

## PROTOTIPAGEM DE UMA RÉGUA DIGITAL DE PRECISÃO UTILIZANDO O ARDUÍNO PARA A INSPEÇÃO GEOMÉTRICA DO TRILHO FERROVIÁRIO PERLÍTICO PÓS SOLDAGEM

PROTOTYPING OF A PRECISION DIGITAL RULER USING THE  
ARDUINO FOR THE GEOMETRIC INSPECTION OF THE PERLITHIC RAILWAY  
POST WELDING

Walace de Barros Silva<sup>1</sup>

Alex Franco Ferreira<sup>2</sup>

**RESUMO:** Nos pátios de soldagem do setor ferroviário, o processo de soldagem por faísca (*Flash Butt Welding* - FBW) de trilhos ferroviários perlíticos é padronizado por meio de Procedimentos Operacionais Padrão (POPs) baseados em normas e processos normativos que regem o controle de qualidade da soldagem durante o processo de soldagem. O objetivo deste trabalho foi desenvolver uma régua de precisão digital para otimizar a inspeção geométrica de soldas brutas. Para a prototipagem, uma placa Arduino foi programada em C++ para receber as leituras do sensor de distância a laser e imprimir a informação correspondente no display LCD, referente ao empenamento vertical e horizontal do tarugo. Depois de soldar. Como a precisão do sensor usado depende de certos fatores que causam variação na leitura, os dados recebidos do sensor variam. Comparado aos programas atuais que utilizam calibradores tipo faca, o protótipo é bem otimizado em programas de inspeção geométrica devido à facilidade de uso e observação das leituras de medição. Para a futura implementação do projeto final, é necessário o uso de sensores com maior precisão e mínima variação (na faixa de um centésimo de milímetro).

711

**Palavras-chave:** Régua. Arduíno. Inspeção. Solda. Trilhos.

**ABSTRACT:** In railroad welding yards, the Flash Butt Welding (FBW) process of pearlitic rail tracks is standardized through Standard Operating Procedures (SOPs) based on normative standards and processes that govern the quality control of welding during the welding process. The objective of this work was to develop a digital precision ruler to optimize the geometric inspection of raw welds. For prototyping, an Arduino board was programmed in C++ to receive the readings from the laser distance sensor and print the corresponding information on the LCD display, regarding the vertical and horizontal warping of the billet. After soldering. Since the accuracy of the sensor used depends on certain factors that cause the reading to vary, the data received from the sensor varies. Compared to current programs that use knife-type gauges, the prototype is well optimized in geometric inspection programs due to the ease of use and observation of the measurement readings. For future implementation of the final project, the use of sensors with higher accuracy and minimal variation (in the range of one hundredth of a millimeter) is required.

**Keywords:** Ruler. Arduino. Inspection. Welding. Rail.

<sup>1</sup> Universidade de Vassouras/ Vassouras/ RJ/ Brasil

<sup>2</sup> Universidade de Vassouras/ Vassouras/ RJ/ Brasil

## INTRODUÇÃO

O paradigma de que produtos e operações podem ser bons o suficiente para serem mantidos, foi quebrado com a chegada do conceito de melhoria contínua, trazendo uma visão de que não existe algo suficientemente bom em uma determinada organização que não possa ser aprimorado. A constante evolução dos procedimentos operacionais, a tecnologia tornou-se uma grande aliada na automatização em busca de melhorias nos processos (SOUZA, 2017).

O Arduíno vem se destacando devido a possibilidade de discentes, docentes, programadores e pesquisadores em desenvolver projetos e colocar em prática através de protótipos a custos acessíveis. Segundo Souza (2017, p.12) “o Arduíno foi concebido para ser duas coisas: barato e fácil, em pé de igualdade em relação de importância”.

No conceito de melhoria contínua, a única forma de um processo ser melhorado, é ele fazer parte de um procedimento operacional POP (FELIX, 2022).

O objetivo do presente trabalho foi o desenvolvimento de um protótipo que viesse apresentar sugestões para otimizar processos de inspeção geométrica do trilho ferroviário perlítico, pós soldagem, realizada dentro de um estaleiro ou em lugares apropriados dentro do padrão para o determinado processo.

O projeto foi baseado na estrutura de ferramentas usuais descritas no procedimento padrão, onde foi possível confeccionar um instrumento para medição com elementos eletrônicos programáveis dentro da plataforma Arduíno, visando a praticidade e facilidade durante a inspeção.

