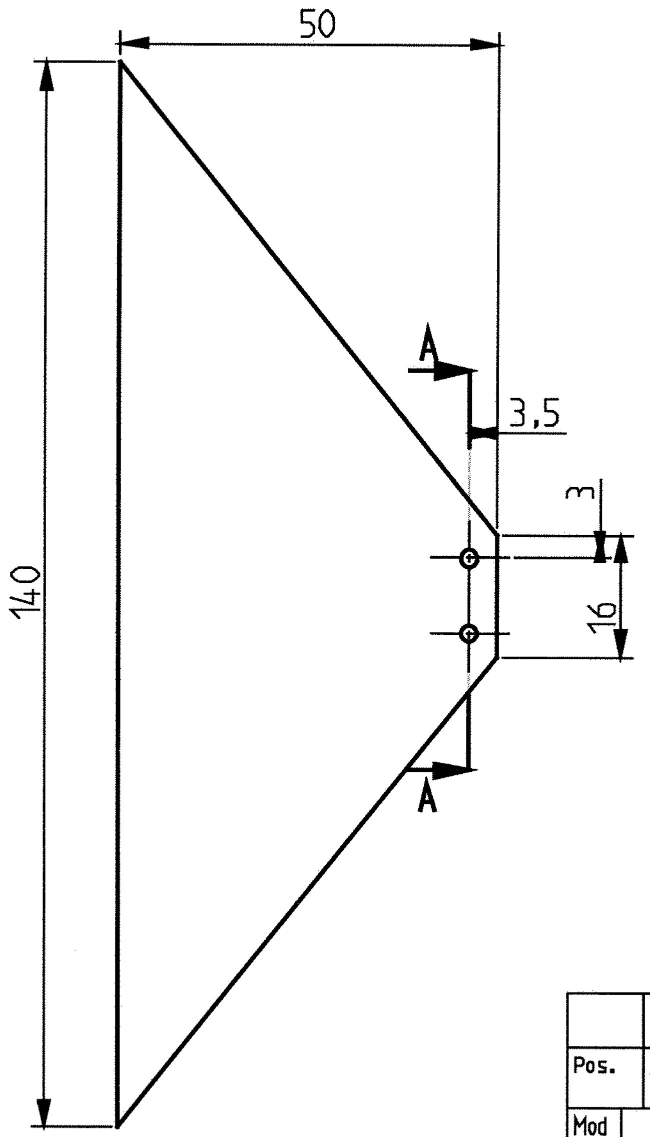
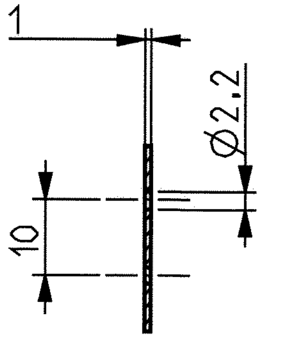
Appendix A

Mechanical drawings



SECTION A-A



Pos.	Quantite	Unite	N° d'identification	Denomination / Caracteristiques
	1			CAUDAL1 / PVC

Mod	Mod	Dessine	Daisy Lachat
		Controle	
		Conf normes	
		Bon execution	

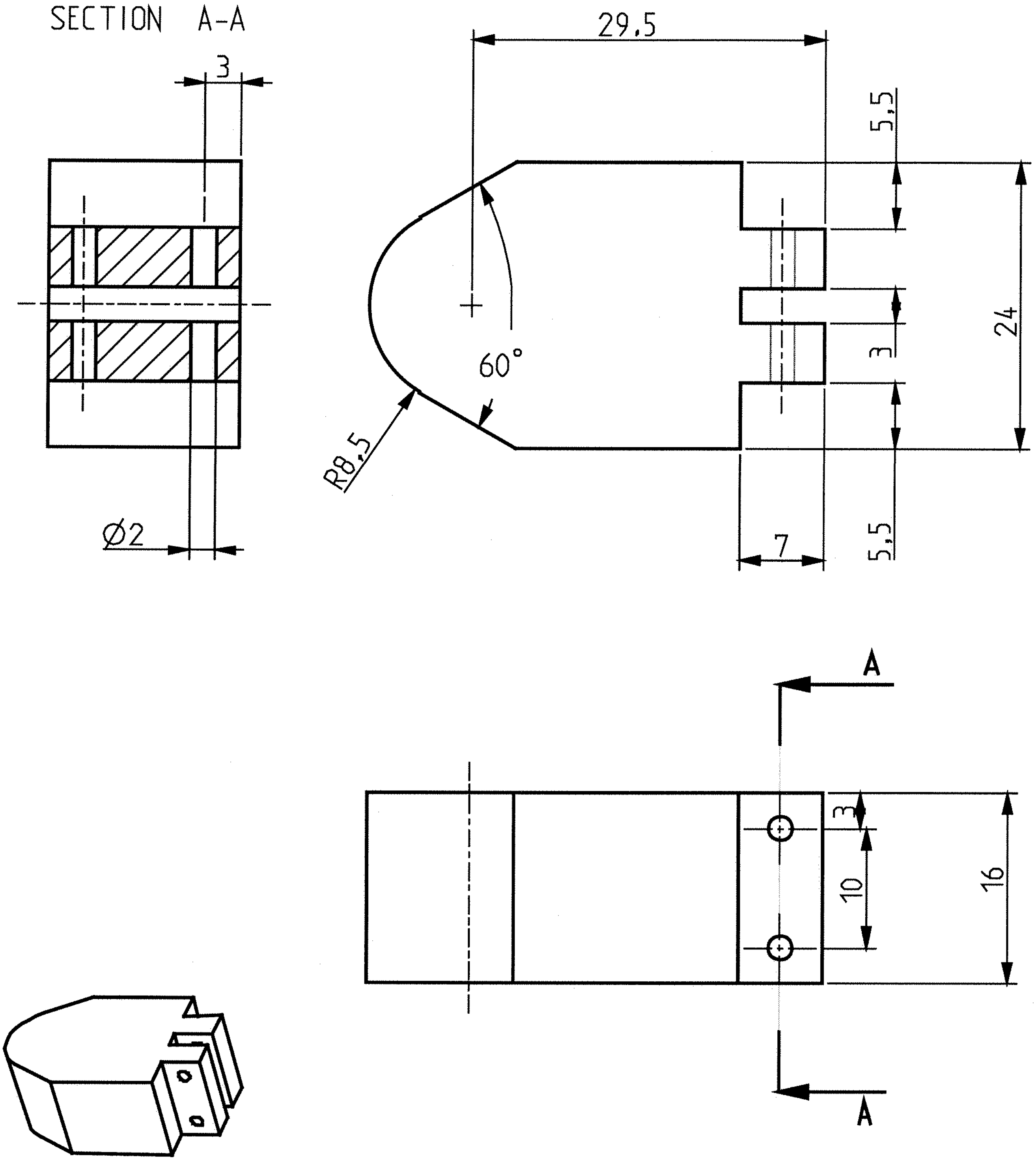
Sans nomenclature separee		N° commande		
Nomenclature sep de meme N°		Origine	Nb feuilles 1	Feuille N° 1
Nomenclature sep de N° different	N° ident	Remplace		

Echelle
1:1

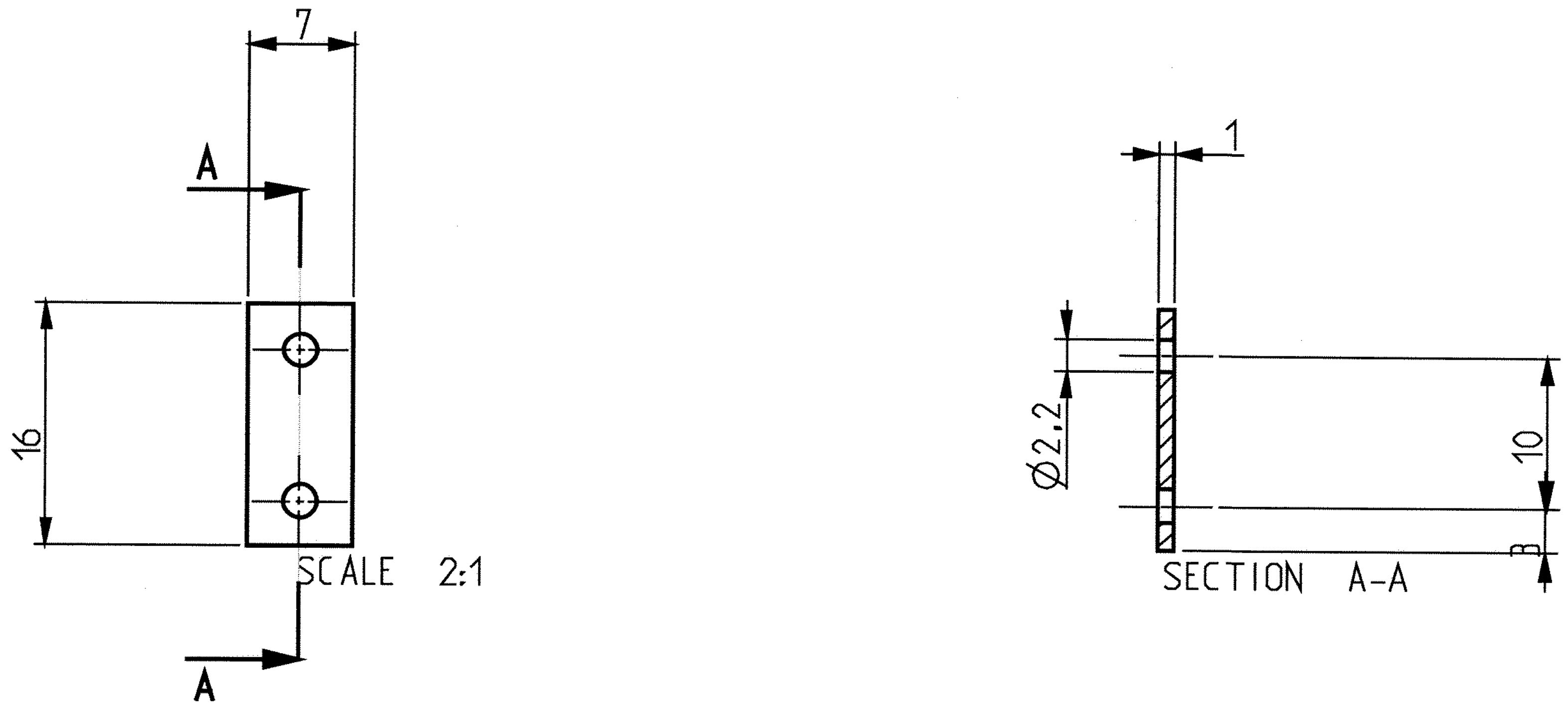
EPFL

CAUDAL1

N° de dessin



Pos.	Quantite 1	Unite	N° d'identification	Denomination / Caracteristiques ENTRAINEMENT-CAUDAL / PUR allégé		
Mod		Mod		Dessine	Daisy Lachat	Echelle 2/1
				Controle		
				Conf normes		
				Bon execution		
Sans nomenclature separee				N° commande		
Nomenclature sep de meme N°				Origine	Nb feuilles 1	Feuille N° 1
Nomenclature sep de N° different	N° ident			Remplace		
EPFL				ENTRAINEMENT-CAUDAL	N° de dessin AMPHI-012	



Pos.	Quantite	Unite	N° d'identification	Denomination / Caracteristiques
	2			CALE-CAUDAL1 / PVC

Mod	Mod	Dessine	Daisy Lachat
		Controle	
		Conf normes	
		Bon execution	

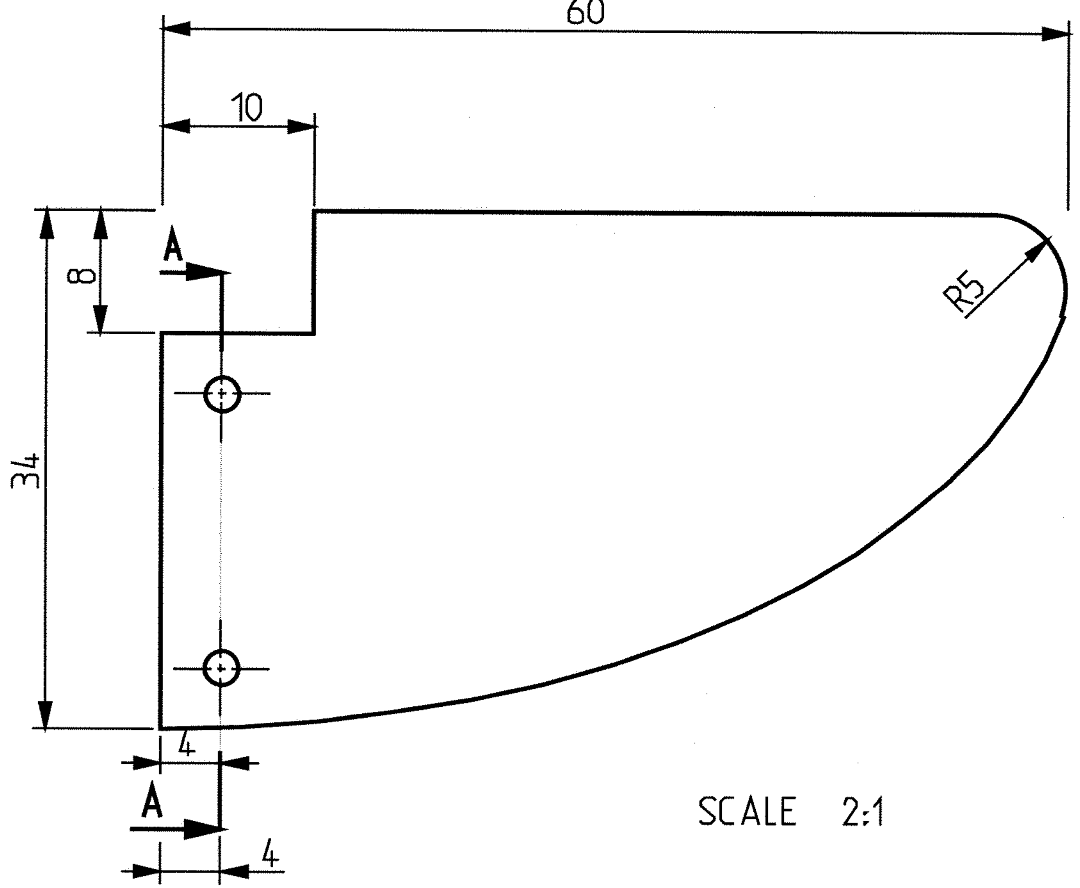
Sans nomenclature separee		N° commande	
Nomenclature sep de meme N°		Origine	
Nomenclature sep de N° different	N° ident	Remplace	

Echelle	Nb feuilles	Feuille N°
1:1	1	1

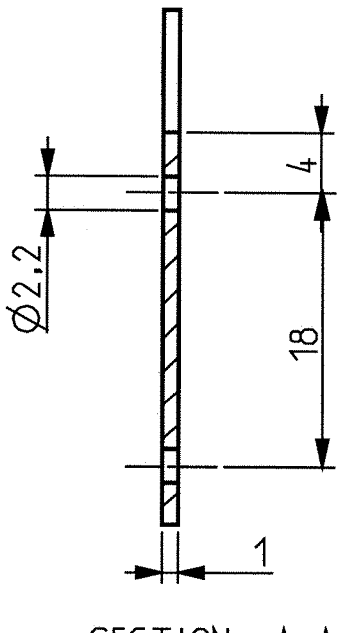
N° de dessin

EPFL

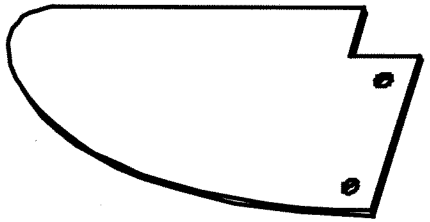
CALE-CAUDAL1



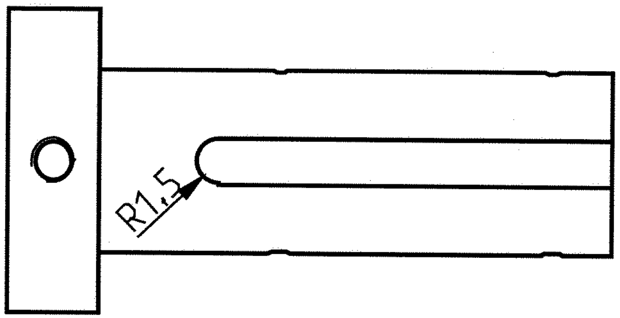
SCALE 2:1

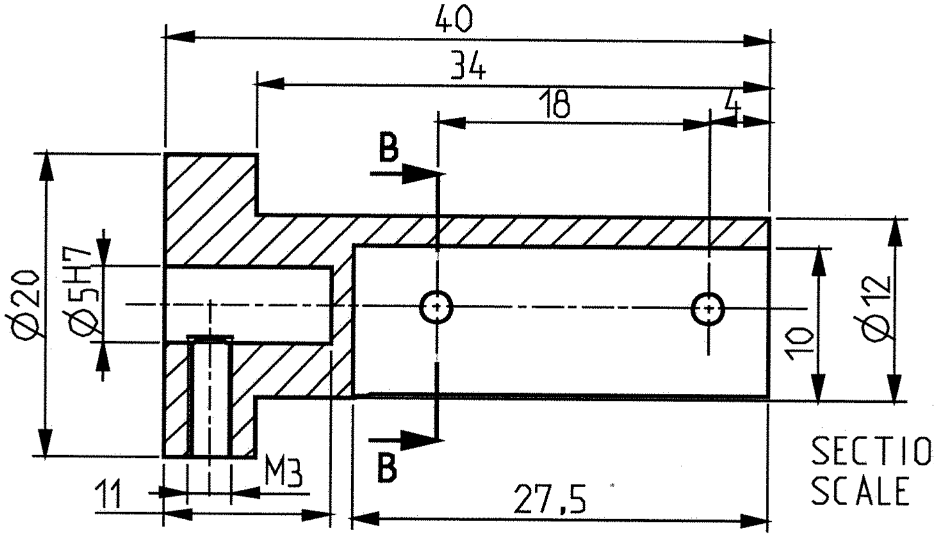


SECTION A-A

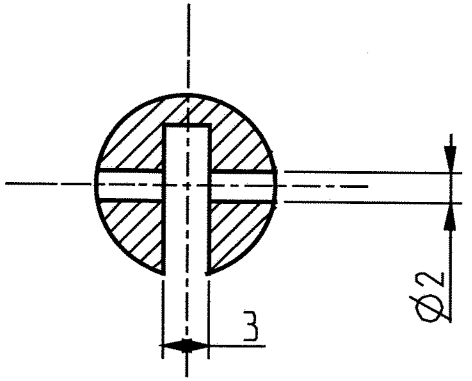


Pos.	Quantité 2	Unite	N° d'identification	Denomination / Caracteristique PECTORAL1 / PERBUNAN 75 shore A			
Mod		Mod		Dessine	Daisy Lachat		Echelle 1:1
				Controle			
				Conf normes			
				Bon execution			
Sans nomenclature separee				N° commande			
Nomenclature sep de meme N°				Origine		Nb feuilles 1	Feuille N° 1
Nomenclature sep de N° different	N° ident			Remplace			
	PECTORAL1					N° de dessin	

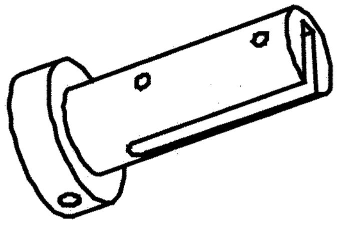




SECTION A-A
SCALE 2:1



SECTION B-B



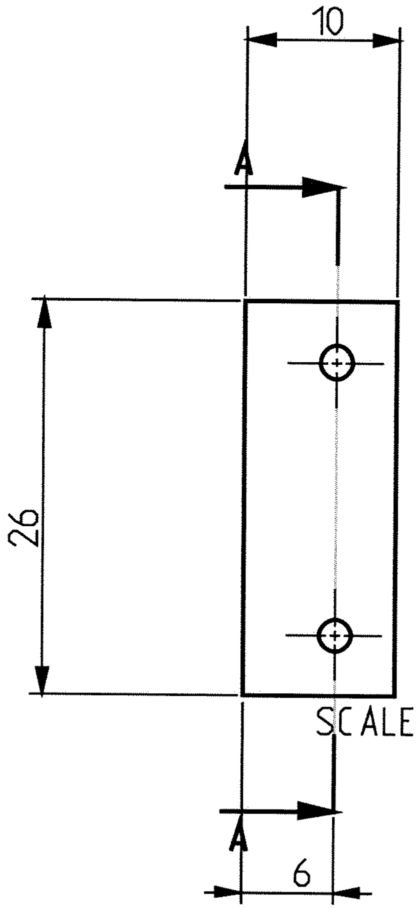
Pos.	Quantit 2	Unite	N° d'identification	Denomination/Caracteristiques ENTRAINEMENT-PECTORAL / POM

Mod	Mod	Dessine	Daísy Lachat		Echelle 1:1
		Controle			
		Conf normes			
		Bon execution			
Sans nomenclature separee		N° commande			
Nomenclature sep de meme N°		Origine		Nb feuilles 1	Feuille N° 1
Nomenclature sep de N° different	N° Ident	Remplace			

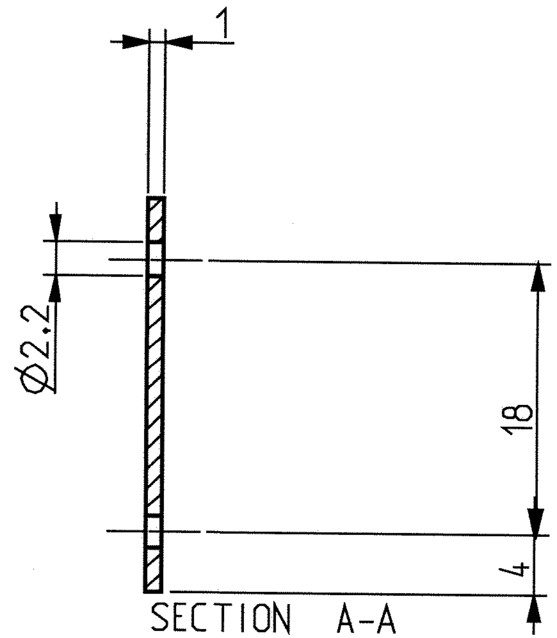
EPFL

ENTRAINEMENT-PECTORAL

N° de dessin



SCALE 2:1



SECTION A-A

Pos.	Quantite	Unite	N° d'identification	Denomination / Caracteristiques
	2			CALE-PECTORAL1 / PVC

Mod

Mod

		Echelle
Dessine	Daisy Lachat	1:1
Controle		
Conf normes		
Bon execution		
N° commande		
Origine		Nb feuilles 1
Remplace		Feuille N° 1

