

Física Experimental VI – 4300314

1º Semestre de 2017

**Instituto de Física
Universidade de São Paulo**

Professor: Antonio Domingues dos Santos

E-mail: adsantos@if.usp.br

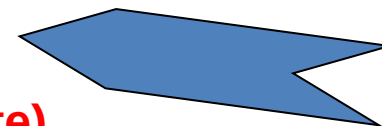
Fone: 3091.6886

Comunicação de dados

Instrumentação $\leftrightarrow$ computador

OPÇÕES:

- 1) Porta paralela do computador
- 2) Porta serial ou USB (mais recente)
- 3) Outras soluções específicas:
 - a) GPIB (HPIB, IEEE788)
 - b) CAMAC
- 4) Rede internet (novo !)



Instrumentação ↔ computador



Exige comandos de acesso às portas lógicas do computador, como:

Input(porta) e **output(porta, valor)**

Porta paralela do computador

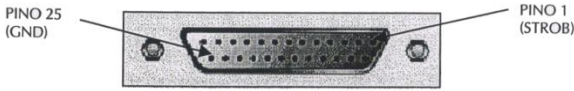


Figura 17.1 - Conector DB 25pinos Fêmea.

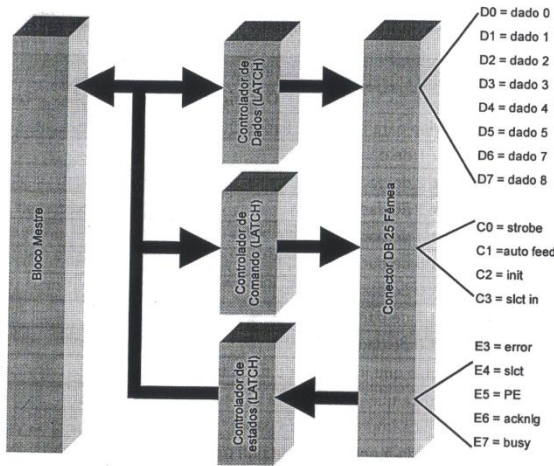


Figura 17.2 - Interface Paralela PC.

Veja a representação da interface paralela do PC em um diagrama de blocos, conforme mostrado na figura 17.2.

Pelo diagrama em blocos da interface paralela (representada na figura 17.2), podemos definir alguns pontos importantes:

- Os sinais D0 até D7 - formam uma barra unidirecional (porta paralela SPP) de saída de dados;
- Os sinais identificados por C0 até C3 formam uma barra unidirecional (porta paralela SPP) de saída de comandos;
- E por fim, os sinais E3 até E7 representam uma barra unidirecional (porta paralela SPP) de entrada do estado de um periférico externo.

As características elétricas da porta paralela (níveis de tensão e corrente) são as mesmas de um dispositivo TTL (lógica transistor transistor) padrão. Portanto, um sinal de nível baixo (nível lógico 0) deve corresponder a uma tensão em torno de 0V e +0,8V e o sinal de nível alto (nível lógico 1) deve estar dentro da faixa de +2,4V até +5V.

A descrição dos pinos do conector DB 25 da porta paralela é dada na tabela de especificação da interface paralela:

DB 25 Fêmea			
Pino	Sinal	Descrição	Direção
1	-strobe	Pulso para leitura de dados	Saída
2	data 0	Bit 1 de dado	Saída
3	data 1	Bit 2 de dado	Saída
4	data 2	Bit 3 de dado	Saída
5	data 3	Bit 4 de dado	Saída
6	data 4	Bit 5 de dado	Saída
7	data 5	Bit 6 de dado	Saída
8	data 6	Bit 7 de dado	Saída
9	data 7	Bit 8 de dado	Saída
10	acknlg	Pulso que indica que o dado foi recebido	Entrada
11	-busy	Indica que a impressora não pode receber dados	Entrada
12	PE	Indica impressora sem papel	Entrada
13	slct	Ligado a um resistor de pull up	Entrada
14	auto feed	Quando em nível 0, o papel é alimentado em uma linha depois da impressão	Saída
15	error	Este sinal vai para nível 0 para sinalizar erro	Entrada
16	init	Quando em nível 0, o controlador da impressora é "resetado" e o buffer é apagado	Saída
17	slct in	Código DC1/DC3 é válido somente para nível 1	Saída
25	GND	Terra do sistema	*****

Tabela 17.1 - Pinagem conector DB25 Fêmea.

O acesso a cada um dos barramentos aqui descritos é feito em software por instruções de entrada (*inportb()*) e saída (*outportb()*) pelos endereços de I/O:

Interface	Endereço
Controladora de dados	378 H
Controladora de comandos	37A H
Controladora de estados	379 H

Tabela 17.2 - Endereço da Controladora LPT1.

Relacionamos em seguida as opções de acesso à interface:

Operação de escrita na controladora de dados - instrução "out" no endereço 378 H.

Referência no microprocessador	D7	D6	D5	D4	D3	D2	D1	D0
Sinal relacionado no DB 25	D7	D6	D5	D4	D3	D2	D1	D0
Pino do conector db 25	9	8	7	6	5	4	3	2

Tabela 17.3 - Relação Pino/Sinal da Controladora de Dados.

Operação de leitura na controladora de estados - instrução "in" no endereço 379 H.

Referência no microprocessador	D7	D6	D5	D4	D3	D2	D1	D0
Sinal relacionado no DB 25	-E7	E6	E5	E4	E3	*	*	*
Pino do conector db 25	11	10	12	13	15	*	*	*

Tabela 17.5 - Relação Pino/Sinal da Controladora de Estados.