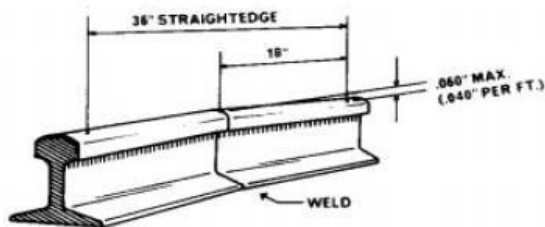
### Inspeção Geométrica

Dentro de um estaleiro ferroviário, os trilhos de 24 m de comprimento recebidos dos fornecedores são soldados através de um processo chamado de *flash butt welding* até formarem um trilho longo soldado (TLS) de 216 metros, ou seja, 9 trilhos e 8 soldas (VIANA, 2014).

Para Wainer (1992 apud Viana, 2014), existem procedimentos baseadas em normas e especificações para evitar problemas relacionados a soldagem, entre elas, a inspeção de trilhos e equipamentos. São definidos os procedimentos operacionais padrão para proporcionar a confiabilidade e a qualidade do processo de soldagem. Os parâmetros de tolerância de desalinhamento vertical (Figura 1 e 2) e horizontal (Figura

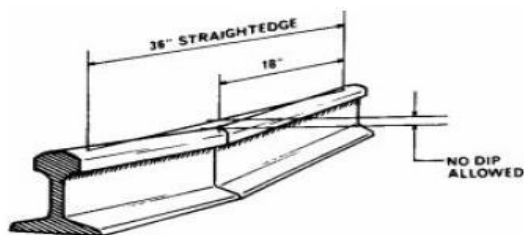
3) para trilhos soldados pós esmerilhamento, são determinados pela norma AREMA (*American Railway Engineering and Maintenance-of-way Association*) (VIANA, 2014).

**Figura 1** – Limite para desalinhamento com curva ascendente após esmerilhamento.



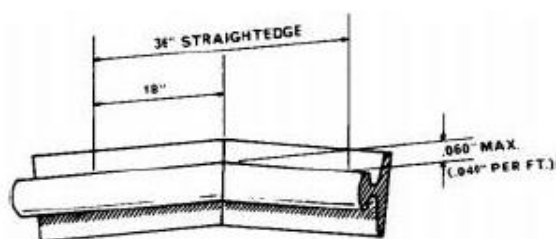
Fonte: AREMA (2017).

**Figura 2** – Não admitido desalinhamento com curva descendente pós esmerilhamento.



Fonte: AREMA (2017).

**Figura 3** – Limite para desalinhamento lateral pós esmerilhamento.



Fonte: AREMA (2017).

Tomando como base essas tolerâncias pós esmerilhamento, adota-se uma tolerância aceitável de até 2 mm tanto no desalinhamento vertical (flecha) como no desalinhamento lateral durante a inspeção geométrica da solda bruta (a quente) durante a pós soldagem. Para realização da inspeção, o inspetor utiliza uma régua de 1 metro de comprimento com entalhe no centro (Figura 4).

**Figura 4** – Régua de aço com 1 metro de comprimento com entalhe no centro.



**Fonte:** Autor (2023).

Para compensar a saliência da solda (Figura 5) e utiliza-se um calibre tipo canivete (Figura 6) para a conferência da tolerância aceitável, durante a inspeção vertical e horizontal (Figura 6 e 7).

**Figura 5** – Inspeção Geométrica pós soldagem do desalinhamento vertical (flecha).



**Fonte:** Autor (2023).

**Figura 6** – Inspeção Geométrica pós soldagem do desalinhamento vertical, utilizando o calibre canivete para a conferência da tolerância aceitável.



**Fonte:** Autor (2023).

**Figura 7** – Inspeção Geométrica pós soldagem do desalinhamento lateral.



**Fonte:** Autor (2023).

## Arduino

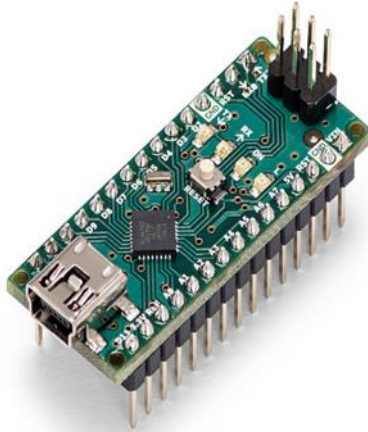
Conforme Stevan Júnior e Silva (2015, p.12) “para todo lado que olhamos vemos exemplos de sistemas que foram automatizados ou que precisam de automação para ganhar padronização, organização e melhoria no desempenho de suas funções”.

