# Project of monitoring the wind tunnel of the ETSEIAT's Aerospace Engineering laboratory (Software)

Annex

**Author**
Oriol Villanueva Pujol

**Director**
David González Diez

**Bachelor's Degree in Aerospace Technologies Engineering Thesis**

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeries
Industrial i Aeronàutica de Terrassa

**Universitat Politècnica de Catalunya**
**Terrassa School of Industrial and Aeronautical Engineering**

**June 2015**

# Contents

1. Summary ........ 1
A. Programming code........ 2
Aidoes.pde ........ 2
Genuine code ........ 14
B. Calibration documents ........ 27
C. Techincal documents and datasheets ........ 32
D. Further information ........ 111
Biblioraphy ........ 123

# List of tables

Table B.1. NTC verification experiment ..................................................... 27
Table B.2. Lambrecht barometer readings and corrections ........................ 27
Table B.3. Lambrecht readings vs sensor readings .................................... 27
Table B.4. Differential pressure verification with U-Tube ............................ 28
Table B.5 Differential pressure verification with alcohol manometer ..... 28
Table B.6 Speed verification ................................................................... 29
Table B.7 Differential pressure sensor calibration .................................. 29
Table B.8 Atmospheric pressure sensor calibration ................................ 29
Table B.9 Alpha constants obtention data .............................................. 30
Table B.10 Uncertainties: Data used ....................................................... 30
Table B.11 Uncertainties: Calculus at different air properties.................. 31
Table D.1 Arduino main characteristics .................................................112
Table D.2 Arduino Due summary of characteristics ............................. 117

# List of figures

Figure D.1. Arduino UNO R3 version board Front and Back views....... 111
Figure D.2. Arduino DUE board Front and Back views ............................ 116
Figure D.3 Arduino DUE, USB ports......................................................... 120
Figure D.4. Wireless shield with XBee module .......................................... 121

# 1. Summary

When writtig such an extensive and crafted thesis like this the author usually needs a lot of interesting information like datasheets or code created to make the project success. Since they all the documents can not go in th report an Annex should be created.

The Annex is going to be divided in several parts, as can be seen in the list of contents (I), according to technical criteria.

First of all the code developed in the whole project is going to be presented. Afterwards the documents used for calibraton and error estimation.will be placed. In the end the technical documentation like data sheets or operation manuals will be included.

Moreover an extra zip file will be included with the used excel files and the code.

# A. Programming code

## Aidoes.pde

This is the code that runs always in the Arduino while the Matlab code is executing the orders.

```
/* Analog and Digital Input and Output Server for MATLAB      */
/* Giampiero Campa, Copyright 2012 The MathWorks, Inc         */
/* This file is meant to be used with the MATLAB arduino IO
   package, however, it can be used from the IDE environment
   (or any other serial terminal) by typing commands like:

   0e0 : assigns digital pin #4 (e) as input
   0f1 : assigns digital pin #5 (f) as output
   0n1 : assigns digital pin #13 (n) as output
   1c  : reads digital pin #2 (c)
   1e  : reads digital pin #4 (e)
   2n0 : sets digital pin #13 (n) low
   2n1 : sets digital pin #13 (n) high
   2f1 : sets digital pin #5 (f) high
   2f0 : sets digital pin #5 (f) low
   4j2 : sets digital pin #9 (j) to  50=ascii(2) over 255
   4jz : sets digital pin #9 (j) to 122=ascii(z) over 255
   3a  : reads analog pin #0 (a)
   3f  : reads analog pin #5 (f)
   5j    : reads status (attached/detached) of servo on pin #9
   5k    : reads status (attached/detached) of servo on pin #10
   6j1   : attaches servo on pin #9
   8jz   : moves servo on pin #9 of 122 degrees (122=ascii(z))
   7j    : reads angle of servo on pin #9
   6j0   : detaches servo on pin #9

   E0cd  : attaches encoder #0 (0) on pins 2 (c) and 3 (d)
   E1st  : attaches encoder #1 on pins 18 (s) and 19 (t)
   E2vu  : attaches encoder #2 on pins 21 (v) and 20 (u)
   G0    : gets 0 position of encoder #0
   I0u   : sets debounce delay to 20 (2ms) for encoder #0
   H1    : resets position of encoder #1
   F2    : detaches encoder #2
   R0    : sets analog reference to DEFAULT
   R1    : sets analog reference to INTERNAL
   R2    : sets analog reference to EXTERNAL
   X3    : roundtrip example case returning the input (ascii(3))
   99    : returns script type (0 adio.pde ... 3 motor.pde ) */
```

```
#include ¡Servo.h¿

/* define internal for the MEGA as 1.1V (as as for the 328)  */
#if defined(¨AVR˙ATmega1280¨) —— defined(¨AVR˙ATmega2560¨)
#define INTERNAL INTERNAL1V1
#endif
/* define encoder structure                              */
typedef struct – int pinA; int pinB; int pos; int del;˝ Encoder;
volatile Encoder Enc[3] = ––0,0,0,0˝, –0,0,0,0˝, –0,0,0,0˝˝;
/* create servo vector                                    */
Servo servo[70];
void setup() –
  /* initialize serial                                    */
  Serial.begin(115200);
˝
void loop() –

  /* variables declaration and initialization                 */

  static int  s    = -1;     /* state                       */
  static int  pin = 13;     /* generic pin number            */
  static int  enc = 0;      /* generic encoder number        */
  int  val =  0;            /* generic value read from serial */
  int  agv =  0;            /* generic analog value          */
  int  dgv =  0;            /* generic digital value         */

  /* The following instruction constantly checks if anything
     is available on the serial port. Nothing gets executed in
     the loop if nothing is available to be read, but as soon
     as anything becomes available, then the part coded after
     the if statement (that is the real stuff) gets executed */
  if (Serial.available() ¿0) –
    /* whatever is available from the serial is read here     */
    val = Serial.read();

    /* This part basically implements a state machine that
       reads the serial port and makes just one transition
       to a new state, depending on both the previous state
       and the command that is read from the serial port.
       Some commands need additional inputs from the serial
       port, so they need 2 or 3 state transitions (each one
       happening as soon as anything new is available from
       the serial port) to be fully executed. After a command
       is fully executed the state returns to its initial
       value s=-1                                         */
    switch (s)
```

```
/* s=-1 means NOTHING RECEIVED YET ****************** */
case -1:
/* calculate next state                          */
if (val¿47 && val¡90) –
    /* the first received value indicates the mode
   49 is ascii for 1, ... 90 is ascii for Z
   s=0 is change-pin mode;
   s=10 is DI;  s=20 is DO;  s=30 is AI;  s=40 is AO;
   s=50 is servo status; s=60 is aervo attach/detach;
   s=70 is servo read;   s=80 is servo write;
   s=90 is query script type (1 basic, 2 motor);
   s=210 is encoder attach; s=220 is encoder detach;
   s=230 is get encoder position; s=240 is encoder reset;
   s=250 is set encoder debounce delay;
   s=340 is change analog reference;
   s=400 example echo returning the input argument;
                                      */
 s=10*(val-48);
 /* the following statements are needed to handle
  unexpected first values coming from the serial (if
  the value is unrecognized then it defaults to s=-1) */
if ((s¿90 && s¡210) —— (s¿250 && s!=340 && s!=400)) –
 s=-1;
/* the break statements gets out of the switch-case, so
/* we go back and wait for new serial data          */
break; /* s=-1 (initial state) taken care of          */
/* s=0 or 1 means CHANGE PIN MODE                       */
   case 0:
/* the second received value indicates the pin
  from abs('c')=99, pin 2, to abs('Π')=166, pin 69    */
if (val¿98 && val¡167) –
 pin=val-97;               /* calculate pin          */
 s=1; /* next we will need to get 0 or 1 from serial  */
    else –
 s=-1; /* if value is not a pin then return to -1      */
   break; /* s=0 taken care of                          */
case 1:
/* the third received value indicates the value 0 or 1 */
if (val¿47 && val¡50) –
 /* set pin mode                               */
 if (val==48) –
  pinMode(pin,INPUT)
 else –
  pinMode(pin,OUTPUT);
s=-1;  /* we are done with CHANGE PIN so go to -1       */
break; /* s=1 taken care of                          */
```

```
/* s=10 means DIGITAL INPUT ************************** */
case 10:
/* the second received value indicates the pin
   from abs('c')=99, pin 2, to abs('Π')=166, pin 69    */
if (val¿98 && val¡167) –
  pin=val-97;              /* calculate pin          */
  dgv=digitalRead(pin);    /* perform Digital Input  */
  Serial.println(dgv);     /* send value via serial  */
s=-1;  /* we are done with DI so next state is -1      */
break; /* s=10 taken care of                         */
/* s=20 or 21 means DIGITAL OUTPUT ******************* */
case 20:
/* the second received value indicates the pin
   from abs('c')=99, pin 2, to abs('Π')=166, pin 69    */
if (val¿98 && val¡167) –
  pin=val-97;              /* calculate pin          */
  s=21; /* next we will need to get 0 or 1 from serial */
else –
  s=-1; /* if value is not a pin then return to -1     */
break; /* s=20 taken care of                         */
case 21:
/* the third received value indicates the value 0 or 1 */
if (val¿47 && val¡50) –
  dgv=val-48;              /* calculate value        */
    digitalWrite(pin,dgv);   /* perform Digital Output */
s=-1;  /* we are done with DO so next state is -1      */
break; /* s=21 taken care of                         */
/* s=30 means ANALOG INPUT *************************** */

case 30:
/* the second received value indicates the pin
   from abs('a')=97, pin 0, to abs('p')=112, pin 15    */
if (val¿96 && val¡113) –
  pin=val-97;              /* calculate pin          */
  agv=analogRead(pin);     /* perform Analog Input   */
    Serial.println(agv);     /* send value via serial  */
s=-1;  /* we are done with AI so next state is -1      */
break; /* s=30 taken care of                         */
/* s=40 or 41 means ANALOG OUTPUT ******************** */
case 40:
/* the second received value indicates the pin
   from abs('c')=99, pin 2, to abs('Π')=166, pin 69    */
if (val¿98 && val¡167) –
  pin=val-97;              /* calculate pin          */
  s=41; /* next we will need to get value from serial  */
else –
  s=-1; /* if value is not a pin then return to -1     */
```

```
break; /* s=40 taken care of                    */
case 41:
/* the third received value indicates the analog value */
analogWrite(pin,val);      /* perform Analog Output  */
s=-1;  /* we are done with AO so next state is -1     */
break; /* s=41 taken care of                    */
/* s=50 means SERVO STATUS (ATTACHED/DETACHED) ******* */
case 50:
/* the second value indicates the servo attachment pin
   from abs('c')=99, pin 2, to abs('Π')=166, pin 69    */
if (val¿98 && val¡167) –
  pin=val-97;              /* calculate pin         */
  dgv=servo[pin].attached();         /* read status */
  Serial.println(dgv);      /* send value via serial  */
s=-1;  /* we are done with servo status so return to -1*/
break; /* s=50 taken care of                    */
/* s=60 or 61 means SERVO ATTACH/DETACH ************** */
case 60:
/* the second value indicates the servo attachment pin
   from abs('c')=99, pin 2, to abs('Π')=166, pin 69    */
if (val¿98 && val¡167) –
  pin=val-97;              /* calculate pin         */
  s=61; /* next we will need to get 0 or 1 from serial */
else –
  s=-1; /* if value is not a servo then return to -1   */
break; /* s=60 taken care of                    */
case 61:
/* the third received value indicates the value 0 or 1
   0 for detach and 1 for attach                */
if (val¿47 && val¡50) –
  dgv=val-48;              /* calculate value       */
  if (dgv) servo[pin].attach(pin);    /* attach servo */
  else servo[pin].detach();          /* detach servo */
s=-1;  /* we are done with servo attach/detach so -1   */
break; /* s=61 taken care of                    */
/* s=70 means SERVO READ **************************** */
case 70:
/* the second value indicates the servo attachment pin
   from abs('c')=99, pin 2, to abs('Π')=166, pin 69    */
if (val¿98 && val¡167) –
  pin=val-97;              /* calculate pin         */
  agv=servo[pin].read();     /* read value           */
    Serial.println(agv);      /* send value via serial  */
s=-1;  /* we are done with servo read so go to -1 next */
break; /* s=70 taken care of                    */
```

```
/* s=80 or 81 means SERVO WRITE   ******************** */
case 80:
/* the second value indicates the servo attachment pin
   from abs('c')=99, pin 2, to abs('Π')=166, pin 69    */
if (val¿98 && val¡167) –
  pin=val-97;               /* calculate pin          */
  s=81; /* next we will need to get value from serial  */
else –
  s=-1; /* if value is not a servo then return to -1   */
break; /* s=80 taken care of                          */
case 81:
/* the third received value indicates the servo angle  */
servo[pin].write(val);                /* write value */
s=-1;  /* we are done with servo write so go to -1 next*/
break; /* s=81 taken care of                          */
/* s=90 means Query Script Type:
   (0 adio, 1 adioenc, 2 adiosrv, 3 motor)            */
case 90:
if (val==57) –
  /* if string sent is 99  send script type via serial */
  Serial.println(2);
s=-1;  /* we are done with this so next state is -1    */
break; /* s=90 taken care of                          */
/* s=210 to 212 means ENCODER ATTACH ***************** */
case 210:
/* the second value indicates the encoder number:
   either 0, 1 or 2                                */
if (val¿47 && val¡51) –
  enc=val-48;         /* calculate encoder number      */
  s=211;  /* next we need the first attachment pin     */
else –
  s=-1; /* if value is not an encoder then return to -1*/
break; /* s=210 taken care of                         */
case 211:
/* the third received value indicates the first pin
   from abs('c')=99, pin 2, to abs('Π')=166, pin 69    */
if (val¿98 && val¡167) –
  pin=val-97;               /* calculate pin          */
  Enc[enc].pinA=pin;          /* set pin A              */
  s=212;  /* next we need the second attachment pin    */
else –
  s=-1; /* if value is not a servo then return to -1   */
break; /* s=211 taken care of                         */
case 212:
/* the fourth received value indicates the second pin
```

```
   from abs('c')=99, pin 2, to abs('Π')=166, pin 69     */
if (val¿98 && val¡167) –
  pin=val-97;                /* calculate pin          */
  Enc[enc].pinB=pin;         /* set pin B              */
  /* set encoder pins as inputs                       */
  pinMode(Enc[enc].pinA, INPUT);
  pinMode(Enc[enc].pinB, INPUT);
  /* turn on pullup resistors                         */
  digitalWrite(Enc[enc].pinA, HIGH);
  digitalWrite(Enc[enc].pinB, HIGH);
  /* attach interrupts                                */
  switch(enc) –
    case 0:
      attachInterrupt(getIntNum(Enc[0].pinA), isrPinAEn0, CHANGE);
      attachInterrupt(getIntNum(Enc[0].pinB), isrPinBEn0, CHANGE);
      break;
    case 1:
      attachInterrupt(getIntNum(Enc[1].pinA), isrPinAEn1, CHANGE);
      attachInterrupt(getIntNum(Enc[1].pinB), isrPinBEn1, CHANGE);
      break;
    case 2:
      attachInterrupt(getIntNum(Enc[2].pinA), isrPinAEn2, CHANGE);
      attachInterrupt(getIntNum(Enc[2].pinB), isrPinBEn2, CHANGE);
      break;
s=-1; /* we are done with encoder attach so -1          */
break; /* s=212 taken care of                           */
/* s=220 means ENCODER DETACH   ********************** */
case 220:
/* the second value indicates the encoder number:
   either 0, 1 or 2                                   */
if (val¿47 && val¡51) –
  enc=val-48;        /* calculate encoder number       */
  /* detach interrupts */
  detachInterrupt(getIntNum(Enc[enc].pinA));
s=-1;  /* we are done with encoder detach so -1         */
break; /* s=220 taken care of                           */
/* s=230 means GET ENCODER POSITION ***************** */
case 230:
/* the second value indicates the encoder number:
   either 0, 1 or 2                                   */
if (val¿47 && val¡51) –
  enc=val-48;        /* calculate encoder number       */
  /* send the value back                              */
  Serial.println(Enc[enc].pos);
s=-1;  /* we are done with encoder detach so -1         */
break; /* s=230 taken care of                           */
```

```
/* s=240 means RESET ENCODER POSITION **************** */
case 240:
/* the second value indicates the encoder number:
  either 0, 1 or 2                              */
if (val¿47 && val¡51) –
  enc=val-48;        /* calculate encoder number       */
  /* reset position                              */
  Enc[enc].pos=0;
s=-1;  /* we are done with encoder detach so -1        */
break; /* s=240 taken care of                        */
/* s=250 and 251 mean SET ENCODER DEBOUNCE DELAY ***** */
case 250:
/* the second value indicates the encoder number:
  either 0, 1 or 2                              */
if (val¿47 && val¡51)
  enc=val-48;        /* calculate encoder number       */
  s=251;  /* next we need the first attachment pin      */
else
  s=-1; /* if value is not an encoder then return to -1*/
break; /* s=250 taken care of                        */
case 251:
/* the third received value indicates the debounce
  delay value in units of approximately 0.1 ms each
  from abs('a')=97, 0 units, to abs('Π')=166, 69 units*/
if (val¿96 && val¡167) –
s=-1;  /* we are done with this so next state is -1     */
break; /* s=251 taken care of                        */
/* s=340 or 341 means ANALOG REFERENCE *************** */
case 340:
/* the second received value indicates the reference,
  which is encoded as is 0,1,2 for DEFAULT, INTERNAL
  and EXTERNAL, respectively. Note that this function
  is ignored for boards not featuring AVR or PIC32     */
#if defined(¨AVR¨) —— defined(¨PIC32MX¨)
switch (val) –
  case 48:
  analogReference(DEFAULT);
    break;
  case 49:
  analogReference(INTERNAL);
  break;
  case 50:
  analogReference(EXTERNAL);
  break;
  default:              /* unrecognized, no action   */
```

```
        break;
       ”
#endif
 s=-1;  /* we are done with this so next state is -1     */
      break; /* s=341 taken care of                        */
      /* s=400 roundtrip example function (returns the input)*/
        case 400:
      /* the second value (val) can really be anything here  */
      /* This is an auxiliary function that returns the ASCII
         value of its first argument. It is provided as an
         example for people that want to add their own code  */
      /* your own code goes here instead of the serial print */
      Serial.println(val);
      s=-1;  /* we are done with the aux function so -1       */
      break; /* s=400 taken care of                        */
      /* ******* UNRECOGNIZED STATE, go back to s=-1 ******* */
      default:
      /* we should never get here but if we do it means we
         are in an unexpected state so whatever is the second
         received value we get out of here and back to s=-1  */
      s=-1;  /* go back to the initial state, break unneeded */
     /* end switch on state s                          */
    /* end if serial available                         */
   /* end loop statement                                */
/* auxiliary function to handle encoder attachment            */
int getIntNum(int pin) –
/* returns the interrupt number for a given interrupt pin
   see http://arduino.cc/it/Reference/AttachInterrupt         */
switch(pin) –
  case 2:
    return 0;
  case 3:
    return 1;
  case 21:
    return 2;
  case 20:
    return 3;
  case 19:
    return 4;
  case 18:
    return 5;
  default:
    return -1;
/* auxiliary debouncing function                              */
void debounce(int del) –
  int k;
```

```
  for (k=0;k¡del;k++) –
    /* can't use delay in the ISR so need to waste some time
       perfoming operations, this uses roughly 0.1ms on uno  */
    k = k +0.0 +0.0 -0.0 +3.0 -3.0;
/* Interrupt Service Routine: change on pin A for Encoder 0  */
void isrPinAEn0()–
  /* read pin B right away                                */
  int drB = digitalRead(Enc[0].pinB);
    /* possibly wait before reading pin A, then read it        */
  debounce(Enc[0].del);
  int drA = digitalRead(Enc[0].pinA);
  /* this updates the counter                             */
  if (drA == HIGH) –   /* low-¿high on A? */
    if (drB == LOW) –  /* check pin B */
        Enc[0].pos++;  /* going clockwise: increment          */
     else –
        Enc[0].pos--;  /* going counterclockwise: decrement  */
    else –                   /* must be high to low on A */
    if (drB == HIGH) – /* check pin B */
        Enc[0].pos++;  /* going clockwise: increment          */
   else –
        Enc[0].pos--;  /* going counterclockwise: decrement  */
/* end counter update                                  */
/* end ISR pin A Encoder 0                              */
/* Interrupt Service Routine: change on pin B for Encoder 0  */
void isrPinBEn0()–
  /* read pin A right away                                */
  int drA = digitalRead(Enc[0].pinA);
   /* possibly wait before reading pin B, then read it        */
  debounce(Enc[0].del);
  int drB = digitalRead(Enc[0].pinB);
  /* this updates the counter                             */
  if (drB == HIGH) –   /* low-¿high on B? */
    if (drA == HIGH) – /* check pin A */
        Enc[0].pos++;  /* going clockwise: increment          */
else
        Enc[0].pos--;  /* going counterclockwise: decrement  *
else                   /* must be high to low on B */
    if (drA == LOW) –  /* check pin A */
        Enc[0].pos++;  /* going clockwise: increment          */
     else
        Enc[0].pos--;  /* going counterclockwise: decrement  */
 /* end counter update */
/* end ISR pin B Encoder 0  */
/* Interrupt Service Routine: change on pin A for Encoder 1  */
void isrPinAEn1()–
  /* read pin B right away                                */
```

```
  int drB = digitalRead(Enc[1].pinB);
    /* possibly wait before reading pin A, then read it         */
  debounce(Enc[1].del);
  int drA = digitalRead(Enc[1].pinA);
  /* this updates the counter                          */
  if (drA == HIGH) –   /* low-¿high on A? */
    if (drB == LOW) –  /* check pin B */
         Enc[1].pos++;  /* going clockwise: increment          */
     else –
         Enc[1].pos--;  /* going counterclockwise: decrement   */
    else – /* must be high to low on A                    */
    if (drB == HIGH) – /* check pin B */
         Enc[1].pos++;  /* going clockwise: increment          */
     else –
         Enc[1].pos--;  /* going counterclockwise: decrement   */
  /* end counter update                              */
/* end ISR pin A Encoder 1                             */
/* Interrupt Service Routine: change on pin B for Encoder 1   */
void isrPinBEn1()–
/* read pin A right away                               */
  int drA = digitalRead(Enc[1].pinA);
  /* possibly wait before reading pin B, then read it         */
  debounce(Enc[1].del);
  int drB = digitalRead(Enc[1].pinB);
  /* this updates the counter                          */
  if (drB == HIGH) –   /* low-¿high on B? */
   if (drA == HIGH) – /* check pin A */
         Enc[1].pos++;  /* going clockwise: increment          */
    else –
         Enc[1].pos--;  /* going counterclockwise: decrement   */
 else – /* must be high to low on B                     */
     if (drA == LOW) –  /* check pin A */
         Enc[1].pos++;  /* going clockwise: increment          */
     else –
         Enc[1].pos--;  /* going counterclockwise: decrement   */
 /* end counter update                               */
/* end ISR pin B Encoder 1                             */
/* Interrupt Service Routine: change on pin A for Encoder 2   */
void isrPinAEn2()–
  /* read pin B right away                             */
  int drB = digitalRead(Enc[2].pinB);
  /* possibly wait before reading pin A, then read it         */
  debounce(Enc[2].del);
  int drA = digitalRead(Enc[2].pinA);
  /* this updates the counter                          */
  if (drA == HIGH) –   /* low-¿high on A? */
```

```
    if (drB == LOW) –  /* check pin B */
            Enc[2].pos++;  /* going clockwise: increment          */
     else –
            Enc[2].pos--;  /* going counterclockwise: decrement  */
  else – /* must be high to low on A                         */
    if (drB == HIGH) – /* check pin B */
            Enc[2].pos++;  /* going clockwise: increment          */
   else –
            Enc[2].pos--;  /* going counterclockwise: decrement  */
  /* end counter update                               */
/* end ISR pin A Encoder 2                              */
/* Interrupt Service Routine: change on pin B for Encoder 2  */
void isrPinBEn2()–
/* read pin A right away                              */
  int drA = digitalRead(Enc[2].pinA);
  /* possibly wait before reading pin B, then read it         */
  debounce(Enc[2].del);
  int drB = digitalRead(Enc[2].pinB);
/* this updates the counter                            */
  if (drB == HIGH) –   /* low-¿high on B? */
    if (drA == HIGH) – /* check pin A */
            Enc[2].pos++;  /* going clockwise: increment          */
     else
            Enc[2].pos--;  /* going counterclockwise: decrement  */
   else  /* must be high to low on B                        */

    if (drA == LOW) –  /* check pin A */
            Enc[2].pos++;  /* going clockwise: increment          */
    else –
            Enc[2].pos--;  /* going counterclockwise: decrement  */
  /* end counter update                               */
/* end ISR pin B Encoder 2                              */
```