Sans nomenclature separee

Nomenclature sep de meme N°

Nomenclature sep de N° different | N° ident

EPFL

N° de dessin

CALE-PECTORAL1

Appendix B

Software

B.1 PIC's program

```
/*-----------------------------------------------------------------------------
 * FishRobot.c
 *
 * Author: Daisy Lachat <daisy.lachat@epfl.ch>
 * Date: 25-JUN-05
 * Release: 1.0
 * Purpose: fish robot make a serie of motion's sequences
 *---------------------------------------------------------------------------*/

#include <16F876A.h>
#include "FishRobot.h"
#include "regs.h"
#include "sin.h"
#include "sequences.h"

/*-----------------------------------------------------------------------------
 * Constantes
 *---------------------------------------------------------------------------*/

/* Device control registers (generic) */
const uint8 gen_regs[24][2] = {
  {24, 0},                      /*number of items*/
  {REG_MODE,              MODE_IDLE},
  {REG_SETPOINT_SOURCE, SETPOINT_SRC_I2C},
  {REG_CONTROL_TYPE,      CONTROL_POS},
  {REG_ERROR_FLAGS,       0},
  {REG_SETPOINT,          0},
  {REG_POSITION,          0},
  {REG_FILTER_POS,        1},
  {REG_SAMPLE_TIME_H,     8},
  {REG_SAMPLE_TIME_L, 124},

  {REG_HW_OPTIONS,        0}, /*power up in idle mode*/
  {REG_SW_OPTIONS,        3}  /*enable antireset windup and separate derivative
                                     *term, disable softstop*/
};

/* Device control registers (body element specific) */
const uint8 body_regs[7][2] = {
  {7, 0},                       /*number of items*/
  {REG_KD_POS,            25},
  {REG_KD_DIV_POS,        0},
  {REG_KP_POS,            10},
  {REG_KP_DIV_POS,        0},
  {REG_DIVIDER_SPEED,     5},
  {REG_DIVIDER_POS,       8}
};

/* Device control registers (head element specific) */
const uint8 head_regs[7][2] = {
  {7, 0},                       /*number of items*/
  {REG_KD_POS,           10},
  {REG_KD_DIV_POS,        0},
  {REG_KP_POS,           10},
  {REG_KP_DIV_POS,        0},
  {REG_DIVIDER_SPEED,     5},
  {REG_DIVIDER_POS,       9}
};

/* I²C addresses of the modules */
const uint8 caudal     = 0x01;
const uint8 pect_left  = 0x07;
const uint8 pect_right = 0x1f;

const uint8 addr_fin[FINS]= {caudal, pect_left, pect_right};


/*-----------------------------------------------------------------------------
```

```
 * Global variables
 *-----------------------------------------------------------------------------*/

float time;
uint8 num_seq;           /*number of the current sequence*/

/*Parameters of sines available for the current sequence*/
uint8 sine_ampl[FINS];  /*amplitude of sine*/
float sine_phas[FINS];  /*phase of sine*/
uint8 sine_freq[FINS];  /*frequencies of sine mult by 20(resolution=0.05Hz)*/
int8  sine_offs[FINS];  /*offsets of sine*/

/*-------------------------------------------------------------------------------
 * Interrupt service routine
 *-----------------------------------------------------------------------------*/

/*----------------------------------------------------
 * Increment time and update the number of the
 * current sequence. Interrupt every 546.621 [us].
 * out: time, num_seq
 *--------------------------------------------------*/
#INT_TIMER2
int_timer_2(){
 uint8 i;

 time+=DELTA_T;
 if(time >= tab_time[num_seq])
   num_seq++;
}

/*-------------------------------------------------------------------------------
 * Functions
 *-----------------------------------------------------------------------------*/

/*----------------------------------------------------
 * Write a value in a register of a module.
 * Value is unsigned but we can send a signed number
 * and it would be interpreted correctly.
 * in: addr : adress of the module
 *     reg  : adress of the register
 *     value: value to write
 * return 1 if the register is set, 0 otherwise
 *--------------------------------------------------*/
int1 set_reg_value(uint8 addr, uint8 reg, uint8 value){
  uint8 k;
  int1 ack=0; /*ack==1 -> no ack from slave*/

  for(k=0 ; k < 5 ; k++){
    i2c_start();
    ack = i2c_write(addr<<1);
    if(ack==0) ack = i2c_write(reg);
    if(ack==0) ack = i2c_write(value);
    i2c_stop();

    if(ack==0)
      return 1;  /*value transmitted*/
  }

  PDEBUG("set_reg_value: value %u not wrote in reg 0x%x of module 0x%x\r\n",
         value, reg, addr);
  return 0;      /*value not transmitted*/
}

/*----------------------------------------------------
 * Read a register of a module. Value is positive but
 * can be interpreted as a signed number if requiered
 * (ex: position) and it would be interpreted correctly.
 * in: addr: address of the module
 *     reg:  address of the register
 * return:   the value read
 *----------------------------------------------------*/
uint8 get_reg_value(uint8 addr, uint8 reg){
  uint8 value, k;
  int1 ack; /*ack==1 -> no ack from slave*/

  for(k=0 ; k < 5 ; k++){

    i2c_start();
    ack = i2c_write(addr<<1);
    ack = ack || i2c_write(reg);
    i2c_start();  /*repeated start*/
    ack = ack || i2c_write((addr<<1) | 1); /* reading */
    value=i2c_read(0); /*0 = no ack from master*/
    i2c_stop();

    if(ack==0)
      break;
  }
  if(ack!=0)
    PDEBUG("get_reg_value: reg 0x%x not read from module 0x%x\r\n", reg, addr);

  return value;
}

/*------------------------------------------------------
 * Set sine configuration for the nth fin for the cur-
 * rent sequence.
 * in: a: amplitude, p: phase, f: frequency, o: offset
 *     n: number of the fin
 *----------------------------------------------------*/
/*  n = number of the fin */
void set_sine(uint8 a, float p, uint8 f, int8 o, uint8 n){

  sine_ampl[n]=a;
  sine_offs[n]=o;
  sine_phas[n]=p;
  sine_freq[n]=f;
}

#ifdef GRAPH
#inline
/*------------------------------------------------------
 * Send a float to RS232 in graphique mode
 * in: t: float to transmit
 *----------------------------------------------------*/
void send_float(float t)
{
  uint8 i;
  for (i=0; i<4; i++) putc(*(((uint8 *) &t) + i));
}
#endif

/*------------------------------------------------------
 * Set each fin in normal mode
 *----------------------------------------------------*/
void start_fish(){
  uint8 i;
  for(i=0 ; i<FINS ; i++) set_reg_value(addr_fin[i], REG_MODE, MODE_NORMAL);
}

/*------------------------------------------------------
 * Set each fin in idle mode
 *----------------------------------------------------*/
void stop_fish(){
  uint8 i;
  for(i=0 ; i<FINS ; i++) set_reg_value(addr_fin[i], REG_MODE, MODE_IDLE);
```

```
}

/*------------------------------------------------------
 * Set the configuration to sequence number seq
 * in: seq: number of the sequence to configure
 * mod: sine_ampl, sine_phas, sine_freq, sine_offs
 *-----------------------------------------------------*/
void set_seq(uint8 seq){
  static uint8 num=0;
  uint8 j;
  PDEBUG("set_seq %d à t=%f\r\n",seq,time);

  for(j=0 ; j<FINS ; j++) {
    /*Set the fin in position of offset*/
    set_reg_value(addr_fin[j], REG_SETPOINT, tab_offset[seq][j]);
    set_sine(tab_ampl[seq][j], M_PI*((float)(tab_phas[seq][j])),
    tab_freq[seq][j], tab_offset[seq][j], j);
  }
}


/*------------------------------------------------------
 * Check if number of sequence have to be change and
 * call set_seq if required.
 * return: 1 if every sequence have been passed
 *         0 otherwise
 *-----------------------------------------------------*/
int1 check_seq(void){
  uint8 i;
  /*flag to verify if the current sequence has already been configured*/
  static uint8 internal_flag = 255;

  for(i = 0; i<SEQUENCES+1 ; i++){
    /*set the seq only one times per seq*/
    if((num_seq==i)&&(internal_flag !=i)){
      internal_flag = i;
      if(i==SEQUENCES)
        return 1;   /*each sequence have been passed*/
      set_seq(i);
      return 0;
    }
  }
  return 0;
}

/*------------------------------------------------------
 * Set the setpoint of module i at the right value
 * in:  i: number of the module to set
 *-----------------------------------------------------*/
void position_motor(uint8 i){
  uint8 val;
  uint32 num;
  float w;

  /*if the frequency is null, it isn't a sine*/
  if(sine_freq[i] != 0){
    /*compute the sine value*/
    num = ( 255 * time * sine_freq[i] * 0.05 )+ (sine_phas[i]*0.5*255/M_PI);
    num -=((uint32)(num/255))*255; /*scaling [0;255]*/
    val = sine_ampl[i] * (float)(lookup_sin[num])/100.0 + sine_offs[i];

    set_reg_value(addr_fin[i], REG_SETPOINT, val);

    #ifdef GRAPH
    send_float(time);   /*send time to rs232 (float)  */
    putc(val);          /*send consigne to rs232 (int) */
    /*send position of the motor to rs232 (int)*/
    putc(get_reg_value(addr_fin[i], REG_POSITION));
    putc(2);            /*sync*/
    #endif
  }
  else{
    #ifdef GRAPH
    send_float(time);   /*send time to rs232 (float)  */
    val=get_reg_value(addr_fin[i], REG_SETPOINT);
    putc(val);          /*send consign to rs232 (int) */
    /*send position of the motor to rs232 (int)*/
    putc(get_reg_value(addr_fin[i], REG_POSITION)); //for checking position
    putc(2);            //sync
    #endif
  }
}

#ifdef GRAPH
/*-------------------------------------------------------
 * Send configurations value to RS232 for graphique mode
 *-----------------------------------------------------*/
void init_graph(void){
  delay_ms(5000); /*wait to let time for open the prog on pc*/
  putc(2);        /*begining*/
  putc(FINS);     /*number of couple (float, int) before return */
  putc(2);        /*number of int variables to printf after the float variable*/
  putc(2);        /*end of init*/
}
#endif

/* Macro to send configuration registers to a single module */
#define send_config(addr, tab); for(j=1 ; j<tab[0][0] ; j++)
{set_reg_value(addr, tab[j][0], tab[j][1]);}

/*-------------------------------------------------------
 * Send registers table to initialize modules
 *-----------------------------------------------------*/
void init_fins(void){
  uint8 i, j;
   /* general config for all fins */
  for(i=0 ; i<FINS ; i++)
    send_config(addr_fin[i], gen_regs);

  /* specific config for caudal fin */
  send_config(caudal, body_regs);

  //specific config for pectoral fins
  send_config(pect_left, head_regs);
  send_config(pect_right, head_regs);
}

/*-----------------------------------------------------------------------------
 * Main function
 *---------------------------------------------------------------------------*/
void main(){

  uint8 i, j, cnt=0;
  PDEBUG("Begin\r\n");
  delay_ms(1000);  //wait that modules are powered

  time = 0;
  num_seq = 0;

  setup_counters(RTCC_INTERNAL,RTCC_DIV_2);
  setup_timer_2(T2_DIV_BY_16,10,16);
  enable_interrupts(INT_TIMER2);

  output_low(LED);      /*led off*/

  init_fins();
  delay_ms(2000);  /*time for placing fins manually*/

  #ifdef GRAPH
  init_graph();
  #endif

  enable_interrupts(GLOBAL);

  /*start to move*/
  start_fish();

  /*main loop*/
  while(1){
    for(i=0 ; i<FINS ; i++)  position_motor(i);
    if(check_seq()==1){
      time = 0;
      num_seq = 0;
      PDEBUG("End of sequences. One more time\r\n");

      #ifdef GRAPH
      break; /*in graphic mode, the sequences are execute only one time*/
      #endif
    }
  }

  /*stop to move*/
  stop_fish();

  #ifdef GRAPH
  /*close RS232 communication*/
  for(i=0 ; i<4 ; i++){
    send_float(0);
    putc(0);
    putc(0);
  }
  #endif

  PDEBUG("End\r\n");
}
```

```
/*-------------------------------------------------------------------------------
 * FishRobot.h
 *
 * Author: Daisy Lachat <daisy.lachat@epfl.ch>
 * Date: 25-JUN-05
 * Release: 1.0
 * Purpose: header file of FishRobot.c
 *-----------------------------------------------------------------------------*/
#ifndef __FISHROBOT_H
#define __FISHROBOT_H

#use delay(clock=19660000)             /* Set clock to 19.66 MHz*/
#fuses HS,NOLVP,NOWDT,NOBROWNOUT
#use I2C(MASTER,Slow,scl=PIN_C3,sda=PIN_C4)

/* debug define */
//#define DEBUG
//#define GRAPH

#ifdef DEBUG
  #define PDEBUG  printf
#else
  #define PDEBUG
#endif

#ifdef DEBUG
  #use rs232(baud=38400,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)
#endif

#ifdef GRAPH
  #ifndef DEBUG
    #use rs232(baud=38400,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)
  #endif
#endif

/* constantes */
#define M_PI         3.141592654
#define DELTA_T      0.000546621
#define FINS  3

/* Define LED */
#define LED                  PIN_C2
#define PINNR_LED              2

/* Typedefs */
typedef unsigned int8  uint8;     /* 8 bits unsigned */
typedef signed int8    sint8;     /* 8 bits signed   */
typedef unsigned int16 uint16;    /* 16 bits unsigned*/
typedef signed int16   sint16;    /* 16 bits signed  */
typedef unsigned int32 uint32;    /* 32 bits unsigned*/
typedef signed int32   sint32;    /* 32 bits signed  */

#endif
```

```
/*---------------------------------------------------------------------------
 * regs.h
 * Purpose: constantes defines of registers for the I2C motor control modules
 *--------------------------------------------------------------------------*/

#ifndef __REGS_H
#define __REGS_H

#define REG_MODE                0x28
#define REG_SETPOINT_SOURCE     0x29
#define REG_HW_OPTIONS          0x2A
#define REG_SW_OPTIONS          0x2B
#define REG_CONTROL_TYPE        0x2C
#define REG_ERROR_FLAGS         0x2D
#define REG_STATUS_FLAGS        0x2E
#define REG_SETPOINT            0x2F
#define REG_POSITION            0x30
#define REG_SPEED               0x31
#define REG_TORQUE              0x32
#define REG_DIVIDER_POS         0x33
#define REG_POSITION_HH         0x34
#define REG_POSITION_HL         0x35
#define REG_POSITION_LH         0x36
#define REG_POSITION_LL         0x37
#define REG_DIVIDER_SPEED       0x38
#define REG_SPEED_H             0x39
#define REG_SPEED_L             0x3A
#define REG_DIVIDER_TORQUE      0x3B
#define REG_TORQUE_H            0x3C
#define REG_TORQUE_L            0x3D
#define REG_TORQUE_BIAS_H       0x3E
#define REG_TORQUE_BIAS_L       0x3F
#define REG_KP_POS              0x40
#define REG_KP_DIV_POS          0x41
#define REG_KD_POS              0x42
#define REG_KD_DIV_POS          0x43
#define REG_FILTER_POS          0x44
#define REG_KP_SPEED            0x45
#define REG_KP_DIV_SPEED        0x46
#define REG_KI_SPEED            0x47
#define REG_KI_DIV_SPEED        0x48
#define REG_KP_TORQUE           0x49
#define REG_KP_DIV_TORQUE       0x4A
#define REG_KI_TORQUE           0x4B
#define REG_KI_DIV_TORQUE       0x4C
#define REG_SAMPLE_TIME_H       0x4D
#define REG_SAMPLE_TIME_L       0x4E
#define REG_BLOCKED_TIME        0x4F
#define REG_INT_GEN_PERIOD      0x50
#define REG_INT_GEN_AMPLITUDE   0x51
#define REG_INT_GEN_OFFSET      0x52
#define REG_SOFT_STOP_MIN       0x53
#define REG_SOFT_STOP_MAX       0x54
#define REG_ACCELERATION        0x55
#define REG_MAX_SPEED           0x56
#define REG_STATIC_FRICTION     0x57
#define REG_HW_CURRENT_LIMIT    0x58
#define REG_SW_CURRENT_LIMIT    0x59
#define REG_DELTA_8BIT          0x5A
#define REG_PID_P_8BIT          0x5B
#define REG_PID_I_8BIT          0x5C
#define REG_PID_D_8BIT          0x5D
#define REG_PID_OUT_8BIT        0x5E
#define REG_EXT_DEVICE          0x61
#define REG_IR_INPUT            0x62
#define REG_RESET_VALUE         0x63

#define MODE_IDLE               0
```

```
#define MODE_NORMAL            1
#define MODE_STOP              2
#define MODE_SLEEP             3
#define MODE_RESET             4
#define MODE_SETPOINT_RESET    6

#define CONTROL_OPEN_LOOP      0
#define CONTROL_POS            1
#define CONTROL_POS_PROFILE    2
#define CONTROL_SPEED          3
#define CONTROL_SPEED_PROFILE  4
#define CONTROL_TORQUE         5
#define CONTROL_ZERO_FRICTION  6

#define SETPOINT_SRC_I2C       0
#define SETPOINT_SRC_ANALOG    1
#define SETPOINT_SRC_SQUARE    2
#define SETPOINT_SRC_TRIANGLE  3
#define SETPOINT_SRC_SINUS     4

#define HWO_STARTUP_NORMAL     0x01
#define HWO_ANALOG_SETPOINT    0x02
#define HWO_LED                0x04
#define HWO_ENC_025            0x08
#define HWO_INVERT_TORQUE      0x10


#endif
```

```
/*--------------------------------------------------------------------------------
 * sequences.h
 *
 * Author: Daisy Lachat <daisy.lachat@epfl.ch>
 * Date: 25-JUN-05
 * Release: 1.0
 * Purpose: constantes' tables for trajectories follow
 * Sequences:
 * 1) forward fast. Caud: 10° 8hz. Pect: 40° 6hz
 * 2) turn left. Caud: max offset. Pect: 40° 6hz
 * 3) forward slower. Caud: 30° 3hz. Pect: 30° 5hz
 * 4) turn left on itself. Caud:offset null. Pect: 50° 5hz, left: offset 180°
 * 5) spinning and forward motion. Caud: 10° 8hz. Pect: 40° 6hz, left: offset
90°
 *    right: offset -90°
 * 6) forward fast. Caud: 10° 8hz. Pect: 40° 6hz
 * 7) backward. Caud: straight. Pect: 40° 6hz
 * 8) diving. Caud: 10° 8hz. Pect: 40° 6hz with up offsets
 * 9) turn left with up motion. Caud: max offset. Pect: 40° 6hz with down offset
 *-------------------------------------------------------------------------------*/

#define SEQUENCES  9 /*number of sequences*/

/*ending time of each sequence*/
const uint8 tab_time[SEQUENCES+1] = {4, 7, 12, 18, 22, 26, 28, 32, 37, 100};

/*table of amplitudes*/
const uint8 tab_ampl[SEQUENCES][3] = {
  {21,   22,   22},
  { 0,   22,   22},
  {64,   17,   17},
  { 0,   28,   28},
  {21,   22,   22},
  {21,   22,   22},
  { 0,   22,   22},
  {21,   22,   22},
  { 0,   22,   22}
};

const uint8 tab_freq[SEQUENCES][3] = {

  {160,   120,   120},
  {  0,   120,   120},
  { 60,   100,   100},
  {  0,   100,   100},
  {160,   120,   120},
  {160,   120,   120},
  {  0,   120,   120},
  {160,   120,   120},
  {  0,   120,   120}
};

const uint8 tab_phas[SEQUENCES][3] = {  //in rad/M_PI
  {0,       1,    0},
  {0,       1,    0},
  {0,       1,    0},
  {0,       0,    0},
  {0,       1,    0},
  {0,       1,    0},
  {0,       1,    0},
  {0,       1,    0},
  {0,       1,    0}
};

const uint8 tab_offset[SEQUENCES][3] = {
  {0,     0,     0},
  {127,   0,     0},
  {0,     0,     0},
  {0,   100,     0},
  {0,    50,     50},
  {0,     0,     0},
  {0,   100,    100},
  {0,    20,    -20},
  {127,  -20,    20}
};
```

```
/*-----------------------------------------------------------------------------
 * sin.h
 *
 * Author: Daisy Lachat <daisy.lachat@epfl.ch>
 * Date: 25-JUN-05
 * Release: 1.0
 * Purpose: lookup table for computing the sinus, values are multiplied by 100
 *----------------------------------------------------------------------------*/
#ifndef __SIN_H
#define __SIN_H

const sint8 lookup_sin[256] =
{0,2,5,7,10,12,15,17,20,22,24,27,29,31,34,36,38,41,
43,45,47,49,52,54,56,58,60,62,64,66,67,69,71,73,74,76,78,79,81,82,83,85,86,87,
88,90,91,92,93,93,94,95,96,97,97,98,98,99,99,99,100,100,100,100,100,100,100,100,
99,99,99,98,98,97,97,96,95,95,94,93,92,91,90,89,88,87,85,84,83,81,80,78,77,75,
73,72,70,68,66,65,63,61,59,57,55,53,51,48,46,44,42,40,37,35,33,30,28,26,23,21,
18,16,14,11,9,6,4,1,-1,-4,-6,-9,-11,-14,-16,-18,-21,-23,-26,-28,-30,-33,-35,-37,
-40,-42,-44,-46,-48,-51,-53,-55,-57,-59,-61,-63,-65,-66,-68,-70,-72,-73,-75,-77,
-78,-80,-81,-83,-84,-85,-87,-88,-89,-90,-91,-92,-93,-94,-95,-95,-96,-97,-97,-98,
-98,-99,-99,-99,-100,-100,-100,-100,-100,-100,-100,-100,-99,-99,-99,-98,-98,-97,
-97,-96,-95,-94,-93,-93,-92,-91,-90,-88,-87,-86,-85,-83,-82,-81,-79,-78,-76,-74,
-73,-71,-69,-67,-66,-64,-62,-60,-58,-56,-54,-52,-49,-47,-45,-43,-41,-38,-36,-34,
-31,-29,-27,-24,-22,-20,-17,-15,-12,-10,-7,-5,-2,0};

#endif
```

B.2 Matlab

```
%--------------------------------------------------------------------------
% generate_sine_table.m
%
% Author: Daisy Lachat <daisy.lachat@epfl.ch>
% Date: 25-JUN-05
% Release: 1.0
% Purpose: create a lookup table for sine wave
%--------------------------------------------------------------------------

N=255;
for i=1:N
   tab(i)=round(100*sin(i*2*pi/N));
end

plot(tab)
fid = fopen('sin.txt','w');
fprintf(fid,'{0 ');
fprintf(fid,';%g',tab);
fprintf(fid,'} ');
fclose(fid);
```

```
%----------------------------------------------------------------------------
% plot_position.m
%
% Author: Daisy Lachat <daisy.lachat@epfl.ch>
% Date: 25-JUN-05
% Release: 1.0
% Purpose: plot the 3 graphics of desired position and measured position
%          and a more visual graphic that show the position of the 3 fins
%          the data file is generated by the software Rs232Interface.cpp
%----------------------------------------------------------------------------

clear
close all
load data.txt  %load data file

%plot graphics of desired position(in green) and measured position (in red)
%caudal fin
figure();
plot(data(:,1),data(:,2),'g',data(:,1),data(:,3),'r');
title('caudal fin');
legend('desired position', 'measured position');
xlabel('time [s]');
ylabel('position');

%left pectoral fin
figure();
plot(data(:,4),data(:,5),'g',data(:,4),data(:,6),'r');
title('pectoral fin left');
legend('desired position', 'measured position');
xlabel('time [s]');
ylabel('position');

%right pectoral fin
figure();
plot(data(:,7),data(:,8),'g',data(:,7),data(:,9),'r');
title('pectoral fin right');
legend('desired position', 'measured position');
xlabel('time [s]');
ylabel('position');

%plot the measured position of each fin in the same graphic
figure();
plot(data(:,7),data(:,8),'b',data(:,1),data(:,2),'r', data(:,4),data(:,5),'g');

legend('right pectoral fin','caudal fin', 'left pectoral fin');
```

```
%-------------------------------------------------------------------------------
% plot_surface.m
%
% Author: Daisy Lachat <daisy.lachat@epfl.ch>
% Date: 25-JUN-05
% Release: 1.0
% Purpose: create a graphical approximated surface with data in x,y,z vectors
%          It is used to represente the dependance of z=speed with x=amplitude
%          and y=frequency of the fins' oscillation
%-------------------------------------------------------------------------------

clear;

%load data from appropriate file
%pect_data;
caudal_data;

%plot the surface
tri = delaunay(x,y);
plot(x,y,'.')
[r,c] = size(tri);
disp(r);  %number of triangles
h = trisurf(tri, x, y, z); hold on;  %plot
axis vis3d;
axis([10 60 1 9 0 0.25]);
%clean it

lighting phong;
shading interp;
colorbar EastOutside;

%title('Speed variation with frequency and amplitude for pecoral fins',
'FontSize', 17);
title('Speed variation with frequency and amplitude for caudal fin', 'FontSize',
17);
ylabel('frequency [Hz]', 'FontSize', 14);
xlabel('amplitude [°]', 'FontSize', 14);
zlabel('speed [m/s]', 'FontSize', 14);

scatter3(x,y,z,'.m');   %plot the measure points on the same graphic

%plot the physical limit: multiple lines in different z position
k=0;
for i=1:50
  zlim=k*ones(9,1);
  plot3(xlim,ylim,zlim, 'k');
  hold on;
  k = k+0.006;
end

hold off;
```

B.3 RS232 interface

```
/*-----------------------------------------------------------------------------
 * Rs232Interface.cpp
 *
 * Author: Alessandro Crespi
 * Modified: Daisy Lachat <daisy.lachat@epfl.ch>
 * Date: 25-JUN-05
 * Release: 1.1
 * Purpose: RS232 interface to catch values send by PIC
 *---------------------------------------------------------------------------*/

#include <stdio.h>
#include <math.h>
#include <windows.h>

#define OCOUNT 3
#define COUNT OCOUNT + 1

void wperror(char *str)
{
DWORD err = GetLastError();
char *msg = NULL;

FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM | 60,
  NULL, err, LANG_NEUTRAL, (char *) &msg, 2048, NULL);
fprintf(stderr, "%s: %s\n", str, msg);
LocalFree(msg);
}

HANDLE open_serial(char *portname, int speed)
{
HANDLE h;
DCB dcb;
COMMTIMEOUTS ct;

h = CreateFile(portname, GENERIC_READ | GENERIC_WRITE, 0, NULL, OPEN_EXISTING,
  0, 0);
if (h==INVALID_HANDLE_VALUE) {
  wperror("CreateFile");
  return h;
}

GetCommState(h, &dcb);
dcb.DCBlength = sizeof(dcb);
dcb.BaudRate = speed;
dcb.ByteSize = 8;
dcb.Parity = NOPARITY;
dcb.StopBits = ONESTOPBIT;
if (!SetCommState(h, &dcb)) {
  wperror("SetCommState");
  CloseHandle(h);
  return INVALID_HANDLE_VALUE;
}

SetupComm(h, 4096, 16);

ZeroMemory(&ct, sizeof(ct));
if (!SetCommTimeouts(h, &ct)) {
  wperror("SetCommTimeouts");
  CloseHandle(h);
  return INVALID_HANDLE_VALUE;
}

return h;
}

bool read_block(HANDLE hSerial, void *buffer, DWORD count)
{
DWORD w;
```

```
if (!ReadFile(hSerial, buffer, count, &w, NULL)) {
  wperror("ReadFile");
  return false;
}

return (count==w);
}

DWORD swap_dword(DWORD d)
{
return (d >> 24) | ((d & 0xFF00) << 8) | ((d & 0xFF0000) >> 8) | ((d & 0xFF) <<
24);
}

double GetTickCountW()
{
__int64 l, f;

QueryPerformanceCounter((LARGE_INTEGER *) &l);
QueryPerformanceFrequency((LARGE_INTEGER *) &f);

return (double) l / (double) f;
}

float dword_to_float(DWORD d)
{
DWORD e = (d & 0x00800000) << 8 | (d & 0x007FFFFF) | (d & 0xFF000000) >> 1;
float res = *((float *)(&e));
return res;
}

int main(int argc, char *argv[])
{
HANDLE h;
FILE *f;
BYTE b;
char num_int, num_bloc;
DWORD d;

h = open_serial("COM1", 38400);
if (h==INVALID_HANDLE_VALUE) return 1;

printf("Wait... ");
double r,s;

do {
  do {
    read_block(h, &b, 1);        //beginning of init
  } while (b!=2);
        printf(".");
        r = GetTickCountW();
    read_block(h, &num_bloc, 1); //read the number of bloc to printf
        read_block(h, &num_int, 1);  //read the number of int in each bloc
        read_block(h, &b, 1);        //end of init
        s = GetTickCountW();
        fflush(stdout);
} while (b!=2);
printf(" Init OK\n");
printf("Data frame rate: %0.2f fps\n", 1/(s-r));

f = fopen("/data/data.txt", "w");
if (f==NULL) {
        wperror("fopen");
        return 2;
}

//beginning of reading
```

```
do {

    for (int k=1; k<(num_bloc+1); k++) {
          read_block(h, &d, 4);
      fprintf(f, "%f ", dword_to_float(swap_dword(d))); //write time (float) and
a space

          for (int j=1; j<(num_int+1); j++) {
             char c;
                 read_block(h, &c, 1);
                 fprintf(f, "%d ", c);                         //write position
(int) and a space
          }

      read_block(h, &b, 1);  //sync
          if (b!=2) {
        fprintf(f, "\nsync lost ");
                fprintf(stderr, "ERROR: synchronization lost!\n");
                CloseHandle(h);
                return 2;
          }
    }
    fprintf(f, "\n"); //carriage return

} while (1);
fflush(f);
fprintf(f, "\nend of prog ");
CloseHandle(h);

return 0;
}
```

Appendix C

Web sites of fish robots

These web sites were visited in march 2005.