Com uma plataforma de interface de hardware que permite uma programação com código-fonte *open-source*, ou *software* livre, o Arduino é baseado em microcontroladores que possibilitam a plataforma interação e controle sobre atuadores e processamento de periféricos como: memória, comunicação serial, conversores analógico-digitais, dispositivos PWM, comparadores e entre outros que podem ser controlados pelos microcontroladores da linha *Atmel*, empregados em diferentes versões de hardwares atribuídos a placas periféricas Arduino ou Shields (STEVAN JÚNIOR e SILVA, 2015).

A programação do *hardware* é realizada através do *software* Arduino *Integrated Development Environment* ou *Arduino Software* (IDE), contendo uma área de trabalho para a programação com várias funções (ARDUINO, 2023).

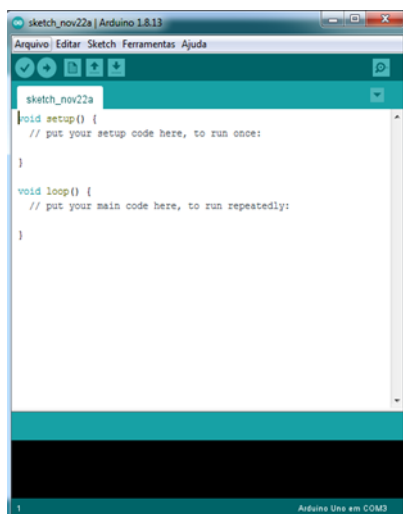
Foi utilizado o hardware na versão Arduino Nano V3 com microcontrolador Atmega328 (Figura 8) e a linguagem de programação C/C++ textualizada no Arduino Software IDE (Figura 9), para realizar a sua programação.

**Figura 8** – Arduino Nano Atmega328.



**Fonte:** Arduino (2022).

**Figura 9** – Arduino Software IDE.



**Fonte:** Arduino (2022).

## Display LCD 16x2 com Protocolo I2C

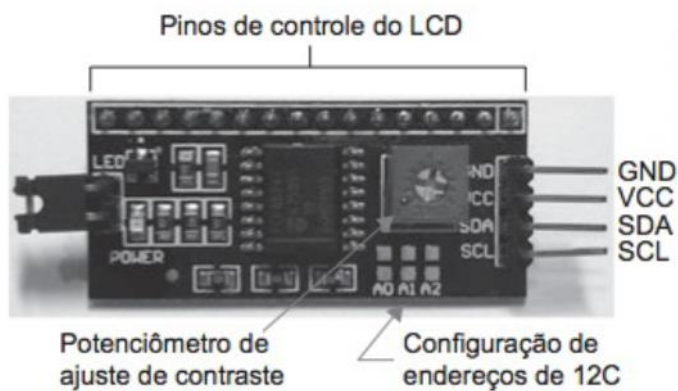
Segundo Evans, Noble e Hockenbaum (2013) a tecnologia *Liquid Crystal Display* (LCD) está entre as principais maneiras de utilização de dados eletrônicos presentes no nosso cotidiano a todo tempo, seja através de um LCD de um despertador, mp3 player, telefone, etc.

O Controlador Hitachi HD44780 é um dos chips mais comuns utilizados em sistemas embarcados que suporta muitas formas e tamanhos na construção de monitores de LCD, com um bastante custo acessível. (EVANS et al. 2013).

Conforme Stevan Júnior e Silva (2015, p.199) “o excesso de pinos que um display de LCD consome pode ser um limitador de expansão de um projeto. Uma alternativa é utilizar o circuito integrado que faça o controle através do protocolo serial I2C”.

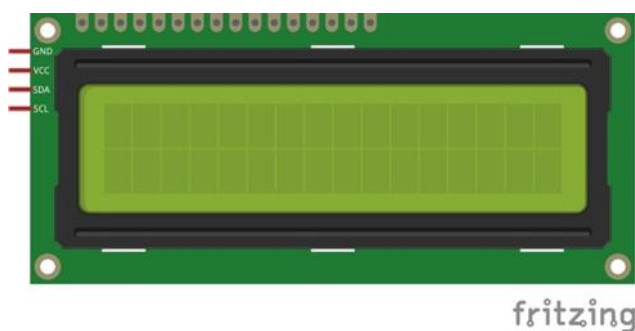
O módulo I2C (Figura 10) para LCD vem com o circuito integrado PCF85574 ao qual é possível transmitir dados a um display de LCD (Figura 11) de 2 linhas (16x2 ou 20x2) utilizando 2 pinos do Arduino: o Serial Data (SDA) e o *Serial Clock* (SCL) (STEVAN JÚNIOR E SILVA, 2015).

