

CAMADA FÍSICA DA COMPUTAÇÃO

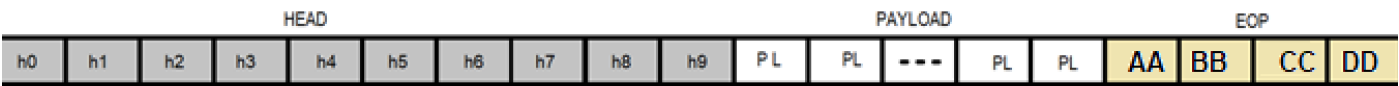
ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto

PROJETO 4 – Protocolo de comunicação UART ponto a ponto.

Agora você é um(a) engenheiro(a) de computação recém-contratado(a) para desenvolver a comunicação entre sensores de campo (que enviam periodicamente dados) e aplicações locais que armazenam os dados em um banco SQL. Os sensores não têm a funcionalidade de envio de dados via protocolo TCP-IP, porém têm um processador que pode rodar uma aplicação Python e também possui um chip UART, podendo comunicar-se serialmente.

Você então tem a tarefa de implementar uma aplicação para os sensores se comunicarem serialmente com padrão UART de maneira segura, sem perda de dados. A comunicação deve ser feita para envio de arquivos para os servidores, sendo uma rotina de envio executada pelo sensor toda vez que este tem um arquivo a ser enviado.

A camada superior da comunicação deve funcionar seguindo uma estratégia já definida, onde os arquivos são enviados em pacotes, respeitando o datagrama definido a seguir.



- h0 – tipo de mensagem
- h1 – livre
- h2 – livre
- h3 – número total de pacotes do arquivo
- h4 – número do pacote sendo enviado
- h5 – se tipo for handshake: id do arquivo (crie um)
- h5 – se tipo for dados: tamanho do payload
- h6 – pacote solicitado para recomeço quando a erro no envio.
- h7 – último pacote recebido com sucesso.
- h8 – h9 – CRC (Por ora deixe em branco. Fará parte do projeto 5)
- PAYLOAD – variável entre 0 e 114 bytes. Reservado à transmissão dos arquivos.
- EOP – 4 bytes: 0xAA 0xBB 0xCC 0xDD

IMPORTANTE: A MÉTRICA PARA SEU SUCESSO SÃO A INTEGRIDADE DOS DADOS RECEBIDOS E O THROUGHPUT !

Documento de definição do protocolo de comunicação

Padrao:
UART, baudrate 115200, sem bit de paridade

Datagrama
Cada envio deve ser feito como um datagrama completo, contendo head, payload e eop, ou seja, não é permitido envios que não contenham head, payload(ocasionalmente nulo) e eop. O tamanho do payload não pode ultrapassar 114 bytes e o tamanho do datagrama não deve ser maior que 128 bytes

Tipos de mensagens
TIPO 1 – Esta mensagem representa um chamado do cliente enviado ao servidor convidando-o para a transmissão. Nesse caso, o head deve conter o byte h0 com o número 1, indicando mensagem tipo 1, e outro byte com um identificador. O identificador é o número do servidor, sendo que quando este receber uma mensagem tipo 1, verifica se é para ele mesmo o envio. A mensagem tipo 1 já deve conter o número total de pacotes que se pretende enviar!
TIPO 2 – Essa mensagem é enviada pelo servidor ao cliente, após o primeiro receber uma mensagem tipo 1 com o número identificador correto. Deve conter no head o número 2 no byte reservado ao tipo de mensagem. O significado de uma mensagem tipo 2 é que o servidor está ocioso e, portanto, pronto para receber o envio dos pacotes.



CAMADA FÍSICA DA COMPUTAÇÃO

ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto

TIPO 3 – A mensagem tipo 3 é a mensagem de dados. Este tipo de mensagem contém de fato um bloco do dado a ser enviado (payload). Deve conter o número 3 no byte reservado ao tipo de mensagem. Essa mensagem deve conter também **o número do pacote que envia (começando do 1) e o total de pacotes a serem enviados**.

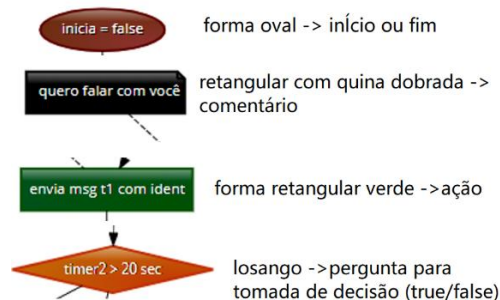
TIPO 4 – Essa mensagem é enviada do servidor para o cliente toda vez que uma mensagem tipo 3 é recebida pelo servidor e averiguada. Quando verificado que a mensagem é realmente o pacote que o servidor estava esperando e que tal mensagem chegou em perfeitas condições (eop no local correto), o servidor envia então a mensagem tipo 4, afirmando que recebeu o pacote. Essa mensagem deve ter o número 4 no byte reservado ao tipo de mensagem. Além disso, deve conter o número do último pacote recebido e já aferido.

TIPO 5 – É uma mensagem de time out. Toda vez que o limite de espera exceder o timer dedicado a isso, em qualquer um dos lados, deve-se enviar essa mensagem e finalizar a conexão. Essa mensagem deve ter o número 5 no byte reservado ao tipo de mensagem.