Comunicação de dados

Instrumentação ↔ computador

Um exemplo de comunicação através da porta paralela do computador

/* Rotina de transferencia de dados atraves da Porta Paralela */

Le_byte(unsigned int canal)

unsigned int canal;

```
{
    unsigned int cont, temp, stale, data, msb, lsb, sh, pronto; /* 4 bits cada */
    stale=8; /* 00001000 -> 378h bit 3 bit "start + ALE" */
    sh=128; /* 10000000 -> 378h inicializa 74157 para lsb ou 0x80 */
    pronto=16; /* 00010000 -> 378h output enable */
```

```
    outp(0x378,canal); /* define canal */
    outp(0x378,canal+stale); /* habilita start e ALE */
    outp(0x378,canal);
    for(cont=0; cont<30000;cont++){
        delay(aux_delay);
        if(cont >= (3000/(aux_delay+1))) return(-1);
        if((inp(0x379) & 8) == 8) break; /* 01110111 detetor de EOC */
    }
```

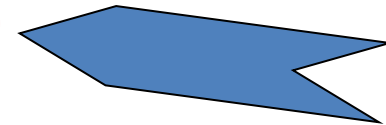
```
    outp(0x378,pronto); /* disponibiliza dado para MSB */
    msb = inp(0x379)^0x8F; /* adquire e ajusta bit 3 invertido */
    outp(0x378,pronto+sh); /* inverte a chave do 74157 via 378h */
    temp = inp(0x379)^0x8F; /* adquire e ajusta bit 3 invertido */
    lsb = temp>>4; /* move os dados 4 bits para direita */
    data = msb+lsb; /* soma */
    return(data); /* devolve valor medido */
}
```

Comunicação de dados

Instrumentação $\leftrightarrow$ computador

OPÇÕES:

- 1) Porta paralela do computador
- 2) Porta serial ou USB (mais recente)
- 3) Outras soluções específicas:
 - a) GPIB (HPIB, IEEE788)
 - b) CAMAC
- 4) Rede internet (?)



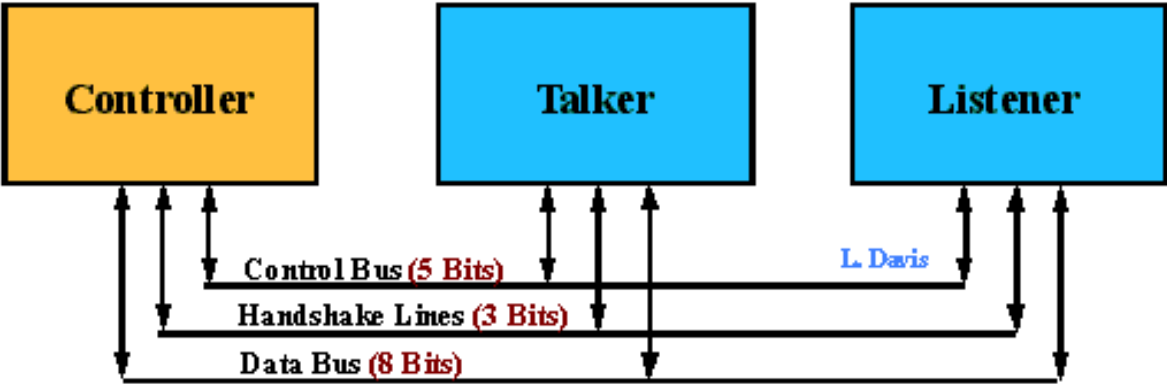
Comunicação de dados
Instrumentação ↔ computador

GPIB

(General Purpose
Interface Board)

Nome antigo: HPIB

GPIB Description [IEEE488]



GPIB System

Comunicação de dados

Instrumentação ↔ computador

GPIB

(General Purpose Interface Board)

A interface GPIB permite a transferência serializada de BYTES de 8 bits (paralelos). Para isto, ela possui:

5 linhas de controle

3 linhas de protocolo (handshake lines)

8 linhas bi-direcionais de dados

Em um cabo de 24 fios.

Com taxa máxima de transferência de 1Mb/s.

Cada equipamento (ou placa GPIB) pode assumir qualquer das seguintes funções:

1. Controller (somente 1 em cada bus)
2. Talker (somente 1 a cada vez)
3. Listener (até 15 equipamentos)

Cada equipamento tem um endereço que é usado pelo controlador para coordenar as ações a serem executadas.

Além das 3 funções básicas (controller, talker e listener) o sistema incorpora outras atividades operacionais, como:

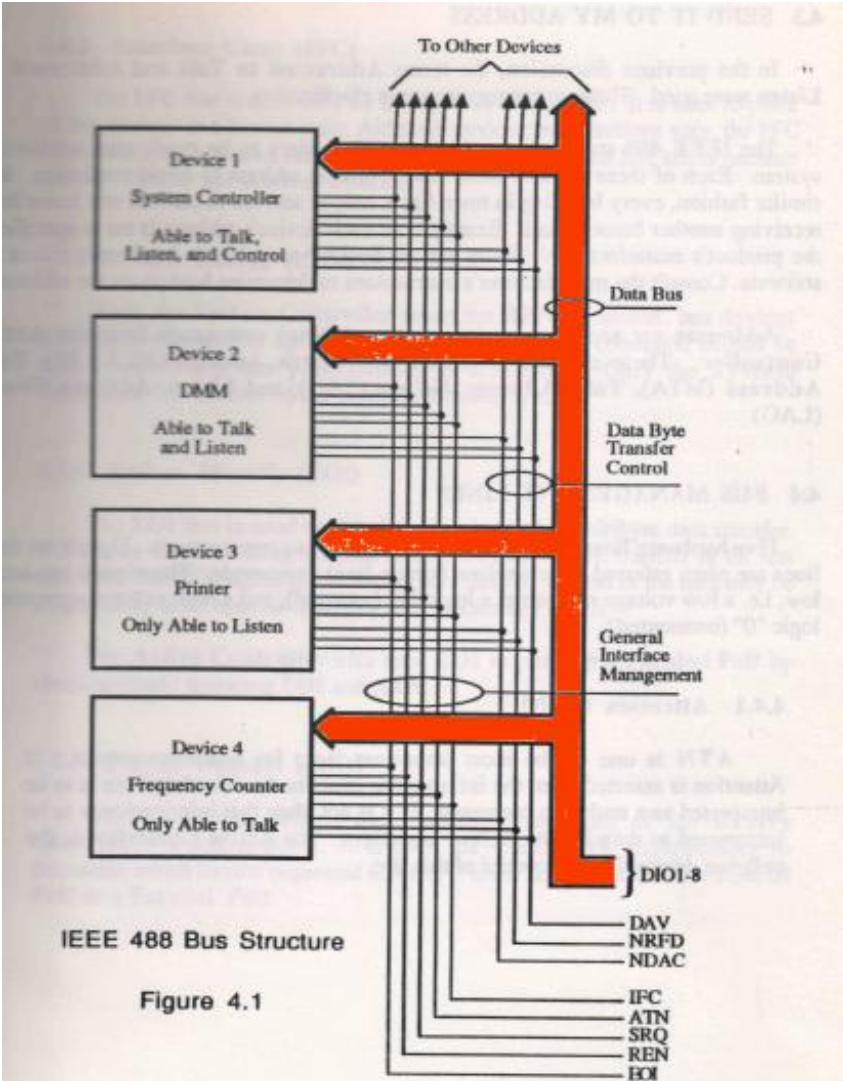
- serial poll,
- parallel poll,
- secondary talk and listen addresses,
- remote/local capability,
- device clear (trigger)

