

Capítulo IV – Uma Proposta de Medição de Nível de combustíveis em Reservatórios Comerciais

Márcio Barbosa de Lima¹²
Prof. MS.c Renato Kazuo Miyamoto¹³
Prof. Fabio Rodrigo Milanez¹⁴
Prof. Wesley Candido da Silva¹⁵

RESUMO

Os sistemas de medição de nível para tanque jaquetados submersos de combustível em postos de gasolina, geralmente utilizam uma central cujo custo de aquisição é elevado, devido aos equipamentos que são utilizados em sua construção. Este trabalho propõe a implementação de um protótipo para medição de nível desenvolvido com equipamentos de baixo custo, podendo reduzir os valores na instalação do sistema. Além disso, o protótipo tem a capacidade de coletar e armazenar os dados no ThingSpeak (nuvem) e conexão com o aplicativo Android. O protótipo conseguiu resultados satisfatórios, realizando a coleta de dados do reservatório e enviando para o ThingSpeak, gerando gráficos em tempo real de nível e porcentagem, além de ser monitorado através de um aplicativo Android.

Palavras-chave: Nível, Nuvem, Sistema IoT, Raspberry Pi.

¹² Pós-graduando em Automação Industrial E-mail: tecmicro2014@gmail.com

¹³ Docente Faculdade da Indústria SENAI Londrina.. E-mail: renato.miyamoto@sistemafiep.org.br

¹⁴ Docente Faculdade da Indústria SENAI Londrina.. E-mail: fabio.milanez@sistemafiep.org.br

¹⁵ Docente Faculdade da Indústria SENAI Londrina. e-mail: wesley.candido@sistemafiep.org.br

A Proposal for Measuring Fuel Levels in Reservoirs commercials

ABSTRACT

The metering central system of the fuel tank level has a much higher value, because of the equipments used in its construction. The prototype of this metering level were developed with low cost equipments, allowing to lower values in the system installation, even if the prototype has the capacity of collect and storing data inside the ThingSpeak and has a connection with an Android App, will reduce its value. This prototype succeeded collecting the reservoir data and send it to ThingSpeak, generating a graphic indicating the quantity and percentage, other than being monitored by an Android App.

Key-words: Level, Cloud, IoT System, Raspberry pi.

3. INTRODUÇÃO

O termo precisão de medidas é um fator importante para atender as necessidades de qualidade da instrumentação industrial. Tal termo está relacionado a diversos setores como farmacêutico, químico, alimentício, minerações e bebidas (LUIS, 2016).

Na história, grandes nomes marcaram suas contribuições. Em 1502 Leonardo da Vinci observou que a quantidade de água por unidade de tempo que escoava em um rio era a mesma em qualquer parte, independente da largura, profundidade e inclinação. Mas o desenvolvimento de dispositivos práticos só foi possível com o surgimento da era industrial e o trabalho de pesquisadores como Bernoulli, Pitot e outros (CASSIOLATA; ALVES, 2006).

A medição de fluidos está presente no dia a dia de muitas pessoas. Por exemplo no hidrômetro de uma residência, no marcador de uma bomba de combustível nos veículos, ou no medidor de níveis de tanques. A medição de nível de um tanque em um posto de combustíveis, por exemplo, consiste em um vaso jaquetado que possui sondas de nível e comunicação serial a uma central para a transmissão dos dados de leitura.

A automação industrial juntamente com a instrumentação está crescendo rapidamente devido ao aprimoramento das tecnologias, à necessidade de trabalho e a adaptação dos funcionários. Com isso, há a melhoria dos resultados de qualidade e eficiência, aliado a redução do tempo de processo.

A automação de tanques de combustíveis para a medição de nível tem grande importância em um processo, pois a precisão em medidas aliada a correta calibração acarreta em um rápido tempo de resposta, otimizando o processo produtivo, e assim resultando em melhorias ao consumidor e ao proprietário do posto.

Os sistemas automatizados são frequentemente encontrados nos ramos industriais, desde simples a complexos controles automatizados. Esses são baseados em uma interação entre software e hardware. Podemos citar softwares como os ambientes de desenvolvimento de programação tais como RSLogix 5000 e a IDE RaspBerry e softwares para a criação de interface Homem-Máquina, tal como o Visual Basic C#. Para os hardwares pode-se citar os microcontroladores, controladores e os circuitos integrados (CIs).

