

## **SISTEMA DE AQUISIÇÃO DE DADOS EM TRATORES: DESENVOLVIMENTO DE UM PROTÓTIPO DE BAIXO CUSTO USANDO A PLATAFORMA ARDUINO**

**Mayron R. L. Ribeiro<sup>1,\*</sup>, Bárbara R. Morais<sup>2</sup>, Isabella R. C. de Faria<sup>1</sup>, Carlos R. Nolli<sup>1</sup> e  
Bruno A. S. Oliveira<sup>2</sup>**

<sup>1</sup>*Instituto Federal de Minas Gerais, Bambuí, MG, Brasil*

<sup>2</sup>*Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil*

**Palavras-chave:** Arduino, Máquinas agrícolas, Sensores, Sistema de Aquisição de Dados, Sistemas embarcados.

**Resumo.** A utilização de equipamentos de sensoriamento em máquinas agrícolas é uma prática cada vez mais comum e traz diversos benefícios para os produtores rurais e/ou pesquisadores. Uma das dificuldades encontradas no uso desta prática é o alto custo de aquisição dos equipamentos, dessa forma, o uso destes em máquinas agrícolas se torna bastante restrito. Com a finalidade de propor uma opção mais viável financeiramente, o presente trabalho tem como objetivo o desenvolvimento de um protótipo capaz de ler informações de diversos equipamentos e as armazená-las de maneira confiável. Para alcançar o que se propõe, a plataforma Arduino é utilizada como um Sistema de Aquisição de Dados que recebe informações de temperatura do ar, velocidade do trator, vazão de pulverização e coordenadas geográficas e as armazena. Para que o Arduino consiga interpretar corretamente os dados, se torna necessário o uso de módulos para que ocorra a comunicação entre os componentes de sensoriamento e a plataforma. Como resultado, tem-se que a plataforma Arduino é capaz de armazenar, de maneira confiável, as informações, tornando-se uma alternativa viável e de baixo custo para substituir os equipamentos que já existem no mercado para tal função.

---

E-mail addresses: [mayronreislacerda@gmail.com](mailto:mayronreislacerda@gmail.com)\*, [barbararobertamorais@ufmg.br](mailto:barbararobertamorais@ufmg.br), [isabella.regina@gmail.com](mailto:isabella.regina@gmail.com), [renato.nolli@ifmg.edu.br](mailto:renato.nolli@ifmg.edu.br), [brunoalbertobambui@ufmg.br](mailto:brunoalbertobambui@ufmg.br).

## **TRACTOR DATA ACQUISITION SYSTEM: DEVELOPMENT OF A LOW COST PROTOTYPE USING THE ARDUINO PLATFORM**

**Keywords:** Agricultural machinery, Arduino, Data Acquisition System, Embedded Systems, Sensors.

**Abstract.** The use of sensing equipment in agricultural machinery is an increasingly common practice and has many benefits for farmers and / or researchers. One of the difficulties encountered in this practice is the high cost of equipment acquisition, thus, their use in agricultural machines becomes very restricted. In order to propose a more financially viable option, this paper aims to develop a prototype capable of reading information from various equipment and storing it reliably. To achieve its purpose, the Arduino platform is used as a Data Acquisition System that receives temperature, velocity, flow and geographic coordinate information and stores it. For Arduino to be able to correctly interpret the data, it is necessary to use modules for communication between the sensing components and the platform. As result, the Arduino platform has been able to reliably store information, making it a viable and a low cost alternative to replace equipment already in the market for such a function.

## 1 INTRODUÇÃO

Máquinas agrícolas que fazem uso de instrumentos de medição modernos já existem no mercado e são empregadas na agricultura de precisão. Sistemas eletrônicos que possibilitam ao utilizador acompanhar informações referentes ao plantio, aplicação de insumos ou preparo da terra já estão presentes nestas máquinas. Armazenar dados de deslocamento, temperatura, vazão, ou outros dados necessários são operações simples para máquinas equipadas, porém seu alto custo de aquisição é um dos principais fatores para que os pequenos e médios produtores não utilizem tais equipamentos [6].

Para que as máquinas agrícolas se tornem fonte de dados, deve-se ter um conjunto de sensores para medição e um sistema que seja capaz de interpretar os dados adquiridos e armazená-los de maneira confiável e eficaz. Para tal função, existem diversas plataformas que podem ser utilizadas.

Entretanto, a plataforma Arduino se destaca por estar crescendo no mercado e ganhando cada vez mais visibilidade e entusiastas. Além de ser um dispositivo de baixo custo, o *hardware* e o *software* são *open source* e a plataforma ainda é compatível com diversos equipamentos [1].

No trabalho de [5] é utilizado um trator agrícola equipado com diversos sensores para analisar a diferença de consumo entre combustíveis distintos. Na pesquisa são comparados os óleos de Macaúba e Diesel, além de verificar a viabilidade de utilização do óleo de Macaúba como substituto para o óleo tradicionalmente utilizado. O autor utilizou diversos equipamentos de medição e um *Datalogger* para a captura e o armazenamento de dados. No presente estudo, foram utilizados os mesmos recursos que [5], exceto pela substituição do *Datalogger* CR1000 por uma proposta mais econômica.

Sabendo que os sistemas presentes no mercado são restritos por possuírem um alto preço como dito em [9, 10], uma alternativa de baixo custo para a substituição de um *Datalogger* torna-se vantajosa para os pequenos e médios produtores. A plataforma Arduino, além de ser muito versátil, possui um custo de aquisição relativamente baixo e isso reduz significativamente o valor final do produto.

O uso de mecanismos de medição em tratores pode ser o foco de diversos pesquisadores, no entanto, a confiabilidade dos dados é um fator muito importante e impacta diretamente nas pesquisas que forem realizadas. Além disso, a utilização de equipamentos com custo reduzido tornará os estudos mais acessíveis para os pesquisadores.

Diante desta situação, uma alternativa de baixo custo para produtores de menor porte torna-se bastante interessante, tendo em vista que o produto poderia ser amplamente explorado no mercado. O equipamento alternativo deve suprir de maneira satisfatória os instrumentos que já existem, apresentando um valor de compra inferior.

Portanto, nesta pesquisa é proposto o desenvolvimento de um sistema capaz de substituir com eficiência o equipamento *Datalogger* CR1000, já consolidado no mercado, pela plataforma Arduino Mega 2560. O sistema apresentado deve ser capaz de ler, armazenar e gerenciar dados de diversos sensores de maneira confiável.

## 2 ESTADO DA ARTE

A utilização da plataforma Arduino no meio agrícola é parte do estudo de [14], que realiza a automação de uma colhedora de café. O autor compara a automação realizada com Arduino e um controlador lógico programável (CLP). No trabalho, foram realizados dois experimentos, sendo o primeiro quando as rodas da colhedora estão em paralelo e o outro quando as rodas estão efetuando movimentos curvilíneos. A validação do experimento compara os resultados

em cada roda e em cada movimento, seja ele paralelo ou curvilíneo. Os resultados mostraram uma grande exatidão para os dois sistemas desenvolvidos. Quanto ao movimento em paralelo para todas as rodas, o sistema CLP foi mais preciso que o sistema Arduino. Para o movimento curvilíneo, os dois sistemas apresentaram grande exatidão, praticamente iguais estatisticamente.

O trabalho de [14] apresenta grande contribuição para o presente estudo, tendo em vista que o autor faz a comparação de um equipamento bastante difundido no meio industrial com a plataforma Arduino. A partir dessa pesquisa, é possível conhecer um pouco mais a capacidade e também as limitações da plataforma Arduino.

