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

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

INSTITUIÇÃO DE EXECUÇÃO:
Universidade Estadual do Rio de Janeiro (UERJ)

Chamada Universal MCTIC/CNPq 2018

12 de setembro de 2018

1 Identificação do Projeto

1.1 Título

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

1.2 Palavras-Chave

internet das coisas, eficiência energética, linguagem de programação síncrona

1.3 Resumo

De acordo com a Agência Internacional de Energia (AIE) [2], o número de dispositivos conectados em rede 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 ativamente. Atualmente, o modo de espera é responsável por 10 a 15% do consumo residencial. Também estima-se que as emissões globais de CO_2 relacionadas ao modo de espera seja equivalente às emissões de 1 milhão de carros.

Os efeitos do consumo de energia sobre o meio ambiente, aliados ao grande crescimento da IoT no curto prazo, 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 hardware e software para detectar períodos de inatividade nos dispositivos, identificar periféricos que devem (ainda) permanecer ligados, e aplicar os modos de energia 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 e a IoT. Céu é baseada no modelo de concorrência síncrono, no qual todas as reações ao mundo externo são computadas em tempo finito, garantindo que as aplicações sempre cheguem a um estado ocioso suscetível ao modo de espera. A linguagem já

[1] G20's Energy Efficiency Action Plan: `https://www.iea-4e.org/projects/g20`

possui suporte recente ao gerenciamento automático de energia. Em testes preliminares, alcançamos economias entre 20 e 90% para aplicações escritas puramente em CÉU.

No que diz respeito ao hardware, iremos adotar microcontroladores e transceptores de rádio de baixo consumo de energia. Além disso, buscamos como requisitos extras o baixo custo e flexibilidade da plataforma para maior adequação ao contexto de pesquisa e educação. As soluções completas de hardware para IoT atuais são pouco flexíveis, pois tipicamente possuem componentes SMD já soldados às placas. Em particular, os módulos de rádio são pré-determinados, criando uma barreira para a experimentação. Mesmo havendo soluções abertas, o método de montagem industrial tipicamente adotado não é adequado para pesquisa e educação. A nossa proposta visa projetar uma solução flexível e de baixo custo baseada em microcontroladores e módulos "de platereira" do mercado brasileiro.

2 Equipe e Instituições

- Francisco Figueiredo Goytacaz Sant'Anna
  - **Função:**
    Pesquisador Proponente e Coordenador
  - **E-mail:**
    `francisco@ime.uerj.br`
  - **Lattes:**
    `http://lattes.cnpq.br/0077491494754494`
  - **Instituição:**
    Departamento de Informática e Ciências da Computação
    Programa de Pós-Graduação em Engenharia Eletrônica (PEL)
    Universidade Estadual do Rio de Janeiro (UERJ)
- Alexandre Sztajnberg
  - **Função:**
    Pesquisador Colaborador
  - **E-mail:**
    `alexszt@ime.uerj.br`

- **Lattes:**
  `http://lattes.cnpq.br/0403732822984772`
- **Instituição:**
  Departamento de Informática e Ciências da Computação
  Programa de Pós-Graduação em Engenharia Eletrônica (PEL)
  Universidade Estadual do Rio de Janeiro (UERJ)

3 Áreas de Conhecimento

- Área predominante:
  - Ciência da Computação — Sistemas de Computação
- Áreas relacionadas:
  - Ciência da Computação — Linguagens de Programação
  - Outra — Microeletrônica — Projeto

4 Objetivos

4.1 Objetivos Gerais

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

4.2 Objetivos Específicos

A plataforma deve alcançar os seguintes objetivos:

Baixo Consumo Energético: 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. A linguagem deve ser 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" comuns no mercado para compras em pequenas quantidades e a custo baixo. O plataforma de software deve ser *open source*.

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

5 Metodologia

O projeto possui duas linhas concomitantes de pesquisa—em *software* e em *hardware*—que se complementam para oferecer uma plataforma completa, mas que também são suficientemente independentes para serem desenvolvidas o máximo em separado.

5.1 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, investigamos como prover tratamento de interrupções e gerenciamento automático do modo de espera diretamente em Céu, já obtendo resultados preliminares com baixo consumo de energia [8]. No entanto, a plataforma de software proposta neste projeto requer uma pesquisa mais extensa, que ainda deve abranger os seguintes temas:

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 dos drivers para os periféricos mais essenciais (item 4).

Uma aluna de mestrado está trabalhando em um ambiente de programação educacional para IoT que tem como alvo alunos no início da graduação em Ciência da Computação. 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. Essa frente de pesquisa será importante para a avaliação de usabilidade da linguagem (item 1).

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.

5.2 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.