Em Weissmann, Douglas, et al, (2012), o autor utiliza três computadores, um HUB, um potenciômetro para calibração no monitoramento de nível. O projeto realiza a leitura de três tanques de combustível com sensores ultrassônicos e enviam dados para três IHMs (Computadores). Por meio da internet os dados são compartilhados com o cliente e com um servidor Web.

Em Luis, (2016) o autor realiza a medição de nível de líquido utilizando um microcontrolador Arduino, por meio de um sensor ultrassônico para monitorar o nível de combustível. O projeto teve as expectativas bem-sucedidas conseguindo verificar a medição e vazão de combustíveis com sucesso, porém não remetem às aplicações industriais.

Atualmente, os equipamentos que realizam medição dos tanques de combustíveis possuem uma central não automatizada que usando processo manual para impressão dos dados do tanque, acarretando em dependência de um colaborador para execução da impressão do histórico, controle e tomada de decisões.

Já existem centrais automatizadas que realizam a leitura e comunicação com um aplicativo de celular por meio da conexão com a internet podendo ter o controle em tempo real. Porém, o elevado custo destas centrais se torna um ponto negativo.

Este trabalho tem como objetivo implementar um controle automatizado de nível em postos de combustível por meio do microcomputador Raspberry. O projeto oferece opção de baixo custo quando comparado a soluções existentes no mercado, e poderá ser acessado em tempo real por meio de um controle supervisionado. Para criação do Protótipo foram usados os conceitos de microcontroladores, microcomputadores e sensores como veremos a partir do tópico 2.

4. FUNDAMENTAÇÃO TEÓRICA

Nos postos de combustíveis é utilizado sistemas de medição de nível como o sistema da Excelbr que tem o ELS que monitora os tanques com sondas e são capazes de perceber a inclinação de um tanque e mandar os dados online a quantidade de combustível exata usando assim utilizando o conceito IoT (Internet das Coisas). Disponível em: < <http://excelbr.com.br/medicao-de-tanque-e-monitoramento-ambiental/>>

Um outro sistema que é utilizado no mercado é o da OPW Console SiteSentinel Nano que traz uma tela interativa e conexão com a internet mostrando o monitoramento em tempo real.

Disponível em: < <http://www.opwbrasil.com.br/produto/console-sitesentinel-nano/> >

A grande dificuldade dos comerciantes de combustíveis na implantação desse sistema de monitoramento é o custo. Assim, a criação do protótipo que será apresentada nos próximos tópicos, tem um custo mais reduzido.

2.1 SISTEMAS EMBARCADOS

Sistemas embarcados são dispositivos que podem ser encontrados em qualquer lugar, de aplicações residenciais a controladores de processos críticos como aviação ou equipamentos médicos.

Almeida, Rodrigo Maximiano et al. (2016) afirma que: “Outro grande motivador é o advento da internet das coisas, plataformas embarcadas que possuem acesso à internet, várias empresas multinacionais vêm investindo nesta área (Intel, Oracle, Advantech, Dell) e as universidades começam a perceber a escassez de mão de obra no mercado”. O sistema embarcado utilizado nos microcontroladores podem substituir as centras de medição existentes com o mesmo desempenho e menor custo.

2.2.1 Microcontroladores e circuito de prototipagem

Os microcontroladores são circuitos integrados programáveis, que podem ser conectados a sensores, atuadores em seus pinos de entrada e saída. Segundo Pietro (2004, cp.1 pág. 15), um microcontrolador é um CI capaz de efetuar processos lógicos com extrema rapidez e precisão. A grande vantagem deste CI é a sua possibilidade de programação, o que o torna adaptável à finalidade desejada, e que possibilita seu ajuste de acordo com a tarefa que deverá executar.

O Arduino é um circuito que foi desenvolvido com o objetivo de ajudar estudantes e pesquisadores na prototipagem de *software*, podendo ser usado em projetos. Se trata de uma plataforma aberta com grande possibilidade de comunicação entre Shields e sensores. Neste projeto, utiliza-se o Arduino como interface de

conversor A/D cujo objetivo é realizar a leitura do sensor e trocar dados com o Raspberry Pi. A placa Arduino UNO está ilustrada na Figura 1.

Figura 1: Plataforma Arduino UNO



Fonte: Autoria própria (2019)

A Tabela 1 apresenta as características e especificações da plataforma Arduino UNO.