Foi desenvolvido na pesquisa de [3] um sistema para monitoramento da produção e consumo de energia em centrais de micro geração fotovoltaicas. Para realizar o monitoramento, o autor utilizou sensores de corrente contínua, sensores de tensão e rádio para a transmissão de dados. Nesta pesquisa, foram utilizadas duas plataformas Arduino. A primeira foi o modelo Mega 2560, que efetuou a coleta dos dados provenientes dos sensores, e para o recebimento dos dados foi utilizado o Arduino Duemilanove. O autor conclui sua tese afirmando que os resultados foram satisfatórios, em que 96% dos dados foram transmitidos corretamente. Assim, o Arduino mostrou-se uma boa solução para o problema proposto.

Analisando-se o estudo de [3], é visto que o autor utiliza a plataforma Arduino em situações em que é necessário uma grande precisão na leitura de dados provenientes de sensores. A proposta deste trabalho é a captação de dados de diversos sensores. Dessa forma, o autor reafirma a capacidade do Arduino para realizar o que é proposto nesta pesquisa.

Na pesquisa de [6] foi apresentado o desenvolvimento de um SAD de baixo custo para a obtenção e armazenamento de dados relacionados à tecnologia de aplicação, como pressão de trabalho, rotação da Tomada de Potência, temperatura e umidade relativa do ar e velocidade de deslocamento de um trator agrícola, além de obter as coordenadas geográficas e determinar a área trabalhada pela máquina por meio de GPS. A pesquisa tem como base a plataforma Arduino Mega 2560, responsável por coordenar os sensores utilizados e realizar a manipulação dos dados obtidos. O autor concluiu que o SAD desenvolvido é um equipamento robusto, de fácil manipulação e com custo acessível para pequenos e médios produtores.

Na mesma referência [6], é apresentada uma das principais contribuições para a realização do presente trabalho. Na dissertação, muitos fatores assemelham-se ao que foi aqui empregado. A utilização do Arduino como Sistema de Aquisição de Dados funcionou de forma eficiente e conseguiu atingir os objetivos determinados pelo autor.

A comparação do Arduino com outras plataformas é um fato comum na literatura, sendo que o CLP é o alvo de comparação do artigo de [15]. No experimento realizado, as duas plataformas em questão são empregadas para a utilização na irrigação. O objetivo do artigo é contribuir para a disseminação da agricultura moderna em que informações provenientes de diversos sensores auxiliam o agricultor em suas atividades rotineiras. A fim de verificar qual a melhor plataforma para determinada função, ambas são analisadas visando identificar qual apresenta maior confiabilidade e melhor custo benefício. A ideia apresentada no artigo sugere que sensores façam o monitoramento de várias áreas de irrigação e avisem ao agricultor caso seja necessário irrigar novamente aquela determinada área em que o sensor se encontra. Foram implementados dois projetos similares, um para Arduino e outro para o CLP, tendo o autor concluído que o Arduino é uma aplicação mais viável e destacou como pontos positivos o baixo custo, a versatilidade, o fácil acesso e a vasta quantidade de informação que se tem sobre a plataforma.

[15] constataram que o CLP é um equipamento mais robusto e isso o torna mais caro. Assim, a plataforma Arduino torna-se viável por ser mais acessível. Outro fato positivo para o

Arduino é a vasta documentação encontrada sobre ele. A linguagem de programação utilizada também deve ser levada em consideração, pois é considerada uma linguagem de fácil acesso se comparada ao CLP.

A agricultura é uma das principais atividades econômicas em diversos países, sendo o México um deles. Nesse sentido, o artigo de [8] mostra que a utilização de sistemas SCADA (*Supervisory Control and Data Acquisition*) poderia facilitar a vida dos produtores de menor porte. O principal problema seria o valor envolvido na aquisição desses sistemas, sendo assim, o autor propõe a utilização da plataforma Arduino para a diminuição de custos. No artigo, não é implementado nenhum equipamento, porém é realizada uma revisão de literatura que mostra a viabilidade de utilização do Arduino como um dos principais componentes na implementação de um sistema SCADA. O autor conclui seu trabalho indicando a utilização do Arduino em conjunto com a plataforma XBEE (são módulos de rádio), que realiza a comunicação remota entre o Arduino e a IHM, responsável pela tomada de decisão.

A revisão de literatura de [8] apresenta a versatilidade da plataforma Arduino, além de expor que sistemas como o SCADA podem fazer o uso do Arduino de maneira eficaz e satisfatória, fator motivante para o desenvolvimento dessa pesquisa. Como pôde ser encontrado no estado da arte, atualmente existem inúmeros estudos que utilizam da plataforma Arduino para automatizarem processos e surgirem como uma alternativa barata para um determinado equipamento. Dito isso, o atual trabalho tem por objetivo sanar os problemas encontrados nas demais soluções parecidas, além de servir como pioneira na substituição do *Datalogger* CR1000.

Além dos trabalhos mencionados, diversos outros autores [11, 13, 2, 4, 12] abordaram o mesmo tema. Grande parte das pesquisas foram aplicadas em ambiente controlado, o que é um diferencial do estudo proposto, uma vez que o mesmo também foi empregado e validado em campo. Além disso, o comparativo realizado no estudo foi entre o *Datalogger* CR1000 e o sistema proposto, sendo que, conforme já apresentado, o *Datalogger* CR1000 é amplamente utilizado nesse tipo de aplicação aqui discutida, o que o torna uma ferramenta extremamente interessante para ser comparada.

### 3 MODELO PROPOSTO

A solução proposta visa a diminuição de custos para que se possa fazer o sensoriamento do desempenho de um trator agrícola. A utilização da plataforma Arduino é o principal componente e colaborador para a redução de custos.

Para que o sistema pudesse atender de maneira satisfatória os usuários um levantamento de requisitos foi realizado. Assim, o sistema deve ler e armazenar dados de sensores de temperatura, de sensores de vazão, de sensores de velocidade, de uma célula de carga e de um GPS. As informações devem ser mostradas em uma tela onde a navegação entre as informações seja intuitiva e de fácil acesso.

A Figura 1 mostra as principais funcionalidades do sistema e suas interações com os usuários.

O modelo apresentado no presente trabalho deseja sensoriar temperatura, vazão, velocidade das rodas, esforço feito pelo trator quando ele é acoplado a um implemento agrícola e também as coordenadas geográficas em que o trator se encontra. Para facilitar a compreensão do que foi referido e do que foi descrito posteriormente a esta seção, a Figura 2 exhibe o diagrama de blocos do sistema.

