

Monitoramento de Consumo Doméstico de Água Utilizando uma Meta-Plataforma de IoT

Home monitoring of water consumption using an IoT meta-platform

Everton Dornelas ¹  orcid.org/0000-0003-4447-0760

Sérgio Campello Oliveira ¹  orcid.org/0000-0003-1058-1139

¹ Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil

E-mail do autor principal: Everton Dornelas evertondornelas@gmail.com

Resumo

Internet das Coisas é um tema bastante debatido atualmente, tanto no meio acadêmico quanto no mercado de negócios, e tem atraído investimentos em pesquisa e desenvolvimento. Objetos que não eram conectados a internet terão essa funcionalidade e facilitarão a vida cotidiana fornecendo informações coletadas por sensores ou interagindo através de atuadores. Um recurso que é imprescindível para o cotidiano da sociedade e que muitas vezes não tem seu consumo adequadamente mensurado, é a água. Este artigo descreve a construção de um protótipo para o monitoramento do consumo de água em um ambiente doméstico, utilizando uma meta-plataforma de código aberto que visa facilitar o desenvolvimento de dispositivos que conectam coisas à internet. O objetivo do sistema é mensurar o consumo instantâneo e disponibilizar os dados gerados em tempo real na internet para serem consultados por outras aplicações. O projeto utilizou um Arduino UNO com um sensor de fluxo de água e um módulo transceptor nRF24L01+ de banda ISM 2,4GHz como dispositivo de medição. Além de um Raspberry Pi 2 Model B também com um módulo nRF24L01+ como gateway. Esse gateway, quando conectado a uma rede local TCP/IP com acesso à internet, disponibiliza na cloud os dados coletados pelo dispositivo. Os testes preliminares indicaram que foi possível mensurar o consumo em tempo real com uma precisão de aproximadamente 96% nos experimentos realizados.

Palavras-Chave: internet das coisas; consumo de água; monitoramento; sistema embarcado; casa inteligente (domótica); medição inteligente.

Abstract

Internet of Things is a subject much debated nowadays, in the academic as well as in the business market, and has attracted considerable investment in research and development. Things that were not connected to the internet will have this functionality and will make everyday life easier by providing information collected by sensors or interacting through actuators. A resource that is crucial to the daily life of society and that often does not have its consumption properly measured is water. This article describes the construction of a prototype for monitoring the water consumption in a domestic environment, using an open-source meta-platform which aims to facilitate the development of devices that connect things to the internet. The system objective is to measure the immediate consumption and to make available the generated data in real time in the internet to be consulted

by other applications. This project has used an Arduino UNO, with a water flow sensor and a nRF24L01+ 2.4GHz ISM band transceiver module as measuring device. Besides that, a Raspberry Pi 2 Model B with another nRF24L01+ module working as a gateway. This gateway, when connected to a local TCP/IP network with internet access, provides to the cloud the data collected by the device. The preliminary tests indicated that it was possible to measure consumption in real time with an accuracy of approximately 96% in the experiments performed.

Key-words: *internet of things; water consumption; monitoring; embedded system; smart home; smart metering.*

1 Introdução

Passada a primeira década do século XXI o Brasil ainda sofre com o abastecimento de água, principalmente nas regiões Norte e Nordeste. No país, quase 35 milhões de brasileiros não possuem acesso ao serviço, uma vez que o percentual da população atendida é de 83,3% [1]. Como agravante, a média do índice de perdas na distribuição de água do país é de 36,95%, por conta de vazamentos, falta de monitoramento, medições incorretas, entre outros problemas [2].

Diferentemente das companhias elétricas, que já iniciaram seus investimentos em redes inteligentes de energia (as *Smart Grids*) que possibilitam ao consumidor o acompanhamento e controle diário do consumo de energia elétrica (como é o caso de CEMIG [3] e CPFL [4]), não se percebe movimento similar das companhias de saneamento básico na chamada rede de água inteligente (do inglês *Smart Water Network* – *SWAN*) com serviços de acompanhamento disponíveis à população.

Distribuída a água para as torneiras da população, tem-se a segunda parte do desperdício, que é feito diretamente pelas pessoas. De uma maneira prática, não é fácil mensurar o quanto se gasta de água num banho, para escovar os dentes, lavar a louça ou até mesmo lavar as mãos. Fazendo novamente um comparativo com o consumo energético, em qualquer simples eletrodoméstico é fácil consultar no manual (ou até mesmo nas etiquetas localizadas no produto) qual a sua potência de consumo de energia em Watts, e com uma simples conta matemática tem-se ideia dos equipamentos com o maior potencial de consumo de energia elétrica, tornando o seu uso bem mais racional.

Levando em consideração que nem as companhias de abastecimento de água conseguem mensurar e monitorar corretamente o que entregam, esperar hidrômetros similares aos medidores inteligentes da energia elétrica - onde é possível consultar o consumo praticamente em tempo real através de um *smartphone* - parece ser algo distante da realidade. Entretanto, existe movimentação no mercado: *startups* lançam projetos de financiamento coletivo (*crowdfunding*) na tentativa de levantar verba suficiente para a fabricação em larga escala de seus produtos, que realizam a medição inteligente de consumo de água - é o caso da estadunidense *FLUID* [5] e da espanhola *Óasys* [6].

A importância de como o acompanhamento do consumo de água pode trazer benefícios é evidenciada

em resultados de um estudo realizado na Austrália, através do programa *H₂OME Smart* [7]. Com milhares de famílias participantes, o estudo mostrou quão notável é a economia no consumo de água a partir do momento que os dados ficam facilmente disponíveis para consulta: famílias que tinham um consumo acima da média conseguiram reduzi-lo após o acompanhamento [7].