- RoboTuna I, http://web.mit.edu/towtank/www/Tuna/Tuna1/tuna1.html
- RoboTuna II, http://web.mit.edu/towtank/www/Tuna/tuna.html
- RoboPike, http://web.mit.edu/towtank/www/Pike/pike.html
- SPC-II,http://www.vieartificielle.com/nouvelle/?id_nouvelle=769
- Dongle,http://www.seattlerobotics.org/encoder /200211/autonomous_robotic_fish.html
- B1, http://www-personal.umich.edu/ bobden/biomechatronic _devices.html#Movie%20of%20our%20Biomechatronic%20Fish
- NMRI's robots, http://www.nmri.go.jp/eng/khirata/fish/index_e.html
- Essex University's robots, http://privatewww.essex.ac.uk / jliua
- Coelacanth fish, http://www.mhi.co.jp/enews/e_0898.html
- Aqua project, http://www.aquarobot.net:8080/AQUA/AQUA/index_html

Appendix D

Translation english-french of fishes' vocabulary

English		French
Trout	=	truite
Tuna	=	thon
Salmon	=	saumon
Swordfish	=	espadon
Mackerel	=	maquereau
Herring	=	hareng
Pike	=	brochet
Carp	=	carpe
Cord	=	morue
Boxfish	=	poisson coffre
Angelfish	=	poisson ange
eel	=	anguille
lamprey	=	lamproie
buoyancy	=	flottabilité
air bladder	=	vessie natatoire
fin	=	nageoire
caudal fin	=	nageoire caudale
pectoral fins	=	nageoires pectorales
pelvic fins	=	nageoires pelviennes ou ventrales
median fins	=	nageoires impaires
spiny rays	=	rayons pineux
soft rays	=	rayons mous

Appendix E

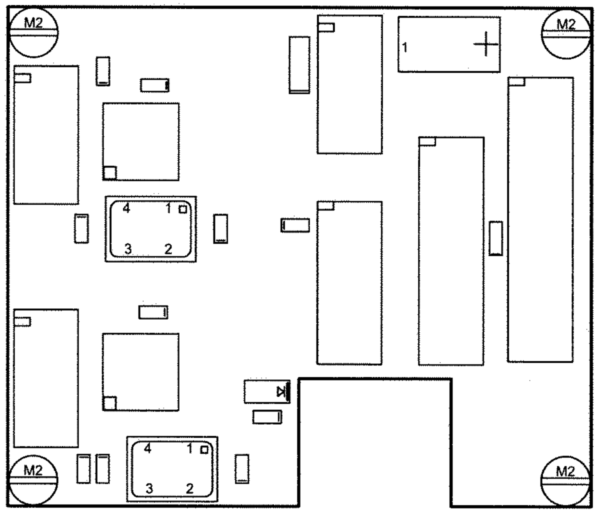
Electronical parts

E.1 Control board Amphibot II - body module

MICROCON2

LSL A.Crespi
Phone: 36630
Fri Apr 8 08:56:06 MET DST 2005

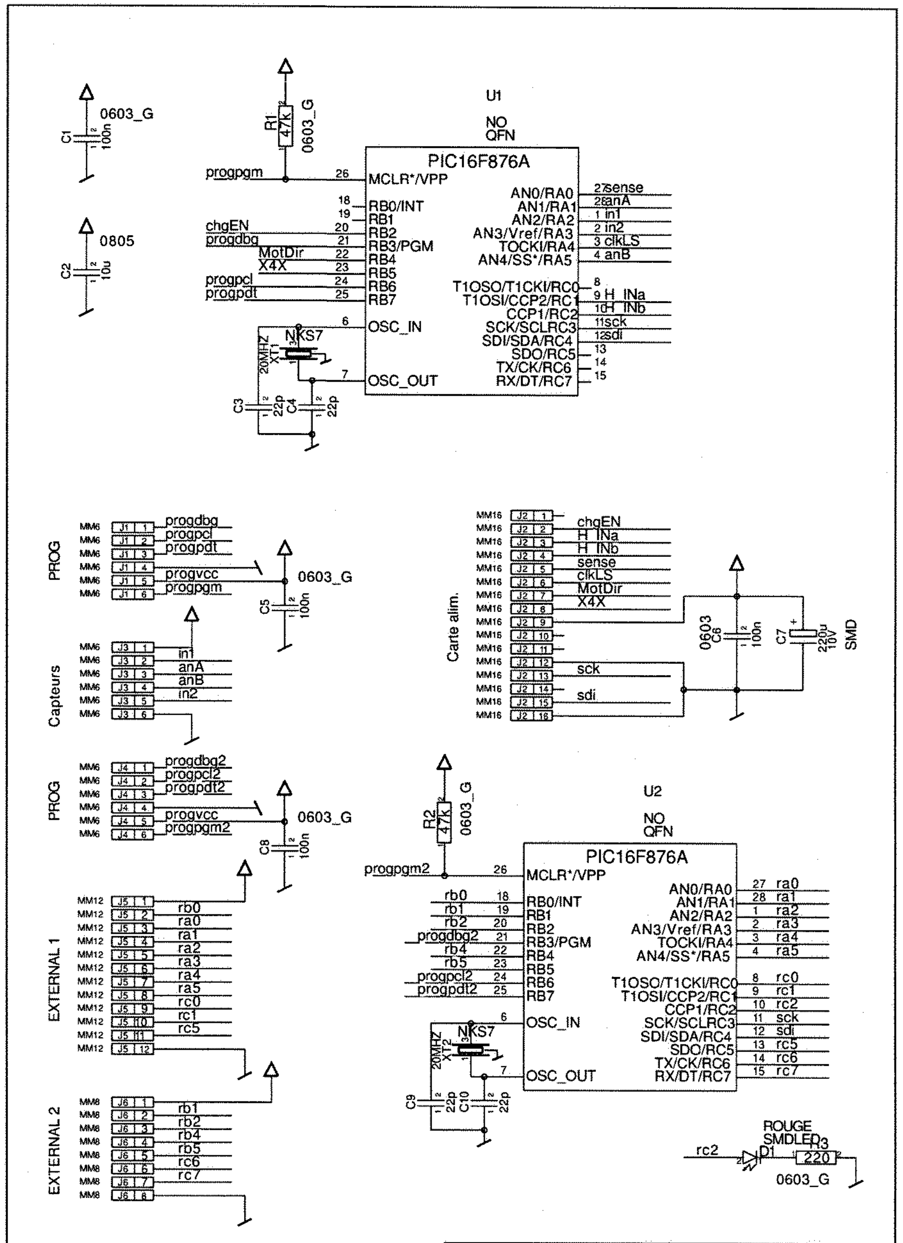
MFG_DATE:	Undefined
QUANTITY:	12
SOLDERMASK:	mtop mbot
MATERIAL:	FR4 : 0.8 mm



CONTENTS	PAGE
Root Schema salamuc1	2
Xref salamuc1	3
Drill	4
Etch	5
Outline_Top	6
Outline_Top_Scale4_0	7
Gerber_mtop	8
Gerber_top	9
Gerber_bot	10
Gerber_mbot	11
Gerber_cont	12
Component Report	13
BOM Report	14
NC-Pins Report	15

CONTENTS	PAGE
Power-Pins Report	16
Single Node Nets	17

ACORT



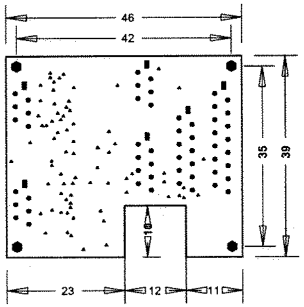
DRAWING	ENGINEER:
microcon2	A. Crespi
DATE:	PAGE: 1/1
Thu Apr 7 14:28:48 2005	acort cuser

Signal Page Ref MICROCON2

salamuc1 Fri Apr 8 08:56:07 MET DST 2005

Ref Des	Pages
anA	
anB	
chgEN	
clkLS	
H_INa	
H_INb	
in1	
in2	
MotDir	
progdbg	
progdbg2	
progpcl	
progpcl2	
progpdt	
progpdt2	
progpgm	
progpgm2	
progvcc	
ra0	
ra1	
ra2	
ra3	
ra4	
ra5	
rb0	
rb1	
rb2	
rb4	
rb5	
rc0	
rc1	
rc2	
rc5	
rc6	
rc7	
sck	
sdi	
sense	
X4X	

Total Signals count 39

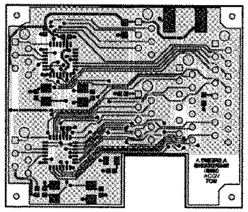


DRILL CHART			
FIGURE	SIZE	PLATED	QTY
·	0.203	PLATED	74
●	0.787	PLATED	54
■	1.499	PLATED	6
⬢	2.2	NOT PLATED	4

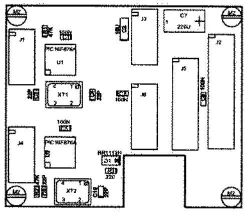
MICROCON2

LABORATOIRE:LSL	DATE:07.04.05
ENGINEER:A.Crespi	EPFL-ACORT-gv

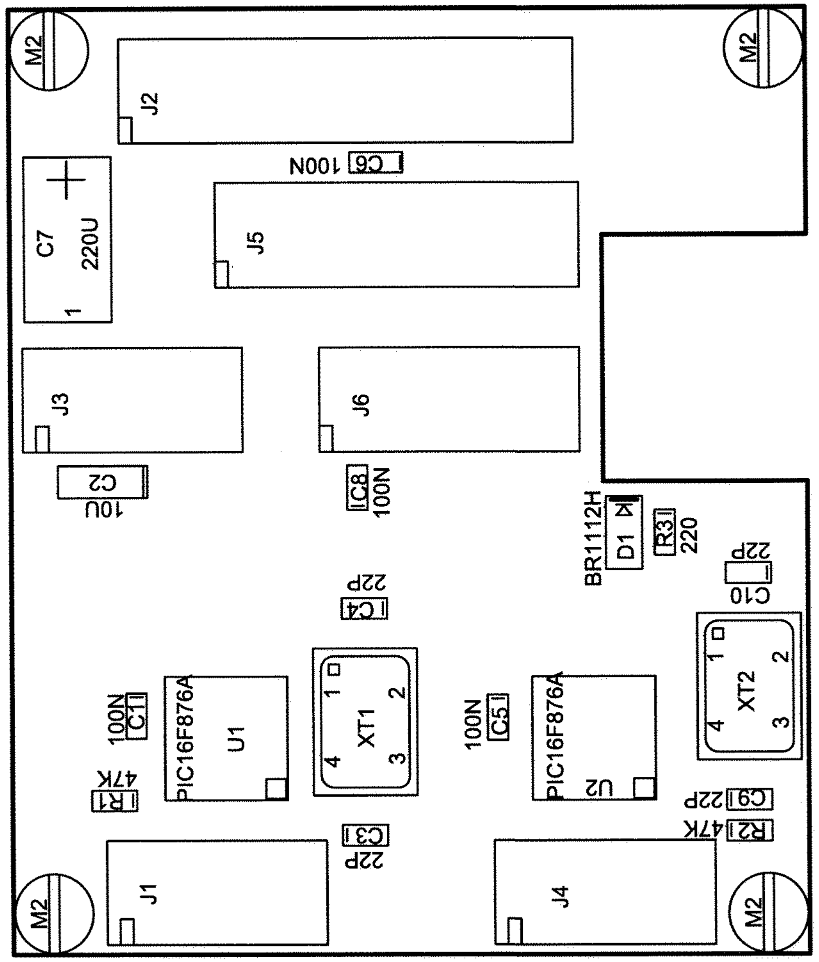
EPFL ACORT Fri 08.04.2005 8:55 Drawing name = MICROCON2 Scale = 1.00 1.000 Inch =



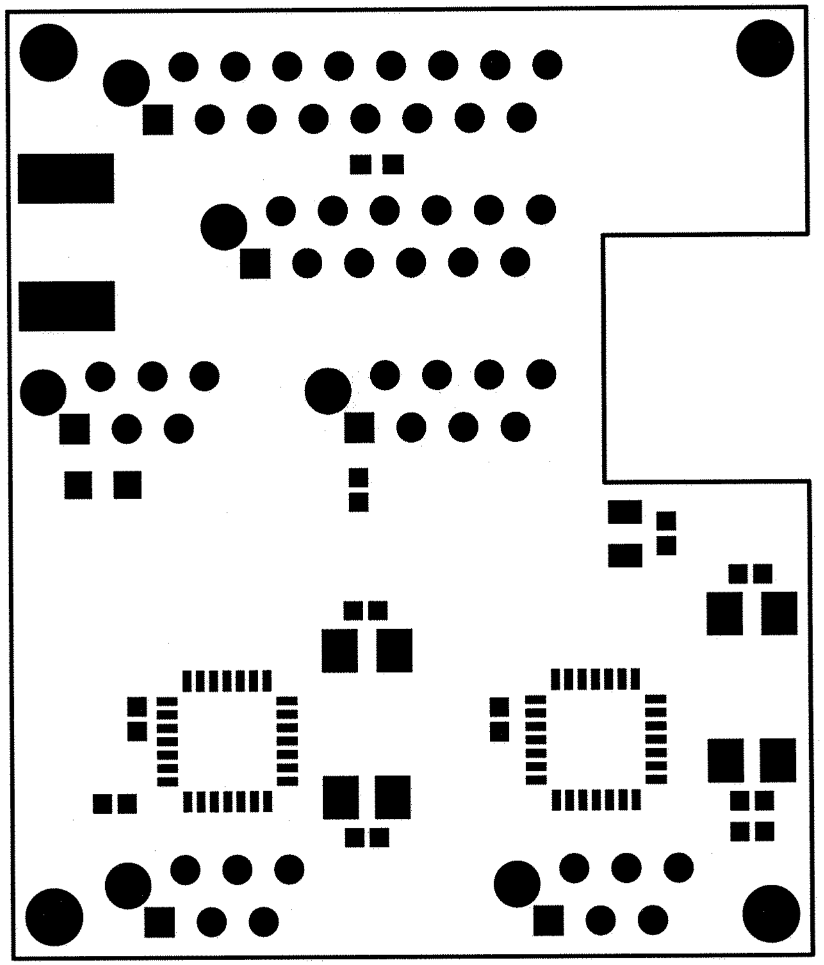
MICROCON2	LABORATOIRE:LSL	DATE:07.04.05
	ENGINEER:A.Crespi	EPFL-ACORT-gv



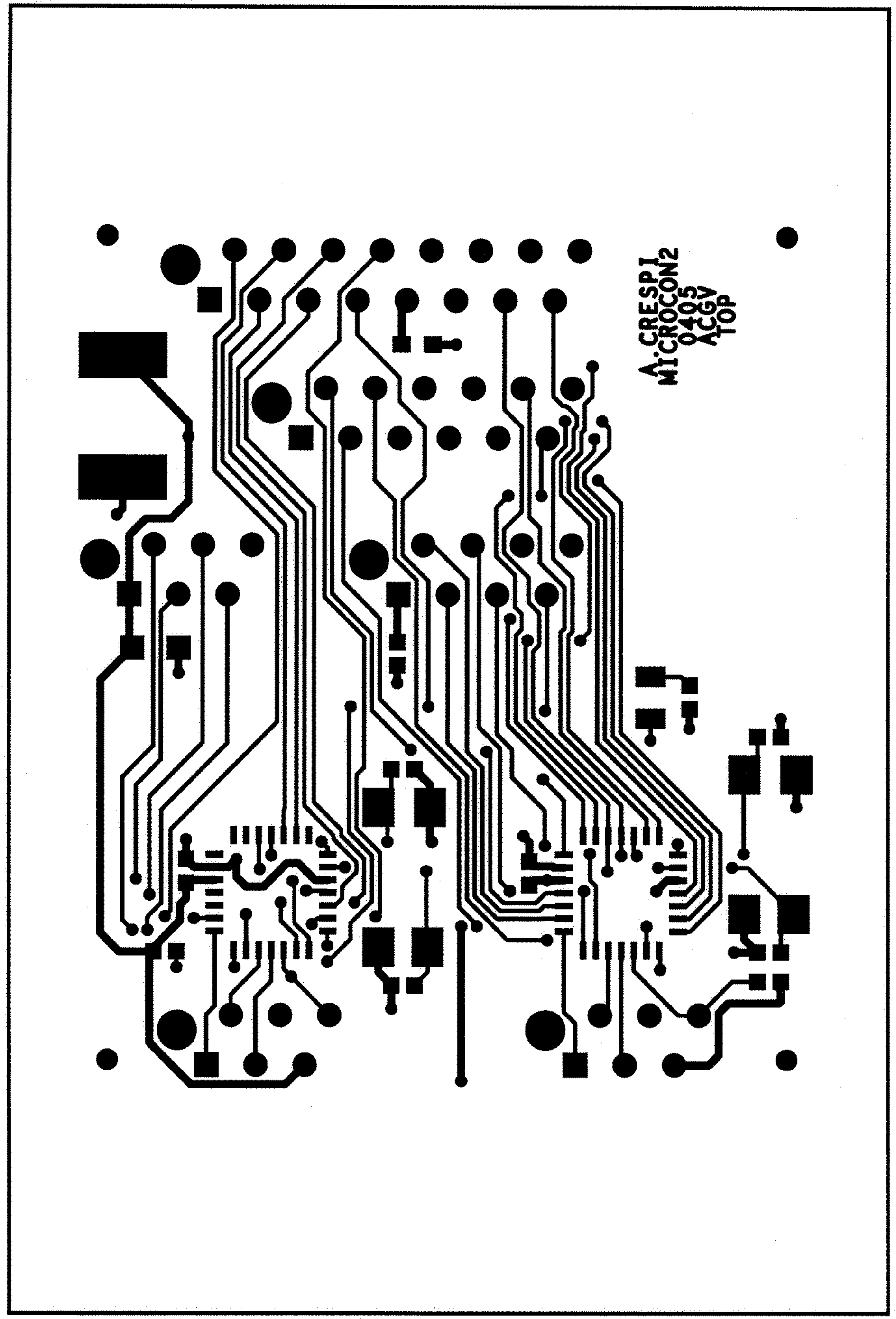
MICROCON2	LABORATOIRE:LSL	DATE:07.04.05
	ENGINEER:A.Crespi	EPFL-ACORT-gv



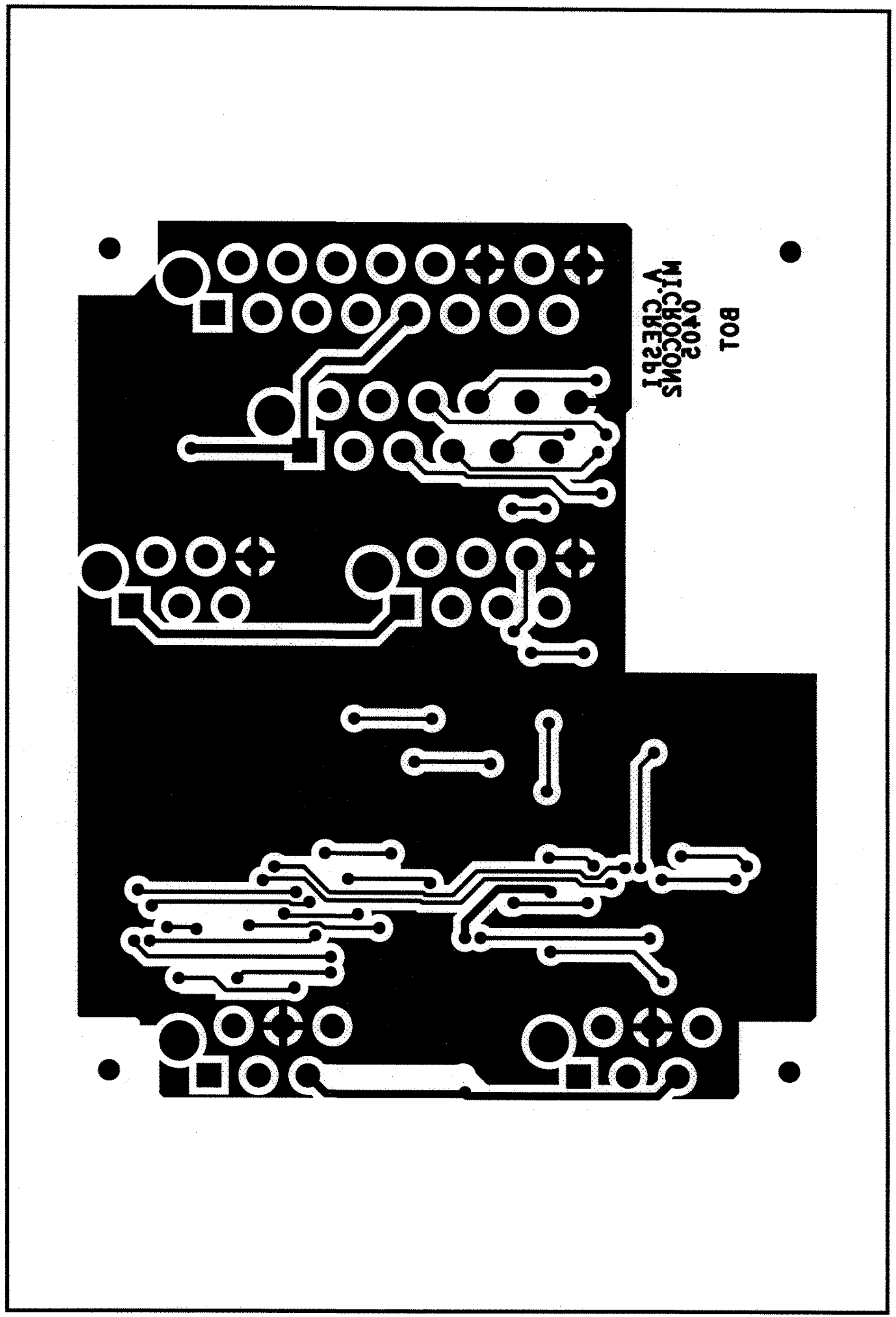
EPFL ACORT Fri 08.04.2005 8:55 Drawing name = MICROCON2 Scale = 4.00 0.250 Inch =



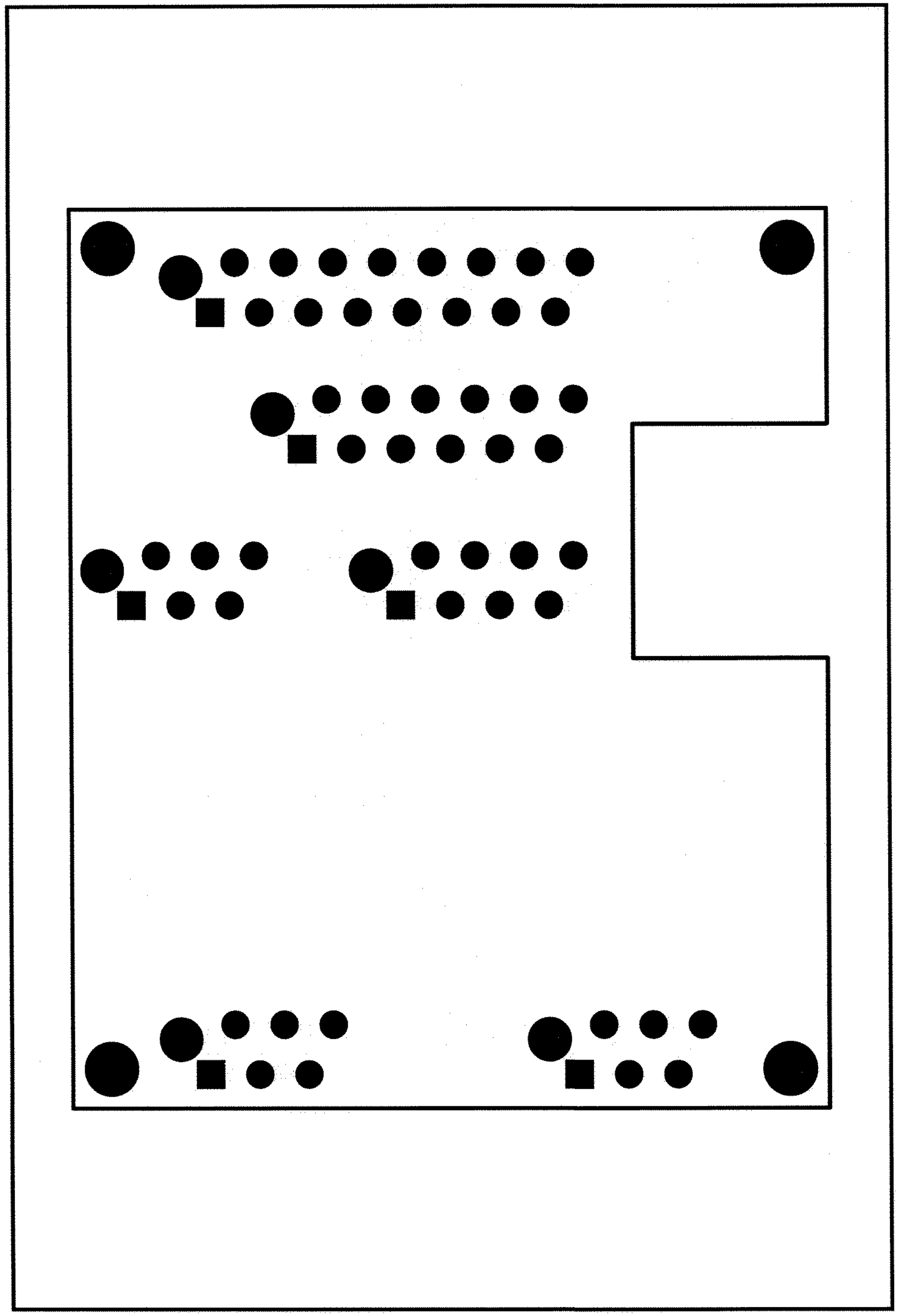
EPFL ACORT Fri Apr 8 08:55:55 2005 Project = microcon2 File = microcon2_gloss,mtop Scale = 4 0.25 cm =



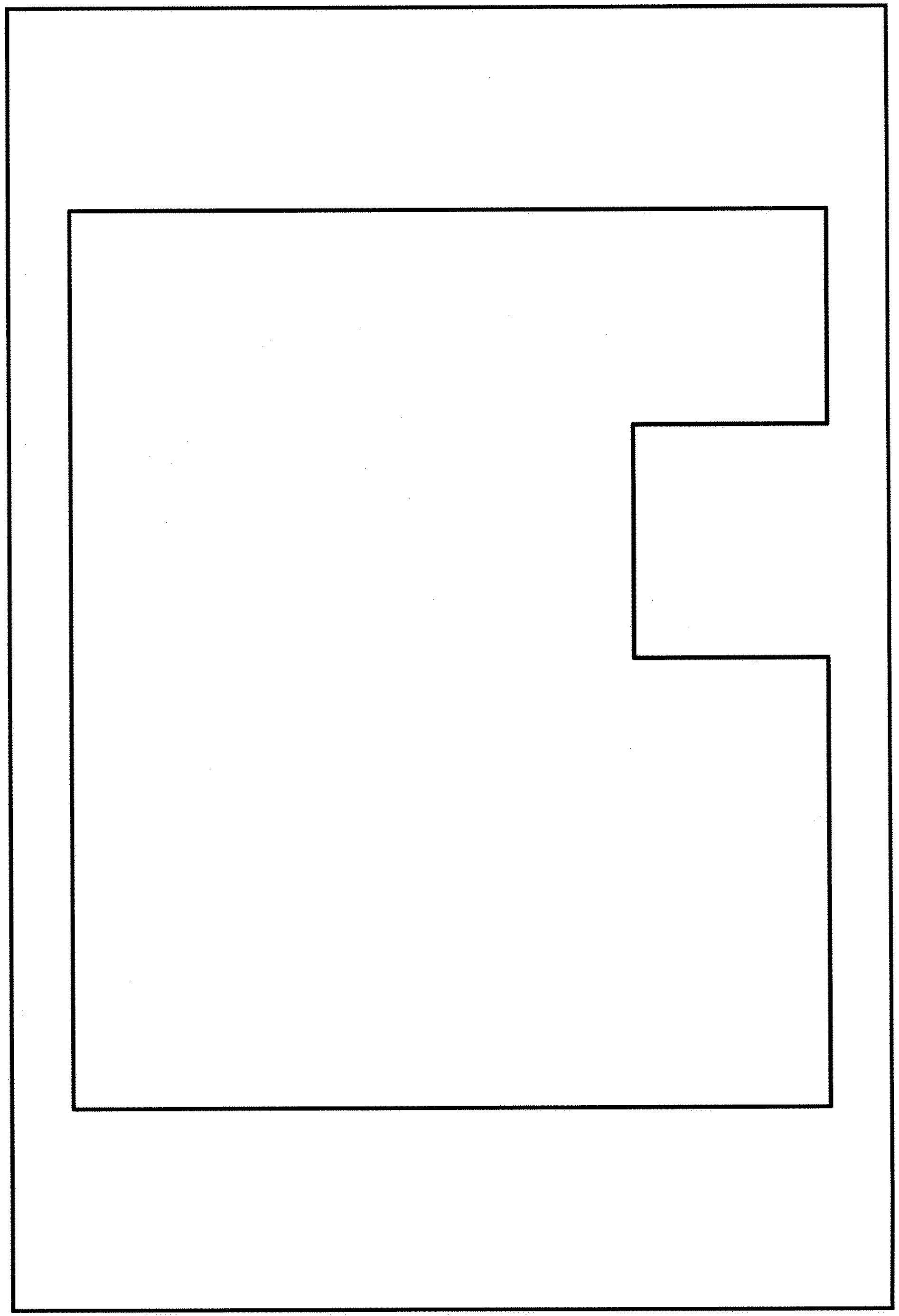
EPFL ACORT Fri Apr 8 08:55:55 2005 Project = microcon2 File = microcon2_gloss,top Scale = 4 0.25 cm = ______



EPFL ACORT Fri Apr 8 08:55:55 2005 Project = microcon2 File = microcon2_gloss,bot Scale = 4 0.25 cm = ______



EPFL ACORT Fri Apr 8 08:55:55 2005 Project = microcon2 File = microcon2_gloss,mbot Scale = 4 0.25 cm = ______



EPFL ACORT Fri Apr 8 08:55:55 2005 Project = microcon2 File = microcon2_gloss,cont Scale = 4 0.25 cm = ______

Component Report MICROCON2

microcon2_gloss Fri Apr 8 08:56:07 MET DST 2005

Ref Des	Device Type	Value	Package Type	x	y	ang	Mir	Remark
C1	0603	100N	0603_G	-11.400	18.400	270.000	NO	
C2	0805	10U	0805	0.000	20.000	180.000	NO	
C3	0603	22P	0603_G	-17.200	7.200	0.000	NO	
C4	0603	22P	0603_G	-6.200	7.200	180.000	NO	
C5	0603	100N	0603_G	-11.500	0.700	270.000	NO	
C6	0603	100N	0603	15.500	6.500	180.000	NO	
C7	CP	220U	CP_2917	11.800	21.700	90.000	NO	
C8	0603	100N	0603_G	-0.300	7.500	90.000	NO	
C9	0603	22P	0603_G	-15.500	-11.600	0.000	NO	
C10	0603	22P	0603_G	-4.500	-11.600	180.000	NO	
D1	BR1112H	BR1112H	SMDLED	-2.512	-5.500	270.000	NO	
J1	CO6		MM6	-21.270	17.358	270.000	NO	
J2	CO16		MM16	17.730	17.208	270.000	NO	
J3	CO6		MM6	2.730	21.358	270.000	NO	
J4	CO6		MM6	-21.270	-1.642	270.000	NO	
J5	CO12		MM12	10.730	12.485	270.000	NO	
J6	CO8		MM8	2.730	7.445	270.000	NO	
R1	R	47K	0603_G	-15.500	19.500	180.000	NO	
R2	R	47K	0603_G	-17.000	-11.600	0.000	NO	
R3	R	220	0603_G	-2.500	-7.500	270.000	NO	
U1	PIC16F876A		MLF28	-12.500	14.000	90.000	NO	
U2	PIC16F876A		MLF28	-12.500	-4.000	90.000	NO	
XT1	XTALGND	20MHZ	NKS7	-11.700	7.200	270.000	NO	
XT2	XTALGND	20MHZ	NKS7	-10.000	-11.600	270.000	NO	