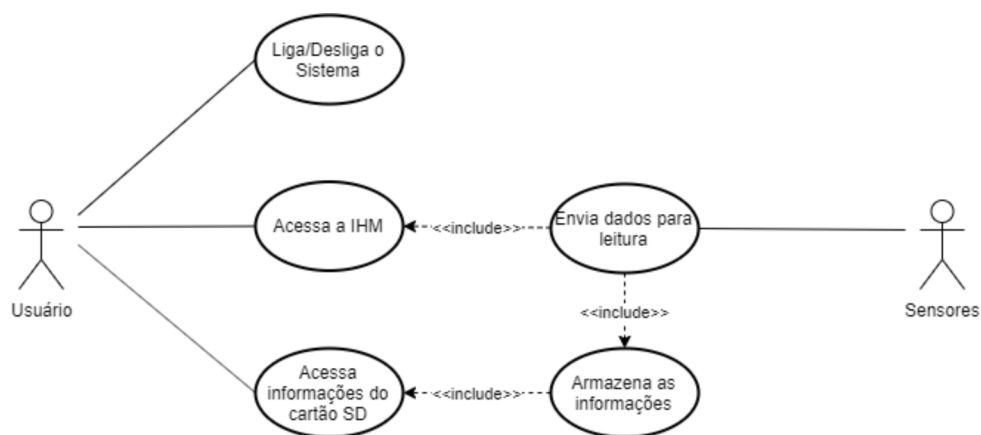


Figura 1: Diagrama de caso de uso do sistema

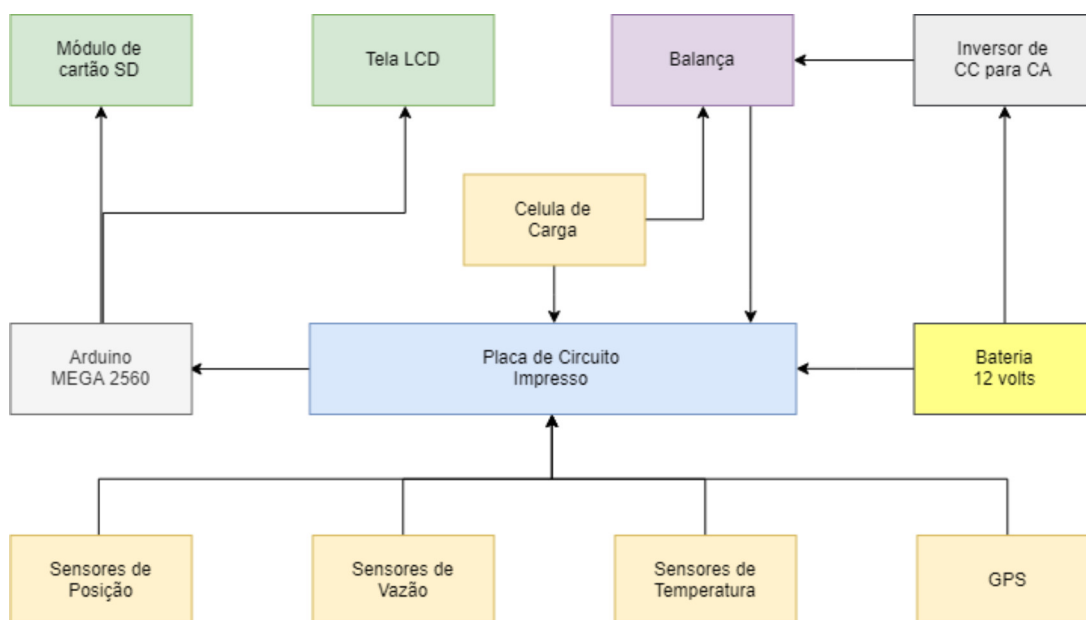


Figura 2: Diagrama de Blocos do sistema proposto

#### 4 MATERIAIS UTILIZADOS

O trator agrícola utilizado é da marca John Deere, modelo 5603. Salienta-se que todo o trabalho gira em torno desse trator, devido ao sistema apresentado permitir armazenar as informações de diversos sensores que podem ser instalados na máquina, sendo útil em várias atividades. O trator também fornecerá ao projeto toda a alimentação necessária, que foi proveniente da sua bateria.

A Plataforma Arduino Mega 2560 trata-se do principal componente deste estudo, pois todos os sensores foram ligados na plataforma, que fará a leitura das informações e posteriormente armazenará os dados em um cartão SD. A Interface Homem Máquina também foi controlada pela plataforma.

O Arduino Uno não faz parte do sistema final desta pesquisa, porém ele é utilizado como emissor de sinais para simulação dos sensores. O Arduino Uno foi de suma importância no desenvolvimento deste estudo, pois evitou a utilização de alguns sensores nas fases iniciais do projeto.



O sensor de temperatura utilizado é do tipo PTC, o qual leva esse nome pois é feito de platina, que é um elemento químico de símbolo Pt, e também porque a 0° C o sensor tem resistência igual a 100  $\Omega$ . O PT100 foi responsável por sensoriar a temperatura do combustível na entrada e saída do tanque. A Figura 3 apresenta os sensores instalados no trator.

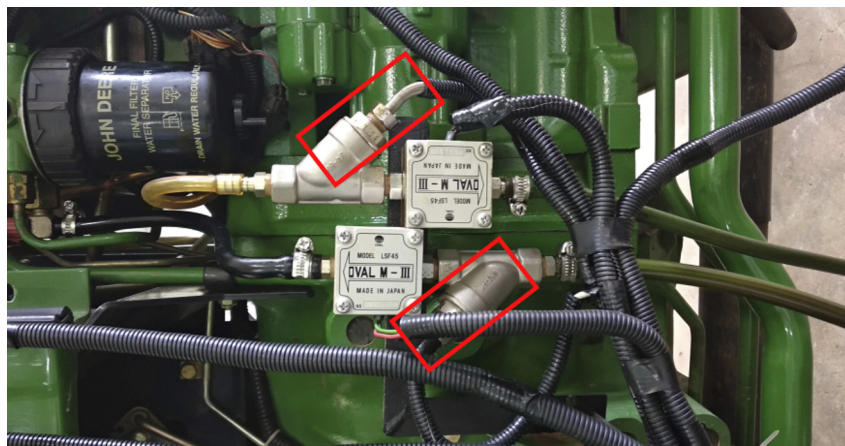


Figura 3: PT100 instalado no trator

O modelo utilizado para o sensor de velocidade foi o PRL30-10DP, que se refere a quatro sensores indutivos que foram acoplados às rodas do trator. Os sensores estão posicionados para detectar a passagem dos parafusos da roda e indicar, por pulsos, sempre que este evento ocorrer. A Figura 4 apresenta o posicionamento dos sensores nas rodas do trator.

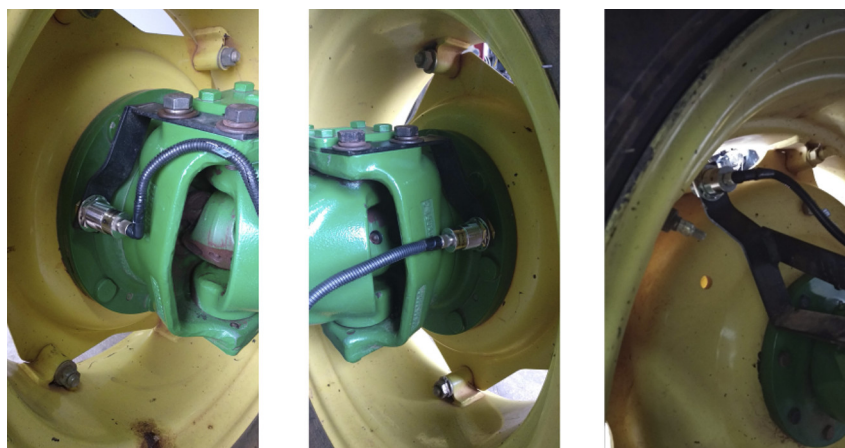


Figura 4: Sensor de velocidade instalado no trator

O sensor de vazão utilizado foi o LSF5 Oval M – III. O sensor têm como função captar o fluxo de entrada e de saída de combustível. A cada 30 mililitros de combustível que passa pelo equipamento é emitido um pulso. Sendo assim, é possível mensurar a vazão nos sensores por meio de manipulações matemáticas. A Figura 5 mostra os sensores instalados.

A célula de carga utilizada ficou instalada na parte traseira do trator, onde geralmente são acoplados os implementos agrícolas. Dessa forma, a célula de carga tem como função registrar a força de tração entre o trator e o implemento agrícola. Assim, é possível medir o esforço realizado pelo trator. A Figura 6 exhibe a célula de carga acoplada.

