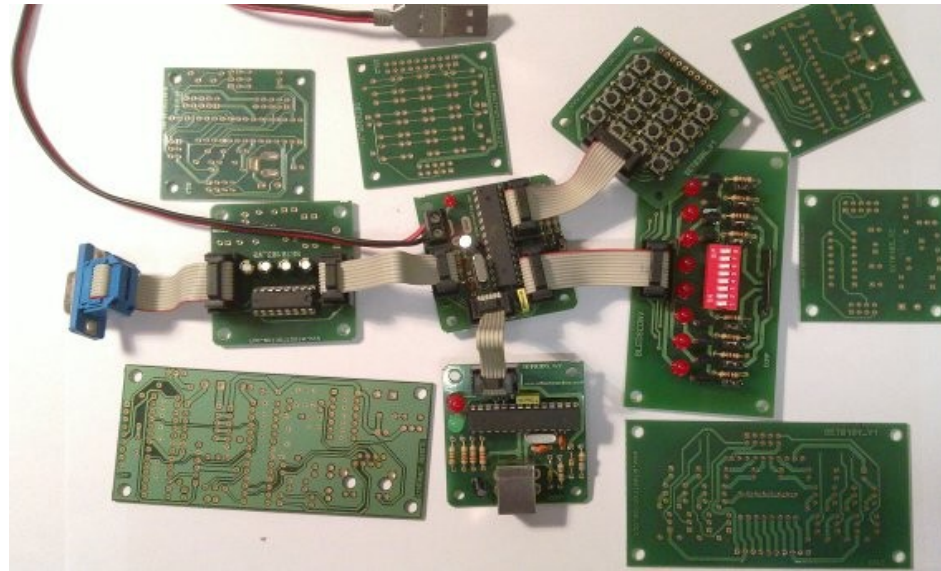
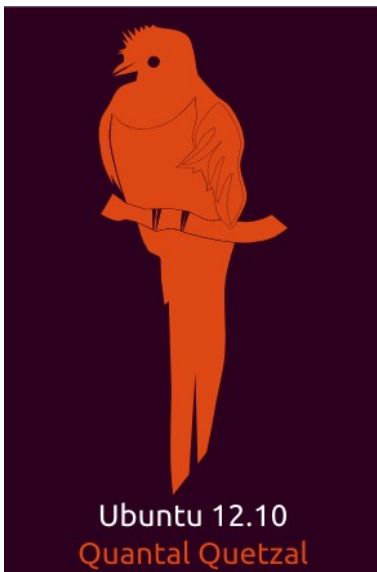


Sessió pràctica de maquinari lliure - OSHW



Festa
Quantal Quetzal
Ubuntu 12.10
10 / 11 / 2012

Jordi Binefa i Martínez

Responsable d'R+D+i a

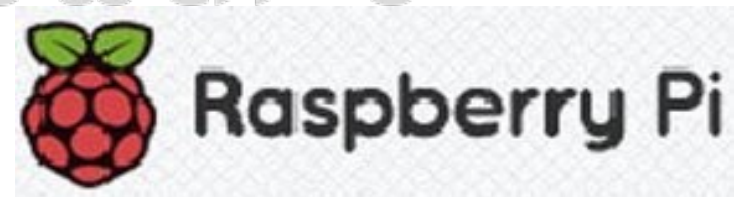
Professor de cicles formatius a



Sessió pràctica de maquinari lliure

Índex

- Introducció a la metadistribució
- Plaques disponibles
- Programació de microcontroladors de 8 bits (ATmega8/48/88)
- Programació de sistemes de 32 bits (Raspberry Pi, Beagleboard, Beaglebone i mini6410)
- Implementació pràctica dels exemples



Sessió pràctica de maquinari lliure

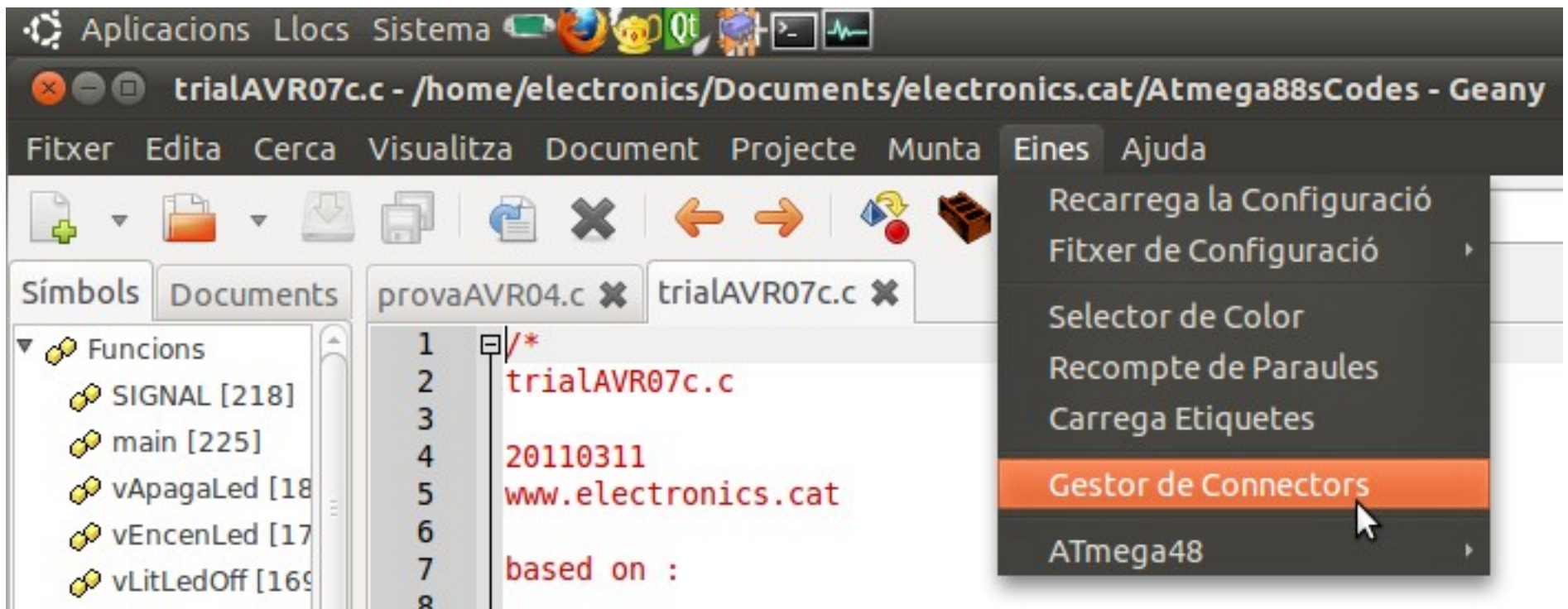
Introducció a la metadistribució

Metadistribució basada en Ubuntu 10.10



Sessió pràctica de maquinari lliure

Introducció a la metadistribució



Sessió pràctica de maquinari lliure

Introducció a la metadistribució

Trieu quins plugins carregar a l'inici:

Actiu	Connector	Fitxer
☐	Atmega48	/usr/lib/geany/geanyHex48.so
☐	Atmega8	/usr/lib/geany/geanyHex8.so
☐	Atmega8535	/usr/lib/geany/geanyHex8535.so
☒	Atmega88	/usr/lib/geany/geanyHex88.so
☐	Caràcters HTML	/usr/lib/geany/htmlchars.so
☐	Constructor de Classes	/usr/lib/geany/classbuilder.so
☐	Desa les Accions	/usr/lib/geany/saveactions.so
☐	Divideix la Finestra	/usr/lib/geany/splitwindow.so
☐	Explorador de Fitxers	/usr/lib/geany/filebrowser.so
☐	Exporta	/usr/lib/geany/export.so

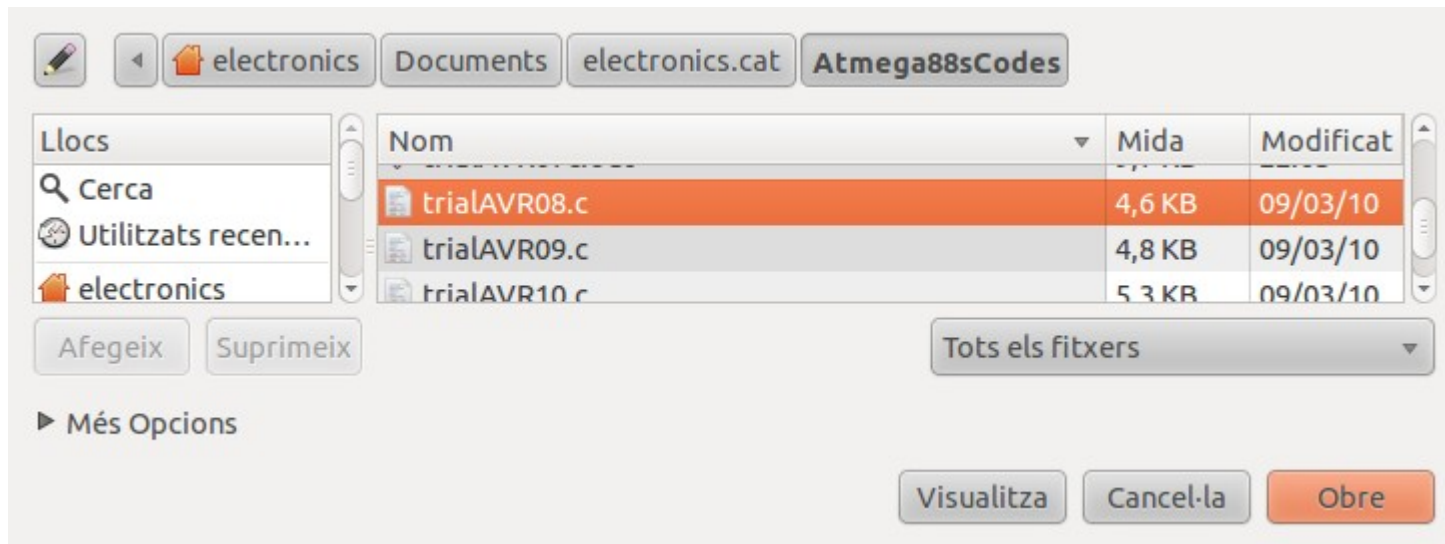
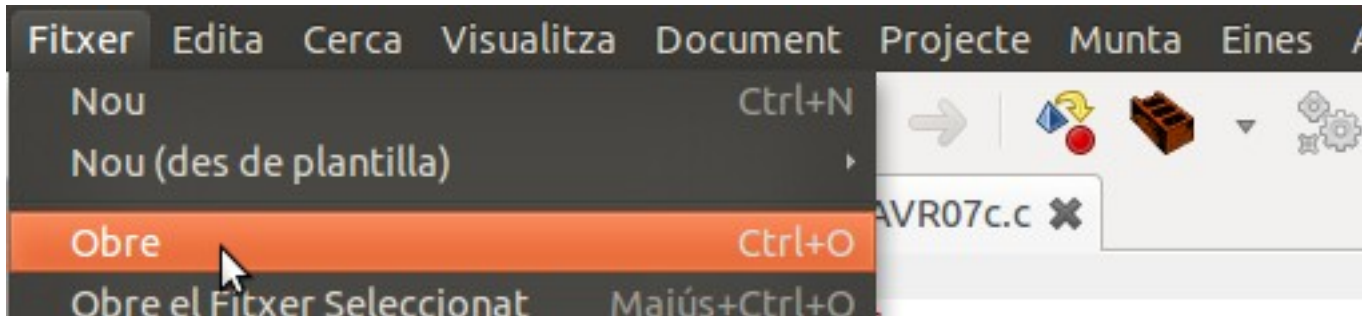
Detalls del connector: [Ajuda](#) [Preferències](#)

Connector:-Atmega88-0.0.14
 Descripció:-Geany's plugin to deal with Atmega88
 Autor(s):-Jordi Binefa <electronics.cat@gmail.com>

[D'acord](#)

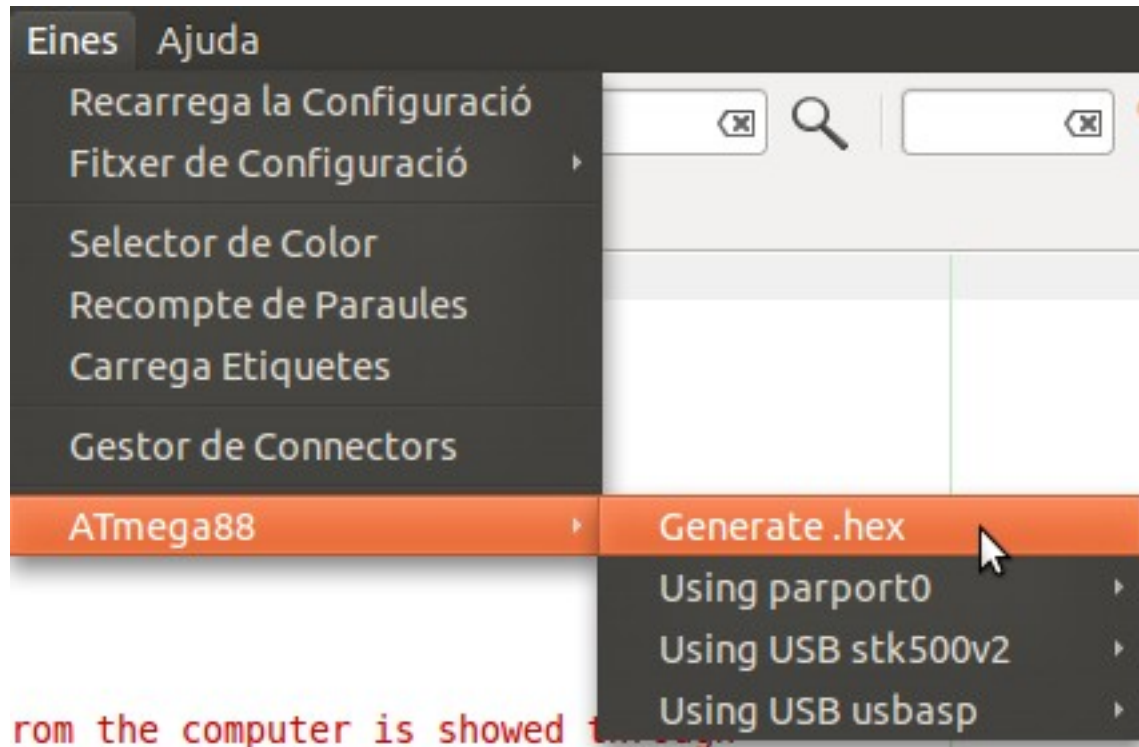
Sessió pràctica de maquinari lliure

Introducció a la metadistribució



Sessió pràctica de maquinari lliure

Introducció a la metadistribució



En el fitxer inclòs des de /home/electronics/Documents/electronics.cat/Atmega88sCodes/trialAVR08.c:45:
/usr/lib/gcc/avr/4.3.5/../../../../avr/include/avr/signal.h:36:2: avís: #warning "This header file is obsolete. Use <avr/interrupt.h>."

