



Revista Ilha Digital

Endereço eletrônico:
<http://ilhadigital.florianopolis.ifsc.edu.br/>



PROJETO INTEGRADOR II

Diogo Tavares - José Nicolau Varela

Resumo: Este trabalho contempla a implementação de parte de uma linha de produção, que consiste em realizar a detecção de um recipiente que percorre um trajeto em uma esteira, tendo este recipiente um certo volume contido. A detecção do recipiente ao longo da esteira é feita por três sensores de presença por infravermelho. Quando é colocado o objeto no início da esteira, o processo é iniciado pelo primeiro sensor e a esteira é ligada, em certo ponto da esteira existe outro sensor de presença, o qual pausa o movimento para a medição do volume bem como a cor da substância contida por outros dois sensores apropriados. Após a medição, a esteira é ativada novamente, e um terceiro sensor de presença no final desta encerra o processo. Quando uma medição é feita, os resultados obtidos são apresentados em um display e enviados via wifi, todo o processo é controlado por um Arduino. Todos os detalhes do projeto, sensores e tecnologias utilizadas serão discutidos neste trabalho. O grande objetivo é desenvolver um sistema que apresente características da indústria 4.0, uma tendência em automação cuja qual utiliza IOT (Internet das Coisas), que é um conceito de sistema, interconectado interna e externamente, o qual pode-se monitorar e controlar remotamente, bem como as partes do sistema podem trocar dados entre si. Este projeto é um simples controle de processo, e não apresenta todas as características citadas referentes ao conceito IOT, no entanto, apresenta seus preceitos básicos.

Palavras-chave: IOT, sensores, Indústria 4.0, Arduino, Automação.

Abstract: This work contemplates the implementation of part of a production line, which consists of performing the detection of a container that runs a path on a conveyor, this container having a certain volume contained. Container detection along the conveyor is made by three infrared presence sensors. When the object is placed at the beginning of the treadmill, the process is started by the first sensor and the treadmill is connected, at some point on the treadmill there is another presence sensor, which pauses the movement for the measurement of the volume as well as the color of the substance contained by two other appropriate sensors. After the measurement, the treadmill is activated again, and a third presence sensor at the end of the treadmill terminates the process. When a measurement is made, the results obtained are presented on a display and sent via wifi, the entire process is controlled by an Arduino. All project details, sensors and technologies used will be discussed in this paper. The main goal is to develop a system that presents features of industry 4.0, a trend in automation which uses IOT (Internet of Things), which is a system concept, interconnected internally and externally, which can be monitored and controlled remotely, as well as the parts of the system can exchange data among themselves. This project is a simple process control, and does not present all the mentioned characteristics related to the IOT concept, however, it presents its basic precepts.

Keywords: IOT, sensors, 4.0 Industry, Arduino, Automation.

1 INTRODUÇÃO

A internet das coisas ainda pode ser vista como um conceito futurista, porém já é uma realidade, e na indústria, é algo praticamente necessário nos dias atuais. O conceito de comunicação entre as máquinas na

era da informação é cada vez mais presente, e a tecnologia tem favorecido este desenvolvimento. As comunicações sem fio estão presentes em todos os lugares, seja com tecnologia *wifi*, *bluetooth*, quanto as mais antigas como o rádio e a televisão.

Com a praticidade que as comunicações sem fio proporcionam, trazer este conceito para linhas de produção é uma tendência natural, onde, as máquinas recebem e enviam informações umas para as outras, mantendo um processo interconectado e com gerenciamento de dados eficiente por parte de quem os controla. A grosso modo, este é o conceito da indústria 4.0, onde a internet das coisas (IoT) está presente.

A IoT não requer necessariamente conexões sem fio, porém, esta tecnologia é a mais difundida. Na indústria, um projeto pode ser completamente monitorado em tempo real (até por aplicativo ou página web), presente ou remotamente, ou ainda, uma linha de produção pode se comunicar com outra em qualquer outro lugar (cidade ou país), e modelos baseados em dados existentes podem ser adotados e implementados com mais facilidade e interoperabilidade. Isso é o que se tem chamado de digitalização da produção, o que converge para as redes industriais.

Tudo isso é um conceito muito promissor, o qual possibilitará uma série de inovações, como por exemplo, um cliente poderia criar um projeto customizado e acompanhar todo o processo de produção, ou pode-se acessar uma gigantesca biblioteca de projetos em um banco de dados com inúmeras simulações e parâmetros já estabelecidos. Mas para toda essa interconexão, também é necessário um sistema de segurança robusto, por razões óbvias, ninguém quer seu sistema invadido por algum *hacker*, ou até mesmo que haja alguma falha de comunicação prejudicial.

Portanto, o Projeto Integrador II tem como objetivo o desenvolvimento de um sistema no modelo da indústria 4.0, dando aos alunos uma introdução ao contexto, problemas e soluções, com tecnologias simples a fim de que se possa desenvolver o projeto e integrá-lo.

2 RECURSOS UTILIZADOS

Para a realização deste projeto, além do acesso ao laboratório de pesquisa, foram necessários os seguintes componentes:

- 1 Arduino uno, programado via sua IDE;
- 3 sensores infravermelho E18-D80NK, para a detecção do objeto ao longo da esteira;
- 1 módulo ultrassônico HC-SR04, para medição da distância do conteúdo do recipiente e cálculo do volume;
- 1 sensor de cor RGB TCS230;
- 1 buzzer 5 V, para sinalizar as etapas do processo;
- 1 display Oled I2C 128x64;
- 1 módulo WiFi ESP8266;
- 1 módulo driver para motor DC L298N, para alimentar o motor com 24 V a partir do comando do Arduino;
- 1 motor DC 24 V acoplado à esteira do processo;
- Suportes para os sensores.

