

Desenvolvimento de um sistema de monitoramento em tempo real baseado em Arduino e Raspberry

Development of a real-time monitoring system based on Arduino and Raspberry

Lucas Alvarez Nogueira¹, Caison Rodrigues Ramos², Leonardo Carillo Bahia³ e Celso Becker Tischer⁴

^{1,2,3,4}Universidade Federal de Santa Maria, Coordenadoria Acadêmica, Cachoeira do Sul, Brasil
lukas_nogueira@hotmail.com; caisonrr@hotmail.com; leo.cariillo@hotmail.com; cbtischer@gmail.com

Resumo

Atualmente, investidores de diversos setores industriais buscam melhorias de processos que visam rentabilidade e maior competitividade no mercado, porém, características como custo, agilidade de informação e produção são pontos determinantes para tal. Plataformas de controle de entrada e saída open source que integram microcontroladores vem ganhando espaço em diversos setores, no qual, programados apresentam capacidade de controlar processos e gerenciar dados. Desta forma, este artigo apresenta o desenvolvimento de um sistema de monitoramento em tempo real de grandezas físicas baseado em Arduino e Raspberry. O sistema completo será composto por dois módulos, "Escravo" e "Mestre" no qual a transferência de dados das medidas de temperatura, umidade, luminosidade, pressão e entre outras será realizada. O escravo, formado por sensores de baixo custo conectados à um Arduino MEGA 2560, será alocado em local remoto para aquisição das grandezas físicas desejadas. Já o Mestre, receberá as informações através de uma shield ethernet, conectada em ambos os módulos, e fará o gerenciamento de banco de dados. As informações armazenadas no servidor web poderão ser acessadas via navegador de internet, seja em sistema desktops ou móveis por meio de uma plataforma amigável em tempo real.

Palavras-chave: Arduino; Raspberry; Monitoramento

Abstract

Actually, investors from several industrial sectors are looking for process improvements that aim at profitability and greater competitiveness in the market, however, characteristics such as cost, information and production agility are determining points for this. Open source input and output control platforms that integrate microcontrollers have been gaining ground in several sectors, where programmers have the ability to control processes and manage data. In this way, this article presents the development of a real time monitoring system of physical quantities based on Arduino and Raspberry. The complete system will consist of two modules, "Slave" and "Master", in which the data transfer of the measurements of temperature, humidity, luminosity, pressure and others will be carried out. The slave, formed by inexpensive sensors connected to an Arduino MEGA 2560, will be allocated at a remote location to acquire the desired physical quantities. The Master, will receive the information through an ethernet shield, connected in both modules, and will manage the database. The information stored on the web server can be accessed via internet browser, either in desktops or mobile systems through a friendly platform in real time.

Keywords: Arduino; Raspberry; Monitoring

1 Introdução

Em diversos ramos de trabalho é estritamente necessário o monitoramento do ambiente, como por exemplo, data centers, câmaras frias, granjas e em previsões do tempo. Os sistemas desenvolvidos hoje para esta finalidade apresentam muitas vezes softwares e protocolos proprietários, sendo em sua maioria, de alto custo, de difícil instalação e utilizam cabeamento para a transmissão de dados até o módulo de processamento. (MELLO, 2017, DONATÉ, 2016).

Com a crescente utilização de smartphones, desktops e notebooks, permitiu-se maiores facilidades em acessar informações dispondo apenas de uma rede wireless. Tal inovação possibilita hoje monitorar e atuar em processos e gerenciar dados em tempo real. Nesta perspectiva, o monitoramento de processos que envolve diversos ramos, permite analisar e verificar o andamento do sistema com dados em tempo real que informa as condições de uma dada grandeza. Com isso, é possível identificar, corrigir falhas e até mesmo atuar, como supervisor, sem afetar o funcionamento do processo de maneira preventiva, sendo que na maioria dos casos envolvem custos.

As aplicações típicas em um sistema de supervisão são divididas em três grupos, tais como: Controle, que permite o gerenciamento do fluxo do sistema de potência e confiabilidade; Medição: Admite a apresentação dos dados coletados dos sistemas em forma de relatório por meio de supervisão e controle de dados (SCADA), bem como armazenamentos dos dados coletados para posterior análise e previsões de eventos; e Gerenciamento de Ativos, que busca informações nos equipamentos relativas a histórico de operação favorecendo ações de manutenção preditiva.

