

UTILIZAÇÃO DO PROTOCOLO MQTT E FRAMEWORK ROS NA INSTRUMENTAÇÃO DE TRATORES AGRÍCOLAS

FÁBIO OLIVEIRA DE PAULA,*BRUNO HENRIQUE GROENNER BARBOSA,*SÉRGIO DE OLIVEIRA,†GUSTAVO HENRIQUE DE OLIVEIRA FONTES,*EDUARDO LÚCIO LASMAR JÚNIOR,*CARLOS EDUARDO SILVA VOLPATO*

**Universidade Federal de Lavras
Departamento de Engenharia
Lavras, Minas Gerais, Brasil*

*†Universidade Federal de São João del-Rei
Departamento de Tecnologia em Engenharia Civil, Computação e Humanidades
Ouro Branco, Minas Gerais, Brasil*

Emails: fabio.olp@gmail.com, brunohb@gmail.com, sergioool@ufs.j.edu.br,
guhof1822@gmail.com, lasmaredu@gmail.com, volpato@deg.ufla.br

Abstract— Considering the world population growth and the consequent increase of the food production, the optimization of agricultural process is crucial to circumvent this problem. For this, Precision Agriculture (AP) seeks to reduce the amount of nutrients and other inputs used, increasing the quality and productivity of crops without adding cultivated area. Through advances in electronic instrumentation and embedded systems, it has been possible to accelerate AP developments. Tools such as the Message Queuing Telemetry Transport (MQTT) protocol and the Robot Operating System (ROS) framework enable monitoring of system variables, communication between all units in the machine, communication with remote servers, and control of any actuator in that system. This work deals with the development of a modular embedded system for agricultural machines, with the ROS and the MQTT protocol, capable of collecting data during the operation of the tractor-implement assembly through specific sensors. It also addresses the validation of this system with a focus on the integrity of the data transmitted and their speed of transmission. With the tests carried out, it was observed that the system allows the acquisition of data by the ROS system at a sampling frequency of 3700 Hz which, compared to other works, is an excellent value for control systems. It was also observed that the system guarantees 100% integrity of the transmitted data.

Keywords— Precision agriculture. Agricultural Management. Embedded system. Electronic Instrumentation.

Resumo— Um grande motivador em se otimizar processos agrícolas é pela estimativa de crescimento da população mundial e o consequente aumento da produção de alimentos. Para isso, a Agricultura de Precisão (AP) busca reduzir a quantidade de nutrientes e outros insumos utilizados, aumentando a qualidade e a produtividade das culturas sem acréscimo de área cultivada. Por meio dos avanços da instrumentação eletrônica e de sistemas embarcados tem sido possível progredir de modo acelerado os desenvolvimentos da AP. Ferramentas como o protocolo de comunicação MQTT (*Message Queuing Telemetry Transport*) e o *framework Robot Operating System* (ROS) possibilitam o monitoramento das variáveis do sistema, a comunicação entre todas as unidades que houver na máquina e implementos, a comunicação com servidores remotos, e ainda o controle de qualquer atuador que houver nesse sistema. Esse trabalho trata do desenvolvimento de um sistema embarcado modular para máquinas agrícolas, com o ROS e o protocolo MQTT, capaz de coletar dados sobre o funcionamento do conjunto trator-implemento por meio de sensores específicos. Trata também da validação desse sistema com foco na integridade dos dados transmitidos e na velocidade de transmissão desses dados. Com os ensaios realizados, observou-se que o sistema permite a aquisição de dados pelo sistema ROS à uma frequência de amostragem de 3700 Hz que, em comparação com outros trabalhos, consiste de um valor excelente para sistemas de controle. Observou-se também que o sistema garante a integridade em 100% dos dados transmitidos.

Palavras-chave— Agricultura de Precisão. Gestão Agrícola. Sistema Embarcado. Instrumentação.

1 Introdução

A Agricultura de Precisão (AP) é um conceito moderno de gestão de sistemas de produção agrícola que utiliza técnicas digitais para monitorar e otimizar os processos de produção. Para isso, a AP busca reduzir a quantidade de nutrientes e outros insumos utilizados, aumentando a qualidade e a produtividade das culturas sem acréscimo de área cultivada. Sendo assim, a obtenção e processamento de informações detalhadas e georreferenciadas sobre as áreas de cultivo auxiliam na definição de estratégias de manejo mais eficientes (Union, 2016; Pierce and Nowak, 1999).

