

Uma Plataforma de Baixo Consumo de  
Energia, Flexível e Barata para a Internet das  
Coisas

PROPONENTE:

Francisco Figueiredo Goytacaz Sant'Anna  
francisco@ime.uerj.br

INSTITUIÇÃO DE EXECUÇÃO:

Departamento de Informática e Ciências da Computação  
Instituto de Matemática e Estatística (IME)  
Universidade Estadual do Rio de Janeiro (UERJ)

Chamada Universal MCTIC/CNPq 2018

18 de agosto de 2018

# 1 Identificação do Projeto

## 1.1 Título

Uma Plataforma de Baixo Consumo de Energia, Flexível e Barata para a Internet das Coisas

## 1.2 Palavras-Chave

internet das coisas, eficiência energética, linguagem síncrona

## 1.3 Resumo

De acordo com a Agência Internacional de Energia (AIE) [2], o número de dispositivos conectados deve atingir 50 bilhões até 2020 com a expansão da Internet das Coisas (IoT). A maior parte do consumo de energia nesses dispositivos será em *modo de espera* (aka *standby mode*), quando eles não estão transmitindo ou processando dados. Em particular, o modo de espera é responsável por aproximadamente 10–15% do consumo residencial. Também estima-se que as emissões de  $CO_2$  relacionadas ao modo de espera mundiais seja equivalente às de 1 milhão de carros.

Os efeitos alarmantes do consumo em modo de espera, aliados ao crescimento estimado da IoT, tornou o modo de espera para dispositivos conectados um dos seis pilares do *Plano de Ação para Eficiência Energética do G20*<sup>1</sup>. No entanto, o uso efetivo do modo de espera requer grandes esforços de software e hardware para detectar períodos de inatividade nos dispositivos, identificar periféricos que devem permanecer ligados, e aplicar os modos mais econômicos sempre que possível.

Este projeto tem como principal objetivo desenvolver uma plataforma de hardware e software de baixo consumo de energia para pesquisa e educação em Internet das Coisas.

No que diz respeito ao software, iremos adotar a linguagem de programação reativa CÉU [3], a qual estamos desenvolvendo durante os últimos 8 anos, e que tem como alvo sistemas embarcados restritos. CÉU é baseada no modelo de concorrência síncrono, que troca poder por confiabilidade e possui um modelo de tempo mais simples que cobre a maioria dos requisitos de aplicações IoT. Nesse modelo, todas as reações ao mundo externo são

---

<sup>1</sup>G20's Energy Efficiency Action Plan: <https://www.iea-4e.org/projects/g20>

computadas em tempo finito, garantindo que as aplicações sempre chegam a um estado ocioso que é suscetível ao modo de espera. A linguagem já possui suporte recente básico a tratamento de interrupções e gerenciamento de energia automático. Em testes preliminares, já alcançamos economias entre 20 e 90% para aplicações escritas puramente em CÉU.

No que diz respeito ao hardware, os requisitos principais são o baixo custo e flexibilidade da plataforma. As soluções completas de hardware atuais são acessíveis mas pouco flexíveis, pois tipicamente possuem componentes SMD montados na placa. Em particular, os módulos de rádio são previamente determinados, criando uma barreira para a pesquisa. Mesmo que muitas dessas soluções sejam abertas, o custo dos componentes é alto para compras em pequenas quantidades e o método de montagem não é adequado para experimentação. A nossa proposta visa projetar uma solução flexível e de baixo custo baseada em microcontroladores e módulos "de platereira" no mercado brasileiro.

## 2 Equipe e Instituições

- Francisco Figueiredo Goytacaz Sant'Anna
  - **Função:**  
Pesquisador Proponente e Coordenador
  - **E-mail:**  
`francisco@ime.uerj.br`
  - **Lattes:**  
<http://buscatextual.cnpq.br/buscatextual/visualizacv.do?id=K4736349P3>
  - **Instituição:**  
Departamento de Informática e Ciências da Computação,  
Instituto de Matemática e Estatística (IME),  
Universidade Estadual do Rio de Janeiro (UERJ)

TODO

## 3 Áreas de Conhecimento

- Área predominante:

- Ciência da Computação — Sistemas de Computação
- Áreas relacionadas:
  - Engenharia de Telecomunicações — Sistemas de Telecomunicações
  - Ciência da Computação — Linguagens de Programação

## 4 Objetivos

## 5 Objetivos Gerais

Criar uma plataforma de hardware e software para pesquisa e educação em Internet das Coisas de baixo consumo de energia, flexível e barata.

## 6 Objetivos Específicos

A plataforma deve alcançar os seguintes objetivos:

**Baixo Consumo de Energia:** O hardware deve possuir modos de economia de energia para todos os seus componentes, sejam eles o microcontrolador, sensores, ou módulos de rádio. O software será baseado em uma linguagem de programação ciente de energia, que seja capaz de detectar quando os componentes estão ociosos para colocá-los em modo de espera automaticamente, sem a intervenção do programador.