Assim, nos últimos anos o uso de microcontroladores nas áreas de tecnologia teve crescimento considerável devido a suas pequenas dimensões, capacidade de controlar processos e permitir o processamento de dados, proporcionadas por novas linguagens de programação e ao fácil acesso à informação por meio da internet. A aplicação de microcontroladores é significativa em diversos setores, tais como: industriais, domésticas, entretenimento, telecomunicações, automotivas, transporte e aviação (SARTORI 2015; ALCIATORE, 2014).

Um microcontrolador é um sistema com várias funções em um único chip, entre elas estão: gerador interno de clock, memórias, conversores analógicos-digitais, digitais-analógicos, temporizadores, contadores, comparadores analógicos, saídas PWM (Pulse Width Modulation), interfaces de comunicação e periféricos especializados de acordo com cada fabricante (ARDUINO, 2018). Sua utilização agregada a uma pequena quantidade de componentes eletrônicos permite desenvolver diferentes tipos de sistemas em um curto período de tempo (CABÓ, 2015), no qual, pode ser encontrado no Arduino.

Plataformas de controle de entradas e saídas open-source que integram microcontroladores, chamada Arduino, vem ganhando espaço em processos e na utilização de sistemas embarcados, no qual, são desenvolvidos para exercer uma funcionalidade durante sua vida útil (EVANS, 2013). Assim, pretende-se com esse projeto desenvolver um sistema de monitoramento de ambiente utilizando interface web, permitindo o acesso direto as informações via navegador de internet, aos quais verificam constantemente as atuais condições de temperatura, umidade, luminosidade, pressão entre outros de um local específico. Tal sistema será baseado na utilização de Arduino e Raspberry.

2 Arduino

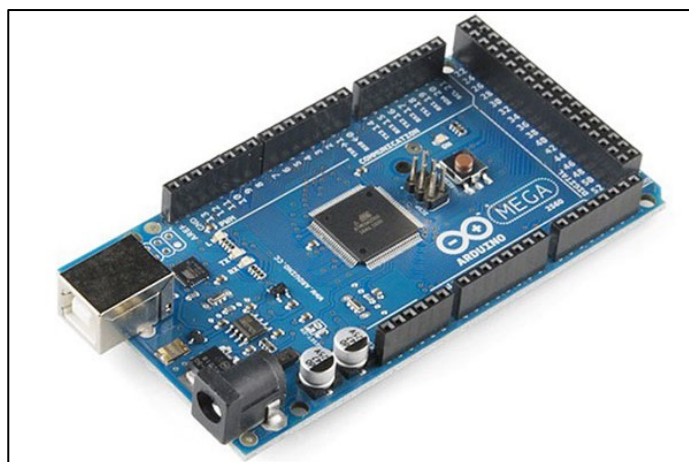
Arduino é uma plataforma de prototipagem eletrônica baseada em *software* e *hardware* livre. Esta plataforma pode perceber o ambiente ao seu redor através de sensores e atuar sobre ele por meio de atuadores. Tal equipamento é disponibilizado na forma de placas para o desenvolvimento de dispositivos diversos, onde sistemas digitais, ligados a

sensores e atuadores, formam uma placa de circuito impresso, possibilitando medir grandezas do meio físico, realizar cálculos numéricos e tomar decisões lógicas programadas corretamente.

Neste projeto será utilizado o Arduino Mega2560, conforme mostrado na Figura 1, devido seu baixo custo e capacidade de atender as necessidades do sistema. Esse modelo possui um microcontrolador ATmega2560, 54 pinos de entradas/saídas digitais, 16 entradas analógicas, 4 portas seriais de hardware, um oscilador a cristal de 16MHz, uma conexão USB, uma entrada de alimentação, uma conexão usada para gravar o software no microcontrolador e um botão de reset.

O microcontrolador ATmega2560, possui 256KB de memória flash para armazenamento de código, 8KB de SRAM e 4KB de EEPROM. Cada um dos 54 pinos digitais pode ser utilizado como uma entrada ou saída. Eles operam com uma tensão de 5 volts para nível alto e 0 volts para nível baixo. Cada pino pode fornecer ou drenar no máximo uma corrente elétrica de 40mA (Miliampères). Porém a soma das correntes não pode ultrapassar 200mA.