Com o contínuo avanço da eletrônica, o aumento da densidade de circuitos integrados e o

movimento crescente de produtos de prototipagem, os sistemas embarcados têm se tornado ferramentas cada vez mais poderosas e com maior capacidade de processamento. Com isso, ferramentas da Tecnologia da Informação e Comunicação (TIC), que antes eram restritas a centrais de grande processamento, hoje migram para sistemas menores. Atualmente, a TIC tem sido amplamente utilizada em todos os campos da agricultura e promovem o desenvolvimento da modernização agrícola (Talavera et al., 2017).

As tendências futuras das TICs, no contexto da agricultura de precisão, caminham para a integração de equipamentos agrícolas, sistemas de informação e automação de processos e operações agrícolas, computação ubíqua e em nuvem, apli-

cações geoespaciais, sistemas de suporte a decisão, protocolos e normas de produção. Esta é considerada uma importante fronteira do desenvolvimento da AP no mundo (Queiros et al., 2001).

Dentre as subáreas da TIC, destaca-se a *Internet of Things* (IoT) e seus protocolos de comunicação. Por ser um protocolo de mensagens leve, o MQTT (*Message Queuing Telemetry Transport*) é amplamente adotado e foi desenvolvido para dispositivos com restrições de energia, conectividade e memória (MQTT, 2017).

Em uma época em que a robótica cresce na área agrícola (de Sousa et al., 2014; Tabile et al., 2008), o ROS (*Robot Operating System*) também ganha importância e visibilidade nesse meio. ROS é um conjunto de bibliotecas e ferramentas que auxiliam na construção de aplicações para robôs. Assim, o ROS é capaz de integrar diversos sistemas, sensores e atuadores e tem maior aplicabilidade dentro da engenharia de controle e robótica, sobretudo na área industrial, como observa-se em Best et al. (2013), Cashmore et al. (2015) e Sa and Corke (2012a).

O ROS possibilita o monitoramento das variáveis do sistema, a comunicação entre todas as unidades que houver na máquina e implementos e o controle de qualquer atuador que houver nesse sistema. Desta forma, o ROS e o MQTT podem constituir assim, uma excelente ferramenta para a Agricultura de Precisão.

Este trabalho trata do desenvolvimento de um sistema embarcado modular para máquinas agrícolas, com o *framework* ROS e protocolo de comunicação MQTT, capaz de coletar dados sobre o funcionamento do conjunto trator-implemento. Tais dados podem ser utilizados posteriormente para implementação de leis de controle ou análise de dados sobre o desempenho do maquinário. Compõem esse sistema módulos com sensores específicos para cada parâmetro do maquinário agrícola e uma central que gerencia as conexões com os módulos. Trata também da validação desse sistema com foco na integridade dos dados transmitidos e na velocidade de transmissão.

De forma a obter redução dos custos, maior difusão e alcance entre produtores e desenvolvedores, priorizou-se utilizar apenas elementos de baixo custo no sistema proposto. Pretendeu-se que o sistema embarcado desenvolvido seja instalado e utilizado em máquinas agrícolas de baixo custo que não possuem quaisquer tipos de instrumentação ou central eletrônica. Assim, ao usufruir de tecnologia avançada em seu maquinário, o pequeno produtor tem a chance de tornar-se mais competitivo no mercado.

2 Referencial Teórico

2.1 Instrumentação Eletrônica na Agricultura de Precisão

A instrumentação, como ramo da engenharia que trata da medição, monitoramento e controle de variáveis, pode ser aplicada em sistemas de automação de máquinas e equipamentos agrícolas e de coleta de dados dos sensores do trator, do implemento (por exemplo, máquinas de preparo do solo, colheita, pulverizadores, etc), de sensores na lavoura, informações climatológicas, dentre outros (Pino, 2013; Seidel, 2011; Resende et al., 2010).