D'acord

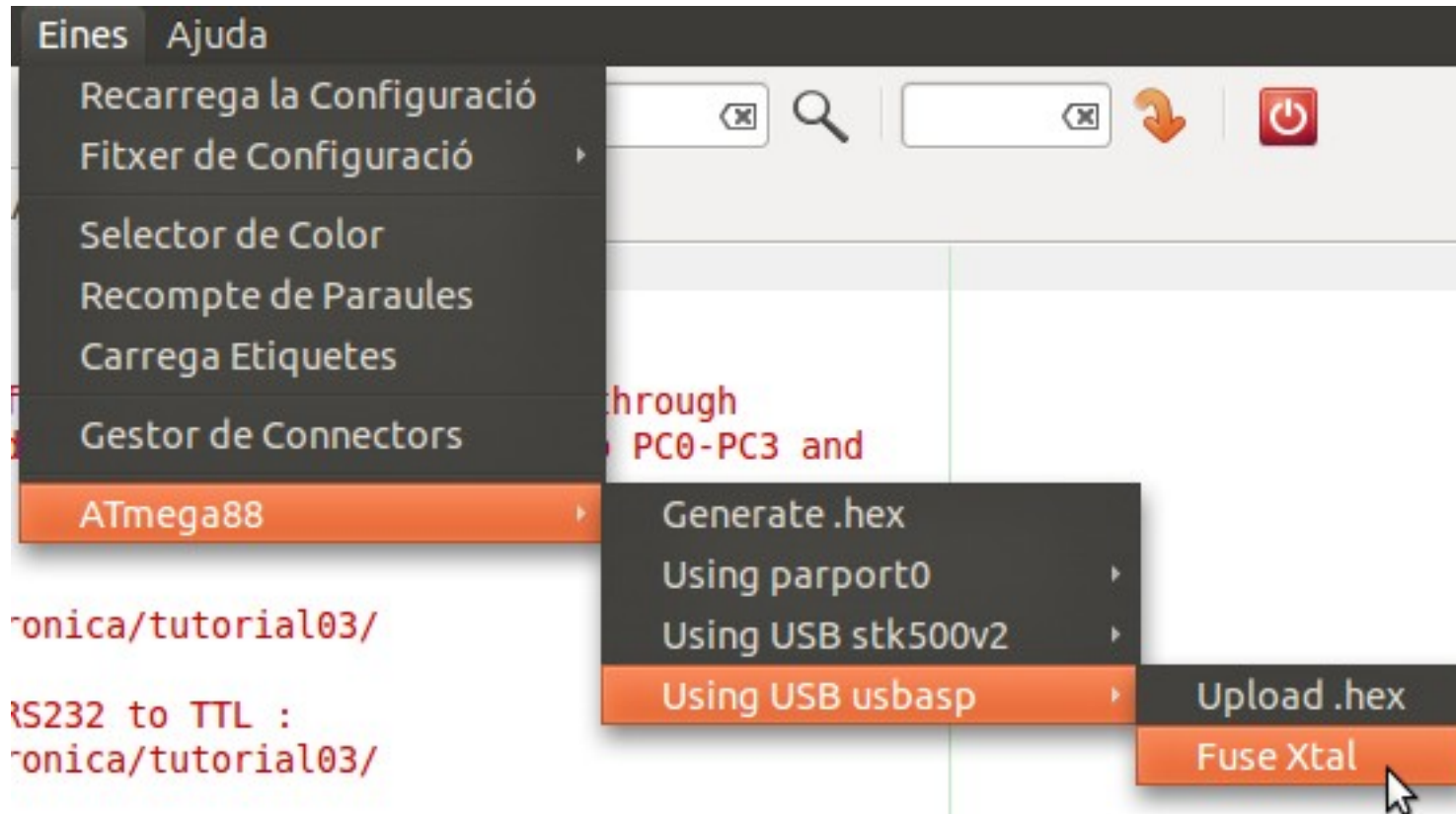


File /home/electronics/Documents/electronics.cat/
Atmega88sCodes/trialAVR08.hex has been created

D'acord

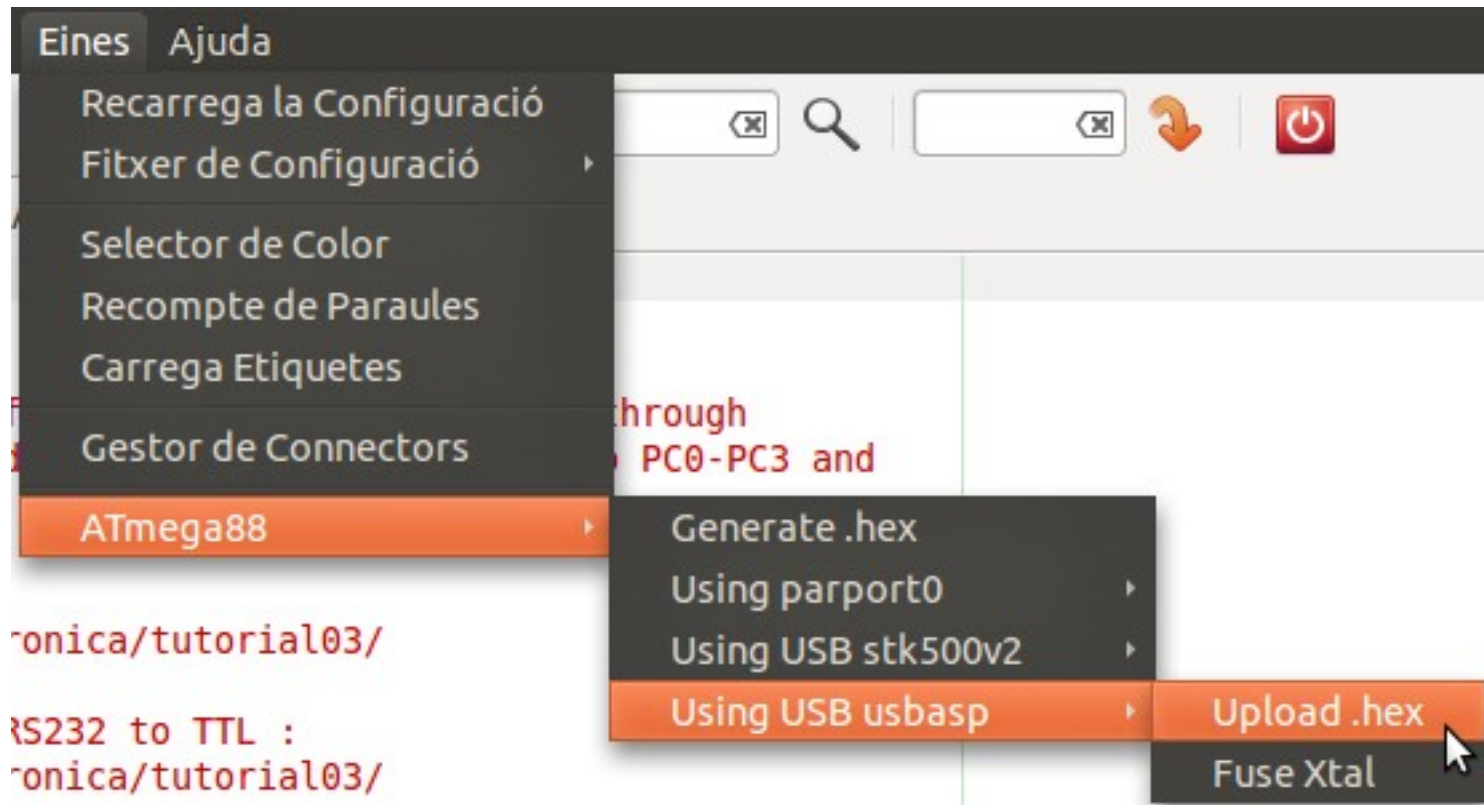
Sessió pràctica de maquinari lliure

Introducció a la metadistribució



Sessió pràctica de maquinari lliure

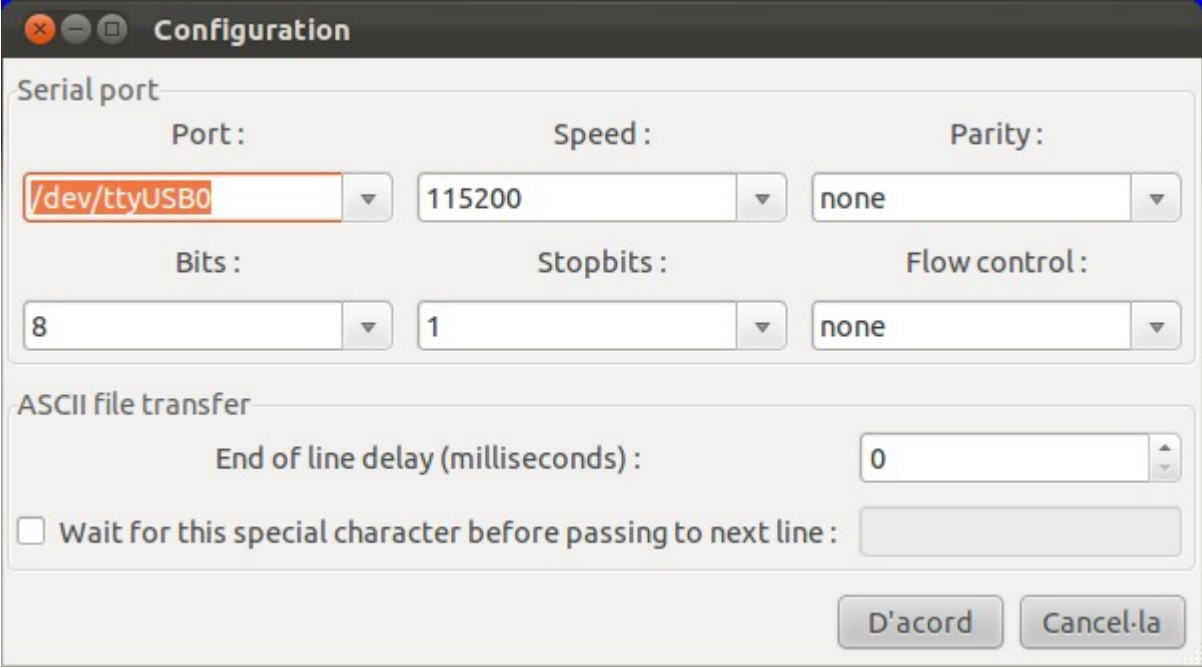
Introducció a la metadistribució



Sessió pràctica de maquinari lliure

Introducció a la metadistribució

GtkTerm



The image shows a 'Configuration' dialog box for GtkTerm. It is divided into two main sections: 'Serial port' and 'ASCII file transfer'. The 'Serial port' section contains six dropdown menus arranged in two rows. The first row includes 'Port:' (set to '/dev/ttyUSB0'), 'Speed:' (set to '115200'), and 'Parity:' (set to 'none'). The second row includes 'Bits:' (set to '8'), 'Stopbits:' (set to '1'), and 'Flow control:' (set to 'none'). The 'ASCII file transfer' section includes a label 'ASCII file transfer', a text input field for 'End of line delay (milliseconds):' (set to '0'), and a checkbox labeled 'Wait for this special character before passing to next line:' which is currently unchecked. At the bottom right, there are two buttons: 'D'acord' and 'Cancel·la'.

Configuration

Serial port

Port :

Speed :

Parity :

Bits :

Stopbits :

Flow control :

ASCII file transfer

End of line delay (milliseconds) :

☐ Wait for this special character before passing to next line :

D'acord Cancel·la

Sessió pràctica de maquinari lliure

Plaques disponibles

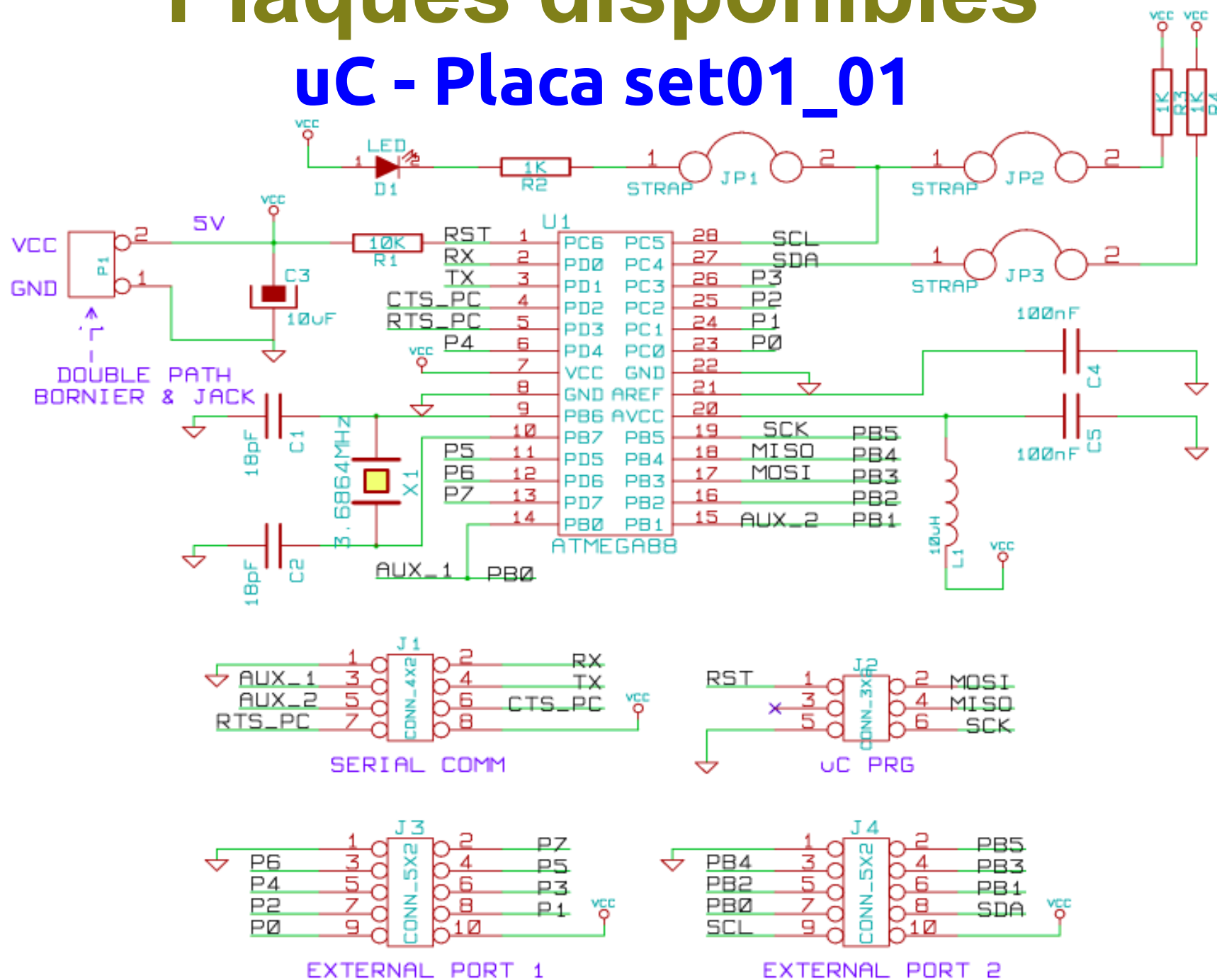
uC - Placa set01_01



Sessió pràctica de maquinari lliure

Plaques disponibles

uC - Placa set01_01



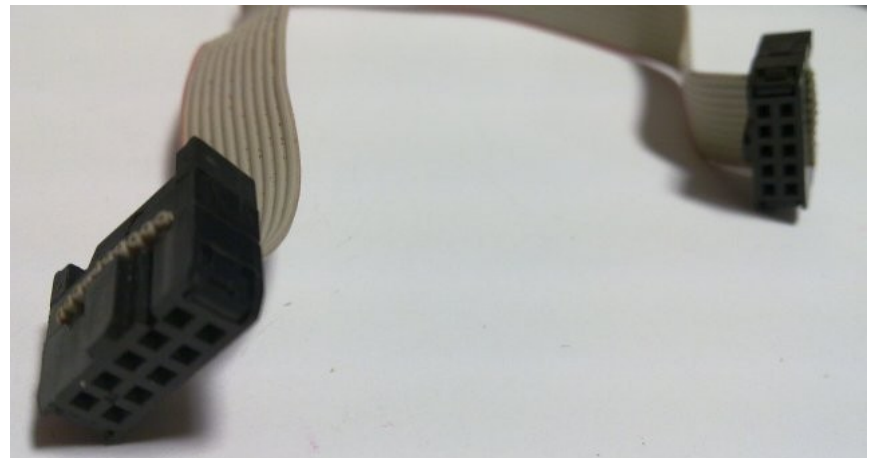
Sessió pràctica de maquinari lliure

Plaques disponibles

uC - Placa set01_01

Descripció

Aquesta placa té dos ports de 8 bits d'entrada / sortida. Un d'ells preparat per treballar amb el protocol TWI, també conegut com a I2C. La connectivitat a les plaques d'entrada/sortida es fa mitjançant el cable set01_04_A (10 fils, connectors de 5x2).