3 DESENVOLVIMENTO DO PROJETO

As etapas deste projeto consistiram em realizar os testes com os sensores isoladamente, para a sua compreensão, incorporando-os em seguida. O processo foi desenvolvido na ordem em que será apresentada a seguir, com o estudo de cada componente e sua implementação.

Todos os detalhes sobre as ligações, pinos e o código não serão discutidos ao longo do texto, pois podem ser analisados no apêndice.

3.1 A plataforma Arduino

O principal componente deste projeto é o seu controlador, neste caso foi utilizado o popular Arduino Uno, que é um microcontrolador Atmega328 com uma série de periféricos e outros recursos que facilitam a

prototipagem de pequenos projetos de automação. Sua programação pode ser feita de várias formas, inclusive, existe uma plataforma que utiliza blocos gráficos, para os mais inexperientes em programação, chamada ArduBlock. No entanto, por experiência prévia com programação e para o maior controle sobre o processo, foi optado por utilizar a sua IDE (Ambiente de desenvolvimento e programação) que utiliza a linguagem C/C++.

Uma grande facilidade ao se utilizar o Arduino, é que existem bibliotecas disponíveis para os mais diversos tipos de sensores e aplicações, e para este projeto algumas foram utilizadas, como será discutido posteriormente em cada caso.

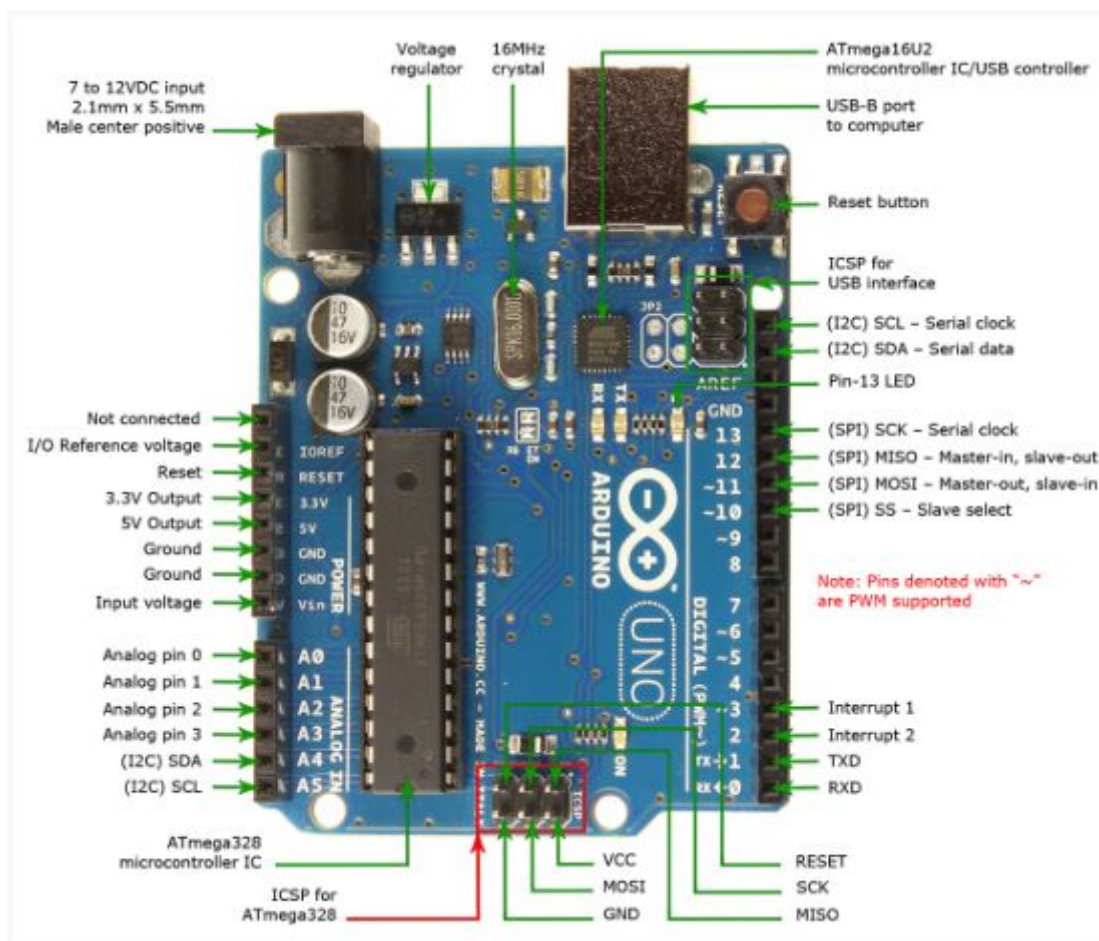


Figura 1. Arduino Uno

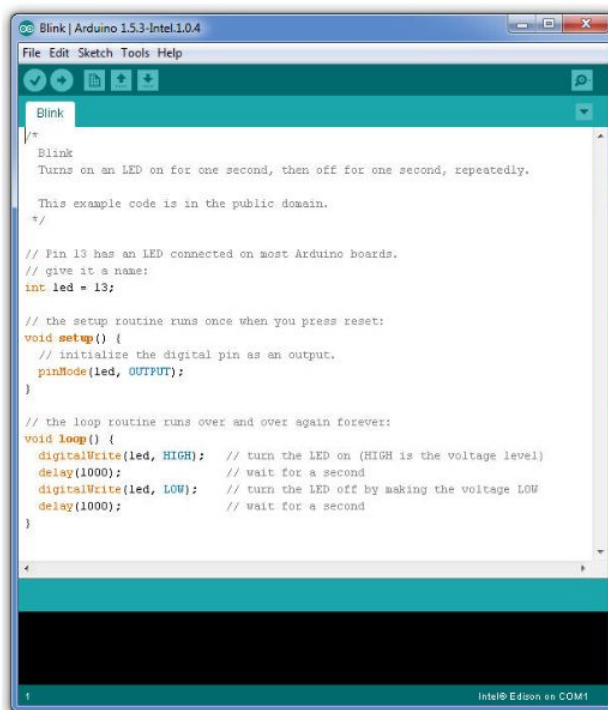


Figura 2. Arduino IDE

3.2 Sensor infravermelho

Os gatilhos do sistema ocorrem quando o recipiente atinge determinadas posições, portanto, o primeiro sensor a ser analisado é o de presença.

