

# UMA PLATAFORMA ABERTA E DE BAIXO CUSTO DE ROBÔ MÓVEL PARA PROPÓSITO GERAL

MATSUMURA, Takao

Universidade Estadual Paulista Júlio de Mesquita Filho (Unesp)  
matsumura.tko@gmail.com

FERASOLI FILHO, Humberto

Universidade Estadual Paulista Júlio de Mesquita Filho (Unesp)  
Bauru - SP  
ferasoli@fc.unesp.br

MARRANGHELLO, Norian

Universidade Estadual Paulista Júlio de Mesquita Filho (Unesp)  
São José do Rio Preto - SP  
norian@ibilce.unesp.br

**RESUMO:** Pesquisas na robótica móvel tem sido impulsionadas pelos avanços tecnológicos. Existem frentes de pesquisas que abordam diferentes aspectos e desafios da robótica móvel, dentre os quais é possível citar tópicos como locomoção, navegação e arquitetura de controle. Esse crescimento em pesquisas acarreta uma maior necessidade por plataformas de robôs que possam ser destinadas à pesquisa e também para fins educacionais. Este trabalho propõe uma alternativa de plataforma de robô móvel de baixo custo, de arquitetura de hardware, software e controle aberta, destinada a atividades de propósitos gerais. A plataforma almeja a facilitação e flexibilização do processo de desenvolvimento de estudos e aplicações robóticas por meio de uma interface de comunicação simplificada e pela abstração da heterogeneidade dos dispositivos periféricos de hardware, sendo assim, capaz de oferecer maior liberdade em relação às linguagens de programação, paradigmas de controle e tecnologias de controle.

**Palavras-chave:** plataforma de robô móvel, arquitetura aberta, interface de comunicação.

**ABSTRACT:** *The research on mobile robots has been driven by technological advances. There are research fronts on different aspects and challenges of mobile robots, among which are topics such as locomotion, navigation and control architectures. This growth in researches also leads to greater need for mobile robot platforms that can be used for research and educational purposes. This work presents an alternative of low cost mobile robot platform with an open hardware, software and control architecture, for use in general purposes activities. This platform aims to ease and turn more flexible the development process of research and application on robotics through a simplified communication interface and abstracting the heterogeneity of the peripheral hardware devices, being able to offer greater freedom in relation to programming languages, control paradigms and control technologies.*

**Keywords:** *mobile robot platform, open architecture, communication interface.*

## 1. INTRODUÇÃO

O campo da robótica móvel vem sendo cada vez mais explorado. É uma realidade que acompanha o progresso tecnológico. Hoje em dia, robôs móveis podem ser frequentemente vistos em diversas áreas além dos tradicionais setores industriais, como já podem ser associados a aplicações agrícolas, hospitalares e domésticas. Essa popularização da robótica móvel tem esti-

mulado pesquisas nessa área, que envolve outras áreas como mecânica, matemática, física, eletrônica, computação e biologia. E devido à sua natureza multidisciplinar, a robótica móvel pode ser abordada a partir de diferentes perspectivas.

Existem frentes de pesquisas que abordam diferentes aspectos e desafios da robótica móvel, dentre os quais estão tópicos como locomoção, navegação e arquitetura de controle. O crescimento em pesquisas

também acarreta uma maior necessidade por bases móveis, em outras palavras, por plataformas de robôs móveis que possam ser destinadas às iniciativas que envolvam a robótica móvel. Embora existam soluções comerciais para essas plataformas de robôs móveis, ainda existem sérias limitações em relação à flexibilidade tanto em termos de *hardware* utilizado, como sensores e atuadores, assim como em relação aos *software*, que costumam ser proprietários e fechados. Essas limitações se estendem em relação às necessidades de alterações, que são comuns em termos de *hardware*, fatores que acabam influenciando de forma decisiva sobre a empregabilidade de tais soluções tanto em aplicações gerais, como no âmbito científico e didático.

Existem outras iniciativas na literatura que propuseram ambientes e arquiteturas voltadas para o desenvolvimento de robôs móveis, que objetivaram a facilitação e a simplificação do processo de desenvolvimento de robôs móveis. É o caso de trabalhos como os propostos por [1,3,4], que propõem ferramentas e ambientes de programação robóticos, com bibliotecas de código aberto concebidas em diversas linguagens de programação.

Desta forma, uma plataforma de robô móvel de arquitetura de controle, de *software* e de *hardware*, aberta, pode apresentar-se como uma alternativa diante de tal cenário. Adicionalmente, atributos como simplicidade e manobrabilidade, associados à facilidade de alterações físicas (sensores e atuadores) podem contribuir de forma decisiva na aplicabilidade de uma plataforma de robô móvel. Ao buscar tais características, uma plataforma de robô móvel pode apresentar-se como uma alternativa de ferramenta interessante no desenvolvimento de aplicações envolvendo a robótica móvel. E pode, ainda, ser empregada como ferramenta de ensino e base de investigações de assuntos relacionados à robótica móvel.

