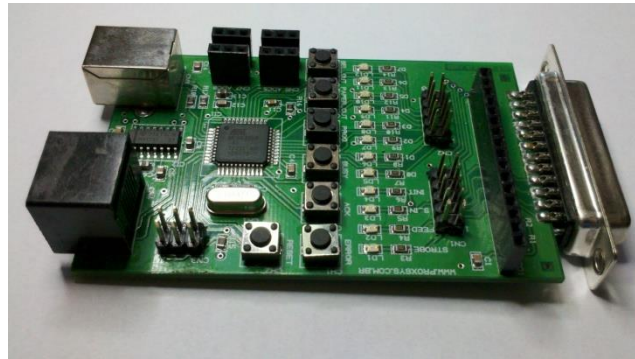


Placa Lab_Uino



A placa Lab_Uino foi desenvolvida como uma plataforma aberta para o desenvolvimento e aprendizagem de sistemas microprocessados. Utilizando o processador ATmega32U4 da Atmel, a placa permite sua utilização e gravação apenas utilizando uma interface USB, possibilitando o desenvolvimento de pequenos sistemas até dispositivos que utilizam esta interface.

Interfaces disponíveis na placa:

- Interface USB para uso geral e gravação (CN9)
- 7 chaves para simulação de entrada e reset (CH1/CH7)
- 12 leds para indicação de saída (LD1/LD12)
- 1 conector padrão ISP (CN3)
- 2 conectores de expansão para placa padrão passo 100 mils (0,1 pol) (CN1/CN2)
- 1 conector para módulos de LCD (LCD1)
- 4 conectores de interface analógica (CN5/CN8)
- 1 conector DB25-Fêmea similar interface paralela-PC (Borda da placa)
- 1 conector RJ-45 com interface RS-232 (CN4)

A figura 1 ilustra as interfaces e dispositivos disponíveis na placa.

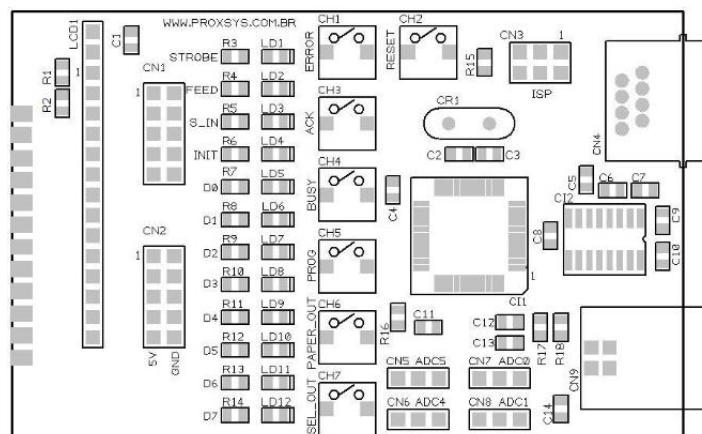


Fig 1. Dispositivos da placa

Descrição das interfaces

Conector DB-25

O conector DB-25 possui os mesmos sinais que um conector da interface paralela dos computadores tipo IBM- PC, possibilitando a conexão de outras placas de interface disponíveis no mercado.

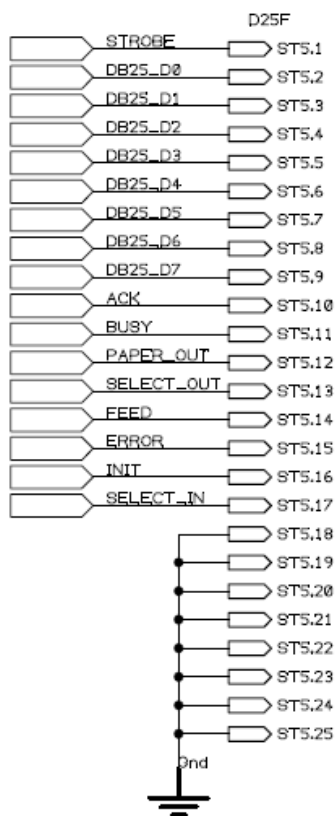


Fig 2. Pinagem conector DB-25

Os sinais deste conector estão ligados diretamente ao processador ATmega32U4 como também em outros dispositivos da placa.