Para realizar a detecção do objeto se movendo ao longo da esteira, foi utilizado um sensor que utiliza a emissão de raios infravermelhos.

Este sensor funciona com um LED emitindo uma onda no espectro infravermelho, se algum objeto estiver em frente ao sensor, o receptor (Fotodiodo) detectará a reflexão deste sinal. Um CI comparador LM393 coloca a saída em nível lógico 0 (zero) quando ocorre a detecção. A distância entre o sensor e o objeto pode ser calibrada por um trimpot.



Figura 3. Sensor infravermelho E18-D80NK

Especificações:

- Tensão de operação: 3.3 à 5V DC;
- Emissor e receptor IR;

- Distância de detecção: 2 à 80 cm;
- Potenciômetro para ajuste da distância;
- Ângulo de detecção 35°;
- LEDs: Verde - Sensor ativado, Vermelho - Sensor alimentado;
- Dimensões: 37 x 14 x 6 mm;
- Nível lógico em detecção: 0.

3.3 Primeira implementação utilizando Buzzer e acionamento do motor DC

Chegou o momento de realizar a primeira implementação já integrando o E18-D80NK com um buzzer. Trata-se de um teste simples, o efeito prático do sensor é como o acionamento de um botão, quando acionado este ativa o buzzer. O tempo de acionamento do buzzer é controlado para ser apenas um “bip” (alguns milissegundos).

A implementação do Arduino consiste simplesmente em definir quais os pinos utilizados e o modo dos pinos (INPUT/OUTPUT). No caso do sensor é INPUT e o buzzer, OUTPUT. Alguns condicionais para realizar os comandos de acordo como que se deseja, que no caso é, se o pino do sensor estiver em LOW, o pino do buzzer fica em HIGH por 200 ms. É interessante deixar a entrada do sensor em PULL UP para que esta se mantenha sempre no nível alto quando não houver medição para que não haja flutuação na entrada e ocorra alguma medição errada.

Os códigos e comandos no Arduino não serão aprofundados aqui para não estender demais o texto, o código final pode ser analisado posteriormente.



Figura 4. Buzzer

Feito este teste, foi implementado então o acionamento do motor enquanto o sensor infravermelho está ativo. Para isto, não é possível ligar diretamente um motor nos pinos do Arduino, pois este não possui tensão e corrente suficientes. É necessário a utilização de um driver, com alimentação externa adequada, e é para ele que os comandos do motor são enviados.

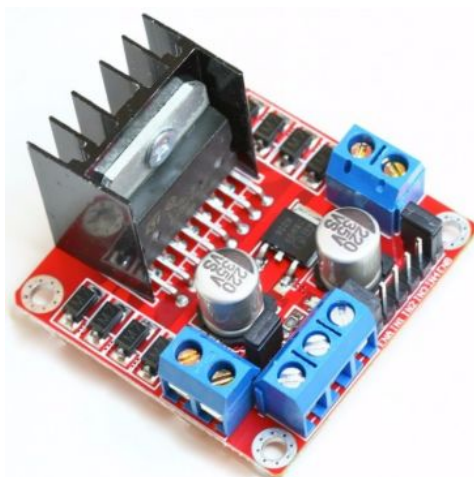


Figura 5. Módulo driver motor DC

É possível fazer o controle da velocidade do motor com PWM, que nada mais é do que uma onda quadrada, mas que pode ter seu tempo em nível alto (ciclo ativo) controlado sem alterar a sua frequência, ou seja, é possível deixar um ciclo com 10% em alto e 90% em baixo, 80% em alto e 20% em baixo e assim por diante. Com isso, pode-se controlar a tensão média resultante, e assim, a velocidade do motor DC. Trata-se de uma emulação de saída analógica que o Arduino possui, sendo utilizada a função `analogWrite(pino,valor)` para isso. A escala de valor é de 0 a 255, pois é um microcontrolador com registradores de 8 bits, e representa valores de tensão de 0 a 5 volts. A frequência é de 980 Hz nos pinos 5 e 6, e 490 Hz nos demais pinos PWM. Neste projeto o PWM é utilizado, mas sua velocidade é sempre constante, ou seja, o valor de escala de tensão é constante, não é a frequência do PWM que controla a velocidade do motor e sim a tensão média.

Este sinal PWM é enviado ao driver, que alimenta o motor com tensão proporcional à recebida do sinal do controlador. A tensão de alimentação do driver é de 24 V.

Feitos estes testes de acionamento, foi implementada então a máquina de estados para a sequência de acionamentos com os 3 sensores de presença. Quando o primeiro sensor é ativado, a esteira (motor) sai de *standby* e é ligada, e permanece assim até que o segundo sensor seja ativado. Neste momento a esteira é parada, e a medição do volume e cor é realizada durante 1 segundo, a esteira é ligada novamente, e permanece até que o terceiro sensor de presença seja ativado, colocando a máquina em *standby* novamente. Neste ponto as medições ainda não foram implementadas, mas o código da máquina já possui os escopos das funções de medição em espera.

A máquina de estados implementada apresenta um comportamento estável e seguro, sinais que não os que cada estado espera não interferem no sistema.

