Localização RSSI Assistida por Sensor Ultrassom

Autores

- Arthur Bridi Guazzelli
- Caio Pereira Oliveira
- Salomão Rodrigues Jacinto

Motivação

A localização dos nodos em uma RSSF é uma funcionalidade extremamente importante. Em aplicações como rastreamento de objetos, monitoramento do meio ambiente, rastreamento de inimigos em campo de batalha e sensoriamento para previsão de tempo, as informações coletadas pelos sensores apenas possuem sentido se atreladas a uma posição geográfica. [1] Especialmente em RSSF em ambientes internos, a estimativa de distância via RSSI é muito prejudicada pela presença de objetos, esse fenômeno é conhecido como *shadowing*. [2]

Objetivos

Esse trabalho tem como objetivo atacar o fenômeno de shadowing na medida do RSSI com o auxílio de um sensor ultrassom. Esperamos que com os dados do sensor seja possível realizar uma melhor estimativa da localizaçã do nó. Ou, ao menos, melhorar a medida do grau de confiança sobre a informação de localização.

Metodologia

Utilizaremos a placa EPOSMote III que conta com o protocolo TSTP no qual são feitas as estimativas de localização e distância. O sensor ultrassom que será utilizado é o HC-SR04 e o motor de passo 28BYJ-48 que orientará o sensor na direção correta.

Tarefas

- Implementar driver do sensor HC-SR04 para o EPOS.
- Implementar driver do motor de passo 28BYJ-48 para o EPOS.
- Abstrair o controle do motor de passo para um sistema de coordenadas cartesianas.
- Melhorar a localização do protocolo TSTP do EPOS utilizando a leitura do sensor ultrassom.

Entrega

27/11: Entrega do código produzido e documentação do projeto na página do EPOS Makers.

Documentação

Todo código produzido nesse projeto (incluindo testes) pode ser encontrado através do github https://github.com/MaoRodriguesJ/INE5424

Motor de passo 28BYJ-48 e driver ULN2003

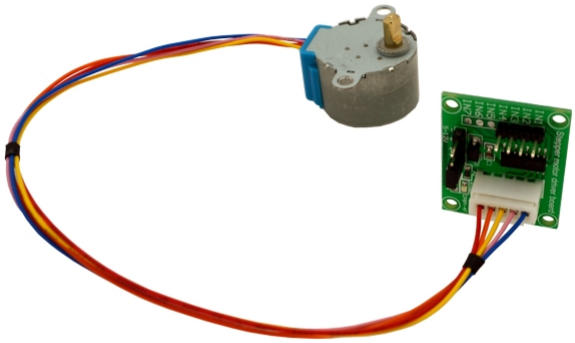

Figura: motor de passo 28BY-J48 (esquerda) e driver ULN2003 (direita)

A partir do estudo do funcionamento de motores de passo[6] e das características do motor de passo 28BYJ-48[5], foi desenvolvido um driver do motor para a plataforma EPOSMote III. O driver utiliza o método Wave Drive para comandar as bobinas do motor, ou seja, apenas um estágio está ativo em cada passo, como mostra a tabela a seguir. Outros métodos de controle existem para fornecer maior torque ou precisão ao motor e podem ser implementados no driver futuramente.

Wave Drive	Passos			
Estágios	1	2	3	4
Azul (A)	**HIGH**	LOW	LOW	LOW
Rosa (B)	LOW	**HIGH**	LOW	LOW
Amarelo (C)	LOW	LOW	**HIGH**	LOW
Laranja (D)	LOW	LOW	LOW	**HIGH**

	28BYJ-48 com Wave Drive	
Motor	Ângulo de passo	11,25°
	N° de passos por revolução	32
Eixo de Saída	Ângulo de passo	~0,17578°
	N de passos por revolução	2048

Para usar o driver do motor de passo em sua aplicação é necessário instanciar um StepperDriver, fornecendo os pinos do EPOSMote que farão o controle do motor e a velocidade do motor em passos por segundo. Opcionalmente também é possível fornecer o ângulo que o motor tomará como inicial para o controle através de um sistema de coordenadas.

StepperDriver constructor

StepperDriver(GPIO IN1, GPIO IN2, GPIO IN3, GPIO IN4, float motor_steps_per_second, float current_angle = 0);

Existem dois métodos para controlar o motor, um por número de passos e outro por ângulo de rotação. O

controle por número de passos é a base do funcionamento do motor, entretanto, não é muito intuitivo para o desenvolvedor da aplicação. O segundo método faz o controle do motor através de um ângulo de rotação em graus. O ângulo mínimo de rotação do motor é de 0,17578125° (1 passo). Em ambos os métodos, valores negativos rotacionam o motor no sentido horário e valores positivos no sentido anti-horário.

