

MODELO DE COMUNICAÇÃO

A comunicação entre o software (driver do servidor de hardware) e o hardware (possivelmente um PIC) é feita através de uma determinada porta que pertence a um conjunto de portas específicas para uma determinada experiência, definido no ficheiro de configuração.

O driver utilizado deve ser o mais genérico possível, sendo que as definições de comunicação com cada um dos hardwares de experiências são definidas num ficheiro de configuração em XML.

A comunicação é iniciada pelo driver do servidor de hardware que começa por procurar uma dada experiência perguntando o identificador em cada conexão que estabelece em cada porta ou falhando no caso de não obter resposta em tempo útil, continuando a percorrer as portas em ciclo até ser encontrada.

Na troca de mensagens por porta RS232 entre o driver e o PIC, as mensagens do driver são escritas em caracteres minúsculos sendo que cada instrução deve ser definida no máximo por três caracteres, enquanto que as mensagens do PIC são escritas em caracteres maiúsculos. Cada parâmetro da mensagem é separado por um caracter tab (\t). O final de uma mensagem é marcada por um caracter carriage-return (\r) excepto quando estão a ser recebidos dados em binário provenientes do PIC. As mensagens recebidas pelo PIC provenientes do driver devem ser enviadas de volta ao driver após a sua recepção para garantir ao servidor de hardware a sua recepção pelo PIC.

O driver deve aguardar um número de segundos arbitrário, definido no ficheiro de configuração do driver para cada instrução, para que cada execução de instrução seja terminada. Caso contrário a execução será dada como falhada e irá ser iniciado o procedimento de reset por parte do driver, enviando uma mensagem *rst*.

MENSAGENS RS232

As mensagens trocadas entre driver e PIC são definidas por um conjunto de instruções e uma sintaxe definida nesta secção:

A mensagem *ids* é despoletada pelo driver em dois momentos:

- para iniciar uma conversaç o com um hardware, momento em que   importante determinar qual a identificaç o desse hardware;
- durante uma conversaç o para determinar o estado actual do hardware, por m nunca deve ser usado dentro de uma troca de mensagens iniciada por *DAT* ou *BIN*;

O PIC deve responder *IDS* seguido do identificador  nico da experi ncia e do estado actual da experi ncia.

Driver ⇒ PIC	<i>ids</i>
PIC ⇒ Driver	<i>IDS</i> <id do hardware> <status>

A mensagem *IDS* deve ser enviada frequentemente pelo PIC para as portas série para que o servidor de hardware tenha conhecimento da ligação com uma experiência e do estado actual em que esta se encontra.

A mensagem *cfg* é enviada pelo driver para o hardware para definir configurações para a experiência. O PIC deve responder com a lista de parâmetros recebidos e *CFGOK* quando as configurações forem aplicadas.

Driver ⇒ PIC	cfg <parâmetro 1> <parâmetro 2> ... <parâmetro N>
PIC ⇒ Driver	CFG <parâmetro 1> <parâmetro 2> ... <parâmetro N>
PIC ⇒ Driver	CFGOK

A mensagem *cur* é enviada pelo driver para que o hardware responda quais os parâmetros de configuração que estão actualmente definidos para a experiência.

Driver ⇒ PIC	cur
PIC ⇒ Driver	CUR <parâmetro 1> <parâmetro 2> ... <parâmetro N>

A mensagem *str* é enviada pelo driver para que o hardware dê início a uma experiência. O PIC deve responder com *STR*.

Driver ⇒ PIC	str
PIC ⇒ Driver	STR

Após a inicialização da experiência, o PIC deve enviar uma mensagem *DAT* ou *BIN* que servirá para iniciar a transferência de dados e também para avisar que a inicialização da experiência ocorreu com sucesso.

A mensagem *DAT* é enviada pelo PIC para que se inicie uma transmissão de dados adquiridos na experiência em forma textual. Em cada linha seguinte deve conter os dados de cada um canais que o driver espera receber, e pode ser opcionalmente acrescentado no final de cada linha o valor relativo de relógio para cada linha de dados adquirida, devendo ser a primeira linha de dados sempre relativa ao tempo de relógio zero.

PIC ⇒ Driver	DAT
PIC ⇒ Driver	<canal 1> <canal 2> ... <canal N> [clock relativo]
PIC ⇒ Driver	<canal 1> <canal 2> ... <canal N> [clock relativo]
PIC ⇒ Driver	...
PIC ⇒ Driver	END

A mensagem *BIN* é enviada pelo PIC para que se inicie uma transmissão de dados adquiridos na experiência em forma binária. Esta mensagem deve ser seguida do

número de bytes que o driver deve esperar receber a seguir ao return-carriage no fim desta resposta. Esse número de bytes de dados são enviados em seguida e não são terminados por return-carriage.

PIC ⇒ Driver	BIN <número de bytes>
PIC ⇒ Driver	<dados em binário>

A mensagem *stp* é enviada pelo driver para o hardware para que este termine uma experiência. O PIC responde com *STP*, e responde *STPOK* quando a finalização tenha terminado com sucesso.

