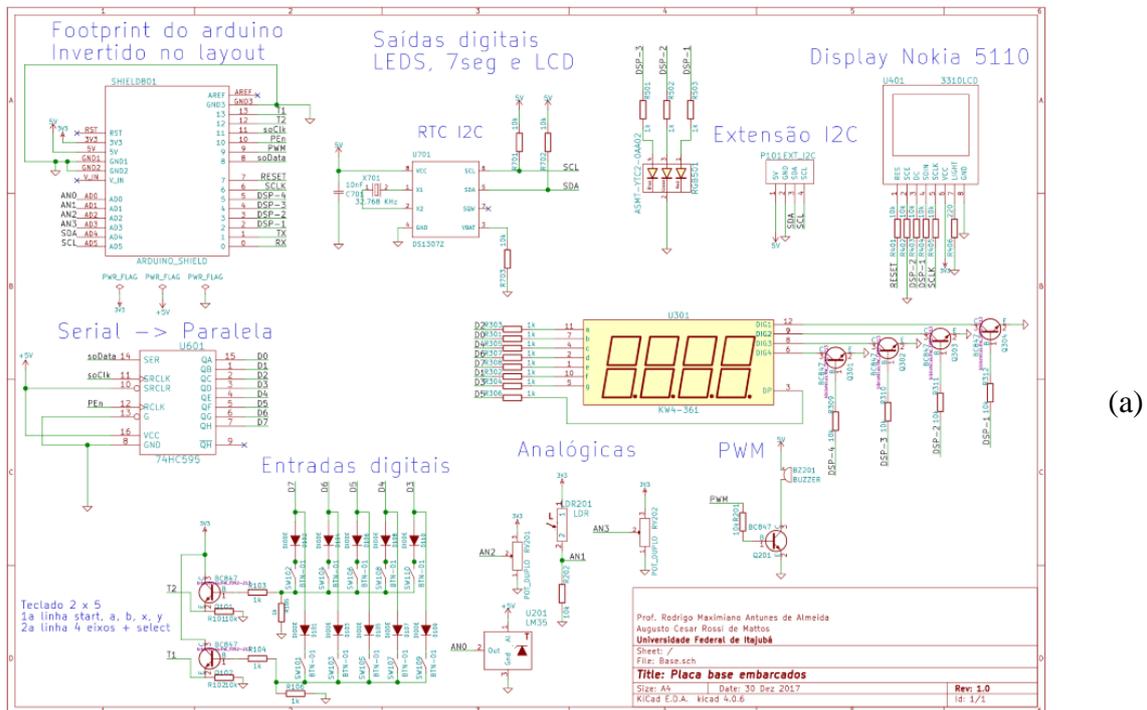
placa controladora que possua o mesmo footprint da plataforma Arduino. A Tabela 1 indica todos os periféricos incorporados na PQDB bem como os conceitos que podem ser explorados pela utilização da plataforma. A Figura 1.a apresenta o esquemático da placa e a Figura 1.b ilustra o perfil final da placa fabricada e pronta para a utilização.

