

Sistema de Navegação para Robôs Autônomos em Ambientes Fechados.

Odivio Caio Santos Matos¹; Anfranserai Morais Dias²;

1. Bolsista PIBIC/CNPq, Graduando em Engenharia da Computação, Universidade Estadual de Feira de Santana, e-mail: odiviobzr@gmail.com

2. Orientador, Departamento de Tecnologia, Universidade Estadual de Feira de Santana, e-mail: anfranserai@ecomp.uefs.br

PALAVRAS-CHAVE: Sistemas Inteligentes; Automação; Microcontrolador

INTRODUÇÃO

O projeto colônia de robôs foi concebido inicialmente, com o objetivo de utilizar microcontroladores como unidade central de processamento para os robôs. Vários sistemas foram desenvolvidos para os robôs durante pesquisas passadas, porém com o passar do tempo e com as novas tecnologias, essa arquitetura mostrou-se limitada. Com isso boa parte dos novos projetos acabavam por realizar o desenvolvimento de trabalhos existentes, em uma nova tecnologia, para poder prosseguir com a pesquisa.

Visando agilizar a continuação de trabalhos de pesquisa, uma atualização geral no projeto era necessária. O laboratório realizou a aquisição de quatro placas Raspberry Pi modelo A+, figura 1. Essas placas são mini-computadores do tamanho de um cartão, que possibilitam a instalação de sistemas operacionais. Com isso, foram necessários o desenvolvimento de novas placas de controle, instalação do ROS (*Robotics operating system*), que é um sistema operacional de robótica, e a integração do novo sistema na colônia de robôs.

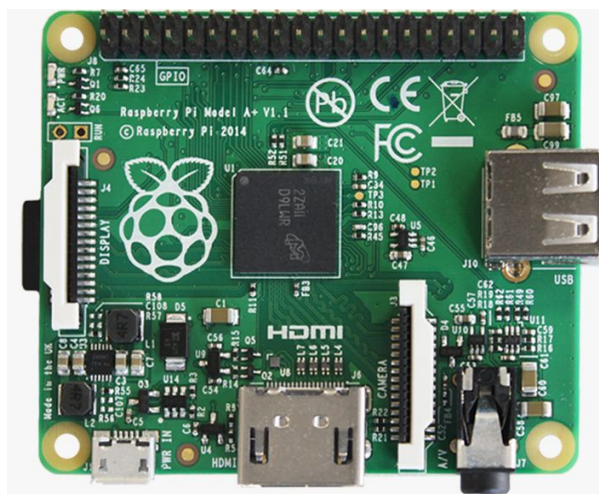


Figura 1: Raspberry Pi utilizado.

MATERIAL E MÉTODOS OU METODOLOGIA (ou equivalente)

A comunicação entre as placas foi realizada utilizando um protocolo serial. A placa desenvolvida envia determinadas informações, como valores da odometria e sensoramento, ao Raspberry Pi, no caso do ROS, responsável por realizar a análise dos

dados e responder ao PIC, caso necessário. Este modelo foi proposto devido a situações onde o PIC se torna incapaz de processar todas as informações simultaneamente enquanto faz a coleta de dados dos sensores. Este modelo também facilita a adição de outros componentes ao sistema, partindo do princípio que o Raspberry Pi possui WIFI, esta comunicação pode ser utilizada para se comunicar com o um servidor, por exemplo.

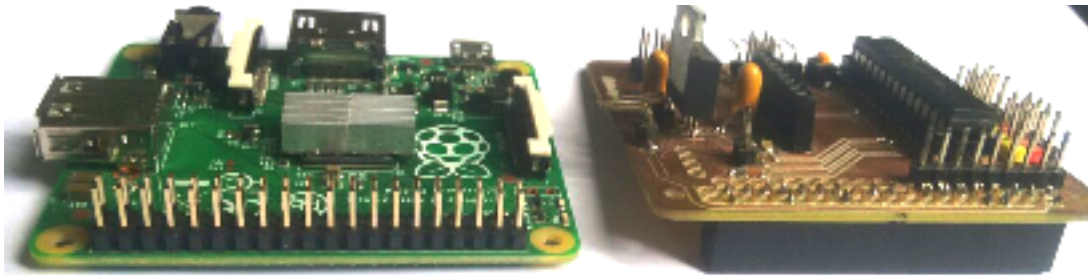


Figura 4:Placas utilizadas durante o projeto, Raspberry e Quengo.