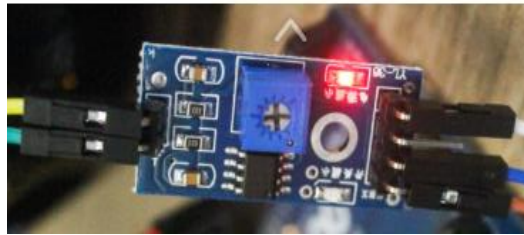
Tabela 1: Características do Arduino UNO

Corrente de entrada	5v
Corrente de saída	3.3 – 5v
Microcontrolador	Atmel
Entrada/Saída Digital	14
Entrada/Saída Analógica	5

2.2.1 Sensores

Em sistemas de controle de variáveis de processo, se faz necessário a utilização de sensores para aquisição dos sinais da variável controlada. Assim, para esta pesquisa foram utilizadas um módulo de sensor de umidade e uma boia de nível resistiva. Assim, o sensor de modelo LM393 utilizado nesta pesquisa está ilustrado na Figura 2.

Figura 2: Módulo Sensor de Umidade do Solo.



Fonte: Autoria própria (2019)

O referido sensor possui uma saída digital (D0), e uma saída analógica (A0). O sinal digital é ajustado através de um potenciômetro presente na placa comparadora. A Tabela 2 informa as especificações do sensor.

Tabela 2: Características do módulo LM393

Tensão	3.3 – 5v
Sensibilidade	Ajustável via potenciômetro
Saída	TTL (DO) e analógica (AO)
Dimensões	30mm x 16mm
GND	GND

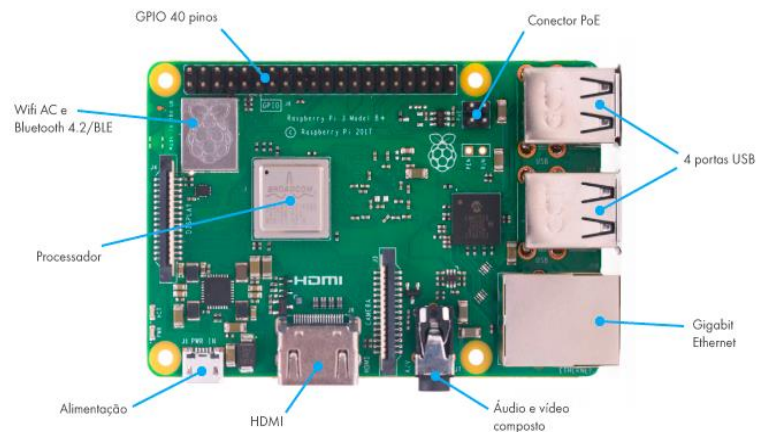
Fonte: Baú da Eletrônica

2.4 Microcomputador Raspberry Pi 3

Com os avanços na tecnologia na área de *hardware*, os dados que antes eram transmitidos por computadores *desktops* hoje podem ser facilmente tratados por computadores que cabem na palma da mão, resultando em espaço e mobilidade ao controlar circuitos em pequenos espaços físicos.

A Raspberry Pi Foundation (Fundação Raspberry Pi), criou o microcomputador no ano de 2009 e tendo uma evolução contínua sua última versão é o Raspberry Pi 3B+ que suporta sistemas operacionais e traz a IDE para programação em Python e com um *hardware* capaz de processar e armazenar dados de forma eficiente uma grande vantagem é seu tamanho físico que facilita a instalação em pequenos ambientes como ilustra a Figura 3.

Figura 3: Microcomputador Raspberry PI 3



Fonte: Filiflop

Podemos verificar as especificações técnicas do Raspberry Pi 3 na Tabela 3.

Tabela 3: Características do Microcomputador

Entrada de Fonte DC micro USB	5v/2.5A
Processador Broadcom	BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
Rede	wireless LAN(RJ45), Bluetooth
USB	4 Portas
GPIO	40 PINOS
CONECTOR CSI/DSI	Câmera/display capacitivo
Saída de vídeo/áudio	Plug P4
Slot cartão micro SD	Para carregamento do OS e armazenamento de arquivos
Transmissão de Vídeo	HDMI

Disponível em: <<https://blog.fazedores.com/nova-raspberry-pi-3-b-plus/>>

2.5 SISTEMA IoT

O sistema IoT (*Internet of Things*) possibilitou o controle dos equipamentos através da internet, podendo efetuar ações e monitoramento por meio de smartphones, tablet, notebook e computadores. Com o auxílio de aplicativos, é possível efetuar acionamento de sistema de irrigação em jardins de residências e bem como monitorar o nível de um tanque de combustível em um posto.