Tabela 1 – Periféricos incorporados na PQDB

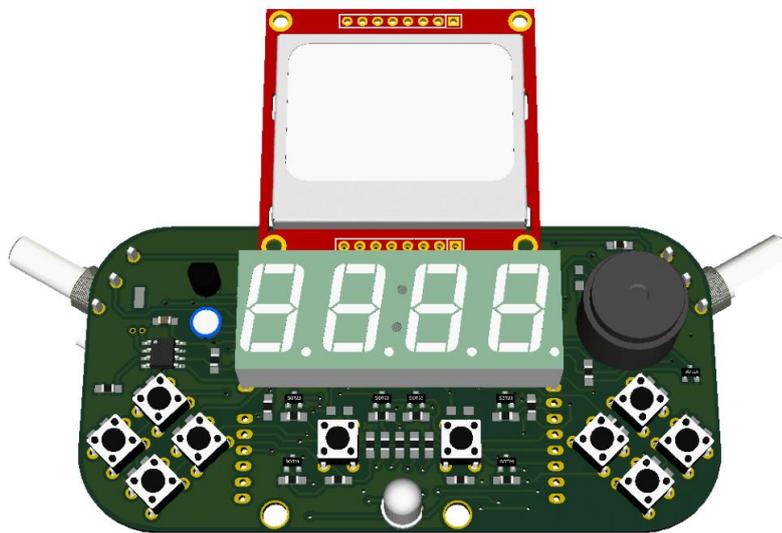
<b>Periférico</b>	<b>Conceitos explorados</b>
LCD 16X2 (compatível com HD44780)	<ul style="list-style-type: none"><li>● Saídas digitais</li><li>● Protocolos de comunicação</li></ul>
Display de 7 segmentos	<ul style="list-style-type: none"><li>● Saídas digitais</li><li>● Multiplexação temporal</li></ul>
RTC DS1307	<ul style="list-style-type: none"><li>● Comunicação serial I2C</li><li>● Armazenamento de dados remotos</li><li>● Operação com relógios</li></ul>
Teclado matricial	<ul style="list-style-type: none"><li>● Entradas digitais</li><li>● Varredura</li><li>● <i>Debounce</i> de teclas</li></ul>
Potenciômetro/LDR/LM35	<ul style="list-style-type: none"><li>● Elementos sensores diversos</li><li>● Conversores analógico-digital</li></ul>
Led RGB	<ul style="list-style-type: none"><li>● Saídas digitais</li><li>● Composição de bits</li><li>● PWM</li></ul>
Buzzer	<ul style="list-style-type: none"><li>● Emissão de sons</li><li>● Controle de frequência</li><li>● PWM</li></ul>
Periféricos internos	<ul style="list-style-type: none"><li>● Timer</li><li>● Conversores analógico-digital</li><li>● Interrupção</li><li>● <i>Watchdog</i></li></ul>

Fonte: <https://www.catarse.me/pqdb>

Figura 1 - Esquemático e imagem 3D da PQDB



(a)



(b)

Fonte: <https://github.com/projetopqdb>

A PQDB vem para contribuir com o movimento open hardware de modo que todo o esquemático, documentação e instruções para que toda pessoa possa analisar, modificar e fabricar por iniciativa própria está disponível na plataforma de compartilhamento de código

GitHub (ALMEIDA, 2017). Foi desenvolvida para ser uma alternativa de baixo custo, possuindo poucos componentes e sendo fabricável com placas de circuito impresso de face simples. Possui um livro base desenvolvido e publicado (ALMEIDA, MORAES e SERAPHIM, 2016) além de códigos e bibliotecas utilizando diversas plataformas de desenvolvimento disponíveis no GitHub em (ALMEIDA, 2017) e (SILVA, 2018).

### 3. Metodologia

Como estudo de caso para apresentar a facilidade de utilização da placa como plataforma de ensino, vamos apresentar o procedimento para a portabilidade de um sistema operacional e o desenvolvimento das bibliotecas de software utilizando a PQDB. Para isso foram criadas quatro atividades. A primeira não faz uso do sistema operacional e serve como atividade de ambientação para a plataforma desenvolvida. A segunda introduz o conceito de sistema operacional. Na terceira são adicionados os periféricos do teclado e da comunicação serial. Por fim, quarta atividade foi realizada

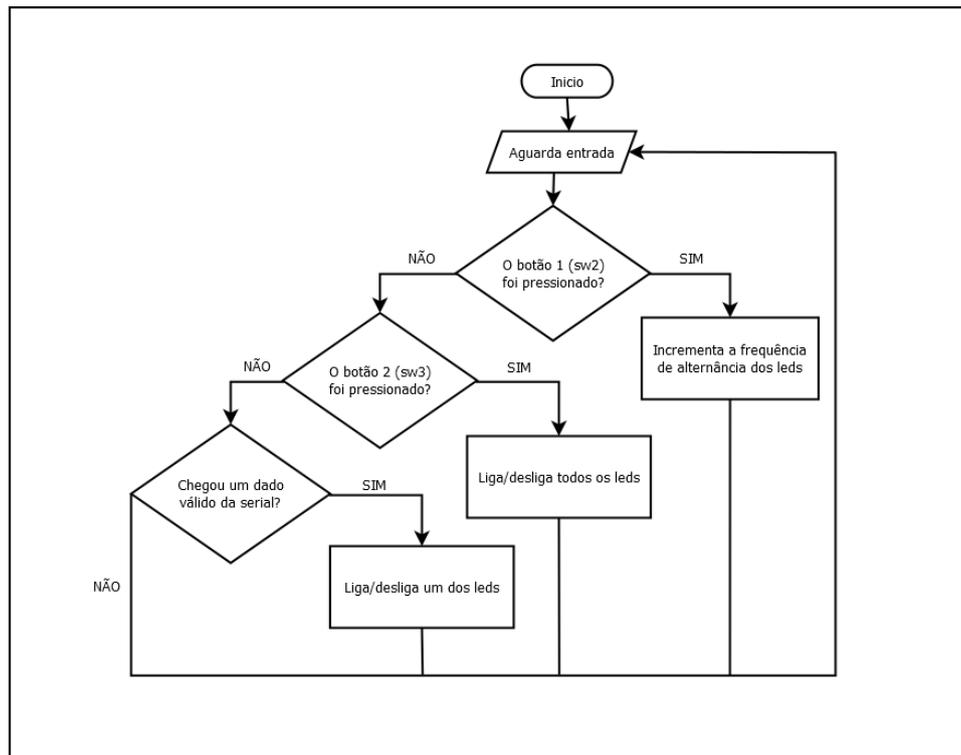
A primeira atividade se inicia com o aprendizado da IDE e a familiarização com a SDK escolhida. A escolha da SDK foi baseada no suporte à plataforma de desenvolvimento e à última versão do FreeRTOS. Dessa maneira, são utilizados a KSDK 2.0, a Kinetis Design Studio 3.0 e a plataforma de desenvolvimento FRDM-K64F.