Total Component count 24

BOM Report MICROCON2

microcon2_gloss Fri Apr 8 08:56:08 MET DST 2005

Device	Package	Value	Nb	Reference Designators	Remark
0603-1-100N	0603_G	100N	3	C1 C5 C8	
0603-1-22P	0603_G	22P	4	C3 C4 C9 C10	
0603-100N	0603	100N	1	C6	
0805-10U	0805	10U	1	C2	
BR1112H	SMDLED	BR1112H	1	D1	
CO12-15	MM12		1	J5	
CO16-10	MM16		1	J2	
CO6-5	MM6		3	J1 J3 J4	
CO8-22	MM8		1	J6	
CP-10V-SMD220UF-220U	CP_2917	220U	1	C7	
PIC16F876A	MLF28		2	U1 U2	
R-23-220	0603_G	220	1	R3	
R-23-47K	0603_G	47K	2	R1 R2	
XTALGND-0-20MHZ	NKS7	20MHZ	2	XT1 XT2	

Total Component count 24

NC Pins Report MICROCON2

microcon2_gloss Fri Apr 8 08:56:08 MET DST 2005

Ref Des	Device	Nb	Not Connected Pins	Remark
J2	CO16-10	4	1 10 11 14	
U1	PIC16F876A	6	8 13 14 15 18 19	

Total count 10

Power Pins Report MICROCON2

microcon2_gloss Fri Apr 8 08:56:09 MET DST 2005

Ref Des	Device	Name	Power Pins	Remark
U1	PIC16F876A	GND	5 16	
		VCC	17	
U2	PIC16F876A	GND	5 16	
		VCC	17	

Total count 2

Single Node Nets Report MICROCON2

microcon2_gloss Fri Apr 8 08:56:09 MET DST 2005

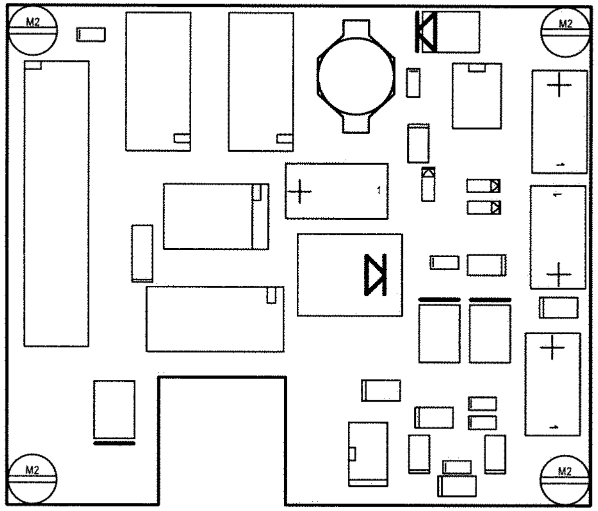
Netname	Node	Device	Remark
Total Nets count 0			

E.2 Power board Amphibot II - body module

SALAMPOWER

LSL A.Crespi
Phone: 36630
Mon Apr 11 08:21:00 MET DST 2005

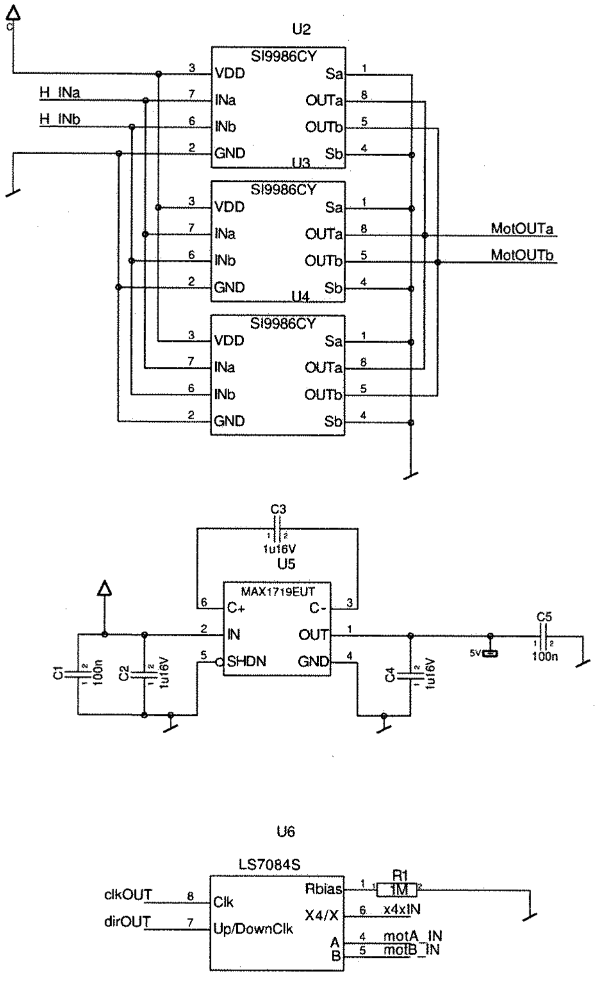
MFG_DATE:	Undefined
QUANTITY:	12
SOLDERMASK:	mtop mbot
MATERIAL:	FR4 : 0.8 mm



CONTENTS	PAGE
Root Schema salammotor	2 - 5
Xref salammotor	6
Drill	7
Etch	8
Outline_Bot_Scale2_0	9
Outline_Bot_Scale4_0	10
Outline_Top	11
Outline_Top_Scale2_0	12
Outline_Top_Scale4_0	13
Gerber_mtop	14
Gerber_top	15
Gerber_bot	16
Gerber_mbot	17
Gerber_cont	18

CONTENTS	PAGE
Component Report	19 - 20
BOM Report	21
NC-Pins Report	22
Power-Pins Report	23
Single Node Nets	24

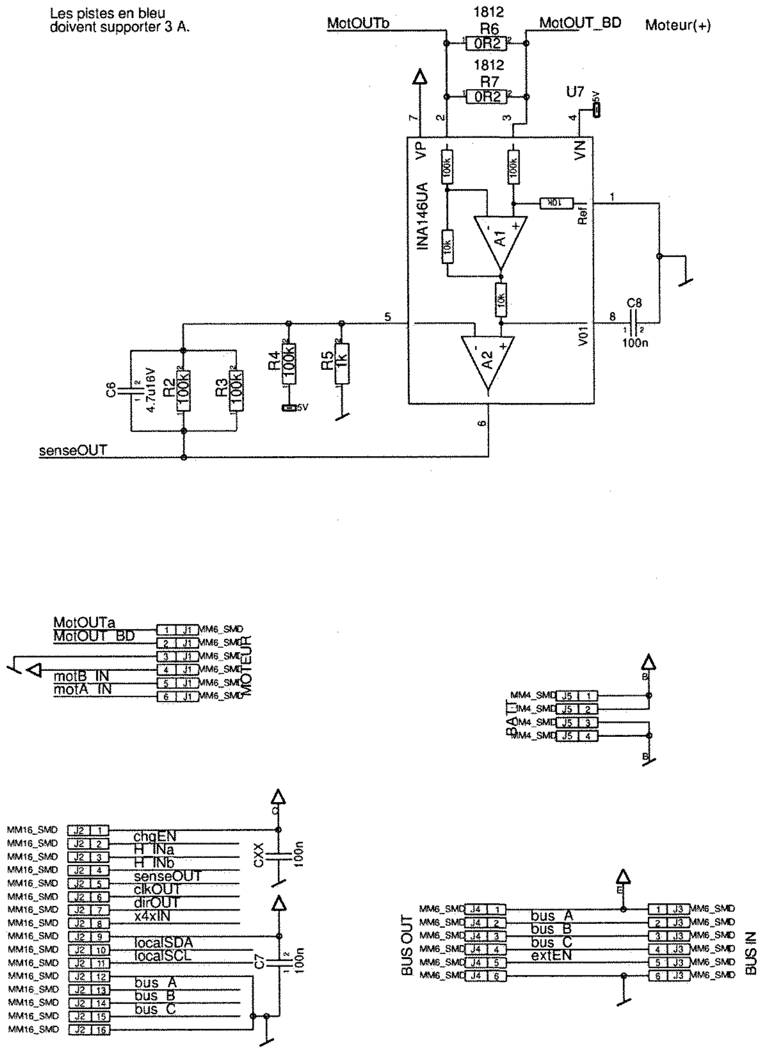
ACORT



DRAWING	ENGINEER:
salampower	A. Crespi
DATE: Fri Apr 8 10:59:42 2005	PAGE: 1/4 acort cuser

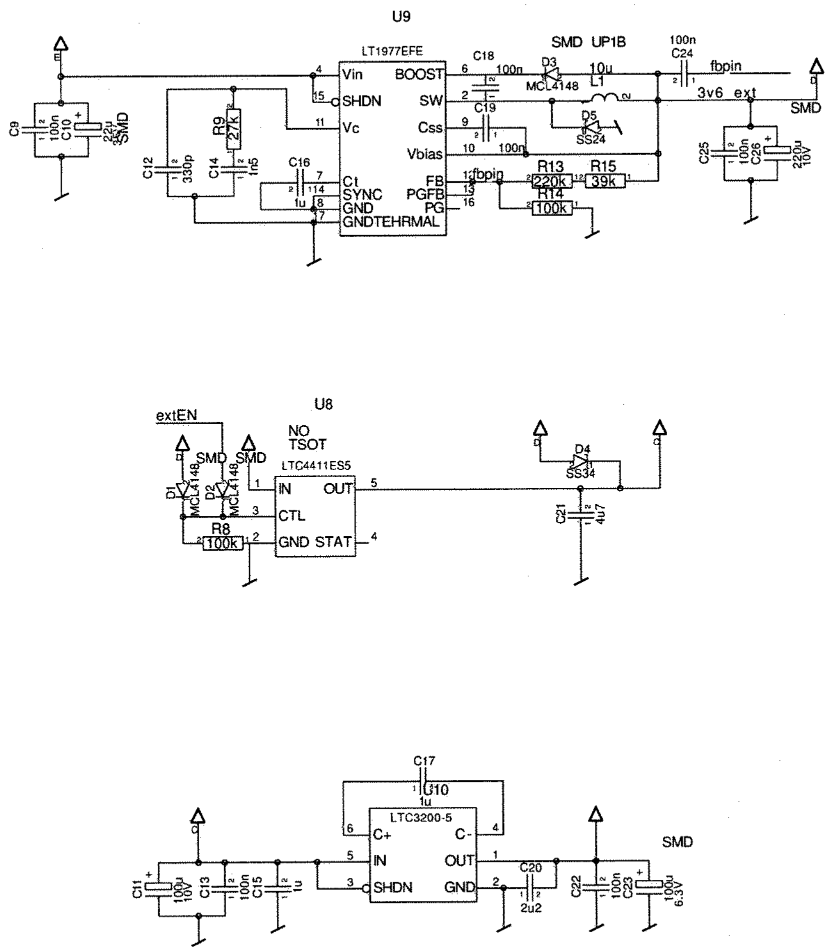
ATTENTION

Les pistes en bleu
doivent supporter 3 A.

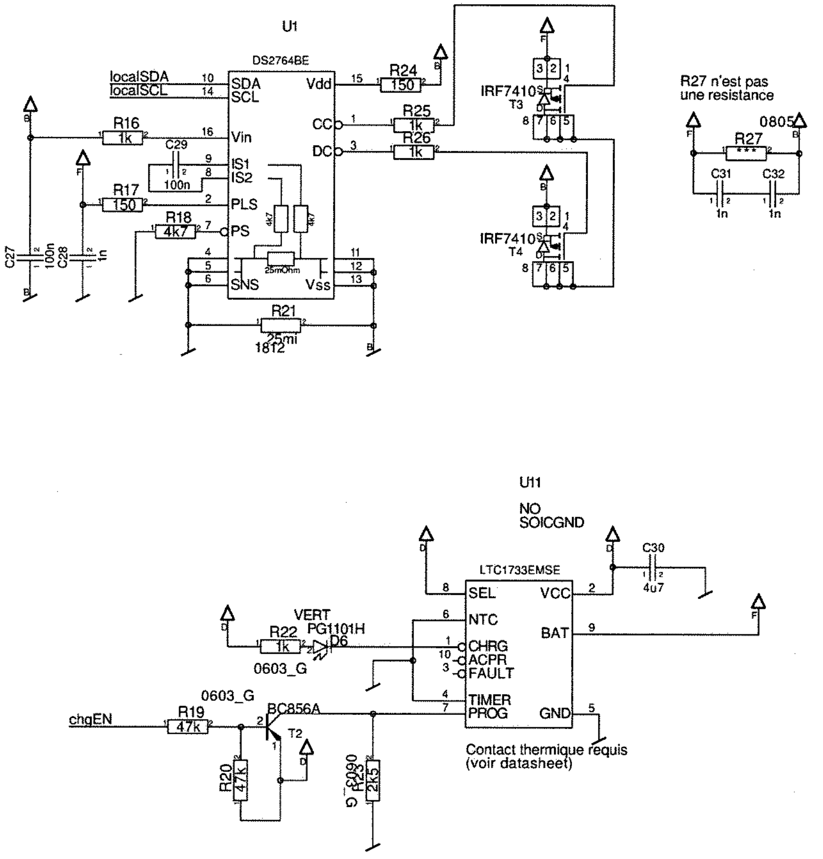


DRAWING salampower	ENGINEER: A. Crespi
DATE: Fri Apr 8 10:59:38 2005	PAGE: 2/4 acort cuser

ATTENTION!
(1) pistes en vert (min 1 A)
(2) pistes en bleu (min 3 A)
(3) LAYOUT SELON DATASHEET DU LT1977 (PISTES COURTES!)



DRAWING salampower	ENGINEER: A. Crespi
DATE: Fri Apr 8 10:59:34 2005	PAGE: 3/4 acort cuser



DRAWING salampower	ENGINEER: A. Crespi
DATE: Fri Apr 8 10:59:28 2005	PAGE: 4/4 acort cuser

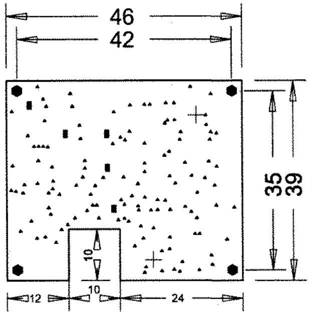
Signal Page Ref SALAMPOWER

salammotor

Mon Apr 11 08:21:06 MET DST 2005

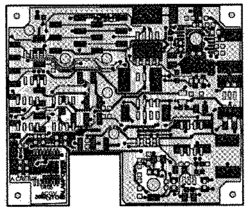
Ref Des	Pages
3v6_ext	3
bus_A	2 2
bus_B	2 2
bus_C	2 2
chgEN	2 4
clkOUT	1 2
dirOUT	1 2
extEN	2 3
fbpin	3 3
H_INa	1 2
H_INb	1 2
localSCL	2 4
localSDA	2 4
motA_IN	1 2
motB_IN	1 2
MotOUTa	1 2
MotOUTb	1 2
MotOUT_BD	2 2
senseOUT	2 2
x4xIN	1 2

Total Signals count 20

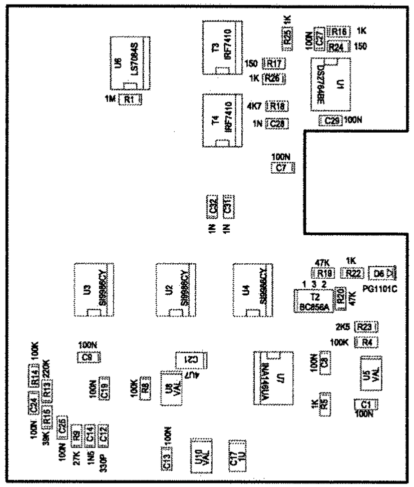


DRILL CHART			
FIGURE	SIZE	PLATED	QTY
·	0.203	PLATED	123
■	1.499	PLATED	5
+	2.184	PLATED	2
⬢	2.2	NOT PLATED	4

SALAMPOWER	LABORATOIRE:LSL	DATE:11.04.05
	ENGINEER:A.Crespi	EPFL-ACORT-gv

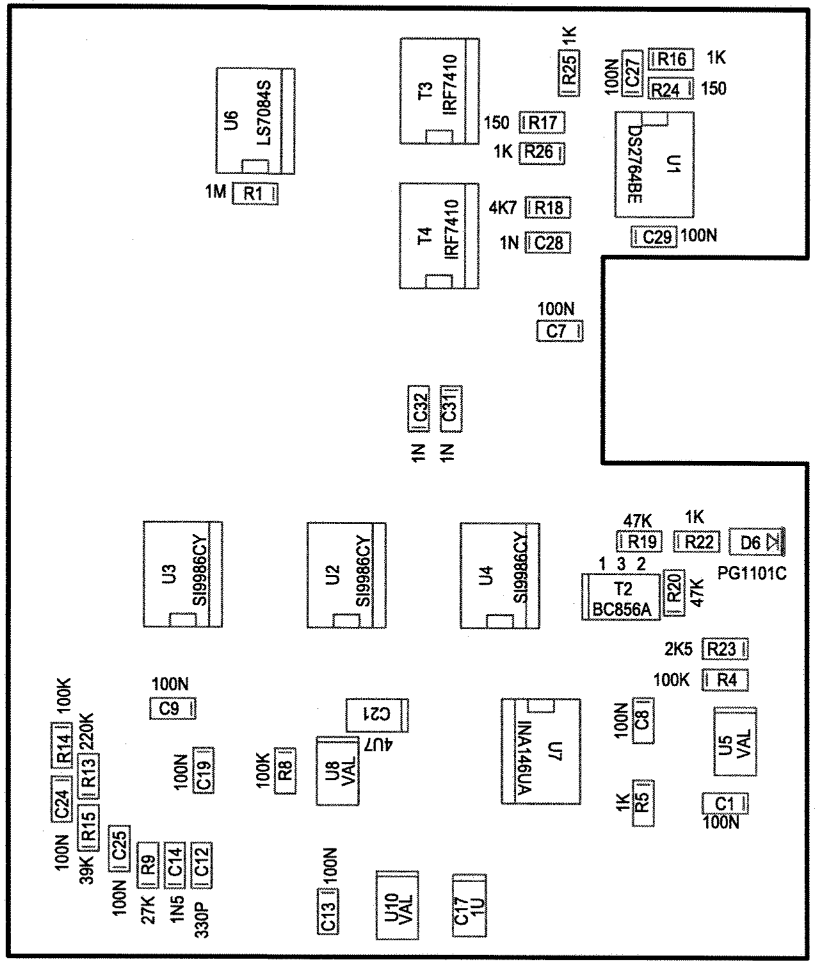


SALAMPOWER	LABORATOIRE:LSL	DATE:11.04.05
	ENGINEER:A.Crespi	EPFL-ACORT-gv



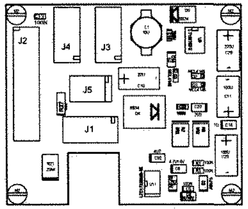
EPFL ACORT Mon 11.04.2005 8:20 Drawing name = SALAMPOWER Scale = 2.00 0.500 Inch =

Page 9

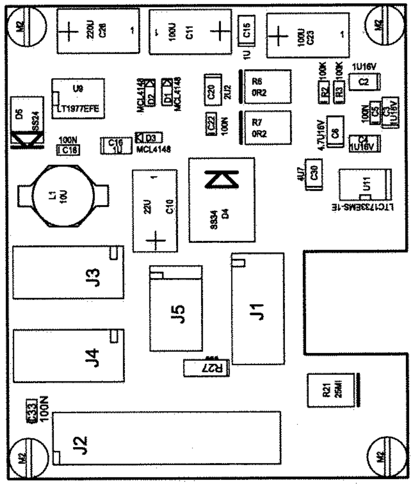


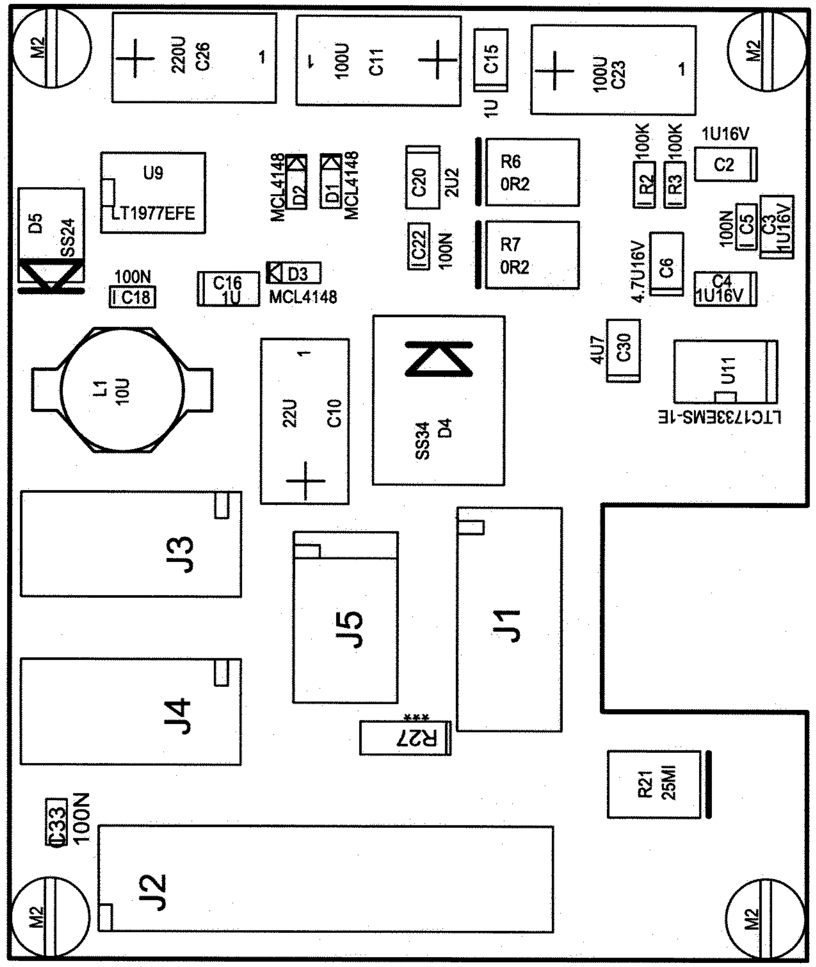
EPFL ACORT Mon 11.04.2005 8:20 Drawing name = SALAMPOWER Scale = 4.00 0.250 Inch =