**Figura 10** – Módulo I2C para LCD.



**Fonte:** Arduino (2022).

**Figura 11** – Display LCD 16x2 com controlador HITACHI HD44780.



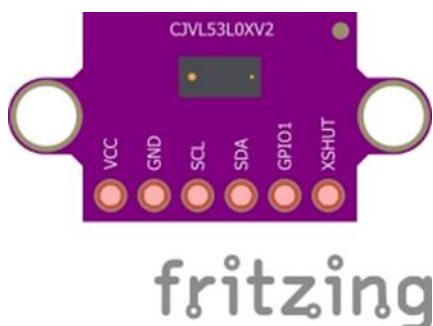
**Fonte:** Fritzing (2022).

## Sensor de Distância a Laser VL53LoX

Na Figura 12 mostra-se o sensor de Distância a Laser VL53LoX



**Figura 12** - Sensor de Distância a Laser VL53LoX.



Fonte: Fritzing (2022).

## Arquitetura Física do Projeto

Para a construção do projeto, foram utilizados:

- Tubo Retangular (1 metro) Metalon 30 x 50mm, com entalhe no centro para a construção da régua digital;
- Placa Arduino (01 unidade) Nano V3 com microcontrolador Atmega328, para o controle da arquitetura e este programado através da linguagem de programação C/C++ realizada no Arduino Software IDE instalado na plataforma Windows11, onde o programa foi compilado e carregado para o hardware através do cabo USB ligado a uma porta COM do terminal de programação;
- Bateria de 9Vcc (01 unidade) para alimentação da placa Arduino;
- Chave liga/desliga para habilitar e desabilitar alimentação da placa Arduino;
- 2 sensor modelo VL53LoX com protocolo I2C para a aferição do desalinhamento da curva ascendente vertical e do desalinhamento curva lateral do trilho pós soldado, ambos alimentados através do GND e do pino 5Vcc output da Arduino e comunicados através da comunicação serial I2C (SDA E SCL) para transferência de dados. O pino XSHUT do sensor 1 e do sensor 2, estão ligados aos pinos 6 e 7 da placa Arduino, isto foi realizado para desligar e religar os sensores em tempos diferentes, para que os sensores não venham se iniciar com o mesmo endereço I2C por padrão. Porém o pino XSHUT dos sensores dever alimentados com tensão de 3.6Vcc, caso o contrário, podem ser danificados com tensão de 5Vcc. Desta forma, os pinos XSHUT dos sensores foram determinados como entrada na placa Arduino (pinos 6 e 7). Feito isso o *pull-up* interno do sensor automaticamente faz o pino XSHUT ficar em nível alto  $\pm 3,2Vcc$ .

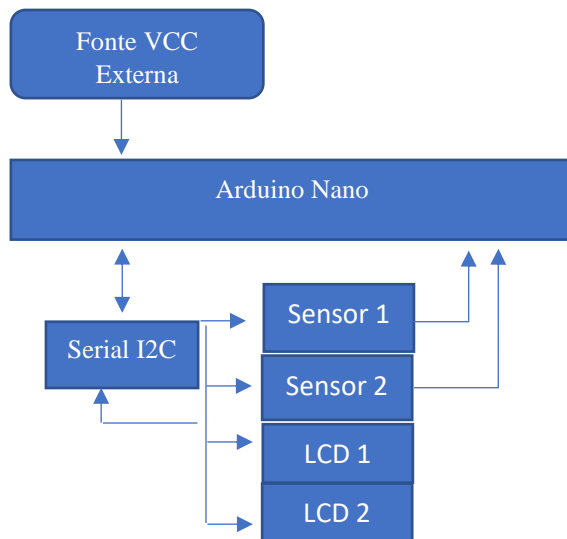


- Display LCD 16x2 (02 unidades) com protocolo de comunicação I2C, para printar informações de leituras dos sensores. Foi configurado os endereços da serial em cada módulo para que os mesmos não sejam iguais.
- Cabos *jumpers* (25 unidades) para a ligação do circuito;