A primeira atividade desenvolvida foi planejada para ser o mais simples possível, a fim de obter maior familiarização com as ferramentas utilizadas na pesquisa. Consistiu em um software bare-metal, sem utilização de um sistema operacional, que acessa as interfaces de I/O digitais para ligar e desligar o led RGB com um tempo de delay pré-definido.

Na segunda atividade se inicia a ambientação com o FreeRTOS na placa de desenvolvimento utilizada. Com um simples trocar de estado das saídas digitais implementadas em bare-metal o próximo passo é implementar a mesma aplicação utilizando o sistema operacional de tempo real. Como a atividade se baseia somente no piscar do led RGB é necessária a criação de somente uma tarefa com prioridade máxima, com isso não há problemas de escalonamento de processos.

Na terceira atividade o aluno utiliza os periféricos disponíveis na plataforma de desenvolvimento através do sistema operacional. Dessa maneira utiliza-se a serial, os botões e o led RGB disponíveis na placa. A interface de comunicação serial utilizada é a UART a fim de se comunicar com o computador. A Figura 2 ilustra o funcionamento do programa.

Figura 2 - Diagrama de fluxo da aplicação



A etapa de início implementa todas as configurações básicas da placa, habilitação do clock às portas, configuração para atuar como GPIO e configuração da direção dos pinos (entrada ou saída). A comunicação serial UART deve ser estabelecida com um baud rate (taxa de transmissão de dados) de 115200 bps. Um protocolo básico de comunicação é implementado com a serial de modo que quando um caractere ‘r’, ‘g’ ou ‘b’ for recebido o sistema execute a ação de mudança de estado dos leds vermelho, verde e azul, respectivamente.

Na quarta atividade (LCD + ADC) o aluno realiza a comunicação com o display LCD. Primeiro é necessário conseguir enviar uma string ao display e vê-la impressa, para isso é necessário inicializar o dispositivo LCD, enviar um comando indicando em que posição de linha e coluna o texto será escrito, e em seguida enviar o dado de interesse de acordo com seu tipo (caractere, string ou um número).

Assim realizada a comunicação com o LCD, é acessado o conversor AD para fazer as leituras dos dispositivos analógicos e imprimir o resultado no display. Para isso é necessário inicializar o conversor AD e em seguida fazer a leitura baseada no canal de interesse. Cada canal do conversor se relaciona a um dispositivo, o canal 0 corresponde ao sensor de temperatura, o canal 1 ao sensor de luminosidade e o canal 2 ao potenciômetro, cuja ordem

descrita define a ordem que deve ser seguida no desenvolvimento da atividade.

Na quinta atividade (7 seg) se realiza a comunicação com o display de 7 segmentos, de modo que antes de tudo seja feita a sua devida inicialização. Os quatro displays conectados à placa estão multiplexados, dessa maneira cada um deles é ligado e desligado periodicamente de maneira que esta operação não seja perceptível a olho nu.

Em seguida é necessário enviar o dígito escolhido e a posição que ele deve se localizar entre os displays. Por fim, é necessário fazer a atualização contínua dos displays de modo que se realize a alternância entre eles produzindo o efeito desejado ao final.