Este trabalho apresenta a implementação de uma plataforma de robô móvel aberta e de baixo custo, de arquitetura de *hardware* e *software* aberta. Esta plataforma oferece suporte a diferentes cenários e tecnologias de sistemas de controle através de uma interface padronizada de comunica-

ção. Além disso, esta plataforma implementa conceitos simples e bem estabelecidos, como arquitetura cliente-servidor, sistema fortemente acoplado e interface de comunicação serial. Em virtude da simplicidade conceitual envolvida, esta plataforma pode ser facilmente empregada em aplicações gerais envolvendo a robótica móvel. Com isso, esta plataforma se apresenta como uma alternativa viável para a facilitação do processo de desenvolvimento de *software* de controle. Além de seu baixo custo relativo, a plataforma apresentada motiva o reaproveitamento de seus módulos de modo independente ao esquema de controle ou à tecnologia de controle adotada.

De modo a validar esta plataforma, dois experimentos foram elaborados e realizados para demonstrar que a arquitetura apresentada oferece flexibilidade em relação à plataforma de controle. No primeiro experimento, foi elaborado um cenário investigativo envolvendo a navegação de robôs móveis. Neste primeiro cenário, utilizou-se a plataforma controlando-a de forma remota através de um *software* de navegação sendo executado em um computador pessoal. Já no segundo experimento, foi avaliada a flexibilidade da plataforma em termos de tecnologia de sistemas de controle e sua capacidade de adaptação ao acrescentar dispositivos de *hardware*. Para o sistema de controle da plataforma (servidor), os *software* das aplicações e suas bibliotecas foram implementadas em C++, para ambientes Windows e, então, adaptadas para o Arduino.

## 2. ARQUITETURA DA PLATAFORMA DE ROBÔ MÓVEL

Considerando a destinação de propósito geral da plataforma de robô móvel apresentada, é apropriada a adoção de uma arquitetura que ofereça facilidade de entendimento e ainda permita um grau de flexibilidade de *hardware* para ser modificada de acordo com as necessidades das aplicações. Desta forma, optou-se por adotar uma arquitetura preferencialmente simples, tanto em termos conceituais como técnico. Isso permite fácil compreensão e praticidade em termos de implementação,

tornando o uso da plataforma interessante e motivador. Assim sendo, a disposição da arquitetura da plataforma de robô móvel e seus componentes foram divididos em três camadas, sendo estas:

- Camada Superior: Camada na qual é implementada o *software* de controle, responsável pelo processamento dos dados e leituras dos sensores, tomada de decisões e ainda pelo controle de acionamento dos atuadores;
- Camada Intermediária: Camada responsável por estabelecer a comunicação entre a Camada Superior e a Camada Inferior. Esta camada abrange a interface comum de *hardware*, o protocolo de comunicação padronizado e APIs e bibliotecas de interface, responsáveis pela abstração dos detalhes específicos de todos os dispositivos periféricos localizados na Camada Inferior;
- Camada Inferior: Corresponde aos dispositivos periféricos de *hardware* implementados e funcionais, abrangendo tanto sensores como atuadores.

A Figura 1 ilustra a disposição da arquitetura da plataforma e suas camadas.

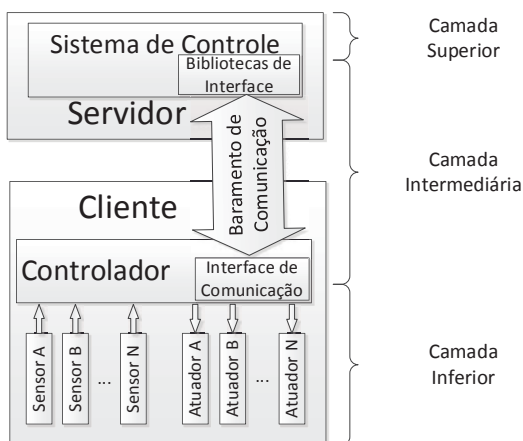


Figura 1 - Disposição da Arquitetura.

Esta plataforma adota uma arquitetura de *software* do tipo cliente-servidor

onde o Cliente representa a base do robô móvel e o Servidor corresponde ao sistema disponível para o controle. O Servidor pode tanto ser embarcado, com a conexão sendo estabelecida diretamente no barramento de comunicação, ou remoto, sendo a comunicação estabelecida com a utilização de tecnologias de conexão sem fio, como RF, Wi-Fi ou Bluetooth.