Outra atividade foi o estudo e utilização de um bootloader, configuração utilizada para propiciar a gravação de microcontroladores sem a utilização de hardwares específicos, onde a sua função foi substituir o gravador que apresentou defeito durante o projeto. Para o desenvolvimento foi utilizado o Harmony, um “firmware” integrado, em conjunto com o MPLABX, IDE (*Integrated Development Environment*) que possui o objetivo de tornar o desenvolvimento mais rápido, oferecendo bibliotecas de periféricos, drivers e sistemas de serviço acessíveis e configuráveis.

RESULTADOS E/OU DISCUSSÃO (ou Análise e discussão dos resultados)

Primeiramente foi criado, como uma forma de entender melhor o novo sistema, uma máquina virtual para simular o raspberry pi, utilizando a ferramenta chamada QEMU, onde foi feita a primeira instalação do ROS. A utilização da máquina virtual foi de fundamental importância, que serviu como uma configuração inicial do ROS, pois não estava sendo possível utilizar o raspberry pi físico. O problema foi devido ao modelo utilizado possuir somente uma porta USB, onde não era possível a utilização de múltiplos periféricos, como mouse teclado e módulo WIFI simultaneamente.

Como uma forma de solucionar o problema primeiramente foi pensado a utilização de um USB HUB, o que não foi possível devido a concorrente, de cerca de 1A, do raspberry não ser suficiente para alimentar aparelho. Então outra solução encontrada foi a utilização do protocolo SSH (*Secure Shell*) para controlar o Raspberry através do WIFI. Porém devido ao projeto não possuir um modem dedicado, ocorreu um certo atraso com a obtenção e configuração do aparelho no laboratório.

Com a aquisição um modem para ser utilizado como rede principal do projeto de pesquisa, foi possível realizar o acesso via SSH ao Raspberry Pi. A Partir daí, o

primeiro passo foi iniciar a instalação do ROS, que levou cerca de 2 mês, isto devido aos problemas como falta de espaço no cartão de memória utilizado e ao baixo processamento do raspberry pi, onde algumas instalações de dependências levaram mais de seis horas de duração. Em paralelo, foi desenvolvido o projeto da nova estrutura da placa de controle e o estudo bibliográfico do novo microcontrolador a ser utilizado, no caso o PIC32.