Segundo Resende et al. (2010), dentre as principais aplicações da AP no manejo de solo, destacam os sistemas de automação, sensoramento remoto, sistemas de guia, monitores de colheita, amostragem georreferenciada de solos, mapeamento da fertilidade e aplicação de corretivos e fertilizantes a taxa variável.

Dentre as importâncias da instrumentação eletrônica em máquinas agrícolas, Mantovani et al. (1999) afirmam que, com a instrumentação eletrônica, de custo razoável, montada no trator e implementos, pode-se ter a aquisição automática dos dados, que aumenta a facilidade na realização dos testes, e promove melhorias na obtenção de dados mais precisos medidos no campo. E dizem ainda que os parâmetros do conjunto trator-implemento que foram instrumentados permitiram descrever completamente o funcionamento do conjunto.

2.2 Sistemas Embarcados

São diversos os trabalhos de sistemas embarcados na área agrícola. Incluem dispositivos de coleta de informações do maquinário agrícola, como parâmetros referentes ao desempenho do maquinário; do meio de cultivo, como parâmetros referentes ao ar, ao solo, etc (Talavera et al., 2017); georreferenciamento com GPS/DGPS (Galante and Garcia, 2014; Keskina and Say, 2006); temperatura e umidade ambiente (Zhang, Ma and Yang, 2016); intensidade da luz, dentre outros; além de sistemas de gerenciamento de dados (Du et al., 2016; Zhu et al., 2016; Zhang, Hu, Huang, Zhang and Ma, 2016; Maia et al., 2016) e comunicação (Ibayashi et al., 2016; Julia et al., 2014).

Segundo Maia et al. (2016), é cada vez mais comum a utilização de Unidades de Controle Eletrônico (do inglês, *Electronic Control Unit* - ECU) no setor da agricultura, mais especificamente nos veículos e implementos, acompanhando uma tendência global. Isso contribui para a produtividade, melhora o conforto do operador e torna os veículos agrícolas mais seguros.

Dentre as opções de placas de desenvolvimento no mercado brasileiro, pode-se destacar o Raspberry Pi, por ser um computador em módulo

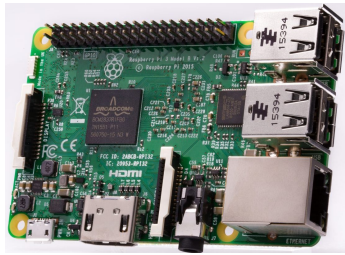


Figura 1: Raspberry Pi 3
Fonte: (RaspberryPi, 2017)

com grande potencial de aplicações, e a placa NodeMCU, por seu *hardware* versátil e facilidade de programação.

2.2.1 Raspberry Pi

O Raspberry Pi (Figura 1) é um computador de placa única desenvolvido no Reino Unido pela empresa *Raspberry Pi Foundation* (Cellan-Jones, 2011). O modelo e a ideia originais tornaram-se muito populares e chegou à aplicações complexas, como a robótica.

O modelo Raspberry Pi 3 modelo B é composto por um microprocessador *Broadcom BCM2837 quad-core* de 64 bits, 1,2GHz de *clock* e 1Gb de memória RAM. Possui módulos *Wi-Fi* e *Bluetooth* embutidos e *slot* de cartão de memória SD para armazenamento do sistema operacional (RaspberryPi, 2017).

2.2.2 NodeMCU

NodeMCU é uma plataforma de desenvolvimento *open-source* que tem sido muito utilizada em projetos de IoT devido à sua interface *Wi-Fi* (Figura 2). Nativamente, ela utiliza a linguagem de *script* Lua, mas permite a programação por meio da IDE do Arduino. Possui sistema operacional *XTOS*, memória de 128kB e armazenamento de 4MB. A CPU é um chip *ESP8266* (NodeMCU, 2017).

