# Comunicação com o C<sup>4</sup>D – versão com Teensy

Versão 1.03

do Lago, Claudimir Lucio

Comunicação com o C4D – versão com Teensy

São Paulo, 2011, 2014

Copyright © 2011 by Claudimir Lucio do Lago

# **Índice Geral**

Introdução	3
Comunicação com o Injetor	4
Connect (UART)	
Program Press & Hold	
Run	
Halt	4
Status	5
Comunicação com o Detector	6
Connect (UART)	
Set (UART)	
Get (UART)	
Get (I <sup>2</sup> C)	
Zero	8
Comunicação com o Marcador Térmico	9
Program	9
Sync	
Run	9
Halt	
Test	
Status	

# Introdução

A comunicação se dá através da porta serial (USB) entre o computador e a microcontroladora ou via I<sup>2</sup>C.

As *strings* de comando são formadas por caracteres ASCII imprimíveis entre 33 e 126. Letras maiúsculas e minúsculas são caracteres distintos. O primeiro caractere da *string* de comando (posição 0) define o módulo destinatário. O segundo caractere da *string* de comando (posição 1) define o módulo que fez a solicitação (remetente). A resposta será dada usando-se este caractere como o destinatário da mensagem de retorno. O terceiro caractere da *string* de comando define o comando propriamente dito, o qual determina se o restante da *string* deve ou não conter informações adicionais relevantes. Cada declaração sempre é encerrada com um ";" (#59). Uma string de comando deve possuir no máximo 32 caracteres, incluindo o ";".

No caso particular da comunicação via I<sup>2</sup>C, a mensagem de retorno pode ter formato livre, pois cabe ao módulo solicitante (mestre) interpretar diretamente a resposta em função do comando enviado. Esta liberdade permite reduzir a carga no barramento I<sup>2</sup>C, pois valores numéricos podem ser enviados no formato nativo ao invés de ASCII.

No caso particular da comunicação via UART, caracteres abaixo de espaço (#32) são suprimidos na composição da *string*. Assim, um campo numérico, quando necessário, deve conter '0' ao invés de espaços nos dígitos à esquerda. A terminação '\0' (caractere nulo) é necessária em linguagem C para compor a string e deve ser computada no comprimento total da mesma:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
																														;	\0

# Comunicação com o Injetor

Nos comandos a seguir, x é sempre o remetente.

# **Connect (UART)**

Estabelece ou interrompe conexão com o injetor. Enquanto não há conexão, o injetor envia de tempos em tempos a mensagem de broadcast.

string de comando:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
i	х	X	у	;	0/																										

string de retorno:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
x	i	X	у	;	\0																										

y − 'N' ativa a conexão; 'F' desativa a conexão.

# **Program Press & Hold**

Estabelece os tempos de pressurização e de retenção.

string de comando:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
i	x	P	p	p	p	p	p	p	p	p	h	h	h	h	h	h	h	h	;	\0											

p – tempo de pressurização, de 00 000 000 a 99 999 999 milissegundos.

*h* – tempo de retenção, de 00 000 000 a 99 999 999 milissegundos.

#### Run

Realiza uma injeção como programado anteriormente.

string de comando:

(	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
i	x	R	;	\0																											

### Halt

Aborta a injeção.

string de comando:

		0																													
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
i	x	Н	;	\0																											

# **Status**

Retorna a condição de funcionamento do injetor.

# string de comando:

		<u> </u>																													
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
i	x	S	;	\0																											

# string de retorno:

0	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
x		i	S	y	;	\0																										

y – retorna '1' se estiver injetando ou '0' em caso contrário.

# Comunicação com o Detector

Os quatro conversores A/D são identificados de 0 a 3. Não há informação sobre qual a finalidade de cada um deles. No entanto, o uso padrão deve ser conversores 0 e 1 para detectores, 2 para o sensor de pressão e 3 para a corrente eletroforética.

Nos comandos a seguir, x é sempre o remetente.

# **Connect (UART)**

Estabelece ou interrompe conexão com o detector. Enquanto não há conexão, o detector envia de tempos em tempos a mensagem de broadcast.

#### string de comando:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
d	x	X	y	;	\0																										

#### string de retorno:

0 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$x \mid d$	X	y	;	0/																										

y - 'N' ativa a conexão; 'F' desativa a conexão.

# Set (UART)

Informa qual o tempo desde a última reinicialização do cronômetro do módulo detector e uma leitura dos conversores A/D. Este é o comando a ser utilizado por sistemas que façam aquisição de dados via porta UART do equipamento.

### string de comando:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
d	х	S	y	t	$a_0$	$a_1$	$a_2$	$a_3$	;	\0																					

y – indica o caractere a ser utilizado como separador. 't' deve ser utilizado para tab e 's'
 para espaço. O uso de 'f' indica que a resposta deve ser formatada (ver comando Get - UART). Qualquer outro caractere imprimível pode ser também utilizado.

t-1' indica que o *time stamp* deve ser incluído na resposta não formatada.

 $a_x$  – '1' indica que a resposta do ADC x deve ser incluído na resposta.