Nesta arquitetura, o Servidor é composto pelo Sistema de Software de suporte para a Arquitetura de Controle, APIs e bibliotecas responsáveis pela interface e abstração da comunicação estabelecida com o Cliente. A Camada Superior compreende tecnologias computacionais como computadores pessoais, computadores portáteis ou dispositivos embarcáveis de processamento, como microcontroladores, FPGA, Arduino, Raspberry Pi, dentre outros. Já o Cliente corresponde à base de robô móvel, onde se encontra o Controlador e os módulos de hardware periféricos.

Para o Controlador, foi adotado o conceito de sistema fortemente acoplado, atribuindo a um único agente a tarefa de gerenciar todos os dispositivos de sensoriamento e atuação. Isso confere à plataforma menor complexidade em comparação à abordagem conceitual que um sistema fracamente acoplado acarreta. Um outro fator que levou à essa escolha foi a necessidade de um número reduzido de dispositivos dedicados de gerenciamento dos dispositivos periféricos, e portanto, menor custo de implementação e manutenção. Além disso, com esta abordagem, é possível mitigar erros e falhas relacionadas ao desenvolvimento, implementação e alterações dos módulos localizados na Camada Inferior.

Ao deixar em aberto questões como arquitetura de controle e arquitetura de software, esta plataforma oferece neutralidade em relação à implementação de qualquer paradigma de controle apresentados por [5,7], implicando também em uma neutralidade em relação à linguagem de programação utilizada. Essa abordagem oferece ao usuário maior liberdade na escolha da linguagem de programação, que melhor atenda às necessidades da aplicação ou simplesmente em razão da familiaridade com determinada linguagem ou ambiente

de desenvolvimento. E essa liberdade, por sua vez, se estende também à tecnologia de controle empregada.

Para a comunicação de dados, foi adotado o serial, compatível e com suporte ao padrão RS-232, embora existam outros padrões seriais de comunicação, como o padrão USB. De acordo com [8], o RS-232 é uma interface cuja conectividade pode ser direcionada em aplicações que exijam pouca ou média largura de banda. E, apesar de se tratar de um padrão legatário, o serial apresenta aspectos favoráveis em relação ao tipo paralelo e, até mesmo, em relação a padrões mais modernos, como o USB. O principal aspecto considerado para tal escolha foi sua simplicidade, uma vez que o porto serial apresenta-se como uma alternativa de fácil entendimento, implementação e uso. Padrões mais modernos, como o USB, por possuírem diversos modos de operação, conectividades e recursos, como o *Plug-and-Play*, apresentam extensas documentações e acabam exigindo conhecimentos mais avançados. Isso pode, num primeiro instante, desencorajar sua utilização. Além disso, o padrão serial ainda pode ser encontrado em dispositivos embarcáveis, como FPGAs e Raspberry Pi. Já o USART é um padrão de comunicação de dados serial ponto a ponto, sendo comumente empregado em conjunto com o padrão RS-232, RS-485, dentre outros. A combinação da simplicidade oferecida pelo padrão USART em conjunto com o padrão RS-232 é, ainda, interessante e conveniente tanto em termos didáticos como técnicos, o que permite fácil implementação e utilização por parte dos usuários. Com isso, é possível concentrar maior parte dos esforços nas questões relacionadas à robótica ou ao tópico de interesse e sem desestimular o usuário, desde o estudante até um pesquisador. E de forma a oferecer maior flexibilidade em relação à conectividade, uma linha adicional de comunicação direta oferece ao usuário da plataforma de robô móvel, acesso ao terminal EUSART do microcontrolador utilizado, dando maior liberdade de acoplar diferentes esquemas físicos de plataformas de controle ou dispositivos de comunicação sem fio (rádio, Wi-Fi ou bluetooth).

### 3. A PLATAFORMA DE ROBÔ MÓVEL

O processo de desenvolvimento do corpo de um robô móvel deve ser feito buscando um formato que melhor se adapte aos requisitos da aplicação. No entanto, este processo depende também de outros aspectos como limitações tecnológicas e econômicas. Desta forma, o custo total de aquisição dos materiais utilizados é um fator fundamental, principalmente ao se propor uma alternativa de baixo custo. Para este trabalho, o custo total dos componentes e materiais ficou em torno de R\$500,00. Neste cálculo foram desconsiderados os custos do seu desenvolvimento e montagem, já que serve apenas para elucidar o baixo custo relativo do robô. Além disso, o valor do custo real depende de variáveis de mercado que fogem do escopo deste trabalho.