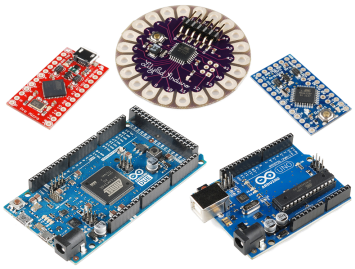

Figura 1: Modelos de Arduino

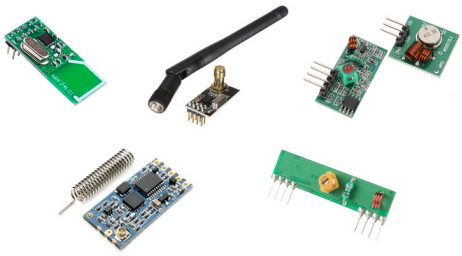

Figura 2: Módulos RF

Os microcontroladores de 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 módulos RF disponíveis, alguns com custo inferior a R$10,00. Idealmente, cada um dos módulos considerados deve ser acoplável e facilmente intercambiável 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.

A plataforma também deve permitir o acoplamento de sensores e outros periféricos (item 3). Além disso, deve oferecer mecanismos para desligá-los 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 pode ser realizado em separado da avaliação dos módulos de rádio (item 2).

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

Figura 3: Componentes *Through-Hole* (esquerda) e *Surface-Mount* (direita).

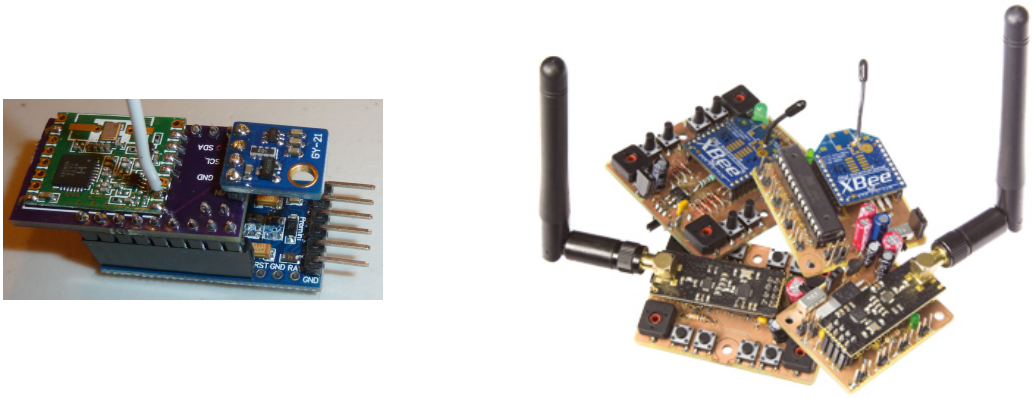

Figura 4: Protótipos de dispositivos IoT.

da frente de pesquisa de software em Céu, e tampouco depende do desenvolvimento de softwares complexos (mesmo que em outras linguagens). Como iremos utilizar hardware de commodity, já existe oferta suficiente de bibliotecas de software e aplicações open-source para testes. Apenas para a avaliação de consumo de energia que as duas frentes deverão ser necessariamente integradas.

6 Cronograma de Atividades

O projeto deve se estender por 2 anos (8 trimestres) conforme retratado na Figura 5. O projeto é dividido em 5 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 serem finalizados. Também inclui atividades eventuais de manutenção da linguagem durante todo o projeto.

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 primeiro protótipo de hardware 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 (i.e., sem modo de espera). Essa avaliação usará *microbenchmarks* para as diversas propriedades e não depende da linguagem Céu pois usará bibliotecas já disponíveis em C.

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 computacionais para as aplicações desenvolvidas. Esperamos obter economias de energia significativas pelo gerenciamento de energia automático de Céu.

Fase III: Essa fase complementa as Fases I e II com outros sensores, periféricos, mecanismos de economia de energia e também ajustes e acabamentos do protótipo. Ao fim do projeto, esperamos ter uma solução

Atividade \ Trimestre			1	2	3	4	5	6	7	8
Fase 0	SW	Manutenção da Linguagem Céu	X	X	X	X	X	X	X	X
	SW	Suporte para Tratamento de Interrupções e Gerenciamento de Energia em Céu	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 em Céu		X	X	X				
	SW	Desenvolvimento de Aplicações IoT em Céu		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 em Céu					X	X	X	X
	ART	Solução HW/SW completa para IoT							X	X
Fase IV	SW	Ambiente Educacional para IoT	X	X	X	X	X	X	X	X
	ART	Avaliação Qualitativa de Céu para IoT							X	X

Frentes de Trabalho		
	HW	Hardware
	SW	Software
	ART	Artigo Científico

Figura 5: Cronograma de Atividades

completa de IoT de hardware e software para um artigo de maior impacto.