## Genuine code

### · Matlab code

Now the programming code developed by the author in Matlab is presented, clearly separatd by functions/scripts.

## Global_sensors

```
clc;
clear all;
close all;

%Select here the number of samples that the software is going to gather.
p=0.1;%time between messures
n_sample=400; %T_sampling=n_sample*(p+0.15) aproximately

Rgas=8.3144621;%J/molK
Mair=28.9645e-3;%kg/mol
Rair=Rgas/Mair;%J/kgK

%the pin of the fan is defined and the duty initialized to 0.
fan=9;
duty=zeros(n_sample, 1);

%Diferential pressure transducer calibration values
alpha1=0.9933;
alpha2=1.0018;

%PID controlller adjustment
kp=1;
ki=0.3;
kd=0;% Finally the derivative action has been beittled
eant=0;
e_int=0;
%Read the manufacturer's table of T as a function of Rntc
[ R , Tc , Tk ] = Excel_read( );

%connection of MAtlab with Arduino is achieved
port= 'COM6' ;
A=arduino(port);
port.BaudRate=115200;
%As soon as it the connection is started a 0 is writeen in the output, so
%the fan is set at v=0rpm;
A.pinMode(fan, 'output');
A.analogWrite(fan, 0);

VmaxA=5;%Arduino maximum voltage in
VminA=0;%Arduino minimum voltage in


%initial values and global variables for the graphical user's interface
global Te
global Patmo
global Vmax
global mmax
global dy
global det
global inc
global Slival
global mode
mode=0;
[ lecturNTC,lecturPD,lecturPA ] = GetData(A);
```

```
voffset=lecturPD*5/1023;
Te= NTC( VmaxA , lecturNTC, R, Tk);%in K
Patmo= AtmP ( VmaxA , lecturPA );% in Pa
dif_Pmax = 6000;%in Pa
Vmax=Velocity ( Te, dif_Pmax , Patmo, alpha1 , alpha2, Rair );
mmax=Vmax/10; %maximum slope defined slope
a=0;%auxiliar variable

GUI;
v_req=0;


if mode==1
Graph=figure('Name','Sensor Values Plot','NumberTitle','off',
'units','normalized','outerposition',[0 0 .7 1]);
else
Graph=figure('Name','Sensor Values Plot','NumberTitle','off',
'units','normalized','outerposition',[0 0 1 1]);
end

i=1;
t0=tic;
% t(1)=t0;
while (i<n_sample)
%   Required velocity calculus depending of the mode.
  if mode == 1
  v_req=(Slival/100)*Vmax;
  end
  if mode == 2
      if (i>=3)
        if (t(i-1)>3 && a==0)%ramp starts at 3 seconds
          dti=t(i-1)-t(i-2);% this is only calculated the first time.
          inct=det/dti;
          incy=dy/inct;
          v_req=v_req+incy;
          a=1;
        end
        if t(i-1) >= 3%ramp starts at 3 seconds
           if v_req<dy
           v_req=v_req+incy;
           else
           v_req=dy;
           end
        end
       else
          v_req=0;
       end
  end
  if mode == 3
      v_req=inc;
  end
  %data gathering thanks to Arduino, values in bits from 0 to 1023
  [ lecturaNTC(i),lecturaPD(i),lecturaPA(i) ] = GetData(A);

  T(i)= NTC( VmaxA , lecturaNTC(i), R, Tk);%in K

  dif_P(i)   = DifP( VmaxA, lecturaPD(i), voffset );%in Pa
  if (i>4) dif_P(i)=(dif_P(i)+dif_P(i-1)+dif_P(i-3)+dif_P(i-2))/4;
  end   %this step is done to avoid peaks in the readings and smooth the
results
  Patm(i) = AtmP ( VmaxA , lecturaPA(i) );% in Pa
  %once all phyisical data is collected in proper unities the speed is
  %calculated
  v (i) = Velocity ( T(i), dif_P(i) , Patm(i), alpha1 , alpha2, Rair );
  % the required velocity is compared with the current one
  error(i)=v_req-v(i);
  if abs(error(i))<0.1
```

```
    error(i)=0;% added because there will be a minimum of error due to
Arduino's output precision.
  end

 [duty(i) , eant, e_int]=PID(error(i), kp, ki, kd, eant, e_int);
  m(i)=round(duty(i));%the result is rounded to send an integer to Arduino
  if m(i)<0
      m(i)=0;
  elseif m(i)>255
      m(i)=255;
  end
  %The calculated duty is send to Arduino's fan pin (9).
  if v_req==0
  A.analogWrite(fan, 0);
  else
  A.analogWrite(fan, m(i));
  end
  t(i)=toc(t0);
  t(i)=t(i)-t(1);

  Plot( t, T, Patm, dif_P, v , v_req, Graph);

  pause(p);%for stability reasons
  i=i+1;

end
%when the code finishes it writes a 0 and stopes the fan.
fi=0;
A.analogWrite(fan, fi);
%data is saved here in a textfile
SaveData( t,T,dif_P,Patm,v,n_sample );
%finally, connection with Arduino is eliminated.
delete(A);
```

## Excel_read

```
function [ R , Tc , Tk ] = Excel_read( )
%This function reads the excel created with "My Vishay NTC curve" applet
%that the brand provides for users of their Hardware. In this case the
%model NTCLE100E3104JB0 has been used
filename='RT_curve.xls';
sheet='Hoja1';
TempRange='B30:B1230';
RRange='C30:C1230';
Tc=xlsread(filename,sheet,TempRange);
R=xlsread(filename,sheet,RRange);
Tk=Tc+273.15;
end
```