### Diagrama de Blocos da Arquitetura

Adotou-se a arquitetura apresentada na Figura 13

**Figura 13** - Diagrama de blocos da arquitetura.



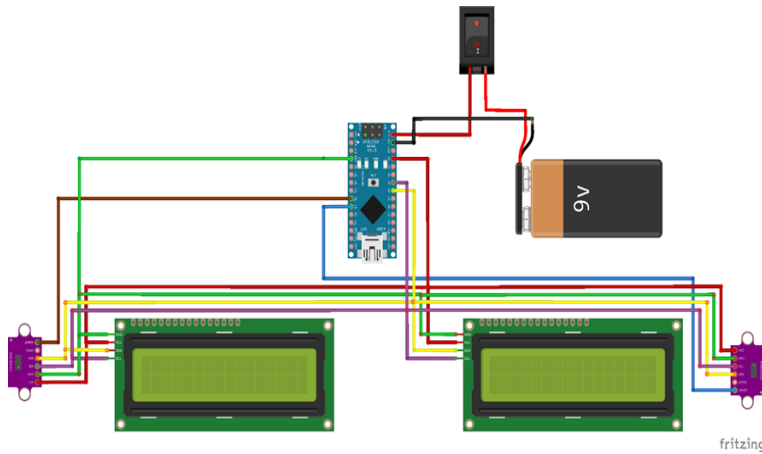
**Fonte:** Autor (2023).

### Esquema de Ligação do Projeto

Para a arquitetura de ligação do projeto, foi utilizado o *software Fritzing* (Figuras 14 e 15).

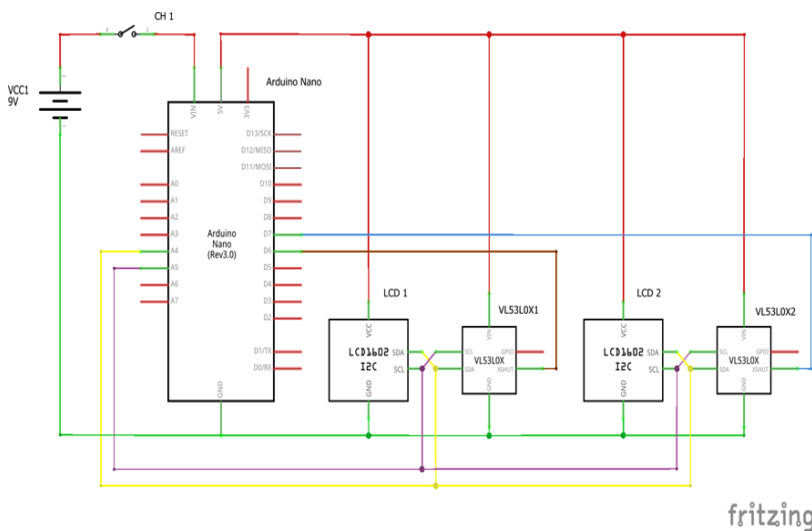
O *Fritzing* permite aos usuários documentar e compartilhar seus protótipos, motivando pessoas a aprenderem eletrônica e a fabricar seus próprios PCBs profissionais através de uma maneira fácil e criativa, em um ambiente de hardware *open-surce* (código aberto) para processo e Arduino (FRITZING, 2022).

Figura 14 – Circuito do Projeto.



Fonte: Autor (2023).

Figura 15 - Diagrama de Ligação do Projeto.



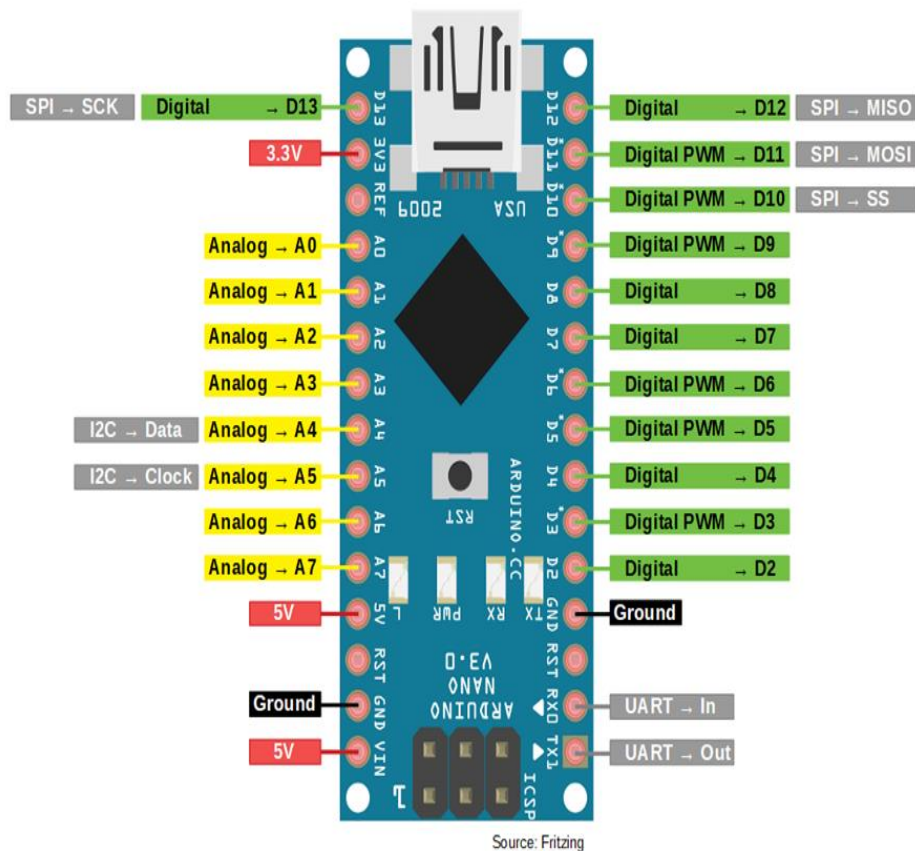
Fonte: Autor (2023).