Método de rotação em passos

```
void rotate_steps(int steps_to_move);
```

Método de rotação em graus

```
void rotate_degrees(float degrees_to_move);
```

A velocidade de rotação do motor também pode ser alterada após a instaciação utilizando o método abaixo, fornecendo a nova velocidade em passos por segundo.

Método para alterar a velocidade de rotação

```
void set_speed(const float motor_steps_per_second);
```

// Sessão em construção //

Sensor Ultrassom HC-SR04

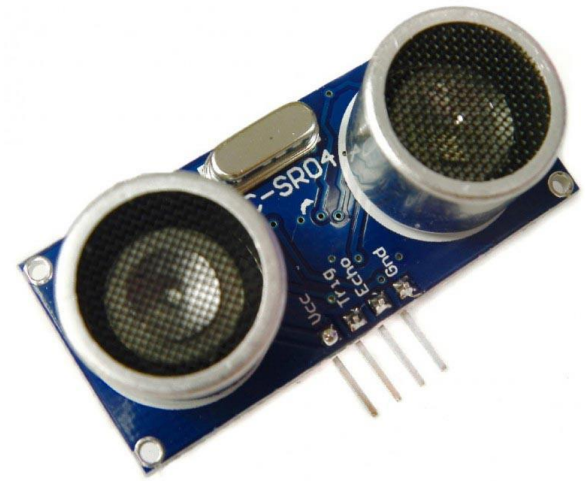

Figura: sensor ultrassom HC-SR04

Como o driver para o sensor HC-SR04 presente na biblioteca do EPOS não estava funcionando como esperado por razões desconhecidas, foi feita uma implementação do sensor de forma similar ao motor de passo.

O funcionamento do sensor é relativamente simples. Primeiro enviar um pulso de 10 microssegundos no pino Trigger, então o sensor enviará uma onda ultrasônica da direção em que estiver apontado e colocando o pino Echo em alta até que o sensor detecte a reflexão da onda. A medida da distância se dá através do período em que o pino Echo permaneceu em alta.[4]

Alcance Máximo	400cm
Alcance Mínimo	2cm

Ângulo de medida	15°

As complicações na implementação do driver para o sensor desapareceram após o uso do cronômetro presente na biblioteca do EPOS para medir o período em alta do pino Echo.

Para utilizar o sensor em sua aplicação, deve-se instanciar o driver fornecendo os pinos Trigger e Echo que serão usados.

HC-SR04 constructor

```
Sensor_HCSR04(GPIO trigger, GPIO echo);
```

Após isso, a leitura da distância pode ser feita através do método *sense* que retorna a distância em centímetros.

Metódo de leitura do sensor ultrassom

```
unsigned int sense();
```

// Sessão em construção //

Integração com EPOSMote III

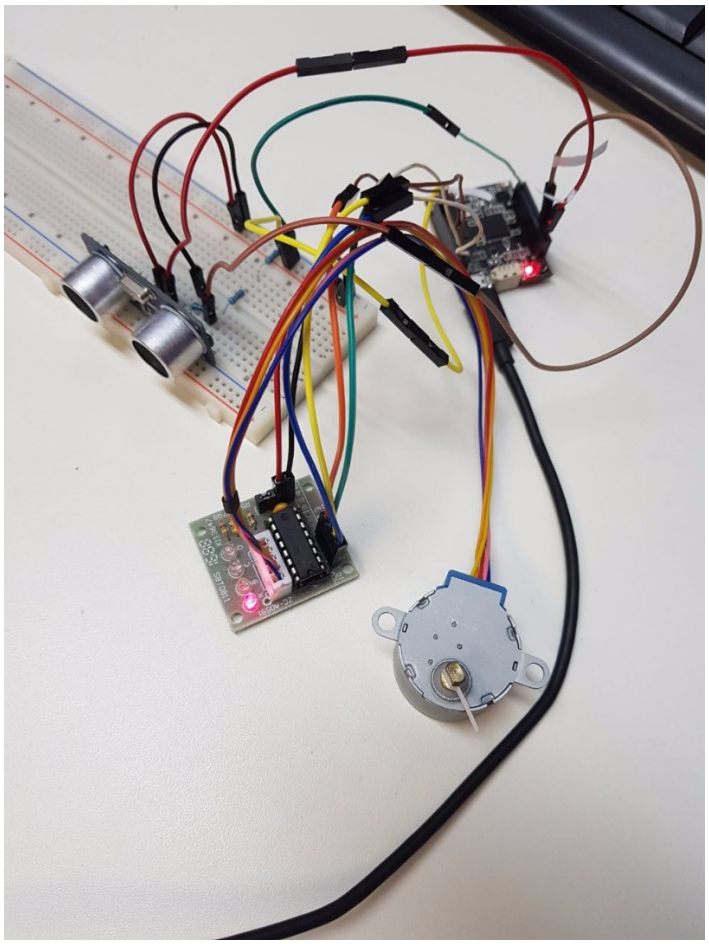

Figura: foto do circuito final do projeto

Todos os componentes eletrônicos usados nesse projeto necessitam de 5V de alimentação, para isso, foi usado o pino 5V USB do EPOSMote. É importante notar que os pinos digitais do Mote são de 3.3V, assim, foi usado um divisor de tensão entre o pino Echo do sensor ultrassom e o Mote para que a tensão chegue

em níveis nominais ao EPOSMote. Ilustrado nas imagens a seguir.