# Get (UART)

Informa qual o tempo desde a última reinicialização do cronômetro do módulo detector e uma leitura dos conversores A/D. Este é o comando a ser utilizado por sistemas que façam aquisição de dados via porta UART do equipamento.

string de comando:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
d	x	G	у	;	\0																										

y – 'r' inicia o envio contínuo de dados pela porta UART; 'h' interrompe o envio contínuo de dados; 'w' espera pelo disparo externo, zera o cronômetro e inicia o envio contínuo de dados pela porta UART; 't' espera pelo disparo externo, zera o cronômetro e inicia o envio contínuo de dados pela porta UART até que o sinal de *stop* externo seja recebido pela microcontroladora (A mensagem "xdH;" é gerada ao final). Qualquer outro valor para y irá resultar no envio de um só dado como resposta.

string de retorno para opção 'f' do comando Set:

	$x \mid d \mid G \mid t \mid a_w \mid a_x \mid $	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
$ x  d  G  t  t  t  t  t  t  t  t  a_w  a_w  a_w  a_w  a_w  a_w  a_x  $		x	d	G	t	t	t	t	t	t	t	$a_w$	$a_x$													

24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
$a_{y}$	$a_z$	;	\0																					

t – tempo no qual as leituras dos conversores foram realizadas (em milissegundos). Número inteiro positivo de 7 dígitos.

 $a_w$  a  $a_z$  – leituras dos conversores A/D. Número inteiro entre 0 e 4 194 304.

string de retorno para opção não formatada do comando Set:

		<del>-</del>					1		- I	3													~ -	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
t	t	t	t	t	t	t	s	$a_w$	S	$a_x$	$a_x$	$a_x$	$a_x$	$a_x$	$ a_x $	$a_x$								

24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
s	$a_y$	$a_{y}$	$a_{y}$	$a_{y}$	$a_{y}$	$a_{y}$	$a_{y}$	s	$a_z$	lf														

- t tempo no qual as leituras dos conversores foram realizadas (em milissegundos).
  Número inteiro positivo de 7 dígitos. Pode ou não ser incluído na resposta (ver comando Set).
- s caractere escolhido como separador no comando Set.
- $a_w$  a  $a_z$  leituras dos conversores A/D. Número inteiro entre 0 e 4 194 304. Pode ou não ser incluído na resposta (ver comando Set).

lf − *line feed* ou final de linha.

# Get (I<sup>2</sup>C)

Informa qual o tempo desde a última reinicialização do cronômetro do módulo detector e uma leitura dos conversores A/D. Este comando deve ser utilizado somente por módulos instalados no barramento I<sup>2</sup>C interno do equipamento.

#### string de comando:

0	1	2	3
d	G	;	\0

string de retorno:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
t	t	t	t	$a_0$	$a_0$	$a_0$	$a_0$	$a_{I}$	$a_1$	$a_1$	$a_1$	$a_2$	$a_2$	$a_2$	$a_2$	$a_3$	$a_3$	$a_3$	$a_3$

Todos os valores são do tipo unsigned long (inteiro sem sinal de 4 bytes).

*t* – tempo no qual as leituras dos conversores foram realizadas (em milissegundos).

 $a_w$  – leituras dos conversores A/D. Número inteiro entre 0 e 4.194.304.

## **Zero**

Zera o cronômetro.

### string de comando:

~ -		ອ ~					•																								
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
d	x	$\mathbf{Z}$	;	\0																											

# Comunicação com o Marcador Térmico

Nos comandos a seguir, x é sempre o remetente.

# **Program**

Programa o marcador térmico para uso subsequente.

string de comando:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
p	x	P	w	w	w	w	p	p	p	d	d	d	d	d	d	d	c	c	c	c	c	n	n	;	\0						

w – largura do pulso (ms).

p – potência do pulso (0 a 100).

d – tempo de aguardo (ms) antes do primeiro pulso.

*c* − período (ms) de cada ciclo.

n – número de ciclos.

# **Sync**

Sincroniza o disparo do marcador com o disparo de aquisição de dados do detector. Quando disparado, realiza um teste do marcador antes de iniciar o programa.

string de comando:

501		<b>5</b> ~					•																								
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
p	x	W	y	;	\0																										

y – 'N' ativa a sincronização; 'F' desativa a sincronização.

#### Run

Executa o programa do marcador térmico. Desativa automaticamente o modo sincronizado e realiza um teste do marcador antes de iniciar a corrida.

string de comando:

_		_	_																													
(	)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	5	х	R	;	\0																											

#### Halt

Aborta a execução do programa do marcador térmico. Desativa automaticamente o modo sincronizado.

string de comando:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	x	Н	;	\0																											

#### **Test**

Executa um teste de integridade dos dispositivos de potência do marcador térmico. Desativa automaticamente o modo sincronizado e aborta a execução do programa, caso este esteja em curso. O resultado do teste pode ser lido pelo comando *Status*.

string de comando:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
p	x	T	;	\0																											

#### **Status**

Solicita o status do marcador térmico. Os estados do filamento e transistor correspondem àquele de quando foi executado o último comando *Test* ou imediatamente antes da execução do programa, o que for mais recente.

string de comando:

		<u> </u>																													
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
p	x	S	;	0/																											

string de retorno:

		<b>o</b> '					-																								
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
p	x	S	f	t	m	w	x	x	n	n	;	\0																			

f – '1' se o filamento estiver queimado; '0' em caso contrário.

t-'1' se o transistor estiver queimado; '0' em caso contrário.

m-1'se o programa estiver em execução; '0' em caso contrário.

w-'1' se o marcador estiver sincronizado com o detector; '0' em caso contrário.

xx – número de ciclos que faltam ser executados.

*nn* – número total de ciclos programados.