Comunicação de dados

Instrumentação ↔ computador

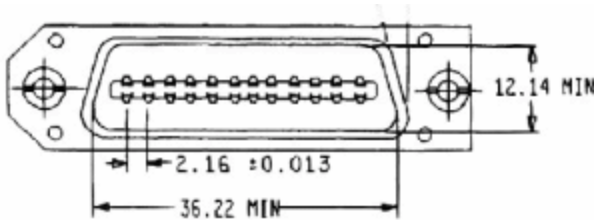
GPIB

(General Purpose Interface Board)



Comunicação de dados

Instrumentação ↔ computador



GPIB

(General Purpose Interface Board)

Linhas de dados

Pin #	Signal Names	Signal Description
1	DIO1	Data Input/Output Bit 1
2	DIO2	Data Input/Output Bit 2
3	DIO3	Data Input/Output Bit 3
4	DIO4	Data Input/Output Bit 4
5	EIO	End-Or-Identify

Linhas de protocolo

6	DAV	Data Valid
7	NRFD	Not Ready For Data
8	NDAC	Not Data Accepted

Linhas de controle

9	IFC	Interface Clear
10	SRQ	Service Request
11	ATN	Attention
12	Shield	Chassis Ground

Pin #	Signal Names	Signal Description
13	DIO5	Data Input/Output Bit 5
14	DIO6	Data Input/Output Bit 6
15	DIO7	Data Input/Output Bit 7
16	DIO8	Data Input/Output Bit 8
17	REN	Remote Enable
18	Shield	Ground (DAV)
19	Shield	Ground (NRFD)
20	Shield	Ground (NDAC)
21	Shield	Ground (IFC)
22	Shield	Ground (SRQ)
23	Shield	Ground (ATN)
24	Single GND	Single Ground

Comunicação de dados

Instrumentação ↔ computador

Comandos da placa controladora

GPIB

(General Purpose Interface Board)

Table 1-1. List of NI-488 Device-Level Functions

Function	Purpose
ibask	Return information about software configuration parameters
ibbna	Change the access board of a device
ibclr	Clear a specific device
ibconfig	Change the software configuration parameters
ibdev	Open and initialize a device
ibeos	Configure the end-of-string (EOS) termination mode or character
ibeot	Enable or disable the automatic assertion of the GPIB EOI line at the end of write I/O operations
iblines	Return the status of the eight GPIB control lines
ibln	Check for the presence of a device on the bus
ibloc	Go to local
ibonl	Place the device online or offline
ibpad	Change the primary address
ibpct	Pass control to another GPIB device with Controller capability
ibppc	Parallel poll configure
ibrd	Read data from a device into a user buffer
ibrda	Read data asynchronously from a device into a user buffer
ibrdf	Read data from a device into a file
ibrdi	Read data from a device into a user integer buffer
ibrdia	Read data asynchronously from a device into a user integer buffer
ibrpp	Conduct a parallel poll
ibrsp	Conduct a serial poll
ibsad	Change or disable the secondary address
ibstop	Abort asynchronous I/O operation

Table 1-1. List of NI-488 Device-Level Functions (Continued)

Function	Purpose
ibtmo	Change or disable the I/O timeout period
ibtrg	Trigger selected device
ibwait	Wait for GPIB events
ibwrt	Write data to a device from a user buffer
ibwrta	Write data asynchronously to a device from a user buffer
ibwrtf	Write data to a device from a file
ibwrti	Write data to a device from a user integer buffer
ibwrtia	Write data asynchronously to a device from a user integer buffer

Table 1-2. List of NI-488 Board-Level Functions

Function	Purpose
ibask	Return information about software configuration parameters
ibcac	Become Active Controller
ibcmd	Send GPIB commands
ibcmda	Send GPIB commands asynchronously
ibconfig	Change the software configuration parameters
ibdma	Enable or disable DMA
ibeos	Configure the end-of-string (EOS) termination mode or character
ibeot	Enable or disable the automatic assertion of the GPIB EOI line at the end of write I/O operations
ibevent	Return the oldest event
ibfind	Open and initialize a GPIB board
ibgts	Go from Active Controller to Standby
ibist	Set or clear the board individual status bit for parallel polls
iblines	Return the status of the eight GPIB control lines
ibln	Check for the presence of a device on the bus
ibloc	Go to local
ibonl	Place the interface board online or offline
ibpad	Change the primary address

(continues)

Table 1-2. List of NI-488 Board-Level Functions (Continued)

Function	Purpose
ibppc	Parallel poll configure
ibrd	Read data from a device into a user buffer
ibrda	Read data asynchronously from a device into a user buffer
ibrdf	Read data from a device into a file
ibrdi	Read data from a device into a user integer buffer
ibrdia	Read data asynchronously from a device into a user integer buffer
ibrpp	Conduct a parallel poll
ibrsc	Request or release system control
ibrsv	Request service and change the serial poll status byte
ibsad	Change or disable the secondary address
ibsic	Assert interface clear
ibstre	Set or clear the Remote Enable (REN) line
ibsrq	Request an SRQ "interrupt routine"
ibstop	Abort asynchronous I/O operation
ibtmo	Change or disable the I/O timeout period
ibtrap	Configure the Applications Monitor
ibwait	Wait for GPIB events
ibwrt	Write data to a device from a user buffer
ibwrta	Write data asynchronously to a device from a user buffer
ibwrtf	Write data to a device from a file
ibwrti	Write data to a device from a user integer buffer
ibwrtia	Write data asynchronously to a device from a user integer buffer

Comandos Principais:

Ibrd (leitura), ibwrt (escrita), ibconfig (configuração), ibrsp (serial poll - sondagem)

