

Water Monitor

1. Arquitetura da aplicação

Esta aplicação tem como objetivo a realização periódica de coleta de dados da água. Esses dados são coletados a partir de dois sensores de luminosidade, um sensor de temperatura e um sensor de umidade. Os dados são enviados de forma espontânea, sem a necessidade de solicitação de envio.

Os próprios motes realizam o roteamento da mensagem até a aplicação de coleta, por isso, o endereço Ipv6 da interface na qual a aplicação de coleta se comunica deve ser conhecido por todos os motes. Além disso, utiliza-se um modelo Push, onde os sensores enviam periodicamente os dados captados à aplicação de coleta em um PC.

Detalhes de implementação:

- O período de realização da coleta dos dados é definida pela constante `COLLECT_PERIOD` (Water.h).
- Os dados são coletados por cada sensor um determinado número de vezes, definido pela constante `MAX_DATA_SET`. Assim que os dados são coletados, os mesmos são armazenados em uma estrutura do tipo `'data_t'` (Water.h), um vetor de dados de `MAX_DATA_SET` posições.
- Quando o vetor é totalmente preenchido, o mesmo é enviado juntamente a um serial, em uma estrutura do tipo `'packet_t'` (Water.h).
- Após o envio do pacote, a aplicação responsável por receber e tratar os dados deve enviar uma mensagem informando que recebeu as informações com sucesso (ACK). Dessa forma, o mote deve receber uma estrutura do tipo `'ack_t'` (Water.h), contendo o serial do pacote enviado anteriormente.
- O não recebimento da mensagem de ACK implica em falha ao enviar os dados coletados. Para essa caso, são tomadas as seguintes precauções:
 - Assim que os dados são enviados, inicia-se um período de espera pela mensagem de ACK, definido pela constante `ACK_WAIT_TIME` (Water.h).
 - Caso esse período acabe e a mensagem de ACK não tenha sido recebida, o pacote é retransmitido. O número de retransmissões é controlado pela constante `MAX_RETRANSMISSIONS` (Water.h).
 - Caso o pacote não seja retransmitido em nenhuma das tentativas, os dados são descartados, porém o serial é mantido para o próximo envio.
- Caso haja sucesso no envio, o mote deve receber a mensagem de ACK. Então, a aplicação nos motes incrementa o número serial e descarta os dados atuais do vetor.
- Se o coletor receber um pacote com serial repetido, isso significa que o mote não recebeu o ACK, então envia o ACK novamente e descarta os dados pois já foram computados.

2. Constantes definidas em Water.h

Abaixo, uma descrição das constantes utilizadas para a aplicação que executa nos motes:

Constante	Descrição
COLLECT_PERIOD	Intervalo de tempo definido para realização da coleta, definido em milisegundos.
ACK_WAIT_TIME	Tempo de espera pelo recebimento da mensagem de ACK da aplicação Ipv6, definido em milisegundos.
MAX_RETRANSMISSIONS	Número máximo de retransmissões no caso da aplicação dos motes não ter recebido o ACK em ACK_WAIT_TIME milisegundos.
MAX_DATA_SET	Define o número de coletas a serem realizadas por cada sensor.
DATA_NUM	Número de dados sendo coletados pelos sensores (temperatura, luminosidade, oxigênio)
SERVER	Armazena o endereço IPv6 da interface através da qual a aplicação IPv6 se comunica.

3. Estrutura das Mensagens

Mensagem	<i>data_t</i>	
Descrição	Estrutura de dados responsável por armazenar as informações de coleta do sensor. A cada coleta, preenche-se uma estrutura deste tipo.	
Campos	Tipo	Descrição
d1	uint16_t	Cada campo armazena uma informação coletada pelo sensor (temperatura ou luminosidade ou umidade).
d2	uint16_t	
d3	uint16_t	
d4	uint16_t	

Mensagem	<i>packet_t</i>	
Descrição	Estrutura de dados responsável por transportar um conjunto de coletas realizadas pelos sensores e um número serial.	
Campos	Tipo	Descrição
SERIAL	uint16_t	Serial utilizado para indicar sucesso ou falha no envio dos dados.
VETOR [MAX_DATA_SET]	data_t	Armazena um conjunto de coletas realizadas pelo sensor.

Mensagem	<i>ack_t</i>	
Descrição	Estrutura de dados para transportar a mensagem de reconhecimento (ACK) de recebimento dos dados dos sensores	
Campos	Tipo	Descrição
ACK	uint8_t	Inteiro para indicar que é uma mensagem de ACK. Seu valor é sempre 1.
SERIAL	uint16_t	Serial utilizado para indicar sucesso ou falha no envio dos dados.

4. Fluxo das Mensagens