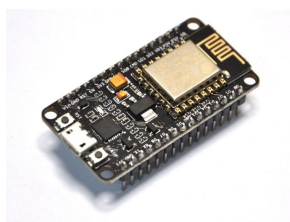


Figura 2: Módulo NodeMCU
Fonte: (NodeMCU, 2017)

2.3 Protocolo de Comunicação MQTT

O protocolo de mensagens MQTT (*Message Queue Telemetry Transport*) é projetado para um baixo

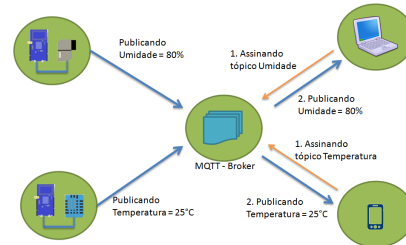


Figura 3: Comunicação MQTT
Fonte: (AmebaIoT, 2017)

consumo de banda de rede e requisitos de *hardware* sendo extremamente simples e leve (MQTT, 2017).

O protocolo utiliza o paradigma de comunicação *publish/subscribe* para a troca de mensagens. Esse paradigma utiliza um dispositivo central, chamado de *broker*, que intermedia toda a comunicação. O *broker* é responsável por receber, enfileirar e reenviar as mensagens recebidas dos *publishers* para os *subscribers*. O *publisher* é responsável por se conectar ao *broker* e publicar mensagens. O *subscriber* é responsável por se conectar ao *broker* e receber as mensagens que ele tiver interesse. A Figura 3 apresenta um esquemático da dinâmica de comunicação do protocolo MQTT.

Para coletar dados sobre o campo para análise contínua e para o desenvolvimento e implantação de tecnologia inteligente para o setor agrícola buscando a melhoria da sustentabilidade ambiental e agrícola, e melhoraria na rastreabilidade das culturas, Pooja et al. (2017) propõem um sistema baseado em IoT que utiliza o MQTT como protocolo de comunicação. Pooja et al. (2017) consideraram o MQTT como um protocolo leve e muito flexível para o uso no controle e monitoramento de parâmetros ambientais. Propuseram que para um produto final, os dados ambientais fossem apresentados ao usuário de uma maneira mais amigável por meio de gráficos interativos.

2.4 Desempenho de rede

A norma RFC 2544, de abrangência internacional, propõe e define os testes utilizados para descrever e comparar as características de desempenho dos dispositivos conectados à uma rede. Para a avaliação de dispositivos de comunicação, a norma propõe testes de desempenho que visam quantificar o desempenho do equipamento em diferentes cenários de utilização (Bradner and McQuaid, 1999).

Dentre esses testes destaca-se o teste de latência, que diz respeito ao tempo total que o último bit de um pacote de dados demora para ir da origem ao destino.

Uma das formas de realizar o cálculo de latência, segundo a norma, é fazer com que um

pacote de dados de teste seja enviado pelo dispositivo de origem, recebido pelo dispositivo de destino e reenviado, por este último, ao dispositivo de origem. A latência é dada pela metade do tempo decorrido para o pacote de dados sair e voltar ao dispositivo de origem. Segundo Bradner and McQuaid (1999), a unidade que os valores de tempo de latência devem ser apresentados é microsegundos (μs).

Outro teste importante descrito pela norma RFC 2544 é o teste de perda de pacotes de dados. O valor encontrado nesse teste é definido como o número de pacotes de dados que foram transmitidos pela origem e não foram recebidos no destino (Bradner and McQuaid, 1999).

2.5 Robot Operating System (ROS)

Robot Operating System (ROS) é um *framework*, isto é, um conjunto de bibliotecas e ferramentas *open-source* que auxiliam na construção de aplicações para robôs. É uma coleção de algoritmos e convenções que visam simplificar a tarefa de criar um comportamento complexo e robusto de robôs através de uma ampla variedade de plataformas robóticas. Suas aplicações vão de *drivers* até algoritmos e ferramentas de desenvolvimento para projetos de robótica complexos (ROS, 2017).

Com o mesmo paradigma de comunicação do protocolo MQTT, o ROS utiliza um dispositivo intermediário denominado de *master* e clientes denominados *nós*. As variáveis onde são disponibilizados os dados são chamados de *tópicos*. Os nós publicam e assinam mensagens através dos tópicos.

A principal vantagem da utilização do ROS é a facilidade gerada no desenvolvimento de *softwares* para sistemas robóticos, que viabiliza a reutilização de código e oferece uma interface modular e distribuída (ROS, 2017).