A importância do consumo consciente de água, para tornar o mundo cada vez mais sustentável, aliado a escassez de tecnologia disponível para o auxílio desse objetivo, foram as motivações para este trabalho. O objetivo é propor um sistema embarcado capaz de fornecer um monitoramento mínimo do consumo hídrico doméstico, podendo ser expandido para um ambiente comunitário (como escolas, empresas, condomínios, etc.), utilizando soluções de código aberto (aproveitando principalmente a evolução da Internet das Coisas) e recursos de *hardware* simples, com boa disponibilidade no mercado e preço acessível.

2 Internet das Coisas

Internet das Coisas (*Internet of Things* – *IoT*) tem sido um termo bem popular (*buzzword*) no meio da Tecnologia da Informação e Comunicação (TIC), e que já movimentou milhares de dólares em pesquisas, novos produtos e *startups*.

A definição do que é Internet das Coisas ainda é algo bem discutido. O *IEEE* (Instituto de Engenheiros Eletricistas e Eletrônicos) descreve a expressão *Internet of Things* como “Uma rede de itens – cada um embarcado com sensores – que estão conectados à internet” [8]. Uma das mais completas definições é a da Cisco [9], que precisou de outro conceito, a Internet de Tudo (*Internet of Everything* - *IoE*), para explicar melhor a Internet das Coisas:

- Internet de Tudo: é a conexão em rede de pessoas, dados, processos e coisas. A *IoE* é composta de muitas transições de tecnologia, incluindo a Internet das Coisas.
- Internet das Coisas: é a conexão em rede de objetos físicos. *IoT* é uma das muitas transições tecnológicas que permitem a *IoE*.

Os números da Internet das Coisas impressionam: são esperados que 75% dos carros produzidos até 2020 saiam de fábrica prontos para conexão, cidades devem investir 133 bilhões de dólares em *IoT* até 2019, na agricultura cerca de 75 milhões de dispositivos devem estar conectados até 2020. O

principal motivo dessa corrida não difere do que já vivenciamos até hoje: economia [10].

Um estudo da *General Electric (GE)* [11] aponta para uma economia de pelo menos 63 bilhões de dólares apenas na área da saúde, com a adoção de objetos conectados. Em um outro estudo, da *Accenture* [12], se fala em 10,6 trilhões de dólares de contribuição cumulativa da *IoT* quando aplicada a indústria, e de acordo com a *IHS Technology* [13], o setor industrial será responsável por quase um terço dos novos dispositivos conectados entre 2015 e 2025.

Internet das Coisas aparece no pico da curva de expectativas inflacionadas para tecnologias emergentes do tradicional gráfico de ciclo de *hype* do *Gartner (Hype Cycle for Emerging Technologies)* em 2015 [14]. Um ponto interessante a ser observado, é que tanto para o gráfico de 2015 quanto para o de 2016 [15] existe um subtema da *IoT* presente na curva "Estímulo de Inovação" (*Innovation Trigger*): Plataforma de Internet das Coisas (*IoT Platform*).

Os ambientes de Internet das Coisas são caracterizados por um alto grau de heterogeneidade, por conta da especificidade de cada solução, que pode variar dentro de alguns pilares como alcance de comunicação, taxa de transmissão, consumo de energia e custo [16]. Tratar essa heterogeneidade é o objetivo de uma plataforma de Internet das Coisas.

Essa heterogeneidade gera um grande desafio para a criação e definição de padrões de comunicação para as coisas na internet, similar ao que já existe hoje na *World Wide Web (WWW)*. Ainda é difícil falar em padrões dentro desse universo, evidenciando a importância das plataformas, que nada mais são do que uma infraestrutura que visa facilitar a interação dos dispositivos (*hardware*, as coisas) com as aplicações (*software*). Uma plataforma pode ser de *software* – que é um conjunto de serviços responsáveis pelo tráfego e armazenamento dos dados – ou de *hardware* – microcontroladores com seu conjunto de sensores, atuadores e rádios de comunicação.

Um levantamento feito por *Matt Turck* [17] - um investidor em capital de risco - exibe o panorama 2016 do que está sendo trabalhado em *IoT*. Foram mapeadas 1266 companhias criando soluções e plataformas de Internet das Coisas nos últimos três anos. Ou seja, no cenário atual, cada um tem seu próprio sistema, com seu próprio método de comunicação (protocolos e/ou rádios de transmissão), e que provavelmente não possui a capacidade de interagir com aplicações de terceiros, afinal padrões ainda estão sendo discutidos. Pensando em facilitar esse cenário, surgiu o *KNoT* com a ideia de ser um mediador (*middleware*) entre plataformas de Internet

das Coisas.

Por exemplo, uma solução que utilizasse o *Arduino* como plataforma de *hardware* interagindo com a plataforma de *software AWS IoT* da *Amazon*, teria dificuldades para interagir também com a plataforma *Azure IoT* da *Microsoft*, uma vez que os protocolos de comunicação suportados pelas plataformas podem ser diferentes. Utilizando um *middleware*, os dispositivos da plataforma de *hardware* interagiriam com as plataformas de *software* de forma transparente, através da mediação realizada pelo *KNoT*.

2.1 *KNoT*

O *KNoT* é uma meta-plataforma de código aberto para a Internet das Coisas, que visa facilitar o desenvolvimento de soluções para conectar coisas [18]. Ele provê dispositivos de baixo custo que podem ser customizados de acordo com a necessidade da solução. Além disso, realiza a integração de plataformas de *IoT* criando uma solução *full stack (hardware e software)*, ou seja, torna possível que um dispositivo de uma determinada plataforma de *hardware* troque dados com um aplicativo para *smartphone* através de compartilhamento de dados via *cloud* de uma plataforma de *software*.

A ideia é que com uma simples configuração no roteador (*Gateway*) junto com a escrita de poucas funções no dispositivo (*Thing*) para descrever o comportamento do(s) sensor(es) e/ou atuador(es), a solução já esteja funcional. Os dados coletados pelos dispositivos trafegarão até a nuvem (*Cloud*) e poderão ser acessados via aplicações *web* ou *mobile*.