“...O sistema IoT segundo Ashton, (1999) Se tivéssemos computadores que soubessem tudo sobre as coisas em geral - usando dados que coletassem sem a nossa ajuda - seríamos capazes de rastrear e contar tudo, e reduzir bastante o desperdício, a perda e os custos. Nós saberíamos quando é necessário substituir, reparar ou fazer um recall de um produto, e se estão

novos ou ultrapassados. Precisamos capacitar os computadores com seus próprios meios de coletar informações, para que possam ver, ouvir e cheirar o mundo sozinhos, com toda a sua glória aleatória. O RFID e a tecnologia de sensores capacitam os computadores a observar, identificar e entender o mundo sem as limitações dos dados inseridos pelos humanos.”

2.5.1 Web Server

Usando o Aplicativo que foi desenvolvido para plataforma Android podemos fazer com que o cliente acesse os seus dados com mobilidade através do Aplicativo criado para esse objetivo, podemos observar as telas do Aplicativo, na Figura 4.

Figura 4: Aplicativo para Smartphone



Elaborado pelo Autor (2019)

Dessa forma pode-se visualizar o nível do tanque de forma rápida e de onde o cliente estiver.

2.5.2 Nuvem da internet

Com a chegada da tecnologia na área de automação industrial, a quantidade de transmissão e recepção de dados tem aumentado continuamente no processo de produção. A definição de nuvem é uma estrutura de computadores que proporciona serviços aos seus usuários através da Internet.

Como a necessidade de armazenamento em um Big Data várias empresas surgiram oferecendo o serviço em Nuvem (Cloud) como por exemplo: Dropbox,

Google Drive e ThingSpeak da Matlab. A metodologia empregada na coleta de dados e no desenvolvimento deste projeto será relatado no Capítulo 3.

3 METODOLOGIA

Este estudo é quantitativo, pois serão coletados dados do nível de combustível de um tanque, onde serão enviados a nuvem para geração de gráficos para monitoramento. Os testes foram realizados em um protótipo experimental.

3.1 Montagem do Protótipo

Para a montagem do protótipo foi conectado da porta USB do Arduino com a porta R232 do Raspberry. O Arduino foi conectado com o módulo sensor de umidade (PCB) através da porta de Entrada Analógica A0 e conexão de tensão 3.3v e GND.

O Raspberry pi 3 foi usado como Mestre e o Arduino Uno como escravo. Assim, o arduino possui como função receber o sinal do sensor de umidade conectado aos sinais da boia enviando para o Raspberry pi 3 conforme (Apêndice A). Com os sinais coletado a traves de código o Raspberry pi 3 faz o tratamento dos sinais e envia para o ThingSpeak (Nuvem) conforme (Apêndice B).

3.3 Programação de monitoramento via WiFi

Por meio do Raspberry pi 3 com a wi-fi on-board é feita a comunicação de monitoramento dos dados coletados que são armazenados no Web Service. Foi criado Web Server, com código fonte desenvolvido na IDE Spring Tools Suíte na linguagem Java. O Web Server foi hospedado na plataforma FireBase, para a conexão com o ThingSpeak (Conforme Apêndice D).

Para execução do monitoramento do nível de líquido do reservatório foi utilizado um raspberry como mestre e um microcontrolador Arduino UNO como escravo. Foi utilizado um módulo de umidade e um sensor resistivo de nível de veículo Monza 87. No Microcontrolador Arduino UNO foi desenvolvido um código para receber o sinal enviado pelo sensor de umidade que está conectado ao sensor (boia).