Sessió pràctica de maquinari lliure

Plaques disponibles

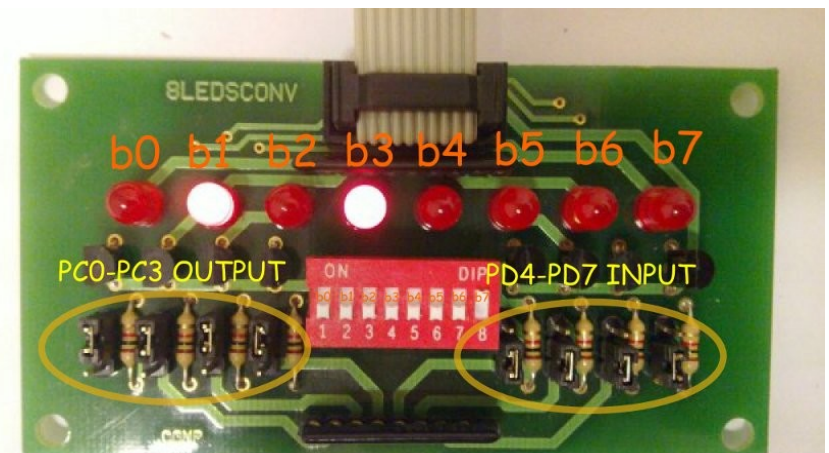
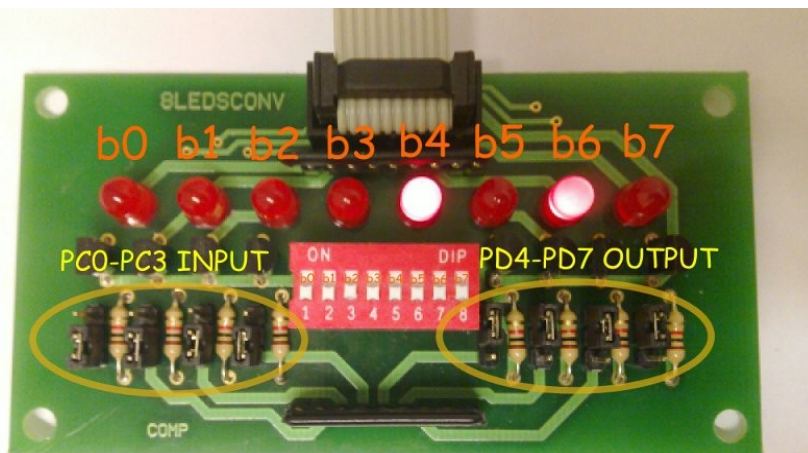
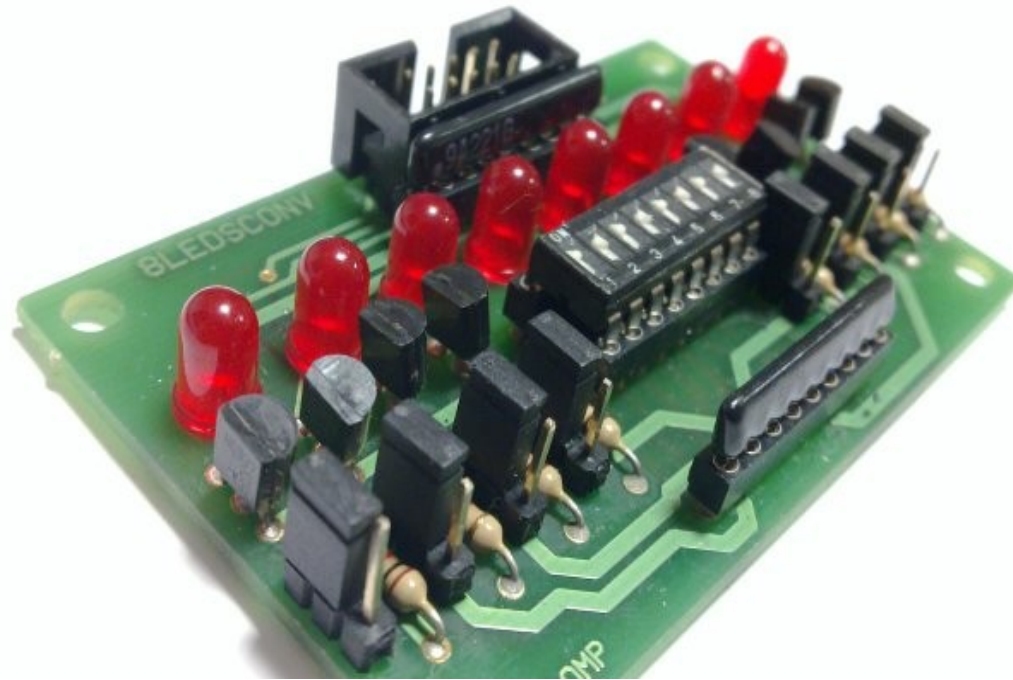
uC - Placa set01_01

- * La comunicació sèrie es pot realitzar per USB (set01_05) o per RS232 (set01_03) mitjançant el cable set03_11 (8 fils, connectors de 4x2). Aquestes plaques de comunicacions també poden alimentar aquesta placa.
- * La programació del xip es pot fer mitjançant USB (set03_09) o pel port paral·lel (set01_02) amb el cable set03_10 (6 fils, connectors de 3x2).
- * Cada funció té un connector de diferent mida per estalviar errades de connectivitat.
- * Hi ha 3 jumpers. Un per a un led de proves i els altres dos per permetre la comunicació TWI.
- * Al dur un cristall de quars s'ha de tenir cura de configurar els fusibles abans, o després, de fer la programació. Altrament no hi hauria precisió per a les comunicacions sèrie.