O *KNoT* utiliza uma arquitetura que segue uma topologia de rede em estrela conhecida como *spoke-hub* (ou *hub and spoke*). Essa arquitetura, que utiliza um *gateway* entre os *things* e a *cloud*, é indicada para dispositivos que possuem recursos reduzidos (*thin clients*) [19]. Arquiteturas em que os *things* conversam diretamente com a *cloud (front-loaded)* e até mesmo com outros *things (smart client)* são mais complexas, necessitam de dispositivos com maior poder de processamento e conseqüentemente tem maior custo [20].

A arquitetura *hub and spoke* é composta por três principais atores: *thing*, *gateway* e *cloud*. Os sensores presentes em cada *thing* geram dados - a partir de suas leituras - que são enviados para o *gateway* via rádio transceptor encapsulados em um protocolo específico. O *gateway* armazena esses dados em um repositório local (chamado de *fog*) que é sincronizado

com a *cloud*. Uma vez disponibilizados em nuvem, os dados podem ser acessados por aplicações através da internet. As aplicações também podem demandar ações para os *things* que possuem atuadores. Essas ações são dados que trafegam no caminho inverso: da *cloud* até o *thing*. A Figura 1 ilustra a arquitetura descrita.

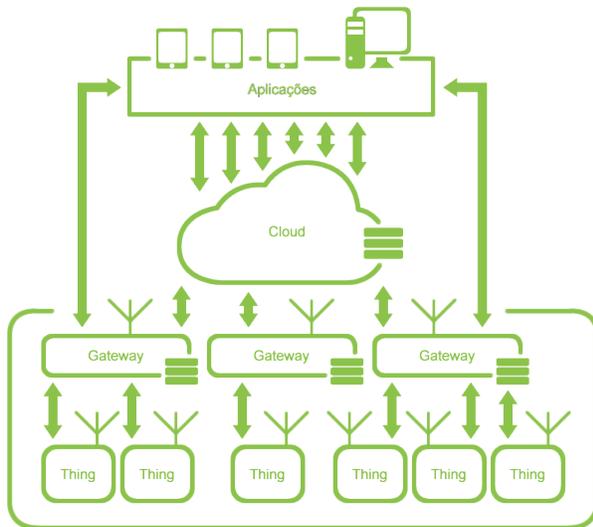


Figura 1: Arquitetura suportada pelo KNoT.

A versão do *KNoT* no momento em que este trabalho foi desenvolvido é a *01.02*, e tem como principais componentes de *software* o *KNoT Thing*, *KNoT Gateway*, *KNoT Service* e o *KNoT Cloud* que são definidos a seguir:

- *KNoT Thing*: é a biblioteca para o desenvolvimento do dispositivo. Depende de outras duas bibliotecas, uma que define o protocolo da camada de aplicação do *KNoT* (*KNoT Protocol*) e outra que cria a abstração para os periféricos do *hardware* (*KNoT HAL*). Nesta versão existe apenas o suporte para a plataforma do *Arduino*.
- *KNoT Gateway*: é o sistema operacional do roteador. É uma distribuição Linux customizada que contém todo o *software* necessário para o funcionamento do *gateway* dentro da rede. Para esta versão, o *hardware* suportado é limitado a placa *Raspberry Pi*.
- *KNoT Service*: é o serviço do *gateway* que funciona como interface entre os dispositivos cadastrados da rede local *KNoT* (*things*) e a nuvem (*cloud*), realizando o gerenciamentos dos

dispositivos e facilitando a troca de dados entre *thing* e *cloud*;

- *KNoT Cloud*: é o serviço *web* que interliga a rede local e as aplicações. Nesta versão, é um *fork* da plataforma *Meshblu* [21]. É embarcado no *KNoT Gateway* funcionando como uma instância local da *cloud* (chamada de *fog*) e também em servidores dedicados atuando como *cloud*.

Nas próximas versões, mais plataformas de *software* e *hardware* serão suportadas. Este trabalho utilizou a versão *01.02*, e por esse motivo se restringiu a implementar a solução utilizando os *hardwares* suportados.

3 Hardware

A escolha do *hardware* do projeto seguiu a lista de componentes suportados pela plataforma de *software*. O *KNoT*, na versão utilizada, tem suporte nativo para *Arduino* pelo *KNoT Thing*, para a *Raspberry Pi* pelo *KNoT Gateway* e para rádio de frequência o *nRF24L01+*. O custo total dos componentes de *hardware* utilizados no projeto foi de aproximadamente R\$ 250,00, conforme os preços detalhados na Tabela 1.

Tabela 1: Preço dos componentes de hardware utilizados.

Item	Quantidade	Preço
Raspberry Pi 2 Model B	1	R\$ 150,00
Arduino UNO	1	R\$ 50,00
Sensor de fluxo de água	1	R\$ 20,00
Módulo do nRF24L01+	2	R\$ 15,00

A organização dos componentes do *thing* consiste no sensor de fluxo instalado no ponto de entrada da distribuição de água através de uma conexão tubular com diâmetro nominal de meia polegada (1/2" NPS) e conectado eletricamente ao *Arduino UNO*, que possui capacidade de conexão sem fio através do transceptor utilizado. O *gateway*, que utiliza um *Raspberry Pi 2 Model B*, possui uma instância da *cloud* (*fog*) com capacidade de armazenamento de dados local e também realiza transferência dados sem fio com um transceptor igual ao utilizado no *thing*. Essa organização é ilustrada na Figura 2.