Para permitir que a plataforma possa ser destinada às aplicações gerais, é desejável que o hardware do robô móvel seja reutilizável e, ainda, permita modificações de modo rápido e descomplicado. Esta possibilidade de reutilização pode acelerar o desenvolvimento de aplicações robóticas além de reduzir substancialmente a ocorrência de erros. No entanto, a reutilização dos módulos de hardware em sistemas robóticos, assim como a possibilidade de adicionar componentes, continua ainda muito limitada. Como as arquiteturas dos módulos de hardware robóticos costumam ser extremamente especializadas e voltadas às aplicações finais, isto acaba dificultando ainda mais o processo de adição ou alteração dos módulos de hardware. Esse aspecto acaba se tornando um grande obstáculo, principalmente em iniciativas didáticas e científicas envolvendo a robótica móvel. A solução de utilizar uma arquitetura modular de hardware, em outras palavras, subsistemas, ainda que seja uma abordagem válida, não se mostra suficientemente flexível e, ainda, suporta uma pequena quantidade de funcionalidades [6].

A Figura 2 ilustra a base de robô móvel concebida, denominado Kihon.



Figura 2 - Robô Móvel Kihon.

Em relação ao aspecto físico e mecânico da base do robô móvel Kihon apresentado, foi considerado que seu ambiente de aplicação se limita às aplicações em superfícies planas com poucas irregularidades. Assim, para a locomoção do Kihon, dentre outras alternativas possíveis, adotou-se a locomoção baseada em rodas, uma vez que são as mais comumente encontradas na robótica móvel. Uma série de fatores justificam tal escolha, dentre os quais é possível citar o baixo custo, seja na implementação como manutenção, estabilidade e também pela simplicidade, tanto mecânica como de controle. As rodas apresentam elevado grau de eficiência em aplicações terrestres, especificamente em superfícies planas. E, ainda, a alternativa adotada se adequa ao ambiente de aplicação no qual a plataforma de robô móvel será destinada. Além disso, decidiu-se pela adoção do modelo de tração diferencial, por ser comumente empregado na robótica móvel. O modelo cinemático da tração diferencial é menos complexo em relação às outras disposições possíveis. Ainda é possível citar algumas características dessa disposição, como simplicidade, relativo baixo custo e facilidade de construção. E, por fim, sua compreensão é mais intuitiva, o que facilita o desenvolvimento de seu controle de locomoção.

De modo a suportar uma quantidade inicial de aplicações, foi implementado um conjunto padrão de módulos de hardware na plataforma apresentada. Desta forma, foi implementado um sistema de

locomoção, além de mecanismos de suporte às técnicas relativas de localização e navegação.

Os dispositivos atuadores implementados no Kihon foram:

- Dois motores de corrente contínua com caixa de redução;
- Um servomotor.

Já os dispositivos sensores implementados foram:

- Dois sensores ópticos reflexivos;
- Três módulos de sensores digitais de proximidade ópticos digitais;
- Dois sensores ópticos reflexivos de solo;
- Um sonar.

O gerenciamento dos dispositivos periféricos apresentados acima é realizado por meio de um único agente embarcado na base da plataforma, o Controlador de Robô Móvel. Dentre outras possibilidades de escolha para este Controlador, optou-se pela utilização do PIC18F4550, um modelo de microcontrolador da família PIC produzido pela Microchip. Diversos fatores contribuíram para sua escolha, como enquadramento ao escopo do projeto, baixo custo, baixo consumo e disponibilidade no mercado. Além disso, este microcontrolador possui funcionalidades que permitem programação otimizada em compiladores C, por meio do compilador MPLAB da Microchip.

A Figura 3 ilustra o diagrama em blocos dos dispositivos de *hardware* que foram implementados no Kihon.

O software embarcado é executado pelo microcontrolador. A Figura 4 ilustra o diagrama do Controlador concebido.

Devido à limitada taxa de transferência suportada pelo barramento de comunicação adotado, um simples protocolo de comunicação foi concebido de modo a alcançar o grau de abstração suficiente para garantir a conectividade das mensagens entre o Sistema de Controle e o Controlador. Neste protocolo, a comunicação é sempre iniciada pelo Sistema de Controle e deve ser iniciada e finalizada por caracteres