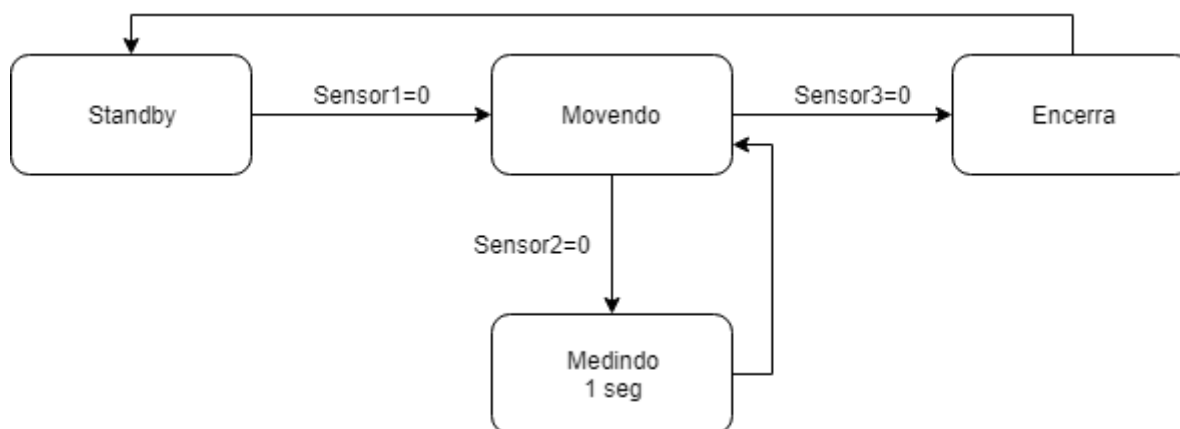


Figura 6. Máquina de estados do sistema

Para a implementação deste projeto será utilizada uma placa de prototipação disponibilizada pelo laboratório para o acionamento do motor, que integra o módulo driver do motor DC e concentra os pinos de todos os sensores. Foi desenvolvido um pequeno *shield* (módulo com periféricos) acoplado ao controlador que contém o display e o buzzer, e as demais conexões, feitas com jumpers soldados ao shield. A conexão inicialmente seria feita com um chicote conector,

3.4 Display Oled e sensor ultrassônico

3.4.1 Display Oled

Para fazer o acompanhamento dos dados, foi feita a implementação do display, para verificar também os passos do projeto em seu desenvolvimento.

A primeira opção para o display seria um tradicional LCD 16x2, porém, este display necessita de pelo menos 7 pinos do controlador, em contrapartida, o display Oled necessita de apenas 2, uma vez que utiliza o protocolo I2C, então, por este fator, este foi o display escolhido. Outra característica é por ser um display gráfico, que oferece uma resolução muito boa. As suas desvantagens são seu tamanho menor, e a relativa complexidade de operação, muito maior que a de um LCD comum. Foi necessário utilizar uma biblioteca, e a U8glib.h foi escolhida por ser leve e eficiente, outras foram testadas, porém extremamente pesadas e consumiam até 80% da memória do controlador e ainda necessitavam configurações em outros arquivos.

Este display não necessita de luz de fundo (*backlight*), o que o torna muito mais econômico no consumo de energia. OLED significa Organic Light-Emitting Diode, ou Diodo Emissor de Luz Orgânico. Esse tipo de display é composto por uma camada de material orgânico que é colocado entre dois eletrodos sendo que um deles geralmente é transparente. Esse material orgânico, ao ser estimulado por um campo eletromagnético, emite luz, dispensando assim o uso do *backlight*.

Este display utiliza I2C, que é um protocolo de apenas 2 barramentos, onde podem ser conectados diversos dispositivos paralelamente ao mesmo barramento, estes são identificados por números de identificação, no caso do display é 0x3C, mas isto não precisa ser configurado com a biblioteca utilizada. Este protocolo utiliza o modelo mestre/escravo, sendo o controlador o mestre. Os pinos I2C são chamados de SCL (barramento de clock) e SDA (barramento de dados), que são os pinos A5 e A4 do Arduino respectivamente. A biblioteca configura automaticamente os pinos, bastando informar no construtor do objeto do display o modelo deste.



Figura 7. Display Oled 128x64

Especificações do display:

- Conexão: I2C (endereço 0x3C);
- Área visível: 0.96";
- Resolução: 128 x 64;
- Controlador: SSD1306;
- Tensão de Alimentação: 3.3V / 5V;
- Tensão de I/O: 3.3V / 5V;
- Cor Letra: Azul;
- Dimensões da placa: 28x27mm

3.4.2 Módulo ultrassônico e medida de volume

Este sensor é utilizado para medir a distância de objetos, com precisão de 3 mm. O funcionamento do HC-SR04 (datasheet) se baseia no envio de sinais ultrassônicos pelo sensor, que aguarda o retorno (echo) do sinal, e com base no tempo entre envio e retorno, calcula a distância entre o sensor e o objeto detectado.

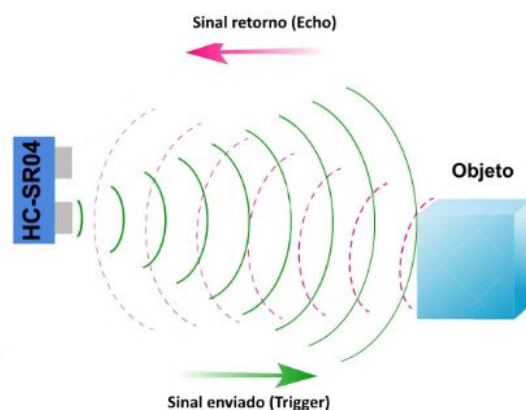


Figura 8. Princípio do sensor ultrassônico

Primeiramente é enviado um pulso de 10 μ s, indicando o início da transmissão de dados. Depois disso, são enviados 8 pulsos de 40 KHz e o sensor então aguarda o retorno (em nível alto/high), para determinar a distância entre o sensor e o objeto, utilizando a equação $Distância = (Tempo\ echo\ em\ nível\ alto * velocidade\ do\ som) / 2$.