## Get_data

```
function [ lecturaNTC,lecturaPD,lecturaPA ] = GetData(A)
%GetData Summary of this function goes here
%   Detailed explanation goes here
NTC_sens = 1;  %define a pin for the NTC sensor
TransducerPin = 2;  %define a pin for the pressure sensor
AtmPressPin = 0;  %define a pin for the sensor
lecturaNTC=analogRead(A,NTC_sens);%in bits number
lecturaPD=analogRead(A,TransducerPin);%in bits number
lecturaPA=analogRead(A,AtmPressPin);%in bits number
end
```

## NTC

```
function [ T ] = NTC( VmaxA , lecturaNTC, R, Tk )
%Circuit without opam
V1=5;%Supply tension, in V
R1=99700;%in Ohm 99700
V_circ=lecturaNTC*(VmaxA/1023);%Translate te bits to voltage
R_circ=(V_circ*R1/V1)/(1-(V_circ/V1)); %calculate resistance offered by the
NTC
T= Temp_NTC( R_circ, R, Tk );
end
```

## Temp_NTC

```
function [ T_out ] = Temp_NTC( rin, R, Tk )
for i=1:1000
rest(i)=R(i)-rin;
end
[m,j]=min(abs(rest));
%Looking for the value to interpolate.
intx(1)=R(j);
inty(1)=Tk(j);
if(m~=0)
    if(abs(R(j+1)-rin)<abs(R(j-1)-rin))
     intx(2)=R(j+1);
     inty(2)=Tk(j+1);
     %interpolate between two points
     T_out=interp1(intx,inty,rin);%in Kelvin
    else
     intx(2)=R(j-1);
     inty(2)=Tk(j-1);
     %interpolate between two points
     T_out=interp1(intx,inty,rin);%in Kelvin

    end
else
    T_out=Tk(j);
end
end
```

## AtmP

```
function [ Patm ] = AtmP ( VmaxA , lecturaPA )
V_read=lecturaPA*(VmaxA/1023);%Translate te bits to voltage
Patm=(83.1846*V_read+891.9367)*100;
end
```

## dif_P

```
function [ dif_P ] = DifP( VmaxA, lecturaPD,Voffset )
V_read=lecturaPD*(VmaxA/1023)-Voffset;%Translate the bits to voltage
%Recalibration procedure. It was recalibrated taking into account the
%conditioning circuit, so the read value has just to be multiplied.
dp_dv=1153.006658;

dif_P=dp_dv*V_read;
if dif_P<0
    dif_P=0;
end
end
```

## Velocity

```
function [ v ] = Velocity ( T, dif_P , Patm, alpha1 , alpha2, Rair)
v=sqrt((2*alpha1*dif_P*Rair*T)/(Patm-(alpha2*dif_P)));
end
```

## GUI

```
scrsz = get( 0 , 'ScreenSize' );
global mode
mode=0;
SelectMode( scrsz );
while mode==0
    pause (0.1);
end
close all;
if mode==1
    Slival=0;
    Slider1(Te, scrsz, Patmo, Vmax);
elseif mode==2
    dy='0';
    det='0';
    global out
    out=0;
    rampa(Te, scrsz, Patmo, Vmax, mmax);
    while out==0
    pause (0.1);
    end
    close all;
    slope=0;
    dy=str2num(dy);
    det=str2num(det);
    slope=dy/det;

    if slope>mmax
        close all;
        errormsg='WARNING! Maximum slope exceeded';
        errordlg(errormsg, 'ERROR');
    end
elseif mode==3;
    inc=0;
    global out
    out=0;
    step(Te, scrsz, Patmo, Vmax);
    while out==0
    pause (0.1);
    end
    close all;
    inc=str2num(inc);
    if inc>Vmax
        close all;
        errormsg='WARNING! Maximum speed exceeded';
        errordlg(errormsg, 'ERROR');
    end
end
```

## Slider1

```
function  Slider1(Temp, scrsz, Patm, Vmax)
%SelectFaces It is the Graphical User's Interface (UI) that asks for the
% number of faces that will appear in the photo.It saves the chosen
% option in a file that the "main" will read afterwards.
% It has been created with  4 pushbuttons thanks to the help of "guide",
% which is a  Matlab's utility.
%
```

```
% Input
%           Scrsz: screen size (0 0 width height)

Fig = figure( 'Name','Manual
Mode','Menubar','none','Resize','off','WindowStyle','modal','Position',[
scrsz(3)-510 scrsz(4)-325 500 300]);
set( Fig , 'NumberTitle' , 'off' );

 pb = uipanel( 'Parent' , Fig , 'Units' , 'pixels' ,...
            'Position' , [5 5 490 290] );
%  sd = uipanel ( '
vm=round(Vmax*10)/10;
% Display thing
slid1 = uicontrol( pb , 'Style', 'slider',  'Min', 0, 'Max' ,100 ,
'Position' ,  [50 80 300 20] , 'callback' , @Slider );
t1 = uicontrol( pb ,'Style', 'text','String' , 'Velocity' , 'Position' ,
[180 105 50 15] );
t2 = uicontrol( pb ,'Style', 'text','String' , 'Temperature (K)' ,
'Position' , [50 215 100 25] );
t3 = uicontrol( pb ,'Style', 'text','String' , Temp , 'Position' , [50 200
50 25] );
t4 = uicontrol( pb ,'Style', 'text','String' , 'Ambient Pressure(Pa)' ,
'Position' , [50 170 130 25] );
t5 = uicontrol( pb ,'Style', 'text','String' , Patm , 'Position' , [55 155
50 25] );
% b3 = uicontrol( pb ,'String' , 'Step Function' , 'Position' , [50 25 200
75] , 'callback' , @threeFace);
t6 = uicontrol( pb ,'Style', 'text','String' , 'Vmin=0 m/s' , 'Position' ,
[20 50 100 25] );
t7 = uicontrol( pb ,'Style', 'text','String' , 'Vmax=' , 'Position' , [320
50 35 25] );
t8 = uicontrol( pb ,'Style', 'text','String' , vm , 'Position' , [355 40 30
35] );
t9 = uicontrol( pb ,'Style', 'text','String' , 'm/s' , 'Position' , [385 50
20 25] );
t10 = uicontrol( pb ,'Style', 'text','String' , '0%' , 'Position' , [50 105
20 15] );
t11 = uicontrol( pb ,'Style', 'text','String' , '100%' , 'Position' , [320
105 30 15] );
end

function Slider(source,callbackdata)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%         val =source.Value
        % For R2014a and earlier:
        global Slival
        Slival =get(source,'Value');
end
```

## Rampa

```
function  [] = rampa(Temp, scrsz, Patm, Vmax, mmax)
%SelectFaces It is the Graphical User's Interface (UI) that asks for the
% number of faces that will appear in the photo.It saves the chosen
% option in a file that the "main" will read afterwards.
% It has been created with  4 pushbuttons thanks to the help of "guide",
% which is a  Matlab's utility.
%
% Input
%           Scrsz: screen size (0 0 width height)
Fig = figure( 'Name','Ramp
Mode','Menubar','none','Resize','off','WindowStyle','modal','Position',[
scrsz(3)-510 scrsz(4)-325 500 300]);
set( Fig , 'NumberTitle' , 'off' );
```

```
 pb = uipanel( 'Parent' , Fig , 'Units' , 'pixels' ,...
             'Position' , [5 5 490 290] );
% Display thing s

t2 = uicontrol( pb ,'Style', 'text','String' , 'Temperature (K)=' ,
'Position' , [50 215 100 25] );
t3 = uicontrol( pb ,'Style', 'text','String' , Temp , 'Position' , [142 215
30 25] );
t4 = uicontrol( pb ,'Style', 'text','String' , 'Ambient Pressure(Pa)=' ,
'Position' , [48 200 130 25] );
t5 = uicontrol( pb ,'Style', 'text','String' , Patm , 'Position' , [175 200
40 25] );
t7 = uicontrol( pb ,'Style', 'text','String' , 'Vmax(m/s)=' , 'Position' ,
[55 175 65 25] );
t8 = uicontrol( pb ,'Style', 'text','String' , Vmax , 'Position' , [122 185
15 15] );
t1 = uicontrol( pb ,'Style', 'text','String' , 'Write here the parameters
and press: RUN' , 'Position' , [50 140 200 25] );
t6 = uicontrol( pb ,'Style', 'text','String' , 'Time Step: ' , 'Position' ,
[40 80 80 25] );
tx = uicontrol( pb ,'Style', 'text','String' , 'Speed Step: ' , 'Position' ,
[43 45 80 25] );
ti = uicontrol (pb , 'Style', 'edit','String',  0, 'Position' , [130 85 40
25], 'Callback', @time );
y =  uicontrol (pb , 'Style', 'edit','String',  0, 'Position' , [130 50 40
25], 'Callback', @increment );
b1 = uicontrol( pb , 'String' , 'RUN' , 'BackgroundColor',[0.196 0.804
0.196], 'Position' ,  [360 190 40 25] , 'callback' , @RUN );
ax=axes('Units','pixels', 'Position', [320 20 150 150] );
imshow('ramp.jpg');
casilast = uicontrol( pb ,'Style', 'text','String' , 'WARNING! The maximum
ramp slope is : ' , 'BackgroundColor',[1 0 0],...
'FontSize',10, 'Position' , [15 15 250 25] );
b2 = uicontrol( pb , 'String' , 'CLOSE' , 'BackgroundColor',[1 0 0],
'Position' ,  [415 250 70 35] ,'FontSize',15, 'callback' , @close );
last = uicontrol( pb ,'Style', 'text','String' , mmax , 'Position' , [265 23
25 15] );
end

function time(hObject, eventdata , handles)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 global det
det = get(gcbo, 'String');
  end
function increment(hObject, eventdata , handles)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global dy
dy = get(gcbo, 'String');
end
function RUN(hObject, eventdata , handles)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global out
out=1;
end
function close(hObject, eventdata , handles)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global out
out=1;
end
```

## Step

```
function  [] = step(Temp, scrsz, Patm, Vmax)
%SelectFaces It is the Graphical User's Interface (UI) that asks for the
% number of faces that will appear in the photo.It saves the chosen
% option in a file that the "main" will read afterwards.
% It has been created with  4 pushbuttons thanks to the help of "guide",
% which is a  Matlab's utility.
%
% Input
%            Scrsz: screen size (0 0 width height)

Fig = figure( 'Name','Ramp
Mode','Menubar','none','Resize','off','WindowStyle','modal','Position',[
scrsz(3)-510 scrsz(4)-325 500 300]);
set( Fig , 'NumberTitle' , 'off' );

 pb = uipanel( 'Parent' , Fig , 'Units' , 'pixels' ,...
             'Position' , [5 5 490 290] );

t2 = uicontrol( pb ,'Style', 'text','String' , 'Temperature (K)=' ,
'Position' , [50 215 100 25] );
t3 = uicontrol( pb ,'Style', 'text','String' , Temp , 'Position' , [142 215
30 25] );
t4 = uicontrol( pb ,'Style', 'text','String' , 'Ambient Pressure(Pa)=' ,
'Position' , [48 200 130 25] );
t5 = uicontrol( pb ,'Style', 'text','String' , Patm , 'Position' , [175 200
40 25] );
t7 = uicontrol( pb ,'Style', 'text','String' , 'Vmax(m/s)=' , 'Position' ,
[55 175 65 25] );
t8 = uicontrol( pb ,'Style', 'text','String' , Vmax , 'Position' , [122 185
15 15] );
t1 = uicontrol( pb ,'Style', 'text','String' , 'Write here the speed step
and press: RUN' , 'Position' , [50 140 200 25] );
t6 = uicontrol( pb ,'Style', 'text','String' , 'Speed Step: ' , 'Position' ,
[43 80 80 25] );
y = uicontrol (pb , 'Style', 'edit','String',  0, 'Position' , [130 85 40
25], 'Callback', @increment );
b1 = uicontrol( pb , 'String' , 'RUN' , 'BackgroundColor',[0.196 0.804
0.196], 'Position' ,  [350 90 80 50] ,...
'FontSize',15, 'Callback' , @RUN );
b2 = uicontrol( pb , 'String' , 'CLOSE' , 'BackgroundColor',[1 0 0],
'Position' ,  [415 250 70 35] ,'FontSize',15, 'Callback' , @close );
casilast = uicontrol( pb ,'Style', 'text','String' , 'WARNING! The maximum
speed is : ' , 'BackgroundColor',[1 0 0],...
'FontSize',10, 'Position' , [15 25 230 20] );
last = uicontrol( pb ,'Style', 'text','String' , Vmax , 'Position' , [245 27
20 15] );
end
function increment(hObject, eventdata , handles)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
  global inc
  inc = get(gcbo, 'String');
end
function RUN(hObject, eventdata , handles)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global out
out=1;
end
function close(hObject, eventdata , handles)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global out
out=1;
end
```

### SelectMode

```
function  []  = SelectMode ( scrsz )
%SelectFaces It is the Graphical User's Interface (UI) that asks for the
% number of faces that will appear in the photo.It saves the chosen
% option in a file that the "main" will read afterwards.
% It has been created with  4 pushbuttons thanks to the help of "guide",
% which is a  Matlab's utility.
%
% Input
%            Scrsz: screen size (0 0 width height)
Fig = figure( 'Name','Mode
Selection','Menubar','none','Resize','off','WindowStyle','modal','Position',
[ scrsz(3)-312 scrsz(4)-368 309 335]);

set( Fig , 'NumberTitle' , 'off' );

pb = uipanel( 'Parent' , Fig , 'Units' , 'pixels' ,...
            'Position' , [5 5 300 325] );

% Display the buttons
b1 = uicontrol( pb , 'String' , 'Manual Mode' , 'Position' ,  [50 225 200
75] , 'callback' , @Manual );
b2 = uicontrol( pb ,'String' , 'Ramp Function' , 'Position' , [50 125 200
75] , 'callback' , @Ramp);
b3 = uicontrol( pb ,'String' , 'Step Function' , 'Position' , [50 25 200 75]
, 'callback' , @Step);

end
function Manual(hObject, eventdata , handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global mode
mode=1;
end
function Ramp(hObject, eventdata , handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global mode
mode=2;
end
function Step(hObject, eventdata , handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global mode
mode=3;
end
```

### PID

```
function [ duty , eant, e_int] = PID( e, kp, ki, kd, eant , e_int )
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here
P=e*kp;

e_int=e_int+e;
%anti wind up
if(e_int>=(1024*0.8/ki))% we are limiting the error to a maximum value.
    e_int=(1024*0.8/ki);
end
if (e_int<=-(1024*0.8/ki))
    e_int=-(1024*0.8/ki);
end
```

```
Int=ki*e_int;

e_der=(e-eant);
eant=e;% to avoid problems of indexing position 0.
D=e_der*kd;
duty1=P+Int+D;
if(duty1>=973)
    duty1=973;
end % duty is limited to maximum of 95%
if(duty1<=50)
    duty1=50;
end % duty is limited to minimum of 5%
%in the end the output is not a 10 bit number but a 8 bit, so the duty is
%reduced from 0-1023  to 0-255
duty=duty1/4;
end
```

## Plot

```
function [  ] = Plot( t, T, Patm, dif_P,v,v_req , Graph)
%UNTITLED3 Summary of this function goes here
%   Detailed explanation goes here
figure(Graph);
subplot(2,2,1)
plot(t,T)
title('Subplot 1: Temperature')
ylim([0 325])

subplot(2,2,2)
plot(t,dif_P)
title('Subplot 2: Diferential Pressure')

subplot(2,2,3)
plot(t,Patm)
title('Subplot 3: Atmospheric pressure')
ylim([95000 110000])

subplot(2,2,4)
 plot(t,v)
 hold on
 plot(get(gca,'xlim'), [v_req v_req], 'r'); % Adapts to x limits of current
axes
 hold off
 title('Subplot 4: velocity')
 end
```