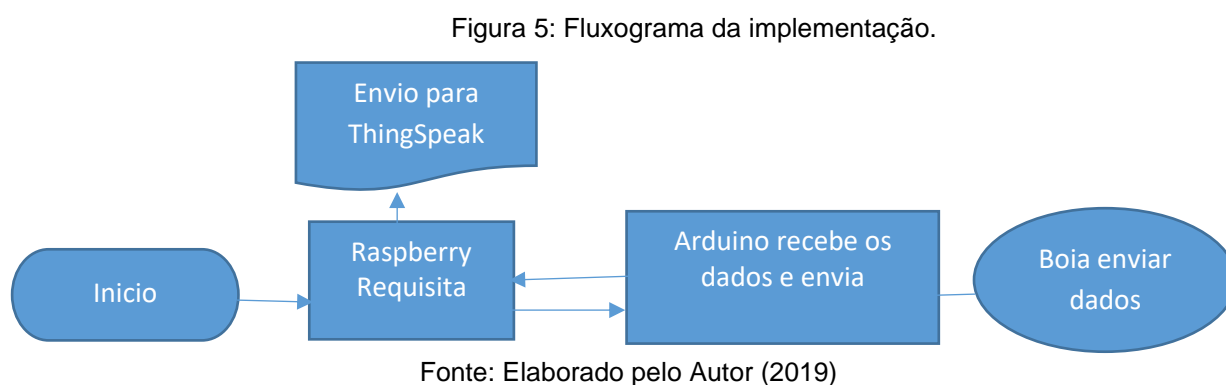
(Conforme Apêndice D). Para o envio de dados a nuvem, foi criado um código fonte de rotina para efetuar a conexão com a cloud da ThingSpeak.

Na comunicação com ThingSpeak foi implementado, um código de programação responsável por fazer a aquisição dos valores em litros e o percentual do reservatório. Foi necessário a implementação de comando para possibilitar o envio de dados a nuvem e gerar gráficos.

3.3.2 Nuvem ThingSpeak

Foi usada uma nuvem para gerar gráficos e armazenar os dados. A nuvem é da empresa Matlab e foi necessário a criação de uma conta para acesso. Após a confirmação da conta, foi criado 1 canal e 2 campos para a geração dos gráficos de litro e porcentagem. Cada canal é identificado por uma numeração e possui chaves escritas para a sua identificação. Esses dados são inseridos no código fonte para efetuar a comunicação com a nuvem.

Para armazenar os dados coletados à nuvem, são exportados manualmente em formato de arquivo JSON, XML, orCSVdata para uma região de armazenamento conforme ilustra o diagrama da Figura 5.



4. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

Neste tópico são apresentados, os resultados do envio de dados monitorados a nuvem, obtidos com os testes aplicados no protótipo apresentado neste trabalho.

4.1 Monitoramento do Nível

A comunicação do Raspberry, com a rede de WiFi, não ocorreu nenhum tipo de ruído, para a verificação foi usado o software para requisição dos dados IDE Python como mostrado na Figura 6, código conforme (Apêndice D).

Figura 6: Dados requisitados e enviados por código python

```

import serial
import time
import http.client as httpplib, urllib

32
33 headers = {"Content-type": "application/x-www-form-urlencoded", "Accept": "text/plain"}
34
35 conn = httpplib.HTTPConnection("api.thingspeak.com:80")
36 conn.request("POST", "/update", params, headers)
37

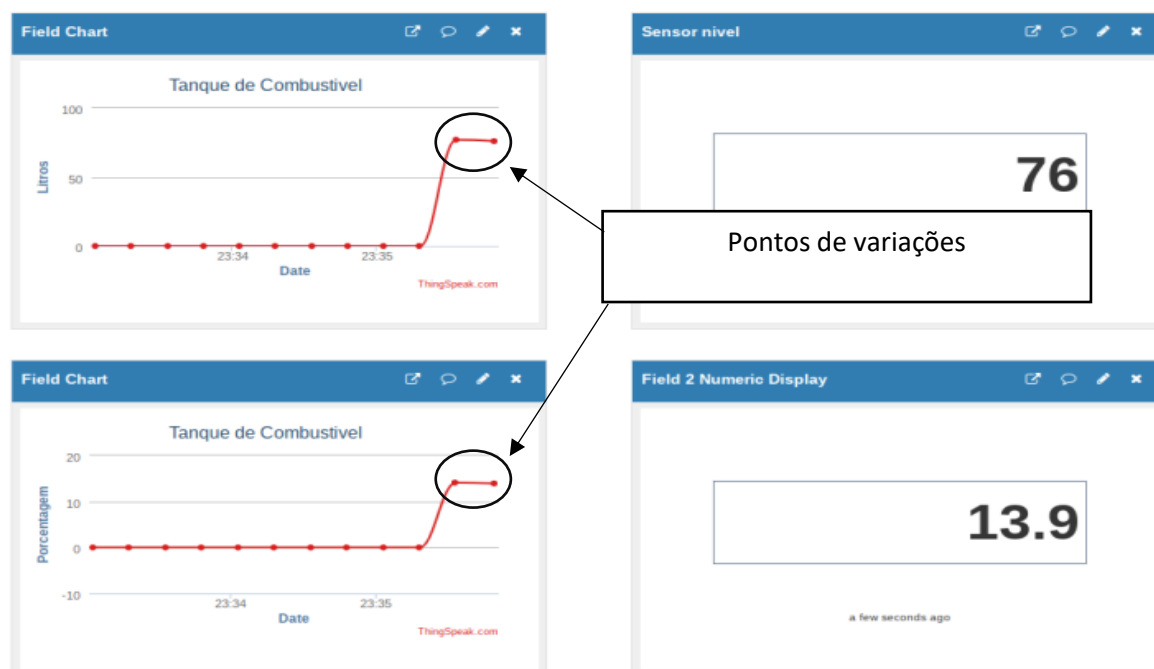
```