3.1 Radio

O *nRF24L01+* é um rádio de baixo custo, fácil de encontrar no mercado, tem uso já consolidado com diversas plataformas de *hardware* e *software*, um alcance que pode chegar até 1km (quando utilizado com uma antena externa e em campo aberto) e um baixo consumo de energia - que pode variar entre 26µA (em modo de espera) e 13.5mA (em potência máxima de recepção) [22]. Essas características influenciaram na sua escolha para ser o primeiro rádio suportado pela plataforma.

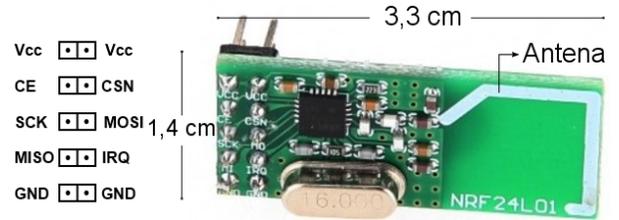


Figura 3: nRF24L01+ board.

3.2 Sensor

A escolha do sensor, assim como no caso do rádio, seguiu o critério custo e disponibilidade. O custo relativamente baixo em relação a outras opções dar-se ao fato de seu modo de funcionamento ser extremamente simples. É um mecanismo que possui uma válvula em formato de cata-vento com um ímã acoplado que trabalha em conjunto com um sensor *hall* para enviar um pulso; é conhecido também como *pinwheel sensor*.

Através dos pulsos gerados é possível mensurar a vazão de água. Esse sensor, o *YIFA YF-S201*, gera 450 pulsos por litro, com uma margem de erro de 5%. A vazão máxima é de 30 litros por minuto e a tensão de alimentação suportada é de 5V a 24V [23]. O sensor, mostrado na Figura 4, possui três fios: o vermelho (que é ligado ao 5V do *Arduino*), o preto (terra) e o amarelo (*hall effect pulse*) que é ligado a um pino do microcontrolador com suporte a interrupção externa.

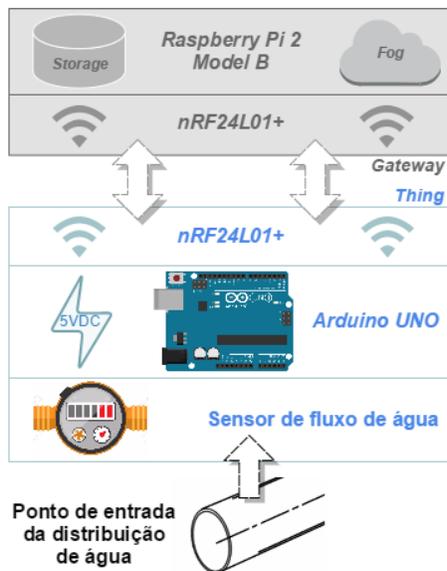


Figura 2: Organização dos componentes.

O rádio, que tem banda *ISM* de 2,4GHz de frequência e taxa de transferência máxima de 2Mbps, é controlado pelo microcontrolador via interface *SPI* (*Serial Peripheral Interface*) [22]. Como pode ser visto na Figura 3, a placa que encapsula o rádio possui uma antena interna e expõe os pinos das conexões *SPI*, que são conectados ao *Arduino*. Os dois pinos que recebem a tensão elétrica (*VCC* 1,9V ~ 3,6V) foram interligados e conectados a saída de tensão da placa do microcontrolador.



Figura 4: Sensor de fluxo de água.

Este sensor se mostrou sensível a passagem de ar, evidenciado que seu mecanismo de funcionamento simples tem limitações. Para contornar esse problema, algumas soluções disponíveis no mercado, que afirmam bloquear a passagem de ar, podem ser utilizadas; porém não possuem homologação de institutos de certificação de qualidade e em testes informais apresentaram algumas deficiências [24]. Caso seja necessário uma maior precisão na medição, este sensor mostrou não ser o mais indicado (existem alguns modelos de sensores do tipo turbina ou ultrassônico que possuem uma margem de erro menor, de cerca de 1%), mas para este trabalho ele atendeu sem maiores problemas.

3.3 Thing

Dentre as versões de *Arduino* suportadas pela biblioteca *KNoT Thing*, *UNO* e *Pro Mini*, a escolha se deu pela primeira opção. Apesar do *Pro Mini* ter um custo bem menor (cerca de um terço do preço), o *UNO* tem duas vantagens para este tipo de trabalho: é mais fácil para um protótipo pois é mais facilmente encontrado no mercado e não precisa ter componentes soldados como é o caso do *Pro Mini*.

O *Arduino UNO* [25] é uma placa de microcontrolador baseada no *Atmel AVR ATmega328P* [26]. A placa possui 14 pinos de entrada/saída digitais, 6 entradas analógicas, um cristal de quartzo de 16MHz e o *In-Circuit Serial Programming (ISCP)* via porta de conexão *USB*, que facilita o desenvolvimento.

O rádio, controlado via interface *SPI*, é conectado ao *Arduino* como ilustra a Figura 5. Já o sensor utilizado, pode ter a saída de sinal ligada ao pino 2 ou 3, pois nesses pinos existe o suporte para a interrupção externa.

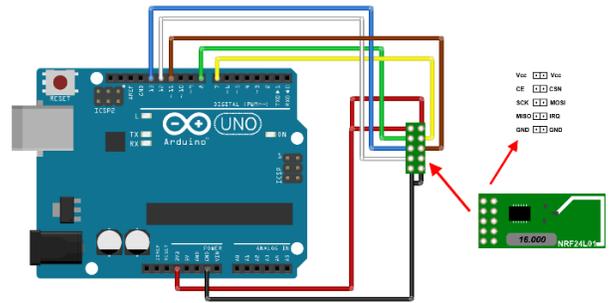


Figura 5: Arduino UNO com o nRF24L01+.

3.4 Gateway

O *KNoT Gateway* tem suporte para as três versões da *Raspberry Pi Model B*. Todas possuem uma boa disponibilidade no mercado, e o preço varia de acordo com a versão: quanto mais recente é mais atualizada, e conseqüentemente tem um custo mais elevado. Para este trabalho a versão escolhida foi a 2, por reunir um melhor custo benefício.