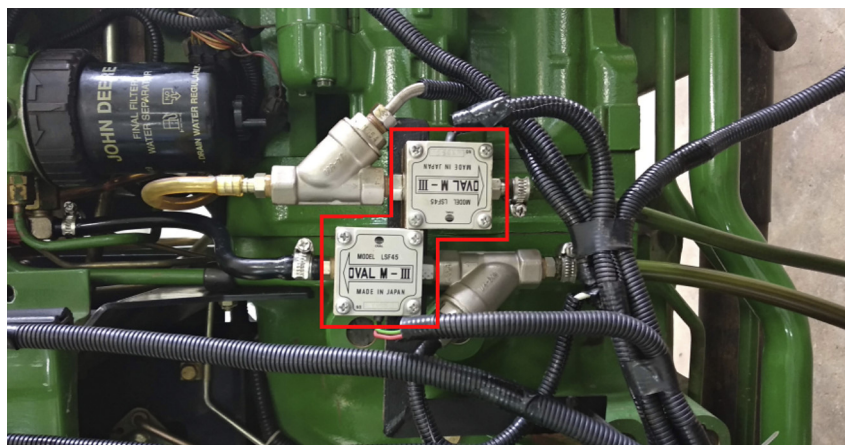


Figura 5: Sensor de vazão Oval M-III instalado

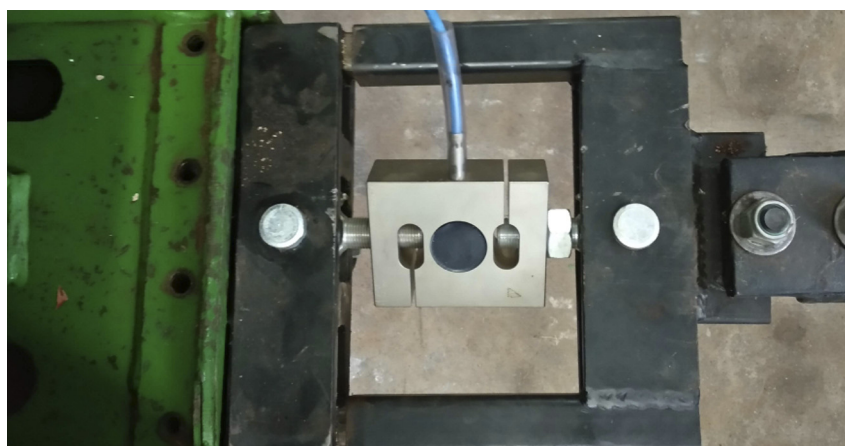


Figura 6: Célula de carga instalada no trator

No sistema proposto, a balança tem como função auxiliar no sensoriamento da célula de carga. Dessa maneira, a célula de carga não se comunica diretamente com o Arduino, uma vez que ela é ligada na balança e esta, por sua vez, informa ao Arduino por meio de comunicação serial sobre os dados que estão sendo medidos.

Ao contrário de todos os outros componentes deste trabalho, a alimentação da balança é um caso a parte, necessitando de 127 V em tensão alternada, e portanto, sendo necessário um conversor de corrente contínua para alternada CC/CA (inversor de frequência).

O inversor de 400 W 12 VDC/127 V é produzido pela Hayonic. O equipamento recebe a tensão da bateria do trator e a transforma em 127 V.

O GPS de modelo 16X-HVS é fabricado pela Garmin. Deste dispositivo foram obtidas informações de latitude, longitude, data, hora e velocidade de deslocamento. Os dados foram enviados para o Arduino por meio de comunicação serial RS232. A Figura 7 apresenta o GPS utilizado.

*Secure Digital Card* ou Cartão SD é o responsável por armazenar as informações coletadas. O cartão armazena apenas um arquivo no formato CSV (*Comma-separated values*), e em cada coluna do arquivo foram gravados dados de um determinado equipamento.

A capacidade de armazenamento do cartão utilizado é de 32 Gigabytes, mas poderia ser menor, pois o Arduino tem capacidade de gravação limitada a quatro Gigabytes. Assim, mesmo que a capacidade do cartão seja superior a quatro Gigabytes, a plataforma limita seu uso.





Figura 7: GPS instalado no trator

As informações coletadas de diversos sensores foram armazenadas em um cartão SD. O módulo funciona por meio de uma interface SPI e, dessa maneira, os dados são gravados de forma síncrona.

O *display* LCD (*Liquid Crystal Display*) 1602A foi utilizado para a visualização das informações em tempo real, sendo, portanto, parte fundamental da IHM do sistema. O *display* conta com duas linhas de 16 caracteres cada. O componente possui 16 pinos, dos quais são utilizados oito para a transmissão de dados, três para alimentação, três de controle e dois para a luz de fundo.

O teclado matricial de Membrana foi utilizado na IHM do projeto. O componente conta com 16 teclas em que cada uma é associada a alguma informação dos sensores. Cada tecla faz a conexão entre as informações que estão sendo lidas pelo Arduino e as exibe no *display* LCD.

O Driver Conversor de Nível trata-se de um receptor que tem como finalidade converter TIA/EIA-232-F para níveis TTL/CMOS 5V. O componente pode receber entradas de até 30 V e fornece sinais de saída de 5 V, além de inverter a tensão negativa, se for o caso.

Um problema encontrado na alimentação do LCD é a corrente que o mesmo demanda do Arduino. Quando esse valor é mais elevado, pode ser que o Arduino não consiga suprir tal necessidade. Por isso, faz-se necessário o uso de um regulador de tensão que tem como objetivo garantir a tensão final de 5 V e uma corrente de até 1 A. A tensão deve ser de 5 V porque é esse valor que o LCD precisa.

A placa de fenolite foi utilizada para a fabricação da placa do circuito. É nela que foram soldados todos os componentes necessários para que os sensores fossem lidos de forma adequada pelo Arduino. A plataforma Arduino também foi acoplada à placa com a finalidade de organizar e facilitar a ligação dos diversos componentes. Assim, os equipamentos de medição se ligaram ao Arduino por meio dos barramentos da placa de circuito gerada.

No geral, foram utilizados 18 resistores, sendo 12 destinados aos divisores de tensão, os quais foram necessários nos sensores de velocidade e nos sensores de vazão. Para a ligação dos sensores de temperatura, foram empregados dois resistores, enquanto o teclado utilizou quatro resistores.

*Bornes* foram utilizados na placa de circuito para que todos os componentes fossem ligados sem que precisem ser soldados. No Total, foram empregados 12 *bornes*, sendo um para cada componente de leitura, um para alimentação e um para saída de tensão.

A IDE utilizada é a de versão 1.8.8 e é nela que foi codificado o algoritmo desenvolvido. A linguagem de programação adotada foi a C++ e o paradigma de programação foi o estruturado.

## 5 METODOLOGIA

Com base nos *datasheets* dos componentes, foi verificado que os itens precisariam de um sistema adicional para funcionar, isto é, eles não poderiam ser ligados diretamente na plataforma Arduino sem um tratamento prévio.

Para que se pudesse dar início ao desenvolvimento do algoritmo utilizado para a leitura e armazenamento dos dados colhidos, um Arduino auxiliar foi usado para simular os sensores. Assim, o Arduino Uno foi o responsável pela referida simulação.

A simulação foi realizada de diferentes maneiras. Para os sensores de velocidade, é sabido que eles trabalham com emissão de pulsos sempre que detectam algum elemento metálico. Assim, foi emitido um pulso em um intervalo de tempo determinado previamente. A leitura desse pulso emitido pelo Arduino Uno é realizada pelo Arduino Mega 2560 e, dessa maneira, pôde-se trabalhar como se fosse o sensor real.

Para o sensor de vazão, o procedimento utilizado foi o mesmo, já que o sensor também funciona por meio de pulsos, diferindo nas manipulações realizadas posteriormente em nível de código. A Figura 8 apresenta o momento em que os sensores eram simulados.