Como sexta atividade foi desenvolvido um workshop para ensinar os conceitos de programação embarcada utilizando a placa, para verificar a facilidade de ensino com a ferramenta. Para garantir um efeito lúdico, as palestras foram chamadas “como fazer seu próprio game boy”. Esse workshop foi apresentado em dois eventos, na CPBR11 (MOREIRA, 2018, Como Fazer Seu Próprio Gameboy – CPBR11) e na 4ª SpACE (Semana para Computação Automação e Eletrônica) (MOREIRA, 2018, Como Fazer Seu Próprio Gameboy – SpACE UNIFEI) realizada na Universidade Federal de Itajubá. Em cada evento foram disponibilizadas 20 placas e atendido um grupo entre 30 e 40 pessoas.

Durante os treinamentos os participantes terminaram todas as atividades propostas. A maioria dos alunos requisitou auxílio dos instrutores pelo menos uma vez, geralmente para solucionar dúvidas com relação a atividades de programação. Não houveram problemas quanto ao funcionamento da plataforma PQDB.

#### 4. Resultados

Em termos técnicos um problema recorrente em ambientes embarcados é o alto consumo de memória. Para isso foi realizado um conjunto de testes para saber se as atividades planejadas estão de acordo com a plataforma escolhida. A Tabela 2 indica o tamanho que cada código ocupou na memória.

Tabela 2 – Quantidade de memória utilizada

	text	data	bss	dec
1ª atividade (led em <i>bare metal</i> )	13580	120	2448	16148
2ª atividade (led com FreeRTOS)	23656	124	13196	36976
3ª atividade (led + uart + teclado + FreeRTOS)	25236	124	13196	38556

O campo text representa o que termina especificamente na FLASH e possui o código e os dados constantes. O campo data possui somente os dados de inicialização. O campo bss contém os dados não inicializados, e geralmente se mantém na RAM. O campo dec é a soma dos três valores mencionados em decimal.

É possível perceber que para resultar no mesmo objetivo a utilização resultou em um aumento da utilização da FLASH, e em um aumento da utilização da RAM. Esse aumento considerável decorre dos diversos módulos implementados pelo sistema operacional, desde o algoritmo de escalonamento, até o gerenciador da fila, das prioridades e dos códigos de troca de contexto.

Contudo, é possível perceber que mesmo com a inserção de mais tarefas no sistema o aumento de ocupação de memória FLASH não é tão grande e que não há acréscimo de uso da RAM. Isto se deve ao fato de que os módulos do sistema operacional, que influenciam significativamente na quantidade de memória utilizada, já estão implementados e o que é inserido não apresenta muito gasto da FLASH.

Para comprovar o funcionamento das bibliotecas foram desenvolvidas as atividades 4 e 5, ambas utilizando do sistema operacional FreeRTOS. A Figura 3 representa o resultado da atividade 4 onde foi feita a leitura da temperatura, da luminosidade e do potenciômetro, respectivamente, no conversor AD. A Figura 4 representa o resultado da atividade 5 a qual faz a utilização do led RGB e do display de 7 segmentos.