Figura 1 - Arduino Mega 2560



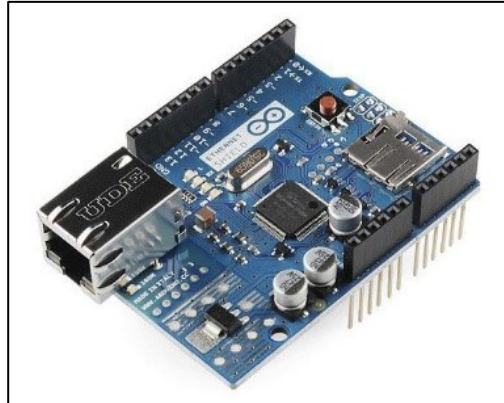
2.1 Linguagem e ambiente de programação

A linguagem de programação para o *Arduino* é baseada nas linguagens C/C++, no qual os programas são chamados de *sketch*, que contém as linhas de códigos de instrução que são compiladas, enviadas e executadas para o *Arduino*. Um *sketch* típico consiste de duas partes ou rotinas: a primeira é a rotina de inicialização chamada *setup*, e a segunda é a rotina chamada *loop*, que geralmente contém o corpo principal do código.

Para a realizar a gravação de dados no *Arduino* uma das possibilidades é a utilização da memória EEPROM do microcontrolador. Com esse método de armazenamento, os dados gravados não se perdem quando acontece uma queda de energia.

A integração de comunicação do *Arduino* com qualquer outra aplicação de linguagem diferente se faz através dos componentes de comunicação que o *Arduino* dispõe, no trabalho proposto o *Arduino* possui o componente *Ethernet Shield*, mostrado na Figura 2, o que possibilita a comunicação com qualquer plataforma via rede.

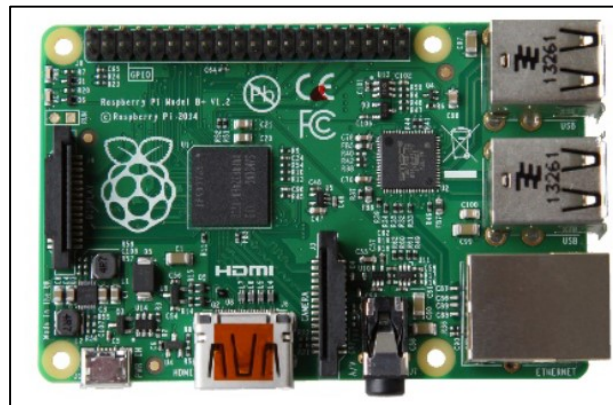
Figura 2 - Módulo Ethernet Shield.



3 Raspberry pi

O Raspberry Pi é um computador single-board com as dimensões de um cartão de crédito no qual é possível conectar periféricos como mouse, teclado e monitor, detalhado na Figura 3. Tal equipamento foi construído como uma alternativa de baixo custo e proporciona ao usuário, perante a instalação de um sistema operacional, a construção de tabelas de texto, a criação de softwares, a navegação na internet e outras funcionalidades de computadores de mesa.

Figura 3 – Raspberry Pi 3 Model B

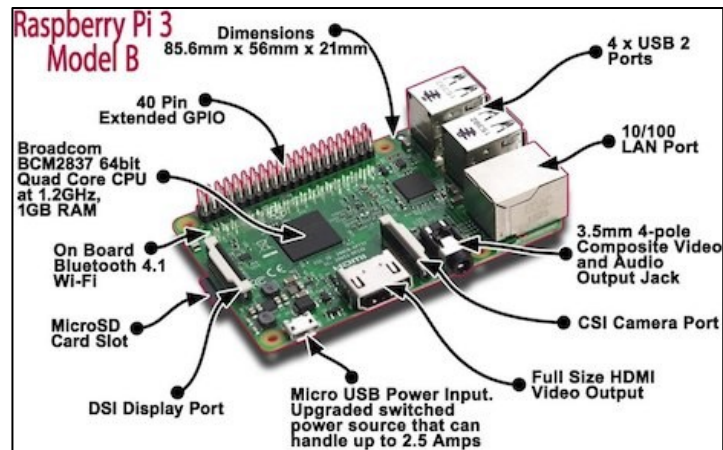


Inicialmente o Raspberry Pi foi desenvolvido por alunos da Universidade de Cambridge para incentivar o estudo da Ciência da Programação básica em escolas do Reino Unido. Mesmo com configurações de hardware limitadas em comparação aos notebooks e desktops, o Raspberry ultrapassou a barreira do aprendizado e alcançou aplicações na área de engenharia. Devido suas especificações e seu modelo construtivo é possível utiliza-lo como um sistema embarcado (RICHARDSON e WALLACE, 2013). A designação Pi se deve a linguagem de programação idealizada para ser mais utilizada o Python. Todavia é possível implementar qualquer linguagem de programação na placa.

