

PSIU - Protocolo Simples de Intercomunicação Unificado

February 1, 2012

Introdução

Um protocolo de comunicação nada mais é do que um **conjunto de convenções que rege o tratamento e, especialmente, a formatação dos dados num sistema de comunicação**. Seria a "gramática" de uma "linguagem" de comunicação padronizada. Conhecemos vários protocolos de comunicação e fazemos uso deles diariamente, mas não pensamos neles como protocolos de comunicação. O mais antigo deles é a língua falada: duas pessoas que emitem sons audíveis aos ouvidos humanos podem se comunicar. Neste exemplo, o protocolo de comunicação é a emissão de sons numa dada faixa de frequência, o código utilizado é a língua falada e a mensagem é o conteúdo do que se fala. Conforme o site *Informática na Aldeia* [1],

Na robótica recorremos o tempo todo a protocolos de comunicação para enviar comandos à robôs, obter informação de sensores, efetuar telemetria de dados, etc. De uma forma não unificada, na maioria dos casos, cada projetista cria seus próprios comandos e desenvolvem programas para controlar seus robôs a partir de computadores, tablets ou outros dispositivos ou mesmos para robôs trocar informações entre si. Como esses comandos não são normatizados, eles acabam ficando limitados ao controle de poucos robôs desenvolvidos pela mesma pessoa ou empresa e não permite que os robôs se comuniquem entre si.

A proposta do PSIU é normatizar comandos, informações e formas de comunicação para que a conversa ocorra entre dispositivos independente dos programas de controle utilizados.

Entre inúmeros protocolos de comunicação que já foram criados, o modelo OSI é universalmente adotado e utilizado inclusive no TCP/IP (protocolo da Internet). De forma a ilustrar a abrangência do PSIU, ele representaria uma única camada do modelo OSI, a de aplicação. Isso quer dizer que o protocolo PSIU não normaliza as camadas mais baixas que definem o canal de comunicação e a forma que a mensagem chega até o destinatário. Ele apenas define palavras formadas em um padrão já consolidado, chamado ASCII, e essas palavras podem ser traduzidas em ações ou informações para robôs independente de sua arquitetura.

Sendo assim o protocolo PSIU pode ser utilizado em um canal serial (UART232, comN) consolidado no Sistema operacional de um PC, por exemplo, a partir de um driver específico (FTDI) e com comunicação por cabo. Como pode utilizar um driver similar consolidado através de uma comunicação bluetooth. Os mesmos comandos também podem trafegar em textos dentro de mensagens HTML ou sobre qualquer outro protocolo. O que o PSIU vem normatizar é o padrão em que o texto dos comandos são formatados.

Para garantir o envio de comandos em dispositivos com processamento limitado também existe a versão simplificada do PSIU que descreve comandos a partir de códigos hexadecimais de 8 bits sem a utilização da codificação ASCII.

[1] Informática na Aldeia, <http://www.numaboa.com.br/informatica/internet/526-protocolos>, acessado em fevereiro 2012

PSIU

O PSIU procura definir uma forma de comunicação para diferentes tipos de aplicação e volume de dados e diferentes topologias. Todas as mensagens (comandos e respostas) enviadas no PSIU são escritos em caractere ASCII. Um pacote básico de informação neste protocolo é mostrado abaixo:

Destinatário	Tamanho	Comando	Parâmetro	Checksum
--------------	---------	---------	-----------	----------

O campo destinatário é o nome de identificação contido em cada robô. O próprio robô conhece seu nome e portanto sabe se o destinatário está correto ou não.

O tamanho é o tamanho do pacote, ou seja, a quantidade total de bytes contido na mensagem.

O comando é o nome do comando e o parâmetro pode ser um inteiro, um float ou um char e vai depender do comando. Se o comando tiver parâmetros eles devem estar na mensagem, caso contrário, será mandada uma resposta de erro.

Checksum é a soma de todos os bytes da mensagem.

Um exemplo de uma mensagem de comando no PSIU está descrita abaixo:

MNERIM029parafrente 100 01884

Onde: *Destinatário = MNERIM* , *Tamanho = 029* , *Comando = parafrente*
Parâmetro = 100 , *Checksum = 01884*

Campos do pacote

Destinatário

Destinatário é o campo onde vai conter o nome do dispositivo a ser enviado a mensagem. Este nome pode conter letra maiúsculas, minúsculas, numerais, espaços e caracteres especiais, desde que sejam caracteres ASCII.