De acordo com o conceito de modularidade do sistema ROS, apresentado por Quigley et al. (2009), o ROS revela-se uma poderosa ferramenta para o desenvolvimento de sistemas embarcados.

Best et al. (2013) realizaram a classificação da cobertura do solo para favorecer o deslocamento de robôs móveis com a utilização de *Support Vector Machine* e com o algoritmo implementado no ROS. Para o sucesso de seu desenvolvimento, foi utilizada a taxa de amostragem das variáveis de entrada à 20Hz. Já Sa and Corke (2012b), para controle de um *quadcopter*, utilizaram a taxa de amostragem de 10Hz.

3 MATERIAIS E MÉTODOS

O projeto consistiu na implementação da infraestrutura de uma rede sem fio embarcada no maquinário agrícola dotada de uma central e módulos

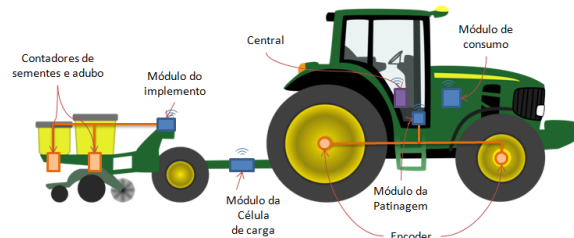


Figura 4: Diagrama da rede e elementos

periféricos. Os elementos desse sistema são apresentados na Figura 4 e descritos na subseção 3.1.

3.1 Arquitetura proposta

A central é também conhecida na infraestrutura da rede como *gateway*. Ela proveu a conexão *Wi-Fi* e atuou como *broker* para o protocolo MQTT e como *master* para o ROS. Foi constituída pelo Raspberry Pi 3, por ser um computador de placa única que permite utilizar o Linux embarcado. Ela gerencia a conexão com os módulos periféricos, centraliza todas as informações do sistema e registra os dados em um banco de dados local, em um cartão de memória.

Os módulos periféricos foram desenvolvidos com base no módulo NodeMCU, devido sua interface *Wi-Fi* e apresentam *hardware* simples e limitados computacionalmente. Realizaram a instrumentação de parâmetros específicos do maquinário, como exemplo, a leitura da patinagem, da força exercida na célula de carga, do consumo, dentre outros, e atuaram como clientes na rede.

Com os dispositivos conectados nessa rede, os módulos periféricos se comunicam com a central por meio do protocolo MQTT. A central, ao receber dados por MQTT, converte-os ao protocolo utilizado pelo ROS.

Essa dinâmica de comunicação foi escolhida devido o fato que sistemas microcontrolados podem ser programados como clientes MQTT, mas não suportam o ROS. Isso permite usufruir do ROS em sistemas microcontrolados de baixo custo.

Outro requisito importante desse projeto é não tornar o sistema proprietário, para que haja maior aceitação na comunidade acadêmica e no mercado. Com base nisso, foi adotado o protocolo MQTT por ser bem difundido na comunidade de desenvolvimento.

A central foi instalada na cabine do trator, atrás do acento do condutor. O módulo periférico, alimentado por uma bateria alcalina, foi instalado no implemento, à cerca de 2 metros da central.

3.2 Metodologia experimental

Os ensaios realizados neste trabalho se baseiam nos testes descritos na norma RFC 2544, com foco

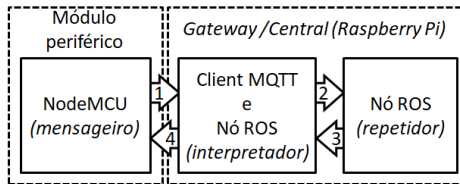


Figura 5: Diagrama da comunicação dos dados

no teste de latência e no teste de perda de dados.

Para ambos ensaios, o caminho percorrido pelo pacote de dados é representado pelo diagrama da Figura 5, seguindo a numeração das setas, e conta com a utilização de três algoritmos.