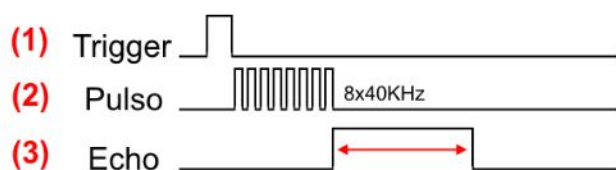


Figura 9. Sinais do sensor



Figura 10. Módulo ultrassônico HC-SR04

O controle deste sensor é simples, e uma biblioteca foi utilizada para encapsular as funções e simplificar o código. A biblioteca Ultrasonic.h cria um objeto o qual possui o método que retorna a distância do objeto, este valor é um inteiro e sua escala está de 1 cm, a qual foi a precisão adotada. Foram feitos testes com valores com ponto flutuante, porém houve oscilação nas medidas.

Para realizar o cálculo de volume, foram obtidas as medidas do recipiente, que consiste em um segmento de cone. A equação do volume do recipiente é a seguinte:

$$V = \pi * h * (R^2 + R * r + r^2) / 3$$

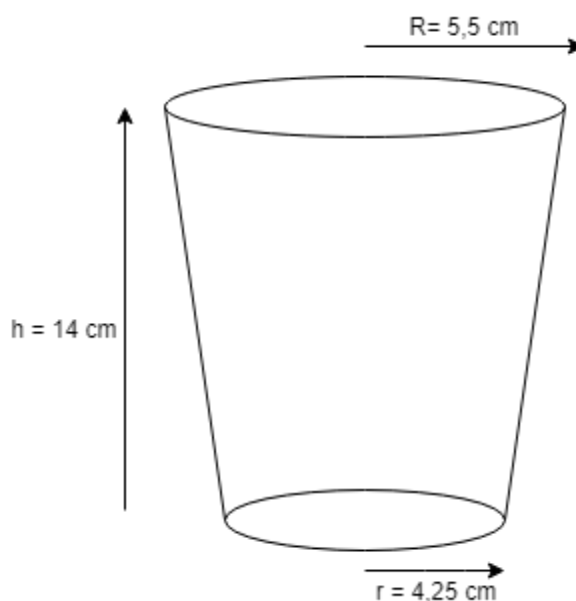


Figura 11. Volume do recipiente

Como o valor de R para o volume depende da altura h, foi feito um equacionamento para deixar tudo em função de h ($h_{\text{novo}} = 14 - h_{\text{medido}}$), e simplificado o máximo possível, eliminando qualquer fator trigonométrico e excessivas divisões e multiplicações, principalmente em se tratando de números flutuantes.

O Arduino é um controlador muito limitado em termos de processamento, e apesar da ilusão da IDE fornecer várias funções para cálculos e orientação a objetos e tudo mais, é preciso ter muito cuidado, pois ao

colocar um processamento muito pesado ele pode demorar ou resultar em erros de arredondamento e overflow. Para um processamento mais robusto seria necessário um controlador de 32 bits ou ainda um DSP.

As medidas são feitas no intervalo de 1 segundo enquanto a esteira é parada, neste período é realizada uma média das distâncias medidas, e ao final, o volume é calculado pela seguinte equação:

$$V = \pi * h(54,18 + 0,286 * h + 0.0005 * h^2)/3$$

É possível que haja truncamento para um valor muito pequeno das parcelas, e com isso algum erro de medida, mas é uma parcela muito pequena do valor, o que pode ser aceitável. O valor do cálculo é então multiplicado por 0,001, para conversão de cm³ para litros.

3.5 Medição de cor

Hoje em dia é possível diferenciar produtos em uma linha de produção, pela determinação de sua cor, desde que possuam cores distintas. O sensor de cor RGB TCS230 (figura 12), pode detectar uma grande variedade de cores com base em seu comprimento de onda.



Figura 12. Sensor de cor RGB TCS230

Especificações do sensor:

Tensão de Alimentação: 2.7V a 5.5V

Tamanho: 28,4 x 28,4 mm (1,12 x 1,12 ")

Interface: TTL digital

Conversão de alta resolução de intensidade de luz em frequência

Frequência de saída programável em cores e em escala

Comunica-se diretamente ao microcontrolador

Ele detecta a luz colorida com a ajuda de uma matriz de 8 x 8 fotodiodos. Um fotodiodo é simplesmente um dispositivo semicondutor que converte luz em corrente e utilizando um conversor de corrente-para-frequência, as leituras dos fotodiodos são convertidas em uma onda quadrada com uma frequência diretamente proporcional à intensidade da luz. Finalmente, usando a placa Arduino, podemos ler a saída de onda quadrada e obter os resultados para a cor. Na placa do sensor estão instalados 4 leds brancos para iluminar o objeto a ser medido.

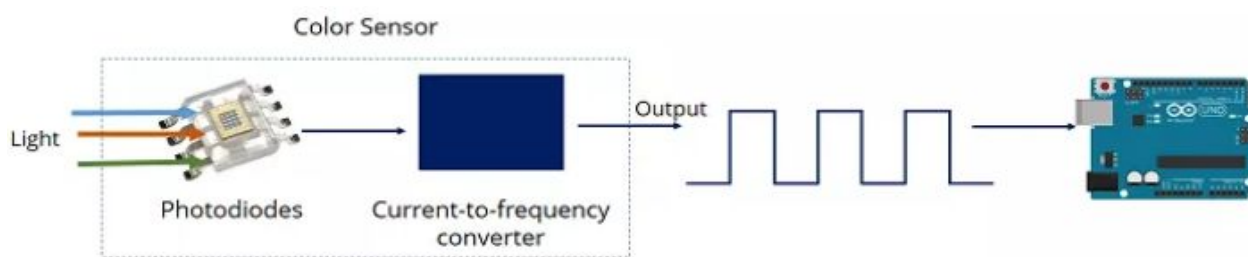


Figura 13 . Pinagem do sensor TCS230

O sensor tem:

- 16 fotodiodos com filtro vermelho - sensível ao comprimento de onda vermelho
- 16 fotodiodos com filtro verde - sensível ao comprimento de onda verde
- 16 fotodiodos com filtro azul - sensível ao comprimento de onda azul
- 16 fotodiodos sem filtro.