Figura 3 - Resultado da atividade 4



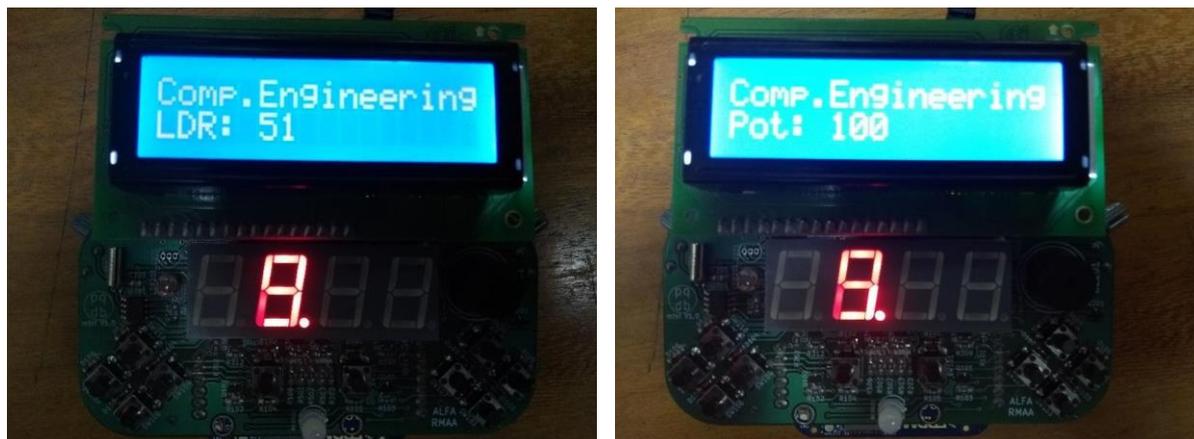
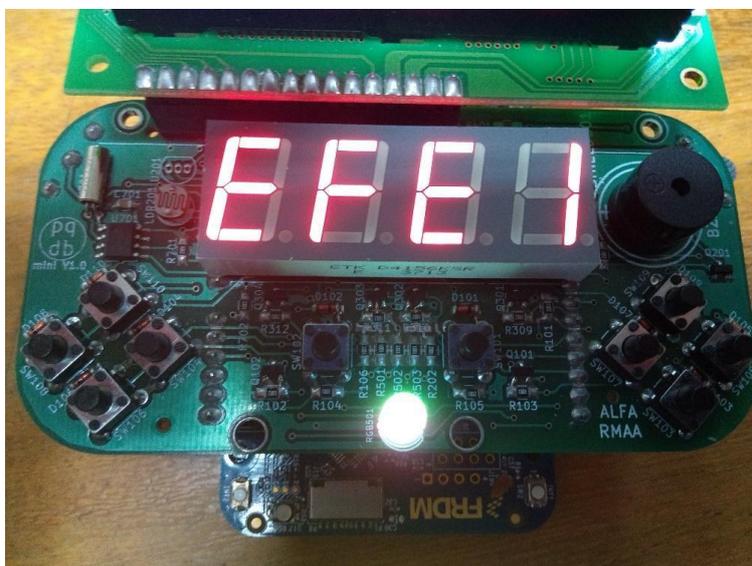


Figura 4 - Resultado da atividade 5



## 5. Conclusão

A placa PQDB oferece diversos recursos que, se bem explorados, garantem um ótimo início na área de desenvolvimento de softwares embarcados. Todos os periféricos estão conectados como de acordo com a pinagem do Arduino e são facilmente acessados.

Mesmo fazendo uso do FreeRTOS o processador escolhido conseguiu atender todas as demandas. A maior dificuldade encontrada foi no uso do microcontrolador, já que a placa PQDB simplificou o acesso os periféricos. As placas foram bem utilizadas nos workshops e permitiram que todos os integrantes finalizassem suas atividades com sucesso e tivessem o conhecimento para ingressar na área de programação embarcada.

Os próximos passos são continuar o desenvolvimento de bibliotecas integrando o FreeRTOS e a KSDK 2.0 na plataforma FRDM-K64F em conjunto com a PQDB, assim como oferecer suporte a outros sistemas microcontrolados.

## Referências

Almeida, R. M. A., Moraes, C. H. V., & Seraphim, T. F. P. (2016). *Programação de sistemas embarcados: Desenvolvendo software para microcontroladores em linguagem C*. Elsevier.

Almeida, R. M. A. (2017). Projetoqdb - Overview. Retrieved November, 2018, from <https://github.com/projetopqdb>

Silva, F. R. (2018). FRDM-K64F\_PQDB\_FreeRTOS\_2017\_2018. Retrieved November, 2018, from [https://github.com/fabiorocha22/FRDM-K64F\\_PQDB\\_FreeRTOS\\_2017\\_2018](https://github.com/fabiorocha22/FRDM-K64F_PQDB_FreeRTOS_2017_2018)

Moreira, R. (2018). Como Fazer Seu Próprio Gameboy – CPBR11. Retrieved November, 2018, from <https://pt.slideshare.net/rafaelmmoreira/como-fazer-seu-prprio-gameboy-cpbr11>

Moreira, R. (2018). Como Fazer Seu Próprio Gameboy - SpACE UNIFEI. Retrieved from <https://www.slideshare.net/rafaelmmoreira/como-fazer-seu-prprio-gameboy-space-unifei>

## Porcentagem de contribuição de cada autor no manuscrito

Fábio Rocha da Silva - 40%

Augusto Cesar Rossi de Mattos - 15%

Gabriel Silva Marcatto - 15%

Rafael de Moura Moreira - 15%

Rodrigo Maximiano Antunes de Almeida – 15%