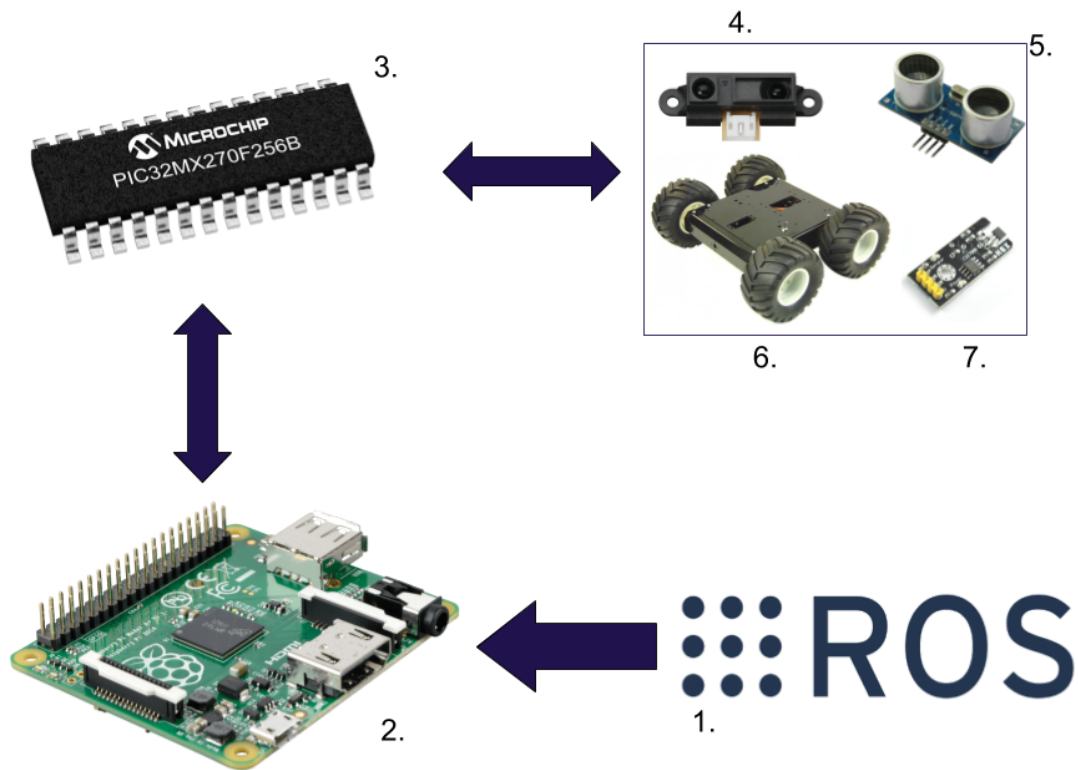


Figura 2: Diagrama do Projeto.

Legenda: 1. Robotic Operating System., 2. Raspberry Pi A+., 3. PIC32MX270F256B, 4. Sensor Infravermelho., 5. Sensor UltraSom, 6. Rover 4wd1(lynxmotion)., 7. Sensor de efeito Hall US1881., 8. *P. venosa* (Mart.) Baehni. (+) = presente; (-) = ausente.

A arquitetura final construída, figura 2, apresenta todos os componentes envolvidos no projeto. Como podemos observar o microcontrolador tem como um das funções coletar os dados dos sensores e enviá-los, através do protocolo RS232, ao Raspberry Pi, responsável por realizar o cálculo da odometria e de posicionamento do robô. Essa arquitetura foi necessária devido a problemas em projetos passados, onde o microcontrolador não possuía processamento suficiente para dar conta da coleta e análise de todos os dados dos periféricos.

CONSIDERAÇÕES FINAIS (ou Conclusão)

Os objetivos principais foram alcançados, como o desenvolvimento da nova placa de controle, figura 3, a modificação do sistema do robô e a criação parcial do seu novo algoritmo de controle. O projeto descrito no plano de trabalho, estava focado em atualizar a arquitetura, com a utilização do PIC32 em conjunto com o Raspberry Pi,

figura 4. Essa mudança garante uma maior fluidez para o desenvolvimento dos trabalhos e facilita a entrada de novas tecnologias no processo, sem que o trabalho seja refeito do zero.



Figura 4: Placa 'Quengo_04' desenvolvida.

Contudo, como discutido, a atualização do sistema de controle para o robô utilizado neste projeto não foi concluída com sucesso, possuindo somente uma parte do algoritmo finalizado. Isto foi devido aos problemas encontrados durante o desenvolvimento, como constantes paralisações da instituição de ensino, o aguardo para a entrega de determinados componentes e dificuldades enfrentadas pelo próprio bolsista durante a vigência da bolsa .

REFERÊNCIAS

DARYL ALBANO, Developing a Computer Vision System for Autonomous Rover Navigation. Disponível em: <http://astro.uhh.hawaii.edu/Alumni/Astronomy%20-%20Daryl%20Albano%20Fall/astr432_f16-Albano_Daryl.pdf> Data de acesso 11 março de 2017.

BRIGIT SCHROEDER, Robot Vision: Letting Robots See. Disponível em: <http://www.cs.uml.edu/~holly/teaching/91450/spring2013/bschroeder_vision_robotics1.pdf> Data de acesso 15 março de 2017.

CUAHEMOC CARBAJAL, Computer Vision Using Simplecv and the Raspberry Pi. Disponível em: <<http://homepage.cem.itesm.mx/carbajal/EmbeddedSystems/SLIDES/Computer%20Vision/Computer%20Vision%20using%20SimpleCV%20and%20the%20Raspberry%20Pi.pdf>> Data de acesso 15 março de 2017.

RICHARD SZELISKI, Computer Vision: Algorithms and Applications. Disponível em: <http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf> Data de acesso 17 março de 2017.