A figura 14 mostra a pinagem do sensor.



Figura 14 . Pinagem do sensor TCS230

NOME DO PIN	I/O	DESCRIÇÃO
GND		Ground (Referência)
OE	I	Ativar para frequência de saída (ativo baixo)
OUT	O	Frequência de saída
S0,S1	I	Entradas de seleção de escala de frequência de saída
S2,S3	I	Entradas de seleção do tipo fotodiodo
VDD		Alimentação de tensão

Tabela 1

Para seleccionar a cor lida pelo fotodiodo, usa-se os pinos de controle S2 e S3. Como os fotodiodos são conectados em paralelo, a configuração de S2 e S3 LOW e HIGH em diferentes combinações permite seleccionar diferentes fotodiodos, como mostra a tabela a seguir:

Escala de frequência de saída	S0	S1
desativado	L	L
2%	L	H
20%	H	L
100%	H	H

Tabela 2

Para o Arduino, é comum usar uma escala de frequência de 20%.

Este sensor não é muito preciso, mas funciona bem para detectar cores em projetos simples.

Pode-se dividir em dois momentos na codificação para a detecção de cor:

1- Leitura e exibição da frequência de saída no monitor serial. Nesta parte precisamos anotar os valores de frequência quando colocamos cores diferentes na frente do sensor, praticamente uma calibração do sensor.

2- Distinguir entre cores diferentes. Neste momento inserimos os valores de frequência selecionados anteriormente no código, para que o sensor pudesse distinguir entre cores diferentes. Detectamos cores vermelhas, verdes e azuis.

3.6 Módulo WiFi

Para o monitoramento remoto, optamos pela rede wifi e utilizamos o módulo ESP 8266 mostrado na figura 13..



Figura 15. Módulo wifi ESP8266

Especificações:

- Chip: ESP8266
- Modelo: ESP-01
- Tensão de operação: 3,3V
- Suporte à redes: 802.11 b/g/n
- Alcance: 90m aprox.
- Comunicação: Serial (TX/RX)
- Suporta comunicação TCP e UDP
- Conectores: GPIO, I2C, SPI, UART, Entrada ADC, Saída PWM e Sensor de Temperatura interno.
- Modo de segurança: OPEN/WEP/WPA_PSK/WPA2_PSK/WPA_WPA2_PSK
- Dimensões: 25 x 14 x 1mm
- Peso: 7g

Ele consiste basicamente em uma unidade de microcontrolador e um transceptor WIFI. Com este módulo, podemos obter comunicação com a rede WIFI local e se necessário, com a internet.

Para programar-lo , usamos a IDE do arduino e conectamos na usb através de um conversor usb-serial, o módulo FTDI-232, esquemático mostrado na figura 1x.

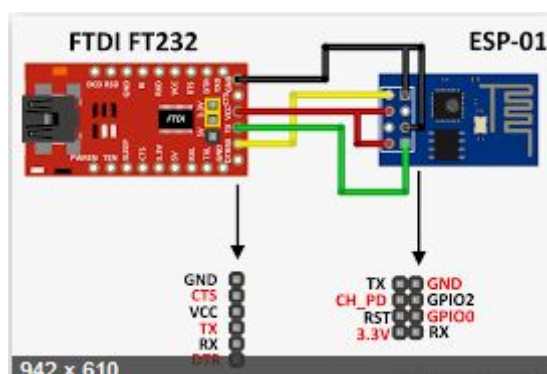


Figura 16 . Esquemático ESP8266 - FTDI-232.

A figura 16 mostra o formato e pinagem do ESP8266.

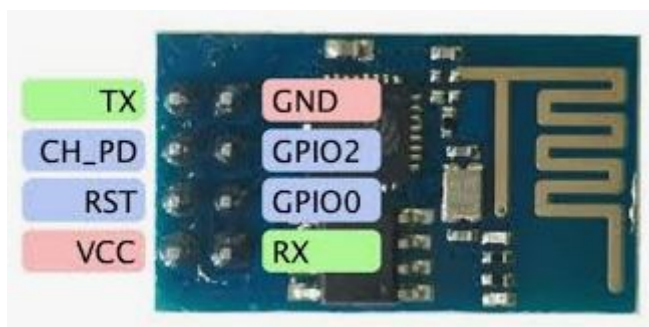


Figura 17. Pinagem do wifi ESP8266

Para visualizar os dados, tanto em um browser de um computador desktop quanto em um smartfone, foi desenvolvida uma página HTML (Hyper Text Markup Language), que é padrão para aplicações WEB. Esta consiste em uma linguagem de marcação, onde o conteúdo da página é descrito em um código (HTML) e o browser o interpreta, apresentando-o na tela. Para os estilos, centralização, cores, etc, é utilizada a linguagem CSS (Cascade Style Sheet) que define como será a aparência da tela.

Foi desenvolvido o código da página, e então este foi implementado como uma string no código do módulo WiFi, que, ao receber os dados do Arduino por comunicação serial, concatena estes dados com a string da página existente no código, e então os envia para a rede.