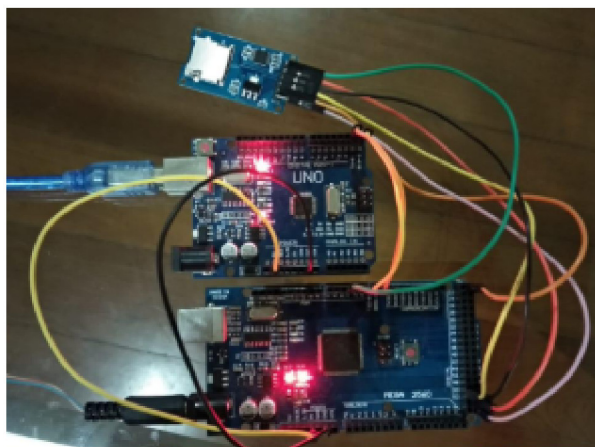


Figura 8: Simulação dos sensores com auxílio do Arduino

O GPS e a célula de carga transmitem dados por meio de comunicação serial, assim, inicialmente, os componentes foram lidos com auxílio do *software* Putty, o qual consegue ler o que está chegando em uma entrada USB de um computador. Entendendo como os dados são transmitidos, o Arduino Uno também simulou os componentes, enviando *strings* de caracteres idênticos aos que foram lidos pelo Putty.

O sensor PT100 foi o único que não foi simulado pelo Arduino Uno. Como se trata de uma leitura analógica de dados, o sensor foi simulado com o auxílio de um potenciômetro, já que a variação de resistência nesse componente é similar ao do sensor PT100.

### 5.1 Montagem dos Sensores de Velocidade

Os sensores de velocidade são empregados para sensoriar a velocidade de movimento em cada uma das rodas do trator. Nessas rodas, são contabilizados oito parafusos em cada, que estão posicionados equidistantes um do outro, dessa forma, a cada inclinação de 45 graus, tem-se um parafuso. O cálculo da velocidade em cada roda do trator é realizado por meio do período entre um pulso e outro e também pelo ângulo entre os parafusos.

Para verificar a que velocidade se encontra a roda no momento da leitura, deve-se, primeiramente, calcular a velocidade angular,  $\omega$ , que é dada pela diferença entre um deslocamento angular e o período deste deslocamento:

$$\omega = \frac{\Delta\Theta}{\Delta T} = \frac{2\pi}{\Delta T} = \frac{\frac{2\pi}{8}}{\Delta T} = \frac{\frac{\pi}{4}}{\Delta T} = \frac{\pi}{4\Delta T}. \quad (1)$$

Na Eq. (1),  $\Delta\Theta$  tem valor de  $360^\circ$  ou  $2\pi$ , já que se refere à circunferência da roda. Sabendo-se que se tem oito parafusos equidistantes, o valor é dividido por oito. O período,  $\Delta T$ , é o tempo entre um pulso e outro. Conhecendo tal valor, a velocidade angular em uma determinada roda pode ser encontrada. O próximo passo é a determinação da velocidade linear, para isso é necessário ter conhecimento do raio de cada roda. Das quatro rodas do trator, as duas traseiras são maiores que as dianteiras e medem 1,48 m de diâmetro. As rodas dianteiras medem 1,12 m de diâmetro. A equação para encontrar a velocidade linear,  $V$ , é:

$$V = \omega r. \quad (2)$$

Tendo conhecimento dessas equações foi possível codificar um algoritmo para interpretar cada pulso recebido pelo sensor e, assim, armazenar a velocidade das rodas.

Após a simulação, foi necessário desenvolver um divisor de tensão para que o sensor pudesse ser ligado ao Arduino. Essa ação não pôde ser feita de maneira direta porque o sensor trabalha com 12 V e o Arduino opera apenas com 5 V. Dessa forma, ligar diretamente o sensor poderia danificar permanentemente o Arduino.

A técnica do divisor de tensão teve por objetivo manipular a tensão elétrica de entrada ( $V_{in}$ ), de forma que a tensão de saída ( $V_{out}$ ) seja um valor desejável.

## 5.2 Montagem dos Sensores de Vazão

O sensoriamento da vazão do tanque de combustível exige um processo similar ao que foi aplicado nos sensores de velocidade, sendo que o que muda é basicamente a formulação matemática aplicada. Sempre que é detectado um fluxo de 30 ml, o sensor emite um pulso. Sendo assim, é necessário saber a quantidade de pulsos emitidos,  $N$ , e o tempo decorrido desde o início do monitoramento,  $\Delta T$ . A equação que relaciona essas variáveis com a vazão,  $V_a$ , é

$$V_a = \frac{(0,03)N}{\Delta T}. \quad (3)$$

## 5.3 Montagem dos Sensores de Temperatura

A variação de temperatura é a única medição realizada de forma analógica. A temperatura pode chegar até no máximo  $100^\circ\text{C}$ . Por conseguinte, foram trabalhados para que a temperatura pudesse ser medida entre  $0^\circ\text{C}$  e  $100^\circ\text{C}$ .

## 5.4 Montagem do receptor de sinal GNSS

O receptor de sinal GNSS Garmin 16X-HVS foi responsável por fornecer dados geográficos e de posicionamento do trator, além de fornecer a hora atual. Para a comunicação, foram utilizadas as portas seriais, RX/TX. O receptor GNSS não pôde ser ligado diretamente ao Arduino porque a tensão de saída chega a picos de 9 V negativos, por isso, é necessário a utilização de um circuito para diminuir e inverter a tensão. Para isso, o componente MAX232 foi empregado.

Para o perfeito funcionamento do componente, foi necessária a utilização de cinco capacitores. As portas 13 e 8 foram utilizadas para entrada dos dados provenientes do receptor GNSS e da balança. Já as portas 12 e 9, foram as saídas que se ligaram ao Arduino.

O receptor GNSS se comunica enviando os dados por meio de uma *string*, em que o protocolo utilizado é o NMEA (*National Marine Electronics Association*) e por esse motivo uma série de regras são adotadas. De acordo com [7], as regras são as seguintes:

- início de transmissão deve começar com \$;
- os cinco caracteres seguintes são para identificação, sendo os dois primeiros para identificar a origem da mensagem e os outros três para identificar o tipo de mensagem;
- todos os campos são delimitados por vírgulas;
- quando não há dado para transmissão, o *byte* deve receber valor nulo;
- quando houver checagem de dados (*checksum*), o primeiro caracter do último campo deve ser obrigatoriamente o "\*"; e
- o *checksum* é representado por dois dígitos em hexadecimais depois do "\*".

O receptor de sinal GNSS pode ser configurado para fornecer três diferentes tipos de *strings* de dados. Os formatos são GPRMC, GPGGA, GPVTG. Os dois primeiros são enviados por padrão e a terceira deve ser configurada via *software*.

A Tabela 1 apresenta como a sentença identificada por \$GPRMC dispõe os dados e a Tabela 2 mostra como os dados da segunda sentença são dispostos na *string*. As tabelas supracitadas estão de acordo com o manual do receptor GNSS.

Tabela 1: Identificação dos dados GPRMC

Campo	Exemplo	Observação
ID da sentença	\$GPRMC	-
Hora UTC	092204.999	hhmmss.sss
Status	A	A=Válido, V=Inválido
Latitude	4250.5589	ddmm.mmmm
Indicador N/S	N	N = Norte, S = Sul
Longitude	14718.5084	dddmm.mmmm
E/W Indicador	E	E = Leste, W = Oeste
Velocidade	0.00	Nós
Curso	0.00	Graus
Data UTC	211200	DDMMAA
Var. Magnética	-	Graus
Var. Magnética	-	E = Leste, W = Oeste
Checksum	*25	-
Terminador	CR/LF	-

Como não são todos os dados necessários para o sistema proposto, algumas posições da *string* foram desprezadas.

## 5.5 Módulo de cartão SD

O armazenamento das informações é uma das partes mais importantes do trabalho e o responsável por esse quesito é o Módulo de Cartão SD. Este suporta os formatos FAT16 e FAT32 e ele pode ser alimentado com 3,3 V ou 5 V.