A placa conta com um processador *ARM Cortex-A7* quad-core de 900MHz, 1GB de *RAM*, 40 pinos *GPIO* (incluindo o suporte ao *SPI*, necessário para o rádio), porta *Ethernet* (necessária para comunicação com a rede local *TCP/IP*) e entrada para cartão de memória (que armazena o sistema operacional). Com essa configuração o *gateway* consegue funcionar bem com um sistema operacional embarcado baseado em *Linux* e ter até 240 *things* conectados em sua rede local *KNoT*.

O rádio é conectado a interface *SPI* da *Raspberry Pi 2* por meio dos pinos 8 (*SS*), 9 (*MISO*), 10 (*MOSI*), 11 (*SCK*) e 25 (*CE*), enquanto a tensão é fornecida através do pino *VCC* de 3,3V da placa. O conjunto de pinos *GPIO* da placa é ilustrado na Figura 6 [27].

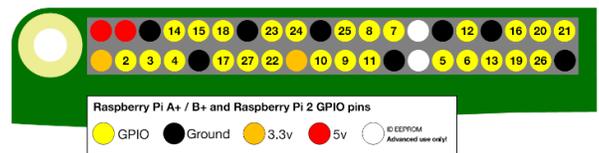


Figura 6: Pinos GPIO Raspberry Pi 2.

4 O Aplicativo AquaControle

Com a utilização do *KNoT*, o desenvolvimento do *software* se restringe ao código do *thing* e da aplicação que interpreta os dados da *cloud* (que não fez parte do escopo inicial deste projeto). Para o *gateway*, é necessário apenas a geração da imagem de seu sistema operacional: uma distribuição Linux para sistemas embarcados com os artefatos necessários para o gerenciamento dos dispositivos na rede local *KNoT*.

4.1 *KNoT Thing*

Para facilitar o desenvolvimento do *software* do dispositivo, a plataforma disponibiliza uma biblioteca a ser utilizada chamada de *KNoT Thing*. Sua utilização simplifica a codificação, sendo necessário apenas o desenvolvimento de poucas funções com objetivos simples: realizar a leitura de dados do sensor e demandar configurações para o atuador.

O processo de fluxo de dados entre o *thing* e *gateway* torna-se transparente com a utilização da biblioteca: o que for lido pelo sensor é enviado do *Arduino* para a *Raspberry Pi*, e o caminho inverso é feito demandando configurações para o atuador. Essa troca de dados é realizada utilizando o protocolo *KNoT* sobre o rádio *nRF24L01+*, que é suportado nativamente pela plataforma.

Toda essa abstração é possível graças ao uso das bibliotecas *KNoT HAL* e *KNoT Protocol*. O objetivo principal da *KNoT Thing* é criar uma interface entre o código customizado do dispositivo e as bibliotecas de abstração. A Figura 7 ilustra a configuração completa (*hardware* e *software*) de um *KNoT Thing*. O *KNoT μOS* mostrado na Figura 7, é a evolução da biblioteca *KNoT Thing* para um micro sistema de tempo real e estará disponível nas próximas versões da plataforma.

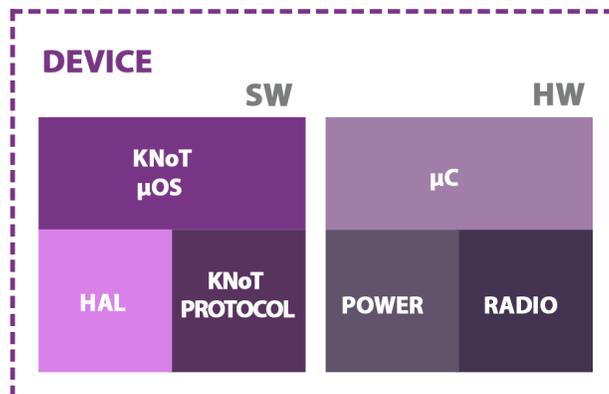


Figura 7: Camadas do *KNoT Thing*.

Para o funcionamento do *Arduino*, é necessária a implementação de duas funções: *setup* e *loop*. Na primeira, que é executada uma única vez, configura-se o dispositivo: pino de entrada do sensor, interrupção, velocidade da serial (utilizada para depuração) e a chamada das funções da biblioteca *KNoT Thing*. Já a segunda função, tem o objetivo de executar as tarefas necessárias enquanto o dispositivo estiver ativo.

Na primeira etapa (Etapa 1 da Figura 8), que é de configuração, as funções chamadas tem os seguintes propósitos: habilitar a interrupção externa, inicializar o dispositivo como um *thing* na rede *KNoT* local e configurar e registrar o(s) sensor(es) e atuador(es) do dispositivos através das respectivas funções de leitura e escrita.

O primeiro passo da etapa de configuração é habilitar a interrupção externa para capturar os pulsos gerados pelo sensor de fluxo – configuração que acontece no passo 1.1 ilustrado na Figura 8. Cada pulso é contabilizado, e a partir do montante de pulsos, a função de leitura de dados do sensor - que é chamada pela biblioteca *KNoT Thing* - calcula o quanto os pulsos representam em volume. O projeto utilizou apenas um sensor, e por esse motivo nenhum atuador é configurado e registrado.

No passo seguinte, 1.2 da Figura 8, o dispositivo é inicializado como um *thing* na rede local *KNoT* e um nome lhe é atribuído. No terceiro passo, a função de leitura responsável por recuperar as informações do sensor de fluxo de água é registrada e configurada para trabalhar com volume, sendo o litro a unidade utilizada - essa configuração acontece no passo 1.3 na Figura 8.