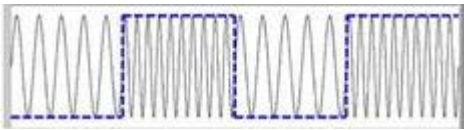
TIPO 6 – É uma mensagem de erro. O servidor deve enviar esta mensagem ao cliente toda vez que receber uma mensagem tipo 3 inválida, seja por estar com bytes faltando, fora do formato correto ou por não ser o pacote esperado pelo servidor (pacote repetido ou fora da ordem). Essa mensagem deve ter o número 6 no byte reservado ao tipo de mensagem. Além disso, deve conter o número correto do pacote esperado pelo servidor na posição h6, independentemente do problema que invalidou a mensagem. Isso orienta sempre o cliente para o reenvio.

A comunicação deve ocorrer de acordo com os diagramas a seguir (atente para a legenda):

Legenda:

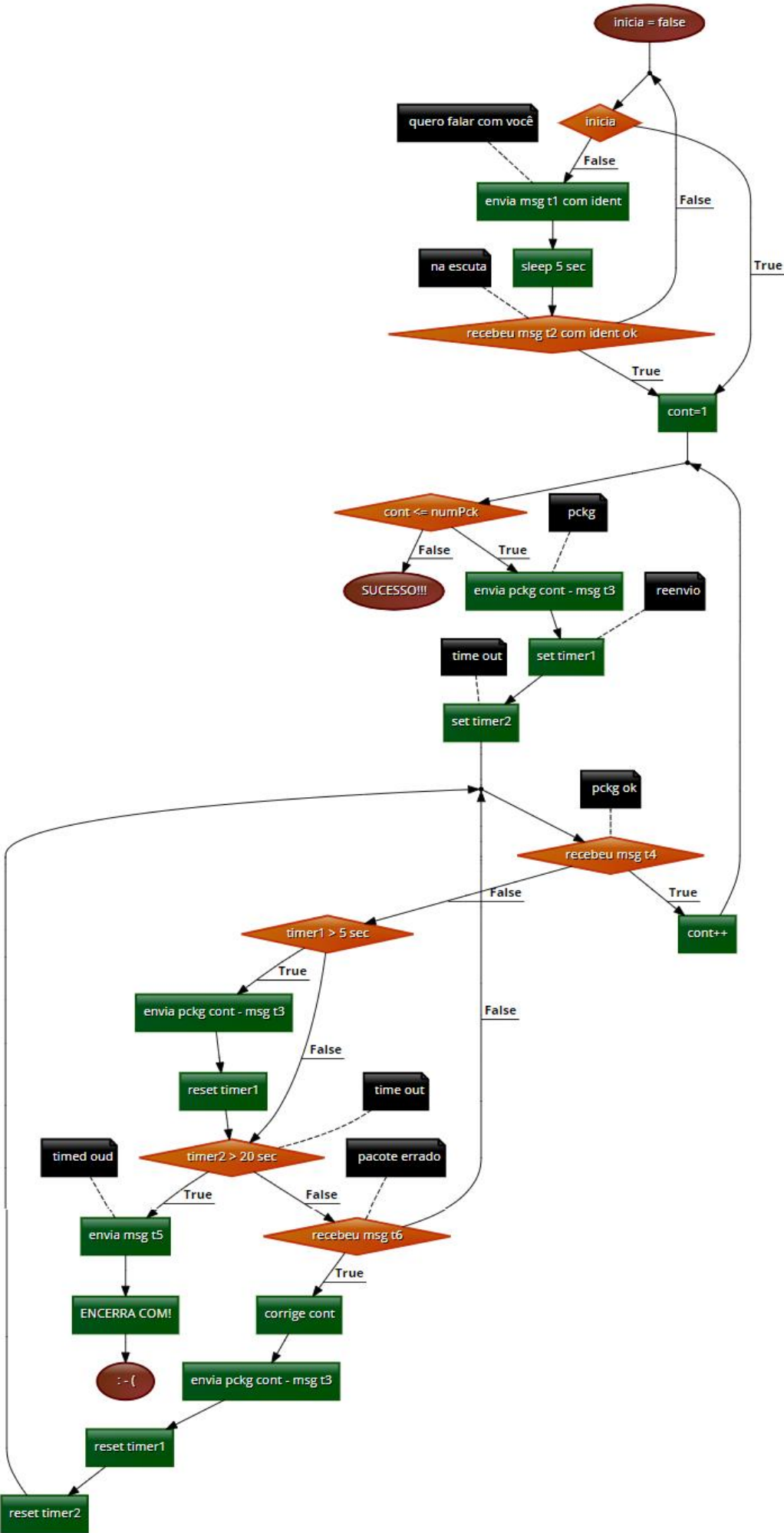


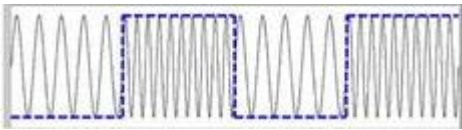
1) PROTOCOLO CLIENTE



CAMADA FÍSICA DA COMPUTAÇÃO

ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto





CAMADA FÍSICA DA COMPUTAÇÃO

ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto

2) PROTOCOLO SERVER