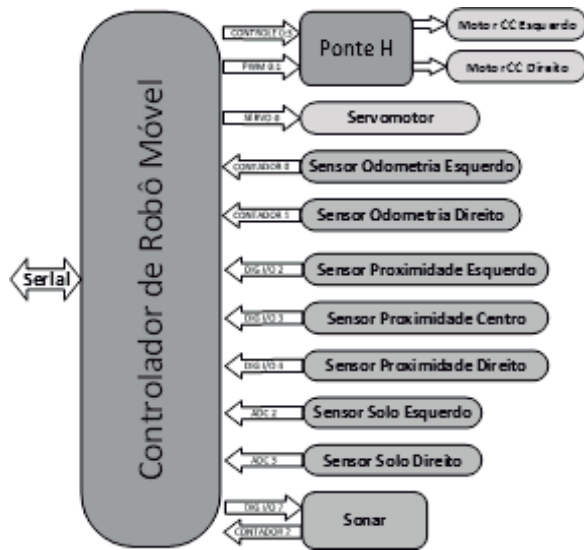


Figura 3 - Diagrama em blocos dos dispositivos implementados

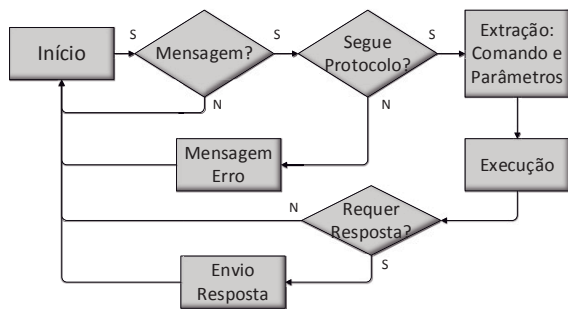


Figura 4 - Fluxograma do firmware do Controlador

específicos. Os comandos devem conter caracteres que correspondam ao código de operação, seguido por parâmetros, quando necessário. Por fim, uma API do Controlador da Plataforma de Robô Móvel foi implementada com o intuito de intermediar e facilitar o processo de comunicação entre o *software* de Controle e o Controlador. Esta API tem a função de abstrair o envio de comandos, interpretar os dados de resposta enviados pelo Controlador e, ainda, é encarregada de efetuar eventuais cálculos e conversões necessários para a configuração dos dispositivos do PIC, de modo a aliviar a carga de processamento do microcontrolador utilizado. Ao utilizar esta API para desenvolver aplicações com a plataforma, não é necessário conhecer a fundo todos os detalhes técnicos dos módulos de dispositivos de *hardware*.

A API foi escrita em linguagem C++, uma vez que esta linguagem oferece uma série de facilidades de comunicação com portos de E/S (entrada e saída) e também por se tratar de uma linguagem popular, além de possuir uma vasta base de conhecimento na internet.

#### 4. TESTES E RESULTADOS

Dois experimentos foram elaborados com o intuito de averiguar os quesitos da plataforma que motivaram o seu desenvolvimento. Buscou-se, ainda, através destes experimentos, verificar a questão da facilidade de uso da plataforma, tanto em termos de *hardware* como de *software*. Do ponto de vista de *hardware*, os aspectos observados foram sua adaptabilidade e facilidade de manutenção, importantes, principalmente, em vista do direcionamento de propósito geral dado a este trabalho. Já do ponto de vista de *software*, os aspectos observados foram se a plataforma oferece flexibilidade, por ser de código aberto e, ainda, independente da linguagem de programação do Sistema de Controle, já que sua interface de comunicação oferece tal possibilidade. Por último, foi avaliada a flexibilidade do barramento de comunicação da plataforma, quando submetida a diferentes tecnologias de Sistema de Controle.

##### 4.1 EXPERIMENTO 1

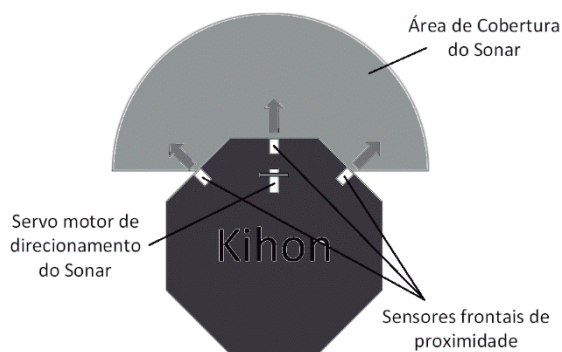
Neste primeiro experimento, buscou-se avaliar a capacidade da plataforma em suportar uma aplicação envolvendo um estudo básico e hipotético sobre navegação de robôs móveis. A questão da usabilidade da API concebida para a plataforma também foi verificada. Para tal, foi elaborado um experimento para verificar a capacidade da plataforma de navegar através de um ambiente, respondendo aos estímulos do ambiente em tempo real, quando controlado de forma remota. Um dos aspectos que também pode ser observado neste experimento é se a disposição dos sensores e dos atuadores escolhidos e implementados no robô móvel o torna capaz de se locomover e navegar de forma apropriada através do seu ambiente de aplicação. Neste sentido,

como tarefa escolhida para o experimento, foi determinado que o robô deveria navegar pelo ambiente seguindo uma parede e evitando colisões. A averiguação se deu tendo como base o comportamento do robô móvel Kihon ao interagir com o ambiente.

