Projeto AMedCA - Documentação

**1. Nome do Projeto:**  AMedCA – Anel Medidor de Consumo de Água

**2. Breve Descrição do Projeto**

O consumo residencial e predial de água geralmente é realizado por meio de medidores físicos acoplados a canos. Estes dispositivos raramente são digitais e fornecem apenas informações básicas de metros cúbicos de água consumidos.

Diversos medidores e registros eletrônicos disponíveis no mercado permitem a medição e coleta de informações relacionadas ao consumo de água. Contudo, estes dispositivos requerem a instrumentação do cano, ou seja, é preciso instalar fisicamente o medidor em um ponto específico. Tal instalação muitas vezes é complicada e requer a interrupção do fornecimento de água, quebra de paredes e outros procedimentos complexos acompanhados de custos adicionais de instalação. Além disso, os medidores eletrônicos existentes fornecem informações pouco precisas sobre qual é a torneira, chuveiro ou outro ponto de saída de água responsável pelo consumo.

A motivação deste projeto é baseada na afirmação: o que não pode ser medido não pode ser controlado. Além disso, a precisão da medição pode influenciar muito em como o controle e consumo consciente de água será realizado.

O projeto AMedCA (Anel Medidor de Consumo de Água) tem como objetivo fornecer informações precisas sobre o consumo de água residencial e predial sem a necessidade de instrumentação complexa.

Basicamente o projeto AMedCA assume a forma de um anel de velcro com um sensor de vibração acoplado. Durante a implantação do sistema o anel deve ser fixado no lado externo do cano de modo que o sensor captará a vibração do cano gerada pela da água sendo passada dentro do cano. Basicamente o projeto utilizará a placa de desenvolvimento Freedom para microcontroladores Kinetis K (FRDM-K64F) junto com um sensor de vibração do tipo Piezo e analisa os dados do sinal do sensor utilizando técnicas de processamento digital de sinais.

A partir de um treinamento o sistema associado ao AMedCA capta o um padrão de vibrações emitido pelo cano. Este padrão será armazenado em uma solução na nuvem que detectará e associará a vibração captada pelo sensor com a quantidade de água sendo consumida e também qual torneira ou outro ponto de saída de água da residência for acionado.

A partir dos dados coletados um backend apresentará diversas informações individuais e gerais de consumo por dia, final de semana, mês e ano. Além disso, o sistema também fará previsões sobre consumos futuros e contará com um módulo que detectará anomalias e variações nos dados. Esta detecção será empregada para, entre outras aplicações, detectar vazamentos em canos, consumo em horários anormais devido a torneiras que ficaram abertas e identificação de padrões de consumo individuais para prédios e condomínios que não possuem um sistema de medição individual por unidade e que realizam o rateio da conta de água.

**3. Vídeo do Funcionamento do Projeto**

O vídeo de apresentação do projeto está hospedado no YouTube no seguinte link:

<https://youtu.be/IaxPOV3pXJw>

Os slides utilizados no vídeos estão hospedados no SlideShare no link abaixo:

<http://pt.slideshare.net/pichiliani/projeto-amedca-concurso-casa-conectada-2015>

Segue a descrição do áudio utilizado no vídeo do projeto:

SLIDE 1

Olá, meu nome é Mauro Pichiliani e este é o video de apresentação do projeto AMedCa.

SLIDE 2

O contexto do projeto é a medição e o consumo consciente de água em prédios e residências.

A medição de consumo de água é feita por registos e medidores conectados à tubulação.

Porém, tais dispositivos nem sempre indicam o consumo individual por apartamento.

Eles também não fornecem informações precisas sobre cada ponto de saída de água, por exemplo, se uma torneira está meio aberta.

Além disso, a instalação de medidores é complexa e demanda reserva do horário para realização do serviço, custos de mão de obra, interrupção do fornecimento de água e quebra de paredes.

SLIDE 3

O objetivo do projeto AMedCA é facilitar a instalação de um medidor que informa o consumo de água.