Page 10

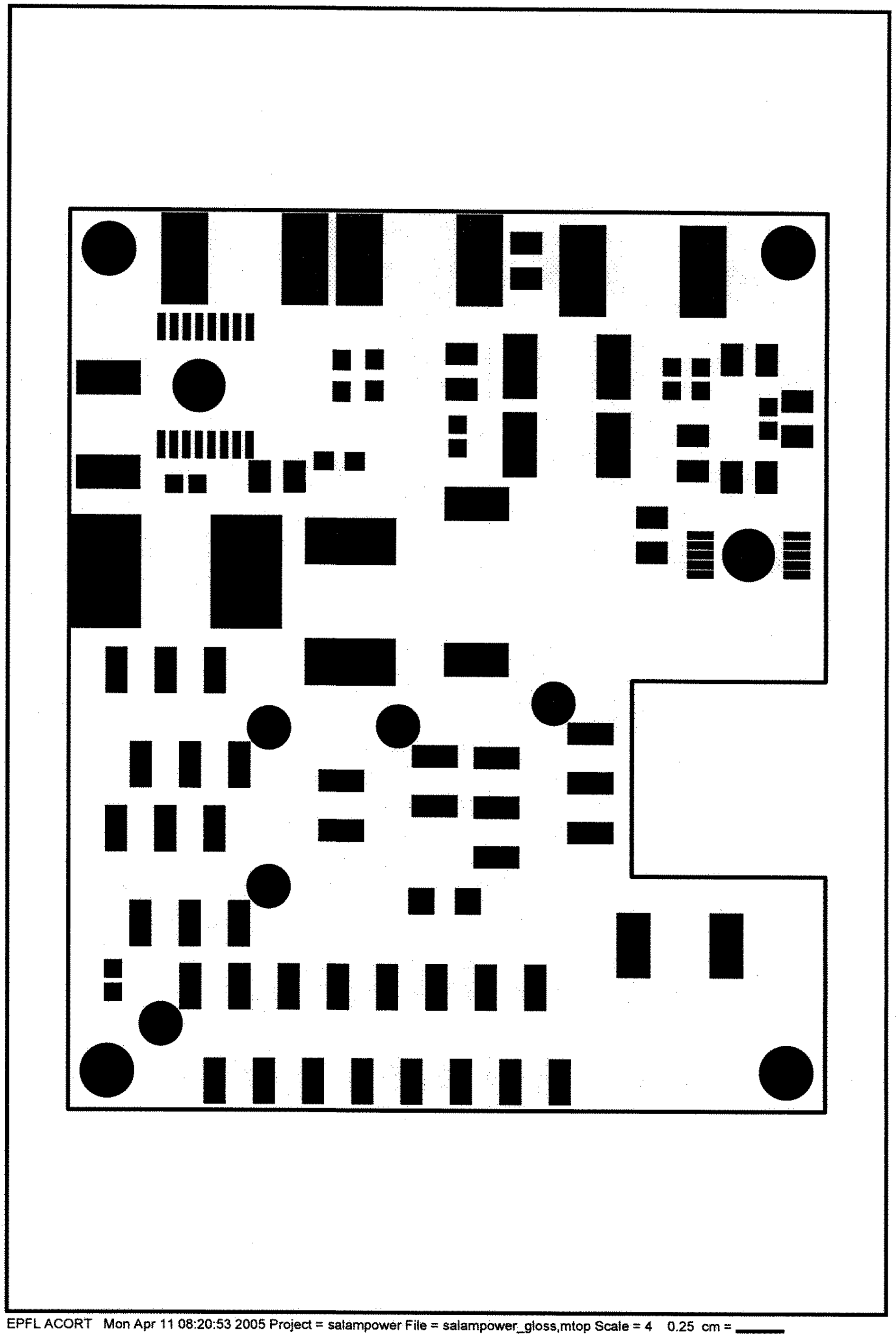


SALAMPOWER	LABORATOIRE:LSL	DATE:11.04.05
	ENGINEER:A.Crespi	EPFL-ACORT-gv

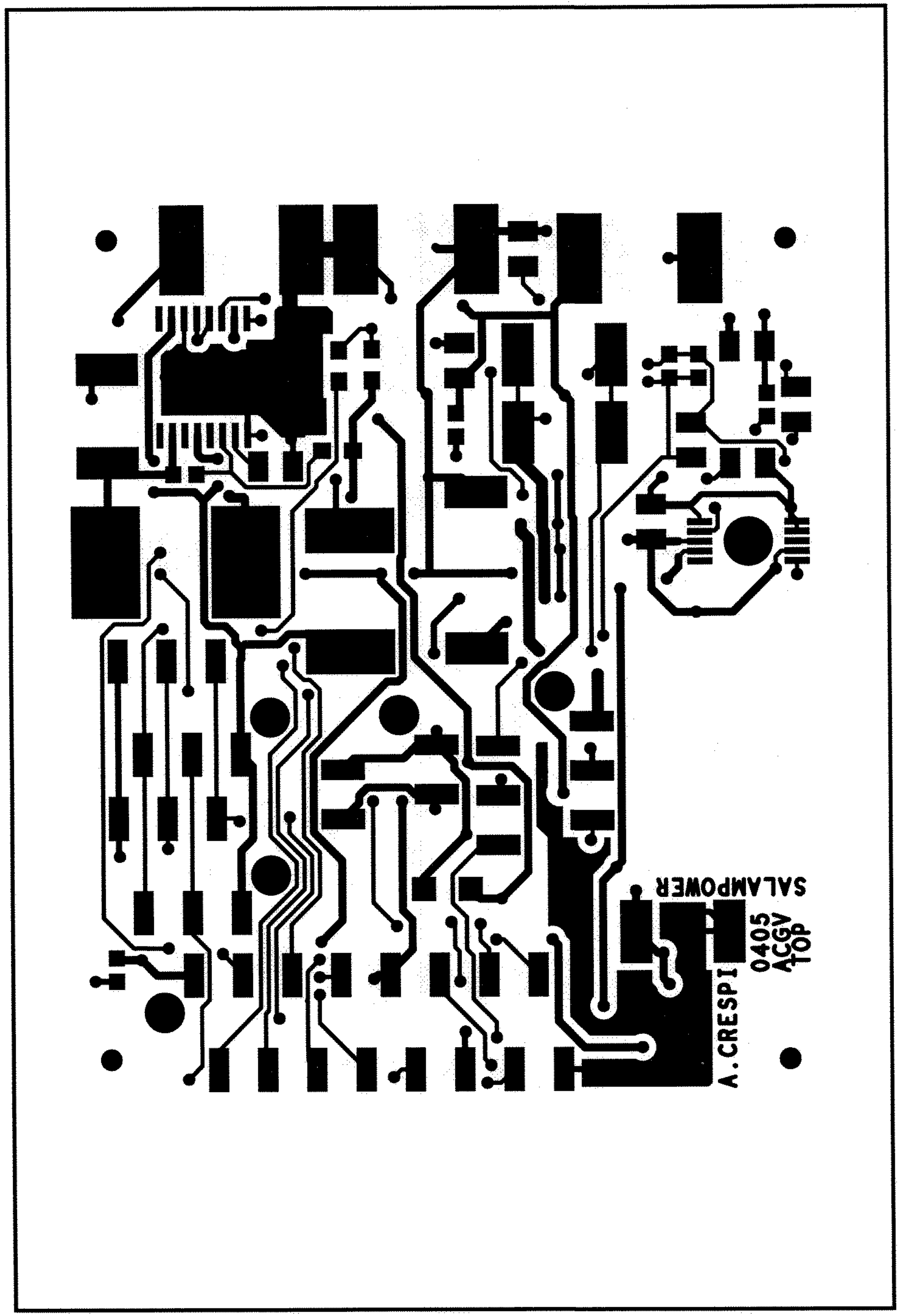




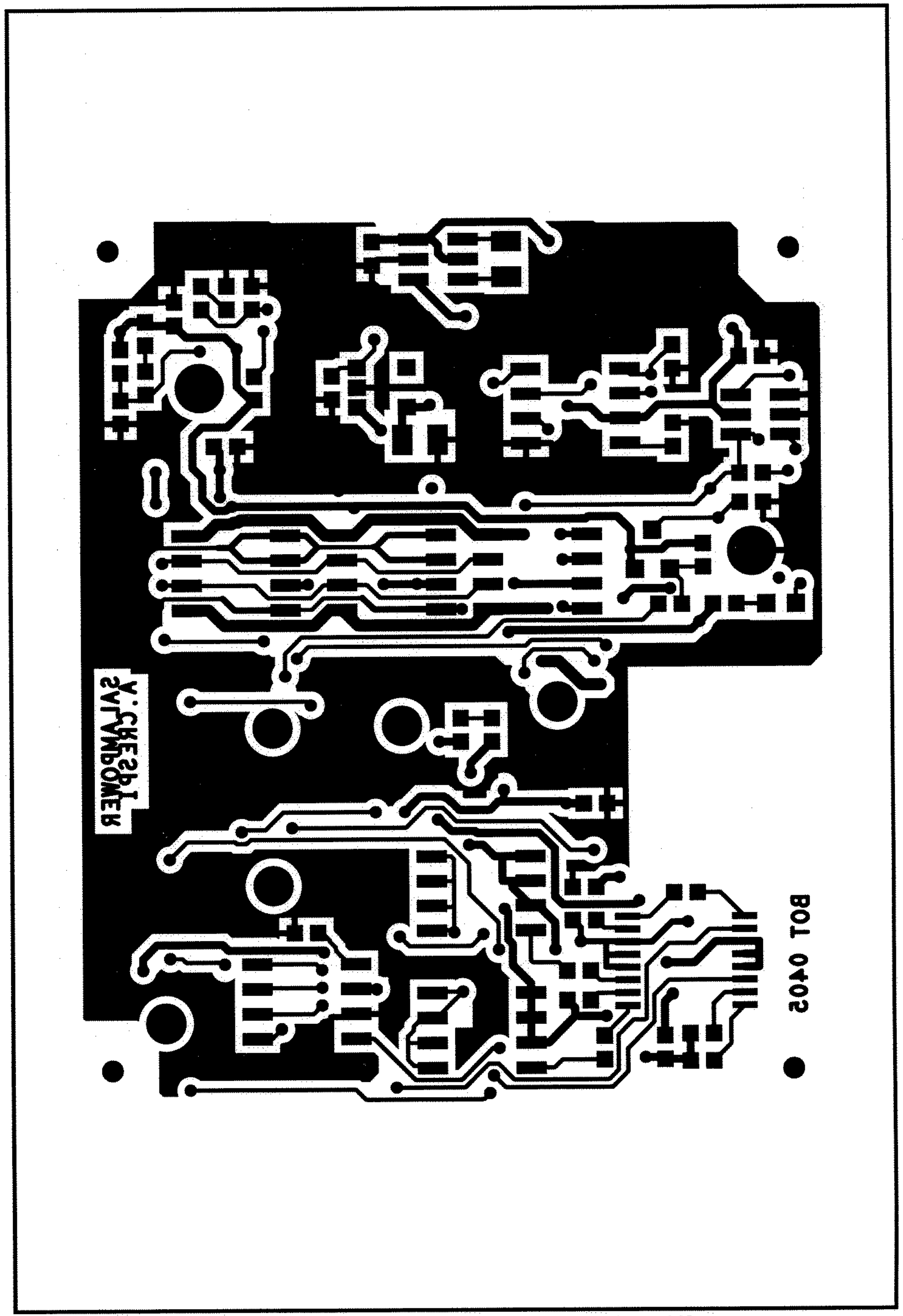
EPFL ACORT Mon 11.04.2005 8:20 Drawing name = SALAMPOWER Scale = 4.00 0.250 Inch =



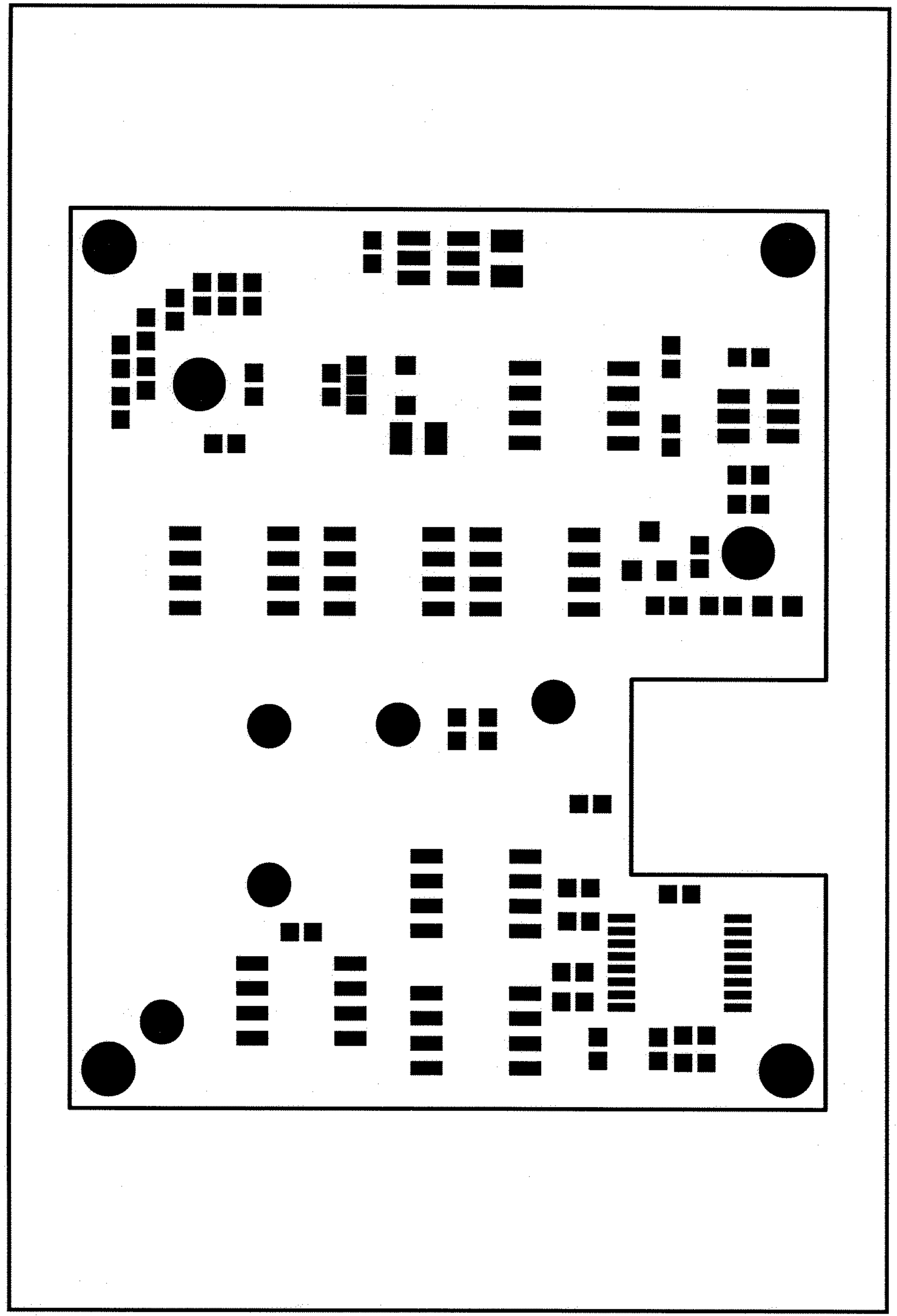
EPFL ACORT Mon Apr 11 08:20:53 2005 Project = salampower File = salampower_gloss,mtop Scale = 4 0.25 cm =



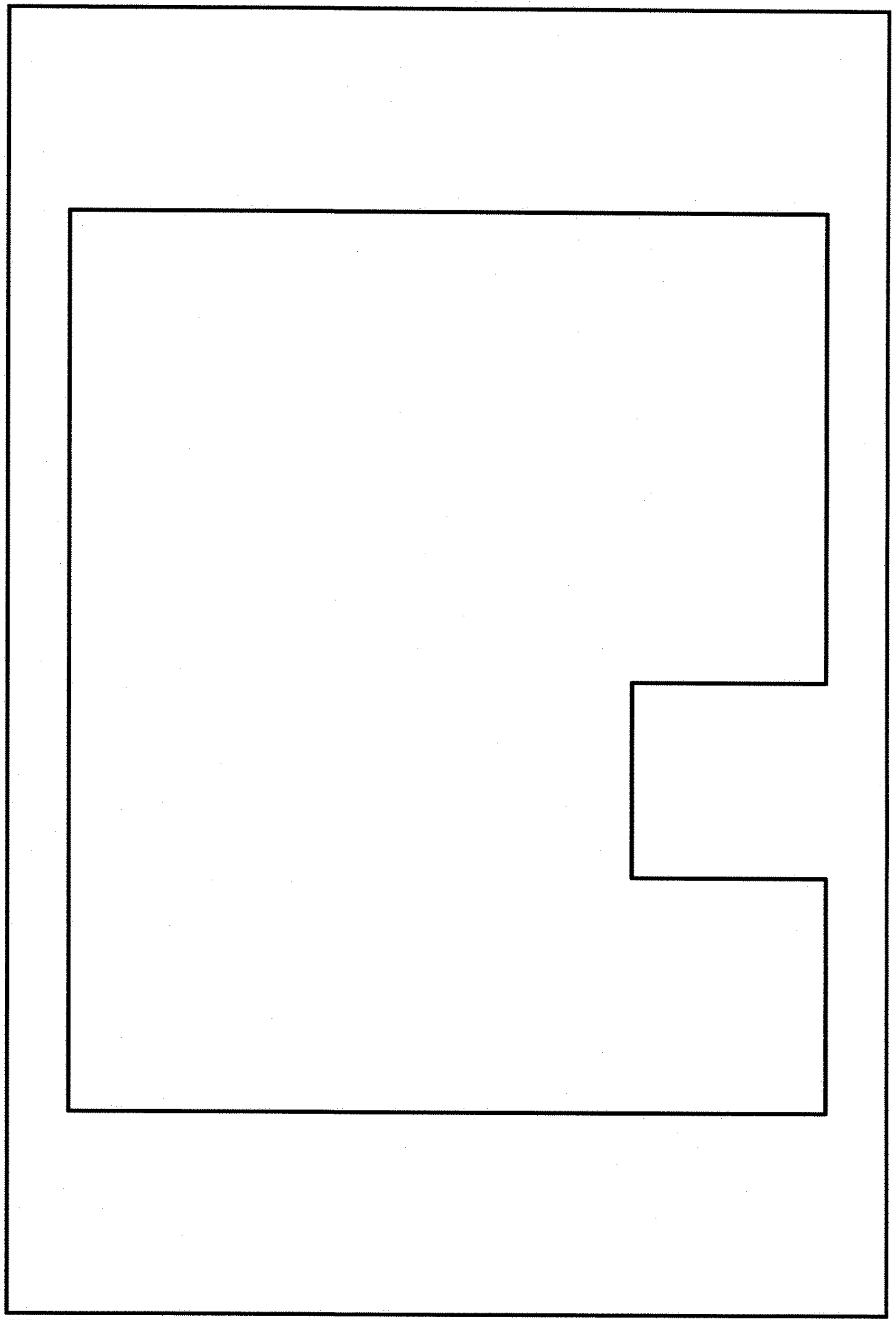
EPFL ACORT Mon Apr 11 08:20:53 2005 Project = salampower File = salampower_gloss,top Scale = 4 0.25 cm =



EPFL ACORT Mon Apr 11 08:20:53 2005 Project = salampower File = salampower_gloss,bot Scale = 4 0.25 cm =



EPFL ACORT Mon Apr 11 08:20:53 2005 Project = salampower File = salampower_gloss,mbot Scale = 4 0.25 cm = ______



EPFL ACORT Mon Apr 11 08:20:53 2005 Project = salampower File = salampower_gloss,cont Scale = 4 0.25 cm = ______

Component Report SALAMPOWER

salampower_gloss Mon Apr 11 08:21:07 MET DST 2005

Ref Des	Device Type	Value	Package Type	x	y	ang	Mir	Remark
C1	0603	100N	0603_G	15.500	-10.500	180.000	YES	
C2	0805	1U16V	0805_G	15.500	-10.500	180.000	NO	
C3	0805	1U16V	0805_G	12.500	-13.000	90.000	NO	
C4	0805	1U16V	0805_G	9.500	-10.500	180.000	NO	
C5	0603	100N	0603_G	12.500	-11.500	90.000	NO	
C6	0805	4.7U16V	0805_G	10.700	-7.600	90.000	NO	
C7	0603	100N	0603_G	-7.400	-2.400	180.000	YES	
C8	0603	100N	0603_G	11.500	-6.500	90.000	YES	
C9	0603	100N	0603_G	11.000	16.500	180.000	YES	
C10	CP	22U	CP_2917	3.000	10.000	270.000	NO	
C11	CP	100U	CP_2917	20.500	6.500	0.000	NO	
C12	0603	330P	0603_G	18.600	15.100	270.000	YES	
C13	0603	100N	0603_G	20.800	8.900	90.000	YES	
C14	0603	1N5	0603_G	18.600	16.400	270.000	YES	
C15	0805	1U	0805_G	20.500	1.000	90.000	NO	
C16	0805	1U	0805_G	9.400	13.800	0.000	NO	
C17	0805	1U	0805_G	20.500	2.000	90.000	YES	
C18	0603	100N	0603_G	9.000	18.500	0.000	NO	
C19	0603	100N	0603_G	14.000	15.000	90.000	YES	
C20	0805	2U2	0805_G	14.800	4.300	270.000	NO	
C21	0805	4U7	0805_G	11.300	6.500	180.000	YES	
C22	0603	100N	0603_G	11.500	4.500	90.000	NO	
C23	CP	100U	CP_2917	20.000	-5.000	180.000	NO	
C24	0603	100N	0603_G	15.400	21.900	90.000	YES	
C25	0603	100N	0603_G	17.800	19.100	270.000	YES	
C26	CP	220U	CP_2917	20.500	15.500	180.000	NO	
C27	0603	100N	0603_G	-19.900	-5.900	270.000	YES	
C28	0603	1N	0603_G	-11.700	-1.800	0.000	YES	
C29	0603	100N	0603_G	-12.000	-7.000	0.000	YES	
C30	0805	4U7	0805_G	6.500	-5.500	90.000	NO	
C31	0603	1N	0603_G	-3.600	2.900	90.000	YES	
C32	0603	1N	0603_G	-3.600	4.500	270.000	YES	
C33	0603	100N	0603_G	-16.400	22.200	90.000	NO	
D1	1N4148	MCL4148	MICROMELF	14.600	8.800	270.000	NO	
D2	1N4148	MCL4148	MICROMELF	14.550	10.500	270.000	NO	
D3	1N4148	MCL4148	MICROMELF	10.200	10.600	0.000	NO	
D4	SS34_SMD	SS34	DO214AB	4.050	3.500	270.000	NO	
D5	SS24	SS24	1812_D	12.000	22.500	90.000	NO	
D6	PG1101H	PG1101C	SMDLED_MINI	2.800	-12.000	180.000	YES	
J1	CO6		MM6_SMD	-3.700	1.300	180.000	NO	
J2	CO16		MM16_SMD	-20.370	18.207	270.000	NO	
J3	CO6		MM6_SMD	-1.700	15.700	90.000	NO	
J4	CO6		MM6_SMD	-9.800	15.700	90.000	NO	
J5	CO4		MM4_SMD	-4.900	9.300	180.000	NO	
L1	L	10U	UP1B	4.500	19.000	270.000	NO	
R1	R	1M	0603_G	-14.000	12.500	180.000	YES	
R2	R	100K	0603_G	14.500	-6.500	90.000	NO	
R3	R	100K	0603_G	14.500	-8.000	90.000	NO	
R4	R	100K	0603_G	9.500	-10.500	0.000	YES	
R5	R	1K	0603_G	15.500	-6.500	90.000	YES	
R6	R	0R2	1812	15.100	-1.100	0.000	NO	
R7	R	0R2	1812	11.100	-1.100	0.000	NO	
R8	R	100K	0603_G	14.000	11.000	90.000	YES	
R9	R	27K	0603_G	18.600	17.700	270.000	YES	
R13	R	220K	0603_G	14.300	20.600	270.000	YES	
R14	R	100K	0603_G	12.800	21.900	90.000	YES	
R15	R	39K	0603_G	16.800	20.600	270.000	YES	
R16	R	1K	0603_G	-20.600	-7.800	0.000	YES	

Component Report SALAMPOWER

salampower_gloss Mon Apr 11 08:21:07 MET DST 2005

Ref Des	Device Type	Value	Package Type	x	y	ang	Mir	Remark
R17	R	150	0603_G	-17.500	-1.500	0.000	YES	
R18	R	4K7	0603_G	-13.400	-1.800	0.000	YES	
R19	R	47K	0603_G	2.800	-6.300	0.000	YES	
R20	R	47K	0603_G	5.300	-8.000	270.000	YES	
R21	R	25MI	1812	-14.500	-7.000	180.000	NO	
R22	R	1K	0603_G	2.800	-9.100	0.000	YES	
R23	R	2K5	0603_G	8.000	-10.500	180.000	YES	
R24	R	150	0603_G	-19.200	-7.800	180.000	YES	
R25	R	1K	0603_G	-19.900	-2.800	270.000	YES	
R26	R	1K	0603_G	-16.000	-1.500	180.000	YES	
R27	R	***	0805	-12.300	5.100	180.000	NO	
T2	BC856A	BC856A	SOT23	5.600	-5.400	180.000	YES	
T3	IRF7410	IRF7410	SO8	-19.000	3.500	270.000	YES	
T4	IRF7410	IRF7410	SO8	-12.000	3.500	270.000	YES	
U1	DS2764BE		TSSOP16	-15.500	-7.005	90.000	YES	
U2	SI9986CY		SO8	4.500	8.000	270.000	YES	
U3	SI9986CY		SO8	4.500	16.000	270.000	YES	
U4	SI9986CY		SO8	4.500	0.500	270.000	YES	
U5	MAX1719EUT		SOT23_6	12.500	-11.000	90.000	YES	
U6	LS7084S		SO8	-17.500	12.500	270.000	YES	
U7	INA146UA		SO8	13.000	-1.500	90.000	YES	
U8	LTC4411ES5		SOT23_5G	14.000	8.437	90.000	YES	
U9	LT1977EFE		TSSOP16_REF	14.005	17.500	0.000	NO	
U10	LTC3200		SOT23_6	20.500	5.500	90.000	YES	
U11	LTC1733EMS		SO10_05MM_GND	5.500	-10.500	90.000	NO	

Total Component count 83

BOM Report SALAMPOWER

salampower_gloss Mon Apr 11 08:21:08 MET DST 2005

Device	Package	Value	Nb	Reference Designators	Remark
0603-1-100N	0603_G	100N	14	C1 C5 C7 C8 C9 C13 C18 C19 C22 C24 C25 C27 C29 C33	
0603-1-1N	0603_G	1N	3	C28 C31 C32	
0603-1-1N5	0603_G	1N5	1	C14	
0603-1-330P	0603_G	330P	1	C12	
0805-1-1U	0805_G	1U	3	C15 C16 C17	
0805-1-1U16V	0805_G	1U16V	3	C2 C3 C4	
0805-1-2U2	0805_G	2U2	1	C20	
0805-1-4.7U16V	0805_G	4.7U16V	1	C6	
0805-1-4U7	0805_G	4U7	2	C21 C30	
1N4148-SMD1-MCL4148	MICROMELF	MCL4148	3	D1 D2 D3	
BC856A	SOT23	BC856A	1	T2	
CO16-26	MM16_SMD		1	J2	
CO4-33	MM4_SMD		1	J5	
CO6-31	MM6_SMD		3	J1 J3 J4	
CP-10V-SMD220UF-220U	CP_2917	220U	1	C26	
CP-10VSMD100U-100U	CP_2917	100U	1	C11	
CP-35V-SMD22UF-22U	CP_2917	22U	1	C10	
CP-6.3VSMD100U-OBSOL	CP_2917	100U	1	C23	
DS2764BE	TSSOP16		1	U1	
INA146UA	SO8		1	U7	
IRF7410	SO8	IRF7410	2	T3 T4	
L-27K-10U	UP1B	10U	1	L1	
LS7084S	SO8		1	U6	
LT1977EFE	TSSOP16_REF		1	U9	
LTC1733EMS-1E	SO10_05MM_GND		1	U11	
LTC3200-5	SOT23_6		1	U10	
LTC4411ES5	SOT23_5G		1	U8	
MAX1719EUT	SOT23_6		1	U5	
PG1101H	SMDLED_MINI	PG1101C	1	D6	
R-13-***	0805	***	1	R27	
R-16-0R2	1812	0R2	2	R6 R7	
R-16-25MI	1812	25MI	1	R21	
R-23-100K	0603_G	100K	5	R2 R3 R4 R8 R14	
R-23-150	0603_G	150	2	R17 R24	
R-23-1K	0603_G	1K	5	R5 R16 R22 R25 R26	
R-23-1M	0603_G	1M	1	R1	
R-23-220K	0603_G	220K	1	R13	
R-23-27K	0603_G	27K	1	R9	
R-23-2K5	0603_G	2K5	1	R23	
R-23-39K	0603_G	39K	1	R15	
R-23-47K	0603_G	47K	2	R19 R20	
R-23-4K7	0603_G	4K7	1	R18	
SI9986CY	SO8		3	U2 U3 U4	
SS24	1812_D	SS24	1	D5	
SS34_SMD	DO214AB	SS34	1	D4	

Total Component count 83

NC Pins Report SALAMPOWER

salampower_gloss Mon Apr 11 08:21:08 MET DST 2005

Ref Des	Device	Nb	Not Connected Pins	Remark
U8	LTC4411ES5	1	4	
U9	LT1977EFE	1	16	
U11	LTC1733EMS-1E	2	3 10	

Total count 4

Power Pins Report SALAMPOWER

salampower_gloss — Mon Apr 11 08:21:09 MET DST 2005

Ref Des	Device	Name	Power Pins	Remark
U1	DS2764BE	GND	4 5 6	
		GNDB	11 12 13	
U2	SI9986CY	GND	1 2 4	
		VCCC	3	
U3	SI9986CY	GND	1 2 4	
		VCCC	3	
U4	SI9986CY	GND	1 2 4	
		VCCC	3	
U5	MAX1719EUT	-5V	1	
		GND	4 5	
		VCC	2	
U6	LS7084S	GND	3	
		VCC	2	
U7	INA146UA	-5V	4	
		GND	1	
		VCC	7	
U8	LTC4411ES5	GND	2	
		VCCC	5	
		VCCF	1	
U9	LT1977EFE	GND	8 14 17	
		VCCD	10	
		VCCE	4 15	
U10	LTC3200-5	GND	2	
		VCC	1	
		VCCC	3 5	
U11	LTC1733EMS-1E	GND	4 5 6 11	
		VCCD	2 8	
		VCCF	9	

Total count 11

Single Node Nets Report SALAMPOWER

salampower_gloss — Mon Apr 11 08:21:09 MET DST 2005

Netname	Node	Device	Remark

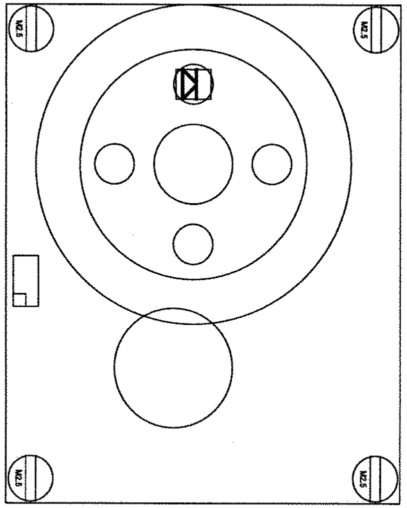
Total Nets count 0

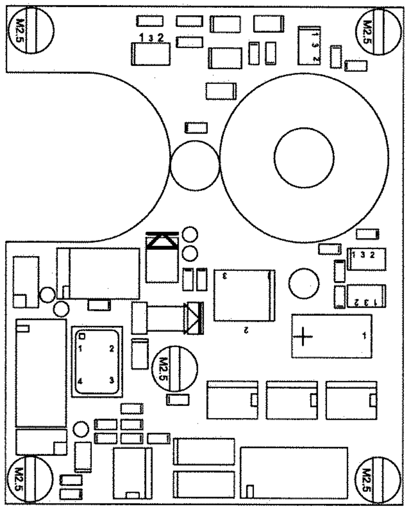
E.3 Power board Amphibot I - head module