Sessió pràctica de maquinari lliure

Plaques disponibles

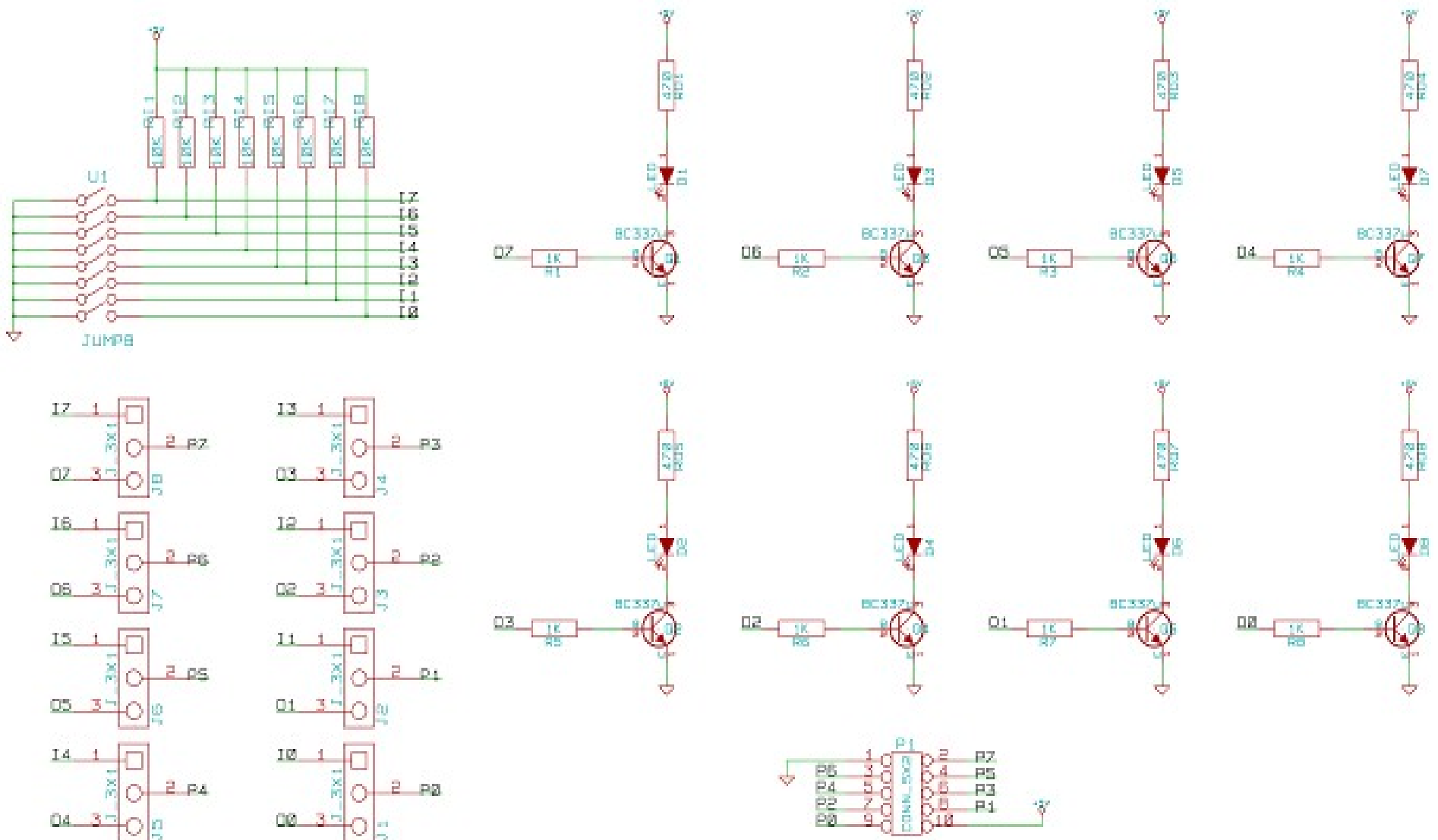
I/O - Placa set01_04



Sessió pràctica de maquinari lliure

Plaques disponibles

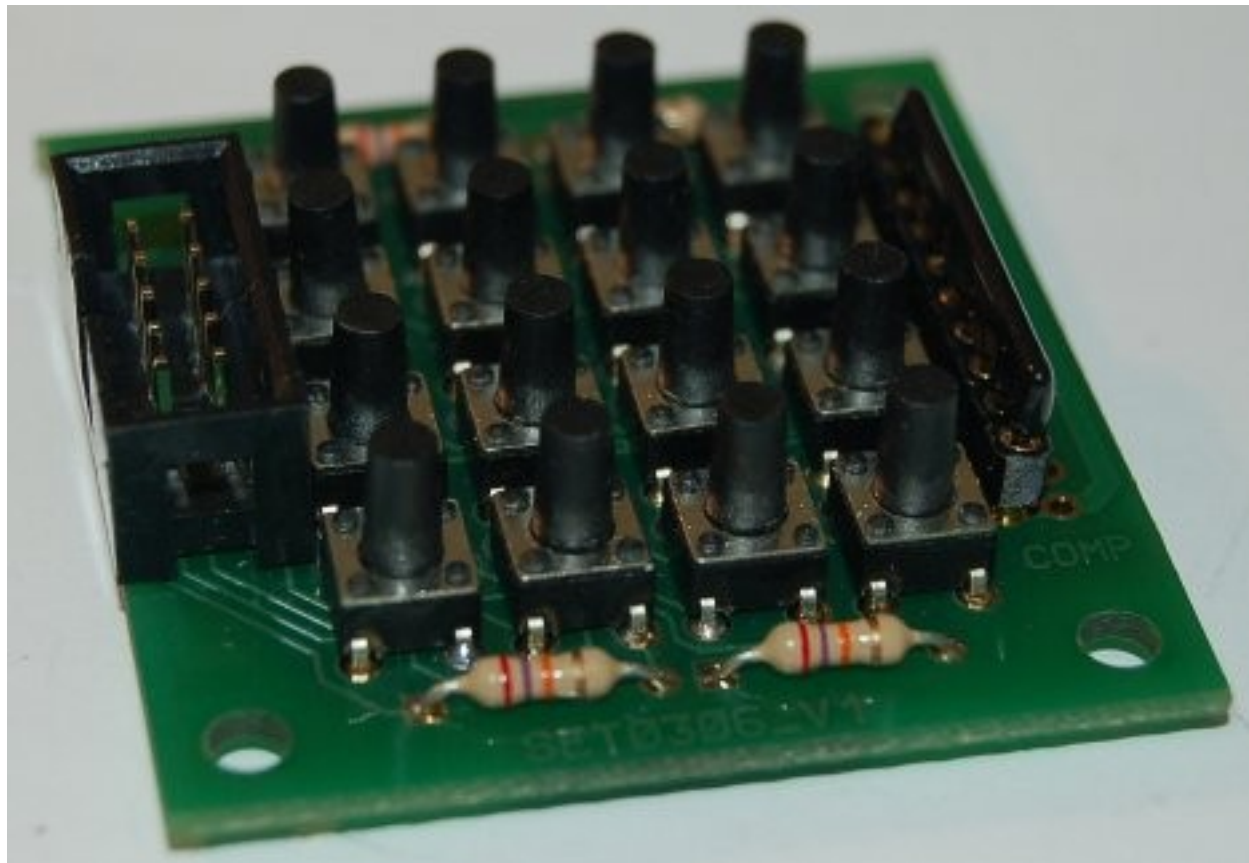
I/O - Placa set01_04



Sessió pràctica de maquinari lliure

Plaques disponibles

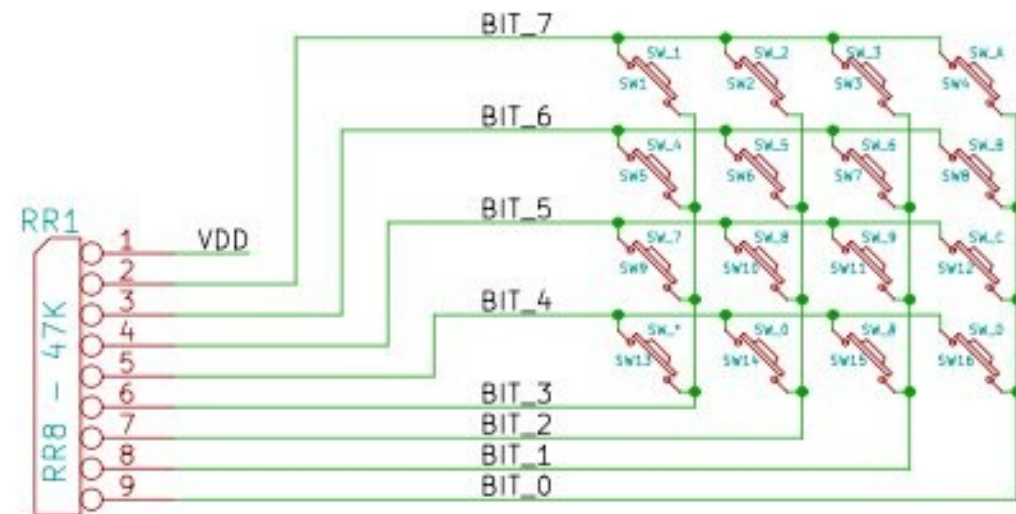
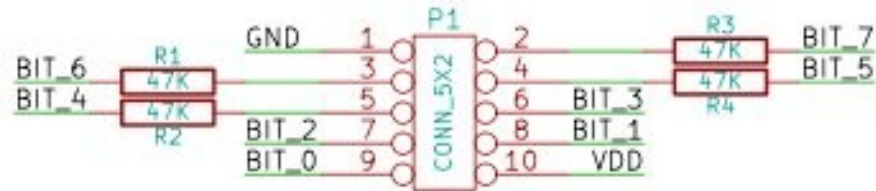
I/O - Placa set03_06



Sessió pràctica de maquinari lliure

Plaques disponibles

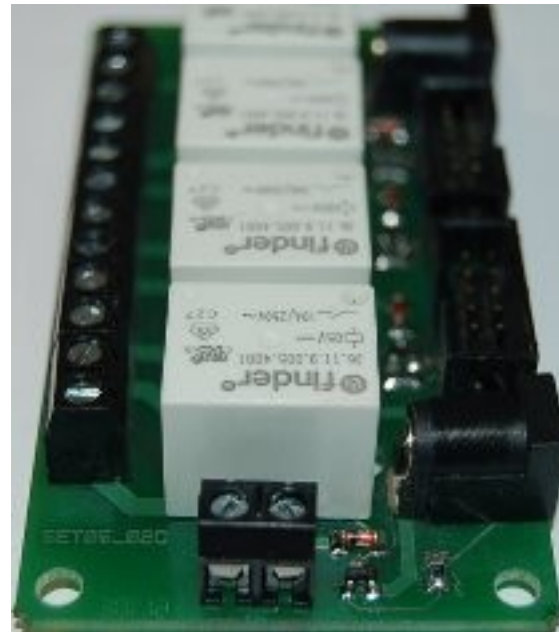
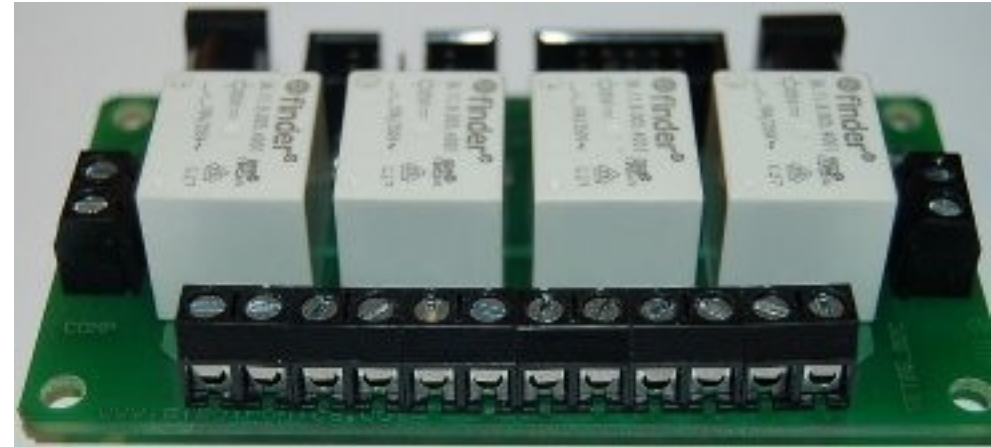
I/O - Placa set03_06



Sessió pràctica de maquinari lliure

Plaques disponibles

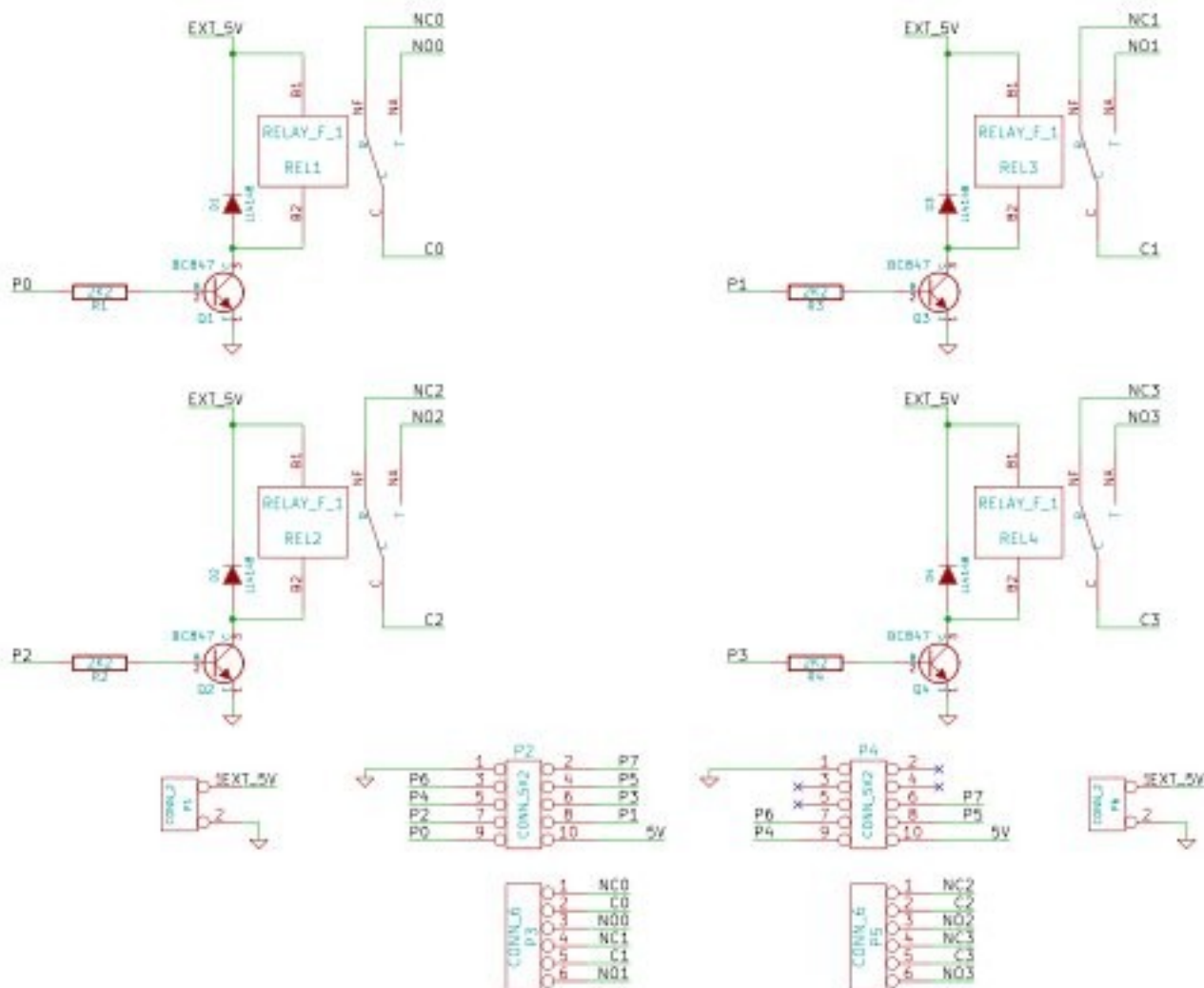
I/O - Placa set05_02



Sessió pràctica de maquinari lliure

Plaques disponibles

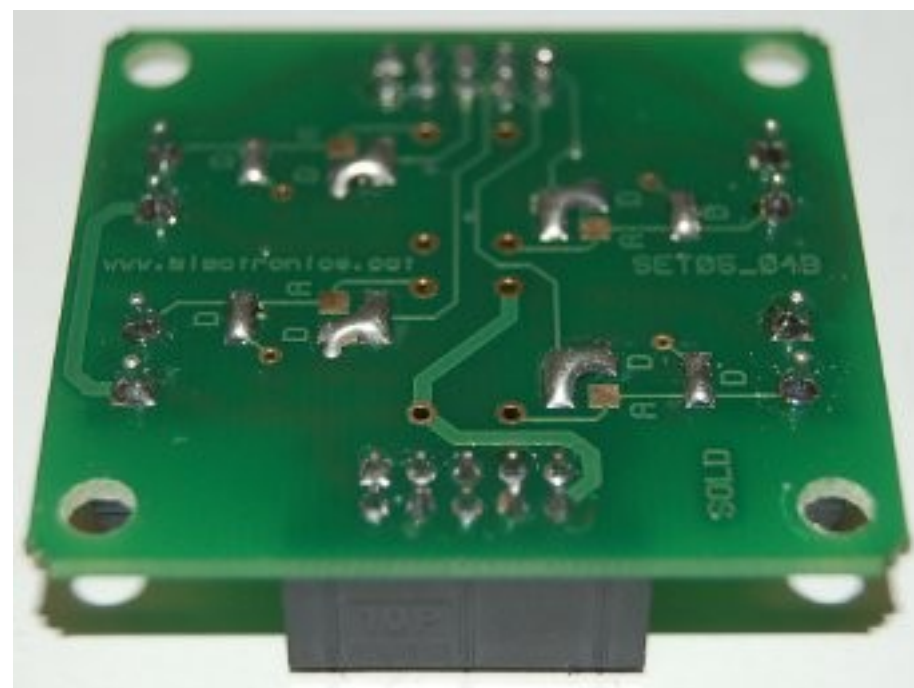
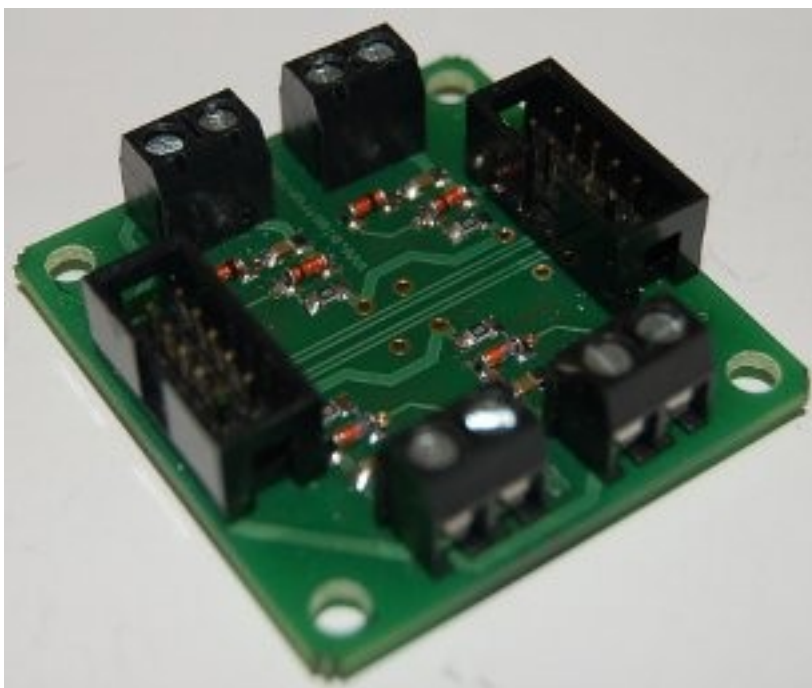
I/O - Placa set05_02