A solução é um anel com um sensor que mede a vibração do cano. É preciso um treinamento inicial do sinal para lidar com vários tipos de canos e níveis de pressão.

Os dados do sensor são captados pela placa FRDM-K64F que os envia para uma solução na nuvem que calcula o uso da água, gera relatórios e identifica de padrões de consumo.

SLIDE 4

A arquitetura do sistema é composta por um cano por onde passa a água instrumentado com um sensor de vibração do tipo piezo.

Este sensor é conectado na placa ligada a um roteador WiFi com acesso a internet. A placa envia os dados do sensor para uma solução na plataforma Windows Azure.

Os usuários então acessam os relatórios, gráficos e indicadores em uma aplicação Web.

SLIDE 5

Nesta demonstração um reservatório foi conectado a canos de PVC de meia polegada e à uma torneira. O sensor foi acoplado ao cano e detecta se a torneira está aberta ou fechada.

Quando a torneira é aberta o sistema detecta este evento e calcula o consumo. O fechamento também é detectado e mostra-se o consumo de água.

SLIDE 6

A análise do consumo de água é importante para a conscientização e controle deste recurso natural.

Soluções existentes requerem muita instrumentação e fornecem pouca precisão em relação ao ponto de saída de água.

O AMedCA mede vibrações do cano e permite um controle do uso de água com relatórios, detecção de padrão de consumo e previsão futura de gastos.

**4. Esquemáticos**

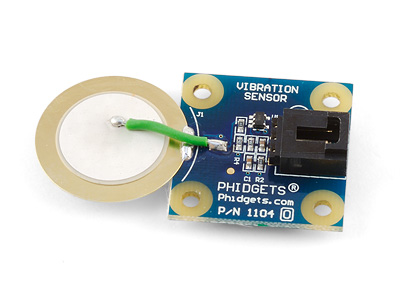
O esquemático deste projeto se resume a ligação da placa FRDM-K64F com um sensor do tipo piezo. O sensor utilizado no projeto foi fornecido pela empresa PhidGets Inc (<http://phidgets.com/>) e possui o código 1104\_0. A Figura 1 mostra os detalhes deste sensor.

Figura 1. Sensor de vibração Phidget 1104\_0, disponível em: <http://www.phidgets.com/products.php?category=1&product_id=1104_0>

Este sensor trabalha com a tensão de 5 volts e possui uma saída de sinal analógico, conforme mostrado na Figura 2 que fornece a indicação dos pinos.

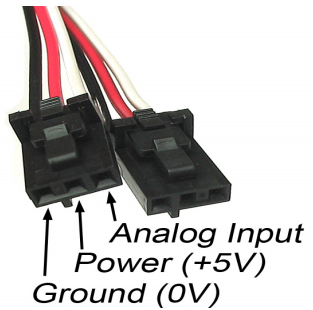


Figura 2. Identificador dos pinos do sensor 1104\_0.

O projeto foi montado utilizando o sensor de vibração do tipo piezo ao invés de um sensor de som cujo código é KY-038 ou equivalente. A troca de sensores em relação à proposta inicial foi devido à precisão superior do sinal quando se utiliza o sensor de vibração encostado ao cano de água testado (PVC de meia polegada).

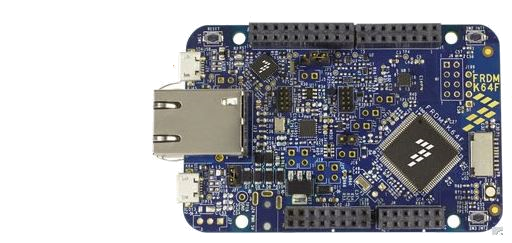
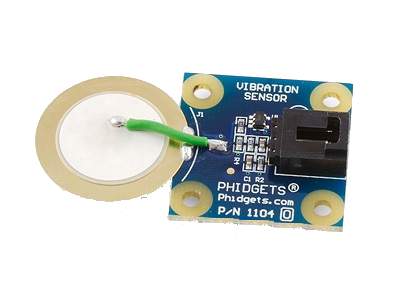
O sensor de vibração código SW18010P (<http://www.filipeflop.com/pd-b8f7a-sensor-de-vibracao-sw18010p.html?ct=41d97&p=2&s=1>) ou o sensor código bzb8780160 (<http://www.bizoner.com/piezo-disk-vibration-sensor-compatible-arduinos-ide-p-392.html>) são alternativas ao sensor 1104\_0 da Phidgets.