O primeiro algoritmo, executado em uma placa NodeMCU e denominado *mensageiro*, publica uma mensagem em um determinado tópico MQTT e aguarda essa mensagem retornar em um outro tópico MQTT subscrito. O segundo, denominado *interpretador*, subscrive todos os tópicos MQTT e os publicam em tópicos homônimos em ROS, sendo assim, esse algoritmo é também um nó ROS. Subscrive ainda um determinado tópico ROS e, quando recebe alguma mensagem por ele, publica-a em um tópico MQTT homônimo. Em outras palavras, qualquer mensagem recebida por esse algoritmo em MQTT é convertida para o ROS, e vice-versa. O terceiro algoritmo, denominado *repetidor*, é um nó ROS. Ele subscrive um determinado tópico ROS e, quando recebe alguma mensagem, publica-a em outro tópico ROS.

3.2.1 Teste de latência

Utilizando os algoritmos descritos anteriormente, o teste de latência inicia-se com o *mensageiro* que envia uma mensagem por MQTT e inicia um contador de tempo. Essa mensagem é recebida pelo *interpretador* e convertida para ROS. O *repetidor* a recebe pelo ROS e envia de volta ao *interpretador*. Este a converte do ROS para MQTT e envia ao *mensageiro*. Este último, assim que a recebe, encerra o contador. Como a norma especifica a latência como sendo o tempo decorrido para o pacote de dados ir do dispositivo de origem ao dispositivo de destino, o valor de tempo pelo contador obtido foi dividido por dois (ida e volta) para que se obtenha o tempo decorrido apenas durante a ida dos dados.

Para maior confiabilidade, esse procedimento foi repetido 1000 vezes e calculou-se dessas medições a média aritmética e seu erro padrão, o desvio padrão, a variância e o coeficiente de variação.

3.2.2 Teste de perda de dados

Adotando os mesmos algoritmos já descritos, o teste de perda de dados tem a mesma dinâmica

Média aritmética	270,001	μs
Desvio Padrão	33,686	μs
Coeficiente de Variação	12,476	%
Erro padrão da média	1,065	μs

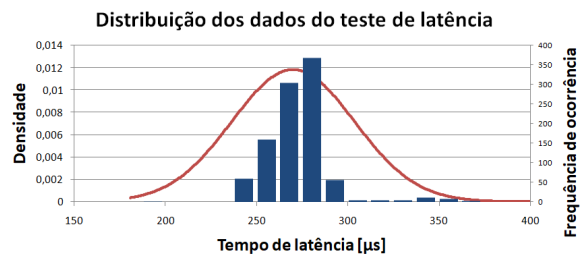


Figura 6: Distribuição dos dados do teste de latência

do teste de latência. A diferença é que as mensagens enviadas pelo *mensageiro* eram variáveis do tipo inteiro com seus valores incrementados à cada repetição. Com isso, a primeira mensagem enviada assumia o valor "1", a segunda assumia o valor "2" e assim por diante até a última que assumia o valor "1000".

Os dados enviados e recebidos pelo *mensageiro* foram tratados para observar se houveram discrepâncias entre eles.

4 Resultados e Discussões

Para o teste de latência, obteve-se dos valores medidos os parâmetros estatísticos indicados na Tabela 1. Assim, o valor médio encontrado corresponde ao tempo médio decorrido entre o envio da mensagem pelo módulo periférico e o recebimento em um nó ROS na central, e seu respectivo erro padrão; o desvio padrão e o coeficiente de variação correspondem à dispersão desses dados.

A Figura 6 apresenta a distribuição dos valores de latência por meio do histograma e densidade de probabilidade. O histograma apresenta a frequência de ocorrência dos valores de latência distribuídos em \sqrt{n} classes, onde n é "1000" referente à quantidade de amostras. A curva representa a função densidade de probabilidade (FDP) da distribuição.

A média encontrada com valor de 270 μs corresponde à uma frequência de amostragem de 3700 Hz. Essa frequência de amostragem atende à sistemas de controle, como Best et al. (2013), que utiliza a frequência de 20 Hz, e Sa and Corke (2012b), que utiliza a frequência de 10 Hz. Nota-se então, que a arquitetura de rede do sistema de instrumentação proposto atende à sistemas de controle que utilizam o *framework* ROS.

O fato do sistema de aquisição e comunicação proposto apresentar um tempo de resposta muito mais rápido quando comparado ao tempo

de resposta dos sistemas de controle dos trabalhos citados, por exemplo, não prejudica o algoritmo de controle. Pelo contrário, os algoritmos de controle poderão ser projetados livremente sem que o tempo de comunicação seja um gargalo.