Comunicação de dados

GPIB

Instrumentação $\longleftrightarrow$ computador

/* Rotinas de acesso a instrumentacao GPIB */

init_gpib(){

```
    libsic(0);
    libsre(0,1);
    libeos(0,0);
    libeot(0,1);
    libdma(0,0);
    libclr(0,equip_1);
    libclr(0,equip_2);
    libclr(0,equip_3);
}
```

send_gpib(mensagem, equip, spoll, espera)

int equip, spoll;

unsigned int espera;

char mensagem[32];

```
{
    int count;
    char *pt_mensagem= mensagem;
    spoll_gpib(equip,spoll,500);
    count= strlen(mensagem);
    libesc(0,equip,pt_mensagem,count);
    delay(espera);
    spoll_gpib(equip,spoll,500);
```

}

int spoll_gpib(equip, spoll, max_count)

int equip, spoll, max_count;

```
{
    int status, l_poll;
    char poll;
    char far *pt_poll=&poll;
    l_poll= 0;
```

if(spoll == 0) return(0);

do {

```
    delay(1);
    l_poll+= 1;
    status= 1;
    liblsp(0,equip,pt_poll);
    status*= (poll & spoll);
    if(l_poll > max_count) return(-1);
} while (status == 0);
```

return(0);

}

Comunicação de dados

GPIB

Instrumentação $\leftrightarrow$ computador

```
float le_f_gpib( mensagem, equip, spoll, amostragem)
```

```
int equip, spoll, amostragem;
```

```
char mensagem[32];
```

```
{
```

```
    char equip_buf[64];
```

```
    char *pt_buf= equip_buf;
```

```
    char *pt_mensagem= mensagem;
```

```
    float sinal, aux;
```

```
    int len, count, t;
```

```
    sinal= 0.0;
```

```
    count= strlen(mensagem);
```

```
    for(t=1;t<=amostragem;t++) {
```

```
        libesc(0,equip,pt_mensagem,count);
```

```
        spoll_gpib(equip,spoll,500);
```

```
        libler(0,equip,pt_buf,&len);
```

```
        sscanf(equip_buf," %g",&aux);
```

```
        sinal+= aux;
```

```
    }
```

```
    sinal/= amostragem;
```

```
    return(sinal);
```

```
}
```

```
int le_i_gpib( mensagem, equip, spoll, amostr)
```

```
int equip, spoll, amostr;
```

```
char mensagem[32];
```

```
{
```

```
    char equip_buf[64];
```

```
    char *pt_buf= equip_buf;
```

```
    char *pt_mensagem= mensagem;
```

```
    long int sinal;
```

```
    int len, count, t, aux;
```

```
    sinal= 0L;
```

```
    count= strlen(mensagem);
```

```
    for(t=1;t<=amostr;t++) {
```

```
        libesc(0,equip,pt_mensagem,count);
```

```
        spoll_gpib(equip,spoll,500);
```

```
        libler(0,equip,pt_buf,&len);
```

```
        sscanf(equip_buf," %d",&aux);
```

```
        sinal+= aux;
```

```
        delay(10);
```

```
    }
```

```
    aux= sinal / amostr;
```

```
    return(aux);
```

```
}
```

Comunicação de dados

GPIB

Instrumentação $\leftrightarrow$ computador

Principais funções de um programa de comunicação:

- Inicialização dos equipamentos
- sincronização das operações
- envio de comandos
- recepção de dados (e informações operacionais)

Todas estas funções devem ser particularizadas a cada equipamento!

Comunicação de dados

Instrumentação ↔ computador

GPIB

(General Purpose Interface Board)

Comandos dos equipamentos

(exemplo: detecção síncrona)

KP:	KEYPRESS reads and resets an internal flag that is set whenever a front-panel key is pressed. If the flag is in the set state when KP is applied, a "1" is reported to the computer, advising it that a keypress has occurred. If the flag is in the reset state when KP is applied, a "0" is reported to the computer. KP always leaves the flag in the reset state so that the next keypress can be detected. The command is well suited to use in programs where the intent is to initiate an external event from the front panel of the Model 5210 while it is operating under computer control.										
LF n:	The LINE FILTER command sets or reads the line filter selection. Sent with the operand, it sets the parameter. Sent without the operand, it reads it. The codes are: <table><tr><th>CODE</th><th>STATUS</th></tr><tr><td>0</td><td>OFF</td></tr><tr><td>1</td><td>2F ON (notch 2X line freq.)</td></tr><tr><td>2</td><td>F ON (notch line freq.)</td></tr><tr><td>3</td><td>BOTH FILTERS ON</td></tr></table>	CODE	STATUS	0	OFF	1	2F ON (notch 2X line freq.)	2	F ON (notch line freq.)	3	BOTH FILTERS ON
CODE	STATUS										
0	OFF										
1	2F ON (notch 2X line freq.)										
2	F ON (notch line freq.)										
3	BOTH FILTERS ON										
LR:	The LR (LOG RATIO) command reads the log of the ratio (X Output to ADC 1 Input) and reports the value 1000 times the log ratio to the host computer. The response range is -3000 to +2000 (log ratio of -3 to +2, that is, five decades).										
LTS n:	The LIGHTS command sets or reads the status of the front-panel lights. Sent with the operand, it sets the parameter. Sent without the operand, it reads it. The codes are: <table><tr><th>CODE</th><th>STATUS</th></tr><tr><td>0</td><td>LIGHTS OFF</td></tr><tr><td>1</td><td>LIGHTS ON</td></tr></table>	CODE	STATUS	0	LIGHTS OFF	1	LIGHTS ON				
CODE	STATUS										
0	LIGHTS OFF										
1	LIGHTS ON										
MAG:	When this command is applied, the Model 5210 reports the signal magnitude to the host computer in standard five-digit format (range ± 15000 , corresponding to ± 1.5 f.s.).										
MP:	This command causes the Model 5210 to report the magnitude and phase to the host computer. The magnitude is sent first (five-digit format), followed by the delimiter, and then the phase in millidegrees (± 180000). Note, however, that since the two least significant digits are always "0", the phase is reported with a resolution of ± 100 millidegrees.										
MSK n:	The SRQ MASK command set or reads the conditions that exist to cause a GPIB service request. Sent with the operand, it sets the mask. Sent without the operand, it reads it.										