O dispositivo utiliza comunicação SPI. Esta é sempre *Full-Duplex*, assim, a troca de dados acontece em ambas as direções. A Figura 9 ilustra o funcionamento da comunicação entre o módulo (Mestre) e o/os Escravo (s) (Arduino).

Tabela 2: Identificação dos dados GPGGA

Campo	Descrição	Observação
0	\$GPGGA	NMEA Identificador da sentença
1	hhmmss	UTC da posição Horas, minutos, segundos
2	1111.111	Latitude: Graus, minutos, milésimos de minutos
3	a	N = Norte, S = Sul
4	nnnnn.nnn	Longitude: Graus, minutos, milésimos de minutos
5	b	E = Leste, W = Oeste
6	t	Qualidade dos GPS, 0 = Sem GPS, 1 = GPS, 2 = DGPS
7	uu	Número de satélites em uso
8	v,v	Diluição horizontal de posição (HDOP)
9	w.w	Altitude da antena em metros
10	M	M = metros
11	x.x	Separação Geoidal em metros
12	M	M = metros
13	y.y	Tempo dos dados
14	zzzz	ID da estação de referência
15	*	Carácter que finaliza a string
16	hh	Checksum
17	<CR><LF>	Terminador

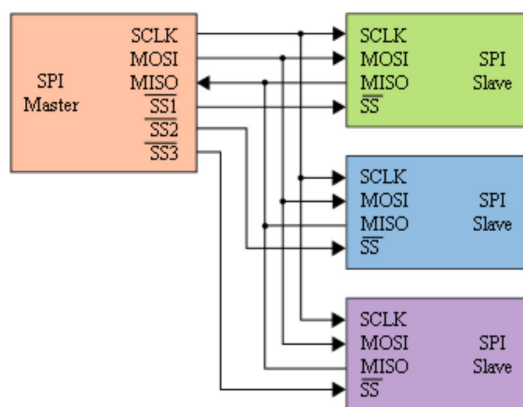


Figura 9: Comunicação SPI

Na Figura 9, tem-se a opção para conexão de até três dispositivos escravos. No módulo em questão, existe apenas um seletor SS. Mesmo que a função pareça desprezível, já que existe apenas um escravo, o módulo pode utilizar o SS para sincronismo de *frame*.

A alimentação do dispositivo foi realizada pelo Arduino e os pinos devem ser ligados especificamente em portas predeterminadas. A Tabela 3 informa a ligação que deve ser respeitada para o funcionamento correto do dispositivo.



Tabela 3: Ligação do Módulo de cartão no Arduino

Módulo	Arduino
MISO	50
MOSI	51
SCLK	52
SS	4

## 5.6 Módulo da Célula de Carga

A célula de carga informará para o usuário qual a força realizada pelo trator para puxar outro equipamento. Esta seção é intitulada como módulo da célula de carga porque, diferentemente de outros materiais utilizados, ela precisa trabalhar com alguns equipamentos extras para transmitir os dados ao Arduino.

Os componentes que compõem este módulo são, obviamente, a célula de carga, uma balança na qual a célula de carga foi ligada, um inversor de tensão responsável por alimentar a balança e o componente MAX232.

Os dados são transmitidos via comunicação serial entre a balança e a plataforma Arduino.

## 5.7 Desenvolvimento da IHM

A IHM é responsável por informar ao usuário o valor atual de cada medição. É composta por um *display* LCD, um teclado de 16 teclas, uma chave liga/desliga e ainda um *led*. No *display*, foi mostrado separadamente o valor de cada sensor ou do receptor de sinal GNSS. As teclas são responsáveis pela navegação entre as informações e o *led* serve para identificar se a chave liga/desliga está ativada.

## 5.8 Desenvolvimento da Placa de Circuito Impresso

Com a finalidade de agrupar em uma única placa todos os circuitos necessários, a placa de circuitos foi uma peça única. Dessa forma, os componentes que não foram acoplados diretamente na placa terão seus cabos de alimentação e de dados ligados a ela.

Para o desenvolvimento da placa, é necessário, inicialmente, o uso do *software* Fritzing para que ele possa fornecer o modelo final das trilhas e o posicionamento de cada um dos componentes. A placa pôde, então ser confeccionada e ter os componentes soldados.

## 6 RESULTADOS E DISCUSSÃO

Desenvolvido o sistema, foi necessário testar e validar os dados armazenados, para isso, primeiro foi realizada a leitura dos dados provenientes da célula de carga e do receptor GNSS com o *software* Putty. A leitura do receptor pode ser visualizada na Figura 10.

A captura desses dados é de suma importância para o desenvolvimento do algoritmo capaz de interpretar os dados e os armazenar de maneira intuitiva. Os sensores que trabalham por meio de pulso são os personagens da próxima etapa. A simulação realizada pelo Arduino Uno proporcionou um ambiente adequado para testar a velocidade e a vazão que foram armazenadas em um arquivo CSV posteriormente.

No *software* Fritzing, foi realizado o desenvolvimento do PCB (*Printed circuit board*) da placa de circuitos, a qual é responsável pelo desenho das trilhas de ligação das placas e das ilhas onde os componentes foram soldados. A Figura 11 mostra o resultado final da PCB desenvolvida.