**Flexível:** O hardware deve prever conexões para uma variedade de sensores e transceptores de rádio frequência. Em particular, os módulos de rádio mais populares devem ser todos acopláveis externamente ao hardware. O software deve ser modular, de maneira que somente os drivers dos dispositivos de interesse sejam compilados junto com as aplicações.

**Barato:** O hardware deve usar microcontroladores e módulos "de prateleira" que sejam encontrados com facilidade no mercado brasileiro para compras em pequenas quantidades e a custo baixo. O plataforma de software deve ser toda baseada em software livre.

Como principais desafios, o hardware deve possuir mecanismos flexíveis que permitam desabilitar periféricos via software e, principalmente, a linguagem de programação deve oferecer mecanismos automáticos para colocar o hardware em modo de espera, sem esforços extras por parte do programador.

## 7 Metodologia

O projeto possui duas linhas de pesquisa, em *software* e em *hardware*, que se complementam para oferecer uma plataforma íntegra, mas que também são suficientemente independentes para serem desenvolvidas por diferentes equipes em simultâneo.

### 7.1 Software

Grande parte da pesquisa em software está fundamentada em trabalhos anteriores nossos com a linguagem CÉU [3, 4, 8]. A linguagem já possui uma implementação estável e foi adotada com sucesso nas áreas de redes de sensores sem fio [3, 1], jogos [6] e multimídia [7].

Mais recentemente, pesquisamos como prover tratamento de interrupções e gerenciamento automático de modo de espera diretamente na linguagem, já obtendo alguns resultados preliminares no contexto de economia de energia [5]. No entanto, a plataforma de software proposta neste projeto requer uma pesquisa mais extensa, que deve abranger os seguintes requisitos:

1. Avaliação qualitativa da usabilidade de CÉU.
2. Avaliação quantitativa do uso de recursos de CÉU, tais como memória, desempenho e, principalmente, consumo de energia.
3. Desenvolvimento de aplicações representativas de IoT que usem comunicação por rádio extensivamente.
4. Cobertura de drivers para periféricos e módulos de rádio diversos.

O coordenador do projeto ficará responsável pelas partes mais críticas da pesquisa, tais como o funcionamento da linguagem e drivers para os periféricos mais essenciais (item 4).

Uma aluna de mestrado recém ingressa está trabalhando com a avaliação de usabilidade da linguagem (item 1). Ela irá desenvolver um ambiente de

programação educacional que tem como alvo alunos no início da graduação em Ciência da Computação. A ideia é introduzir a área IoT e sistemas embarcados já no início do curso. Nossa hipótese é que o vocabulário especializado de CÉU para lidar com o mundo externo (e.g., eventos e concorrência) irá permitir que os alunos desenvolvam projetos de IoT simples mas completos.

O desenvolvimento de aplicações IoT completas (item 3) é fundamental para validar o projeto como um todo. A Internet oferece diversas aplicações de IoT abertas e disponíveis que poderão servir como base para as avaliações qualitativas e quantitativas (itens 2 e 3). Alunos de iniciação científica e projeto final poderão trabalhar em projetos diferentes que servirão para essas avaliações de CÉU.

## 7.2 Hardware

A nossa proposta visa projetar uma solução de hardware flexível e de baixo custo baseada em microcontroladores e módulos "de platereira" no mercado brasileiro com os seguintes requisitos:

- O microcontrolador deve possuir baixo consumo de energia e oferecer modos de espera programáveis.
- Os módulos de rádio mais populares devem ser acopláveis externamente ao hardware para experimentação de diversas propriedades de RF, tais como técnica de modulação, alcance, velocidade, interoperabilidade, etc.
- O hardware deve possuir mecanismos que permitam desabilitar sensores e periféricos via software para maximizar a economia de energia.

Os microcontroladores em placas compatíveis com a plataforma Arduino são os mais populares e disponíveis no mercado brasileiro. A Figura 1 indica alguns modelos que podem ser avaliados para adoção no projeto (item 1). Como exemplo, o modelo *Pro Mini 3.3V* tem tamanho reduzido, baixo consumo de energia e custa em torno de R\$15,00.