Fase IV: Essa fase é independente das demais e deve se estender por todo o período do projeto. O objetivo 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.

7 Resultados Esperados

Ao fim de cada semestre, esperamos obter 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:

1. Um artigo com a avaliação quantitativa de Céu considerando aplicações completas e representativas de IoT.
2. Um protótipo de hardware completo com módulos de RF, sensores, etc.

- Quarto Semestre:

1. Um conjunto de drivers em Céu para sensores e outros periféricos.
2. Um artigo com a solução HW/SW completa da nossa plataforma de IoT.
3. 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.

8 Potencial de Impacto

Como destacado no resumo do projeto, o consumo de energia estimado com o crescimento da IoT terá consequências significativas para o meio ambiente. A nossa proposta para tratamento automático do modo de espera no nível da linguagem de programação pode ser solução uma escalável para esse problema. Potencialmente, todas as aplicações escritas em Céu se beneficiariam desse mecanismo automático, sem esforços extras de programação.

Céu é uma linguagem que vem sendo desenvolvida pelos últimos 8 anos, desde o início do doutorado do pesquisador proponente. Além das nossas próprias contribuições científicas, Céu já foi utilizada como base para a pesquisa de terceiros [9] e também para o desenvolvimento de produtos na indústria de sistemas embarcados. Céu oferece um modelo de concorrência antagônico ao de linguagens mais tradicionais (incluindo linguagens acadêmicas). Isso concede a ela algumas frentes de inovação, como por exemplo o gerenciamento automático de energia.

A nossa proposta é uma plataforma aberta tanto de software como de hardware que pode ser reusada por outros grupos de pesquisa. Propomos uma solução flexível para atender cenários diversos e também de baixo custo para estimular uma maior adoção.

Um objetivo secundário do nosso trabalho é criar um ambiente de programação para educação em IoT. Esse ambiente será usado em um curso introdutório de IoT para alunos no início da graduação. Consideramos importante fomentar o interesse de alunos nesse tema o mais cedo possível.

9 Orçamento Detalhado

A Figura 6 detalha o orçamento por semestre para o período total de 2 anos do projeto. O orçamento total é de R$27.420,00 divididos em recursos de capital (R$11.500,00) e custeio (R$15.920,00).

Os recursos de capital se concentram no primeiro semestre, com a compra de um notebook, monitor e osciloscópio. Além disso, adquiriremos os componentes para o desenvolvimento da plataforma de hardware aos poucos, ao longo dos 4 semestres.

Os recursos de custeio se concentram em viagens para eventos e congressos de IoT e sistemas embarcados. Esperamos ter material para até 4 artigos científicos e deejamos publicar alguns deles em conferências para maior divulgação. Dentre as possíveis conferências, podemos destacar o *Simpósio Brasileiro de Redes de Computadores (SBRC)*, *Brazilian Symposium on Computing Systems Engineering (SBESC)* e o *ACM Conference on Embedded Networked Sensor Systems*. Também reservamos recursos ao longo dos 4 semestres para a confecção de placas de circuito impresso para a plataforma de hardware.

Item \ Semestre (Custo em R$)		1	2	3	4	Total
Capital	Notebook i7	5000	-	-	-	11500
	Monitor 24"	1000	-	-	-	
	Arduinos, Sensores e Módulos de Rádio	1000	500	500	500	
	Osciloscópio Digital	3000	-	-	-	
Custeio	Passagens Nacionais	-	1000	1000	-	15920
	Diárias Nacionais	-	960	960	-	
	Passagens Internacionais	-	-	-	3000	
	Diárias Internacionais	-	-	-	6000	
	Confecção de Placas PCB	500	500	1000	1000	
		10500	2960	3460	10500	**27420**

Figura 6: Orçamento Detalhado

10 Recursos de Outras Fontes

O pesquisador proponente é atualmente coordenador do projeto intitulado *Energy Efficiency for IoT Software in the Large* financiado pelo Instituto Serrapilheira em sua 1a chamada pública[2]. O projeto conta com financiamento de R$70.000 e tem previsão de término para fevereiro de 2019. O escopo do projeto se concentra em desenvolver novos mecanismos de eficiência energética no nível de linguagem de programação. Os recursos financeiros estão sendo usados principalmente para pagamento de bolsas e viagens.

O novo projeto proposto neste documento é uma plataforma completa de hardware e software que já considera os resultados obtidos no projeto em andamento, conforme descrito no início da Seção 5.1.

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] `https://serrapilheira.org/chamada-publica-no1/`

[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.