SALAMPATTE_B

LSL A.Crespi
Phone: 36630
Mon Sep 6 18:20:24 MET DST 2004

MFG_DATE:	06.09.2004
QUANTITY:	4
SOLDERMASK:	mtop mbot
MATERIAL:	FR4 : 0.8 mm

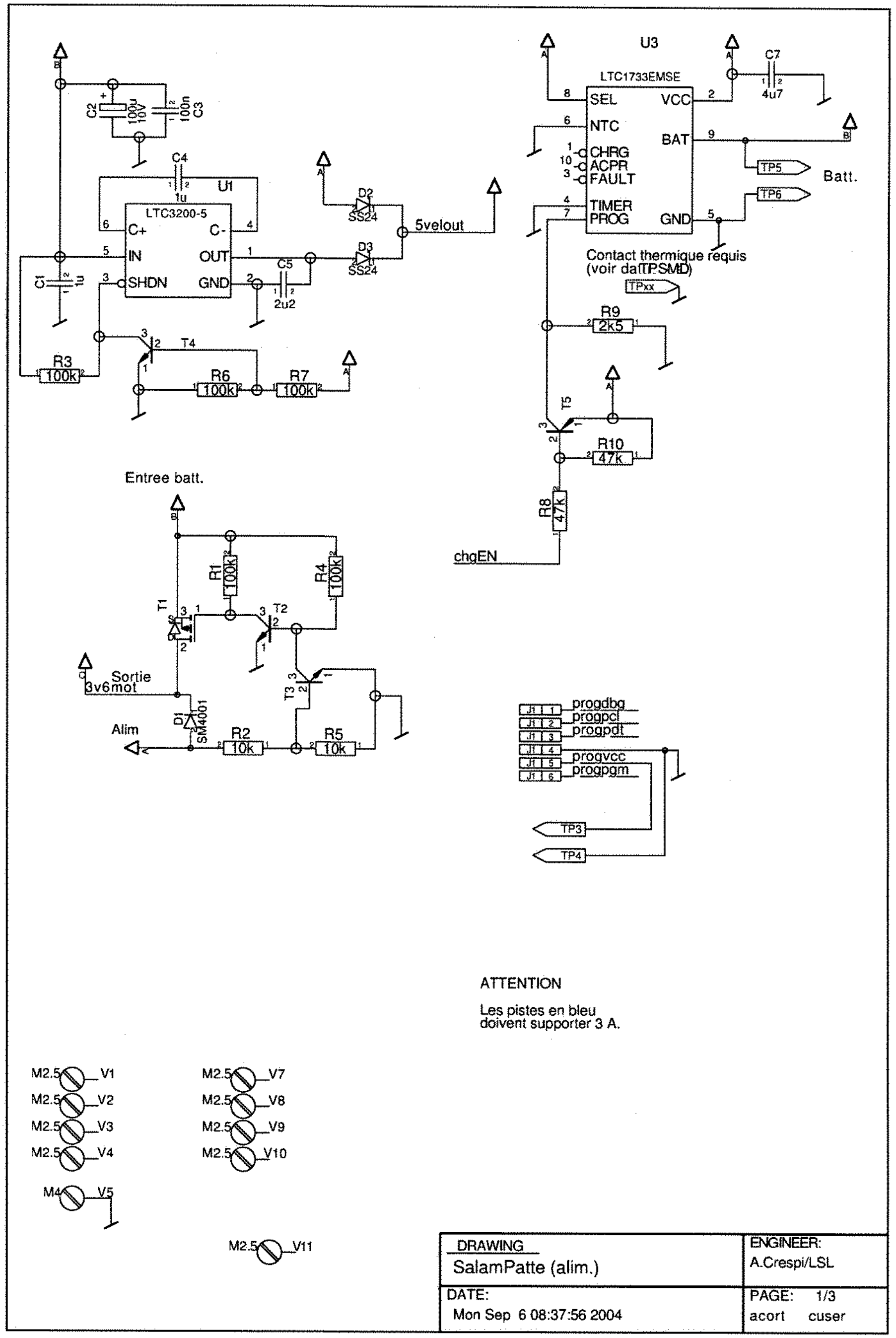




CONTENTS	PAGE
Root Schema salampatte	2 - 4
Xref salampatte	5
Dimensions	6
Drill	7
Etch	8
Outline_Bot_Scale2_0	9
Outline_Top	10
Outline_Top_Scale2_0	11
Gerber_mtop	12
Gerber_top	13
Gerber_bot	14
Gerber_mbot	15
Gerber_cont	16
Component Report	17 - 18

CONTENTS	PAGE
BOM Report	19
NC-Pins Report	20
Power-Pins Report	21
Single Node Nets	22

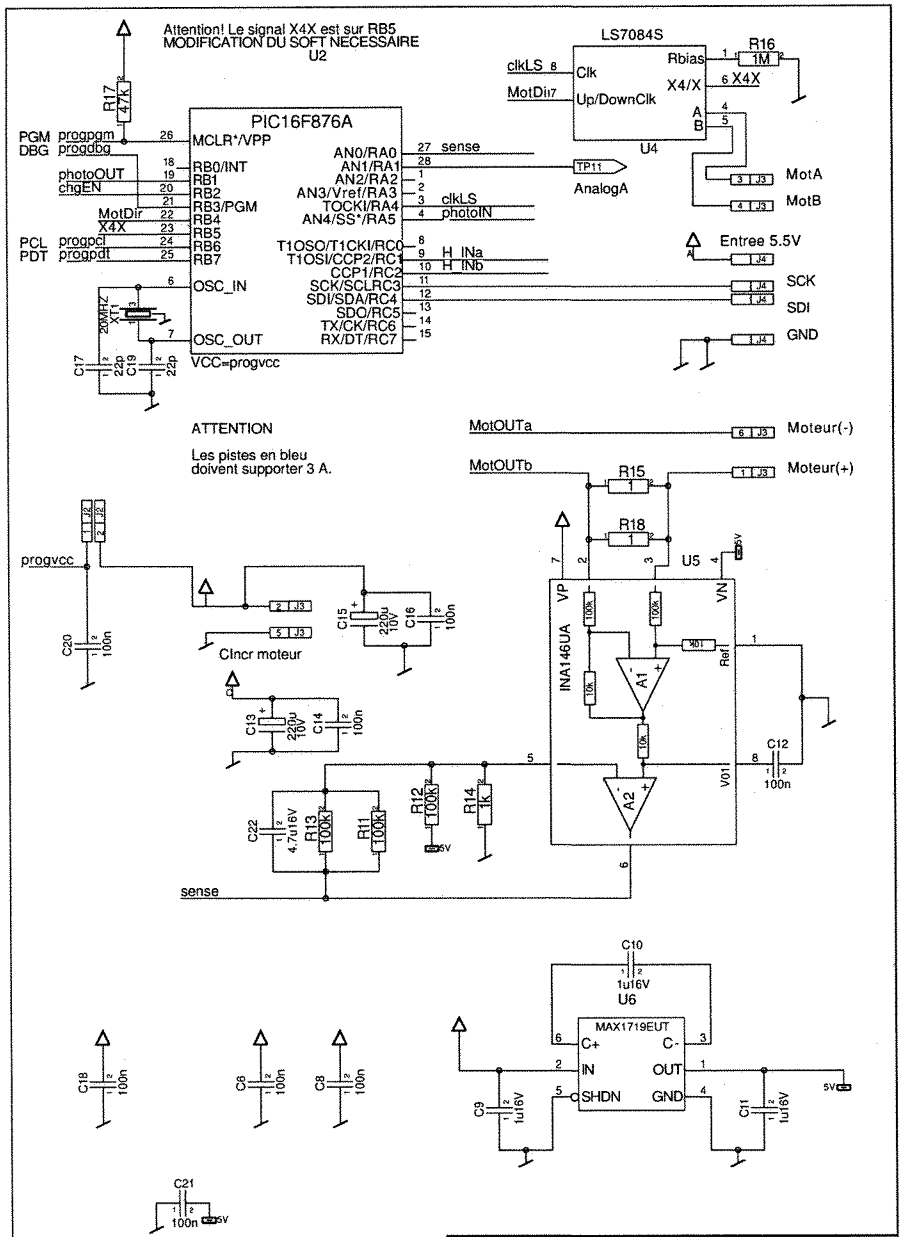
ACORT



ATTENTION

Les pistes en bleu
doivent supporter 3 A.

DRAWING SalamPatte (alim.)	ENGINEER: A.Crespi/LSL
DATE: Mon Sep 6 08:37:56 2004	PAGE: 1/3 acort cuser

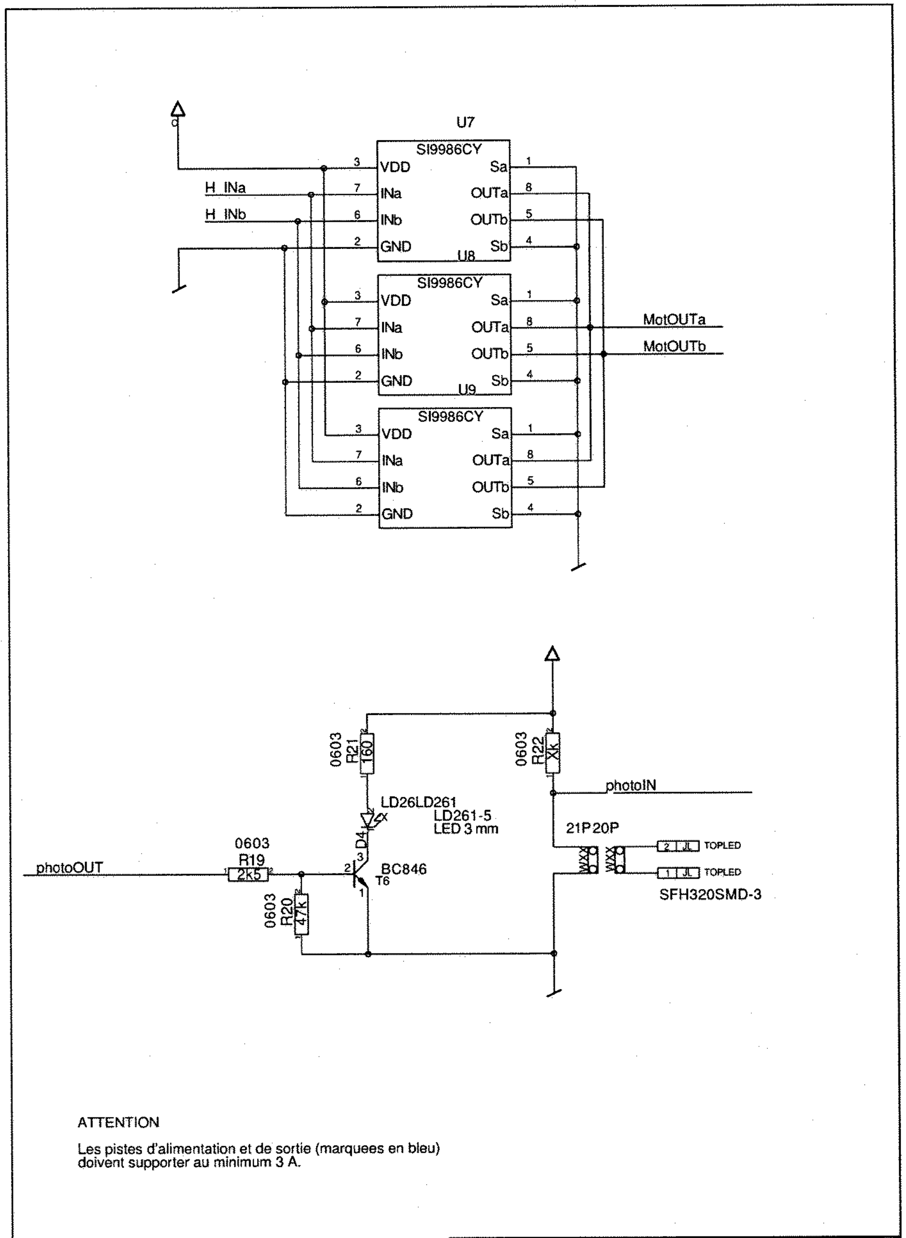


Attention! Le signal X4X est sur RB5
MODIFICATION DU SOFT NECESSAIRE

ATTENTION

Les pistes en bleu
doivent supporter 3 A.

DRAWING	ENGINEER:
SalamPatte (control.)	A.Crespi/LSL
DATE: Mon Sep 6 16:50:28 2004	PAGE: 2/3 acort cuser



ATTENTION

Les pistes d'alimentation et de sortie (marquees en bleu)
doivent supporter au minimum 3 A.

DRAWING	ENGINEER:
SalamPatte (p.H + FT)	A.Crespi/LSL
DATE: Fri Sep 3 11:35:12 2004	PAGE: 3/3 acort cuser

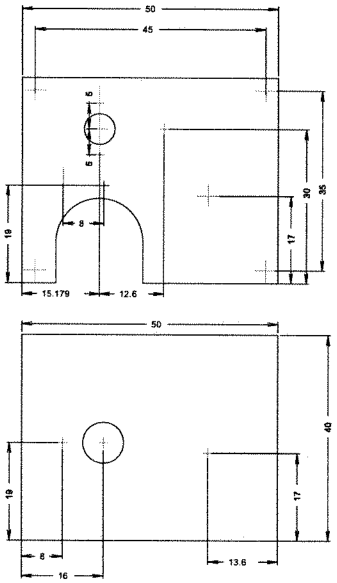
Signal Page Ref SALAMPATTE_B

salampatte

Mon Sep 6 18:20:25 MET DST 2004

Ref Des	Pages	
3v6mot	1	
5velout	1	
chgEN	1	2
clkLS	2	2
H_INa	2	3
H_INb	2	3
MotDir	2	2
MotOUTa	2	3
MotOUTb	2	3
photoIN	2	3
photoOUT	2	3
progdbg	1	2
progpcl	1	2
progpdt	1	2
progpgm	1	2
progvcc	1	2
sense	2	2
X4X	2	2

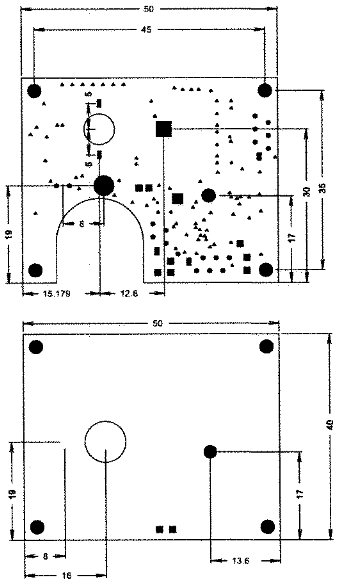
Total Signals count 18



TITRE:SALAMPATTE_B	LABORATOIRE:IC-LSL	DATE:3.09.2004
	ENGINEER:A.Crespi	EPFL-ACORT-PB

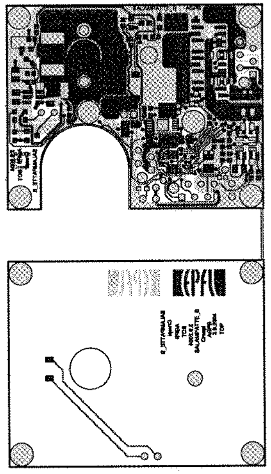
EPFL ACORT Mon 06.09.2004 18:20 Drawing name = SALAMPATTE_B Scale = 1.00 1.000 Inch =

DRILL CHART			
FIGURE	SIZE	PLATED	QTY
·	0.203	PLATED	88
•	0.787	PLATED	19
▪	0.889	PLATED	10
■	1.499	PLATED	3
■	1.6	PLATED	2
■	1.8	PLATED	1
●	2.5	PLATED	9
●	2.5	PLATED	1
■	3.0	PLATED	1
●	4.0	PLATED	1



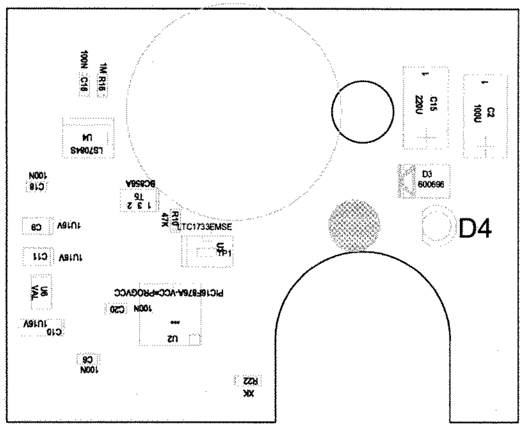
TITRE:SALAMPATTE_B

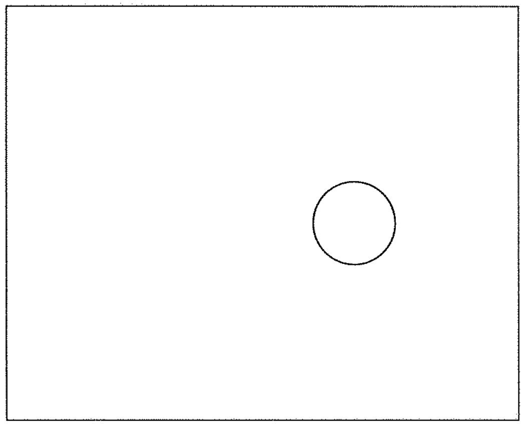
LABORATOIRE:IC-LSL	DATE:3.09.2004
ENGINEER:A.Crespi	EPFL-ACORT-PB

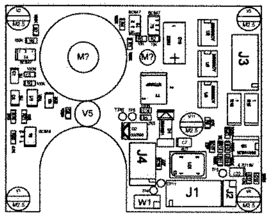


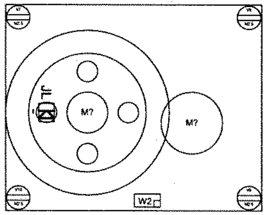
TITRE:SALAMPATTE_B

LABORATOIRE:IC-LSL	DATE:3.09.2004
ENGINEER:A.Crespi	EPFL-ACORT-PB

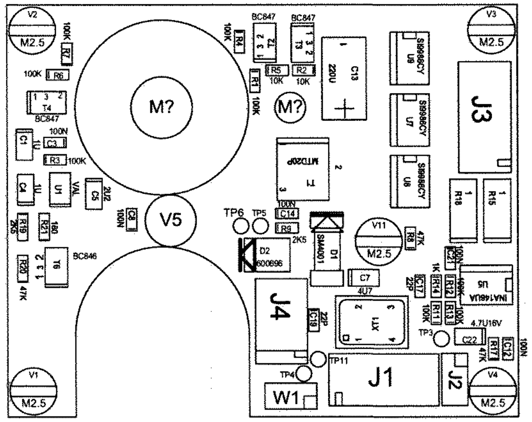


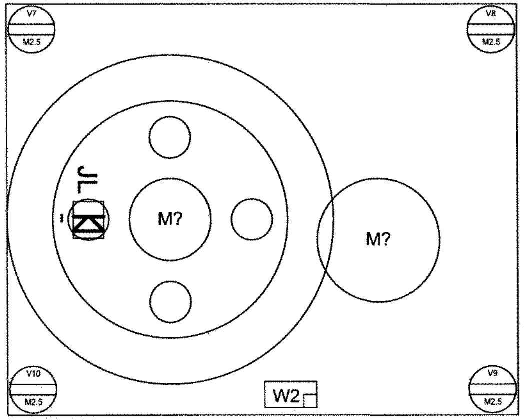




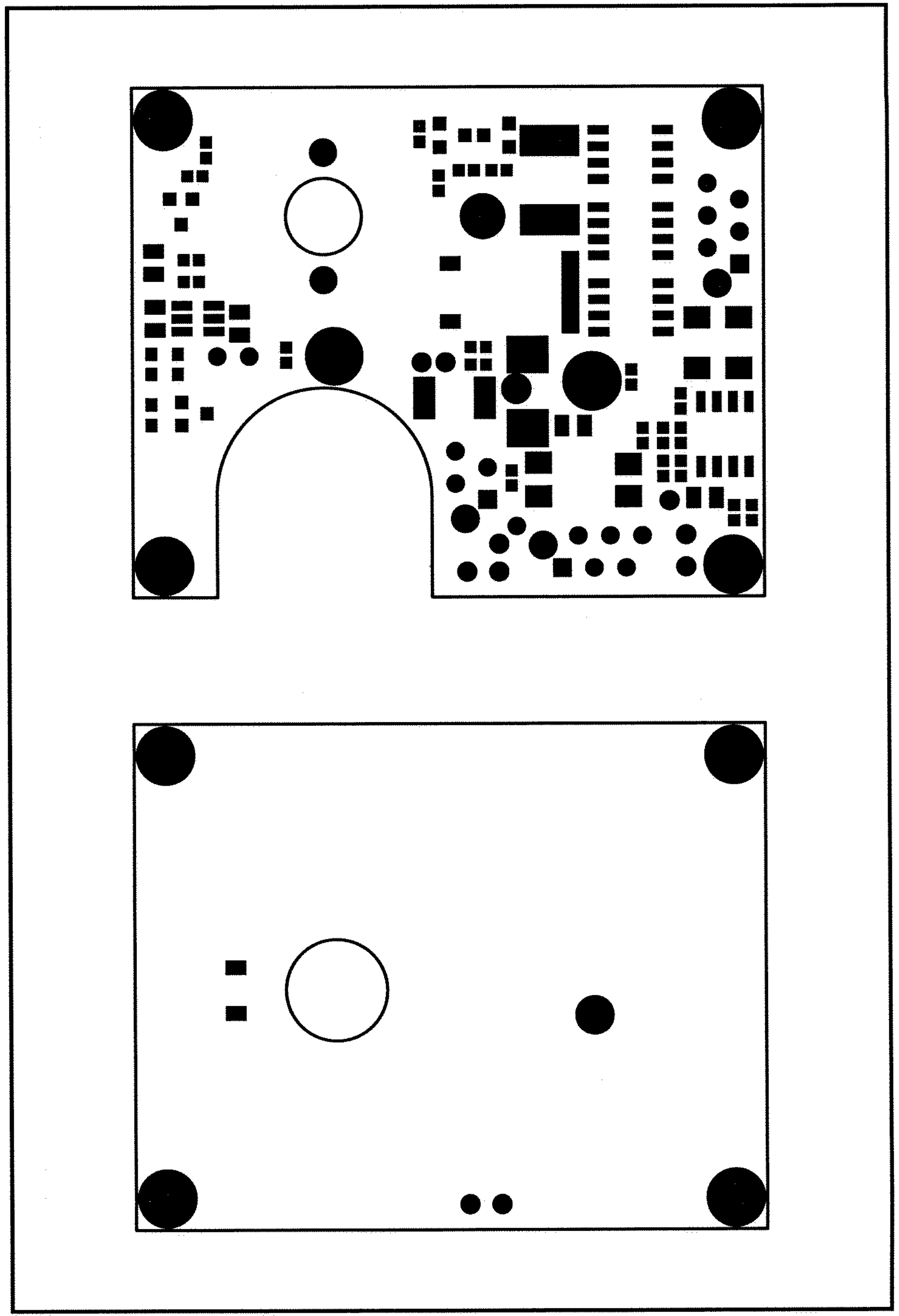


TITRE:SALAMPATTE_B	LABORATOIRE:IC-LSL	DATE:3.09.2004
	ENGINEER:A.Crespi	EPFL-ACORT-PB

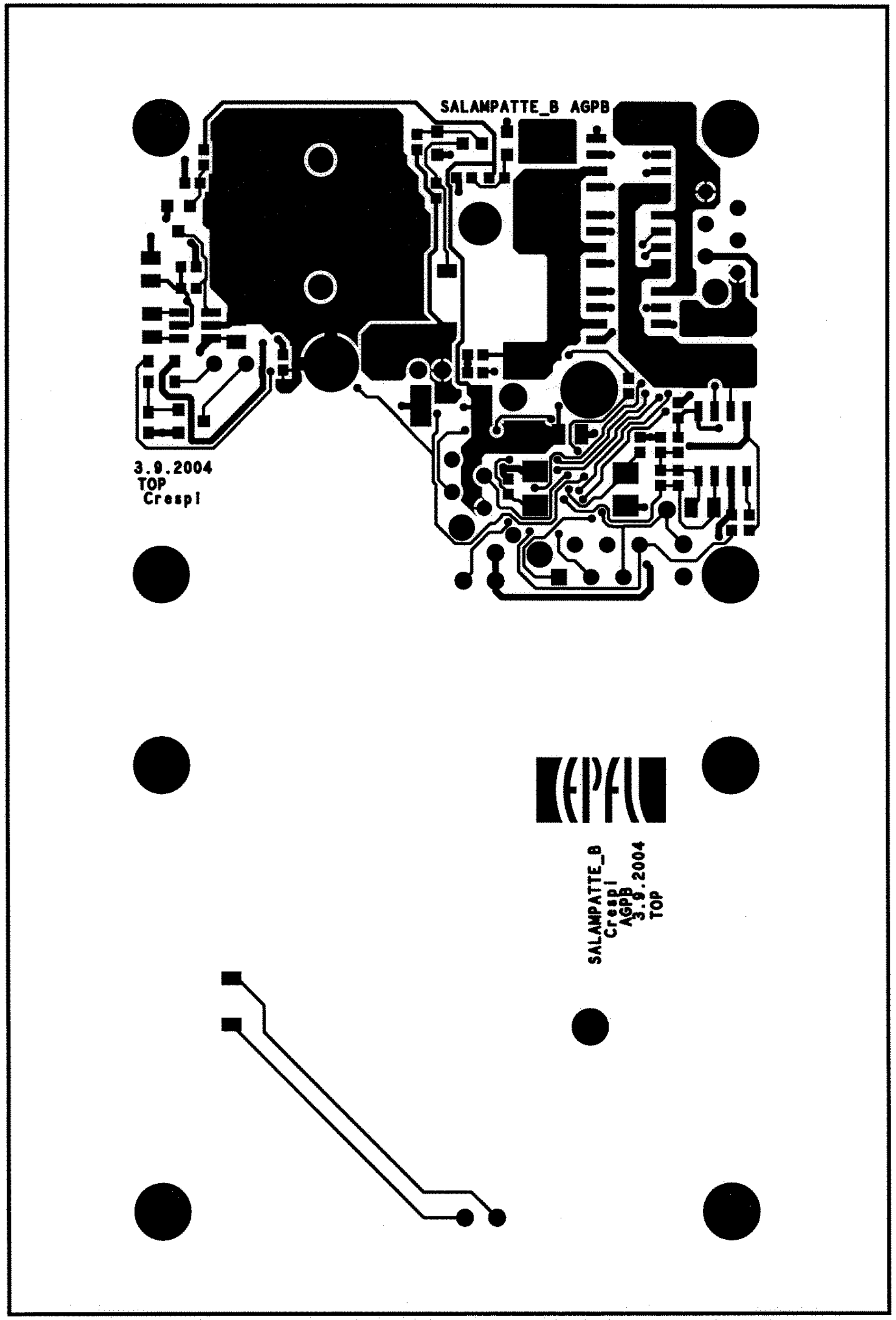




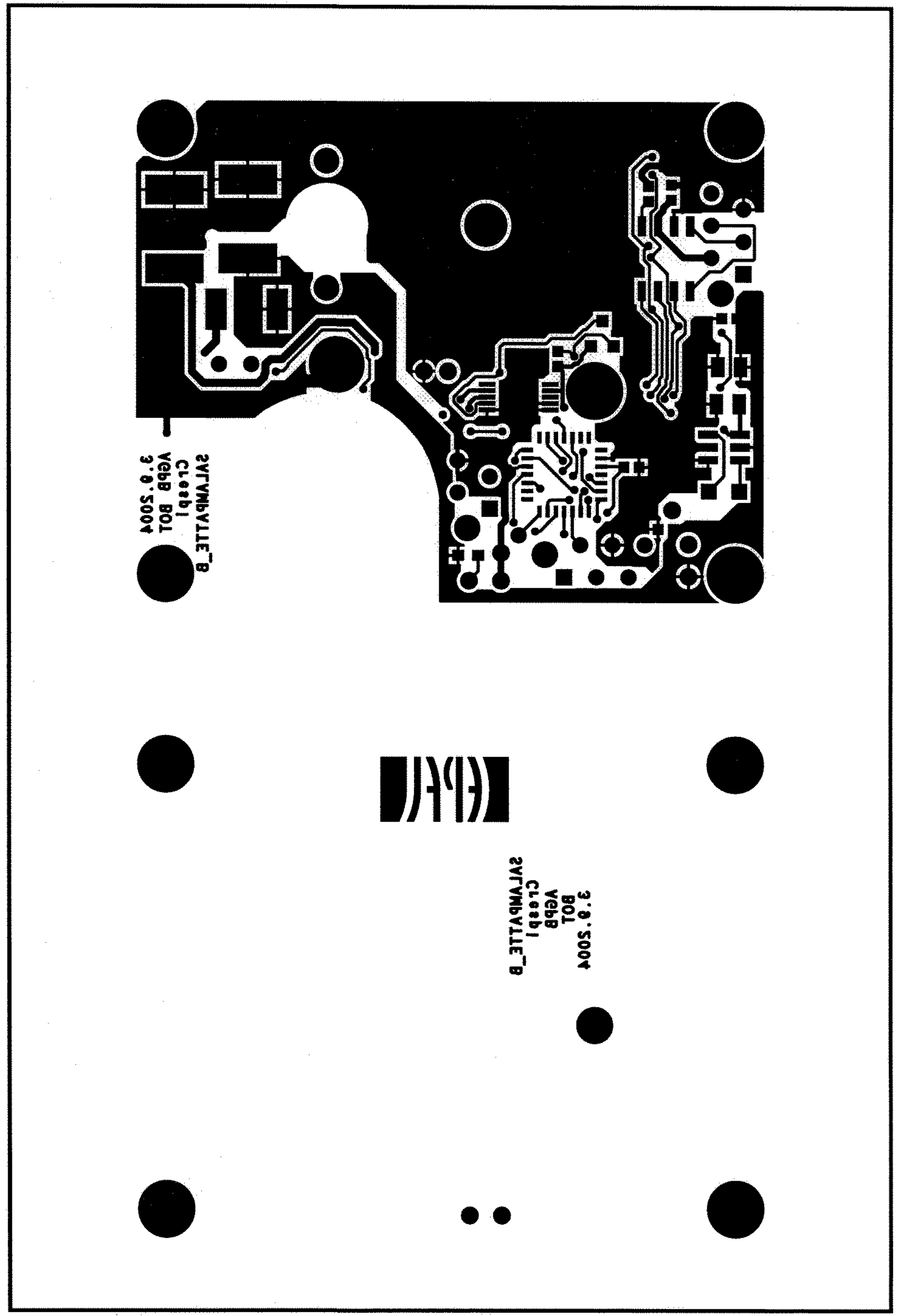
EPFL ACORT Mon 06.09.2004 18:20 Drawing name = SALAMPATTE_B Scale = 2.00 0.500 Inch =