Chaves de simulação de entrada e reset

Para facilitar a depuração de sistemas, a placa possui 6 chaves para simulação de entrada e 1 chave para reset do processador. 5 destas chaves estão ligadas aos sinais de entrada da porta paralela (ack, busy, paper_out, select_out e error), outra chave (prog) está ligada ao sinal HWB do processador, sendo usado para iniciar a gravação do processador após reset (bootloader) ou como uso geral, finalmente temos uma chave ligada ao reset do processador.

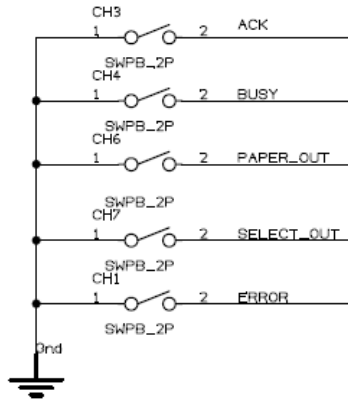


Fig 3. Chaves de simulação de entrada

LEDs indicadores das saídas

Os 12 LEDs (LD1 a LD12) estão conectados aos sinais de saída da porta paralela, sendo que nível lógico 1 no sinal acende o led correspondente.

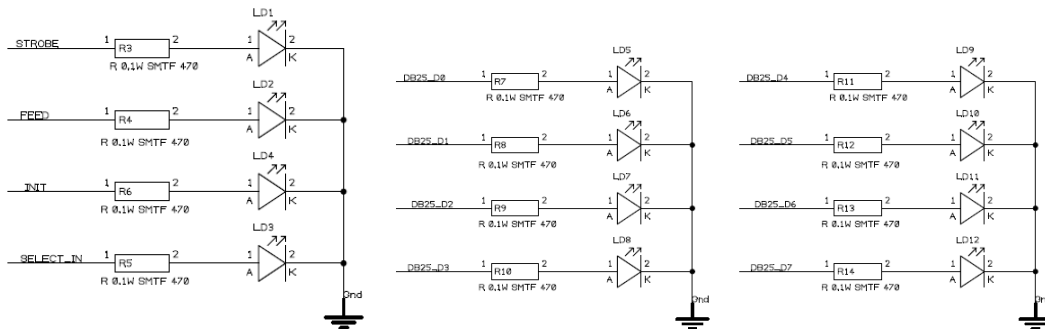


Fig 4. LEDs indicadores

Conectores de expansão

Para facilitar a conexão com outras placas de protótipos, a placa possui 2 conectores (CN1/CN2) com os sinais de entrada, saída, 5V e terra. Estes conectores estão com a distância de passos de 100 mils, compatíveis com placas padrões para protótipos. Estes conectores também podem ser conectados através de conectores para flat-cable tipo 2x5 pinos.

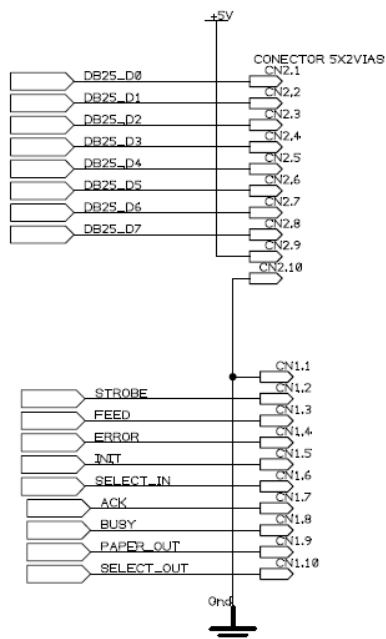
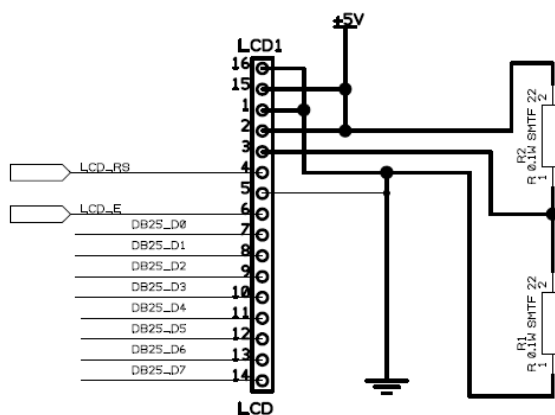