Sua composição de hardware mostrada na Figura 4, possui portas USB, portas HDMI, slot para Cartão de Memória, processador, entrada para câmera e portas GPIO, onde é possível acoplar diversos sensores. Na maioria dos modelos há entradas para módulos *bluetooth*, no entanto, no Raspberry Pi 3 Model B, tais módulos já são acoplados a placa.

Há uma grande variedade de sistemas operacionais para o Raspberry, a mais utilizada é o Raspbian, uma derivação do Linux Debian. Porém é possível instalar versões de Windows 10, Ubuntu e Risc OS.

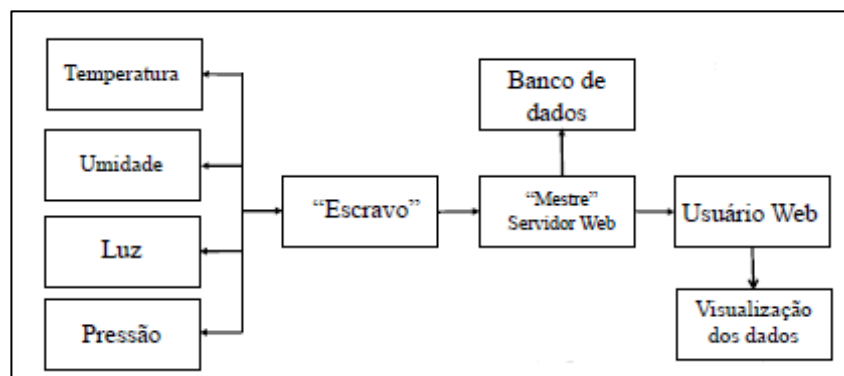
Figura 4 – Composição do Raspberry Pi 3 Model B



4 Descrição geral do sistema

O projeto será dividido em três fases, sendo a primeira medição de sensores, segunda chamada Escravo, para aquisição dos sinais provenientes dos sensores com base em Arduino, e terceira, chamada Mestre, responsável pelo servidor web como mostrado na Figura 5.

Figura 5 - Fluxograma do sistema de monitoramento



4.1 Sensoriamento

Nesta fase, será realizado a programação do microcontrolador presente no Arduino para capturar as grandezas medidas pelos sensores, bem como, para gerar um arquivo de saída padronizado por um protocolo de comunicação a ser utilizado no banco de dados da interface web. No projeto será utilizado sensores para realizar leituras específicas dos ambientes como temperatura, umidade, pressão e luminosidade. Após algumas análises comparativas de sensores, como mostrado na Tabela 1, o sensor de temperatura e umidade chamado DHT 11, mostrado na Figura 6, se destaca, pois permite a medição de duas grandezas físicas e apresenta baixo custo, além de oferecer manutenção reduzida nas aplicações existentes.

O sensor de pressão verificado para a aplicação no projeto é o BMP 280 mostrado na Figura 7, além de possuir ganhos em termos de precisão, tem como característica principal o baixo consumo de energia, permitindo funcionar por longos períodos.

Tabela 1: Comparativo de sensores de temperatura e umidade

Tipos	DHT 11	DHT 22	LM 35	HS15P
Faixa de leitura (Umidade)	20-80%	0-100%	-	20-100%
Precisão (umidade)	5%	5%	-	5%
Faixa de leitura (temperatura)	0/50 °C	-40/125 °C	-55/150°C	-
Precisão (temperatura)		+/- 0.5%	+/- 0.5%	-
Alimentação	3-5.5V	3.3-6V	4-30V	

Figura 6 - Sensor DHT 11

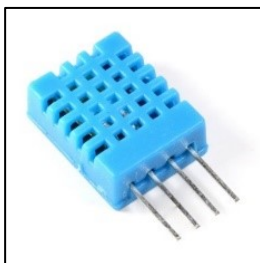
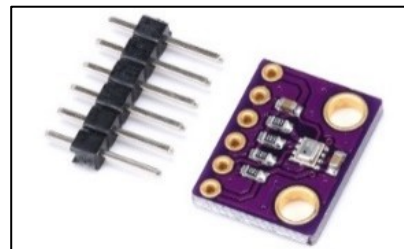