O levantamento, avaliação e montagem dos módulos de rádio é a etapa mais sensível no projeto da plataforma de hardware (item 2). A Figura 2 ilustra a diversidade de implementações de módulos RF disponíveis, alguns com custo inferior a R\$10,00. Idealmente, cada um dos módulos considerados deve ser acoplável e facilmente intercambiáveis na nossa plataforma, dado

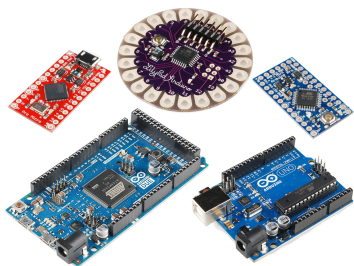


Figura 1: Modelos de Arduino

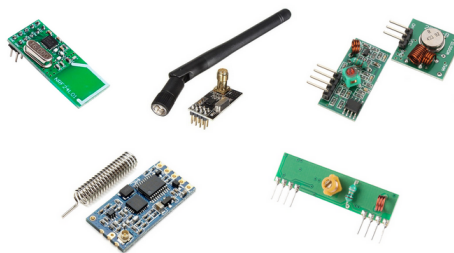


Figura 2: Módulos RF

que temos como objetivo a flexibilidade para experimentação e pesquisa em IoT. Cada módulo poderá ser avaliado e desenvolvido em separado. Para isso, iremos propor a exploração desses módulos em projetos da disciplina de Software Embarcado em nosso programa de pós graduação.

A plataforma também deve permitir o acoplamento de sensores e outros periféricos (item 3). Além disso, deve oferecer mecanismos para desligá-los eletricamente por software de modo a economizar energia. Esses mecanismos podem usar diretamente os pinos do microcontrolador para sensores mais simples ou transistores que funcionarão como chaves eletrônicas. Esse estudo também pode ser realizado em separado por uma outra equipe.

Os componentes eletrônicos devem possuir pinos para montagem *through-hole*, como ilustrado na Figura 7.2, que é a técnica mais flexível e adequada para protótipos considerando o contexto de pesquisa. Esses componentes também costumam ser mais disponíveis no mercado para compras em pequenas quantidades.

É importante destacar que o desenvolvimento do hardware não depende da frente de pesquisa em software em CÉU, e tampouco depende do de-

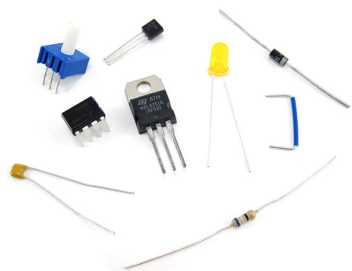


Figura 3: Componentes *Through-Hole*



Figura 4: Componentes *SMD*

envolvimento de softwares complexos. Como iremos utilizar hardware de commodity, já existe oferta suficiente de bibliotecas e aplicações open-source para testes. Apenas para a avaliação de consumo de energia as duas frentes deverão ser necessariamente integradas.

Osciloscópio 3k

Serrapilheira

do modo a permitir.

Inicialmente, não iremos considerar outras tecnologias

IMAGEM: arduinos IMAGEM: RFs baratos. IMAGEM: sensores IMAGEM: smt vs loop-hole IMAGEM: prototipo PCB

//// //

### 7.3 Abstract (in english):

g) Etapas de execução do projeto com respectivo cronograma de atividades; h) Produtos esperados como resultado da execução do projeto, com previsão de cronograma de entregas anuais; i) Potencial de impacto dos resultados do ponto de vista técnico-científico, de inovação, difusão, sócio-econômico e ambiental; j) Colaborações ou parcerias já estabelecidas para a execução do projeto; k) Perspectivas de colaborações interinstitucionais para a execução do projeto; l) Recursos financeiros de outras fontes aprovados para aplicação no projeto; m) Disponibilidade efetiva de infraestrutura e de apoio técnico para o desenvolvimento do projeto; n) Orçamento detalhado.

- Até 36 meses

## Referências

- [1] Adriano Branco et al. Terra: Flexibility and safety in wireless sensor networks. *ACM Trans. Sen. Netw.*, 11(4):59:1–59:27, September 2015.
- [2] OECD/IEA. More data less energy—Making network standby more efficient in billions of connected devices. Technical report, International Energy Agency, 2014.
- [3] Francisco Sant’Anna et al. Safe System-level Concurrency on Resource-Constrained Nodes. In *Proceedings of SenSys’13*. ACM, 2013.



- [4] Francisco Sant’anna et al. The design and implementation of the synchronous language céu. *ACM TECS*, 16(4):98:1–98:26, July 2017.
- [5] Francisco Sant’Anna et al. Wip: Transparent standby for low-power, resource-constrained embedded systems: A programming language-based approach. In *Proceedings of the LCTES’18*, pages 94–98, New York, NY, USA, 2018. ACM.
- [6] Francisco Sant’Anna, Noemi Rodriguez, and Roberto Ierusalimsky. Structured Synchronous Reactive Programming with Céu. In *Proceedings of Modularity’15*, 2015.
- [7] Rodrigo Santos et al. Céu-Media: Local Inter-Media Synchronization Using Céu. In *Proceedings of WebMedia’16*, pages 143–150, New York, NY, USA, 2016. ACM.
- [8] Rodrigo C. M. Santos et al. A Memory-Bounded, Deterministic and Terminating Semantics for the Synchronous Programming Language Céu. In *Proceedings of LCTES’18*, pages 1–18, New York, NY, USA, 2018. ACM.