Driver ⇒ PIC	stp
PIC ⇒ Driver	STP
PIC ⇒ Driver	STPOK

A mensagem *rst* é enviada pelo driver para o hardware para que este execute um reset. O PIC responde com *RST*. Deverá responder *RSTOK* a indicar que o reset foi executado e o PIC vai iniciar o seu ciclo de identificação.

Driver ⇒ PIC	rst
PIC ⇒ Driver	RST
É efectuado o reset do hardware...	
PIC ⇒ Driver	RSTOK

A mensagem *ERR* é enviada pelo PIC para o driver como aviso de erro durante a execução e é seguido do código do erro que ocorreu. Este código poderá ser enviado pelo PIC a qualquer momento. O código deve ser numérico.

PIC ⇒ Driver	ERR <código de erro>
--------------	-----------------------------

FICHEIRO DE DEFINIÇÕES

O ficheiro de definições inclui as definições para cada uma das experiências identificadas pelo seu `UNIQUE_ID` dentro de uma tag hardware. Cada uma das experiências contem:

- tag *rs232* que define as características da conexão à porta série e uma lista de portas, separadas por vírgula, onde esse hardware deve ser procurado;
- tag *parameters* onde são definidos os parâmetros de entrada e saída para a experiência e as funções transferência a aplicar em cada parâmetro;
- tag *channels* onde são definidos os canais de recepção de dados e as suas funções transferência;
- tag *timeout* com os tempos de espera até declarar falha para cada instrução;
- tag *errors* com os códigos e mensagens de erro;

```

<?xml version="1.0" encoding="iso-8859-1" ?>
<hardware id="UNIQUE_ID" num_channels="2" minfrequency="10Hz" maxFrequency="1000Hz">
  <rs232 baud="19200" stopbits="1" paritybits="1" numbits="8" ports_restrict="1,2,3,4" />
  <parameters>
    <parameter output="####" input="####.##" maxvalue="100" minvalue="1" order="1">
      <transfer_function type="output">
        <linear>
          <param weight="1" center="2" />
        </linear>
        <power>
          <param weight="0.5" center="3" power="2" />
          <param weight="0.25" center="2" power="2" />
        </power>
        <exponential>
          <param weight="1" center="0" coefficient="1" />
        </exponential>
        <logarithm>
          <param weight="2" center="10" coefficient="100" />
        </logarithm>
        <sin>
          <param weight="50" coefficient="1" delta="3.1416" />
        </sin>
        <tg>
          <param weight="1" coefficient="3" delta="0" />
        </tg>
      </transfer_function>
      <transfer_function type="input">
        (...)
      </transfer_function>
    </parameter>
  </parameters>
  <channels>
    <channel format="###" order="1">
      <transfer_function>
        (...)
      </transfer_function>
    </channel>
  </channels>
  <timeout>
    <default_timeout time="45" />
    <id time="10" />
    <cfg time="5" />
    <cur time="5" />
    <str time="45" />
    <dat_bin time="12" />
    <dat_no_data time="120" />
    <bin_no_data time="120" />
    <stp time="20" />
    <rst time="20" />
    <hardware_died time="3600" />
  </timeout>
  <errors>
    <error code="0" key="OPS" message="Something wrong happened !" />
    <error code="1" key="SENSOR" message="Sensor has failed." />
    <error code="2" key="OUT" message="Value out of range..." />
  </errors>
</hardware>

```

FUNÇÃO DE TRANSFERÊNCIA

A função de transferência é calculada usando uma série de funções algébricas generalistas que, após preenchidos os seus parâmetros e serem conjugadas, deve satisfazer a função pretendida.

O valor de x é obtido a partir do hardware e os parâmetros a , b e c são definidos no ficheiro XML de configuração.

A expressão da função transferência deve ser do tipo:

$$f(x) = \sum [a \cdot x - b] + \sum [a \cdot (x - b)^c] + \sum [a \cdot e^{c \cdot (x - b)}] + \sum [a \cdot \log_e c \cdot (x - b)] + \sum [a \cdot \sin(c \cdot x - b)] + \sum [a \cdot \text{tg}(c \cdot x - b)]$$

em que cada parâmetro de cada função algébrica é representado por:

função	fórmula	xml tag	a tag	b tag	c tag
Linear	$a \cdot x - b$	linear	weight	center	---
Potência	$a \cdot (x - b)^c$	power	weight	center	power
Exponencial	$a \cdot e^{c \cdot (x - b)}$	exponential	weight	center	coeficient
Logaritmo	$a \cdot \log_e [c \cdot (x - b)]$	logarithm	weight	center	coeficient
Seno	$a \cdot \sin(c \cdot x - b)$	sin	weight	delta	coeficient
Tangente	$a \cdot \text{tg}(c \cdot x - b)$	tg	weight	delta	coeficient