### Programação da Plataforma Arduino

A programação foi realizada no Arduino Software IDE em linguagem C/C++, utilizando bibliotecas, variáveis e funções. As bibliotecas utilizadas foram: *Wire.h*, *VL53LoX.h* e *LiquidCrystal\_I2C.h*.

A biblioteca *Wire.h* permite a comunicação serial com dispositivos I2C, através do pino A4 e A5 do Arduino Nano (Figura 16), respectivamente SDA (*Data Line*) e SCL (*Clock Line*) (ARUINO, 2022).

Figura 16 - Pinagem do Arduino Nano.



Fonte: Fritzing (2022).

Segundo Pololu (2022), a biblioteca VL53LoX “ajuda na interface com o sensor de distância de tempo de voo VL53LoX da ST. A biblioteca simplifica a configuração do sensor e a leitura dos dados de alcance dele via I2C.”

O pino XSHUT do sensor é necessário para o funcionamento, já que ele será responsável em alterar o endereço I2C para que não fiquem com endereços iguais conforme a configuração de fábrica. É sempre necessário quando na utilização de dois ou mais sensores em um projeto. O pino GPIOI é interrupção de leitura, quando uma medição é completada<sup>3</sup> (STMICROELECTRONICS, 2022).

A biblioteca LiquidCrystal\_I2C.h permite comunicação serial, através do módulo I2C integrado em cada display. Os pinos SDA e SCL também são ligados respectivamente aos pinos A4 e A5 do Arduino Nano.

Uma visualização parcial do código incorporando as bibliotecas já referidas pode ser observado na Figura 17.

<sup>3</sup> No caso não se utilizou este pino devido a leitura ser em estado contínuo

**Figura 17** - Parte Inicial do Código Fonte do Programa no Arduino IDE.

```
ProjetoFinal | Arduino 1.8.16
Arquivo Editar Sketch Ferramentas Ajuda

ProjetoFinal
1 #include <Wire.h>
2 #include <VL53L0X.h>
3 #include <LiquidCrystal_I2C.h>
4
5 //Definição dos pinos aos quais o XSHUT está ligado.
6 #define XSHUT_Sensor2 7
7 #define XSHUT_Sensor1 6
8
9 //Definição dos endereços dos sensores VL53L0X.
10 #define Sensor1_endereco 0x42
11 #define Sensor2_endereco 0x43
12
13 //Definição dos sensores.
14 VL53L0X Sensor1;
15 VL53L0X Sensor2;
16
17 //Display LCD 1 no endereço 0x26, 16 linhas e 2 colunas
18 LiquidCrystal_I2C display1(0x26,16,2);
19
20 //Display LCD 2 no endereço 0x27, 16 linhas e 2 colunas
21 LiquidCrystal_I2C display2(0x27,16,2);
22

Compilação terminada.
O sketch usa 10732 bytes (34%) de espaço de armazenamento para programas. O máximo são 30720 bytes.
Variáveis globais usam 486 bytes (23%) de memória dinâmica, deixando 1562 bytes para variáveis locais. O má
```

**Fonte:** Autor (2023).

## Resultados Obtidos

O conceito de melhoria foi baseado no instrumento utilizado (Figuras 18 e 19), priorizando a praticidade, otimização e principalmente precisão.