Sessió pràctica de maquinari lliure

Plaques disponibles

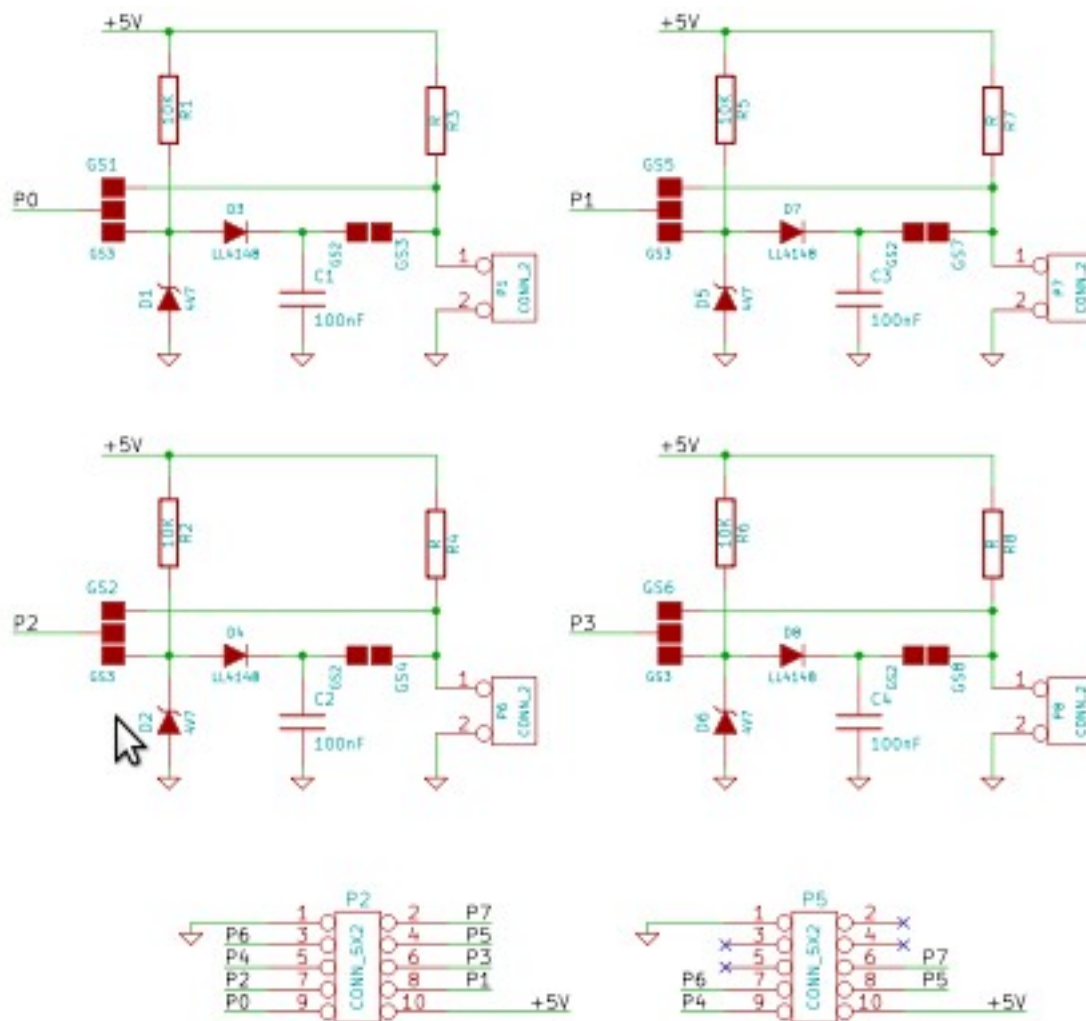
I/O - Placa set05_04



Sessió pràctica de maquinari lliure

Plaques disponibles

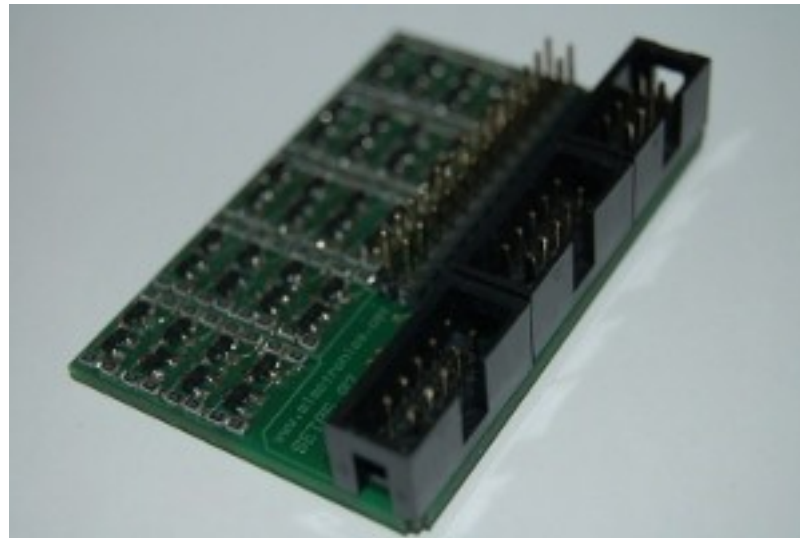
I/O - Placa set05_04



Sessió pràctica de maquinari lliure

Plaques disponibles

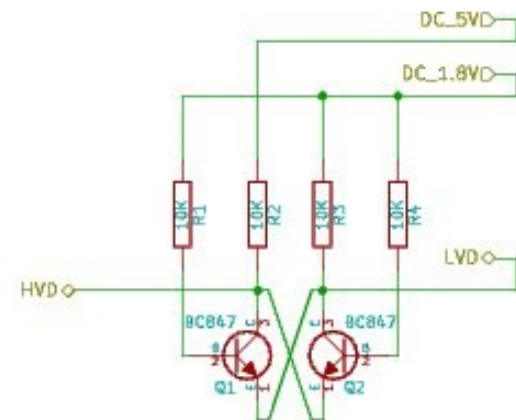
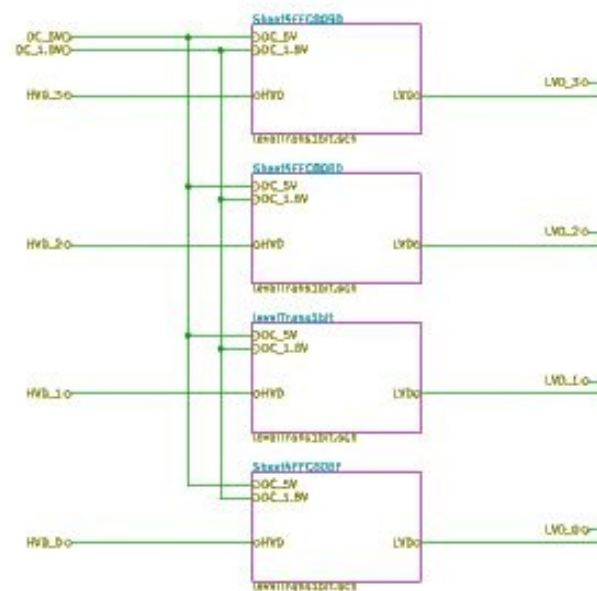
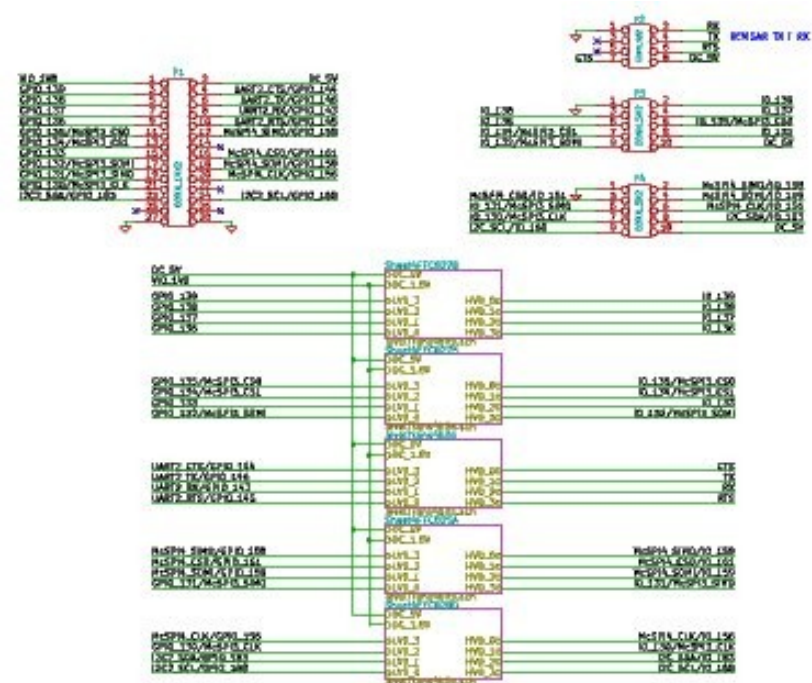
I/O – aturaBoard - Placa set05_07



Sessió pràctica de maquinari lliure

Plaques disponibles

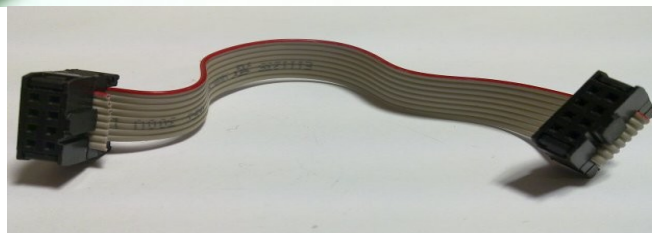
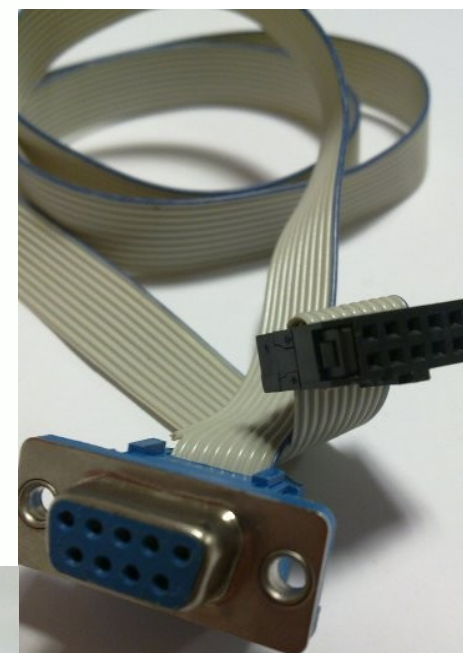
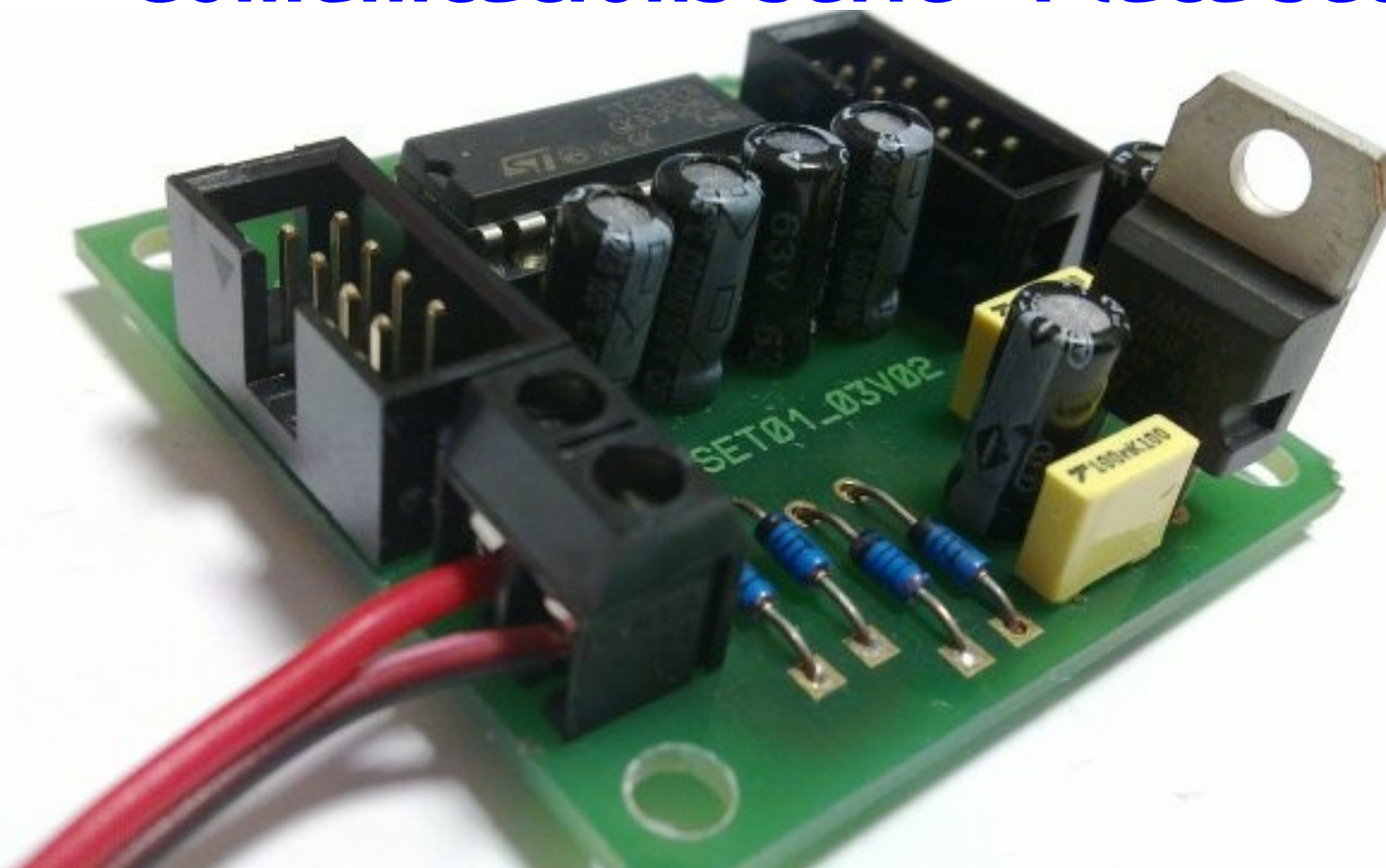
I/O – aturaBoard - Placa set05_07