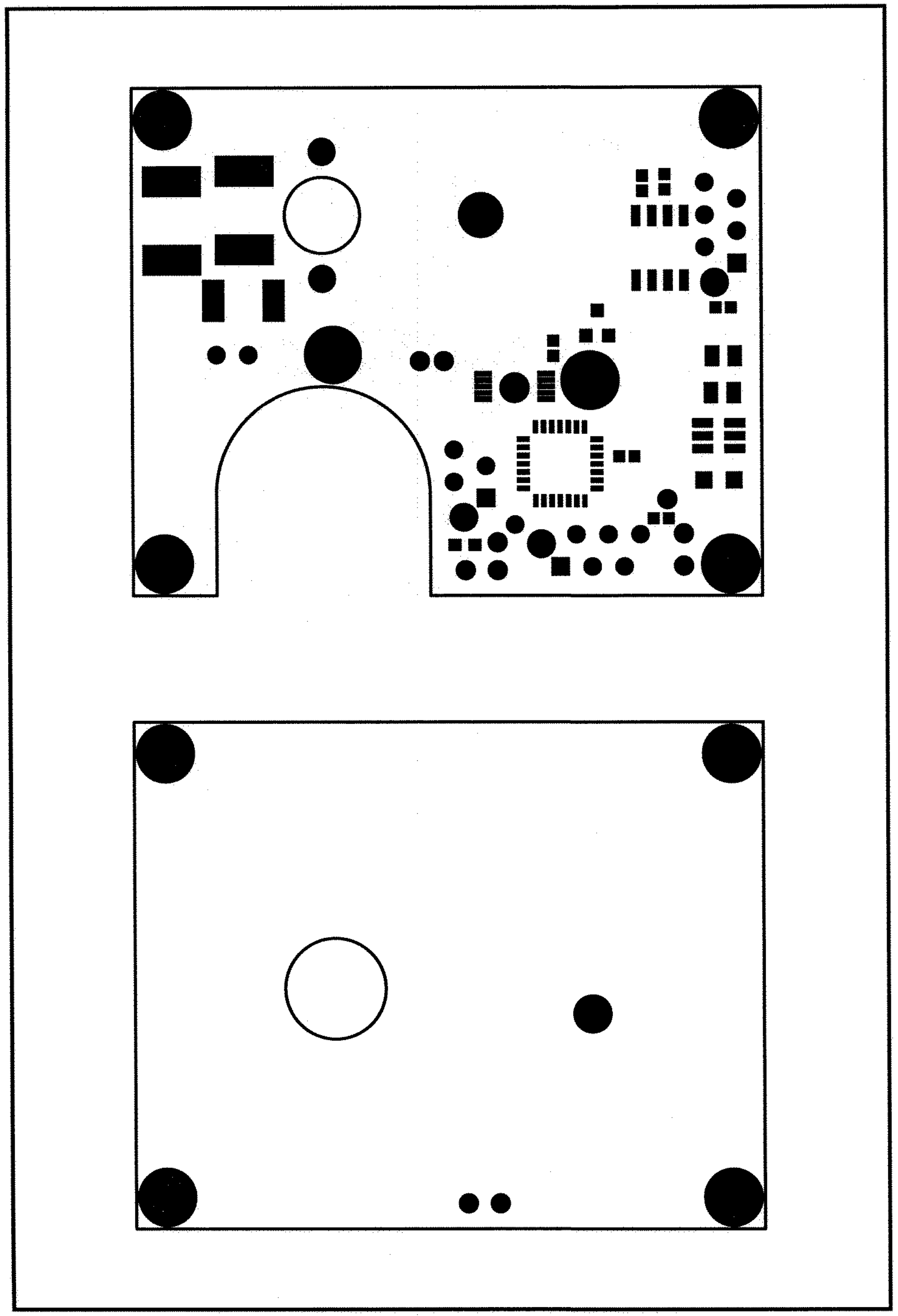
EPFL ACORT Mon Sep 6 18:20:16 2004 Project = salampatte_b File = salampatte_gloss,mtop Scale = 2.6 0.384615384615385 cm =



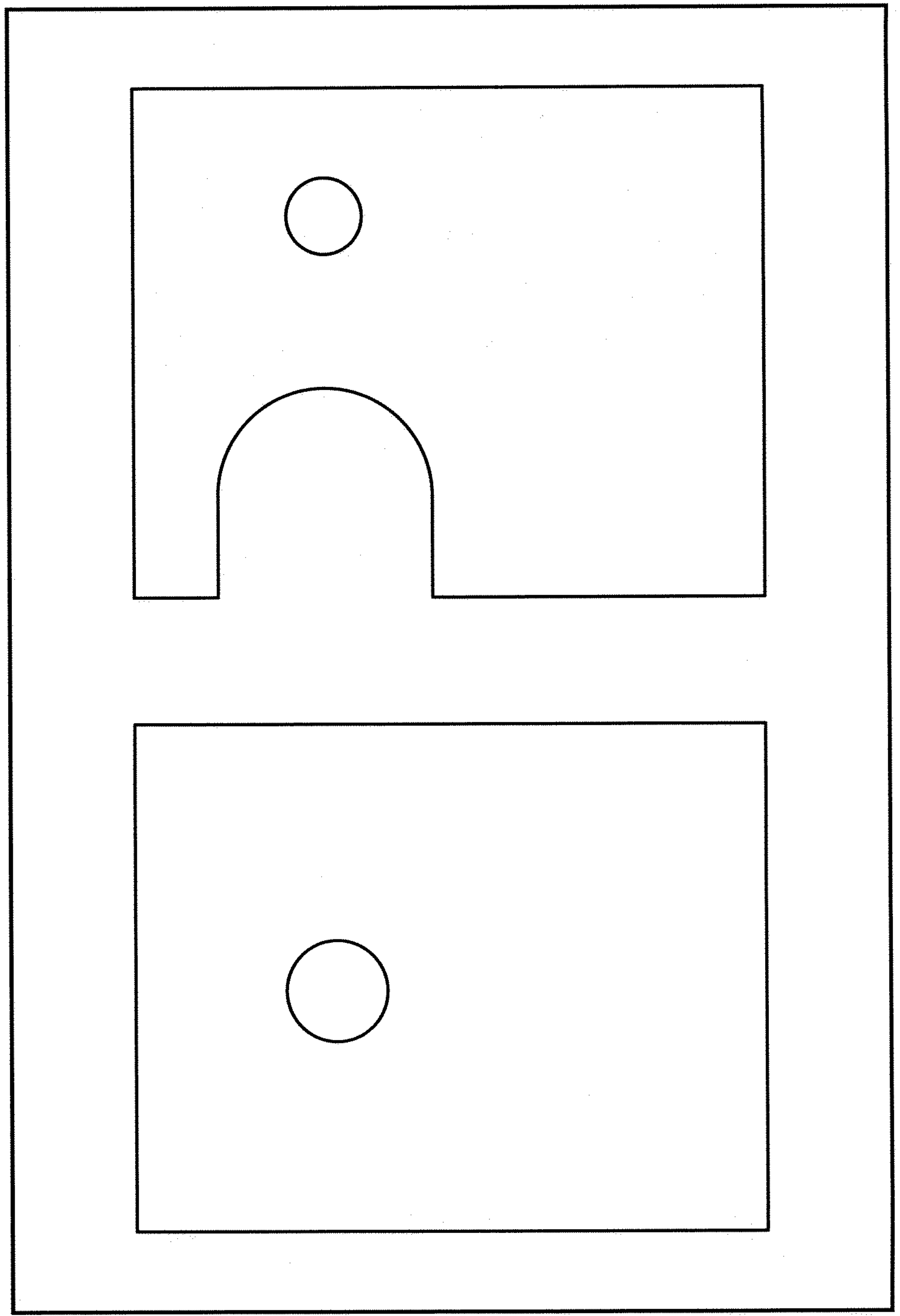
EPFL ACORT Mon Sep 6 18:20:16 2004 Project = salampatte_b File = salampatte_gloss,top Scale = 2.6 0.384615384615385 cm =



EPFL ACORT Mon Sep 6 18:20:16 2004 Project = salampatte_b File = salampatte_gloss,bot Scale = 2.6 0.384615384615385 cm =



EPFL ACORT   Mon Sep 6 18:20:16 2004 Project = salampatte_b File = salampatte_gloss,mbot Scale = 2.6   0.384615384615385 cm =



EPFL ACORT   Mon Sep 6 18:20:16 2004 Project = salampatte_b File = salampatte_gloss,cont Scale = 2.6   0.384615384615385 cm =

Component Report SALAMPATTE_B

salampatte_gloss Mon Sep 6 18:20:26 MET DST 2004

Ref Des	Device Type	Value	Package Type	x	y	ang	Mir	Remark
C1	0805	1U	0805_G	-23.300	26.400	0.000	NO	
C2	CP	100U	CP_2917	-21.800	29.600	0.000	YES	
C3	0603	100N	0603_G	-20.300	26.600	270.000	NO	
C4	0805	1U	0805_G	-23.200	22.000	180.000	NO	
C5	0805	2U2	0805_G	-16.500	21.600	0.000	NO	
C6	0603	100N	0603_G	17.000	6.100	90.000	YES	
C7	0805	4U7	0805_G	9.900	13.500	90.000	NO	
C8	0603	100N	0603_G	-12.800	19.100	180.000	NO	
C9	0805	1U16V	0805_G	22.000	18.900	90.000	YES	
C10	0805	1U16V	0805	21.600	9.100	90.000	YES	
C11	0805	1U16V	0805_G	21.900	16.000	90.000	YES	
C12	0603	100N	0603_G	24.000	6.600	0.000	NO	
C13	CP	220U	CP_2917	8.100	32.800	0.000	NO	
C14	0603	100N	0603_G	2.400	19.700	90.000	NO	
C15	CP	220U	CP_2917	-16.000	30.400	0.000	YES	
C16	0603	100N	0603_G	17.400	32.600	0.000	YES	
C17	0603	22P	0603_G	15.400	12.700	0.000	NO	
C18	0603	100N	0603_G	22.000	22.700	90.000	YES	
C19	0603	22P	0603_G	5.000	9.400	0.000	NO	
C20	0603	100N	0603_G	14.300	11.000	270.000	YES	
C21	0603	100N	0603_G	18.400	15.400	0.000	NO	
C22	0805	4.7U16V	0805_G	20.300	7.800	270.000	NO	
D1	SM4001	SM4001	MELF_D	6.300	16.200	0.000	NO	
D2	SS24	600696	1812_D	0.500	15.700	90.000	NO	
D3	SS24	600696	1812_D	-16.100	23.300	270.000	YES	
D4	1105W90105WJ	X	LED3	-17.000	19.000	0.000	YES	
J1	CO6		MM6	9.000	2.300	0.000	NO	
J2	CO2		JUMP2	18.800	4.900	180.000	NO	
J3	CO6		MM6	23.100	26.200	90.000	NO	
J4	CO4		MM4	3.100	7.700	90.000	NO	
JL	CO2		TOPLED	-17.000	-31.000	0.000	NO	
R1	R	100K	0603_G	-0.700	32.600	0.000	NO	
R2	R	10K	0603_G	4.100	33.600	270.000	NO	
R3	R	100K	0603_G	-20.300	24.900	90.000	NO	
R4	R	100K	0603_G	-2.200	36.400	0.000	NO	
R5	R	10K	0603_G	1.500	33.600	90.000	NO	
R6	R	100K	0603_G	-20.000	33.200	90.000	NO	
R7	R	100K	0603_G	-19.100	35.200	180.000	NO	
R8	R	47K	0603_G	14.500	17.300	180.000	NO	
R9	R	2K5	0603_G	2.400	18.300	90.000	NO	
R10	R	47K	0603_G	8.500	19.500	180.000	YES	
R11	R	100K	0603_G	17.000	10.100	180.000	NO	
R12	R	100K	0603_G	18.400	12.700	0.000	NO	
R13	R	100K	0603_G	18.400	10.100	180.000	NO	
R14	R	1K	0603_G	17.000	12.700	0.000	NO	
R15	R	1	MINIMELF	23.000	20.000	180.000	NO	
R16	R	1M	0603_G	15.600	32.500	180.000	YES	
R17	R	47K	0603_G	22.600	6.600	180.000	NO	
R18	R	1	MINIMELF	19.700	20.000	180.000	NO	
R19	R	2K5	0603	-23.500	18.400	0.000	NO	
R20	R	47K	0603	-23.500	14.400	180.000	NO	
R21	R	160	0603	-21.400	18.400	180.000	NO	
R22	R	XK	0603	1.400	4.000	270.000	YES	
T1	MTD20P	MTD20P	DPAK	4.000	24.000	180.000	NO	
T2	BC847	BC847	SOT23	0.400	36.300	0.000	NO	
T3	BC847	BC847	SOT23	3.900	36.300	180.000	NO	
T4	BC847	BC847	SOT23	-21.100	30.400	270.000	NO	
T5	BC856A	BC856A	SOT23	12.000	21.500	270.000	YES	

Component Report SALAMPATTE_B

salampatte_gloss Mon Sep 6 18:20:26 MET DST 2004

Ref Des	Device Type	Value	Package Type	x	y	ang	Mir	Remark
T6	BC846	BC846	SOT23	-20.100	14.500	0.000	NO	
TP1	TP		TPSMD	5.400	16.400	0.000	YES	
TP3	TP		TP5435	17.500	7.600	0.000	NO	
TP4	TP		TP5435	4.000	4.200	0.000	NO	
TP5	TP		TP5435	-0.200	18.500	0.000	NO	
TP6	TP		TP5435	-2.100	18.500	0.000	NO	
TP11	TP		TP4931	5.400	5.600	0.000	NO	
U1	LTC3200		SOT23_6	-19.800	22.000	180.000	NO	
U2	PIC16F876A		MLF28	9.000	10.400	180.000	YES	
U3	LTC1733EMSE		SO10_05MM	5.400	16.500	0.000	YES	
U4	LS7084S		SO8	17.000	27.400	90.000	YES	
U5	INA146UA		SO8	21.900	12.800	270.000	NO	
U6	MAX1719EUT		SOT23_6	21.600	12.600	0.000	YES	
U7	SI9986CY		SO8	14.500	28.800	0.000	NO	
U8	SI9986CY		SO8	14.500	22.800	0.000	NO	
U9	SI9986CY		SO8	14.500	34.800	0.000	NO	
V1	VIS_METAL		VIS2_5	-22.500	2.500	0.000	NO	
V2	VIS_METAL		VIS2_5	-22.500	37.500	0.000	NO	
V3	VIS_METAL		VIS2_5	22.500	37.500	0.000	NO	
V4	VIS_METAL		VIS2_5	22.500	2.500	0.000	NO	
V5	VIS_METAL		SALAMAXE4	-9.000	19.000	0.000	NO	
V7	VIS_METAL		VIS2_5	-22.500	-12.500	0.000	NO	
V8	VIS_METAL		VIS2_5	22.500	-12.500	0.000	NO	
V9	VIS_METAL		VIS2_5	22.500	-47.500	0.000	NO	
V10	VIS_METAL		VIS2_5	-22.500	-47.500	0.000	NO	
V11	VIS_METAL		VIS2_5	11.400	17.000	0.000	NO	
W1	JUMP2		JUMP2	4.000	2.000	90.000	NO	
W2	JUMP2		JUMP2	4.000	-48.000	90.000	NO	
XT1	XTALGND	20MHZ	NKS7	10.700	9.200	90.000	NO	

Total Component count 87

BOM Report SALAMPATTE_B

salampatte_gloss Mon Sep 6 18:20:27 MET DST 2004

Device	Package	Value	Nb	Reference Designators	Remark
0603-1-100N	0603_G	100N	9	C3 C6 C8 C12 C14 C16 C18 C20 C21	
0603-1-22P	0603_G	22P	2	C17 C19	
0805-1-1U	0805_G	1U	2	C1 C4	
0805-1-1U16V	0805_G	1U16V	2	C9 C11	
0805-1-2U2	0805_G	2U2	1	C5	
0805-1-4.7U16V	0805_G	4.7U16V	1	C22	
0805-1-4U7	0805_G	4U7	1	C7	
0805-1U16V	0805	1U16V	1	C10	
1105W90105WJ-1	LED3	X	1	D4	
BC846	SOT23	BC846	1	T6	
BC847	SOT23	BC847	3	T2 T3 T4	
BC856A	SOT23	BC856A	1	T5	
CO2-3	JUMP2		1	J2	
CO2-33	TOPLED		1	JL	
CO4-14	MM4		1	J4	
CO6-5	MM6		2	J1 J3	
CP-10V-SMDTPS-220UF-	CP_2917	220U	2	C13 C15	
CP-10VSMD100U-100U	CP_2917	100U	1	C2	
INA146UA	SO8		1	U5	
JUMP2-0	JUMP2		2	W1 W2	
LS7084S	SO8		1	U4	
LTC1733EMSE	SO10_05MM		1	U3	
LTC3200-5	SOT23_6		1	U1	
MAX1719EUT	SOT23_6		1	U6	
MTD20P	DPAK	MTD20P	1	T1	
PIC16F876A-VCC=PROGV	MLF28		1	U2	
R-17-1	MINIMELF	1	2	R15 R18	
R-19A-160	0603	160	1	R21	
R-19A-2K5	0603	2K5	1	R19	
R-19A-47K	0603	47K	1	R20	
R-19A-XK	0603	XK	1	R22	
R-23-100K	0603_G	100K	8	R1 R3 R4 R6 R7 R11 R12 R13	
R-23-10K	0603_G	10K	2	R2 R5	
R-23-1K	0603_G	1K	1	R14	
R-23-1M	0603_G	1M	1	R16	
R-23-2K5	0603_G	2K5	1	R9	
R-23-47K	0603_G	47K	3	R8 R10 R17	
SI9986CY	SO8		3	U7 U8 U9	
SM4001-SMD-SM4001	MELF_D	SM4001	1	D1	
SS24	1812_D	600696	2	D2 D3	
TP-19	TP4931		1	TP11	
TP-2	TP5435		4	TP3 TP4 TP5 TP6	
TP-5	TPSMD		1	TP1	
VIS_METAL-M2.5	VIS2_5		9	V1 V2 V3 V4 V7 V8 V9 V10 V11	
VIS_METAL-M4-SALAMAX	SALAMAXE4		1	V5	
XTALGND-0-20MHZ	NKS7	20MHZ	1	XT1	

Total Component count 87

NC Pins Report SALAMPATTE_B

salampatte_gloss Mon Sep 6 18:20:27 MET DST 2004

Ref Des	Device	Nb	Not Connected Pins	Remark
U2	PIC16F876A-VCC=PROGV	7	1 2 8 13 14 15 18	
U3	LTC1733EMSE	3	1 3 10	
V1	VIS_METAL-M2.5	1	1	
V2	VIS_METAL-M2.5	1	1	
V3	VIS_METAL-M2.5	1	1	
V4	VIS_METAL-M2.5	1	1	
V7	VIS_METAL-M2.5	1	1	
V8	VIS_METAL-M2.5	1	1	
V9	VIS_METAL-M2.5	1	1	
V10	VIS_METAL-M2.5	1	1	
V11	VIS_METAL-M2.5	1	1	

Total count 19

Power Pins Report SALAMPATTE_B

salampatte_gloss Mon Sep 6 18:20:28 MET DST 2004

Ref Des	Device	Name	Power Pins	Remark
U1	LTC3200-5	GND	2	
		VCCB	5	
U2	PIC16F876A-VCC=PROGV	GND	5 16	
U3	LTC1733EMSE	GND	4 5 6	
		VCCA	2 8	
		VCCB	9	
U4	LS7084S	GND	3	
		VCC	2	
U5	INA146UA	-5V	4	
		GND	1	
		VCC	7	
U6	MAX1719EUT	-5V	1	
		GND	4 5	
		VCC	2	
U7	SI9986CY	GND	1 2 4	
		VCCC	3	
U8	SI9986CY	GND	1 2 4	
		VCCC	3	
U9	SI9986CY	GND	1 2 4	
		VCCC	3	

Total count 9

Single Node Nets Report SALAMPATTE_B

salampatte_gloss Mon Sep 6 18:20:28 MET DST 2004

Netname	Node	Device	Remark

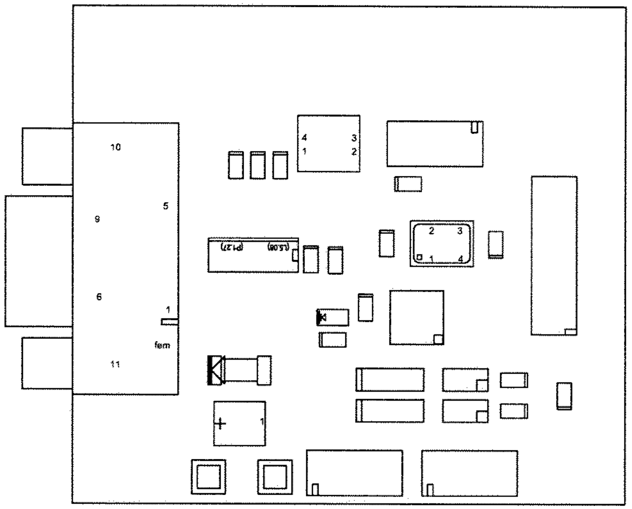
Total Nets count 0

E.4 Development board for PIC16f876A

I2CINTF

LSL A.Crespi
Phone: 36630
Thu Mar 17 11:17:54 MET 2005

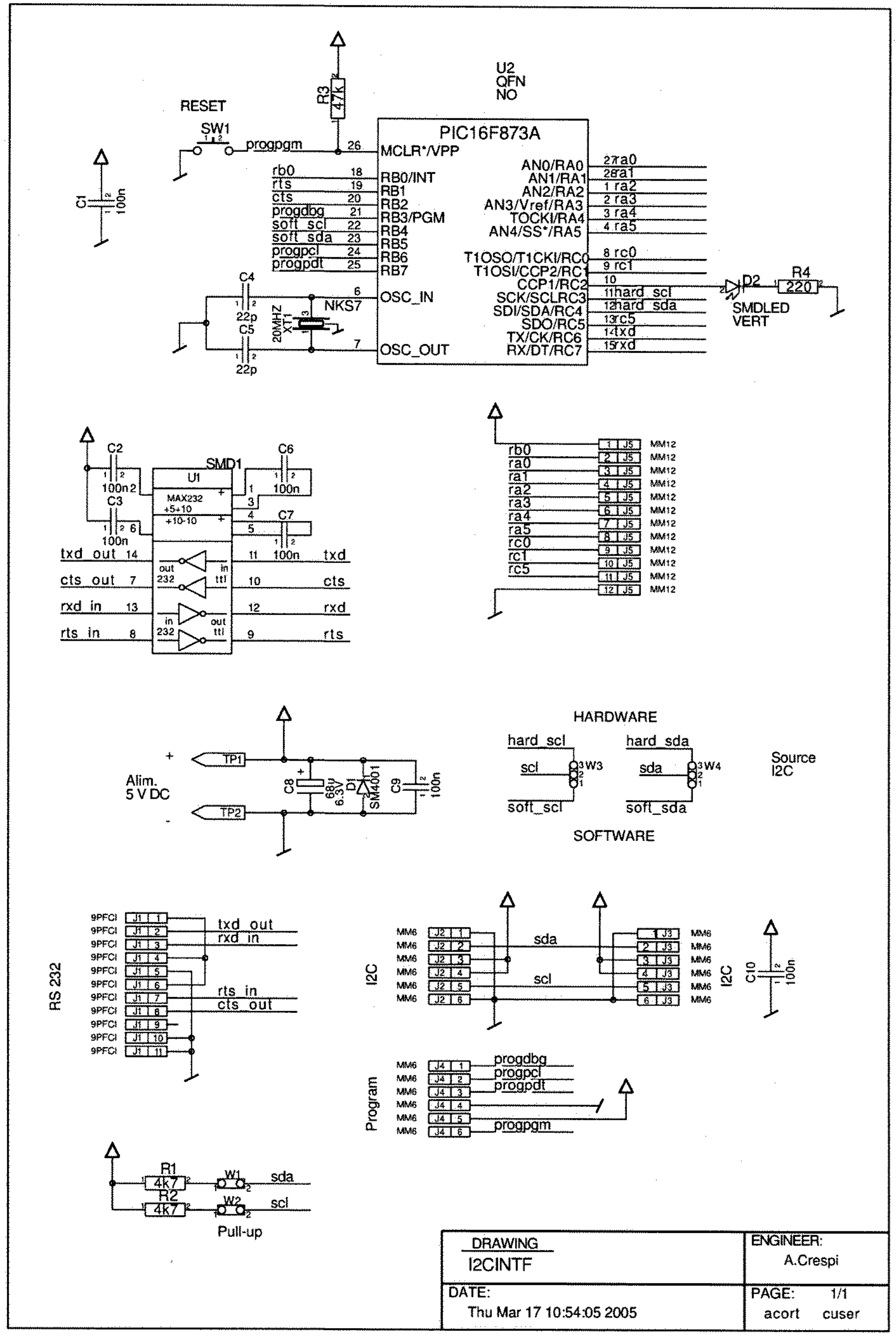
MFG_DATE:	21.03.2005
QUANTITY:	2
SOLDERMASK:	mtop mbot
MATERIAL:	FR4 : 1.6 mm



CONTENTS	PAGE
Root Schema i2cintf	2
Xref i2cintf	3
Drill	4
Etch	5
Outline_Top	6
Gerber_mtop	7
Gerber_top	8
Gerber_bot	9
Gerber_mbot	10
Gerber_cont	11
Component Report	12
BOM Report	13
NC-Pins Report	14
Power-Pins Report	15