Comunicação de dados
Instrumentação ↔ computador

GPIB
(General Purpose
Interface Board)

Comandos dos equipamentos
(exemplo: detecção síncrona)

LF n: The LINE FILTER command sets or reads the line filter selection. Sent with the operand, it sets the parameter. Sent without the operand, it reads it. The codes are:

CODE	STATUS
0	OFF
1	2F ON (notch 2X line freq.)
2	F ON (notch line freq.)
3	BOTH FILTERS ON

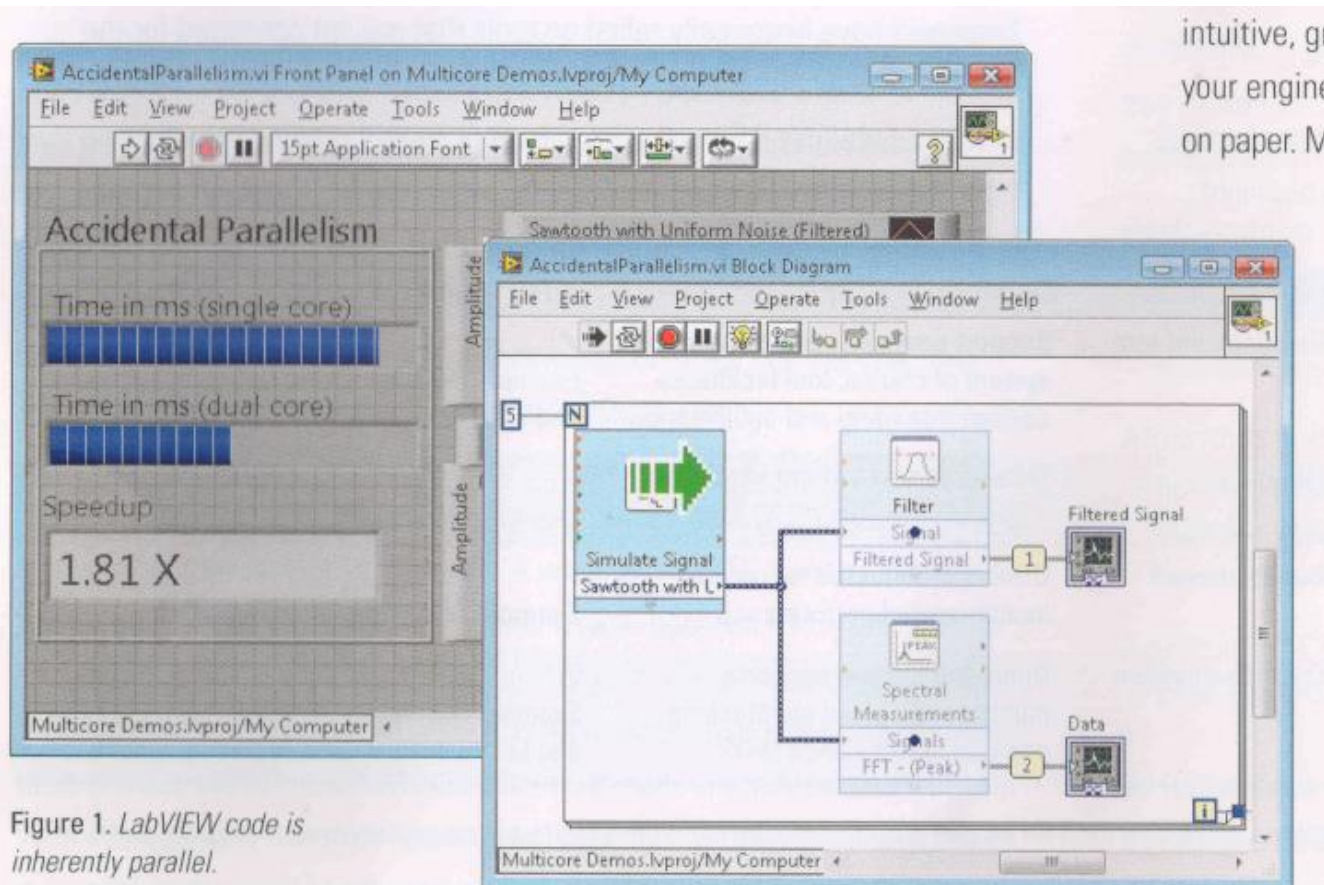
MAG: When this command is applied, the Model 5210 reports the signal magnitude to the host computer in standard five-digit format (range ± 15000 , corresponding to ± 1.5 f.s.).

Comunicação de dados

GPIB

Instrumentação $\leftrightarrow$ computador

**Software comercial Labview
(linguagem objeto)**

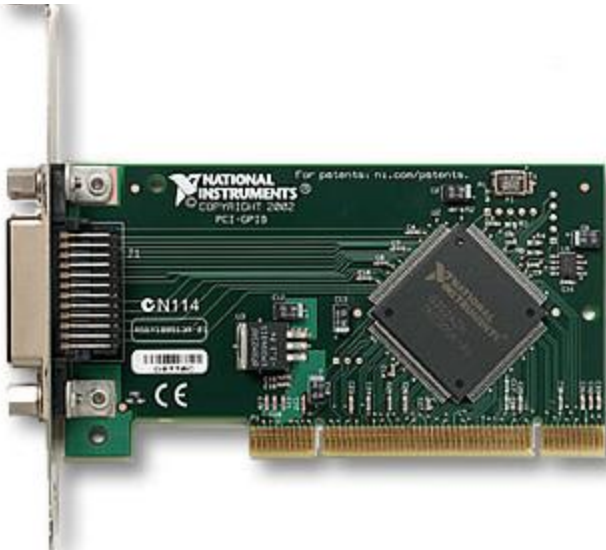


Drivers dos equipamentos fornecidos pelos respectivos fabricantes.