Sessió pràctica de maquinari lliure

Plaques disponibles

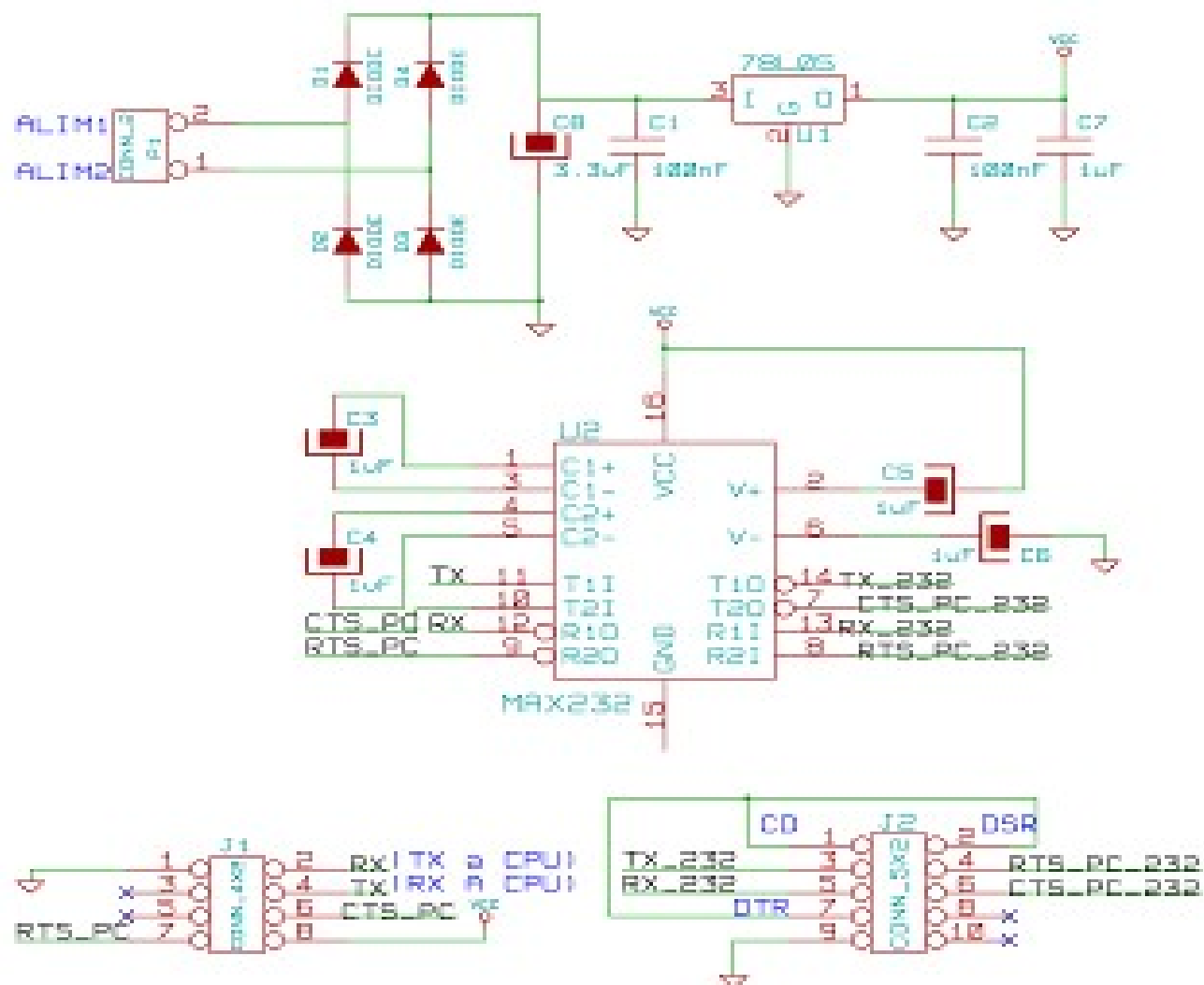
Comunicacions sèrie - Placa set01_03



Sessió pràctica de maquinari lliure

Plaques disponibles

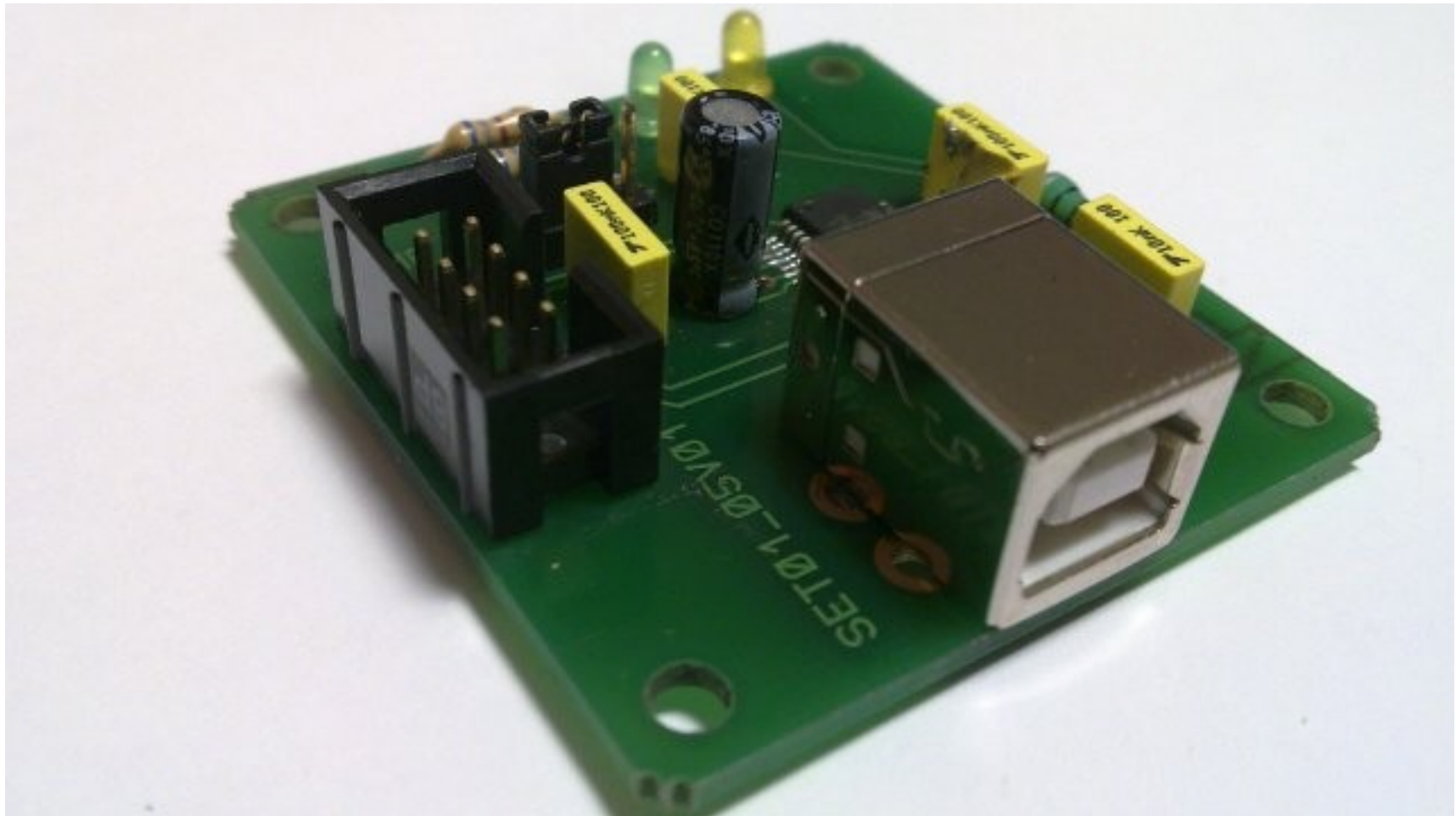
Comunicacions sèrie - Placa set01_03



Sessió pràctica de maquinari lliure

Plaques disponibles

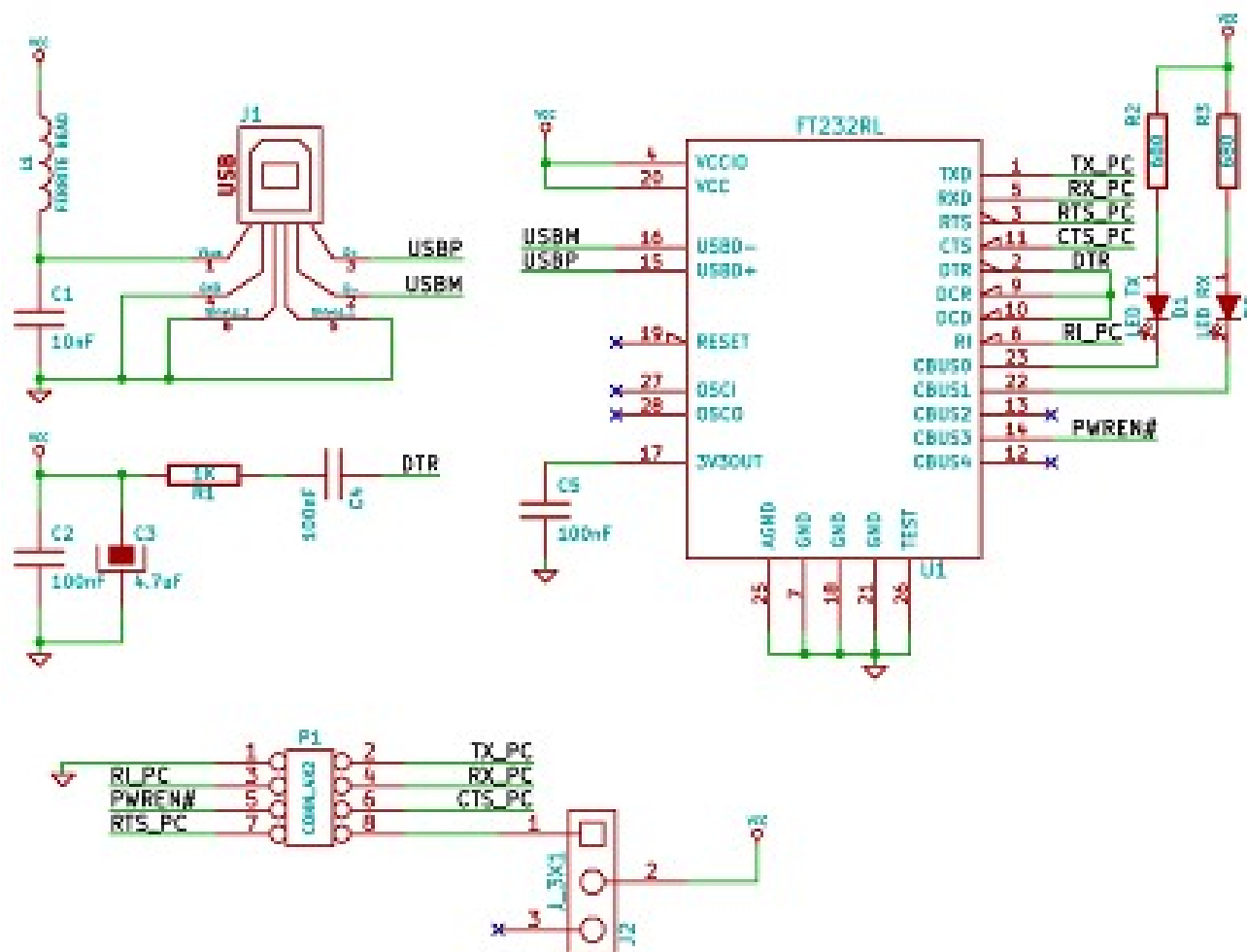
Comunicacions USB/TTL - Placa set01_05



Sessió pràctica de maquinari lliure

Plaques disponibles

Comunicacions USB/TTL - Placa set01_05

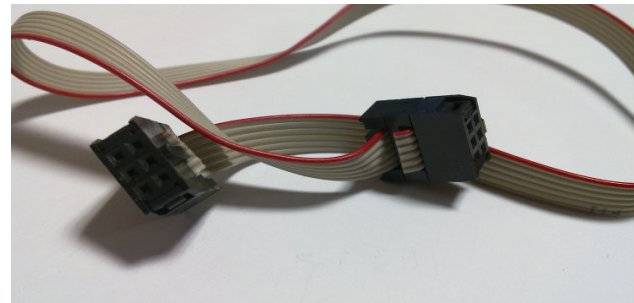
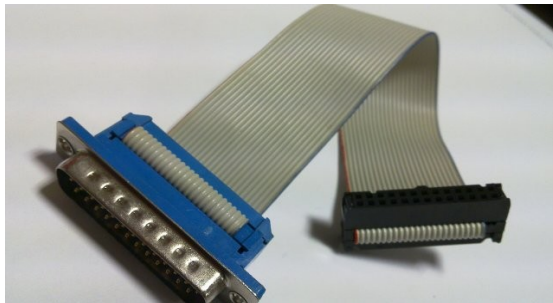
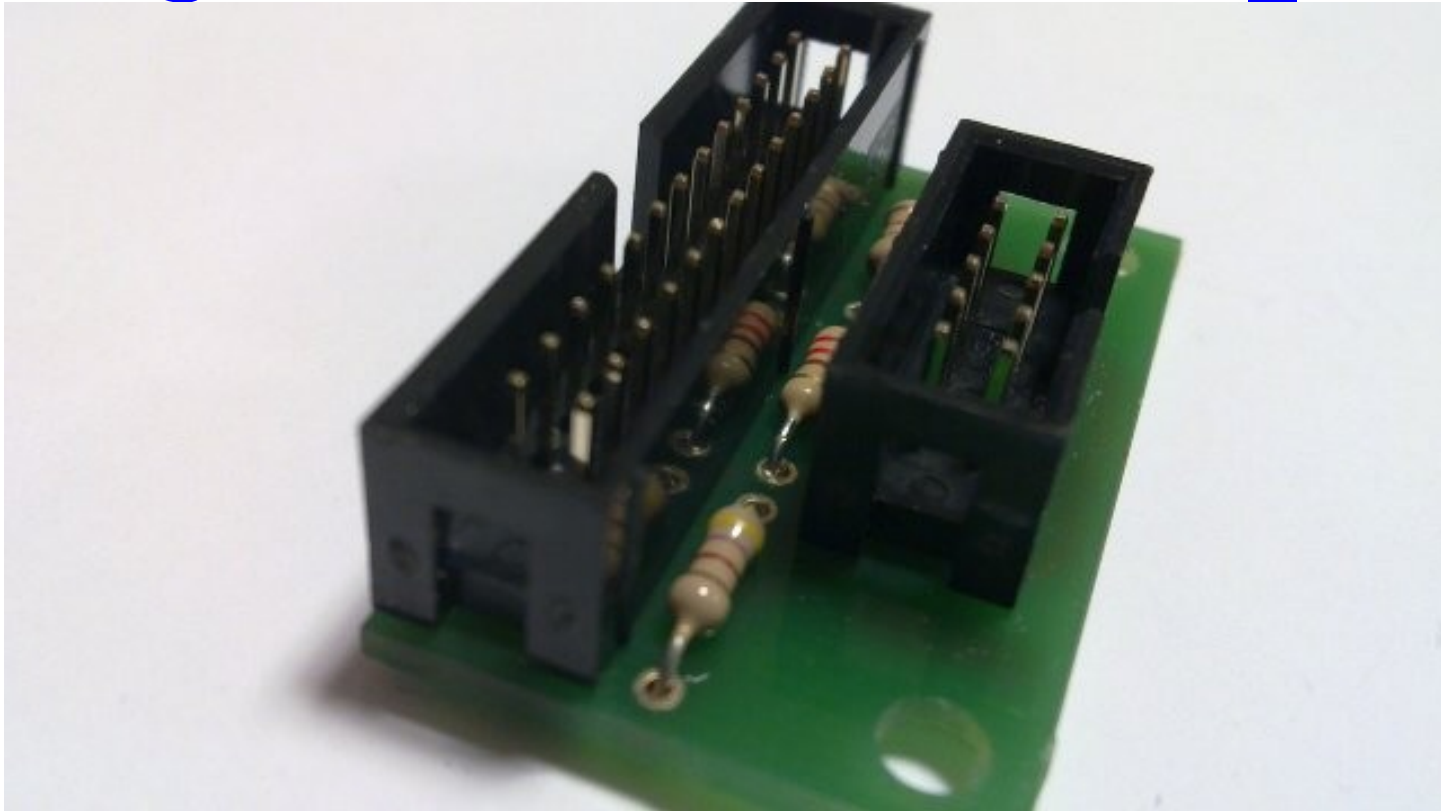


JUMPER BETWEEN PINS 1 AND 2 GIVE SUPPLY TO CONNECTED BOARD
(USB FROM COMPUTER IS ABLE TO GIVE UP TO 100mA)