Este experimento foi realizado empregando um notebook atuando como uma base computacional remota e sua comunicação sem fio foi estabelecida por meio de módulos Xbee. Este experimento buscou aproximar-se das condições usuais, nas quais, por exemplo, alunos utilizam seus computadores pessoais para desenvolver experimentos envolvendo técnicas de navegação e locomoção de robôs móveis. A base computacional remota executa o *software* do Sistema de Controle em um computador cujo sistema operacional é o Microsoft Windows 8. Computadores pessoais, em especial os notebooks, em conjunto com o sistema operacional da plataforma Windows, costumam ser familiares na maior parte dos usuários, tornando tais ambientes uma alternativa comum para o desenvolvimento de *software*, seja por parte dos usuários, como alunos ou pesquisadores. O *software* de navegação do experimento foi escrito em C++, uma vez que a API do Controlador já havia sido concebida anteriormente nessa linguagem. Além disso, o C++ se apresenta frequentemente como uma linguagem didática no desenvolvimento de algoritmos, sendo comum em disciplinas da graduação e, também, como linguagem para desenvolvimento de *software*.

Para realizar este experimento, foi utilizado um sensor de distância (Sonar) acoplado ao servomotor para realizar a varredura da parte frontal do robô móvel e auxiliado pelos três sensores de proximidade frontais. A Figura 5 ilustra a vista superior do Kihon e a disposição dos sensores frontais de proximidade, sonar e do servomotor.

A arquitetura de controle concebida para o experimento se enquadra na arquitetura de subordinação (ou subsunção) apresentada por [2,5], característica do paradigma comportamental, uma vez que baseia-se no comportamento observado em animais. A partir disso, um simples algoritmo de seguir parede foi implementado. Neste algoritmo, uma série de 5 medições



**Figura 5** - Disposição dos sensores do Kihon utilizados no Experimento 1.

do sonar é coletada ao longo dos 180 graus (sendo uma medição a cada 45 graus) da parte frontal da base do robô móvel Kihon. O direcionamento do Sonar foi realizado utilizando o servomotor. Essas medições são feitas de modo a detectar a presença da parede em conjunto com leituras dos sensores de proximidade, na detecção de uma colisão frontal iminente. Os sensores de proximidades auxiliaram na detecção de colisões iminentes nos casos em que o sonar utilizado não foi capaz de detectar os obstáculos. A velocidade de tração dos motores foi controlada por modulação de largura de pulso – PWM.

A partir das medições levantadas pelos sensores, o Kihon foi capaz de navegar através do ambiente acompanhando a parede encontrada. O tempo de resposta do sistema (processamento e comunicação) médio de aproximadamente 20ms não comprometeu a reatividade do Kihon. Logo, o robô móvel foi capaz de cumprir o objetivo do experimento. Quanto à API do controlador utilizado, este foi capaz de oferecer acesso aos módulos de *hardware* implementados no Kihon de forma descomplicada.

## 4.2 EXPERIMENTO 2

O segundo experimento teve como objetivo averiguar a capacidade da plataforma em oferecer suporte a uma aplicação, e que neste caso, foi baseada em um trabalho realizado por [9], onde os autores coletaram e estudaram a umidade relativa do ar

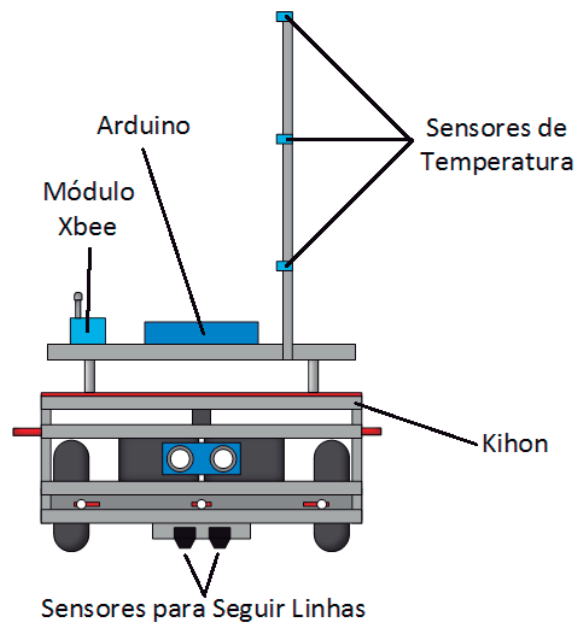
dentro de uma estufa agrícola. A estufa teve sua área mapeada e dividida em 114 pontos de medição. De modo a se obter sua variabilidade espacial, cada um desses pontos tiveram suas medições coletadas manualmente em três diferentes alturas em relação ao solo, sendo estas, 30 cm, 120 cm e 200 cm, em três horários ao longo do dia. Nesse contexto, foi atribuído ao robô móvel, a missão de executar uma tarefa em um cenário similar, seguindo um circuito determinado com linhas pretas no chão e coletando amostras de temperatura em três diferentes alturas, e em determinados pontos de seu plano de navegação. A partir dos dados levantados, é possível, por exemplo, fazer um levantamento espacial da variação de temperatura de uma estufa ou um ambiente qualquer, em diferentes condições como horários, estações, climatizações, etc.