CONTENTS	PAGE
Single Node Nets	16

ACORT



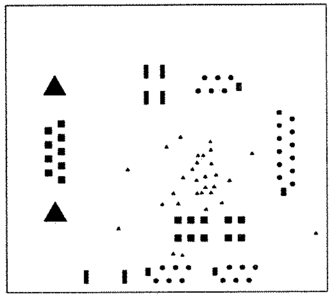
Signal Page Ref I2CINTF

i2cintf

Thu Mar 17 11:17:55 MET 2005

Ref Des	Pages
cts	
cts_out	
hard_scl	
hard_sda	
progdbg	
progpcl	
progpdt	
progpgm	
ra0	
ra1	
ra2	
ra3	
ra4	
ra5	
rb0	
rc0	
rc1	
rc5	
rts	
rts_in	
rxd	
rxd_in	
scl	
sda	
soft_scl	
soft_sda	
txd	
txd_out	

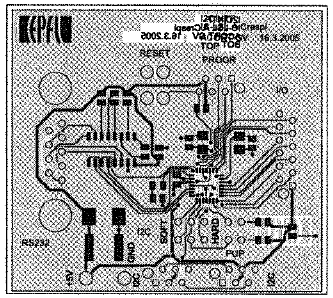
Total Signals count 28



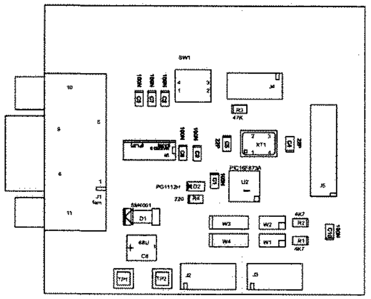
DRILL CHART			
FIGURE	SIZE	PLATED	QTY
·	0.203	PLATED	30
●	0.787	PLATED	30
■	0.889	PLATED	19
■	0.94	PLATED	2
■	1.118	PLATED	4
■	1.499	PLATED	4
▲	3.175	PLATED	2

TITRE:I2CINTF	LABORATOIRE:IC-LSL	DATE:16.3.2005
	ENGINEER:A.Crespi	EPFL-ACORT-GV

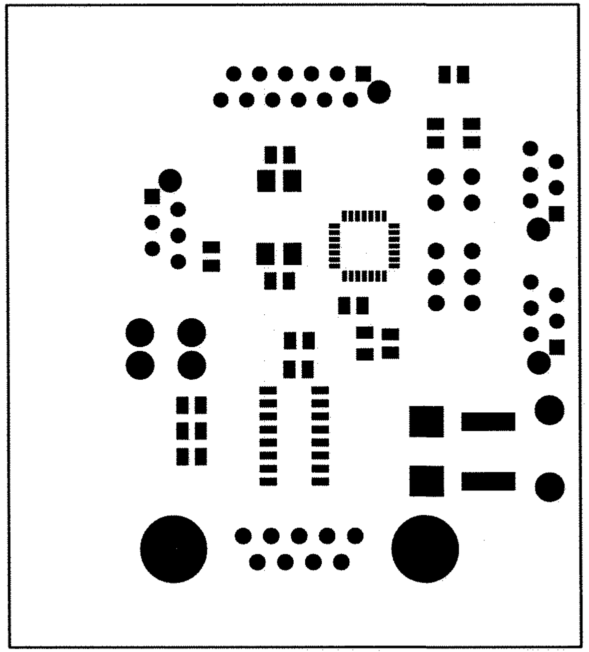
EPFL ACORT Thu 17.03.2005 11:16 Drawing name = I2CINTF Scale = 1.00 1.000 Inch =



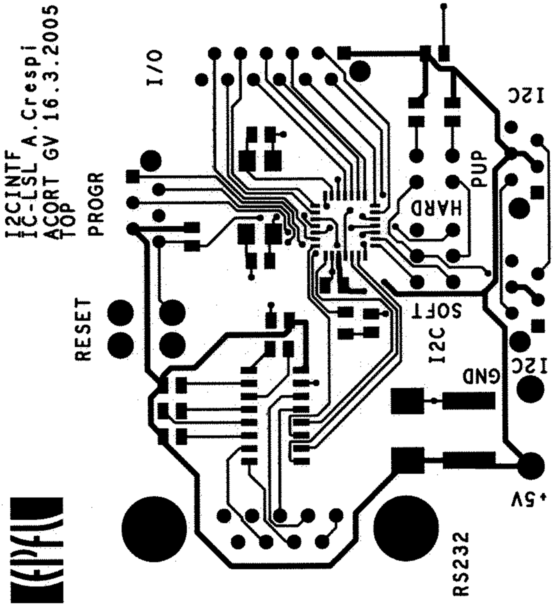
TITRE:I2CINTF	LABORATOIRE:IC-LSL	DATE:16.3.2005
	ENGINEER:A.Crespi	EPFL-ACORT-GV



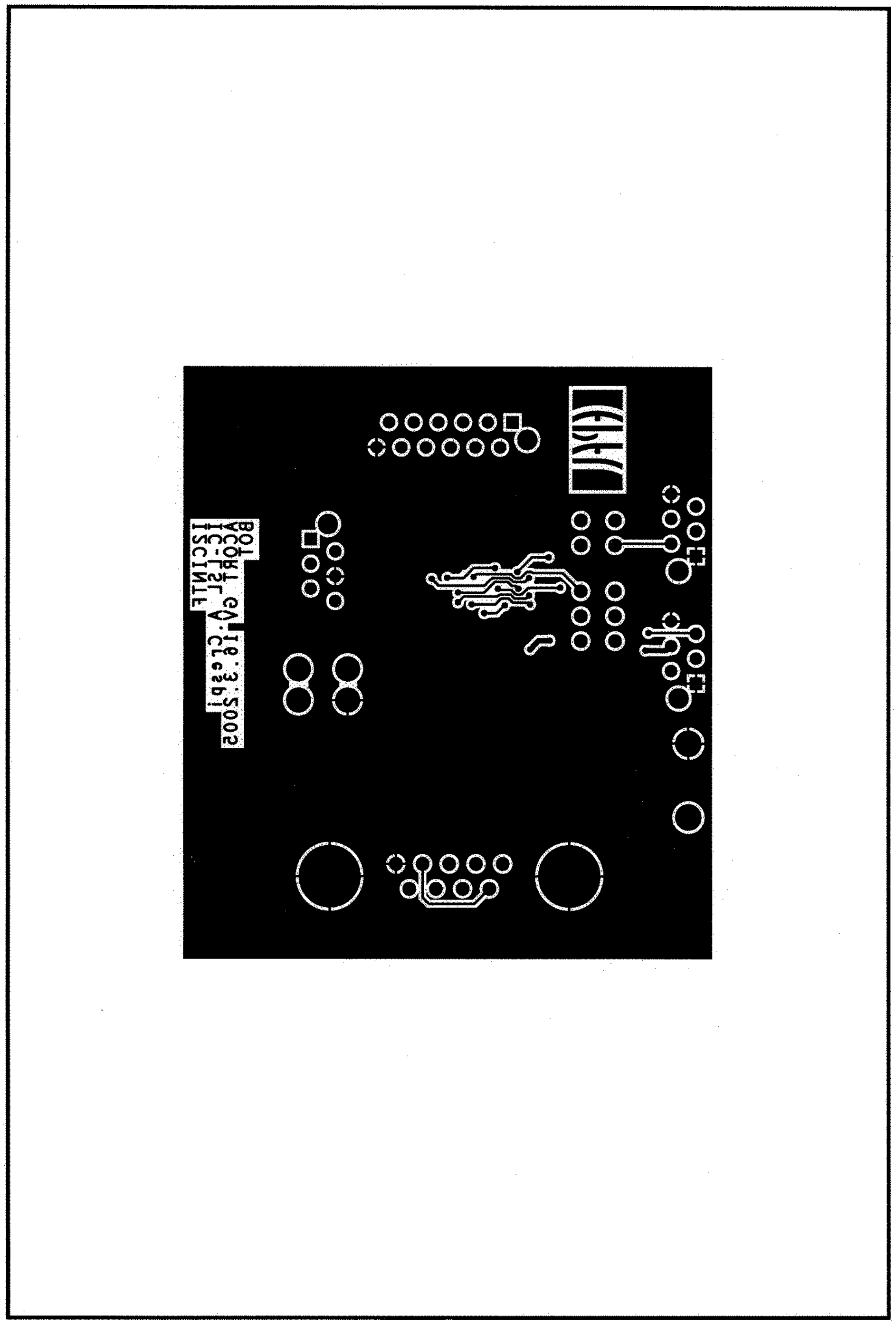
TITRE:I2CINTF	LABORATOIRE:IC-LSL	DATE:16.3.2005
	ENGINEER:A.Crespi	EPFL-ACORT-GV



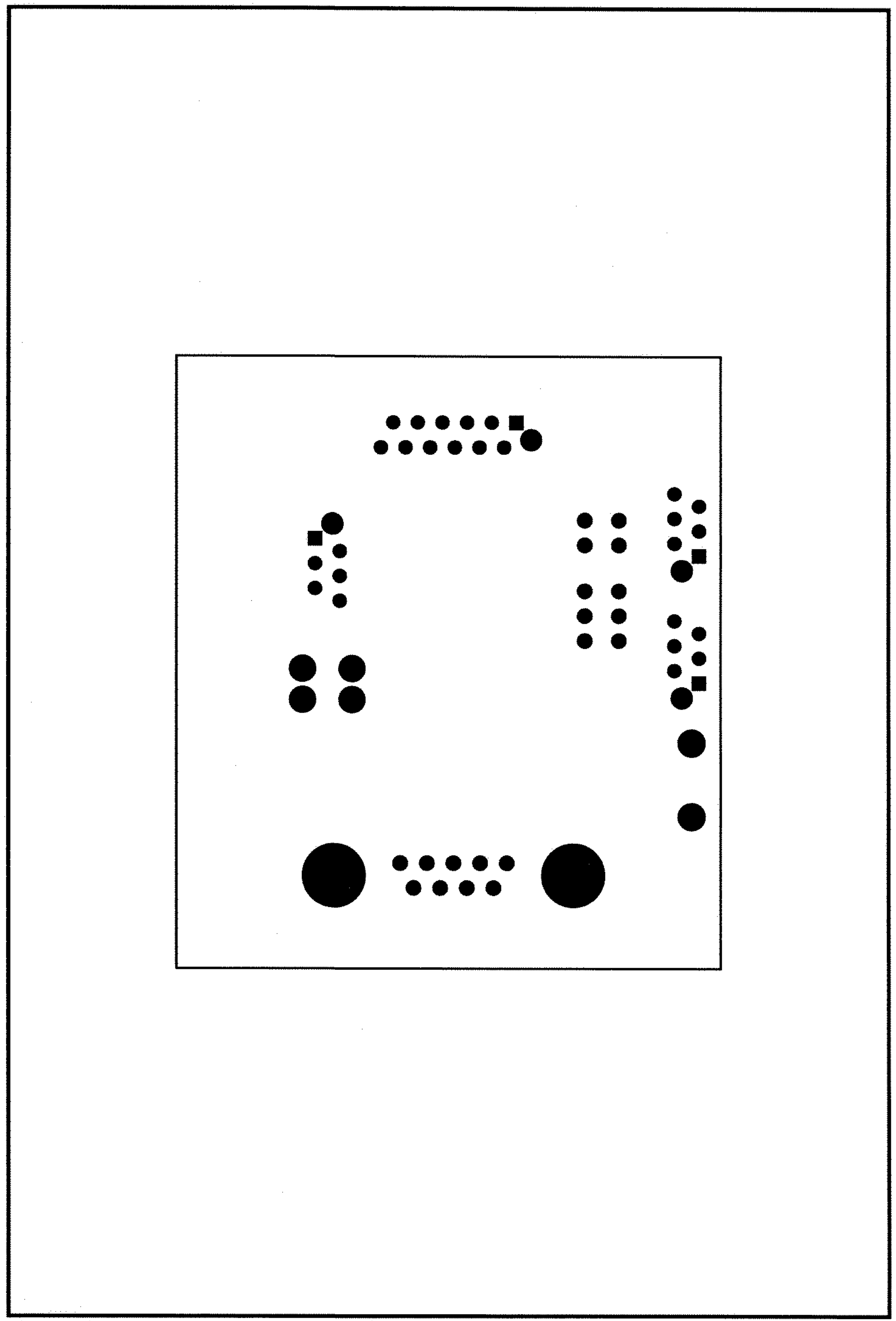
EPFL ACORT Thu Mar 17 11:16:48 2005 Project = i2cintf File = i2cintf_gloss,mtop Scale = 2 0.5 cm =



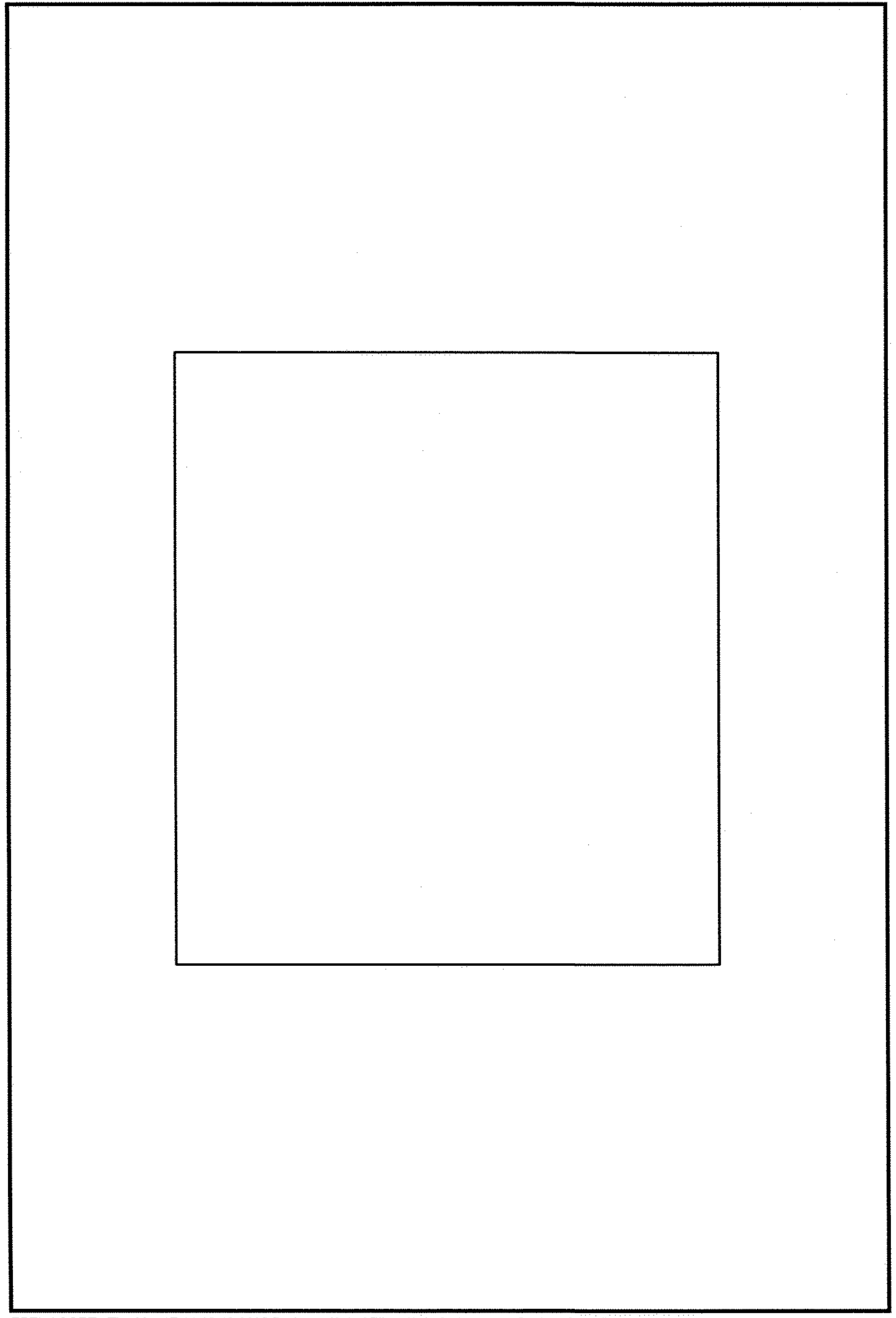
EPFL ACORT Thu Mar 17 11:16:48 2005 Project = i2cintf File = i2cintf_gloss,top Scale = 2 0.5 cm =



EPFL ACORT Thu Mar 17 11:16:48 2005 Project = i2cintf File = i2cintf_gloss,bot Scale = 2 0.5 cm =



EPFL ACORT Thu Mar 17 11:16:48 2005 Project = i2cintf File = i2cintf_gloss,mbot Scale = 2 0.5 cm =



EPFL ACORT Thu Mar 17 11:16:48 2005 Project = i2cintf File = i2cintf_gloss,cont Scale = 2 0.5 cm =

Component Report I2CINTF

i2cintf_gloss Thu Mar 17 11:17:56 MET 2005

Ref Des	Device Type	Value	Package Type	x	y	ang	Mir	Remark
C1	0805	100N	0805_G	-20.800	-5.900	0.000	NO	
C2	0805	100N	0805_G	-30.500	10.000	0.000	NO	
C3	0805	100N	0805_G	-35.500	10.000	0.000	NO	
C4	0805	22P	0805_G	-6.100	1.200	180.000	NO	
C5	0805	22P	0805_G	-18.400	1.300	0.000	NO	
C6	0805	100N	0805_G	-27.000	-0.500	0.000	NO	
C7	0805	100N	0805_G	-33.000	10.000	0.000	NO	
C8	CP	68U	CP_2220	-35.000	-19.000	270.000	NO	
C9	0805	100N	0805_G	-24.200	-0.600	0.000	NO	
C10	0805	100N	0805_G	1.700	-15.800	180.000	NO	
D1	SM4001	SM4001	MELF_D	-35.000	-13.000	90.000	NO	
D2	PG1112H	PG1112H	SMDLED	-24.488	-7.000	90.000	NO	
J1	CO11		9PFCI	-43.238	-0.500	90.000	NO	
J2	CO6		MM6	-24.858	-25.770	0.000	NO	
J3	CO6		MM6	-11.858	-25.770	0.000	NO	
J4	CO6		MM6	-10.143	-13.770	180.000	NO	
J5	CO12		MM12	1.770	-6.985	90.000	NO	
R1	R	4K7	0805_G	-4.000	-17.500	270.000	NO	
R2	R	4K7	0805_G	-4.000	-14.000	270.000	NO	
R3	R	47K	0805_G	-16.000	8.000	90.000	NO	
R4	R	220	0805_G	-24.500	-9.500	90.000	NO	
SW1	SWPOU		POUMINI	-25.013	12.500	0.000	NO	
TP1	TP		TPLOGI	-38.500	-25.000	0.000	NO	
TP2	TP		TPLOGI	-31.000	-25.000	0.000	NO	
U1	MAX232		SO16	-33.500	-0.063	270.000	NO	
U2	PIC16F873A		MLF28	-15.000	-7.000	180.000	NO	
W1	JUMP2		JUMP2	-8.230	-17.500	90.000	NO	
W2	JUMP2		JUMP2	-8.230	-14.000	90.000	NO	
W3	JUMP3		JUMP3	-20.540	-16.540	0.000	NO	
W4	JUMP3		JUMP3	-20.540	-20.040	0.000	NO	
XT1	XTALGND	20MHZ	NKS7	-12.200	1.300	90.000	NO	

Total Component count 31

BOM Report I2CINTF

i2cintf_gloss Thu Mar 17 11:17:56 MET 2005

Device	Package	Value	Nb	Reference Designators	Remark
0805-1-100N	0805_G	100N	7	C1 C2 C3 C6 C7 C9 C10	
0805-1-22P	0805_G	22P	2	C4 C5	
CO11-2	9PFCI		1	J1	
CO12-15	MM12		1	J5	
CO6-5	MM6		3	J2 J3 J4	
CP-6.3VSMD-68U-68U	CP_2220	68U	1	C8	
JUMP2-0	JUMP2		2	W1 W2	
JUMP3-1	JUMP3		2	W3 W4	
MAX232-3	SO16		1	U1	
PG1112H	SMDLED	PG1112H	1	D2	
PIC16F873A	MLF28		1	U2	
R-20-220	0805_G	220	1	R4	
R-20-47K	0805_G	47K	1	R3	
R-20-4K7	0805_G	4K7	2	R1 R2	
SM4001-SMD-SM4001	MELF_D	SM4001	1	D1	
SWPOU-0	POUMINI		1	SW1	
TP-4	TPLOGI		2	TP1 TP2	
XTALGND-0-20MHZ	NKS7	20MHZ	1	XT1	

Total Component count 31

NC Pins Report I2CINTF

i2cintf_gloss Thu Mar 17 11:17:57 MET 2005

Ref Des	Device	Nb	Not Connected Pins	Remark
J1	CO11-2	1	9	

Total count 1

Power Pins Report I2CINTF

i2cintf_gloss Thu Mar 17 11:17:57 MET 2005

Ref Des	Device	Name	Power Pins	Remark
U1	MAX232-3	GND	15	
		VCC	16	
U2	PIC16F873A	GND	5 16	
		VCC	17	

Total count 2

Single Node Nets Report I2CINTF

i2cintf_gloss Thu Mar 17 11:17:58 MET 2005

Netname	Node	Device	Remark

Total Nets count 0

Appendix F

motor control register summary [15]

APPENDIX A: REGISTER SUMMARY

The PID controller uses 56 registers for parameter input and output. The following table resumes all these registers. For more information read the corresponding chapters.

Name	Address	Function	Min	Max
Mode	0x28	Selection of mode 0 = Idle Mode 1 = Normal Control Mode 2 = Stop Motor 3 = Sleep Mode 4 = Reset Mode 5 = Save configuration to EEPROM [1]	0	5
SetPointSource	0x29	Source of SetPoint 0 = External I2C 1 = External analogical input 2 = Internal Square Wave Generator 3 = Internal Triangle Generator 4 = Internal Sinus Generator	0	4
HWOptions	0x2A	Hardware options Bit0: Start-up mode, 0=Idle mode / 1=Normal control Bit1: Analogue SetPoint Input, 0=disabled / 1=enabled Bit2: LED, 0=disabled / 1=enabled Bit3: Encoder resolution, 0=100% / 1=25% Bit4: Torque inversion, 0=disabled / 1=enabled Bit5: Driver Option 1 Bit6: Driver Option 2 Bit7: Driver Option 3	Flags	
SWOptions	0x2B	Software options Bit0: Separate D, 0=disabled / 1=enabled Bit1: Antireset Windup, 0=disabled / 1=enabled Bit2: SoftStopMin, 0=disabled / 1=enabled Bit3: SoftStopMax, 0=disabled / 1=enabled Bit4: Error on SoftStop, 0=disabled / 1=enabled Bit5: Stop motor if blocked, 0=disabled / 1=enabled Bit6: SoftwareCurrentControl, 0=disabled / 1=enabled Bit7: Direction inversion, 0=disabled / 1=enabled	Flags	
ControlTyp	0x2C	Type of PID Control 0 = Open Loop 1 = Position control with no speed Profile 2 = Position control with Trapezoidal Speed Profile 3 = Speed control with no speed profile 4 = Speed control with Trapezoidal Speed Profile 5 = Torque control 6 = Zero Friction control	0	6

Table continues next page

[1] A special security procedure has to be followed in order to prevent erroneous overwriting of the module configuration. The values 0x55, 0xAA and 0x05 have to be written one after the other to the *Mode* register.

ErrorFlags	0x2D	Flags indicating an error (read only!) Bit0: Sample time to small Bit1: Watchdog timer overflow Bit2: Brown-out Bit3: SoftStop happened (only if SoftStop enabled) Bit4: Motor blocked (only if MotorBlocked enabled) Bit5: Position out of range (Overflow), 0=No / 1=Yes Bit6: Speed out of range (Overflow), 0=No / 1=Yes Bit7: Torque out of range (Overflow), 0=No / 1=Yes	Flags	
StatusFlags	0x2E	Flags indicating the status of the controller (read only!) Bit0: Movement detected, 0=No / 1=Yes Bit1: Direction of movement, 0=negative / 1=positive Bit2: On SetPoint, 0=No / 1=Yes Bit3: Near SetPoint (+/-5units), 0=No / 1=Yes Bit4: Saturation of driver command, 0=No / 1=Yes Bit5: Antireset Windup overflow, 0=No / 1=Yes Bit6: SoftwareCurrentControl active, 0=No / 1=Yes Bit7: SoftStop active, 0=No / 1=Yes	Flags	
SetPoint	0x2F	SetPoint	-127	127
Position	0x30	Measured position, 8bit signed integer (read and write)	-127	127
Speed	0x31	Measured Speed, 8bit signed integer (read only)	-127	127
Torque	0x32	Measured Torque, 8bit signed integer (read only)	-127	127
DividerPos	0x33	Divider for the position, choose 8bit out of 32bit	0	24
PositionHH PositionHL PositionLH PositionLL	0x34 0x35 0x36 0x37	These 4 variables contain a copy of the 32bit position Always first read the PositionHH register!	0	2^{32}-1
DividerSpeed	0x38	Divider for the speed, choose 8bit out of 16bit	0	8
SpeedH SpeedL	0x39 0x3A	These 2 variables contain a copy of the 16bit Speed Always first read the SpeedH register!	0	2^{16}-1
DividerTorque	0x3B	Divider for the torque, choose 8bit out of 16bit	0	8
TorqueH TorqueL	0x3C 0x3D	These 2 variables contain a copy of the 16bit Torque Always first read the TorqueH register!	0	2^{16}-1
TorqueBiasH TorqueBiasL	0x3E 0x3F	These 2 variables contain the 16bit Bias of the Torque measurement	0	2^{16}-1
KpPos	0x40	Kp of PID Position	0	255
KpDivPos	0x41	Divider for KpPos, Kp=KpPos/2^KpDivPos	0	32
KdPos	0x42	Kd of PID Position	0	255
KdDivPos	0x43	Divider for KdPos, Kd=KdPos/2^KdDivPos	0	32
FilterPos	0x44	Length of 1st order filter of differential part	0	255
KpSpeed	0x45	Kp of PID Speed	0	255
KpDivSpeed	0x46	Divider for KpSpeed, Kp=KpSpeed/2^KpDivSpeed	0	32
KiSpeed	0x47	Ki of PID Speed	0	255
KiDivSpeed	0x48	Divider for KiSpeed, Ki=KiSpeed/2^KiDivSpeed	0	32
KpTorque	0x49	Kp of PID Torque	0	255
KpDivTorque	0x4A	Divider for KpTorque, Kp=KpTorque/2^KpDivTorque	0	32
KiTorque	0x4B	Ki of PID Torque	0	255
KiDivTorque	0x4C	Divider for KiTorque, Ki=KiTorque/2^KiDivTorque	0	32
SampleTimeH SampleTimeL	0x4D 0x4E	Sampling time, 16bit integer h=SampleTime*1.6us	0	2^{16}-1
BlockedTime	0x49	Time to wait before the motor is considered blocked	0	255

Table continues next page

IntGenPeriod	0x50	Period of function generator, T=IntGenPeriod*256*h	0	255
IntGenAmplitude	0x51	Amplitude of internal function generator	0	255
IntGenOffset	0x52	Offset of internal function generator	-127	127
SoftStopMin	0x53	Position of SoftStop minimum	-127	127
SoftStopMax	0x54	Position of SoftStop maximum	-127	127
Acceleration	0x55	Acceleration for trapezoidal speed profile	0	255
MaxSpeed	0x56	Maximal Speed for trapezoidal speed profile	0	127
StaticFriction	0x57	Friction of the system for zero friction control	0	127
HWCurrentLimit	0x58	HardwareCurrentLimit, U=5V/255*HWCurrentLimit	0	255
SWCurrentLimit	0x59	SoftwareCurrentLimit (limits measured torque)	0	127
PID_E_8Bit	0x5A	Delta8Bit=SetPoint-Process variable (read only!)	-127	127
PID_P_8Bit	0x5B	Proportional part of PID controller (read only!)	-127	127
PID_I_8Bit	0x5C	Integral part of PID controller (read only!)	-127	127
PID_D_8Bit	0x5D	Differential part of PID controller (read only!)	-127	127
PID_OUT_8Bit	0x5E	Output of PID Controller (read only!)	-127	127

Table 1: Register summary.