## SaveData

```
function [ ] = SaveData( t,T,dif_P,Patm,v,n_sample )
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
file=fopen(['TunnelData.txt'],'w');
fprintf(file,'%6s %12s %20s %26s %32s\r\n','time(s)','T(K)','Dif
Pressure(Pa)', 'AtmPressure(Pa)', 'velocity(m/s)');
for j=1:n_sample-1
fprintf(file,'%6.4f %12.2f %20.2f %26.2f
%32.3f\r\n',t(j),T(j),dif_P(j),Patm(j),v(j) );
end
fclose(file);

end
```

## Arduino code

This code is basically the same but only makes use of the Arduino board. It requires an external software to save the data printed in the serial monitor, which will be CoolTerm. This has been done concurrently to the subject *Telemetry and Smart Electronic projects,* but it is not really used because it is more difficult to know the values without ploting them and the user's interface is poorer.

```
#include <TimerOne.h> // include TimerOne library

int NTC_pin=1;
int Patm_pin=0;
int Pdif_pin=2;
int fan_pin=9;
int duty=0;

/*******************************USER INTERFACE*********************************/
//Select mode 1for manual control; 2 for ramp mode or 3 for step mode.
int mode=1;// Select mode 1 ,2 , 3
//if mode=2 select the speed increment and the time interval to go from V=0 to V selected.
// BE CAREFUl: MAXIMUM SLOPE IS : Dv/Dt=10;
float Dv=10;// in m/s
float Dt=10;// in seconds
//if mode=3 (step) select above only the target speed(Dv).
/*****************************************************************************/

/***********************************CONSTANTS*********************************/
float Rair=8.3144621/28.9645e-3;//J/kgK  Rgas/Mair
//Diferential pressure transducer calibration values
float alpha1=0.9933;
float alpha2=1.0018;
//Regulator parameters
float kp=0.2;
int ki=0.6;
int kd=0;
float Vmax=150; // m/s
float e_int=0, P=0, I=0, D=0, e_der=0, e_ant=0, v_req=0;
int intensity=0, key=0;
/*****************************************************************************/

void setup(){/************************SETUP BEGIN *****************************/
Serial.begin(9600);
pinMode(NTC_pin, INPUT);
pinMode(Patm_pin, INPUT);
pinMode(Pdif_pin, INPUT);
pinMode(fan_pin, OUTPUT);
Timer1.pwm(fan_pin,duty);
Timer1.initialize(500000);// the time is in microseconds 10-6
Timer1.attachInterrupt(WindTunel);
}/*******************************SETUP END ***********************************/

float voffset=analogRead(Pdif_pin)*(5/1023);//inside the setum it doesn not work.

/*******************************NTC Temperature value**************************/
float NTC(float lt){
float a=738.7;//polynomial obtained with Matlab according to the manufacturer's datasheet
float b=-0.124;
float c=-152.3;
float Vcirc=lt*5/1023;
float Rcirc=(Vcirc*(99700/5))/(1-(Vcirc/5));
float T=a*pow(Rcirc,b)+c;
return T;
```

```
}/*****************************************************************************/

/*******************************Ambient pressure value***************************/
float AtmP(int lp){
float V_read=lp*(5/1023);
float Patm=100*(83.1846*V_read+891.9367);
return Patm;
}/*****************************************************************************/

/*******************************Differential pressure value***************************/
float difP(int lpd, float voff){
float dp_dv=1153.006658;
float vread=lpd*(5/1023)-voff;
float pdif=vread*dp_dv;
return pdif;
}/*****************************************************************************/


/*******************************Speed value***************************/
float Speed(float Rair, float T, float Patm, float Pdif, float alpha1, float alpha2){
float V=sqrt((2*alpha1*Pdif*Rair*T)/(Patm-alpha2*Pdif));
return V;
}/*****************************************************************************/

/*******************************REGULTATOR PARAMETERS*****************************/

void PID(float e){// reb l'error provinent de malab i el diferencial de temps
P=e*kp;// PROPORTIONATL
e_int=e_int+e;
if(e_int>=512)e_int=512;//we are limiting the error to a maximum value. anti wind-up
if (e_int<=-512)e_int=-512;//we are limiting the error to a minimum value.
I=ki*e_int;// INTEGRAL
e_der=(e-e_ant);
e_ant=e;
D=e_der*kd;// DERIVATIVE
float PID=P+I+D+0.5;//0.5 added because the Cast comand doesn't round, it truncates the num.
duty=(int) PID;// PID

if(duty>=973)duty=973; // limita el duty a un mÃ xim del 95% OJO LIMITACIO DP MAX!
//if(duty<=50)duty=50; // limita el duty a un mÃnim del 5%
Timer1.pwm(fan_pin,duty);// thanks to Timer1 the output here it is possible to be a 10 bit number
}
/*********************************************************************************/

/****************************MANUAL MODE. FAN INTENSITY****************************/
int Intensity (){
if (Serial.available() > 0) {
      key = Serial.read();// read what key is pressed. + =43; -=45.
      if (key==43){// if + is pressed, intensity or velocity are //increased
        Serial.print("I received:");// it prints on the serial window what it is inside ()
        Serial.println("+");
        intensity+=5;
        }
      if (key==45){ // otherwise, if â€“ is pressed, they are decreased
        Serial.print("I received: ");
        Serial.println("-");
        intensity-=5;
        }
          if (intensity<=100){
            if (intensity<0)
              {Serial.print("MINIMUM Intensity="); // message to print when intensity is minimum and it
cannot be decreased more
              intensity=0;
              Serial.print(intensity);
              Serial.println("%");
              }else if(intensity==100){
```

```
            Serial.print("MAXIMUM Intensity=");//message to print when intensity is maximum and it cannot
be increased more
            Serial.print(intensity);
            Serial.println("%");
           } else{
            Serial.print("Intensity=");
            Serial.print(intensity);
            Serial.println("%");
           }
   } else { intensity=1023;
            Serial.print("MAXIMUM Intensity=");
            Serial.print(intensity);
            Serial.println("%");
         }
         }
         return intensity;
         }
  /*****************************************************************************/

/*******************************GLOBAL FUNCTION **********************************/
void WindTunel (void){
//GetData();
int lecturaT=analogRead(NTC_pin);//in bits number
int lecturaPd=analogRead(Pdif_pin);//in bits number
int lecturaPa=analogRead(Patm_pin);//in bits number
float Temp=NTC(lecturaT);
float Patm=AtmP(lecturaPa);
float Pdif=difP(lecturaPd, voffset);
float v=Speed( Rair, Temp, Patm, Pdif, alpha1, alpha2);
if (mode==1){
  v_req=Intensity ()*(Vmax/100);
  }else if (mode==2){
    int dti=Dt/0.5;
      if (v_req<Dv){
      v_req=v_req+(Dv/dti);
      }else{
      v_req=Dv;
      }
  }else if (mode==3){
    v_req=Dv;
  }else {
   Serial.print("ERROR:Wrong mode selected");
  }
//everything is printed to the Serial montiro in order that CoolTerm is able to save the data.
Serial.print("Temperatura (K): ");
Serial.println (Temp);
Serial.print("Atmospheric pressure lectura (Pa): ");
Serial.println (Patm);
Serial.print("Differential pressure lectura (Pa): ");
Serial.println (Pdif);
Serial.print("Required speed (m/s): ");
Serial.println (v_req);
Serial.print("Air speed (m/s): ");
Serial.println (v);

float error=v-v_req;
PID (error);

  }
 /*****************************************************************************/

void loop() {

 }
```

## B. Calibration Documents

Here there are presented some of the tables (the most descriptive ones) of the calibration an verifications process. Nonetheless looking at the excels would give the reader a complete information about the procedure and all data of each experiment.

### Sensor verification

### NTC

| day | hour | sensor Testo( ° C) | NTC reading ( ° C) | absolute error ( ° C) |
|---|---|---|---|---|
| 02/06/2015 | 9:30 | 22,9 | 22,96 | 0,0581 |
| 02/06/2015 | 17:00 | 23,3 | 23,29 | 0,0066 |
| 03/06/2015 | 9:20 | 22,8 | 22,84 | 0,0433 |
| 03/06/2015 | 16:30 | 23,6 | 23,59 | 0,0137 |
| 04/06/2015 | 9:15 | 23,1 | 23,13 | 0,0257 |
| 04/06/2015 | 17:25 | 23,8 | 23,84 | 0,0432 |
| 05/06/2015 | 9:00 | 22,7 | 22,75 | 0,0532 |
| 05/06/2015 | 14:00 | 23,4 | 23,43 | 0,0293 |

Table B.1 NTC verification experiment

### Atmospheric pressure validation

| lmabrecht reading (hPa) | T ( ° C) | temperature correction (-hPa) | latitude and gravity correct. (hPa) | height correction (hPa) |
|---|---|---|---|---|
| 989,6 | 22 | 3,928 | -0,369 | 0,058 |
| 988,6 | 22 | 3,924 | -0,369 | 0,058 |
| 988,3 | 22 | 3,923 | -0,369 | 0,058 |
| 988,5 | 22 | 3,923 | -0,369 | 0,058 |

Table B.2 Lambrecht barometer readings and corrections

| sensor read (V) | Pressure read (Pa) | lambrecht value corrected (hPa) | absolute error (Pa) | relative error (%) |
|---|---|---|---|---|
| 3,18 | 92720 | 985,25 | 5804,51 | 5,89 |
| 3,14 | 92560 | 984,25 | 5864,91 | 5,96 |
| 3,13 | 92520 | 983,95 | 5875,03 | 5,97 |
| 3,13 | 92520 | 984,15 | 5895,03 | 5,99 |

Table B.3 Lambrecht readings vs sensor readings

**Differential pressure verification**

| Visual (Pa) Mean | Sensor (Pa) | absolute error (Pa) | relative error (%) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 26,1 | 12,1 | -14,0 | -53,7 |
| 128,0 | 124,3 | -3,7 | -2,9 |
| 322,3 | 336,1 | 13,8 | 4,3 |
| 620,9 | 653,9 | 33,0 | 5,3 |
| 1023,8 | 1075,0 | 51,2 | 5,0 |
| 1530,9 | 1595,9 | 65,0 | 4,2 |
| 2132,9 | 2203,2 | 70,3 | 3,3 |
| 2803,6 | 2900,8 | 97,3 | 3,5 |
| 3732,6 | 3861,0 | 128,4 | 3,4 |
| 4675,8 | 4801,4 | 125,6 | 2,7 |
| 3725,5 | 3868,5 | 143,1 | 3,8 |
| 2805,9 | 2909,9 | 103,9 | 3,7 |
| 2123,4 | 2004,6 | -118,8 | -5,6 |
| 1528,6 | 1769,3 | 240,8 | 15,8 |
| 1026,2 | 1086,6 | 60,5 | 5,9 |
| 611,4 | 663,6 | 52,1 | 8,5 |
| 329,4 | 343,2 | 13,7 | 4,2 |
| 139,8 | 130,3 | -9,5 | -6,8 |
| 35,5 | 12,1 | -23,5 | -66,0 |

Table B.4 Differential pressure verification with U-Tube

| Manometer corrected (Pa) | mean sensor(Pa) | absolute error (Pa) | relative error (%) |
|---|---|---|---|
| 29 | 17,5 | -11,2 | -39,1 |
| 182 | 189,9 | 8,0 | 4,4 |
| 479 | 509,5 | 30,9 | 6,5 |
| 680 | 726,2 | 46,5 | 6,8 |
| 1005 | 1073,6 | 68,5 | 6,8 |
| 1235 | 1322,3 | 87,5 | 7,1 |
| 1484 | 1588,2 | 104,5 | 7,0 |

Table B.5 Differential pressure verification with alcohol manometer

| qc corrected (Pa) | Rho (kg/m^3) | Dif P (Pa) | Vread (m/s) | V˙Arduino (m/s) | error(m/s) |
|---|---|---|---|---|---|
| 68,723 | 1,158 | 69,186 | 10,897 | 10,39 | 0,507 |
| 147,263 | 1,158 | 148,256 | 15,958 | 15,63 | 0,328 |
| 363,248 | 1,158 | 365,698 | 25,091 | 25,1 | -0,009 |
| 490,875 | 1,158 | 494,186 | 29,187 | 28,956 | 0,231 |

| | | | | | |
|---|---|---|---|---|---|
| 883,575 | 1,158 | 889,535 | 39,237 | 39,19 | 0,047 |
| 1276,275 | 1,158 | 1284,884 | 47,253 | 46,9 | 0,353 |
| 1472,625 | 1,158 | 1482,558 | 50,810 | 50,47 | 0,340 |

Table B.6 Speed verification

### Calibration process

### Differential pressure

| Ref.value(Pa) | Corrected(Pa) | Vread(mean) | Std. deviation(V) | ±dv with ic=95% |
|---|---|---|---|---|
| 50 | 49,16875 | 0,026 | 0,00564 | 0,006386445 |
| 100 | 98,3375 | 0,065 | 0,00564 | 0,006386445 |
| 200 | 196,675 | 0,152 | 0,00000 | 0 |
| 400 | 393,35 | 0,321 | 0,00564 | 0,006386445 |
| 800 | 786,7 | 0,664 | 0,00978 | 0,011061647 |
| 1200 | 1180,05 | 1,006 | 0,00978 | 0,011061647 |
| 1600 | 1573,4 | 1,356 | 0,00564 | 0,006386445 |
| 1200 | 1180,05 | 1,004 | 0,00564 | 0,006386445 |
| 800 | 786,7 | 0,666 | 0,00564 | 0,006386445 |
| 400 | 393,35 | 0,324 | 0,00564 | 0,006386445 |
| 200 | 196,675 | 0,153 | 0,00564 | 0,006386445 |
| 100 | 98,3375 | 0,065 | 0,00564 | 0,006386445 |
| 50 | 49,16875 | 0,026 | 0,00564 | 0,006386445 |
| 0 | 0 | 0,005 | 0,00000 | 0 |

Table B.7 Differential pressure sensor calibration

### Atmospheric pressure calibration

| Vard(V) | Patm˙mean(Pa) h=290m |
|---|---|
| 1,162 | 988,696 |
| 1,167 | 989,015 |
| 1,172 | 989,022 |
| 1,177 | 989,921 |
| 1,182 | 990,294 |
| 1,187 | 990,806 |
| 1,191 | 991,175 |
| 1,196 | 991,597 |
| 1,201 | 991,556 |

Table B.8 Atmospheric pressure sensor calibration

### Alpha 1 and Aplha 2 obtention

| qc=f(Dp nozz) | | | | |
|---|---|---|---|---|
| Pitot read (q) mm H2O | q (Pa) | q·mean value(Pa) | transducer(V) | Ap nozz (Pa) |
| 42 | 808,68 | 808,68 | 0,715 | 824,70 |
| 108 | 2079,46 | 2079,46 | 1,821 | 2099,23 |
| 204 | 3927,88 | 3927,88 | 3,449 | 3976,32 |
| 316 | 6084,36 | 6084,36 | 5,310 | 6122,07 |
| 0 | 0 | 0 | 0,002 | 1,80 |
| | 0 | 0 | 0,003 | 3,60 |
| Ap chamber= f(Ap nozz) | | | | |
| Pitot read (patm-APsC) | read(Pa) | Read mean value(Pa) | transducer(V) | Ap nozz (Pa) |
| 42 | 808,68 | 808,68 | 0,690 | 795,18 |
| 100 | 1925,43 | 1925,43 | 1,654 | 1906,68 |
| 203 | 3908,62 | 3908,62 | 3,415 | 3937,12 |
| 303 | 5834,05 | 5834,05 | 5,034 | 5803,84 |
| 0 | 0 | 0 | 0,002 | 1,80 |
| 0 | 0 | 0 | 0,003 | 3,60 |