Na segunda etapa (Etapa 2 da Figura 8), que é um laço infinito, cada iteração realiza a chamada de uma função da biblioteca que executa as tarefas necessárias: verifica se dados foram recebidos da

rede e se eles demandam ações para o atuador, além de enviar para o *gateway* os dados gerados pelo sensor – representado no passo 2.1 da Figura 8.

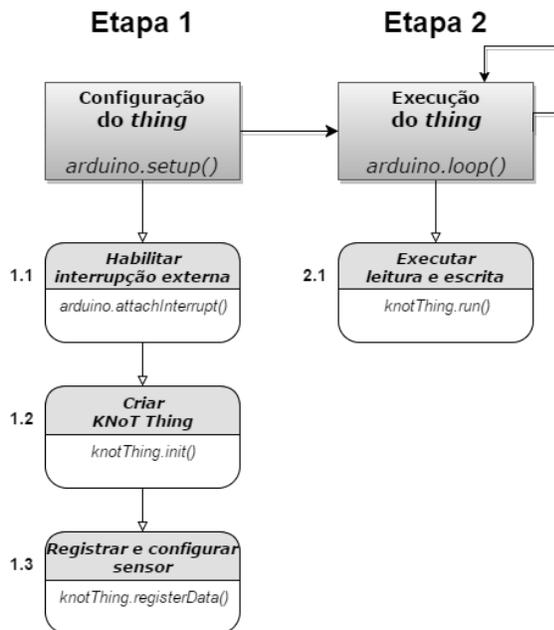


Figura 8: Fluxo do KNoT Thing.

4.2 KNoT Gateway

Utiliza o *Buildroot* [28], uma ferramenta de código livre, eficiente e fácil de usar para gerar sistemas Linux embarcados. A imagem customizada gerada após a compilação contém todos os artefatos necessários para o funcionamento do *gateway*: *KNoT Protocol*, *KNoT HAL*, *KNoT Service* e *KNoT Cloud*. Como pode ser visto na Figura 9, todos esses artefatos funcionam sobre o Linux.

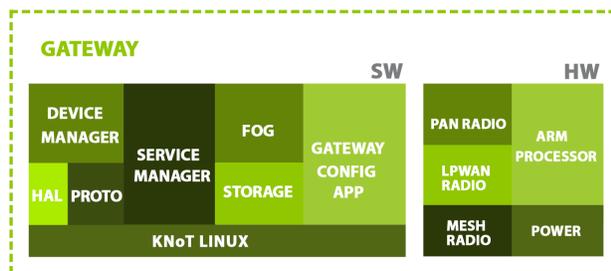


Figura 9: Camadas do KNoT Gateway.

leva um certo tempo, mas é simples e se resume a dois comandos: o primeiro configura a compilação para o *hardware* da *Raspberry Pi 2*, e o segundo inicia a compilação propriamente dita. É disponibilizado junto com o código fonte do *KNoT Gateway* um script que facilita o processo de gravação da imagem gerada no cartão de memória que será conectado a placa: o script formata e copia os arquivos necessários para as partições adequadas. Com a imagem gravada e o cartão conectado à *Raspberry Pi*, o *gateway* está funcional.

O *gateway* funcional é sinônimo de que os serviços e *KNoT Cloud* e *KNoT Service* estão ativos. O primeiro, que no *gateway* atua como uma instância local da *cloud* (e por isso é chamado de *KNoT Fog*), é responsável por armazenar os dados trocados com os *things* da rede local *KNoT* e sincronizá-los com uma *cloud* remota (se configurado para isso). A *Fog*, como ilustrado na Figura 9, depende de uma base de dados (*Storage*) local.

O *KNoT Fog* recebe os dados através do *KNoT Service* – ilustrado na Figura 9 como *Service Manager* – que é o serviço responsável pelo gerenciamento e troca de dados com os dispositivos cadastrados; e assim como a biblioteca *KNoT Thing*, também depende da *KNoT HAL* e da *KNoT Protocol* para as abstrações necessárias – respectivamente representados por *HAL* e *PROTO* na Figura 9.

Para comunicação com rede local *KNoT*, o *gateway* deve utilizar o mesmo rádio dos *things*, e por isso deve suportar o *nRF24L01+*. Para comunicação com a internet (para sincronizar com a *cloud*, por exemplo) utiliza *Ethernet*, presente na *Raspberry Pi 2* e suportada pelo sistema operacional. O uso da *Ethernet* é útil também para acessar através da rede local *TCP/IP* uma aplicação *web* de configuração disponível no *gateway*, chamada de *KNoT Gateway Web UI*.

Como mencionado anteriormente, o *KNoT Cloud/Fog* utiliza o *Meshblu* como aplicação base, que por sua vez utiliza como repositório de dados um sistema gerenciador de banco de dados *NoSQL* orientado a documentos chamado *MongoDB* [29]. Tanto o *Meshblu*, quando o *MongoDB* são aplicações de código livre de licenças flexíveis, o que o tornam compatíveis com a proposta da plataforma *KNoT*. O *Meshblu* é portátil e tem suporte a uma série de protocolos de comunicação (*MQTT*, *CoAP*, *HTTP*, *Websocket*, *AMQP*, etc.) que facilita a troca de dados com aplicações *web* e/ou *mobile*. Já o *MongoDB* consegue ser referência em portabilidade e performance.

O processo de compilação do sistema operacional

5 Funcionamento da Solução

Com *thing* e *gateway* em funcionamento, a primeira troca de mensagem é a solicitação de entrada na rede local *KNoT*. Ao receber a mensagem, o *gateway* aguarda a permissão que deve ser concedida pela interface *web* de configuração (uma única vez para cada novo dispositivo). O acesso a essa interface pode ser realizado através de um smartphone, tablet ou computador. O *thing* inicia no estado *CONNECTING*, e após estabelecer a conexão com o *gateway*, evolui para o estado *REGISTERING*, aguardando as credenciais para fazer parte da rede local *KNoT*.