Adicionalmente, este experimento foi elaborado buscando avaliar a flexibilidade da plataforma de robô móvel em termos de tecnologias de sistemas de controle e sua usabilidade. Para tal, empregou-se um Arduino Uno R3 para realizar o Sistema de Controle. O sistema de controle foi embarcado no robô móvel, fixando-o sobre o suporte do robô móvel Kihon. E por meio de seu porto serial, o Arduino foi conectado diretamente à interface de comunicação do robô móvel Kihon.

Para a realização deste experimento, três sensores de temperatura foram adicionados ao robô móvel, de modo que o permita coletar amostras de temperatura do ambiente em três alturas diferentes a partir do solo. Para isso, uma coluna de alumínio foi fixada sobre o suporte da plataforma do Kihon e cada sensor foi fixado na coluna de modo que sua posição em relação ao solo corresponda às alturas de 50cm, 100cm e 150cm, respectivamente. As saídas dos sensores de temperatura foram conectadas às entradas do conversor AD do Arduino.

A disposição dos dispositivos de hardware utilizados neste experimento no Kihon são mostrados na Figura 6.

Para seguir o traçado do circuito, um par de sensores óptico reflexivos (sensores para seguir linha) foram empregados para efetuarem a leitura da variação da coloração do solo. O circuito foi marcado com

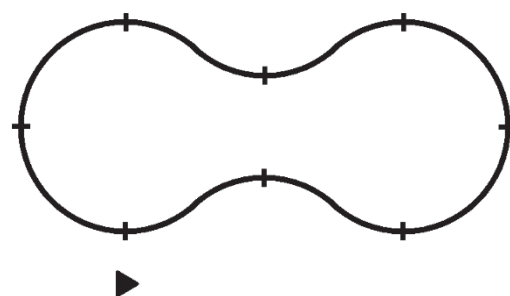


**Figura 6** - Disposição dos dispositivos utilizados no Experimento 2.

uma fita de cor preta sobre o chão de cor clara (branca). O circuito inclui curvas, sem bifurcações e, demarcações indicando os locais de coleta de temperatura.

O traçado do circuito é mostrado na Figura 7.

A partir das medições coletadas pelos sensores empregados para seguir li-



**Figura 7** - Vista superior do circuito elaborado para ser percorrido pelo Kihon.

nha, o robô foi capaz de seguir o traçado marcado, efetuando medições de temperatura nos locais determinados. Os valores de temperatura medidos em cada local foram transmitidos a uma base de coleta de dados remota através de módulos Xbee. Para a recepção desses dados foi utilizado outro mó-



dulo Xbee conectado a um notebook, onde um *software* ficou responsável por armazenar os dados recebidos em um arquivo de registro. A programação desta aplicação foi concebida em C++ e então adaptada para o ambiente Arduino.

Quanto à capacidade em oferecer suporte aos experimentos voltados à pesquisa, o robô móvel foi capaz de fazer uso adequado de seus sensores de solo ao seguir corretamente o traçado estipulado e, em conjunto com sensores acoplados diretamente à plataforma de controle (realizada pelo Arduino), a aplicação foi capaz de realizar as medições de temperatura nos determinados pontos do plano de navegação. Essa capacidade da plataforma de atuar como uma base de robô móvel o torna apto a suportar iniciativas práticas envolvendo a robótica móvel, uma vez que oferece funcionalidades essenciais de navegação, locomoção e ainda, sua arquitetura de *hardware* aberta o possibilita operar em conjunto com dispositivos adicionais externos. Neste caso, tais dispositivos foram a base de coleta de dados remota e os três sensores de temperatura conectados diretamente ao Sistema de Controle. No entanto, em outro cenário de aplicação, poderia ter sido uma câmera de vídeo ou um módulo GPS.

## 5. CONCLUSÕES GERAIS

Para a concepção da plataforma de robô móvel apresentada, foram considerados aspectos como simplicidade e baixo custo, de modo a tornar seu uso mais acessível e motivador. A plataforma apresentada foi averiguada através da concepção do robô móvel Kihon. Por apresentar uma solução de baixo custo, o *hardware* implementado no robô móvel Kihon apresenta poucos componentes, como sensores e atuadores, que podem ser encontrados sem dificuldade no mercado, de modo que a replicação da plataforma apresentada pode ser efetuada facilmente.

A interface de comunicação adotada mostrou-se suficiente para realizar a comunicação de dados entre a plataforma e o *software* de controle sem comprometer sua reatividade. Apesar das evidentes limitações trazidas com a escolha de um pa-

drão legatário, o resultado satisfatório se deu em virtude do simples protocolo de comunicação e também com a abstração das requisições e dos dados. Além disso, atribuindo à API do Controlador a tarefa de efetuar cálculos e conversões permitiu aliviar a carga de processamento sobre o microcontrolador, otimizando seu tempo de resposta aos comandos (requisições) de controle. A interface de comunicação adicional oferecida por meio de um segundo conector físico mostrou ser uma alternativa interessante para conexão de dispositivos que não utilizam o padrão de conector DB9, além de oferecer opções para alimentação de dispositivos de baixa potência.