Table B.9 Alpha constants obtention data

### Uncertainties

| Resol.DP (Pa) | Resol T (˚C) | Resol P (Pa) | a1 | a2 | Rair (J/Kg·K) | Mair kg/mol | Rgas (J/mol·K) |
|---|---|---|---|---|---|---|---|
| 9,8 | 0,1125 | 39,10 | 0,9933 | 1,0018 | 287,06 | 0,02896 | 8,31 |

Table B.10 Uncertainties: Data used

| T(k) | P (Pa) | Dif P(Pa) | V(m/s) | v/Dp | v/T | V/P | total (m/s) |
|---|---|---|---|---|---|---|---|
| 293,15 | 99000 | 343,16 | 24,11 | 0,035 | 0,041 | 0,00012 | 0,344 |
| 293,15 | 99000 | 653,94 | 82,37 | 0,011 | 0,140 | 0,00043 | 0,111 |
| 293,15 | 99000 | 1074,96 | 71,04 | 0,013 | 0,121 | 0,00037 | 0,125 |
| 293,15 | 99000 | 2900,84 | 33,34 | 0,026 | 0,057 | 0,00017 | 0,251 |
| 293,15 | 99000 | 3861,00 | 42,84 | 0,020 | 0,073 | 0,00022 | 0,197 |
| 293,15 | 97000 | 343,16 | 24,36 | 0,036 | 0,042 | 0,00013 | 0,348 |
| 293,15 | 97000 | 653,94 | 33,69 | 0,026 | 0,057 | 0,00017 | 0,253 |
| 293,15 | 97000 | 1074,96 | 43,28 | 0,020 | 0,074 | 0,00023 | 0,199 |
| 293,15 | 97000 | 2900,84 | 71,79 | 0,013 | 0,122 | 0,00038 | 0,126 |
| 293,15 | 97000 | 3861,00 | 83,25 | 0,011 | 0,142 | 0,00045 | 0,112 |
| | | | | | | | |
| 303,15 | 99000 | 343,16 | 24,52 | 0,036 | 0,040 | 0,00012 | 0,350 |
| 303,15 | 99000 | 653,94 | 33,90 | 0,026 | 0,056 | 0,00017 | 0,255 |
| 303,15 | 99000 | 1074,96 | 43,56 | 0,020 | 0,072 | 0,00022 | 0,200 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 303,15 | 99000 | 2900,84 | 72,24 | 0,013 | 0,119 | 0,00038 | 0,127 |
| 303,15 | 99000 | 3861,00 | 83,76 | 0,011 | 0,138 | 0,00044 | 0,113 |
| 303,15 | 97000 | 343,16 | 24,77 | 0,036 | 0,041 | 0,00013 | 0,354 |
| 303,15 | 97000 | 653,94 | 34,25 | 0,026 | 0,056 | 0,00018 | 0,258 |
| 303,15 | 97000 | 1074,96 | 44,02 | 0,021 | 0,073 | 0,00023 | 0,203 |
| 303,15 | 97000 | 2900,84 | 73,00 | 0,013 | 0,120 | 0,00039 | 0,128 |
| 303,15 | 97000 | 3861,00 | 84,66 | 0,011 | 0,140 | 0,00045 | 0,114 |

Table B.11 Uncertainties: Calculus at different air properties

# C. Techincal documents and datasheets

**Altivar 31 Installation Manual**

# Altivar 31H

Installation manual

Variable speed drives
for asynchronous motors


a brand of Schneider Electric

Telemecanique

# Contents

Drive references ____ 2
Mounting ____ 4
Wiring ____ 8

**NOTE: Please also refer to the Programming Manual.**

**When the drive is powered up, the power components and some of the control components are connected to the line supply. It is extremely dangerous to touch them. *The drive cover must be kept closed.***

**In general, the drive power supply must be disconnected before any operation on either the electrical or mechanical parts of the installation or *machine.***
**After the ALTIVAR has been switched off and the display has disappeared completely, *wait for 10 minutes before working on the equipment*. This is the time required for the capacitors to discharge.**
**The motor can be stopped during operation by inhibiting start commands or the speed reference while the drive remains powered up. If personnel safety requires prevention of sudden restarts, this electronic locking system is not sufficient: *fit a cut-off on the power circuit.*.**

**The drive is fitted with safety devices which, in the event of a fault, can shut down the drive and consequently the motor. The motor itself may be stopped by a mechanical blockage. Finally, voltage variations, especially line supply failures, can also cause shutdowns.**
**If the cause of the shutdown disappears, there is a risk of restarting which may endanger certain machines or installations, especially those which must conform to safety regulations.**

***In this case the user must take precautions against the possibility of restarts, in particular by using a low speed detector to cut off power to the drive if the motor performs an unprogrammed shutdown.***

**The drive must be installed and set up in accordance with both international and national standards. Bringing the device into conformity is the responsibility of the systems integrator who must observe the EMC directive among others within the European Union.**
**The specifications contained in this document must be applied in order to comply with the essential requirements of the EMC directive.**

**The Altivar 31 must be considered as a component: it is neither a machine nor a device ready for use in accordance with European directives (machinery directive and electromagnetic compatibility directive). It is the responsibility of the end user to ensure that the machine meets these standards.**

**The drive must not be used as a safety device for machines posing a potential risk of material damage or personal injury (lifting equipment, for example). In such applications, overspeed checks and checks to ensure that the trajectory remains under constant control must be made by separate devices which are independent of the drive.**

**The products and equipment described in this document may be changed or modified at any time, either from a technical point of view or in the way they are operated. Their description can in no way be considered contractual.**

# Drive references

## Single phase supply voltage: 200…240 V 50/60 Hz

3-phase motor 200…240 V

| Motor | Line supply (input) | | | | | Drive (output) | | | Altivar 31 |
|---|---|---|---|---|---|---|---|---|---|
| Power indicated on plate (1) | Max. line current (2) | | Max. prospective line Isc | Apparent power | Max. inrush current (3) | Nominal current In (1) | Max. transient current (1) (4) | Power dissipated at nominal load | Reference (5) |
| | at 200 V | at 240 V | | | | | | | |
| kW/HP | A | A | kA | kVA | A | A | A | W | |
| 0.18/0.25 | 3.0 | 2.5 | 1 | 0.6 | 10 | 1.5 | 2.3 | 24 | **ATV31H018M2** |
| 0.37/0.5 | 5.3 | 4.4 | 1 | 1.0 | 10 | 3.3 | 5.0 | 41 | **ATV31H037M2** |
| 0.55/0.75 | 6.8 | 5.8 | 1 | 1.4 | 10 | 3.7 | 5.6 | 46 | **ATV31H055M2** |
| 0.75/1 | 8.9 | 7.5 | 1 | 1.8 | 10 | 4.8/4.2 (6) | 7.2 | 60 | **ATV31H075M2** |
| 1.1/1.5 | 12.1 | 10.2 | 1 | 2.4 | 19 | 6.9 | 10.4 | 74 | **ATV31HU11M2** |
| 1.5/2 | 15.8 | 13.3 | 1 | 3.2 | 19 | 8.0 | 12.0 | 90 | **ATV31HU15M2** |
| 2.2/3 | 21.9 | 18.4 | 1 | 4.4 | 19 | 11.0 | 16.5 | 123 | **ATV31HU22M2** |

## 3-phase supply voltage: 200…240 V 50/60 Hz

3-phase motor 200…240 V

| Motor | Line supply (input) | | | | | Drive (output) | | | Altivar 31 |
|---|---|---|---|---|---|---|---|---|---|
| Power indicated on plate (1) | Max. line current (2) | | Max. prospective line Isc | Apparent power | Max. inrush current (3) | Nominal current In (1) | Max. transient current (1) (4) | Power dissipated at nominal load | Reference (5) |
| | at 200 V | at 240 V | | | | | | | |
| kW/HP | A | A | kA | kVA | A | A | A | W | |
| 0.18/0.25 | 2.1 | 1.9 | 5 | 0.7 | 10 | 1.5 | 2.3 | 23 | **ATV31H018M3X** |
| 0.37/0.5 | 3.8 | 3.3 | 5 | 1.3 | 10 | 3.3 | 5.0 | 38 | **ATV31H037M3X** |
| 0.55/0.75 | 4.9 | 4.2 | 5 | 1.7 | 10 | 3.7 | 5.6 | 43 | **ATV31H055M3X** |
| 0.75/1 | 6.4 | 5.6 | 5 | 2.2 | 10 | 4.8 | 7.2 | 55 | **ATV31H075M3X** |
| 1.1/1.5 | 8.5 | 7.4 | 5 | 3.0 | 10 | 6.9 | 10.4 | 71 | **ATV31HU11M3X** |
| 1.5/2 | 11.1 | 9.6 | 5 | 3.8 | 10 | 8.0 | 12.0 | 86 | **ATV31HU15M3X** |
| 2.2/3 | 14.9 | 13.0 | 5 | 5.2 | 10 | 11.0 | 16.5 | 114 | **ATV31HU22M3X** |
| 3/3 | 19.1 | 16.6 | 5 | 6.6 | 19 | 13.7 | 20.6 | 146 | **ATV31HU30M3X** |
| 4/5 | 24 | 21.1 | 5 | 8.4 | 19 | 17.5 | 26.3 | 180 | **ATV31HU40M3X** |
| 5.5/7.5 | 36.8 | 32.0 | 22 | 12.8 | 23 | 27.5 | 41.3 | 292 | **ATV31HU55M3X** |
| 7.5/10 | 46.8 | 40.9 | 22 | 16.2 | 23 | 33.0 | 49.5 | 388 | **ATV31HU75M3X** |
| 11/15 | 63.5 | 55.6 | 22 | 22.0 | 93 | 54.0 | 81.0 | 477 | **ATV31HD11M3X** |
| 15/20 | 82.1 | 71.9 | 22 | 28.5 | 93 | 66.0 | 99.0 | 628 | **ATV31HD15M3X** |

(1) These power ratings and currents are for a maximum ambient temperature of 50°C and a switching frequency of 4 kHz in continuous operation.The switching frequency is adjustable from 2 to 16 kHz.
Above 4 kHz, the drive will reduce the switching frequency in the event of excessive temperature rise. The temperature rise is controlled by a PTC probe in the power module. Nonetheless, the nominal drive current should be derated if operation above 4 kHz needs to be continuous.
Derating curves are shown on page 6 as a function of switching frequency, ambient temperature and mounting conditions.

(2) Current on a line supply with the "Max. prospective line Isc" indicated.

(3) Peak current on power-up, for the max. voltage (240 V + 10%).

(4) For 60 seconds.

(5) Reference for a drive with built-in terminal but no control unit. For a drive with control potentiometer and RUN/STOP buttons, add an A at the end of the reference, e.g.: ATV31H018M2A.

(6) 4.8 A at 200 V/4.6 A at 208 V/4.2 A at 230 V and 240 V.

# Drive references

## 3-phase supply voltage: 380…500 V 50/60 Hz

3-phase motor 380…500 V

| Motor | Line supply (input) | | | | | Drive (output) | | | Altivar 31 |
|---|---|---|---|---|---|---|---|---|---|
| Power indicated on plate (1) | Max. line current (2) | | Max. prospective line Isc | Apparent power | Max. inrush current (3) | Nominal current In (1) | Max. transient current (1) (4) | Power dissipated at nominal load | Reference (5) |
| | at 380 V | at 500 V | | | | | | | |
| kW/HP | A | A | kA | kVA | A | A | A | W | |
| 0.37/0.5 | 2.2 | 1.7 | 5 | 1.5 | 10 | 1.5 | 2.3 | 32 | **ATV31H037N4** |
| 0.55/0.75 | 2.8 | 2.2 | 5 | 1.8 | 10 | 1.9 | 2.9 | 37 | **ATV31H055N4** |
| 0.75/1 | 3.6 | 2.7 | 5 | 2.4 | 10 | 2.3 | 3.5 | 41 | **ATV31H075N4** |
| 1.1/1.5 | 4.9 | 3.7 | 5 | 3.2 | 10 | 3.0 | 4.5 | 48 | **ATV31HU11N4** |
| 1.5/2 | 6.4 | 4.8 | 5 | 4.2 | 10 | 4.1 | 6.2 | 61 | **ATV31HU15N4** |
| 2.2/3 | 8.9 | 6.7 | 5 | 5.9 | 10 | 5.5 | 8.3 | 79 | **ATV31HU22N4** |
| 3/3 | 10.9 | 8.3 | 5 | 7.1 | 10 | 7.1 | 10.7 | 125 | **ATV31HU30N4** |
| 4/5 | 13.9 | 10.6 | 5 | 9.2 | 10 | 9.5 | 14.3 | 150 | **ATV31HU40N4** |
| 5.5/7.5 | 21.9 | 16.5 | 22 | 15.0 | 30 | 14.3 | 21.5 | 232 | **ATV31HU55N4** |
| 7.5/10 | 27.7 | 21.0 | 22 | 18.0 | 30 | 17.0 | 25.5 | 269 | **ATV31HU75N4** |
| 11/15 | 37.2 | 28.4 | 22 | 25.0 | 97 | 27.7 | 41.6 | 397 | **ATV31HD11N4** |
| 15/20 | 48.2 | 36.8 | 22 | 32.0 | 97 | 33.0 | 49.5 | 492 | **ATV31HD15N4** |

## 3-phase supply voltage: 525…600 V 50/60 Hz

3-phase motor 525…600 V

| Motor | Line supply (input) | | | | | Drive (output) | | | Altivar 31 |
|---|---|---|---|---|---|---|---|---|---|
| Power indicated on plate (1) | Max. line current (2) | | Max. prospective line Isc | Apparent power | Max. inrush current (3) | Nominal current In (1) | Max. transient current (1) (4) | Power dissipated at nominal load | Reference |
| | at 525 V | at 600 V | | | | | | | |
| kW/HP | A | A | kA | kVA | A | A | A | W | |
| 0.75/1 | 2.8 | 2.4 | 5 | 2.5 | 12 | 1.7 | 2.6 | 36 | **ATV31H075S6X** |
| 1.5/2 | 4.8 | 4.2 | 5 | 4.4 | 12 | 2.7 | 4.1 | 48 | **ATV31HU15S6X** |
| 2.2/3 | 6.4 | 5.6 | 5 | 5.8 | 12 | 3.9 | 5.9 | 62 | **ATV31HU22S6X** |
| 4/5 | 10.7 | 9.3 | 5 | 9.7 | 12 | 6.1 | 9.2 | 94 | **ATV31HU40S6X** |
| 5.5/7.5 | 16.2 | 14.1 | 22 | 15.0 | 36 | 9.0 | 13.5 | 133 | **ATV31HU55S6X** |
| 7.5/10 | 21.3 | 18.5 | 22 | 19.0 | 36 | 11.0 | 16.5 | 165 | **ATV31HU75S6X** |
| 11/15 | 27.8 | 24.4 | 22 | 25.0 | 117 | 17.0 | 25.5 | 257 | **ATV31HD11S6X** |
| 15/20 | 36.4 | 31.8 | 22 | 33.0 | 117 | 22.0 | 33.0 | 335 | **ATV31HD15S6X** |

(1) These power ratings and currents are for a maximum ambient temperature of 50°C and a switching frequency of 4 kHz in continuous operation. The switching frequency is adjustable from 2 to 16 kHz.
Above 4 kHz, the drive will reduce the switching frequency in the event of excessive temperature rise. The temperature rise is controlled by a PTC probe in the power module. Nonetheless, the nominal drive current should be derated if operation above 4 kHz needs to be continuous.
Derating curves are shown on page 6 as a function of switching frequency, ambient temperature and mounting conditions.