A ligação dos pinos do sensor com a placa FRDM-K64F foi feita da seguinte forma: o cabo preto (Ground 0V) foi ligado no pino GND-14, o cabo vermelho (Power +5V) foi ligado no pino P5V\_USB-10 e o cabo branco (Analog Input) foi ligado no pino PTC10-12. A Tabela 1 a apresenta os pinos da placa FRDM-K64F utilizado, a qual cabeçalho da placa eles pertencem e qual é o nome do pino compatível com o Arduino UNO.

|  |  |  |  |
| --- | --- | --- | --- |
| **Pino na placa FRDM-K64F** | **Cor do cabo** | **Cabeçalho**  **na placa FRDM-K64F** | **Pino equivalente**  **no Arduino UNO** |
| GND-14 (Ground) | Preto | J3 I/O Header | GND |
| P5V\_USB-10 (Power +5V) | Vermelho | J3 I/O Header | 5V |
| PTC10-12 (Analog Input) | Branco | J4 I/O Header | A5 |

Tabela 1. Pinos utilizados e compatibilidade com o Arduino

A Figura 3 mostra o esquemático da placa com a ligação de cabos com o sensor.



Pino GND-14

(Ground)

PTC10-12 (Analog Input)

P5V\_USB-10 (Power +5V)

Figura 3. Ligações do sensor de vibração 1104\_0 com a placa FRDM-K64F

**5. Layout**

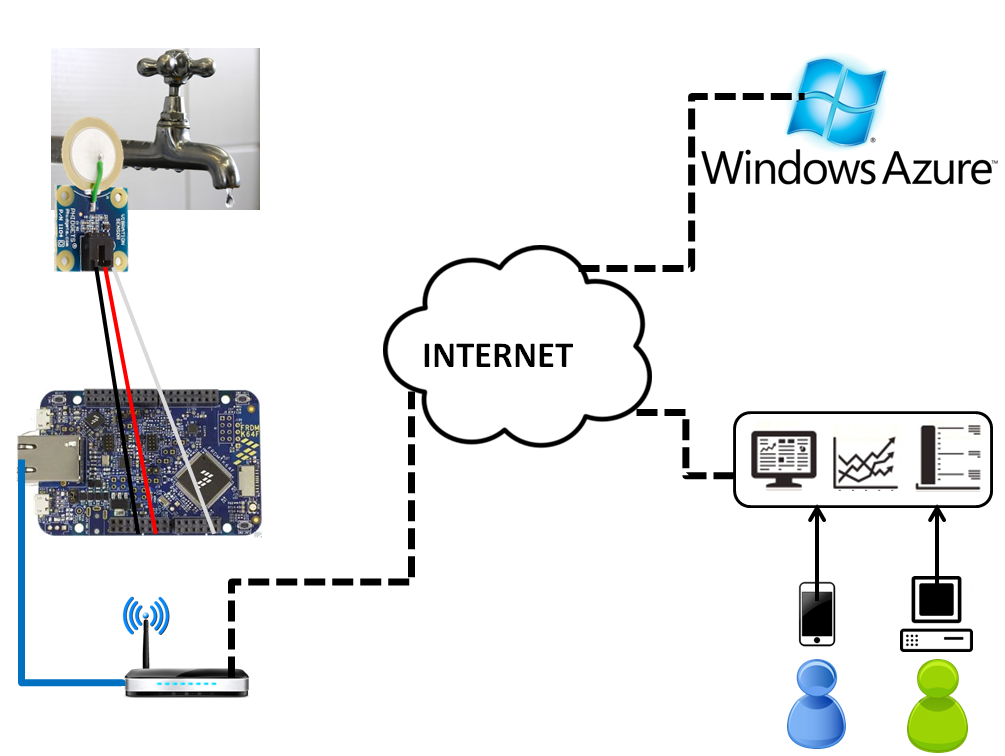
O layout da solução é baseado nos seguintes componentes arquiteturais: um cano que armazena a água corrente e que vibra, o sensor de vibração 1104\_0, a placa FRDM-K64F, um roteador WiFi com acesso à Internet, uma conta na plataforma Windows Azure e os usuários com dispositivos móveis ou desktop para acesso a aplicação Web. A Figura 4 mostra o layout geral da solução.

