

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

19 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*¹. 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 [5], 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

¹G20'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
Programa de Pós-Graduação em Engenharia Eletrônica (PEL)
Universidade Estadual do Rio de Janeiro (UERJ)

TODO

3 Áreas de Conhecimento

- Área predominante:

- Ciência da Computação — Sistemas de Computação
- Áreas relacionadas:
 - 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.

Barata: O hardware deve usar microcontroladores e módulos “de platereira” 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 [5, 3, 7]. A linguagem já possui uma implementação estável e foi adotada com sucesso nas áreas de redes de sensores sem fio [5, 1], jogos [4] e multimídia [6].

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 [8]. 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).

A mestranda colaboradora está trabalhando com a avaliação de usabilidade da linguagem (item 1). Ela vem trabalhando em 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.

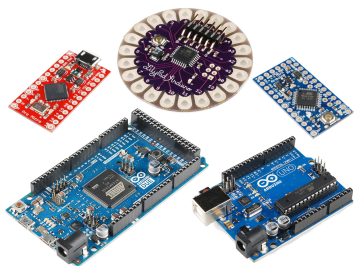


Figura 1: Modelos de Arduino

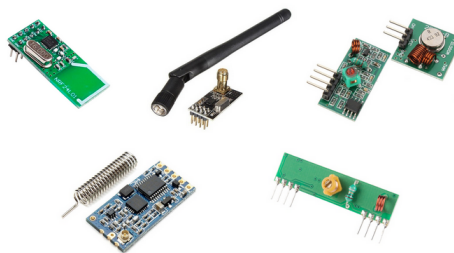


Figura 2: Módulos RF

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:

1. O microcontrolador deve possuir baixo consumo de energia e oferecer modos de espera programáveis.
2. 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.
3. 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.



Figura 3: Componentes *Through-Hole* e *Surface-Mount*.

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 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.

O levantamento dos microcontroladores e módulos de RF, a avaliação das propriedades de interesse e a prototipagem de um hardware flexível aos diversos módulos devem ser tema de uma dissertação de Mestrado.

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 3, que é a técnica mais flexível e adequada para protótipos considerando o contexto de pesquisa. Esses componentes também são mais disponíveis no mercado para compras em pequenas quantidades. Ao fim do projeto, desejamos obter um protótipo de tamanho similar aos da Figura 4.

É importante destacar que o desenvolvimento do hardware não depende da frente de pesquisa de software em CÉU, e tampouco depende do desenvolvimento de softwares complexos. Como iremos utilizar hardware de commodity, já existe oferta suficiente de bibliotecas e aplicações open-source para

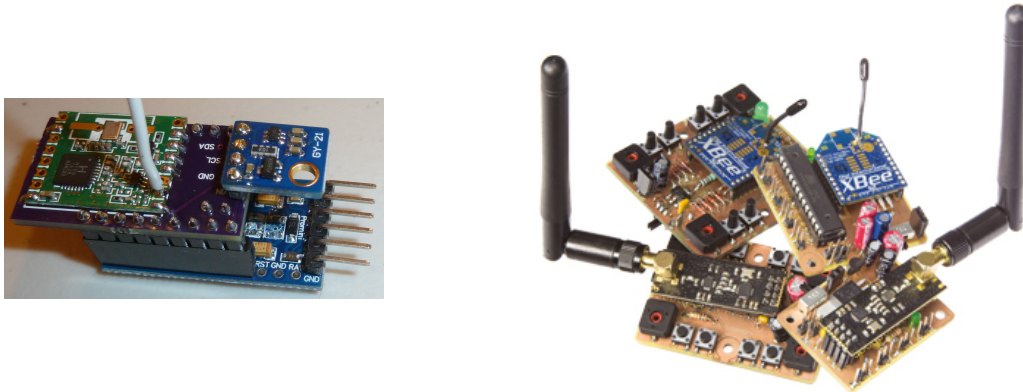


Figura 4: Protótipos de dispositivos IoT.

testes. Apenas para a avaliação de consumo de energia, as duas frentes deverão ser necessariamente integradas.

8 Cronograma de Atividades

O projeto deve se estender por 2 anos (8 trimestres) conforme retratado na Tabela 5. O projeto é dividido 4 em fases:

Fase 0: Essa fase compreende o trabalho de infra-estrutura da linguagem CÉU. Inclui atividades no início do projeto relativas ao suporte recente de interrupções e gerenciamento de energia a ser finalizado. Também inclui atividades eventuais de manutenção da linguagem durante todo o projeto, mas que não devem ser desconsideradas.

Fase I: Essa fase compreende o levantamento de módulos de RF a serem considerados e que em seguida farão parte de plataforma de hardware. O protótipo já deve acomodar os diferentes módulos de RF para testes de campo. Por fim, faremos uma avaliação quantitativa dos módulos considerando diversas propriedades, tais como alcance, velocidade e consumo de energia em modo ativo. Essa avaliação usará *microbenchmarks* para as diversas propriedades e não depende da linguagem CÉU.

Fase II: Essa fase inclui o desenvolvimento em CÉU de drivers para os rádios e aplicações completas de IoT, que devem ser eficientes em termos

de consumo de energia no modo de espera. Faremos uma avaliação quantitativa do uso de recursos de para as aplicações desenvolvidas. Em particular, esperamos economias de energia significativas pelo gerenciamento de energia automático de CÉU.