(2) Current on a line supply with the "Max. prospective line Isc" indicated.

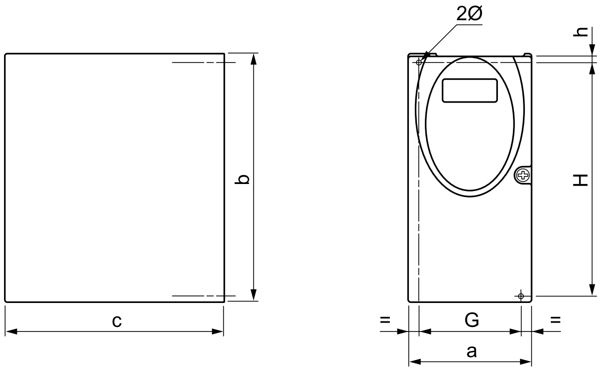
(3) Peak current on power-up, for the max. voltage (500 V + 10%, 600 V + 10%).

(4) For 60 seconds.

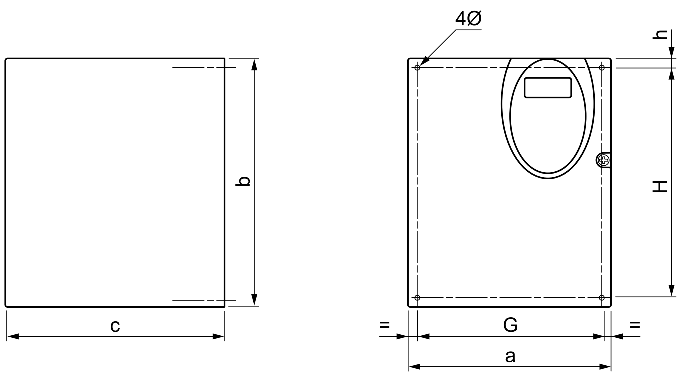
(5) Reference for a drive with built-in terminal but no control unit. For a drive with control potentiometer and RUN/STOP buttons, add an A at the end of the reference, e.g.: ATV31H037N4A.

# Mounting

## Dimensions and weights



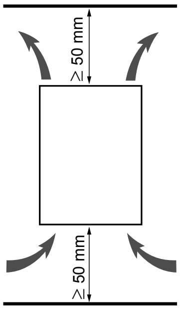

| ATV31 | | a<br>mm | b<br>mm | c (1)<br>mm | G<br>mm | hr<br>mm | H<br>mm | Ø<br>mm | For<br>screw | Weight<br>kg |
|---|---|---|---|---|---|---|---|---|---|---|
| H018M3X, H037M3X | Size 1 | 72 | 145 | 120 | 60±1 | 5 | 121.5±1 | 2 x 5 | M4 | 0.9 |
| H055M3X, H075M3X | Size 2 | 72 | 145 | 130 | 60±1 | 5 | 121.5±1 | 2 x 5 | M4 | 0.9 |
| H018M2, H037M2 | Size 3 | 72 | 145 | 130 | 60±1 | 5 | 121.5±1 | 2 x 5 | M4 | 1.05 |
| H055M2, H075M2 | Size 4 | 72 | 145 | 140 | 60±1 | 5 | 121.5±1 | 2 x 5 | M4 | 1.05 |
| HU11M3X, HU15M3X | Size 5 | 105 | 143 | 130 | 93±1 | 5 | 121.5±1 | 2 x 5 | M4 | 1.25 |
| HU11M2, HU15M2,<br>HU22M3X,<br>H037N4, H055N4, H075N4,<br>HU11N4,HU15N4,<br>H075S6X, HU15S6X | Size 6 | 105 | 143 | 150 | 93±1 | 5 | 121.5±1 | 2 x 5 | M4 | 1.35 |




| ATV31 | | a<br>mm | b<br>mm | c (1)<br>mm | G<br>mm | hr<br>mm | H<br>mm | Ø<br>mm | For<br>screw | Weight<br>kg |
|---|---|---|---|---|---|---|---|---|---|---|
| HU22M2, HU30M3X, HU40M3X,<br>HU22N4, HU30N4, HU40N4,<br>HU22S6X, HU40S6X | Size 7 | 140 | 184 | 150 | 126±1 | 6.5 | 157±1 | 4 x 5 | M4 | 2.35 |
| HU55M3X, HU75M3X,<br>HU55N4, HU75N4,<br>HU55S6X, HU75S6X | Size 8 | 180 | 232 | 170 | 160±1 | 5 | 210±1 | 4 x 5 | M4 | 4.70 |
| HD11M3X, HD15M3X,<br>HD11N4, HD15N4,<br>HD11S6X, HD15S6X | Size 9 | 245 | 330 | 190 | 225±1 | 7 | 295±1 | 4 x 6 | M5 | 9.0 |

(1) For drives in the A range, add 8 mm for the protruding potentiometer button.

## Mounting and temperature conditions

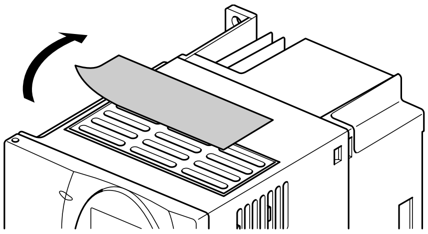



Install the unit vertically, at ± 10°.
Do not place it close to heating elements.
Leave sufficient free space to ensure that the air required for cooling purposes can circulate from the bottom to the top of the unit.

Free space in front of unit: 10 mm minimum.

When IP20 protection is adequate, we recommend that the protective cover on the top of the drive be removed, as shown below.

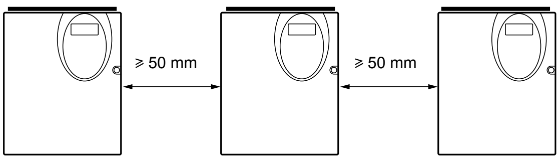
### Removing the protective cover



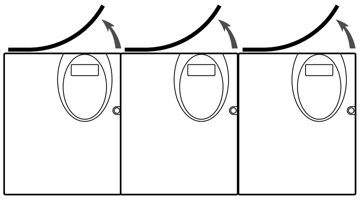
Example ATV31HU11M3X

### 3 types of mounting are possible:

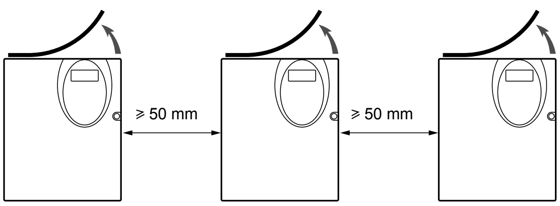
**Type A mounting:** Free space ⩾ 50 mm on each side, with protective cover fitted




**Type B mounting:** Drives mounted side-by-side, protective cover removed (the degree of protection becomes IP20)

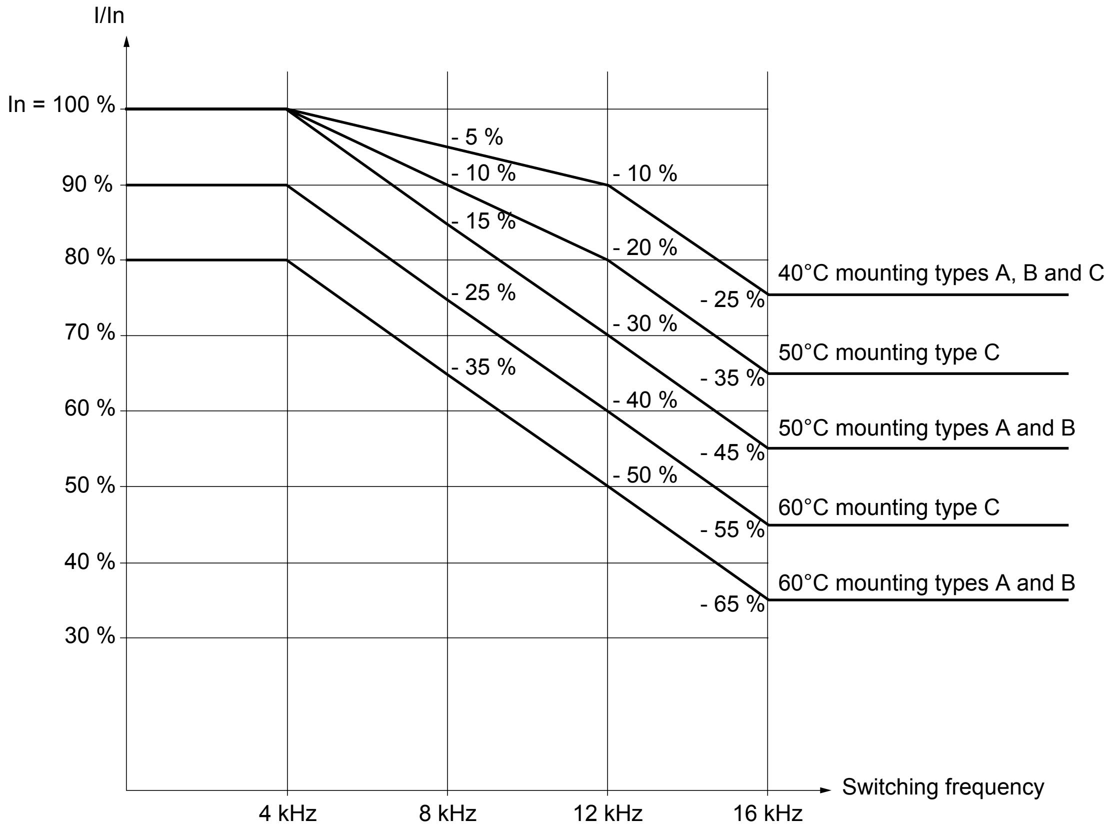


**Type C mounting:** Free space ⩾ 50 mm on each side, protective cover removed (the degree of protection becomes IP20)



# Mounting

Derating curves for the drive current In as a function of the temperature, switching frequency and type of mounting.




For intermediate temperatures (e.g. 55°C), interpolate between 2 curves.

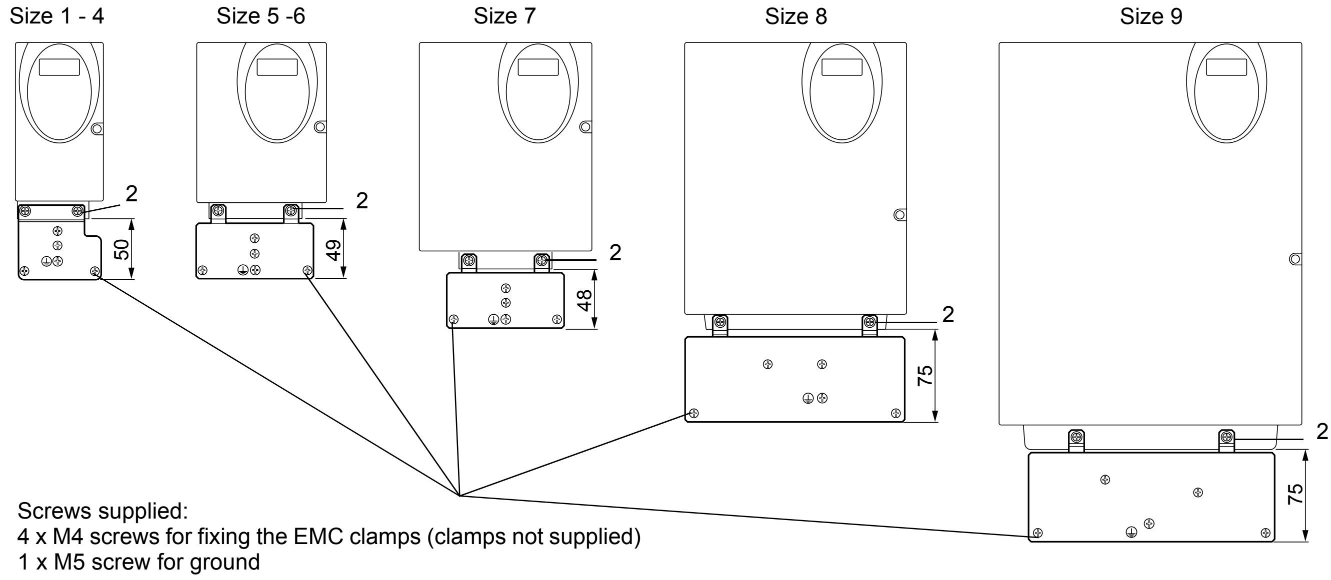
**If you are installing the drives in enclosures, make provision for a flow of air at least equal to the value given in the table below for each drive.**

| ATV31 | Flow rate in $m^3$/hour |
|---|---|
| H018M2, H037M2, H055M2,<br>H018M3X, H037M3X, H055M3X,<br>H037N4, H055N4, H075N4, HU11N4<br>H075S6X, HU15S6X | 18 |
| H075M2, HU11M2, HU15M2<br>H075M3X, HU11M3X, HU15M3X<br>HU15N4, HU22N4<br>HU22S6X, HU40S6X | 33 |
| HU22M2,<br>HU22M3X, HU30M3X, HU40M3X<br>HU30N4, HU40N4<br>HU55S6X, HU75S6X | 93 |
| HU55M3X<br>HU55N4, HU75N4<br>HD11S6X | 102 |
| HU75M3X, HD11M3X,<br>HD11N4, HD15N4<br>HD15S6X | 168 |
| HD15M3X | 216 |

## Electromagnetic compatibility

### EMC mounting plate: Supplied with the drive

Fix the EMC equipotentiality mounting plate to the holes in the ATV 31 heatsink using the 2 screws supplied, as shown in the drawings below.

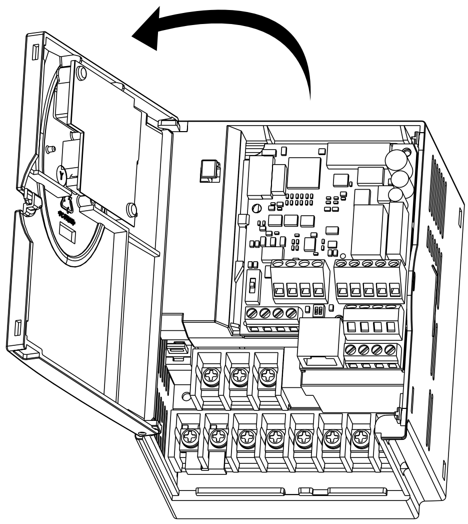



| ATV31 | |
|---|---|
| H018M3X, H037M3X | Size 1 |
| H055M3X, H075M3X | Size 2 |
| H018M2, H037M2 | Size 3 |
| H055M2, H075M2 | Size 4 |
| HU11M3X, HU15M3X | Size 5 |
| HU11M2, HU15M2, HU22M3X,<br>H037N4, H055N4, H075N4, HU11N4, HU15N4,<br>H075S6X, HU15S6X | Size 6 |

| ATV31 | |
|---|---|
| HU22M2, HU30M3X, HU40M3X,<br>HU22N4, HU30N4, HU40N4,<br>HU22S6X, HU40S6X | Size 7 |
| HU55M3X, HU75M3X,<br>HU55N4, HU75N4,<br>HU55S6X, HU75S6X | Size 8 |
| HD11M3X, HD15M3X,<br>HD11N4, HD15N4,<br>HD11S6X, HD15S6X | Size 9 |

## Access to terminals

To access the terminals, open the cover as shown in the example below.