Com a permissão concedida, o *thing* recebe um identificador (código *UUID* - *Universal Unique Identifier*) e um *token*, alterando seu estado para *SCHEMA*. Nesse estado o dispositivo envia para o *gateway* a sua configuração: os sensores e suas unidades de medida. Após esse passo, o *thing* finalmente está *ONLINE*, estado em que enviará os dados lidos pelo sensor e/ou receberá dados para aplicar no atuador. Para o dispositivo desenvolvido neste trabalho, os dados são enviados a cada variação de litro, com o objetivo de diminuir o fluxo de dados e o consumo de energia. A Figura 10 ilustra a máquina de estados.

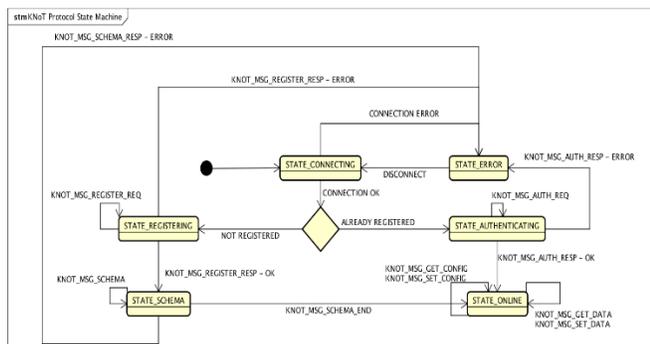


Figura 10: Máquinas de estados *KNoT Protocol*.

Com os dados transmitidos e armazenados na *cloud* (e *fog*), é possível recuperá-los utilizando uma *API HTTP REST* disponibilizada nativamente pelo *Meshblu* [30]. As informações trocadas entre os dispositivos e o *gateway* são rastreáveis a partir do *UUID* de cada *thing*. Dessa forma, aplicações *web* e/ou *mobile* podem ser desenvolvidas para realizar o monitoramento dos dados.

5.1 Experimentos

Foram realizados dois experimentos com o protótipo, para simular o ambiente de leitura do fluxo de água. Um primeiro teste utilizando uma mangueira de conexão de caixa acoplada de aproximadamente 1 metro de comprimento e um segundo teste utilizando uma mangueira de jardim de aproximadamente 15 metros de comprimento.

Utilizando a mangueira menor, com uma extremidade conectada a uma torneira e a outra na conexão NPS 1/2" do sensor de fluxo, abria-se a torneira e a água passada pelo sensor era despejada em um recipiente com 1 litro de capacidade de armazenamento. Com o recipiente cheio, verificava-se a quantidade de pulsos gerados pelo sensor. Esse procedimento foi repetido por cinco vezes e a média de pulsos gerados por litro pelo sensor foi de 464,7.

Com a mangueira maior, de 15 metros, a ideia foi simular um ambiente de tubulação vazia, em que a passagem de água causa um deslocamento de ar. Foi realizado o mesmo teste mencionado anteriormente com o mesmo número de repetições. Nesse segundo teste a média de pulsos por litro foi de 466,9. Considerando que o sensor especifica 450 pulsos por litro, o erro relativo percentual foi de aproximadamente 3,27% e 3,75% para o primeiro e o segundo teste, respectivamente.

Com esse cenário de testes, cada litro contabilizado pelo *thing* gerou uma mensagem com dados para o *gateway*, que foram sincronizados com a *cloud*. Utilizando ferramentas que auxiliam a criação de mensagens *HTTP* (protocolo suportado pela *cloud*), foi possível criar requisições para recuperar os dados armazenados em nuvem.

6 Conclusões

Este trabalho criou um protótipo que faz a medição de consumo hídrico em um ambiente doméstico. Utilizou componentes de *hardware* facilmente encontrados no mercado e com um baixo custo. O tempo gasto no desenvolvimento do *software* foi mínimo, com a implementação de poucas linhas de código fonte, devido ao uso de plataformas de código aberto.

Utilizando um dos múltiplos conceitos da *IoT* – que é o de conectar coisas à internet, o trabalho propôs conectar algo que usualmente não possui nem mesmo a medição eletrônica, e que pode ser aplicado não só para acompanhar o consumo de água, mas também para monitorar se o ambiente está sendo abastecido corretamente.

A solução proposta neste trabalho pode ser evoluída para garantir melhor precisão nas medições, utilizando melhores sensores como os do tipo turbina ou ultrassônicos; além de também incluir atuadores, que podem, por exemplo, bloquear o fluxo. Outra interessante evolução é a utilização da solução em outros ambientes, como condomínios e empresas, o que pode demandar para o sistema o suporte a rádios de longo alcance, como o *LoRa* ou *NB-IOT*.

Em um cenário futuro, fazendo valer a lei de *Moore* - onde tem-se a evolução do *hardware* num determinado período de tempo por um preço constante - o sistema poderá suportar uma arquitetura em que o *thing* tem capacidade de conexão direta com a *cloud* (arquitetura *front-loaded*), evitando a utilização do *gateway*, que é o componente da solução com custo mais elevado.

O próximo passo a ser trabalhado, é a criação de uma aplicação com interface amigável (*mobile* ou *web*), para recuperar os dados armazenados na *cloud* (ou *fog*) e transformá-los em informação. No protótipo criado neste trabalho, foram utilizadas ferramentas como o *Postman* e o *Curl* para recuperar dados puros, sem uma análise sintática.

O *KNoT*, em constante evolução, trará mais funcionalidades para o *KNoT Service* e melhorias de usabilidade para o *KNoT Gateway Web UI* nas próximas versões da plataforma. O gerenciamento da rede local *KNoT (things e gateways)* será facilitado, o que contribuirá para inserção de novos *things* com seus sensores e atuadores, que combinados poderão gerar melhores informações; como por exemplo, mensurar o consumo de água levando em consideração a temperatura do ambiente.