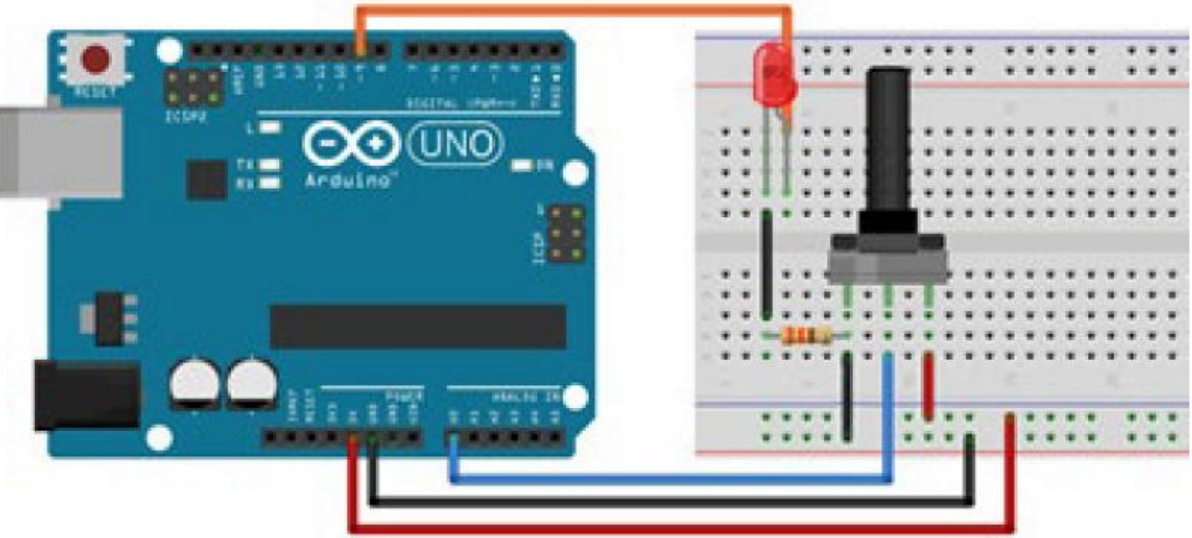
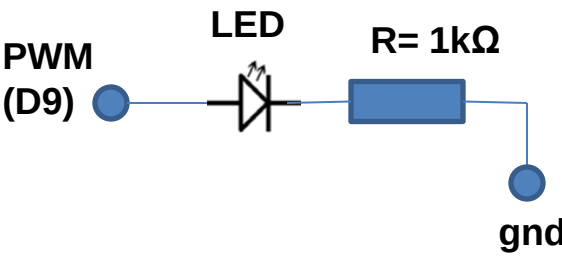
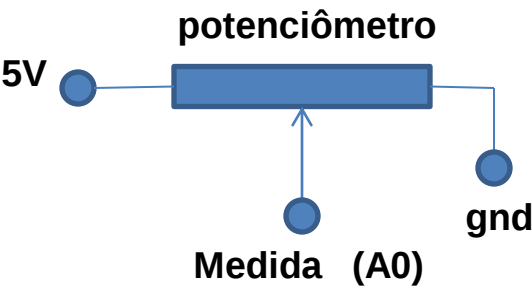
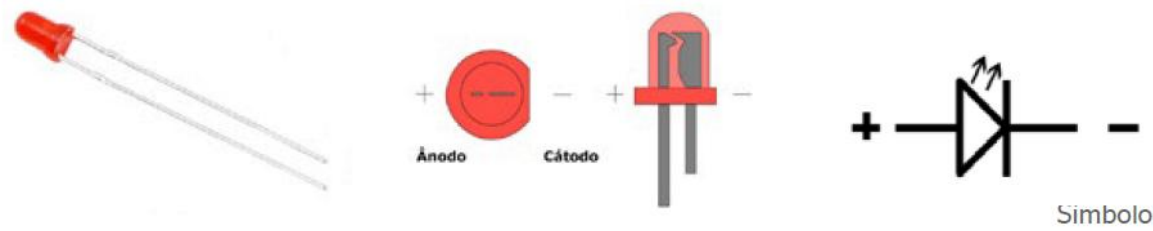
GPIB



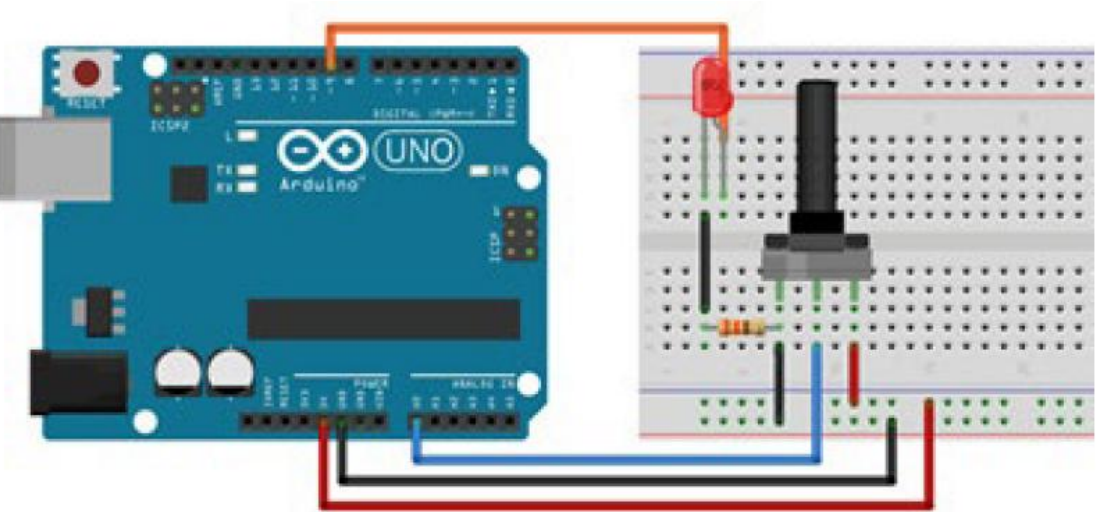
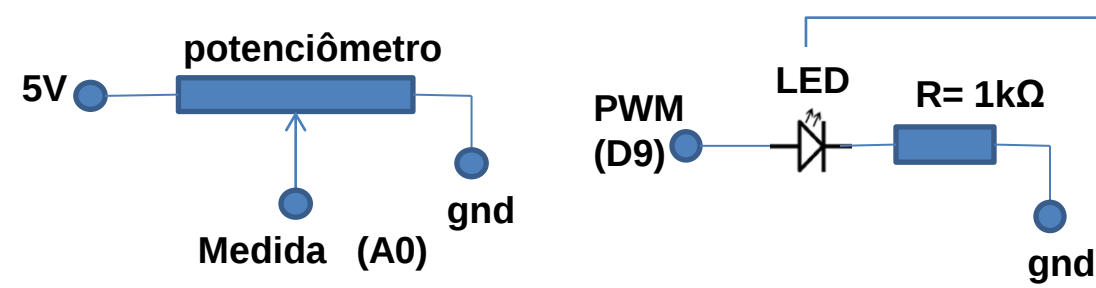
**Cartões de comunicação
no padrão GPIB**