Fonte: Elaborado pelo Autor (2019)

Onde o parâmetro Headers é responsável do tipo do conteúdo que será utilizado e Conn request = especifica o método http, caminho, parâmetros e cabeçalho.

O Shell do Python gerou informações necessárias para enviar para o ThingSpeak e assim podendo monitorar o nível do tanque. Os sinais aqusitados em tempo real são ilustrados na Figura 7.

Figura 7: Gráfico da quantidade de Litros no recipiente



Fonte:Elaborado pelo Autor (2019)

Assim a Figura 7 ilustra a aplicação no Smartphone com o Web Server e com o ThingSpeak (Nuvem) trazendo os dados em tempo real e com a possibilidade de retirada de relatório mostrando os valores mensurados: litros, porcentagem, data e hora.

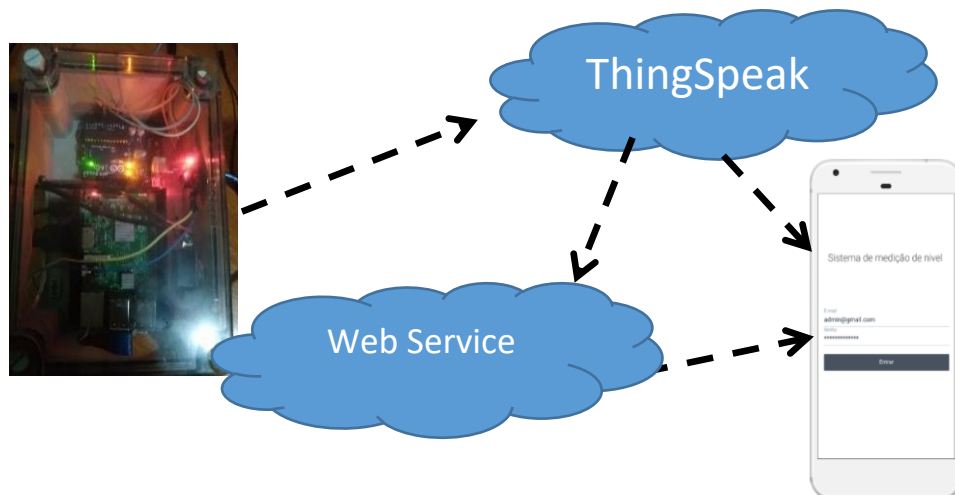
4.2 Monitoramento e aquisição de dados

Os dados coletados pelo protótipo de monitoramento foram enviados a nuvem e gerados os gráficos de litros consumido e porcentagem em tempo real, onde também foi utilizado um alicate amperímetro para comparação dos dados coletados.

Os dados coletados na quantidade de litros, são relatados conforme o horário de acionamento. Deste modo, percebe-se na Figura 7, que as 23:35pm, o valor em Litros era de 76. Observa-se uma leve variação que foi causada pelo sensor (boia) utilizada, houve várias tentativas de calibragem do sinal porem não foi corrigido a variação. Assim, a Figura 8 ilustra a metodologia de conexão entre os dispositivos e o banco de dados.

Figura 8: Conexão entre os dispositivos e o banco de dados

Protótipo



Elaborado pelo Autor (2019)

5. CONSIDERAÇÕES FINAIS

O Protótipo teve uma conexão estável dos dispositivos e em sua montagem o Raspberry se comunicou sem problemas com o Arduino, que coletou os dados do sensor (boia), através do modulo de comparador de tensão. O referido modulo realiza a leitura da tensão enviada pelo sensor (boia). Com o ciclo do processo completo foi feito um tratamento dos dados através de programação Python.