Fig 5. Conectores de expansão

Conector para módulos de LCD

A placa permite a conexão de módulos LCD compatíveis com o padrão “Hitachi HD44780”, muito comum no mercado. Os pinos de dados no módulo estão conectados nos pinos de dados da interface PC. Este conector possui 2 sinais exclusivos para controle do LCD e estão conectados diretamente ao ATmega32U4, sendo o LCD_RS e LCD_E. Estes sinais podem ser utilizados como entrada ou saída genérica caso não se conecte o LCD.

A comunicação com o LCD pode ser feita em 4 ou 8 bits, conforme o software de controle.



Conector padrão ISP

Este conector permite a gravação dos fusebits do processador ou a conexão de dispositivos com interface SPI. A sua pinagem é compatível com os gravadores ISP da Atmel.

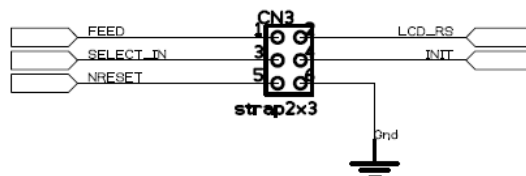


Fig 6. Conector ISP

Interfaces com sinais analógicos

Para efetuar leitura de sinais analógicos, a placa possui 4 conectores (CN5 a CN8) com o pino central ligado diretamente as entradas do conversor analógico digital do processador.

Os pinos externos estão ligados a 5V e terra, desta forma é possível a conexão direta de um potenciômetro para simulação ou sensores de temperatura tipo LM35 como exemplo.

Estes sinais estão ligados as entradas AD0,AD1,AD4 e AD5 do processador. Caso não desejado, estes sinais podem ser programados como entrada ou saída digital. Estes conectores também obedecem o passo de 100 mils.

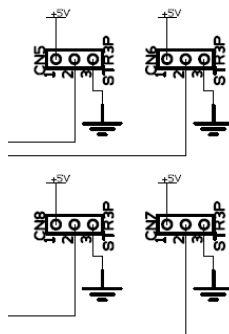


Fig 7. Entradas Analógicas

Interface RS-232

Esta interface possibilita a comunicação serial com outros sistemas no padrão RS-232. O nível do sinal já está no padrão da interface (+/- 9V) . Os sinais de transmissão e recepção estão disponíveis no conector RJ-45 (CN4). Em algumas versões de placa este conector e o driver não estão equipados para possibilitar a conexão com módulos Bluetooth com interface 5V.

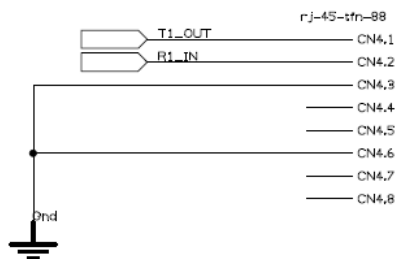


Fig 8. Interface RS-232

Tabela de interligação de sinais

A tabela mostra todas as conexões entre os conectores e o processador, bem como a ligação com o sinal equivalente da placa “Arduino Leonardo” e no Arduino Lab_Uino.