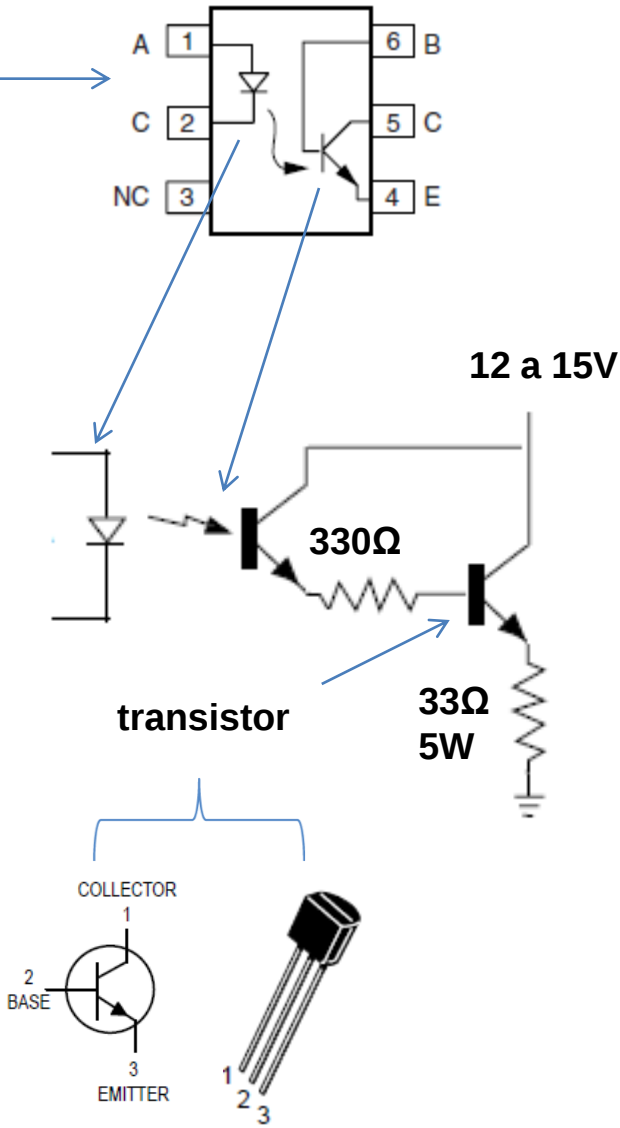
LED com PWM



Aquecedor de Água

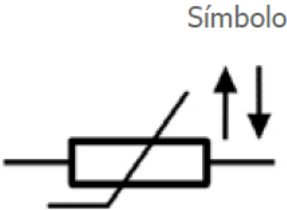


Acoplador ótico (4N37)

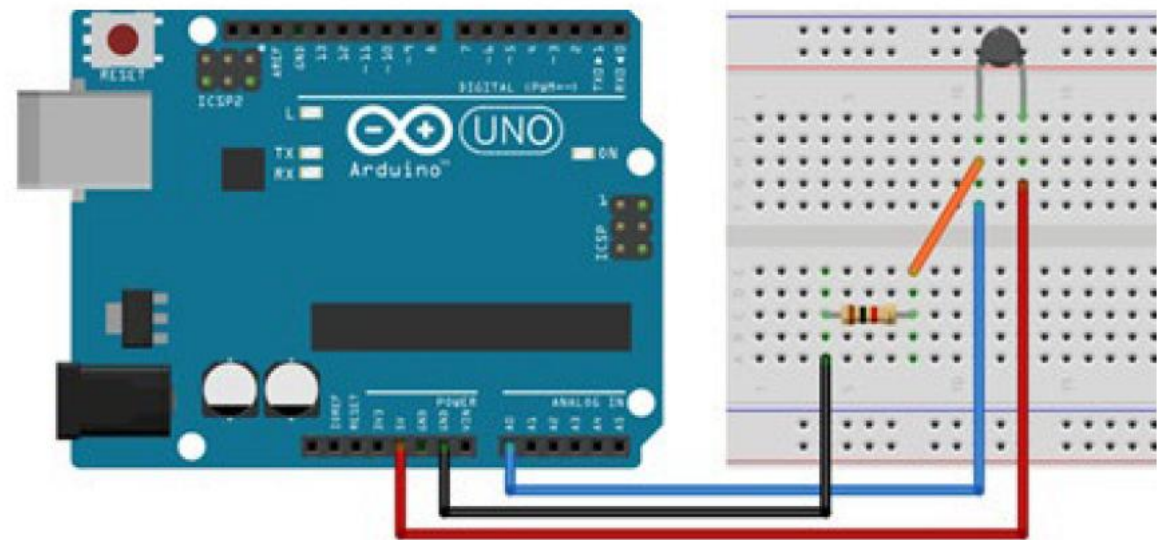


Arduino

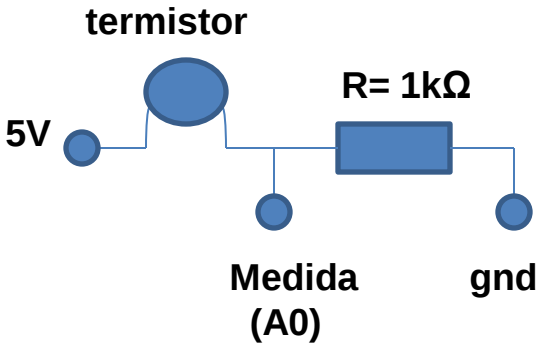
Termistor (NTC – 1kΩ)



O termistor NTC (do inglês *Negative Temperature Coefficient*) é um componente eletrônico semicondutor sensível à temperatura, utilizado para controle, medição ou polarização de circuitos eletrônicos. Possui um coeficiente de variação de resistência que varia negativamente conforme a temperatura aumenta, ou seja, a sua resistência elétrica diminui com o aumento da temperatura.