A comunicação do código fonte com ThingSpeak foi automática apresentando gráficos a cada mudança de nível do reservatório, com os dados na nuvem foi realizado uma comunicação com o Firebase (Banco de Dados).

O aplicativo criado para comunicação remota inicialmente funcionou normalmente, mas ao longo dos dias com o dispositivo ligado e enviando dados para o banco de dados, pode-se perceber o travamento do app e pelo fato da conta criada no FiriBase (Banco de Dados) ser gratuita a quantidade de Armazenamento era limitada.

No entanto ouve muito ruído no sensor utilizado (boia automotiva) podendo ser melhorado com a mudança do sensor que não foi utilizado nesse projeto pelo custo, o uso do microcontrolado SP32.

REFERÊNCIAS

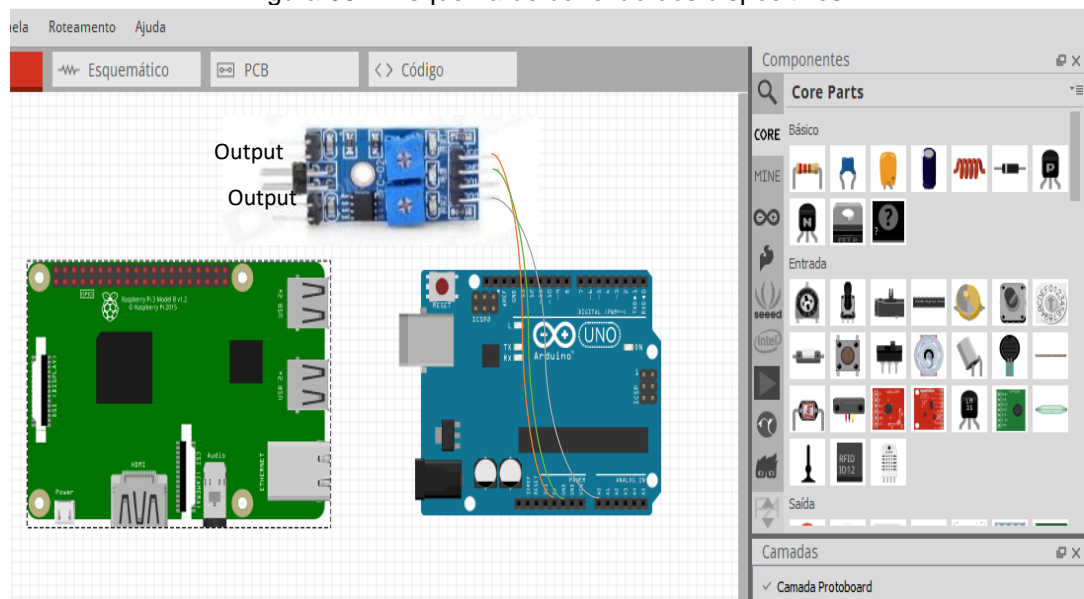
- ASHTON K. **A Coisa da Internet das Coisas**. Disponível em: <https://www.cisco.com/c/dam/global/pt_br/assets/brand/iot/iot/pdfs/lopez_research_an_introduction_to_iiot_102413_final_portuguese.pdf>, Acesso em: 14/03/2019.
- ALMEIDA, R. et al. **Programação de Sistemas Embarcados**. São Paulo: Elsevier, 2016.
- CASSIOLATO, C.; ALVES, E. **Mecatrônica Atual**. 5ed. São Paulo: Ed Saber, 2012.
- DIOVANE P. **MICROCONTROLADOR** 01-12-2004 Disponível em: < http://www.unipac.br/site/bb/bb_tcc_res.php?id=352> Acesso em 14/03/2019.
- LUIS, D. Sistema de Controle de Medição de Vazão para Bombas Peristálticas. **Monografia - Universidade Federal Ouro Preto**, 2016.
- LEAL, F. **O que é a Nuvem**. _____ Disponível em: < <https://www.fabricadejogos.net/posts/artigo-o-que-e-nuvem/>> Acesso em: 14/03/2019
- WEISSMANN D. et al, Sistema de Monitoração Remoto do Volume de Combustível em Tanques de Posto de Abastecimento. **Monografia - Universidade de Taubaté**, 2012.