Embora a norma RFC-2544 especifique tamanhos padrões de pacotes de dados, onde o pacote mínimo seria de 512 bits, a quantidade de dados transmitidos por pacote no sistema proposto foi de 32 bits. Esse tamanho de mensagem utilizado corresponde à 4 bytes, que é o tamanho de uma variável do tipo *float*, muito utilizado em instrumentação e metrologia.

Quanto ao teste de perda de dados, em nenhuma das transmissões observou-se discrepância entre a mensagem enviada e a mensagem recebida. Constituindo assim, 100% de integridade dos dados transmitidos.

5 Conclusões

Com foco na temática da Agricultura de Precisão, apresentou-se a infraestrutura de uma rede sem fio e um sistema embarcado modular para realizar a instrumentação de máquinas agrícolas.

Para a implementação de leis de controle ou análise de dados sobre o desempenho do maquinário agrícola é imprescindível a disponibilidade de dados bons e confiáveis.

Para isso, contou-se com a utilização do protocolo de comunicação MQTT realizando a comunicação entre módulos periféricos, onde estão os sensores, e a central do sistema, onde está instalado o ROS como *software* de análise de dados.

Pretendeu-se com esse sistema criar uma alternativa a ser instalada no maquinário agrícola que não possui nenhum sistema embarcado, por ser um sistema mais acessível, de aplicação ampla e de menor custo de aquisição.

Os resultados dos testes de latência e de perda de dados realizados atestam que o sistema atende à proposta e garante plena integridade dos dados e excelente velocidade de comunicação.

Agradecimentos

Agradecemos à FAPEMIG e à UFSJ/DETEM pelo apoio prestado neste projeto.

Referências

- AmebaIoT (2017). Ameba arduino: Use mqtt to upload and listen to data, *Technical report*, Ameba IoT, Estados Unidos.
- Best, G., Moghadam, P., Kottege, N. and Klee-man, L. (2013). Terrain classification using a hexapod robot, *Proceedings of Australasian Conference on Robotics and Automation*, Sydney, Australia.
- Bradner, S. and McQuaid, J. (1999). *Request for Comments: 2544 - benchmarking methodology for network interconnect devices*.
- Cashmore, M., Fox, M., Long, D., Magazzeni, D., Ridder, B., Carrera, A., Palomerias, N., Hurtos, N. and Carreras, M. (2015). Rosplan: Planning in the robot operating system, *International Conference on Automated Planning and Scheduling*.
- Cellan-Jones, R. (2011). A 15 pound computer to inspire young programmers, Web page. BBC.
- de Sousa, R. V., Lopes, W. C., Tabile, R. A., Porto, A. J. V. and Inamasu, R. Y. (2014). Modelo conceitual para sistema de coleta e comunicação automática de dados para máquina e implemento automatizado e para robô agrícola móvel, *Agricultura de precisão: resultados de um novo olhar*, Bernardi, A. C. de C. and Naime, J. de M. and Resende, A. V. de and Bassoi, L. H. and Inamasu, R. Y., Brasília, DF, pp. 165–172.
- Du, K., Chu, J., Sun, Z., Zheng, F., Xia, Y. and Yang, X. (2016). Design and implementation of monitoring system for agricultural environment based on webgis with internet of things, *Transactions of the Chinese Society of Agricultural Engineering* **32**(4): 171–178.
- Galante, A. and Garcia, R. (2014). Sistema de aquisição de dados de sensores de baixo custo baseado no arduino, *Proceedings of the Congresso Brasileiro de Agricultura de Precisão (ConBAP)*, São Pedro.
- Ibayashi, H., Kaneda, Y., Imahara, J., Oishi, N., Kuroda, M. and Mineno, H. (2016). A reliable wireless control system for tomato hydroponics, *Sensors* **16**(5): 644.
- Jula, A., Sundararajan, E. and Othman, Z. (2014). Cloud computing service composition: A systematic literature review, *Expert Systems with Applications* **41**(8): 3809–3824.
- Keskina, M. and Say, S. M. (2006). Feasibility of low-cost gps receivers for ground speed measurement, *Computers and Electronics in Agriculture* **54**: 36–43.
- Maia, C. M., Inamasu, R. Y. and Poto, A. J. V. (2016). Análise e desenvolvimento de emulador de T-ECU para testes e aplicações didáticas em redes ISO 11783-compatíveis, *Simpósio do Programa de Pós-Graduação em Engenharia Mecânica*.