A visualização das cores é uma aproximação das cores medidas, estas são calibradas para maior resolução da cor a qual é distinguida, para melhores efeitos na visualização. Esta aproximação é feita aumentando os valores da componente da cor em questão, assim, deixando-a mais nítida. O HTML reconhece cores no formato hexadecimal com a sintaxe #ffffff, sendo na ordem, # vermelho/ verde/ azul, com dois algarismos para cada cor.

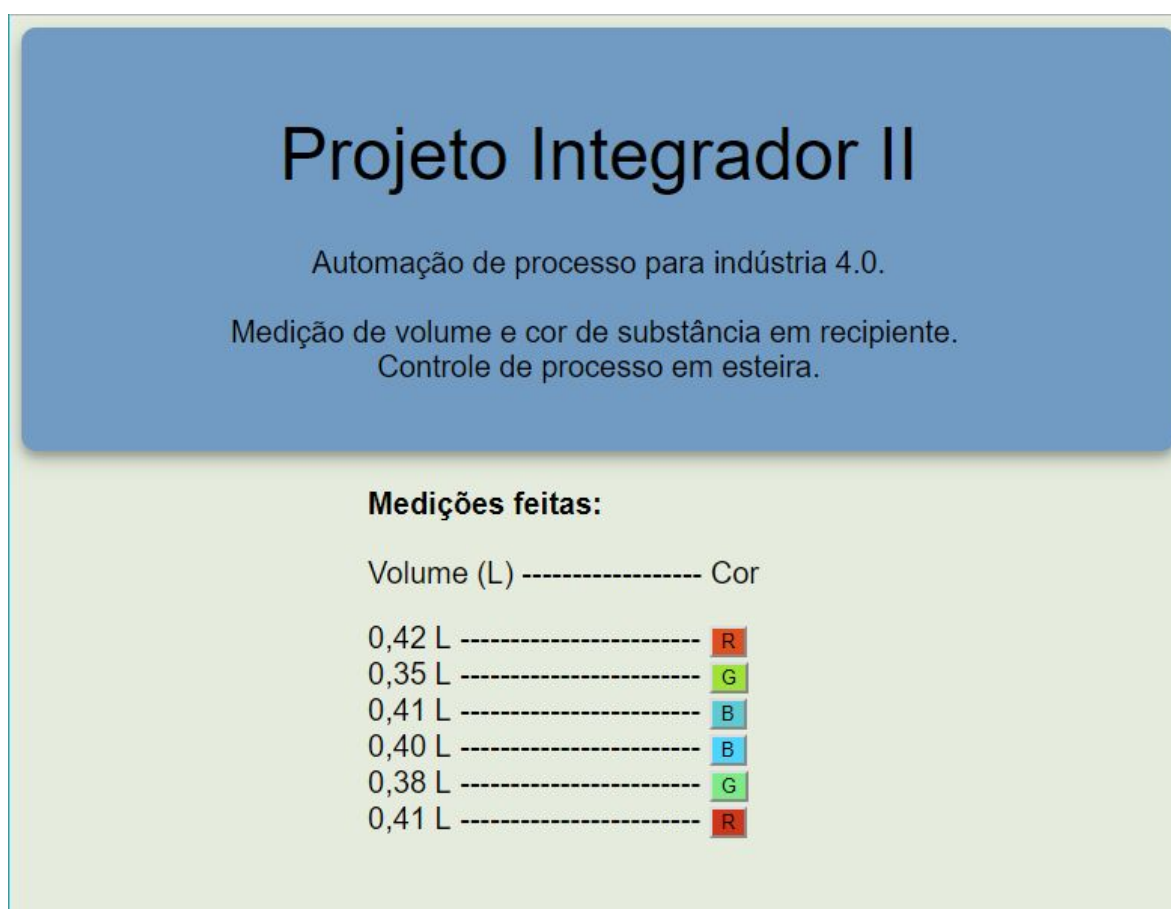


Figura 18. Interface da página HTML.

4 CONCLUSÃO

O sistema desenvolvido apresentou alguns desafios, como o de adquirir sinais de qualidade para as medições, sendo que em alguns momentos estes apresentam variações dos valores. Ao realizar o monitoramento dos sinais que cada sensor enviava, algumas vezes eram muito fora do esperado, não foi identificada a fonte dos ruídos com precisão. Foram utilizadas técnicas no código para melhorar a qualidade dos resultados, como média para a medida de distância e calibração das cores. Contudo, é possível que hajam erros de medida em alguns momentos.

A integração do sistema de maneira geral não é complexa, e as tecnologias e recursos utilizados facilitaram o processo, alguns aspectos, como os citados anteriormente, apresentaram certa demanda de tempo para debug e testes de implementação.

Os resultados obtidos, com o funcionamento correto da máquina de estados, as leituras de distância e cor (apesar de ruidosa em alguns momentos), o envio das informações para o display Oled e o monitoramento WiFi se mostraram muito interessantes e satisfatórios.

Em suma, pode-se concluir que o objetivo foi alcançado e o sistema foi integrado e funciona de forma consistente, a automação realiza todos as etapas e representa bem um modelo de processo que se enquadraria na indústria 4.0.