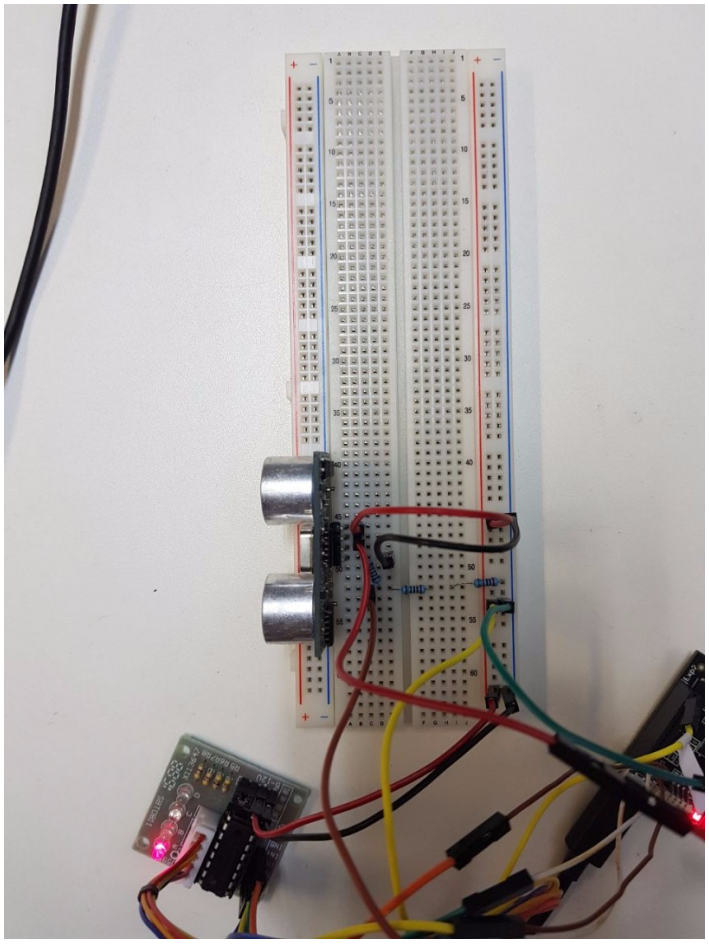

Figura: foto das conexões na protoboard

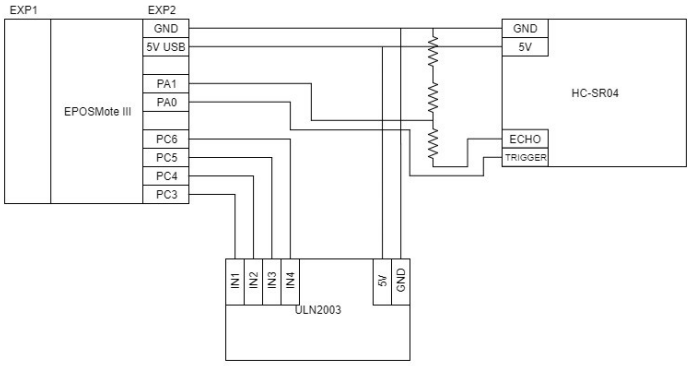

Figura: esquemático do circuito final do projeto

// Sessão em construção //

Bibliografia

1 G. Gracioli, A. A. Fröhlich, Member IEEE, R. P. Pires and L. Wanner. **Evaluation of an RSSI-based Location Algorithm for Wireless Sensor Networks** . Disponível em: http://www.lisha.ufsc.br/pub/Gracioli_IEEEAL_2011.pdf

2 K. Heurtefeux, F. Valois. **Is RSSI a good choice for localization in Wireless Sensor Network?** Disponível em: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6184942&tag=1

3 **EPOSMote III Schematics**. Disponível em: http://epos.lisha.ufsc.br/EPOSMote+III

4 **Ultrasonic Ranging Module HC - SR04 Datasheet**. Disponível em: http://www.micropik.com/PDF/HCSR04.pdf

5 **28BYJ-48 – 5V Stepper Motor Datasheet**. Disponível em: http://robocraft.ru/files/datasheet/28BYJ-48.pdf

6 B. Stateham. **28BYJ-48 Stepper Motor and ULN2003 Driver Intro**. Disponível em: https://www.youtube.com/watch?v=B86nqDRskVU