APÊNDICES

Apêndice A

A Figura 08A apresenta a montagem do Protótipo.

Figura 08A: Esquema de conexão dos dispositivos.



Elaborado pelo Autor (2019)

Apêndice B

A Figura 09B apresenta o canal criado no ThingSpeak para apresentação dos dados.

Figura 09B: Informações do canal

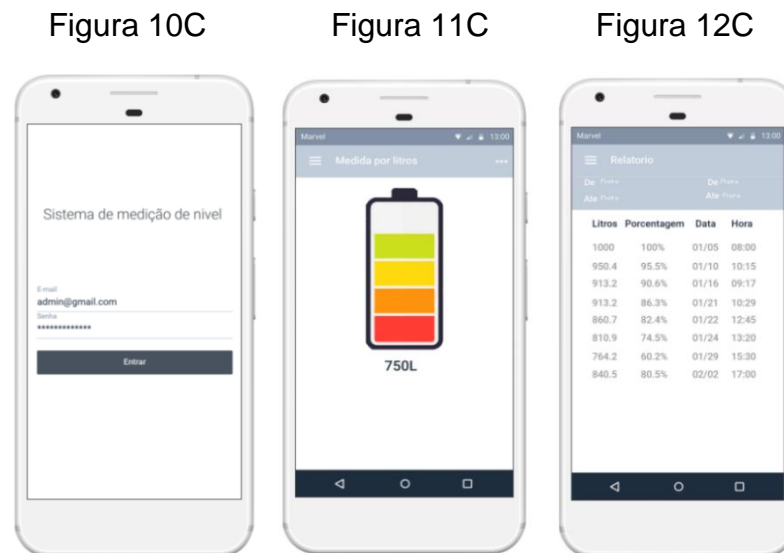
Nome	Criada em	Updated
Tanque de Combustível	2019-03-05	2019-03-16 18:22

Private Públicos Settings Sharing API Keys Data Import / Export

Elaborado pelo Autor (2019)

Apêndice C

As Figuras 10C, 11C e 12C mostram o aplicativo de monitoramento, apresenta a tela de log (Figura 10C), a tela ilustrativa da quantidade de combustível (Figura 11C) e a tela de relatório (Figura 12C).



Elaborado pelo Autor (2019)

Apêndice D

A Figura 13E apresenta o código que foi implementado para comunicação com sensor.

Figura 13E: Código Fonte para leitura de dados do se

```

nível_arduino | Arduino 1.8.8
Arquivo Editar Sketch Ferramentas Ajuda
nível_arduino
#define pinBoia A0

int valor;

void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println(analogRead(pinBoia));

  valor = analogRead(pinBoia);
  valor = map(valor, 500, 0, 0, 100);
  Serial.print(valor);
  delay(5000);
}
Carregado.
  
```

Elaborado pelo Autor (2019)

Apêndice D

O Código de comunicação do Raspberry com o Arduino e com o ThingSpeak apresentado na Figura 14D

Figura 14D: Código Fonte Python

```

1 import serial
2 import time
3 import http.client as httpLib, urllib
4
5 comunicacaoSerial = serial.Serial('/dev/ttyACM0', 9600)
6
7 apiKey = "K27DMMNC5XY9R47G"
8
9 def p(porcentoal):
10     valorAtual = int(porcentoal)
11     valorMax = 500
12
13     if valorAtual <= valorMax:
14         return (valorAtual*100)/valorMax
15     elif valorAtual > valorMax:
16         return (valorMax*100)/valorMax
17
18 while True:
19     valor = comunicacaoSerial.readline()
20     #print(valor.decode("utf-8"))
21     litros = float(valor.decode("utf-8"))
22     porcento = float(valor.decode("utf-8"))
23
24     .
25
26     n = p(porcento)
27
28     print(litros)
29     print(n, "%")
30
31
32     params = urllib.parse.urlencode({'field1': litros, 'field2': n, 'key': apiKey})
33
34     headers = {"Content-type": "application/x-www-form-urlencoded", "Accept": "text/plain"}
35
36     conn = httpLib.HTTPConnection("api.thingspeak.com:80")
37     conn.request("POST", "/update", params, headers)
38
39     response = conn.getresponse()
40
41     print(response.status, response.reason)
42     print("")
43
44     conn.close()
45
46     time.sleep(0.5)
47

```

Elaborado pelo Autor (2019)