Figura 7 - Sensor De Pressão BMP280

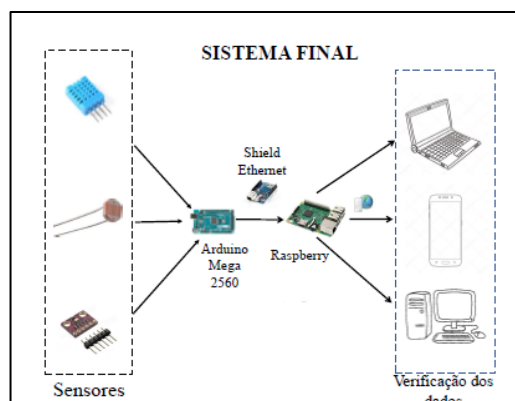


4.2 Sistemas Escravo/Mestre

O modulo escravo será composto por Arduino Mega 2560, por ter facilidade na ligação com os sensores e na possibilidade de futuras expansões do projeto. O módulo mestre do projeto será o Raspberry Pi 3 Model B. O mestre receberá as informações medidas pelos escravos, Arduinos, e salvará os dados em um banco de dados gerenciado por MySQL, que utiliza a linguagem de consulta estruturada. Este banco de dados que conterà os dados de temperatura, pressão e umidade do ambiente, também contará com a data e hora para controle do histórico dos dados.

Os dados armazenados serão disponibilizados em uma interface web, sendo visualizados em um aplicativo de supervisão SCADA. Os arquivos salvos em formato de tabela.csv pelo Raspberry será transformado em JSON e enviado ao servidor Web, possibilitando assim o acesso dos dados tanto no sistema *desktop's* quanto nos móveis, tais como *tablet's*, *ipad's* ou *smartphone's*, por meio de planilhas e gráficos, como mostrado na Figura 8.

Figura 8 - Sistema após finalizado



4 Conclusões

O projeto tende a ser desenvolvido buscando uma forma completa e de fácil acessibilidade, possuindo segurança e confiabilidade na transferência de dados através do protocolo SCADA e assim o usuário pode ter acesso aos dados em qualquer plataforma ligada a rede. A conclusão desse estudo abre uma gama de possibilidades para estudos futuros, como automação residencial de baixo custo.

Referências

ARDUINO. Disponível em: <http://arduino.cc/>. Acesso em abril de 2018.

FONSECA, E. G. P.; VEJA, A. S. Tutorial sobre Introdução a Projetos Utilizando o Kit de Desenvolvimento Arduino.

ALCIATORE, DAVID G.; HISTAND, MICHAEL B. Introdução à Mecatrônica e aos Sistemas de Medições. 4 ed. Porto Alegre – RS: AMGH EDITORA LTDA, 2014.

ARDUINO. Arduino UNO & Genuino UNO. Disponível em <<https://www.arduino.cc/en/Main/ArduinoBoardUno>> Acesso em 2018.

CABÓ, A., LIMA, R., LOPES, D., BEZZERA, L., Sistema microcontrolado de baixo custo para o monitoramento de altas temperaturas em fornos da indústria de cerâmica vermelha. Revista de Sistemas e Computação, Salvador, v. 5, n. 1, p. 15-25, jan./jun. 2015.

DONATÉ, C. Sistema sem fio de monitoramento de Consumo e qualidade de energia. Trabalho de conclusão de curso, Engenharia Mecatrônica, UnB, 2016.

EVANS, M.; NOBLE, J; HOCHENBAUM, J. Arduino em Ação. São Paulo: Novatec, 2013.

MELLO, D. A. P., Solução para monitoramento ambiente utilizando. Trabalho de conclusão de curso, Tecnologia em sistemas para internet, UTFPR, 2017.

RICHARDSON, M.; WALLACE, S. Primeiros passos com Raspberry Pi. São Paulo: Novatec, 2013.

SARTORI, G., MOLINA, L. A., LIMA, W. C. G., Desenvolvimento de um sistema microcontrolado de baixo custo utilizando smartphone para aplicações de automação residencial. Trabalho de conclusão de curso, Departamento Acadêmico de Eletrônica, UTFPR, 2015.