Interligação entre dispositivos

Sinal	CN2	CN1	CN3	DB25		LCD1	32u4	Arduino Leonardo	Arduino Lab_Uino
DB25_D0	1			2	LD5	7	PD.4	DP4	DP5
DB25_D1	2			3	LD6	8	PD.6	DP12	DP6
DB25_D2	3			4	LD7	9	PD.7	DP6	DP7
DB25_D3	4			5	LD8	10	PB.4	DP8	DP8
DB25_D4	5			6	LD9	11	PB.5	DP9	DP9
DB25_D5	6			7	LD10	12	PB.6	DP10	DP10
DB25_D6	7			8	LD11	13	PC.6	DP5	DP11
DB25_D7	8			9	LD12	14	PC.7	DP13	DP12
STROBE		2		1	LD1		PB.7	DP11	DP1
FEED		3	1	14	LD2		PB.3	MOSI	DP2
ERROR		4		15	CH1		PD.0	DP3	DP14
INIT		5	4	16	LD4		PB.2	MISO	DP4
SELECT_IN		6	3	17	LD3		PB.1	SCK	DP3
ACK		7		10	CH3		PD.1	DP2	DP15
BUSY		8		11	CH4		PD.5	TXLED	DP16
PAPER_OUT		9		12	CH6		PF.7	Aln0	DP18
SELEC_OUT		10		13	CH7		PF.6	Aln1	DP19
5V	9					2,15			
GND	10	1	6	18 a 25		1,5,16			
PROG(HWB)					CH5		PE.2	HWB	DP17
LCD_E						6	PE.6	DP7	DP13
LCD_RS			2			4	PB.0	RXLED	DP0
UART_RX							PD.2	DP0	DP21
UART_TX							PD.3	DP1	DP20
ADC5					CN5		PF.5	Aln2	DP25/A3
ADC4					CN6		PF.4	Aln3	DP24/A2
ADC1					CN8		PF.1	Aln4	DP23/A1
ADC0					CN7		PF.0	Aln5	DP22/A0
RESET			5					RESET	

Gravando programas na placa

Após a compilação do programa um arquivo .hex com o código a ser gravado é gerado. Para a gravação deste arquivo é necessário o programa FLIP da Atmel e colocar a placa no modo de programação.

Para a gravação execute os seguintes passos:

- Coloque a placa no modo de gravação apertando simultaneamente os botões PROG e RESET.
- Mantenha o botão PROG pressionado e solte o RESET. A placa entrará no modo de boot loader.
- Abra o programa FLIP
- No menu “device” e ”select” selecione o dispositivo ATMEGA32U4

- Abra a interface USB selecionando USB no ícone de cabo USB
- Abra o arquivo .hex com o programa a ser gravado no menu “file” “load hex”
- Selecione o botão “run” para gravar o programa no processador.
- Desconecte a placa e reconecte novamente para iniciar o software gravado.

Na primeira vez que a placa for conectada ao PC para programação é necessário a instalação do driver de gravação do processador. Para isto aguarde o Windows finalizar a instalação automática, vá até o gerenciador de dispositivos e instale manualmente em atualizar drivers. Os drivers corretos estão na pasta de instalação do Flip, selecione a opção de procurar em pastas e sub pastas. Após a instalação, desconecte a placa e conecte novamente. Este procedimento só será necessário na primeira vez que a placa for conectada no computador.

Aplicativos compilados

Outra aplicação da placa é como módulos funcionais já desenvolvidos com diversas aplicações. Para a ativação destes módulos é necessário a gravação do arquivo .hex correspondente.

Modulo de aquisição com interface HID

Com a gravação de firmware específico a placa Lab_Uino é reconhecida como um dispositivo de interface humana (HID) quando conectada a um PC, com um protocolo de comandos específicos e arquivos de funções em DLL torna-se fácil a utilização de outras ferramentas para controle dos dispositivos da placa.

Esta facilidade permite ao usuário programar sistemas em “C”, “C #”, Delphi, Lab_View e outros compiladores. Estes sistemas podem controlar as saídas e adquirir dados das entradas das placas utilizando chamadas de funções.

Funções do arquivo Lab_Uino.dll

O arquivo Lab_Uino disponibiliza as seguintes funções para controle da placa.