T [°C]	R_Nom [Ω]	R_Min [Ω]	R_Max [Ω]	ΔR/R_Nom [±%]	[±°C]
-55	59 147,00	48 631,00	69 664,00	17,80	2,70
-50	42 661,00	35 521,00	49 781,00	16,70	2,60
-45	31 088,00	26 207,00	35 969,00	15,70	2,50
-40	22 903,00	19 530,00	26 276,00	14,70	2,50
-35	17 062,00	14 700,00	19 406,00	13,80	2,40
-30	12 827,00	11 172,00	14 482,00	12,90	2,30
-25	9 746,00	8 572,00	10 920,00	12,00	2,20
-20	7 477,00	6 638,00	8 316,00	11,20	2,20
-15	5 790,00	5 106,00	6 394,00	10,40	2,10
-10	4 523,00	4 086,00	4 961,00	9,70	2,00
-5	3 564,00	3 246,00	3 883,00	8,90	1,90
0	2 832,00	2 599,00	3 065,00	8,20	1,80
5	2 267,00	2 096,00	2 438,00	7,50	1,70
10	1 829,00	1 703,00	1 956,00	6,80	1,60
15	1 486,00	1 393,00	1 578,00	6,30	1,50
20	1 215,00	1 146,00	1 283,00	5,60	1,40
25	1 000,00	950,00	1 050,00	5,00	1,30
30	828,20	781,60	874,70	5,60	1,50
35	689,90	647,20	732,60	6,20	1,70
40	577,80	539,10	616,60	6,70	1,90
45	486,60	461,40	521,80	7,20	2,10
50	411,80	380,00	443,70	7,70	2,40
55	350,20	321,40	379,00	8,20	2,60
60	299,20	273,20	325,20	8,70	2,80
65	256,70	233,20	280,20	9,20	3,00
70	221,20	200,00	242,40	9,60	3,30
75	191,40	172,20	210,60	10,00	3,50
80	166,20	148,80	183,60	10,50	3,80
85	144,80	129,10	160,60	10,90	4,00
90	126,70	112,40	140,90	11,30	4,30
95	111,20	98,22	124,10	11,70	4,50
100	97,87	86,10	109,60	12,00	4,80
105	86,43	75,72	97,14	12,40	5,00
110	76,66	66,79	86,31	12,80	5,30
115	67,99	59,09	76,90	13,10	5,60
120	60,66	52,42	68,89	13,40	5,90
125	54,07	46,63	61,52	13,80	

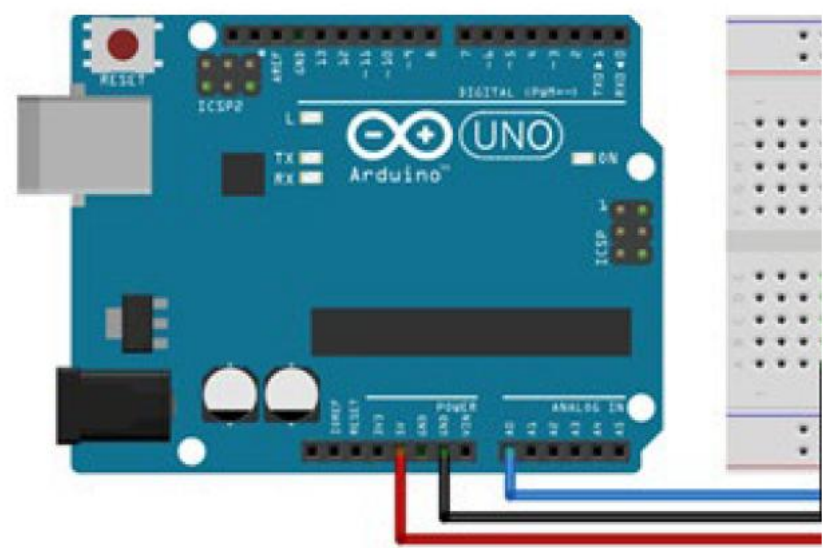


Arduino

Termistor (NTC – 1kΩ)



O termistor NTC (do inglês *Negative Temperature Coefficient*) é um semicondutor sensível à temperatura, utilizado para controle, em circuitos eletrônicos. Possui um coeficiente de variação de resistência conforme a temperatura aumenta, ou seja, a sua resistência elétrica diminui com a temperatura.



T [°C]	R_Nom [Ω]	R_Min [Ω]	R_Max [Ω]	ΔR/R_Nom [±%]	ΔT [±°C]
-55	59.147,00	48.631,00	69.664,00	17,80	2,70
-50	42.661,00	35.521,00	49.781,00	16,70	2,60
-45	31.088,00	26.207,00	35.969,00	15,70	2,50
-40	22.903,00	19.530,00	26.276,00	14,70	2,50
-35	17.062,00	14.700,00	19.405,00	13,80	2,40
-30	12.827,00	11.172,00	14.482,00	12,90	2,30
-25	9.746,00	8.572,00	10.920,00	12,00	2,20
-20	7.477,00	6.638,00	8.316,00	11,20	2,20
-15	5.790,00	5.186,00	6.394,00	10,40	2,10
-10	4.523,00	4.086,00	4.961,00	9,70	2,00
-5	3.564,00	3.246,00	3.883,00	8,90	1,90
0	2.832,00	2.599,00	3.065,00	8,20	1,80
5	2.267,00	2.096,00	2.438,00	7,50	1,70
10	1.829,00	1.703,00	1.955,00	6,90	1,60
15	1.486,00	1.393,00	1.578,00	6,30	1,50
20	1.215,00	1.146,00	1.283,00	5,60	1,40
25	1.000,00	950,00	1.050,00	5,00	1,30
30	828,20	781,60	874,70	5,60	1,50
35	689,90	647,20	732,60	6,20	1,70
40	577,80	539,10	616,60	6,70	1,90
45	486,60	451,40	521,80	7,20	2,10
50	411,80	380,00	443,70	7,70	2,40
55	350,20	321,40	379,00	8,20	2,60
60	299,20	273,20	325,20	8,70	2,80
65	256,70	233,20	280,20	9,20	3,00
70	221,20	200,00	242,40	9,60	3,30
75	191,40	172,20	210,60	10,00	3,50
80	166,20	148,80	183,60	10,50	3,80
85	144,80	129,10	160,60	10,90	4,00
90	126,70	112,40	140,90	11,30	4,30
95	111,20	98,22	124,10	11,70	4,50
100	97,87	86,10	109,60	12,00	4,80
105	86,43	75,72	97,14	12,40	5,00
110	76,55	66,79	86,31	12,80	5,30
115	67,99	59,09	76,90	13,10	5,60
120	60,56	52,42	68,69	13,40	5,90
125	54,07	46,63	61,52	13,80	6,20