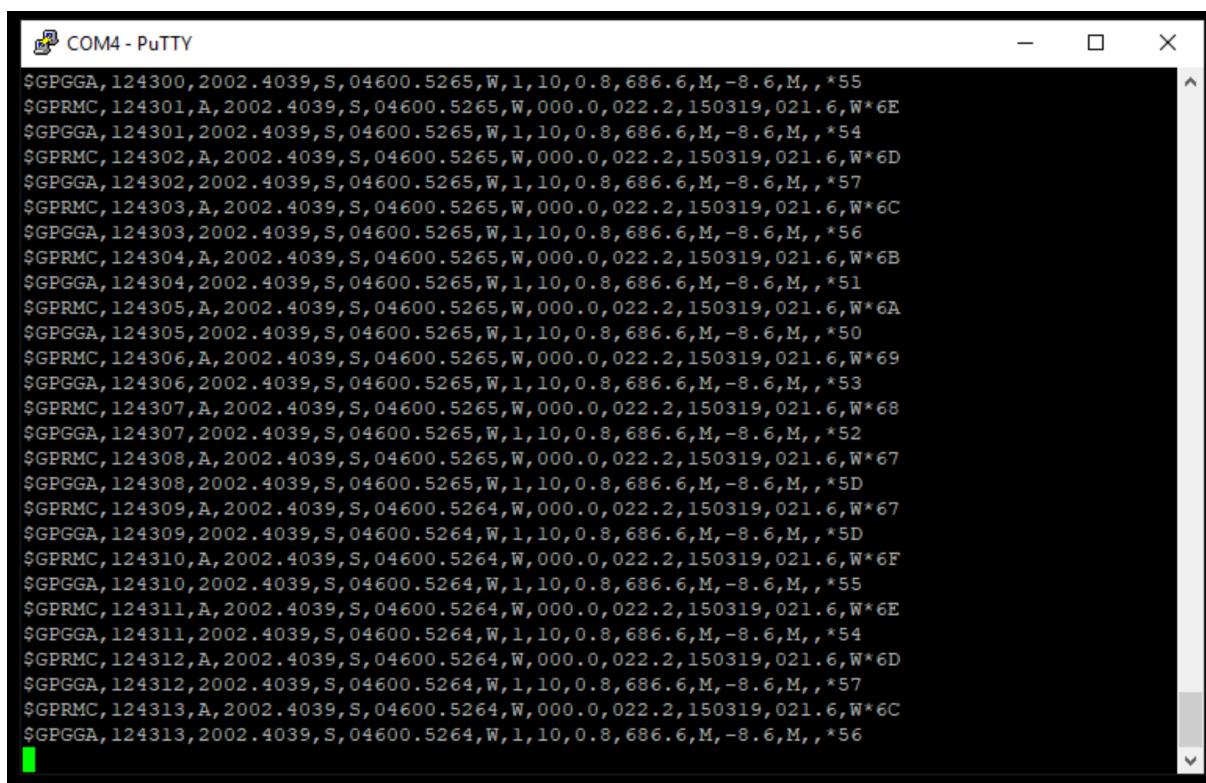


Figura 10: Leitura do receptor GNSS pelo *software* Putty

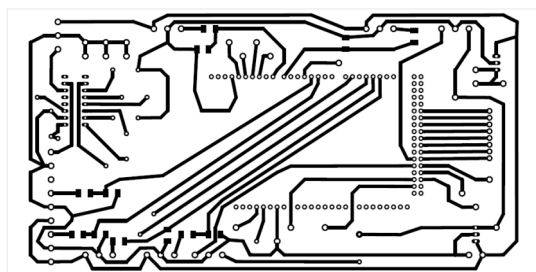


Figura 11: PCB gerado

O passo seguinte foi a confecção da placa. Nesta etapa, foi necessário passar o desenho para a placa de fenolite, corroer a placa com percloroeto de ferro, furar as ilhas onde os componentes se encaixam e, por fim, soldar. O resultado final destas etapas pode ser vistos na Figura 12. O diagrama esquemático gerado pode ser encontrado no endereço eletrônico (link ocultado devido o processo de revisão por pares).

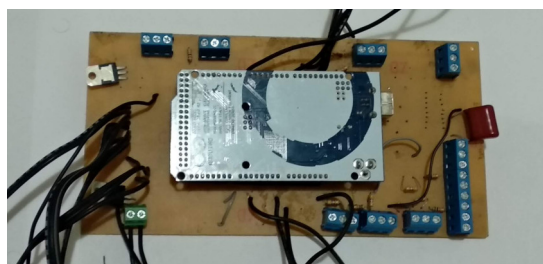


Figura 12: Placa de circuitos pronta

Com todos os componentes soldados, a placa foi colocada em uma caixa que tem como função proteger o equipamento e também deixar a IHM de fácil acesso para o usuário. O resultado pode ser observado na Figura 13.

Para os sensores de velocidade, foi necessário o uso de interrupções do tipo *RISING*, essas que são disparadas apenas na borda de subida, para contabilizar quando um pulso é disparado. O uso de interrupções torna-se necessário tendo em vista que os eventos podem ser gerados diversas vezes em um curto espaço de tempo. Um erro de leitura em uma das rodas poderia influenciar significativamente nos dados gerados, indicando, por exemplo, patinagem em uma roda quando esta não ocorre.



Figura 13: Caixa para proteção dos circuitos e de fácil acesso a IHM

Para os sensores de vazão, que também trabalham com pulsos, o uso de interrupções não foi necessário. Tal fato ocorre devido aos eventos serem gerados em intervalos de tempo mais longos. Sendo assim, os sensores foram ligados em pinos digitais do Arduino que contabilizam sempre que um pulso é disparado.

Para os sensores de temperatura, foi necessário a utilização da biblioteca *OneWire*, que permite que o sensor de temperatura seja lido em uma porta digital via PWM (*Pulse Width Modulation*). Os sensores de temperatura utilizados no modelo final não foram os PT100, pois os sensores originais estavam instalados no trator e o acesso a eles para realização de testes se tornou inviável, sendo assim, utilizou-se sensores do modelo DS18B20.

A leitura correta dos dados do receptor GNSS só foi possível com a utilização da biblioteca *TinnyGPS*. Os dados chegam via comunicação serial e são listados de acordo com a necessidade. Para o modelo final, apenas alguns dados do receptor GNSS foram selecionados para serem gravados, sendo eles: hora, data, latitude, longitude, altitude e velocidade.

Como já evidenciado anteriormente, a célula de carga é ligada a uma balança e essa, por sua vez, transmite os dados para o Arduino por meio de comunicação serial. Tendo em vista que o Arduino Mega 2560 possui quatro portas desse tipo, a leitura paralela de duas fontes de dados distintos não deveria ser um problema. Durante os testes realizados, a obtenção de dados via comunicação serial provenientes de diferentes fontes apresentou problemas, em que os dados de apenas uma porta de comunicação serial eram salvos. Por esse motivo, a leitura serial de duas fontes não foi possível e a preferência foi dada ao receptor GNSS que transmite uma quantidade de informações maior que a célula de carga.

Para o armazenamento de dados no cartão SD, foi necessária a utilização das bibliotecas *SPI* e *SD*. O armazenamento das informações foi feito concatenando as variáveis de cada sensor em

uma *string* e então salvá-la uma única vez, dessa forma, a gravação só foi efetuada uma vez por segundo.

O desenvolvimento da IHM do sistema que é composto pelo *display* LCD, pelo teclado 4x4, por um potenciômetro, por uma chave liga/desliga e por um *led* foi o passo final do algoritmo desenvolvido. Foi preciso utilizar a biblioteca LiquidCrystal para o *display*, já os outros componentes não precisaram de bibliotecas para o funcionamento. Como resultado principal da IHM, tem-se a mostra de informações com o pressionamento de uma tecla. A Tabela 4 mostra como ficou a relação de teclas e informações.

Para que outros pesquisadores possam dar continuidade ao estudo e o reproduzirem, o algoritmo implementado pode ser encontrado no endereço eletrônico (link ocultado devido o processo de revisão por pares).

Tabela 4: Relação de tecla e informação a ser exibida no *display*

Tecla	Função
1	Velocidade roda dianteira direita
2	Velocidade roda dianteira esquerda
3	Velocidade roda traseira direita
4	Velocidade roda traseira esquerda
5	Vazão de entrada
6	Vazão de saída
7	Sensor de temperatura 1
8	Sensor de temperatura 2
9	Latitude e Longitude
0	Altitude
A	Velocidade
B	Hora e Data

Com o algoritmo executando, pôde-se salvar os dados em um cartão SD no formato CSV, em que cada coluna contém armazenado uma informação. A Tabela 5 mostra a relação entre a coluna do arquivo e a informação armazenada nele.

Tabela 5: Relação de coluna e informação do arquivo CSV

Coluna	Informação
1	Label
2	Velocidade roda dianteira direita
3	Velocidade roda dianteira esquerda
4	Velocidade roda traseira direita
5	Velocidade roda traseira esquerda
6	Vazão de entrada
7	Vazão de saída
8	Sensor de temperatura 1
9	Sensor de temperatura 2
10	Latitude e Longitude
11	Altitude
12	Velocidade
13	Hora e Data

Por fim, a etapa final deste trabalho foi o armazenamento das informações. Assim, um ar-

quivo no formato CSV é gerado. Com os sensores ligados ao sistema desenvolvido, foi possível coletar todos os dados desejáveis. Um fragmento dos dados podem ser observados na Figura 6.