Figura 4. Layout geral do projeto AMedCA.

**6. Fluxograma**

O fluxograma do projeto AMedCA descrito nesta sessão é baseado no protótipo apresentado no vídeo que detecta a abertura ou não de uma torneira e calcula o consumo de água. Este projeto é baseado em dois canos de PVC de meia polegadas e 52 centímetros ligados horizontalmente por meio de um conector de 90 graus (cotovelo) e com uma torneira na ponta. A detecção do sinal foi baseado na aberta de 180 graus (uma volta) da torneira. Neste cenário aproximadamente 100ml de água passam pela torneira a cada 3,8 segundos (valores calculados experimentalmente).

O fluxograma do projeto envolve os passos necessários para obter as medidas do sensor, treinar o ruído do sinal, obter a média do ruído, comparar o sinal com a média mais ou menos um percentual de ajuste (0,002% nos teste iniciais), identificar ou não a abertura da torneira, mudar o estado da torneira e calcular o valor de água consumido. A Figura 5 apresenta o fluxo do protótipo de detecção de torneira aberta.

NÃO

NÃO

SIM

SIM

NÃO

INÍCIO

LER DADO

DO SENSOR

TREINAR RUÍDO DO SINAL

RUÍDO DO SINAL TREINADO?

SINAL NA MÉDIA± LIMITE?

TORNEIRA FECHADA?

SIM

MUDAR ESTADO PARA TORNEIRA ABERTA

MUDAR ESTADO PARA TORNEIRA

FECHADA

CALCULAR VALOR CONSUMIDO DE ÁGUA

MOSTAR ÁGUA CONSUMIDA

FIM

Figura 5. Fluxograma do projeto AMedCA no protótipo de detecção de abertura da torneira.

**7. Código fonte**

O código fonte do protótipo do projeto AMedCA foi desenvolvido em duas partes: o código que é executado dentro da placa FRDM-K64F e o código que recebe os dados do sensor e faz a análise do dados para computar o estado da torneira e calcular o consumo de água.

A plataforma mbed (<https://developer.mbed.org/compiler/>) foi utilizada para o desenvolvimento do código fonte que é enviado para a placa. O escolha desta plataforma foi devido à sua facilidade e a grande quantidade de exemplos.

No protótipo inicial do AMedCA o valor do sensor é lido a partir da porta analógica a cada a cada 100 milisegundos. O valor do sensor é representado pelo tipo de dados double e fica em 0.0 e 1.0 com precisão de 6 casas decimais. O código fonte escrito na linguagem C e armazenado no arquivo **main.cpp** para lê os dados do sensor e enviá-los pela porta serial é apresentado na listagem abaixo:

#include "mbed.h"

AnalogIn analog\_value(A5);

// Para rodar este programa é preciso utilizar a plataforma mbed

// disponível on-line em: https://developer.mbed.org/compiler/

int main() {

float meas;

while(1) {

// Melhor configuração: placa com 100ms e gráfico com 50ms

// Em geral, sensor do arduíno lendo no dobro do tempo do computador (serial)

// A linha abaixo lê o valor da entrada A5 analógica A%

// O valor lido é entre 0.0 e 1.0

meas = analog\_value.read();

// Imprime o resultado na porta serial. Um por linha

printf("%f\n", meas);

// Espera 100 ms para ler o valor do sensor novamente

wait(0.1);

}

}

O arquivo cabeçalho mbed.h faz parte da plataforma mbed e, portanto, o código fonte do arquivo **main.cpp** deve ser compilado na plataforma online mbed. O arquivo binário resultado da compilação deve ser jogado driver indicado pelo sistema operacional quando a placa for conectada, de acordo com as instruções de instalação do driver fornecido na página da Freedom.

