

Rede de sensores para detecção de incêndios

Relatório do Projecto

João Jerónimo, N.º 62521

André Garção, N.º 66942

Introdução

O sistema desenvolvido trata-se de uma rede de sensores para detecção de incêndios composta por motes sem fios. A rede auto-configura-se e os motes comunicam entre si para fazerem chegar a um servidor parâmetros ambientais tais como humidade, temperatura e presença de fumo, com o propósito de antecipar potenciais incêndios.

Os motes são programados na linguagem nesC, utilizando o sistema operativo TinyOS, especialmente desenvolvido para este tipo de redes. A instalação é fácil, bastando apenas colocar os nós no sítio onde eles estão destinados. Existe um ambiente de simulação baseado no TOSSIM, que permite simular os eventos, trocas de mensagens, ocorrência de incêndios, etc.

Arquitectura do sistema

Existem 3 tipos de nós: o **nó raiz**, os **nós de *routing*** e os **nós de sensores**.

Os nós de sensores implementam a funcionalidade essencial da rede, que é monitorizar as condições de deflagração de incêndio e detectar fontes de ignição. Fazem-no por meio de 3 tipos de sensor: temperatura, humidade e detecção de fumo.

A função do nó raiz é comunicar a um computador as leituras, feitas pelos nós de sensores, que a ele chegam através da rede.

Já os nós de *routing* servem de intermediários através dos quais os nós de sensores fazem chegar as suas mensagens ao nó raiz.

Todos os nós estão totalmente implementados num único componente nesC.

Sensores e escolhas de hardware

Os nós de sensores têm o propósito de monitorizar as condições existentes de aparecimento e propagação de incêndios na área onde foram colocados. Também é essencial que a sua localização exacta seja conhecida pelo servidor. Como em qualquer rede de sensores, nesta rede os nós de sensores precisam de comunicar entre si.

Os componentes existentes nos nós de sensores reflectem isto. Cada nó de sensores dispõe de um módulo de radiofrequência, um sensor de medição de humidade do ar, um sensor de temperatura, um sensor de fumo e um módulo de GPS. Na tabela seguinte são apresentados com mais detalhe os módulos que foram escolhidos para os motes e os respectivos gastos de energia.

	Tipo	Nome	Voltagem	Corrente	Potência
	GPS	GE 864-GPS	4,5 V	0,0015000 A	0,0067500 W
+	Mote				
	Rádio	TELOSB	3 V	0,0000261 A	0,0000783 W
	Temperatura				
	Fumo	RE46C120	6 V	0,0000045 A	0,0000270 W
					0,0068553 W

As reduzidas potências em causa permitirão que o mote se mantenha operacional durante um longo período sem necessidade de trocar as pilhas.

Comunicação

A rede tem um nó especial, o **nó raiz**, que se encontra ligado a um computador. Este nó centraliza toda a informação recolhida pela rede.

A rede utiliza o protocolo *flooding* para fazer chegar os pacotes do nó raiz. As trocas de mensagens são sempre feitas de forma aberta (broadcast), para que o *routing* se mantenha sob o controlo da aplicação.

O primeiro passo da comunicação consiste no registo dos nós sensor junto do nó raiz. Para esse efeito, cada nó sensor envia uma mensagem que é levada pelos nós de *routing* até ao servidor, indicando as suas coordenadas e o número do nó. Esta mensagem serve também para o nó sensor saber para que nó de *routing* deve enviar as suas mensagens para fazer chegar os seus parâmetros (temperatura, humidade, presença de fumo e coordenadas GPS) ao nó raiz.

Após este passo de configuração, o funcionamento da rede consiste em os vários nós sensor enviarem o estado dos seus sensores através da rede até ao nó raiz. Para tal, o protocolo de *flooding* é implementado de modo a que todos os nós de *routing* reenviem sempre todas as mensagens que a eles cheguem, a menos que a mesma já tenha passado por esse nó.

Instalação

A rede é composta por motes MICAz, que terão de ser adequadamente programados com o código desenvolvido, não sem antes adaptar o mesmo a hardware real.

Adequadamente adaptado o código desenvolvido e programados os nós, estes devem ser espalhados pela área a cobrir, tendo em conta o alcance dos respectivos módulos e comunicação.

Espalhados os motes pela área a cobrir, o sistema deve começar a funcionar sem necessidade de configuração extra. O servidor será então notificado das coordenadas de cada mote assim que estes se forem auto configurando, e será periodicamente notificado da humidade, temperatura e presença de fumo junto a cada mote.

Simulação

O funcionamento da rede pode ser simulado com recurso ao TOSSIM. Para tal, basta compilar o código com o comando “make micaz sim”, e executar o script Python “SensorNetworkSimulation.py”.

Problemas encontrados

O desenvolvimento de um novo tipo de aplicação é sempre um projecto que exige alguma aprendizagem. No caso do sistema implementado notámos alguma dificuldade na adaptação à linguagem nesC, pois a linguagem é bastante rigorosa quanto à ligação dos componentes.

No plano do hardware, tivemos dificuldade a encontrar um sensor de fumo adequado. É fácil encontrar detectores de fumo preparados para interiores, mas não o é tão fácil quanto a detectores adequados para o nosso sistema.

Conclusão

Ao desenhar software para sistemas embebidos há que ter em atenção certas limitações, designadamente a nível de memória RAM, bateria ou custo. Isto motiva uma série de factores de design específicos para este tipo de sistemas. Há que procurar reduzir o tamanho do código, há que escolher componentes adequados que tenham um consumo reduzido de bateria, há que utilizar ferramentas adequadas.

No caso em análise, o sistema operativo TinyOS e a sua linguagem nesC são especialmente desenhados para a implementação de sistemas embebidos do tipo das redes de sensores. O TinyOS e o nesC definem em conjunto uma dinâmica de desenvolvimento que se baseia na transmissão de eventos e comandos. Esta dinâmica permite, por um lado, que os sistemas embebidos não precisem de recorrer ao *multithreading*, o que levaria a grandes gastos de recursos. Permite, por outro lado, manter o código legível, utilizando correntemente as abstrações utilizadas neste tipo de sistemas.