Sessió pràctica de maquinari lliure

Plaques disponibles

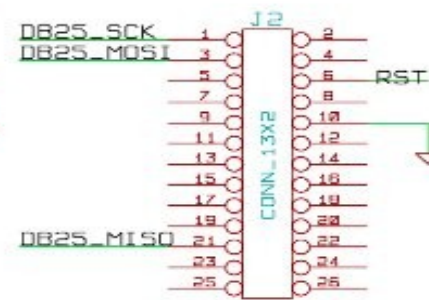
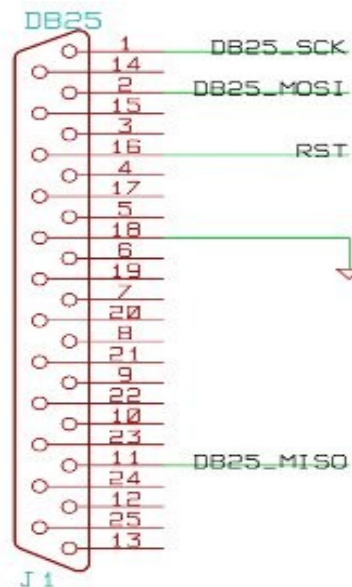
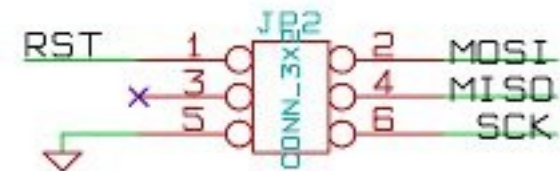
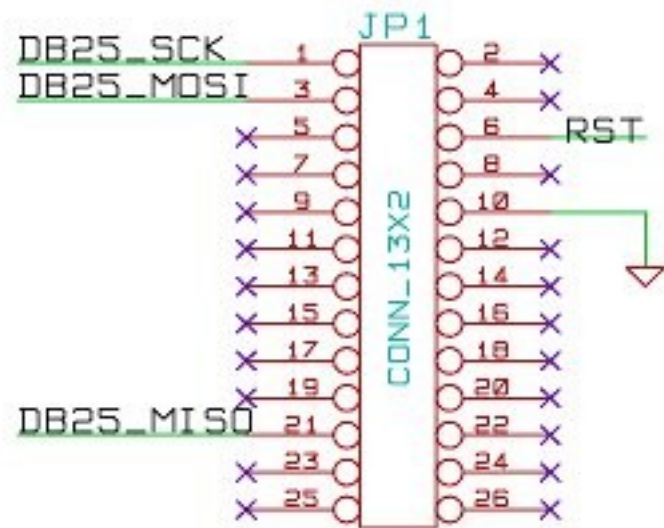
Programadors - Placa set01_02



Sessió pràctica de maquinari lliure

Plaques disponibles

Programadors - Placa set01_02



Sessió pràctica de maquinari lliure

Plaques disponibles

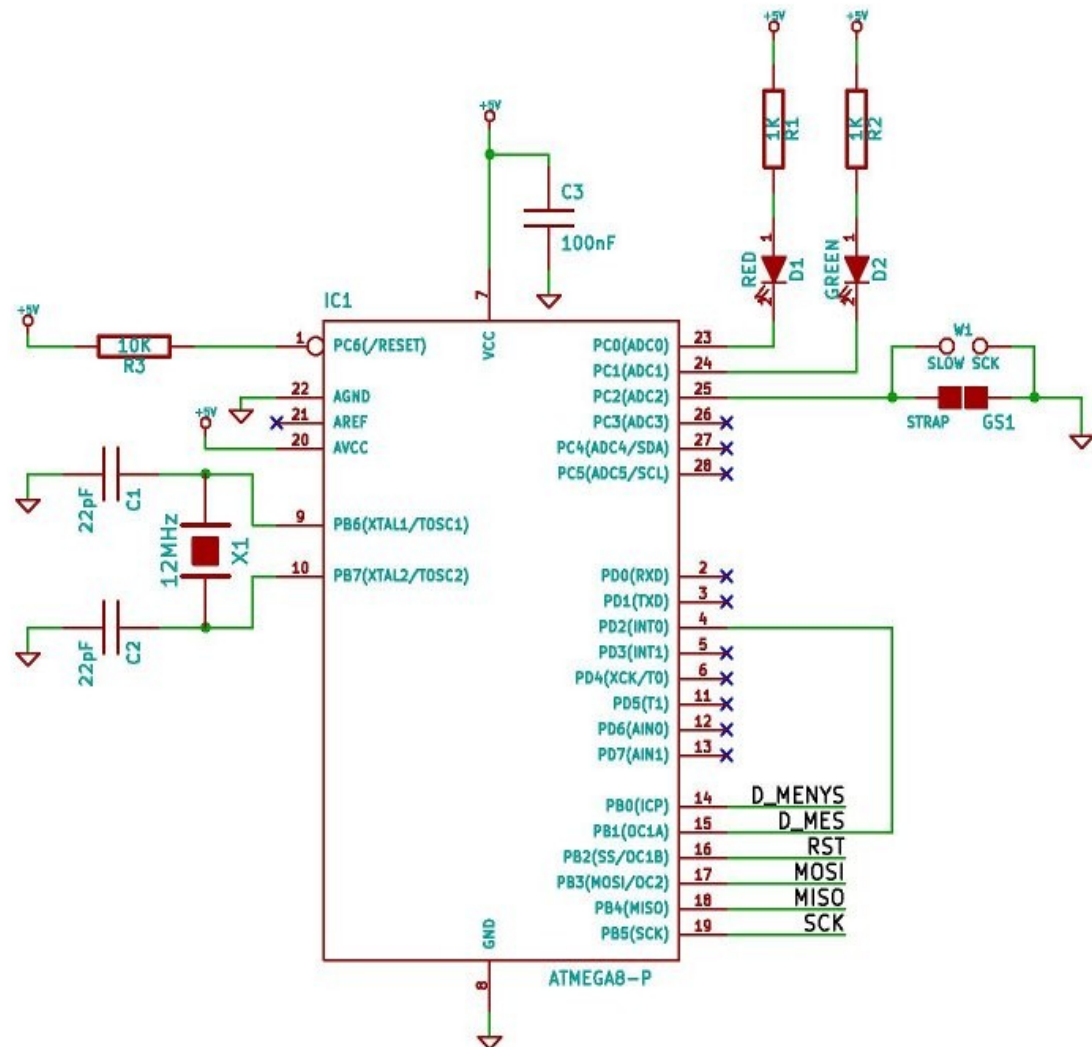
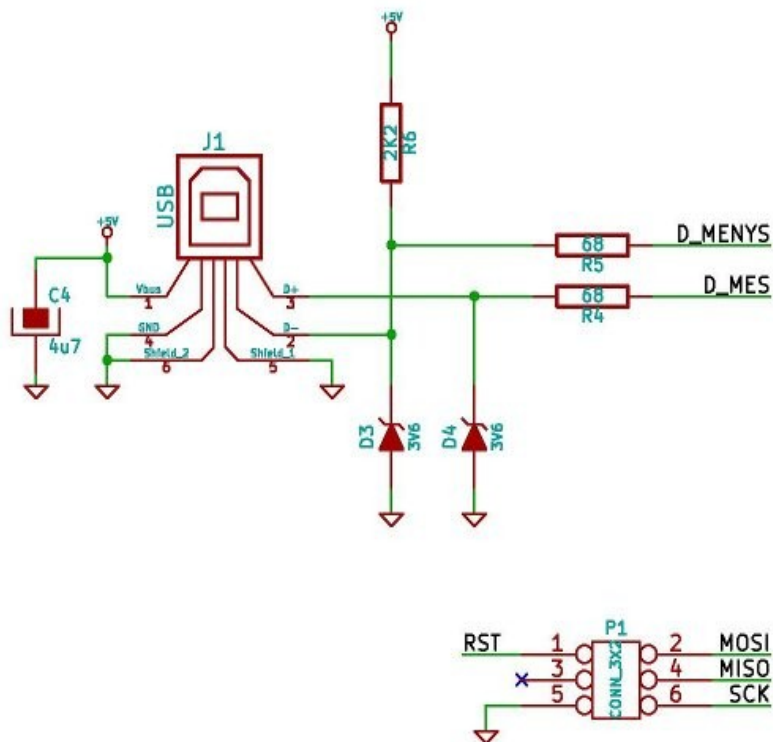
Programadors - Placa set03_09



Sessió pràctica de maquinari lliure

Plaques disponibles

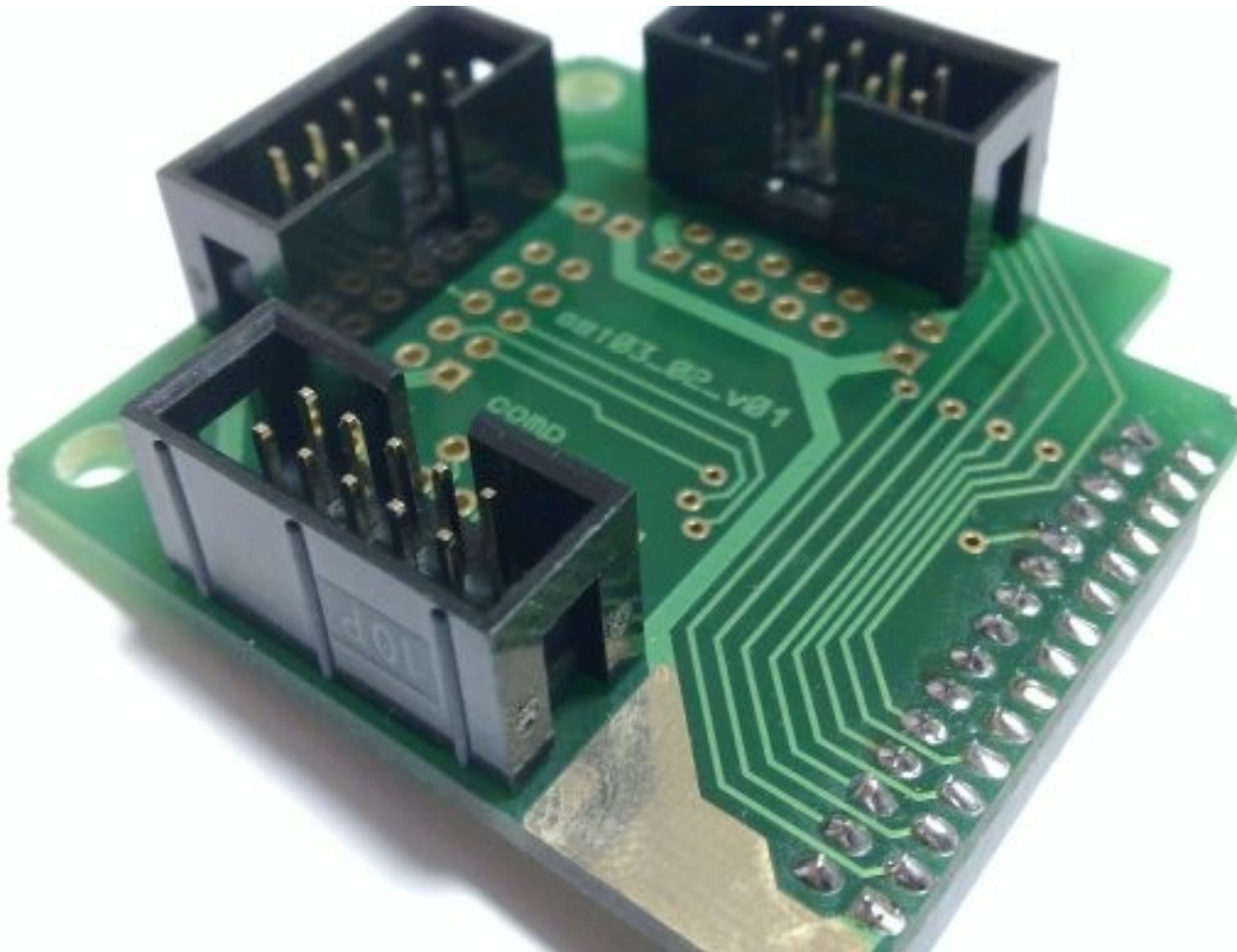
Programadors - Placa set03_09



Sessió pràctica de maquinari lliure

Plaques disponibles

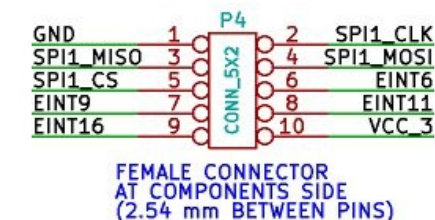
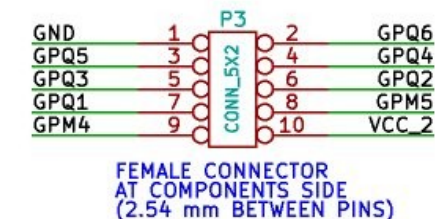
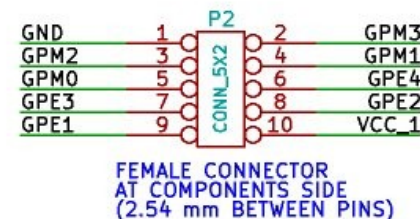
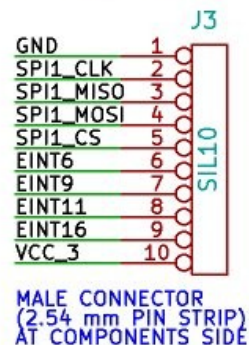
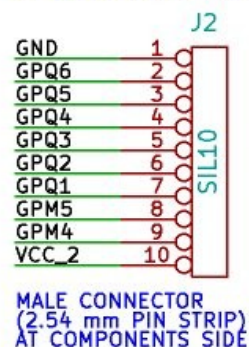
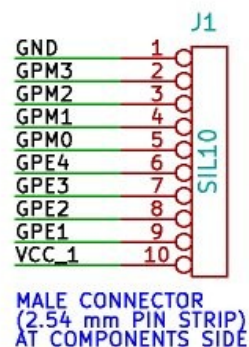
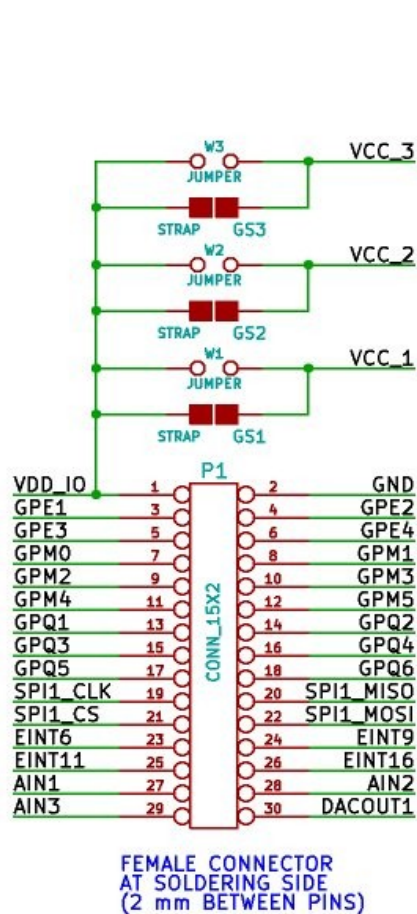
Adaptador mini6410 - Placa set03_02



Sessió pràctica de maquinari lliure

Plaques disponibles

Adaptador mini6410 - Placa set03_02



Sessió pràctica de maquinari lliure

Programació de uC de 8 bits

Exemple 1 (1/3)