Através dos resultados obtidos nos experimentos, é possível concluir que a plataforma concebida neste trabalho é capaz de se apresentar como uma alternativa viável para facilitar o desenvolvimento de aplicações gerais envolvendo a robótica móvel, podendo ainda, ser destinada ao uso didático ou científico. Com a adoção de conceitos simples e o emprego de um número reduzido de componentes, é possível obter uma plataforma de robô móvel que apresenta, dentro de suas limitações, um certo grau de flexibilidade e, ainda, oferece recursos de modo a torná-la funcional para execução de aplicações envolvendo locomoção e navegação de robôs móveis.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BLANK D.; KUMAR D.; MEEDEN L.; YANCO H. *Pyro: a Python-based versatile programming environment for teaching robotics*. Journal on Educational Resources in Computing, v. 3, n. 4, dez. 2003.
- [2] BROOKS, R. A.; *A robusted layered control system for a mobile robot*, IEEE Journal of Robotics and Automation, v. RA2, n. 1, mar. 1986.
- [3] BRUYNINCKX, H. *Open robot control software: the OROCOS project*, Proceedings 2001 ICRA. IEEE International Conference on. IEEE v. 3, 2001. p. 2523-2528.
- [4] GERKEY B.; VAUGHAN R.; HOWARD A. *The Player/Stage project: tools for multi-robot and distributed sensor systems*. Proceedings 11th International Conference on Advanced Robotics, jun 2003. p. 317-323.
- [5] MATARIĆ, M. J. *The robotics primer*. [S.I.]: MIT Press, 2007.

[6] MERTEN, M.; GROSS, H.-M.; *Highly adaptable hardware architecture for scientific and industrial mobile robots*, Proceedings of 2008 IEEE Conference on Robotics, Automation and Mechatronics (RAM 2008), 2008. p. 1130-1135.

[7] MURPHY, R.; *Introduction to AI robotics*, [S.l.] MIT Press, 2000.

[8] SARANLI U.; Avcı, A.; Öztürk, M.C.; *A modular real-time fieldbus architecture for mobile robotic platforms*, IEEE Transactions on Instrumentation and Measurement, v. 60, mar. 2011.

[9] VOLTAN, D. S.; BARBOSA, R. Z.; MARTINS, J. E. M. P.; ZIMBACK, C. R. L.; *Análise da distribuição espacial da temperatura do ar em uma casa de vegetação*, II Simpósio de Geoestatística Aplicada em Ciências Agrárias, mai. 2011.

**Takao Matsumura** é graduado em Bacharelado em Ciência da Computação pela Universidade Estadual Paulista Júlio de Mesquita Filho - UNESP, campus de Bauru (2011). Possui Mestrado em Ciência da Computação pela Universidade Estadual Paulista Júlio de Mesquita Filho - UNESP, campus de São José do Rio Preto (2014). Possui interesse em pesquisas na área de Robótica.

**Humberto Ferasoli Filho** é graduado em Engenharia Civil pela Fundação Educacional de Bauru (1982), possui Mestrado em Ciência da Computação pela Universidade Federal de São Carlos (1992) e Doutorado em Engenharia Elétrica pela Escola Politécnica da Universidade de São Paulo (1999). Atualmente é professor assistente doutor do Departamento de Computação da Faculdade de Ciências do campus de Bauru da Universidade Estadual Paulista Júlio de Mesquita Filho - UNESP. Coordena o Grupo de Integração de Sistemas e Dispositivos Inteligentes GISDI. Atua na área de Robótica, especificamente com arquiteturas de controle de robôs móveis autônomos com foco em robótica social, ou “assistiva”, e em educação.

**Norian Marranghello** é graduado em Engenharia Eletrônica pela Pontifícia Universidade Católica do Rio Grande do Sul (1982), possui Mestrado em Engenharia Elétrica pela Universidade Estadual de Campinas (1987), Doutorado em Engenharia Elétrica pela Universidade Estadual de Campinas (1992), Pós-Doutorado em Sistemas de Computação pela Universidade de Aarhus na Dinamarca (1998) e Livre-Docência em Sistemas Digitais pela Universidade Estadual Paulista (1998). Atualmente é Professor Titular da Universidade Estadual Paulista. Tem experiência nas áreas de Engenharia Elétrica e de Ciência da Computação, com ênfase em Sistemas Digitais, atuando principalmente nos seguintes temas: sistemas digitais integráveis, modelagem e simulação de sistemas, arquiteturas reconfiguráveis, redes de Petri e síntese de sistemas digitais.