A aplicação backend que faz o tratamento dos dados, a classificação do sinal e a mudança de estado da torneira foi desenvolvida utilizando o Visual Studio 2013 Community Edition. Uma solução do tipo Windows Application foi criada a partir do uso da linguagem C#. A Figura 6 mostra a interface da aplicação, que contém um gráfico atualizado dinamicamente com os dados do sensor e botões para abrir a conexão serial (GetData) e treinar o sinal (). Esta aplicação utiliza alguns labels do lado direito da interface para mostrar o estado da torneira e a quantidade de água consumida.

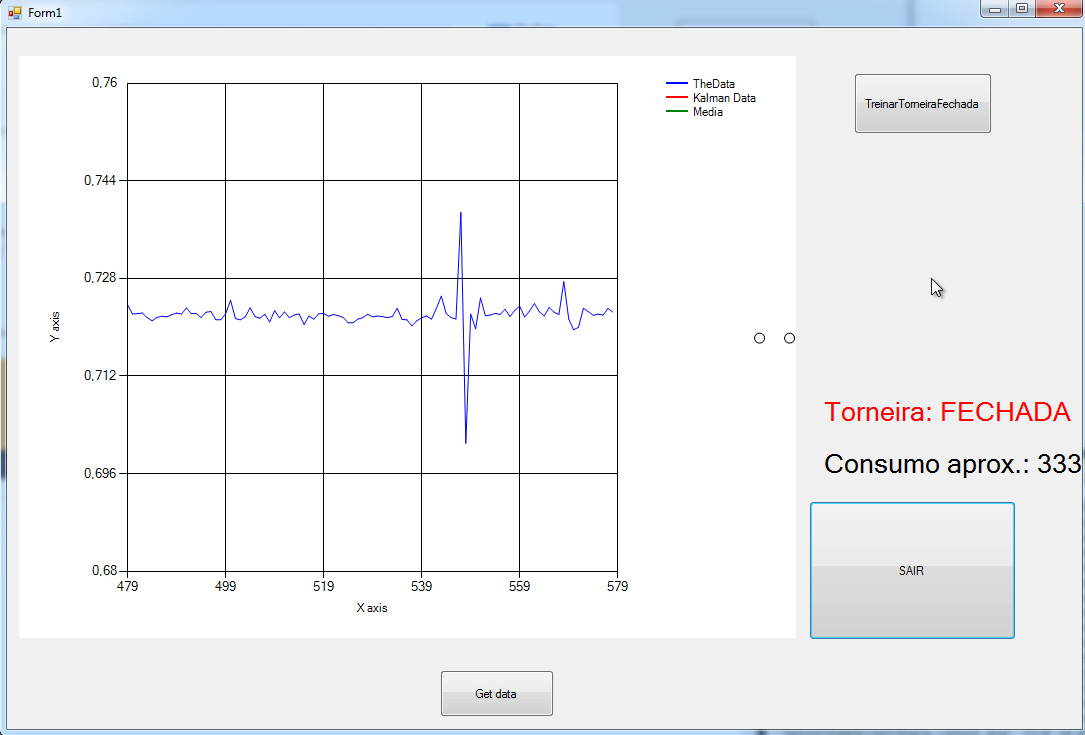


Figura 6. Interface da aplicação do protótipo do projeto AMedCa

O código fonte que vai dentro da placa como a aplicação desenvolvida na plataforma .NET estão disponíveis no seguinte repositório do GitHub: <https://github.com/pichiliani/AMedCa>.