Example ATV31HU11M2

## Power terminals

⚠ **Connect the power terminals before connecting the control terminals.**

### Power terminal characteristics

| Altivar ATV 31 | Maximum connection capacity | | Tightening torque in Nm |
|---|---|---|---|
| | AWG | $mm^2$ | |
| H018M2, H037M2, H055M2, H075M2,<br>H018M3X, H037M3X, H055M3X, H075M3X, HU11M3X, HU15M3X | AWG 14 | 2.5 | 0.8 |
| HU11M2, HU15M2, HU22M2,<br>HU22M3X, HU30M3X, HU40M3X,<br>H037N4, H055N4, H075N4, HU11N4,HU15N4, HU22N4, HU30N4, HU40N4<br>H075S6X, HU15S6X, HU22S6X, HU40S6X | AWG 10 | 6 | 1.2 |
| HU55M3X, HU75M3X,<br>HU55N4, HU75N4,<br>HU55S6X, HU75S6X | AWG 6 | 16 | 2.5 |
| HD11M3X, HD15M3X,<br>HD11N4, HD15N4,<br>HD11S6X, HD15S6X | AWG 3 | 25 | 4.5 |

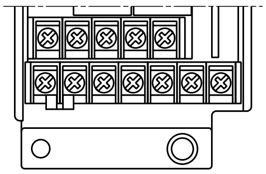
### Power terminal functions

| Terminal | Function | For Altivar ATV 31 |
|---|---|---|
| ⏚ | Ground terminal | All ratings |
| R/L1<br>S/L2 | Power supply | ATV31●●●●M2 |
| R/L1<br>S/L2<br>T/L3 | | ATV31●●●●M3X<br>ATV31●●●●N4<br>ATV31●●●●S6X |
| PO | DC bus + polarity | All ratings |
| PA/+ | Output to braking resistor (+ polarity) | All ratings |
| PB | Output to braking resistor | All ratings |
| PC/- | DC bus - polarity | All ratings |
| U/T1<br>V/T2<br>W/T3 | Outputs to the motor | All ratings |

⚠ **Never remove the commoning link between PO and PA/+. The PO and PA/+ terminal screws must always be fully tightened as a high current flows through the commoning link.**

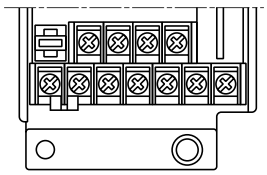
## Arrangement of the power terminals

ATV 31H018M3X, H037M3X, H055M3X, H075M3X



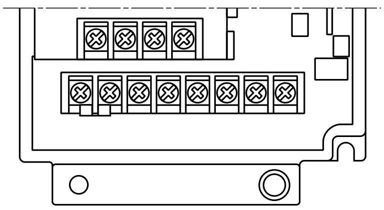
| | | | | | | |
|---|---|---|---|---|---|---|
| ⏚ | ⏚ | R/L1 | S/L2 | T/L3 | | |
| P0 | PA/+ | PB | PC/- | U/T1 | V/T2 | W/T3 |

ATV 31H018M2, H037M2, H055M2, H075M2



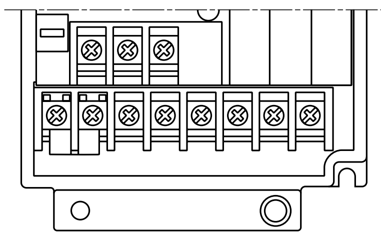
| | | | | | | |
|---|---|---|---|---|---|---|
| ⏚ | ⏚ | R/L1 | S/L2 | | | |
| P0 | PA/+ | PB | PC/- | U/T1 | V/T2 | W/T3 |

ATV 31HU11M3X, HU15M3X, HU22M3X, HU30M3X, HU40M3X, H037N4, H055N4, H075N4, HU11N4, HU15N4, HU22N4, HU30N4, HU40N4, H075S6X, HU15S6X, HU22S6X, HU40S6X



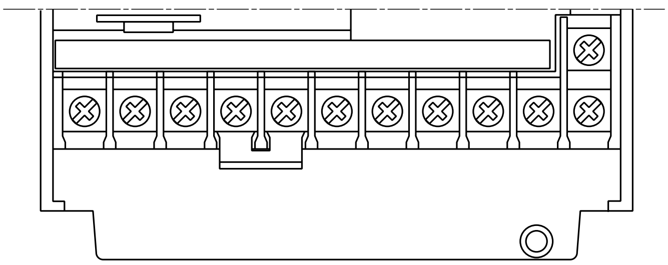
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ⏚ | R/L1 | S/L2 | T/L3 | | | | |
| P0 | PA/+ | PB | PC/- | U/T1 | V/T2 | W/T3 | ⏚ |

ATV 31HU11M2, HU15M2, HU22M2



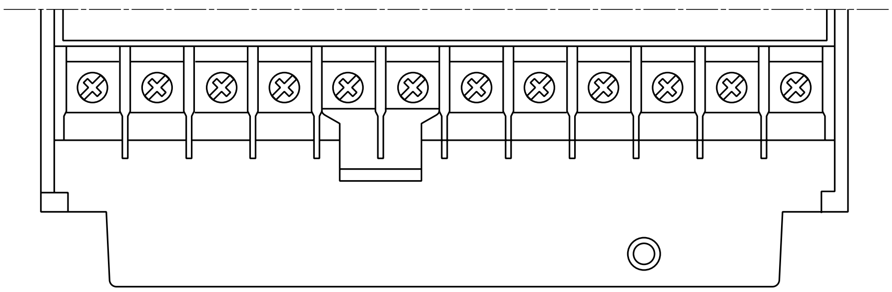
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ⏚ | R/L1 | S/L2 | | | | | |
| P0 | PA/+ | PB | PC/- | U/T1 | V/T2 | W/T3 | ⏚ |

ATV 31HU55M3X, HU75M3X, HU55N4, HU75N4, HU55S6X, HU75S6X



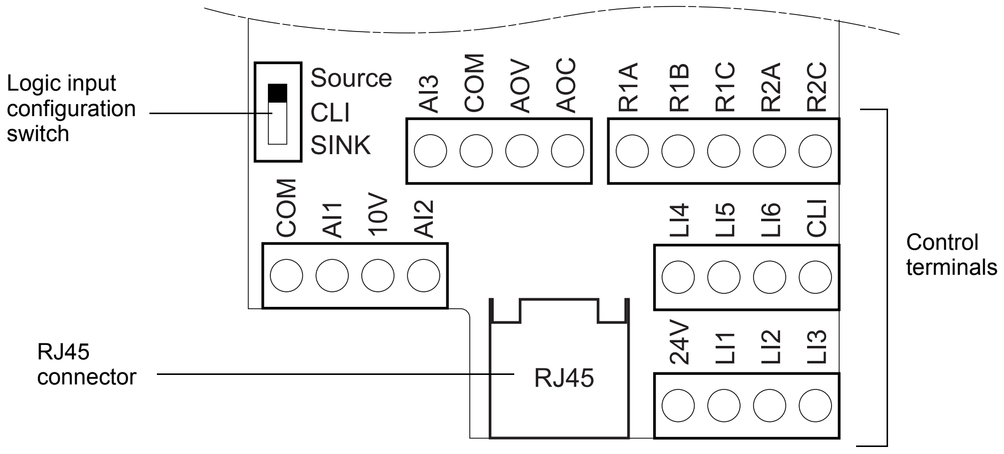
| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | ⏚ |
| R/L1 | S/L2 | T/L3 | P0 | PA/+ | PB | PC/- | U/T1 | V/T2 | W/T3 | ⏚ |

ATV 31HD11M3X, HD15M3X, HD11N4, HD15N4, HD11S6X, HD15S6X



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ⏚ | R/L1 | S/L2 | T/L3 | P0 | PA/+ | PB | PC/- | U/T1 | V/T2 | W/T3 | ⏚ |

## Control terminals




- Maximum connection capacity: 2.5 mm$^2$ - AWG 14
- Max. tightening torque: 0.6 Nm

## Control terminals

### Arrangement, characteristics and functions of the control terminals

| Terminal | Function | Electrical characteristics |
|---|---|---|
| R1A<br>R1B<br>R1C | Common point C/O contact (R1C) of programmable relay R1 | • Min. switching capacity: 10 mA for 5 V ⎓<br>• Max. switching capacity on resistive load (cos φ = 1 and L/R = 0 ms):<br>5 A for 250 V ∿ and 30 V ⎓<br>• Max. switching capacity on inductive load (cos φ = 0.4 and L/R = 7 ms):<br>1.5 A for 250 V ∿ and 30 V ⎓<br>• Sampling time 8 ms<br>• Service life: 100,000 operations at max. switching power<br>1,000,000 operations at min. switching power |
| R2A<br>R2C | N/O contact of programmable relay R2 | |

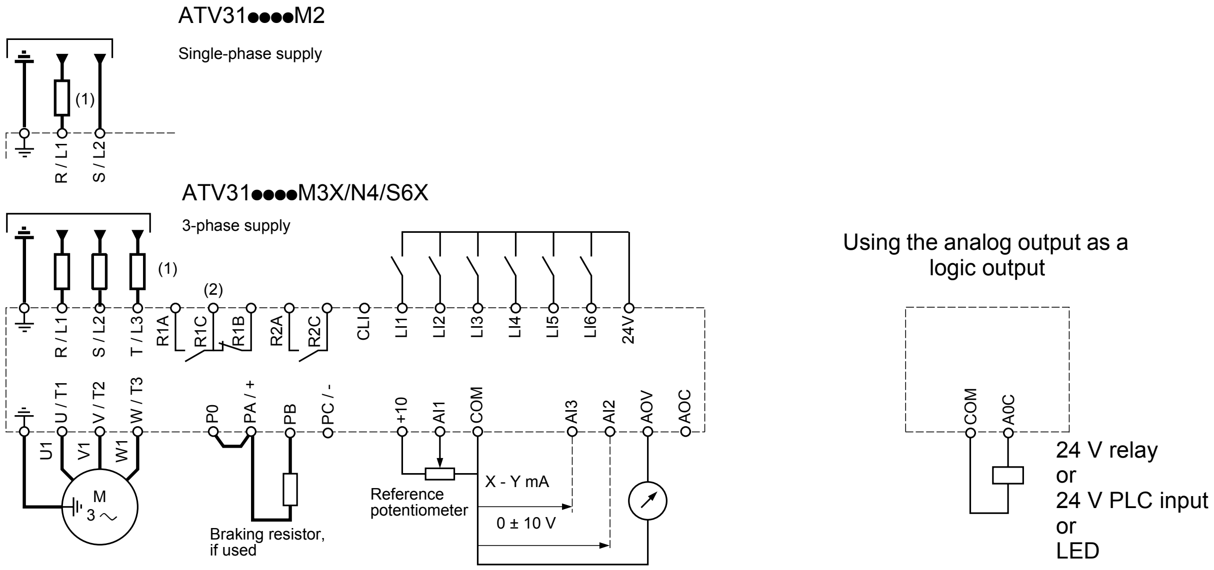
| | | |
|---|---|---|
| COM | Analog I/O common | 0 V |
| AI1 | Analog voltage input | Analog input 0 + 10 V (max. safe voltage 30 V)<br>• Impedance 30 kΩ<br>• Resolution 0.01 V, 10-bit converter<br>• Precision ± 4.3%, linearity ± 0.2%, of max. value<br>• Sampling time 8 ms<br>• Operation with shielded cable 100 m max. |
| 10 V | Power supply for setpoint potentiometer<br>1 to 10 kΩ | +10 V (+ 8% - 0), 10 mA max, protected against short-circuits and overloads |
| AI2 | Analog voltage input | Bipolar analog input 0 ± 10 V (max. safe voltage ± 30 V)<br>**The + or - polarity of the voltage on AI2 affects the direction of the setpoint and therefore the direction of operation.**<br>• Impedance 30 kΩ<br>• Resolution 0.01 V, 10-bit + sign converter<br>• Precision ± 4.3%, linearity ± 0.2%, of max. value<br>• Sampling time 8 ms<br>• Operation with shielded cable 100 m max. |

| | | |
|---|---|---|
| AI3 | Analog current input | Analog input X - Y mA. X and Y can be programmed from 0 to 20 mA<br>• Impedance 250 Ω<br>• Resolution 0.02 mA, 10-bit converter<br>• Precision ± 4.3%, linearity ± 0.2%, of max. value<br>• Sampling time 8 ms |
| COM | Analog I/O common | 0 V |
| AOV<br><br>AOC | Analog voltage output AOV<br>or<br>Analog current output AOC<br>or<br>Logic voltage output AOC<br>AOV or AOC can be assigned (either, but not both) | Analog output 0 to 10 V, min. load impedance 470 Ω<br>or<br>Analog output X - Y mA. X and Y can be programmed from 0 to 20 mA, max. load impedance 800 Ω<br>• Resolution 8 bits (1)<br>• Precision ± 1% (1)<br>• Linearity ± 0.2% (1)<br>• Sampling time 8 ms<br>This analog output can be configured as a 24 V logic output on AOC, min. load impedance 1.2 kΩ.<br>(1) Characteristics of digital/analog converter. |

| | | |
|---|---|---|
| 24 V | Logic input power supply | + 24 V protected against short-circuits and overloads, min. 19 V, max. 30 V<br>Max. customer current available 100 mA |
| LI1<br>LI2<br>LI3 | Logic inputs | Programmable logic inputs<br>• + 24 V power supply (max. 30 V)<br>• Impedance 3.5 kΩ<br>• State 0 if < 5 V, state 1 if > 11 V (voltage difference between LI- and CLI)<br>• Sampling time 4 ms |

| | | |
|---|---|---|
| LI4<br>LI5<br>LI6 | Logic inputs | Programmable logic inputs<br>• + 24 V power supply (max. 30 V)<br>• Impedance 3.5 kΩ<br>• State 0 if < 5 V, state 1 if > 11 V (voltage difference between LI- and CLI)<br>• Sampling time 4 ms |
| CLI | Logic input common | See page 12. |

## Wiring diagram for factory settings



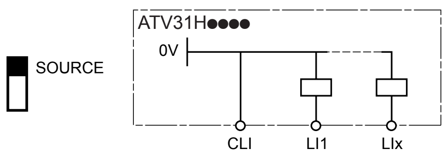

(1) Line choke, if used (single phase or 3-phase)
(2) Fault relay contacts, for remote indication of the drive status.

**Note:** Fit interference suppressors to all inductive circuits near the drive or coupled to the same circuit (relays, contactors, solenoid valves, etc).