Referências

[1] Secretaria Nacional de Saneamento Ambiental. Diagnóstico dos Serviços de Água e Esgotos - 2015. Ministério das Cidades, Brasília, Fevereiro de 2017.

[2] Go Associados. Perdas de Água: Desafios ao Avanço do Saneamento Básico e à Escassez Hídrica. Instituto Trata Brasil, São Paulo, Março de 2015.

[3] CEMIG. A CEMIG e o futuro - Inovação e Sustentabilidade. Disponível em: <<https://goo.gl/09X1R4>>. Acesso em: 7 de Setembro de 2016.

[4] CPFL Energia. Energias sustentáveis - Smart Grid. Disponível em <<http://goo.gl/hzJk97>>. Acesso em: 7 de Setembro de 2016.

[5] FLUID. The Learning Water Meter. Disponível em <<http://goo.gl/e8zXnY>>. Acesso em: 7 de Setembro de 2016.

[6] Ôasys. Visualize, understand, and act on our household water consumption. Disponível em <<http://goo.gl/05Vec2>>. Acesso em: 7 de Setembro de 2016.

[7] M. Anda, F. L. G. Brereton, J. Brennan, E. Paskett. Smart Metering Infrastructure for Residential Water Efficiency: Results of a Trial in a Behavioural Change Program in Perth, Western Australia. In *Anais do First International Conference on Information and Communication Technologies for Sustainability - ICT4S*, Zurique, páginas 116-122, 2013.

[8] R. Minerva, A. Biru, D. Rotondi. Towards a definition of the Internet of Things (IoT) - Revision 1. IEEE Internet of Things, Maio de 2015.

[9] J. Bradley, C. Reberger, A. Dixit, V. Gupta. Internet of Everything: A \$4.6 Trillion Public-Sector Opportunity. Cisco White Paper, 2013.

[10] Peixoto, Eduardo. IoT: Onde estão os negócios? Série IoT - Volume A.2. CESAR, Recife, 2016.

[11] P. C. Evans, M. Annunziata. Industrial Internet: Pushing the Boundaries of Minds and Machines. General Electric White Paper, Novembro de 2012.

[12] M. Purdy, L. Davarzani. The Growth Game-Changer: How the Industrial Internet of Things can drive progress and prosperity. Accenture Strategy White Paper, 2015.

[13] B. Morelli, J. Howell, J. Watson, S. Lucero, L. Ratliff, Y. Kanesin, C. Kim, P. Tomasi. IoT trend watch 2017. IHS Markit, 2017.

[14] Gartner's 2015 Hype Cycle for Emerging Technologies Identifies the Computing Innovations That Organizations Should Monitor. Disponível em <<https://goo.gl/INiSVb>>. Acesso em: 17 de Janeiro de 2017.

[15] Gartner's 2016 Hype Cycle for Emerging
<http://dx.doi.org/10.25286/rep.v2i2.575>

Technologies Identifies the Computing Innovations That Organizations Should Monitor. Disponível em <<https://goo.gl/ZUoCF5>>. Acesso em: 17 de Janeiro de 2017.

[16] CESAR. KNoT - KNoT Network of Things. Slide 4. Disponível em <<https://goo.gl/x0xkq3>>. Acesso em: 1 de Maio de 2017.

[17] Matt Turck. Internet of Things: Are We There Yet? (The 2016 IoT Landscape). Disponível em <<https://goo.gl/kJax3L>>. Acesso em: 1 de Maio de 2017.

[18] KNoT. KNoT Network of Things. Disponível em <<https://goo.gl/qtJEIX>>. Acesso em: 1 de Maio de 2017.

[19] R. Quinell. 3 architectures dominate the Internet of Things. EDN Network, Novembro de 2014.

[20] G. Reiter. Wireless connectivity for the Internet of Things - One size does not fit all. Texas Instruments White Paper, Junho de 2014.

[21] Octoblu Meshblu. Meshblu machine-to-machine (M2M) instant messaging (IM). Disponível em <<https://goo.gl/GCLQBB>>. Acesso em: 01 de Maio de 2017.

[22] Nordic Semiconductor. nRF24L01+ Single Chip 2.4GHz Transceiver Product Specification v1.0. Disponível em <<https://goo.gl/nx3nqr>>. Acesso em: 2 de Maio de 2017.

[23] Adafruit Industries. Sea Water Flow Sensor YF-S201 (Plastic 1/2") Technical Details. Disponível em <<https://goo.gl/jag56k>>. Acesso em: 22 de Maio de 2017.

[24] Globo - G1. Fantástico: Testes avaliam aparelho que promete bloquear ar e baixar conta de água. Disponível em <<https://goo.gl/fXmZWQ>>.

[25] Arduino AG. Arduino Board UNO. Disponível em <<https://goo.gl/RBmkbQ>>. Acesso em: 1 de Maio de 2017.

[26] Atmel Corporation. Datasheet do CI AVR Atmega 328P. Disponível em <<https://goo.gl/UwMvNZ>>. Acesso em: 1 de Maio de 2017.

[27] Raspberry Pi Foundation. Raspberry Pi

Hardware Documentation. Disponível em <<https://goo.gl/agfPNt>>. Acesso em: 3 de Maio de 2017.

[28] Buildroot. Buildroot - Making Embedded Linux Easy. Disponível em <<https://goo.gl/vvJSQ2>>. Acesso em: 01 de Maio de 2017.

[29] MongoDB, Inc. MongoDB. Disponível em <<https://goo.gl/3xHymI>>. Acesso em: 1 de Maio de 2017.

[30] Octoblu Meshblu. Meshblu REST API Documentation. Disponível em <<https://goo.gl/PnTynB>>. Acesso em: 8 de Maio de 2017.