**Figura 18** – Régua de Inspeção utilizada.



**Fonte:** Autor (2023).

Figura 19 – Régua de Inspeção projetada.



Fonte: Autor (2023).

Com isso o instrumento trouxe resultados positivos, demonstrando facilidade no manuseio durante a inspeção do desalinhamento vertical (Figuras 20 e 21).

Figura 20 - Procedimento atual para inspeção do desalinhamento vertical.



Fonte: Autor (2023).

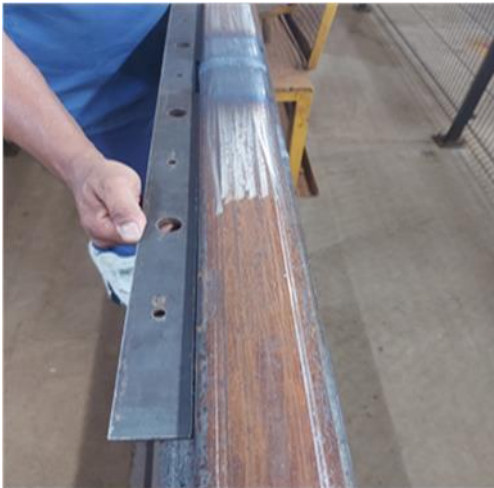
Figura 21 - Procedimento com o instrumento projetado.



Fonte: Autor (2023).

Também trouxe facilidades no manuseio de inspeção no desalinhamento lateral (Figuras 22 e 23), onde um operador/inspetor não era possível segurar a régua e o calibre tipo canivete ao mesmo tempo para aferição, sendo necessário uma segunda pessoa. Observando-se agora a possibilidade de trabalho individual.

Figura 22 - Procedimento atual para inspeção do desalinhamento lateral (necessita da segunda pessoa para o uso do calibre de folga).



Fonte: Autor (2023).

Figura 23 – Procedimento com o instrumento projetado (não necessita da segunda pessoa para inspeção do desalinhamento lateral) .



Fonte: Autor (2023).

Com a régua digital, a segunda pessoa não se faz necessário, isto porque o calibre tipo canivete foi substituído pelo sensor (Figuras 24 e 25). A utilização de displays, trouxe nitidez na leitura até mesmo em ambientes pouco iluminados onde antes dificultaria o trabalho do inspetor ou operador.

**Figura 24** - Procedimento atual para inspeção do desalinhamento vertical com o calibre de folga .



**Fonte:** Autor (2023).

**Figura 25** – Procedimento com o instrumento projetado, sem o uso do calibre de folga.

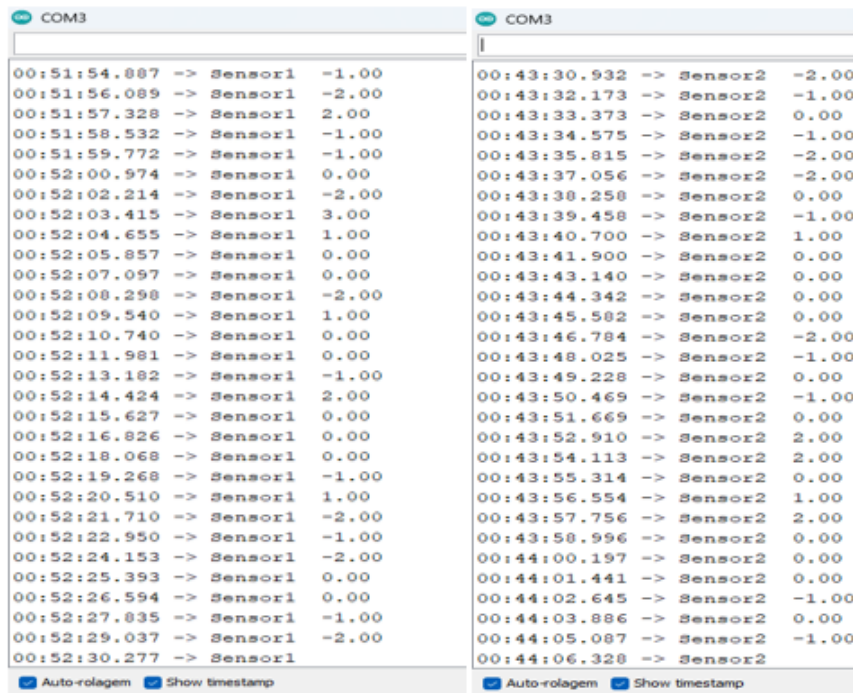


**Fonte:** Autor (2023).

Quanto ao uso do sensor, comparado ao calibre tipo canivete, foi significativo quanto a sua praticidade. Com calibre tipo canivete a verificação é realizada por tentativas até chegar a uma medida da folga do desalinhamento. Já através do sensor, a leitura é imediata e contínua. O sensor utilizado no projeto apresentou uma variação de  $\pm 2\text{mm}$  (Figura 26), prejudicial na precisão durante a leitura, isto porque o sensor VL53LoX é um sensor a laser que apresenta pequenas diferenças de distâncias, pois ele depende da cor do objeto, devido ao seu índice de reflexão.