- Mantovani, E. C., Leplatois, M. and Inamassu, R. Y. (1999). Automação do processo de avaliação de desempenho de tratores e implementos em campo, *Pesquisa Agropecuária Brasileira* **34**: 1241 – 1246.
- MQTT (2017). MQTT community wiki, *Technical report*, MQTT.org.
- NodeMCU (2017). *NodeMCU: Connect Things easy*.
- Pierce, F. and Nowak, P. (1999). Aspects of precision agriculture, *Adv. Agronomy* **67**: 1–85.
- Pino, A. V. (2013). Instrumentação e técnicas de medidas, Apostila da UFRJ.
- Pooja, S., Uday, D. V., Nagesh, U. B. and Talekar, S. G. (2017). Application of mqtt protocol for real time weather monitoring and precision farming, *International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECOT)*, pp. 1–6.
- Queiros, L. R., Luchiari Junior, A., Camargo Neto, J., Massruhá, S. M. F. S., Inamasu, R. Y., Speranza, E. A. and Evangelista, S. R. M. (2001). Análise das possibilidades e tendências do uso da tecnologia da informação e comunicação na agricultura de precisão, *Agricultura de precisão: resultados de um novo olhar*, number 01/2001, Embrapa Informática Agropecuária, Brasília, DF, pp. 97–108.
- Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R. and Ng, A. (2009). Ros: an open-source robot operating system, *Proceedings of the ICRA Workshop on Open Source Software*.
- RaspberryPi (2017). Raspberry pi 3 model b.
- Resende, A. V., Shiratsuchi, L. S., Coelho, A. M., Corazza, E. J., Vilela, M. F., Inamasu, R. Y., Bernardi, A. C. C., Bassoi, L. H. and Naime, J. a. M. (2010). Agricultura de precisão no brasil: Avanços, dificuldades e impactos no manejo e conservação do solo, segurança alimentar e sustentabilidade, *Proceedings of the XVIII Reunião Brasileira de Manejo e Conservação do Solo e da Água*.
- ROS (2017). Robotic operating system, *Technical report*, Open Source Robotics Foundation, Stanford, USA.
- Sa, I. and Corke, P. (2012a). System identification, estimation and control for a cost effective open-source quadcopter, *IEEE International Conference on Robotics and Automation*, pp. 2202–2209.
- Sa, I. and Corke, P. (2012b). System identification, estimation and control for a cost effective open-source quadcopter, *IEEE International Conference on Robotics and Automation* pp. 14–18.
- Seidel, A. R. (2011). *Instrumentação Aplicada*, Editora UFSM, Brasil.
- Tabile, R. A., Inamasu, R. Y. and Porto, A. J. V. (2008). Robótica na agricultura de precisão, *Agricultura de Precisão: um novo olhar*, Embrapa.
- Talavera, J. M., Tobón, L. E., Gómez, J. A., Culman, M. A., Aranda, J. M., Parra, D. T., Quiroz, L. A., Hoyos, A. and Garreta, L. E. (2017). Review of iot applications in agro-industrial and environmental fields, *Computers and Electronics in Agriculture* **142**: 283–297.
- Union, E. (2016). Precision agriculture and the future of farming in europe, *Technical report*, European Union.
- Zhang, J., Hu, J., Huang, L., Zhang, Z. and Ma, Y. (2016). A portable farmland information collection system with multiple sensors, *Sensors* **16**(10).
- Zhang, W., Ma, H. and Yang, S. X. (2016). An inexpensiveinexpensive, stable, and accurate relative humidity measurement method for challenging environments, *Sensors* **16**(3).
- Zhu, W., Chen, W., Hu, Z., Li, Z., Liang, Y. and Chen, J. (2016). Data collection for mobile group consumption: An asynchronous distributed approach, *Sensors* **16**(4).