Tabela 6: Resultados obtidos após a gravação

Label	Data	Hora	Latitude	Altitude	Longitude	Vel. GPS	Vel. 01	Vel. 02	Vel. 03	Vel. 04	Vazão 01	Vazão 02	Tempo 01	Tempo 02
Leitura	04/09/2019	13h8m11s	-20.04	-46.00	693.20	0.00	5.93	5.93	7.84	7.84	0.02	0.02	26.13	26.13
Leitura	04/09/2019	13h8m12s	-20.04	-46.00	693.20	0.00	5.93	5.93	7.84	7.84	0.02	0.02	26.06	26.06
Leitura	04/09/2019	13h8m13s	-20.04	-46.00	693.20	0.00	5.93	5.93	7.84	7.84	0.03	0.03	26.13	26.13
Leitura	04/09/2019	13h8m14s	-20.04	-46.00	693.20	0.00	5.93	5.93	7.84	7.84	0.03	0.03	26.13	26.13
Leitura	04/09/2019	13h8m15s	-20.04	-46.00	693.20	0.00	5.93	5.93	7.84	7.84	0.03	0.03	26.13	26.13
Leitura	04/09/2019	13h8m15s	-20.04	-46.00	693.20	0.00	5.93	5.93	7.84	7.84	0.03	0.03	26.13	26.13
Leitura	04/09/2019	13h8m17s	-20.04	-46.00	693.20	0.00	5.93	5.93	7.84	7.84	0.03	0.03	26.13	26.13
Leitura	04/09/2019	13h8m18s	-20.04	-46.00	693.20	0.00	5.93	5.93	7.84	7.84	0.03	0.03	26.13	26.13
Leitura	04/09/2019	13h8m18s	-20.04	-46.00	693.20	0.00	5.93	5.93	7.84	7.84	0.03	0.03	26.13	26.13
Leitura	04/09/2019	13h8m20s	-20.04	-46.00	693.20	0.00	5.93	5.93	7.84	7.84	0.04	0.04	26.16	26.16

A velocidade das rodas diferem-se nas dianteiras e traseiras. Isso se dá pelo fato de que os pulsos para os quatro sensores estão sendo gerados dentro do mesmo intervalo de tempo e, por isso, as rodas traseiras estão mais rápidas. Essa diferença entre as velocidades das rodas é esperada, já que em ambiente real as rodas traseiras tendem a receber menor quantidade de pulsos por intervalo de tempo, devido ao espaço entre cada parafuso ser maior. As informações de vazão e de temperatura são dados como o esperado.

## 7 CONSIDERAÇÕES FINAIS

O protótipo desenvolvido na presente pesquisa teve como principal proposta a redução de custos de um sistema de aquisição de dados. A plataforma Arduino mostrou-se uma excelente opção para tal função, tendo em vista que é um equipamento acessível financeiramente e que fornece um ambiente em que o desenvolvimento de novos projetos é facilitado.

O material utilizado no desenvolvimento do protótipo do Sistema de Aquisição de Dados teve um valor final de R\$ 150,00, porém a mão de obra envolvida no desenvolvimento do projeto pode ser avaliada em R\$ 4.000,00. Observa-se, portanto, que para o desenvolvimento e implantação do sistema o valor final é definido como R\$ 4.150,00. O *Datalogger* CR1000 tem um custo de aquisição de R\$ 16.100,00, dessa forma, o sistema desenvolvido é uma ferramenta de baixo custo que pode substituir com eficiência o *Datalogger* CR1000. Os valores citados nesta seção são referentes a cotação realizada no ano de 2021.

Desenvolvido o sistema, pôde-se constatar que o Arduino Mega 2560 fez a aquisição de dados de diversos equipamentos e os armazenou de maneira satisfatória. Assim, a plataforma mostrou-se viável para a função que foi proposta e as informações adquiridas são confiáveis para o uso em diversas aplicações.

Vale ressaltar que a plataforma Arduino é utilizada com a finalidade do desenvolvimento de um protótipo, portanto, para projetos destinados à produção em larga escala, o ideal é que seja desenvolvido um *hardware* específico para a função desejada.

Conclui-se que o objetivo proposto na pesquisa pôde ser alcançado. O protótipo desenvolvido é um SAD confiável e eficiente, em que a leitura correta dos equipamentos foi possível devido ao desenvolvimento de módulos que tornam os sensores compatíveis com a plataforma Arduino, enquanto a IHM desenvolvida é intuitiva e de fácil manuseio.

Este estudo pode ser utilizado como base para pesquisas futuras em estudos em que a plataforma Arduino é empregada para diversos fins. Dessa forma, Como sugestões para pesquisas futuras, têm-se: implementar métodos que possibilitem a leitura em paralelo de comunicação



serial; desenvolver um aplicativo para dispositivos móveis (Android/IOS) para a leitura em tempo real das informações; desenvolver um *hardware* dedicado para o produto final, tornando, assim, a produção em larga escala viável

## REFERÊNCIAS

- [1] Arduino S. R. L. *What is Arduino?*, 2019. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acessado em agosto de 2021.
- [2] R. F. Garcia, R. C. de Lima e C. M. C. Manhães. Nota técnica: avaliação de um sensor de fluxo utilizando placa microcontroladora Arduino. *Revista Engenharia na Agricultura - REVENG*, 23(2):162–168, 2015. <https://doi.org/10.13083/reveng.v23i2.569>.
- [3] R. J. Halmeman. *Desenvolvimento de um sistema para monitoramento remoto em centrais de microgeração fotovoltaica*. Tese de Doutorado, Universidade Estadual Paulista Julio de Mesquita Filho - UNESP, Botucatu, SP, Brasil, 2014.
- [4] S. G. Hemeda, A. M. Aboukarima e M. Minyaw. Developing a logging unit for measuring and recording power data using Arduino board. *Misr Journal of Agricultural Engineering*, 34(4-2):2053–2072, 2017. <http://dx.doi.org/10.21608/mjae.2017.96360>.
- [5] D. A. Leite. Desempenho de um trator agrícola utilizando óleo de coco macaúba (*acrocymia aculeata*). In: *Seminário de Iniciação científica IFMG*, 2016.
- [6] R. V. d. O. Machado. *Sistema de aquisição de dados utilizando microcontrolador Arduino para a tecnologia de aplicação de agroquímicos*. Dissertação de Mestrado, Universidade Federal Rural do Semi-Árido, Mossoró, RN, Brasil, 2016.
- [7] National Marine Electronics Association. *NMEA 0183–Standard for interfacing marine electronic devices*. NMEA, 2002.
- [8] J. C. Negrete. SCADAS in the automation of agriculture in Mexico, a overview. *Acta Scientific Agriculture*, 2(7), 2018.
- [9] B. A. S. Oliveira, S. Assis e R. Sasaki. Aplicativo android para avaliar a qualidade da pulverização em condições de campo. *Revista Engenharia e Tecnologia Aplicada-UNG-Ser*, 2(1):15–23, 2019. <http://dx.doi.org/10.33947/2595-6264-v2n1-3617>.
- [10] B. A. S. Oliveira, M. S. Nametala, B. R. Castoril, M. D. Catalano e C. A. L. Nametala. Avaliação de uma rede neural artificial como estimador temporal pluviométrico no sistema de abastecimento Cantareira. *Revista de Informática Aplicada*, 14(1), 2018.
- [11] G. L. Oliveira, A. Russini, L. C. Radtke, C. A. Romani e L. D. de Nazaré Martins. Determinação do desempenho de tratores agrícolas. In: *Anais do Salão Internacional de Ensino, Pesquisa e Extensão*, 8(2), 2017.
- [12] S. Rasool, H. Raheman e G. Upadhyay. Development of an instrumentation system for evaluating the tractive performance of walking tractors. *International Journal of Current Microbiology and Applied Sciences*, 6(10):759–770, 2017. <https://doi.org/10.20546/ijemas.2017.610.092>.
- [13] D. C. Souza, A. Russini, L. D. de Nazaré Martins, C. Galafassi e R. R. de Vargas. Determinação do desempenho de tratores agrícolas utilizando instrumentação eletrônica com auxílio de plataforma aberta. In: *Anais do Salão Internacional de Ensino, Pesquisa e Extensão*, 9(3), 2018.
- [14] G. V. Veloso. *Automação do sistema de direção de colhedora de café*. Dissertação de Mestrado, Universidade Federal de Viçosa, Viçosa, MG, Brasil, 2013.
- [15] M. G. Zaragoza e H. K. Kim. Comparative study of PLC and Arduino in automated irrigation system. *International Journal of Control and Automation*, 10(6):207–218, 2016. <http://dx.doi.org/10.14257/ijca.2017.10.6.20>.