- int Open_Lab(void);
- int Close_Lab(void);
- int Read_Keys(void);
- int Read_Key(unsigned char key_num);
- int Read_ADC(unsigned char adc_num);
- int Write_Data_Led(unsigned char data);
- int Write_Ctr_Led(unsigned char data);
- int Led_On(unsigned char led_num);
- int Led_Off(unsigned char led_num);
- int Led_Toggle(unsigned char led_num);
- int Led_Wr(unsigned char led_num,unsigned char valor);
- int pwm_A(unsigned char valor);
- int pwm_B(unsigned char valor);
- int pwm_C(unsigned char valor);

Função Open_Lab:

Inicia a comunicação com a placa, a função abre o arquivo "AtUsbHid.dll" e executa a sequência necessária para "achar" a placa e "abrir" o dispositivo USB possibilitando futuros acessos. Esta função deve ser chamada uma vez no início do aplicativo.

Função Close_Lab:

Finaliza a comunicação com a placa. Esta função deve ser chamada uma vez no final do aplicativo.

Funções Read_Keys e Read_Key:

A função **Read_keys** executa a leitura de todas as chaves do Lab_Unio. A função retorna um valor formado pelos estados das chaves em cada bit, sendo bit=1 chave apertada ou sinal no pino = 0 (gnd).

D7	D6	D5	D4	D3	D2	D1	D0
0	0	SEL_OUT	PAPER_OUT	PROG	BUSY	ACK	ERROR

Formato do byte

Já a função **Read_key** executa a leitura de uma determinada chave do Lab_Unio. A função retorna 0 ou 1 conforme se a chave está solta ou apertada. A função tem como parâmetro o número da serigrafia da chave que se deseja verificar. (1-error, 3-ack, 4-busy, 5-prog, 6-paper_out, 7-sel_out)

Read_ADC:

A função Read_ADC retorna o valor das entradas dos conversores analógicos/digitais disponíveis no Lab_Unio. A função tem como parâmetro a entrada que se deseja ler o valor (conforme serigrafia da placa) como exemplo Read_ADC(0) para a entrada ADC0.

As entradas ADC0 e ADC1 retornam valor de 0 a 255, conversor de 8 bits, sendo 255=5V.

As entradas ADC4 e ADC5 retornam valor de 0 a 1023, conversor de 10 bits, sendo 1023=5V.

Se utilizar o comando Read_ADC(7), será retornado o valor de temperatura do processador do Lab_Unio.

Funções Write_Data_Led e Write_Ctr_Led:

Similar a porta paralela do PC (outport 0x378,valor), pode-se utilizar as funções Write_Data_Led e Write_Ctr_led. Estas funções permite a escrita de 1 byte, sendo cada bit mapeado em 1 led ou sinal. Sendo 0=gnd Led apagado, 1= 5V Led aceso.

A função Write_Data_Led controla os Leds LD5 a LD12 (sinais D0 a D7), já a função Write_Ctr_Led controla os Leds LD1 a LD4 (sinais strobe, feed, selec_in e init)

	D7	D6	D5	D4	D3	D2	D1	D0
Data	LD12	LD11	LD10	LD9	LD8	LD7	LD6	LD5

Ctr	-	-	-	-	LD4	LD3	LD2	LD1
-----	---	---	---	---	-----	-----	-----	-----

Formato do byte

Funções Led_On, Led_Off, Led_Toggle, Led_Wr:

No caso de se desejar alterar o estado de cada Led individualmente pode-se utilizar os comandos Led_On para acender, Led_Off para apagar e Led-Toggle para alternar o estado do led. As funções tem como parâmetro o número do Led (serigrafia).

A função Led_Wr, escreve o estado de um Led em específico. O número do Led e o estado são passados como parâmetros da função.

Funções pwm_A, pwm_B, pwm_C:

Ativa as saídas de PWM, o parâmetro pode variar de 0 a 255. As saídas estão no Led9, Led10 e Led1 respectivamente.