Fase III: Essa fase complementa a Fase I com outros sensores, periféricos, mecanismos de economia de energia e também ajustes e acabamentos do protótipo. Ao fim do projeto, esperamos ter uma solução completa de IoT de hardware e software para um artigo de maior impacto.

Fase IV: Essa fase é relativamente independente das demais e deve se estender por todo o período do projeto. O objetivo dela é avaliar qualitativamente a usabilidade de CÉU com alunos de graduação. Ela também vai servir para atrair novos alunos para a área de sistemas embarcados e IoT, construir uma comunidade em torno de CÉU, e avaliar o nosso protótipo de hardware em momentos tardios do projeto.

9 Resultados Esperados

Ao fim de cada período indicado, esperamos ter os seguintes resultados:

- Primeiro Semestre:
 1. Uma versão de CÉU com suporte a tratamento de interrupções e gerenciamento de energia automático.
 2. Uma disciplina opcional de IoT preparada para alunos no início da graduação usando o ambiente de programação de CÉU.
- Segundo Semestre:
 1. Um protótipo de hardware com suporte a diversos módulos de RF.
 2. Um artigo com o levantamento e avaliação de módulos de RF com foco em *microbenchmarks* para consumo de energia.
 3. Um conjunto de drivers em CÉU para módulos de RF cientes de energia.
- Terceiro Semestre:

		Atividade \ Trimestre	1	2	3	4	5	6	7	8
Fase 0	Céu	Manutenção da Linguagem	X	X	X	X	X	X	X	X
	Céu	Suporte para Tratamento de Interrupções e Gerenciamento de Energia	X	X						
Fase I	HW	Levantamento de Módulos de RF	X	X						
	HW	Protótipo com RF		X	X	X				
	ART	Avaliação Quantitativa de Módulos de RF			X	X				
Fase II	SW	Desenvolvimento de Drivers RF		X	X	X				
	SW	Desenvolvimento de Aplicações IoT		X	X	X	X			
	ART	Avaliação Quantitativa de Céu para IoT				X	X			
Fase III	HW	Protótipo com RF e Sensores					X	X	X	X
	SW	Desenvolvimento de Drivers					X	X	X	X
	ART	Solução HW/SW completa para IoT							X	X
Fase IV	Céu	Ambiente Educacional para IoT	X	X	X	X	X	X	X	X
	ART	Avaliação Qualitativa de Céu para IoT							X	X

Frentes de Trabalho	Céu	Linguagem Céu
	HW	Hardware
	SW	Software
	ART	Artigo Científico

Figura 5: Cronograma de Atividades

- Um artigo com a avaliação quantitativa de CÉU considerando aplicações completas e representativas de IoT.
 - Um protótipo de hardware completo com módulos de RF, sensores, etc.
- Quarto Semestre:
 - Um conjunto de drivers em CÉU para sensores e outros periféricos.
 - Um artigo com a solução HW/SW completa da nossa plataforma de IoT.
 - Um artigo com a avaliação qualitativa de usabilidade de CÉU considerando todo o software desenvolvido durante o projeto e também a nossa experiência com alunos de graduação.

10 Potencial de Impacto

CÉU é uma linguagem que vem sendo desenvolvida pelos últimos 8 anos, desde o início do doutorado do proponente. Além das nossas próprias contribuições, a linguagem já foi utilizada por terceiros na academia [9] Nos últimos anos, ela vem amadurecendo e sendo usada por outros grupos de

- ceu linguagem 8 anos, usos em outros programas, e industria, ferramenta de difusao - educacao em IoT - hardware flexivel aberto para outras universidade - economia de energia

TODO: - Osciloscópio 3k - Serrapilheira - Falar mais sobre ferramenta da Anny, GSoC

10.1 Abstract (in english):

- 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, Francisco Sant’anna, Roberto Ierusalimschy, Noemi Rodriguez, and Silvana Rossetto. 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, Roberto Ierusalimschy, Noemi Rodriguez, Silvana Rossetto, and Adriano Branco. The design and implementation of the synchronous language céu. *ACM TECS*, 16(4):98:1–98:26, July 2017.
- [4] Francisco Sant’Anna, Noemi Rodriguez, and Roberto Ierusalimschy. Structured Synchronous Reactive Programming with Céu. In *Proceedings of Modularity’15*, 2015.

- [5] Francisco Sant’Anna, Noemi Rodriguez, Roberto Ierusalimschy, Olaf Landsiedel, and Philippas Tsigas. Safe System-level Concurrency on Resource-Constrained Nodes. In *Proceedings of SenSys’13*. ACM, 2013.
- [6] Rodrigo Santos, Guilherme Lima, Francisco Sant’Anna, and Noemi Rodriguez. Céu-Media: Local Inter-Media Synchronization Using Céu. In *Proceedings of WebMedia’16*, pages 143–150, New York, NY, USA, 2016. ACM.
- [7] Rodrigo C. M. Santos, Guilherme F. Lima, Francisco Sant’Anna, Roberto Ierusalimschy, and Edward H. Haeusler. 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.
- [8] Alexandre Sztajnberg, Ana Lúcia de Moura, and Noemi Rodrigues. 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.
- [9] Matthias Terber. Function-oriented decomposition for reactive embedded software. In *43rd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2017, Vienna, Austria, August 30 - Sept. 1, 2017*, pages 288–295, 2017.