Feu que surti a l'inici el missatge de text "ex001_01.c" i s'encengui el led 0 de la placa d'entrades/sortides.

```

188 void vManageCharacter(BYTE by){
189     vWriteCharacter(by);
190 }
191
192 SIGNAL (SIG_USART_RECV) {
193     BYTE by;
194
195     by = UDR0;
196     vManageCharacter(by);
197 }
198
199 int main(void){
200     vStartUp();
201
202     vWriteString ("\nex001_01.c\n");
203     while(1){ /* Infinite loop */
204         vLedIoOn(0);
205     }
206
207     return 0;
208 }

```

```

161 void vLedIoOn(BYTE byBit){
162     BYTE byMask = 0x01;
163     char nI;
164
165     for(nI = 0 ; nI < byBit ; nI++){
166         byMask <= 1;
167         if(byBit < 4){
168             PORTC |= byMask;
169         }else{
170             PORTD |= byMask;
171         }
172     }
173
174 void vLedIoOff(BYTE byBit){
175     BYTE byMask = 0x01;
176     char nI;
177
178     for(nI = 0 ; nI < byBit ; nI++){
179         byMask <= 1;
180         byMask = ~byMask;
181         if(byBit < 4){
182             PORTC &= byMask;
183         }else{
184             PORTD &= byMask;
185         }
186     }

```

Sessió pràctica de maquinari lliure

Programació de uC de 8 bits

Exemple 1 (2/3)

Feu que surti a l'inici el missatge de text "ex001_01.c" i s'encengui el led 0 de la placa d'entrades/sortides.

```

128 BYTE byReadNibblePD4_7(void){
129     BYTE by;
130
131     by = PIND;
132     by >>= 4;
133
134     return by;
135 }
136
137 BYTE byReadNibbleRawPD4_7(void){
138     BYTE by;
139
140     by = PIND;
141     by &= 0xF0;
142
143     return by;
144 }
145
146 BOOL bReadSwitch(BYTE byBit){
147     BYTE byMask = 0x01;
148     char nI;
149
150     for(nI = 0 ; nI < byBit ; nI++){
151         byMask <<= 1;
152         if(byBit < 4){
153             if(byMask & byReadNibblePC0_3())
154                 return TRUE;
155         }else{
156             if(byMask & byReadNibbleRawPD4_7())
157                 return TRUE;
158         }
159     }

```

```

99 void vWriteNibblePD4_7(BYTE byN){
100     byN <<= 4;
101     if(byN & 0x10)
102         PORTD |= 1<<PD4;
103     else
104         PORTD &= ~(1<<PD4);
105     if(byN & 0x20)
106         PORTD |= 1<<PD5;
107     else
108         PORTD &= ~(1<<PD5);
109     if(byN & 0x40)
110         PORTD |= 1<<PD6;
111     else
112         PORTD &= ~(1<<PD6);
113     if(byN & 0x80)
114         PORTD |= 1<<PD7;
115     else
116         PORTD &= ~(1<<PD7);
117 }
118
119 BYTE byReadNibblePC0_3(void){
120     BYTE by;
121
122     by = PINC;
123     by &= 0x0F;
124
125     return by;
126 }
127
128 BYTE byReadNibblePD4_7(void){
129     BYTE by;
130
131     by = PIND;
132     by >>= 4;
133
134     return by;
135 }

```

Sessió pràctica de maquinari lliure

Programació de uC de 8 bits

Exemple 1 (3/3)

Feu que surti a l'inici el missatge de text “ex001_01.c” i s'encengui el led 0 de la placa d'entrades/sortides.

```

59 void vStartUp(void){
60     vStartUART();
61     sei();          /* Enabling interruptions */
62     DDRC |= (1<<DDC3);
63     DDRC |= (1<<DDC2);
64     DDRC |= (1<<DDC1);
65     DDRC |= (1<<DDC0);    /* PC0-PC3 is enabled as output */
66     DDRD |= (1<<DDD7);
67     DDRD |= (1<<DDD6);
68     DDRD |= (1<<DDD5);
69     DDRD |= (1<<DDD4);    /* PD4-PD7 is enabled as output */
70 }
71
72 void vLitLedOn(void){
73     PORTC &= ~(1<<PC5); // it writes 0 to PC5
74 }
75
76 void vLitLedOff(void){
77     PORTC |= 1<<PC5; // it writes a 1 to PC5
78 }
79
80 void vWriteNibblePC0_3(BYTE byN){
81     if(byN & 0x01)
82         PORTC |= 1<<PC0;
83     else
84         PORTC &= ~(1<<PC0);
85     if(byN & 0x02)
86         PORTC |= 1<<PC1;
87     else
88         PORTC &= ~(1<<PC1);
89     if(byN & 0x04)
90         PORTC |= 1<<PC2;
91     else
92         PORTC &= ~(1<<PC2);
93     if(byN & 0x08)
94         PORTC |= 1<<PC3;
95     else
96         PORTC &= ~(1<<PC3);
97 }

```

```

10 #include <avr/io.h>
11 #include <avr/interrupt.h>
12 #include <avr/signal.h>
13 #include <inttypes.h>
14
15 #define F_OSC 3686400 /* Crystal frequency */
16 #define UART_BAUD_RATE 115200 /* UART baud rate */
17 #define UART_BAUD_CALC(UART_BAUD_RATE,F_OSC) ((F_OSC)/((UART_BAUD_RATE)*16L)-1)
18
19 #ifndef BOOL
20     #define BOOL unsigned char
21 #endif
22 #ifndef BYTE
23     #define BYTE unsigned char
24 #endif
25 #ifndef WORD
26     #define WORD unsigned int
27 #endif
28 #ifndef TRUE
29     #define TRUE 1
30 #endif
31 #ifndef FALSE
32     #define FALSE 0
33 #endif
34
35 void vWriteCharacter(unsigned char uc){
36     while(!(UCSR0A & (1<<UDRE0))); /* It waits until UDR is ready */
37     UDR0 = uc; /* Sending character */
38 }
39
40 void vWriteString(char *sz){
41     while (*sz) {
42         vWriteCharacter(*sz);
43         sz++;
44     }
45 }
46
47 void vStartUART(void){
48     // Telling UART baud rate
49     UBR0H = (uint8_t)(UART_BAUD_CALC(UART_BAUD_RATE,F_OSC)>>8); /* !!!! */
50     UBR0L = (uint8_t)UART_BAUD_CALC(UART_BAUD_RATE,F_OSC);
51
52     // Enabling Tx and Rx ( and their interruption )
53     UCSRB = (1<<RXEN0) | (1<<TXEN0) | (1<<RXCIF0);
54
55     // 8 bits of data, without parity and 1 stop bit
56     UCSRC = (1<<UMSEL01) | (3<<UCSZ00);
57 }
58
59 void vStartUp(void){
60     vStartUART();

```

Exercicis amb solucions : www.binefa.cat/dpesti/

Sessió pràctica de maquinari lliure

Programació de uC de 8 bits

Exemple 2

Feu que surti a l'inici el missatge de text "ex08.c" i que al prémer el polsador connectat al bit 4 s'encengui el led 6 de la placa d'entrades/sortides connectada al port 2. I al deixar de prémer-ho s'apagui. En aquest mateix exercici, al prémer el polsador connectat al bit 5 s'ha d'encendre el led 7 de la placa d'entrades/sortides connectada al port 2 i al deixar de prémer-ho s'apagui.

```

200 int main(void){
201     vStartUp();
202
203     vWriteString ("\nex08.c\n");
204     while(1){ /* Infinite loop */
205         if(PINB & 4)
206             PORTB &= ~( 1<< PB4);
207         else
208             PORTB |= ( 1<< PB4);
209
210         if(PINB & 8)
211             PORTB &= ~( 1<< PB5);
212         else
213             PORTB |= ( 1<< PB5);
214
215     }
216
217     return 0;
218 }

```

```

58 void vStartUp(void){
59     vStartUART();
60     sei(); /* Enabling interruptions */
61     DDRB |= (1<<DDC3);
62     DDRB |= (1<<DDC2);
63     DDRB &= ~(1<<DDC1);
64     DDRB &= ~(1<<DDC0); /* PC0-PC3 is enabled as input */
65     DDRB |= ~(1<<DDD7);
66     DDRB |= (1<<DDD6);
67     DDRB &= ~ (1<<DDD5);
68     DDRB &= ~ (1<<DDD4); /* PD4-PD7 is enabled as output */
69 }
70
71 void vLitLedOn(void){
72     PORTB &= ~(1<<PB5);
73     PORTB &= ~(1<<PB4); // it writes 0 to PC5
74 }
75
76 void vLitLedOff(void){
77     PORTB |= 1<<PB5;
78     PORTB |= (1<<PB4); // it writes a 1 to PC5
79 }

```

Sessió pràctica de maquinari lliure

Programació de uC de 8 bits

Exemple 3

Feu que surti a l'inici el missatge de text "cadena02.c" i preguntí per un nombre entre 0 i 255. En cas de que el nombre entrat sigui coherent es visualitzarà en binari per la placa de leds (0 : led apagat, 1: led encès).

```

252 int main(void){
253     BYTE byN;
254
255     vStartUp();
256     vWriteString ("\ncadena02.c\n\nEntra un nombre entre 0 i 255 : ");
257     while(1){ /* Infinite loop */
258         if(bStringHasBeenRead){
259             vWriteString((char *)szBuffer);
260             vWriteCharacter ('\n');
261             if(bParse((char *)szBuffer)){
262                 byN = nString2Number((char *)szBuffer);
263                 if(byN < 256){
264                     vWritePort(byN);
265                     vWriteString((char *)szBuffer);
266                     vWriteString(" escrit a la placa de leds.\n");
267                 }
268             }
269             vWriteString ("\n\nEntra un nombre entre 0 i 255 : ");
270             bStringHasBeenRead = FALSE;
271         }
272     }
273
274     return(0);
275 }

```

```

227 SIGNAL (SIG_USART_RECV) {
228     static unsigned char ucCounter = 0;
229     unsigned char uc;
230
231     uc = UDR0;
232     if((uc == 10) || (uc == 13)) {
233         szBuffer[ucCounter] = '\0';
234         ucCounter = 0;
235         bStringHasBeenRead = TRUE;
236     }else{
237         if(ucCounter < MAX_STRING_LENGTH){
238             if(!bStringHasBeenRead){
239                 szBuffer[ucCounter] = uc;
240                 ucCounter++;
241                 szBuffer[ucCounter] = '\0';
242             }
243         }else{
244             szBuffer[ucCounter] = '\0';
245             ucCounter = 0;
246             bStringHasBeenRead = TRUE;
247         }
248     }
249     vManageCharacter(uc);
250 }

```

Sessió pràctica de maquinari lliure

Programació de uC de 8 bits

Altres exemples

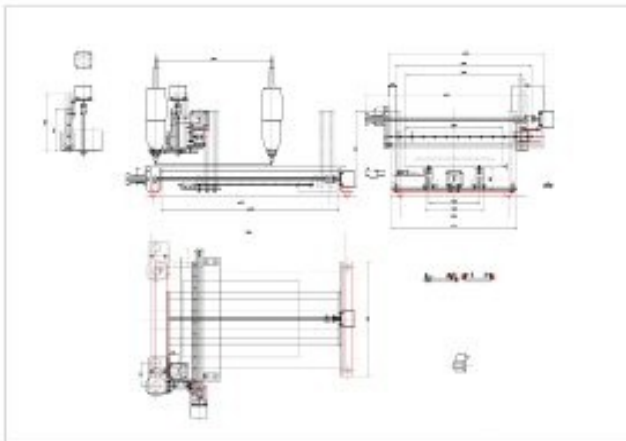
- Programari allotjat a sourceforge.net :
(<http://sourceforge.net/projects/qtncn/files/>)



Home 

Name ▾	Modified ▾	Size ▾	Downloads ▾
 Desktop Software	2011-12-21		
 FirmwareAtmega	2011-12-21		

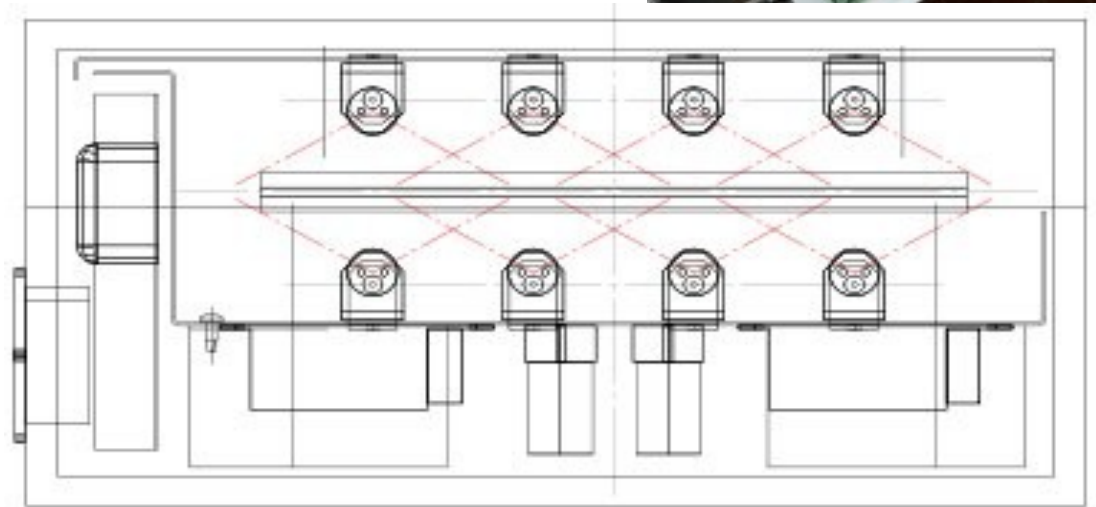
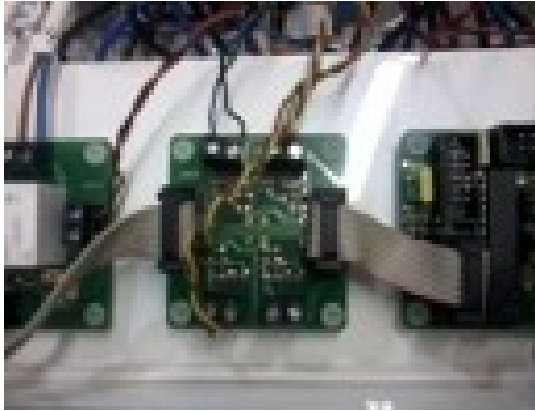
Totals: 2 Items



Sessió pràctica de maquinari lliure

Programació de uC de 8 bits

Altres exemples



Torn de preguntes ...



... i sessió pràctica.

COMIAT

Presentació descarregable a :

<https://dl.dropbox.com/u/65254823/20121108MaqLliurePractic.pdf>

Correu electrònic de contacte : jordibinefa@electronics.cat

twitter



<https://twitter.com/JordiBinefa>



<http://es.linkedin.com/pub/jordi-binefa/13/717/90b>

Plaques disponibles a <http://botiga.electronics.cat/>

Moltes gràcies per la vostra assistència