Cada dispositivo sabe seu nome e com isso ele só irá ler as mensagens que sejam mandadas diretamente para eles.

A quantidade de bytes é definido pelo tamanho do nome do dispositivo.

O número de caracteres do nome também pode variar entre os dispositivos, mas como cada um sabe seu próprio nome é possível indicar onde começa o próximo campo (tamanho da mensagem).

Tamanho

Tamanho é um campo de 3 bytes fixo que vai conter o tamanho total de bytes da mensagem. Sendo assim, o tamanho varia de 0 até 999. Se uma mensagem contém apenas dezenas de bytes, o tamanho de 3 bytes se mantém sempre fixo e no campo da centena será colocado um 0.

Exemplo: 033 (Mensagem de 33 bytes)

Comando

O campo comando vai conter o nome do comando a ser executado pelo dispositivo. Propriamente dito, é uma palavra chave que deve indicar intuitivamente a ação a ser tomada pelo destinatário.

No final de cada string de nome terá que existir um espaço " " (0x20). De forma ao dispositivo determinar o final de um comando.

Parâmetro

Parâmetro não é um campo obrigatório. Ele depende se o comando contém parâmetros ou não. Caso um comando não possua parâmetro este campo deve ser deixado em branco. Um parâmetro assim como o comando, deve conter no final um espaço em branco para determinar o final do parâmetro. Se o comando tiver dois ou mais parâmetros a separação entre eles é dada por espaço " " (0x20). Caracteres do tipo float têm a parte inteira separada da fracionária por ponto (1.25)

Checksum

Checksum é um campo de 5 bytes fixos de tamanho e é a soma de todos os bytes da mensagem (ele não incluído). Ele sempre termina uma mensagem. Como os 5 bytes são fixos, se a soma dos bytes der um número de 4 bytes, deverá ser acrescentado um 0 na frente, assim formando um CheckSum de 5 bytes.

Exemplo: 01884

Comandos

para frente

O comando *para frente* possui um parâmetro inteiro que varia de 0 a 999. Este comando diz ao robô a quantidade em *cm* que ele irá andar para frente. Se o comando for executado corretamente será recebido uma resposta de OK, caso contrário será recebido uma resposta de FALHA.

para trás

O comando *para trás* possui um parâmetro inteiro que varia de 0 a 999. Este comando diz ao robô a quantidade em *cm* que ele irá andar para trás. Se o comando for executado corretamente será recebido uma resposta de OK, caso contrário será recebido uma resposta de FALHA.

gira direita

O comando *gira direita* possui um parâmetro inteiro que varia de 0 a 360. Este comando diz ao robô a quantidade em *graus* que ele irá girar para direita. Se o comando for executado corretamente será recebido uma resposta de OK, caso contrário será recebido uma resposta de FALHA.

gira esquerda

O comando *gira esquerda* possui um parâmetro inteiro que varia de 0 a 360. Este comando diz ao robô a quantidade em *graus* que ele irá girar para esquerda.

Se o comando for executado corretamente será recebido uma resposta de OK, caso contrário será recebido uma resposta de FALHA.

Table 1: Comandos do PSIU

Nome do Comando	Parâmetro	Exemplo de comando
parafrente	1 int (0 : 999)	parafrente 100
paratras	1 int (0 : 999)	paratras 250
giradireita	1 int (0 : 360)	giradireita 120
giraesquerda	1 int (0 : 360)	giraesquerda 180

Mensagens Especiais

Essas mensagens não precisam seguir o padrão do protocolo (NOME TAMANHO COMANDO PARÂMETRO CHECKSUM). Um dispositivo entende a mensagem especial por causa do prefixo "?" (0x3F) incluso em mensagens desse tipo.

?qualseunome

O comando *?qualseunome* pergunta ao dispositivo o seu nome e recebe como resposta o nome do dispositivo.

Se houver interesse em se comunicar com o dispositivo, toda mensagem de comando é começada pelo campo nome. Caso não se saiba o nome do dispositivo, usa-se esse comando.

?listadecomandos

O comando *?listadecomandos* pergunta ao dispositivo suas funções disponíveis. O retorno dessa função é uma lista dos nomes das funções e seus parâmetros.

Respostas

OK

RespOK é enviado se um comando válido foi executado.

FALHA

RespFALHA é enviado se a execução de um comando falhou.