5 BIBLIOGRAFIA

- <https://pt.wikipedia.org/wiki/I%C2%B2C>
- <https://www.filipeflop.com/blog/como-conectar-display-oled-arduino/>
- <https://www.arduino.cc/reference/en/>
- <https://www.filipeflop.com/blog/motor-dc-arduino-ponte-h-l298n/>
- <https://portal.vidadesilicio.com.br/driver-motor-com-ponte-h-l298n/>
- <https://www.filipeflop.com/blog/sensor-infravermelho-arduino/>
- <https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>
- <https://randomnerdtutorials.com/arduino-color-sensor-tcs230-tcs3200/>
- <https://www.filipeflop.com/blog/esp8266-arduino-tutorial/>

6 APÊNDICE

Esquemático do sistema

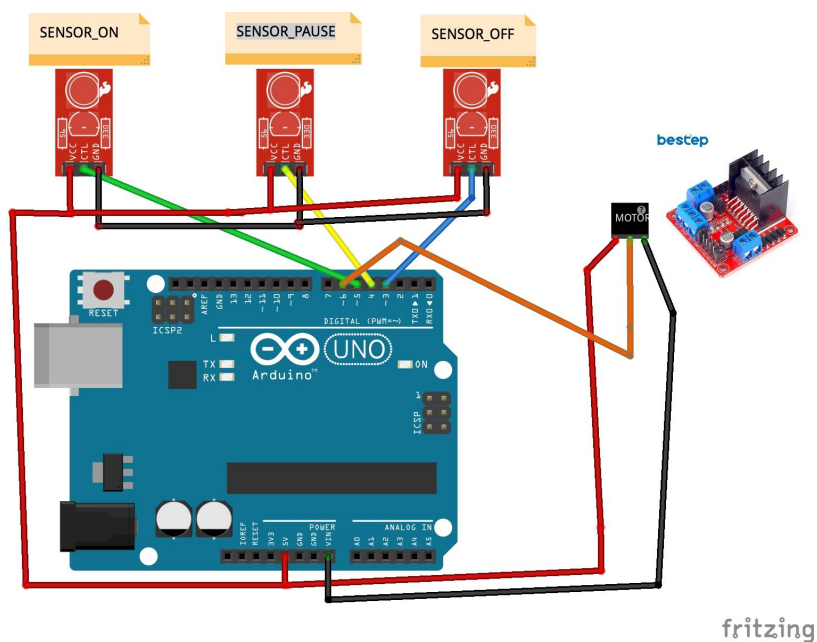


Figura 19 . Esquemático do sensores IR e drive do motor

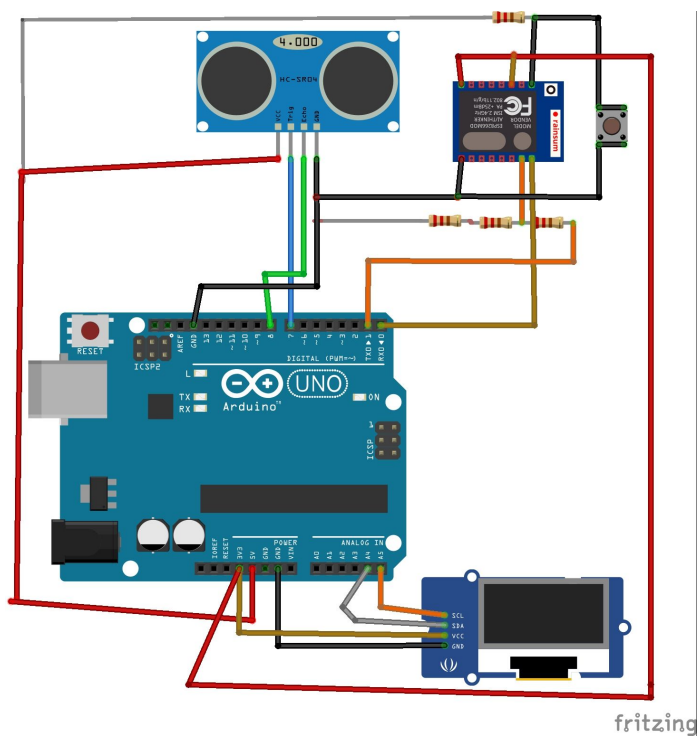


Figura 20 . Esquemático Display, ultra-som e módulo wifi

r

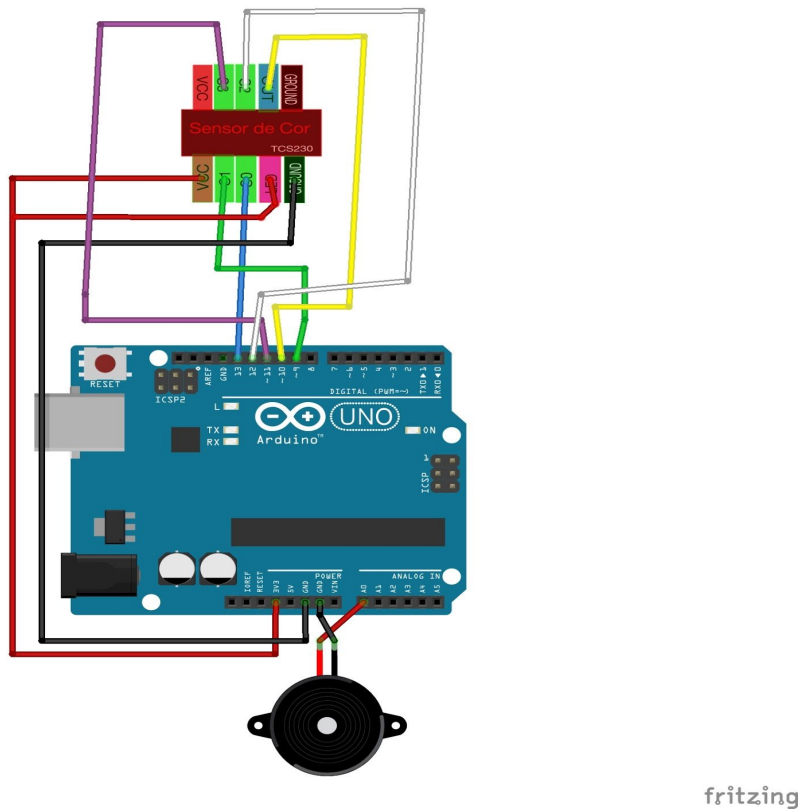


Figura21 . Esquemático do sensor de cor e buzzer