**Figura 26** - Variação do Sensor1 e do Sensor2 registrado no monitor serial Arduino.



Fonte: Autor (2023).

## CONCLUSÃO

O projeto, apesar de ser um protótipo, trouxe propostas dinâmicas e inovadoras por meio da automação, enfatizando a evolução do processo de controle de qualidade da geometria da pista pós-soldagem.

A plataforma Arduino atendeu às expectativas em termos de programação, hardware e comunicação serial I2C. Do lado da detecção, o projeto exigia sensores com maior precisão milimétrica, que varia não mais que um centésimo de milímetro.

A inclusão de um botão de zero com programação Sketch adicional é necessária para autocalibrar o sensor em uma base plana e linear. Os cabos do circuito devem ser revestidos com espaguete de alta temperatura, isso porque a região central da régua recebe condutividade térmica da solda durante a inspeção.

Sendo assim pode-se afirmar que o objetivo do presente trabalho em desenvolver um protótipo que viesse apresentar sugestões para otimizar processos de inspeção geométrica do trilho ferroviário perlítico, pós soldagem, realizada dentro de um estaleiro ou em lugares apropriados dentro do padrão para o determinado processo foi plenamente alcançado.

## REFERÊNCIAS BIBLIOGRÁFICAS

AREMA apud Vicente, Bruno Sabadini. PROPOSTA DE ESPECIFICAÇÃO DE PROCEDIMENTO DE SOLDAGEM POR CENTELHAMENTO DE TRILHOS FERROVIÁRIOS (2017). Trabalho de Conclusão de Curso apresentado à Faculdade de Engenharia da Universidade Federal de Juiz de Fora como requisito parcial para a obtenção do título de Engenheiro Mecânico. Disponível em: <https://www2.ufjf.br/profab//files/2017/11/2017-1-TCC-Proposta-de-EPS-por-FBW-para-trilhos-ferroviarios.pdf> Acessado em: 29 de jan. de 2023.

EDITORA Bookman – Programação com Arduino: Começando com Sketches – Simon Monk – 2ª Edição (2017).

EDITORA Érica – Automação e Instrumentação Industrial com Arduino – Teoria e Projetos – Sérgio Luiz Stevan Jr. E Rodrigo A. Silva – 1ª Edição (2015).

EDITORA Manning Books – Arduino in Action – Martin Evans, Joshua Noble, Jordan Hochenbaum – 1ª Edição (2013).

Estevan Junior, S. L.; Silva, R. A. Automação e instrumentação industrial com Arduino: teoria e projetos. São Paulo: Erica, (2015).

FELIX, Camila. Procedimento Operacional Padrão (POP): como aplicar em empresas? (2022). Disponível em: <https://www.hostgator.com.br/blog/procedimento-operacional-padrao-como-aplicar-empresas/> Acessado em: 23 de jan. de 2023.

Fritzing. Eletronics mode easy (2022). Disponível em: <https://fritzing.org/> Acessado em 14 de fev. de 2023.

GITHUB. VL53LoX library for Arduino (2022). Disponível em: <https://github.com/pololu/vl53lox-arduino> Acessado em 19 de fev. de 2023.

SOUZA, Marco Polo Moreno. Curso Básico de Arduino. (2017) Disponível em: <http://www.arduino.unir.br/images/downloads/apostila-arduino.pdf> . Acessado em: 23 de jan. de 2023.

STMicroelectronics. VL53LoX. Time-of-Flight ranging sensor (2022). Disponível em: <https://www.st.com/resource/en/datasheet/vl53lox.pdf> . Acessado em 14 de fev. de 2023.

VIANA, Thiago Gomes. Implantação de controle de qualidade de soldas em trilho por centelhamento. (2014). Disponível em: <https://www.aeamesp.org.br/biblioteca/stm/20smtf1412Tt34ap.pdf> . Acessado em: 29 de jan. de 2023.