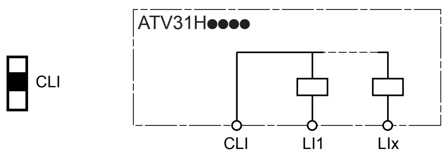
**Choice of associated components:**
Please refer to the catalogue.

## Logic input switch

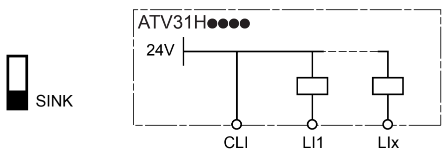
This switch assigns the logic input common link to 0V, 24 V or "floating":




CLI at 0 V (factory setting)




CLI "floating"

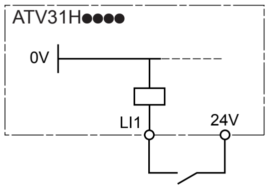



CLI at 24 V

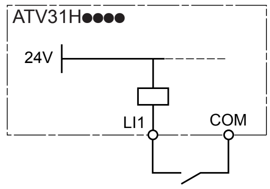
## Examples of recommended circuit diagrams

### Using volt-free contacts

- Switch in "Source" position
  (ATV31 factory setting for types other than ATV31●●●●A)



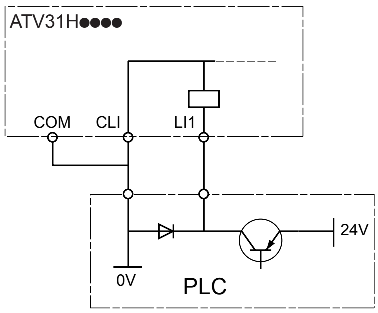

- Switch in "SINK" position
  (factory setting for ATV31●●●●A)



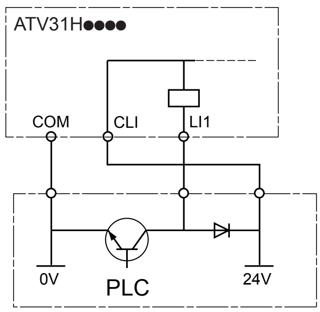

In this instance, the common must never be connected to earth or earth ground, as this presents a risk of unintended equipment operation on the first insulation fault.

### Using PLC transistor outputs

- Switch in CLI position




- Switch in CLI position




## Wiring recommendations

### Power

The drive must be earthed to conform with the regulations concerning high leakage currents (over 3.5 mA).
When upstream protection by means of a "residual current device" is required by the installation standards, a type A device should be used for single-phase drives and type B for 3-phase drives. Choose a suitable model incorporating:

- HF current filtering
- A time delay which prevents tripping caused by the load from stray capacitance on power-up. The time delay is not possible for 30 mA devices. In this case, choose devices with immunity against accidental tripping, for example RCDs with reinforced immunity from the **s.i** range (Merlin Gerin brand).

If the installation includes several drives, provide one "residual current device" per drive.

Keep the power cables separate from circuits in the installation with low-level signals (detectors, PLCs, measuring apparatus, video, telephone).

If you are using cables > 50 m between the drive and the motor, add output filters (please refer to the catalogue).

### Control

Keep the control circuits away from the power cables. For control and speed reference circuits, we recommend using shielded twisted cables with a pitch of between 25 and 50 mm, connecting the shielding to ground at each end.

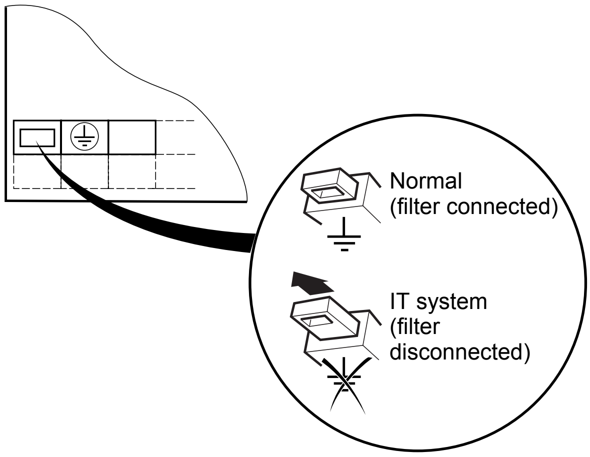
## Operation on an IT system

IT system: Isolated or impedance earthed neutral.
Use a permanent insulation monitor compatible with non-linear loads (a Merlin Gerin type XM200, for example).

ATV 31●●●M2 and N4 drives feature built-in RFI filters. These filters can be isolated from ground for operation on an IT system as follows:

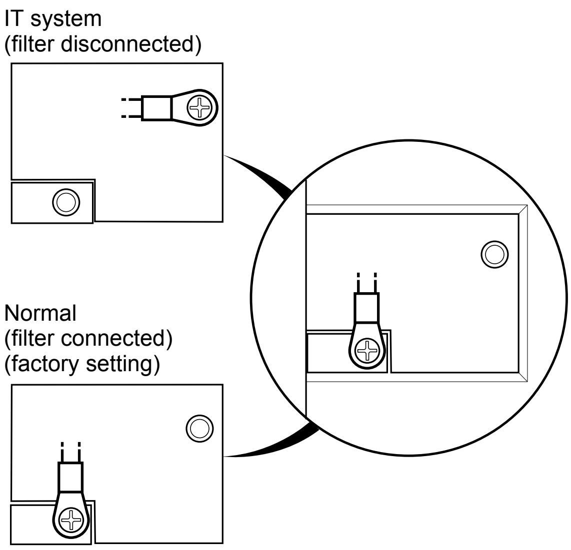
**ATV31H018M2 to U22M2 and ATV31H037N4 to U40N4:**

Pull out the jumper on the left of the ground terminal as illustrated below.




**ATV31HU55N4 to D15N4:**

Move the cable tag on the top left of the power terminals as illustrated below (example ATV31HU55N4):



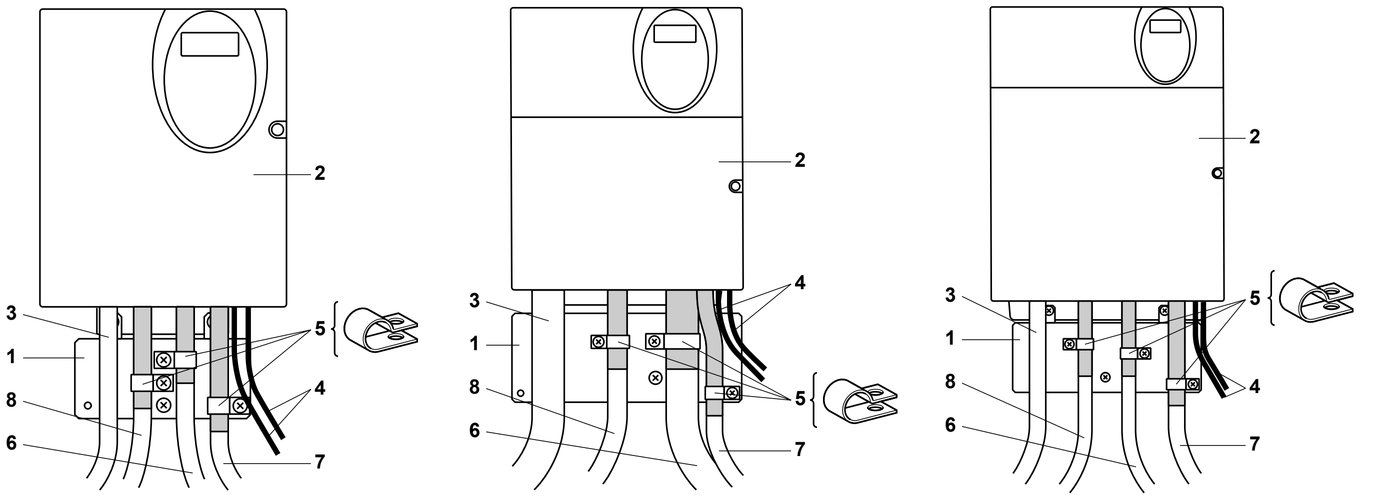
## Electromagnetic compatibility

### Principle

- Grounds between the drive, motor and cable shielding must have "high frequency" equipotentiality.
- Use shielded cables with shielding connected to ground throughout 360° at both ends for the motor cable **6**, braking resistor (if used) **8**, and control-signalling cables **7**. Metal ducting or conduit can be used for part of the shielding length provided that there is no break in continuity.
- Ensure maximum separation between the power supply cable (line supply) and the motor cable.

### Installation diagram (examples)

Sizes 1 to 7 Size 8 Size 9




| | Size 1 | Size 2 | Size 3 | Size 4 | Size 5 | Size 6 | Size 7 | Size 8 | Size 9 |
|---|---|---|---|---|---|---|---|---|---|
| ATV31 | H018M3X,<br>H037M3X | H055M3X,<br>H075M3X | H018M2,<br>H037M2 | H055M2,<br>H075M2 | HU11M3X,<br>HU15M3X | HU11M2, HU15M2<br>HU22M3X<br>H037N4, H055N4,<br>H075N4, HU11N4,<br>HU15N4<br>H075S6X, HU15S6X | HU22M2<br>HU30M3X, HU40M3X<br>HU22N4, HU30N4,<br>HU40N4<br>HU22S6X, HU40S6X | HU55M3X,<br>HU75M3X<br>HU55N4,<br>HU75N4<br>HU55S6X,<br>HU75S6X | HD11M3X,<br>HD15M3X<br>HD11N4,<br>HD15N4<br>HD11S6X,<br>HD15S6X |

**1** Sheet steel grounded plate supplied with the drive, to be fitted as indicated on the diagram.

**2** Altivar 31

**3** Non-shielded power supply wires or cable

**4** Non-shielded wires for relay contacts

**5** Fix and ground the shielding of cables **6**, **7** and **8** as close as possible to the drive:
- Strip the shielding.
- Use stainless steel cable clamps of an appropriate size on the parts from which the shielding has been stripped, to attach them to the plate **1**.
The shielding must be clamped tightly enough to the metal plate to ensure correct contact.

**6** Shielded cable for motor connection with shielding connected to ground at both ends.
The shielding must be continuous and intermediate terminals must be in EMC shielded metal boxes.
For 0.18 to 1.5 kW drives, if the switching frequency is higher than 12 kHz, use cables with low linear capacitance: max. 130 pF (picoFarads) per metre.

**7** Shielded cable for connecting the control/signalling wiring.
For applications requiring several conductors, use cables with a small cross-section (0.5 $mm^2$).
The shielding must be connected to ground at both ends. The shielding must be continuous and intermediate terminals must be in EMC shielded metal boxes.

**8** Shielded cable for connecting braking resistor (if fitted).
The shielding must be continuous and intermediate terminals must be in EMC shielded metal boxes.

**Note:**
- If using an additional input filter, it should be mounted under the drive and connected directly to the line supply via an unshielded cable. Link 3 on the drive is then via the filter output cable.
- The HF equipotential ground connection between the drive, motor and cable shielding does not remove the need to connect the PE protective conductors (green-yellow) to the appropriate terminals on each unit.

**Atmoshperical pressure datasheet**

KELLER

# PIEZORESISTIVE PRESSURE TRANSMITTERS

## STANDARD AND PROGRAMMABLE (PRO) VERSIONS

## SERIES 21
## SERIES 21 PRO

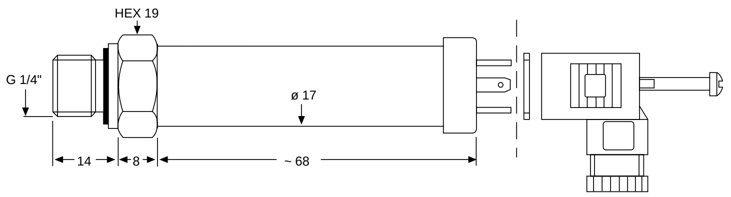
The KELLER Series 21/21PRO are extremely reliable and cost effective pressure transmitters. These pressure transmitters use a programmable KELLER-ASIC microchip called ”ProgRes” (programmable resistors). The ”ProgRes” chip enables the transmitters to be set up and calibrated using automated test equipment, for improved product quality and lower costs. The PRO versions also have the unique feature that they can be programmed or reranged by the end user, via the hand held PP-96 programmer. Since its first introduction in 1989, the Series 21/21PRO was an immediate success, thousands of transmitters and hundreds of programmers are in daily service worldwide.

This transmitter uses the KELLER Series 6SC capsule as the core. The sensor is a piezoresistive silicon pressure sensor, mounted in an oil filled capsule. The media pressure is isolated by a stainless steel diaphragm and transferred to the silicon sensor hydraulically via the oil. Sealed Gauge versions have a TAB (Tape Automated Bonding) sensor floating freely on a flexible print in the oil, this guarantees stability and endurance. Test have shown that even 20 million cycles of rapid changes in pressure from 10 bar to 250 bar do not cause damage to these sensors, and the Series 21 is therefore ideally suited for the toughest hydraulic applications.

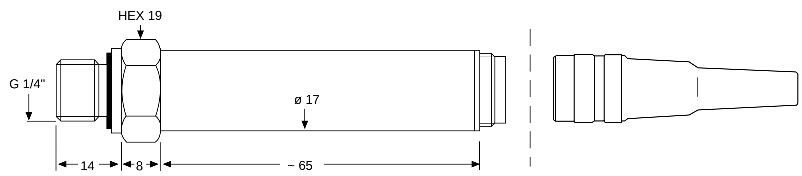
The ”ProgRes” circuit has programmable resistor networks, to set the Zero / Gain and thermal coefficients. All Series 21/21PRO transmitters are subjected to comprehensive test procedures which cycle both pressure and temperature, the various parameters are then automatically calibrated and set by the computer controlled test rig at the end of the run. A final test then checks that the settings are correct and within specification, only then are the transmitters are released for use. Note: The standard Series 21 transmitters are NOT user programmable.

The programmable version transmitters, Series 21PRO has the data lines brought out to an Amphenol 8pin connector, allowing simple and precise adjustment of Zero and Span. The KELLER Programmer PP-96 is used to facilitate programming of individual transmitters. The PP-96-10 can handle up to 10 identical transmitter at one time. The signal output of a PRO version transmitter may be adjusted to give full-scale between 25% and 120% of the nominal range, and ±25% zero adjustment.

**Series 21,** Cable or Plug




**Series 21 PRO,** Amphenol Plug C91 (8-pin)




**ELECTRICAL CONNECTIONS**

Series 21, with Cable

| | 2-WIRE | 3-WIRE |
|---|---|---|
| GREEN | | GND |
| WHITE | OUT/GND | +OUT |
| BROWN | +Vcc | +Vcc |

Series 21, with mPm 393-Connector

| PIN | 2-WIRE | 3-WIRE |
|---|---|---|
| 1 | OUT/GND | GND |
| 2 | | +OUT |
| 3 | +Vcc | +Vcc |

Series 21 / 21 PRO, Amphenol C91-Connector

| PIN | 2-WIRE | 3-WIRE |
|---|---|---|
| 4 | +Vcc | +Vcc |
| 6 | | +OUT |
| 8 | OUT/GND | GND |

Pin Assignment of Programming Lines (Pro)

| PIN | DATA LINE |
|---|---|
| 1 | SIO |
| 2 | VPROG |
| 3 | DATA |
| 5 | WRITE |
| 7 | CLOCK |

Subject to alterations

1/01

**KELLER AG für Druckmesstechnik** St. Gallerstrasse 119 CH-8404 Winterthur Tel. 052 - 235 25 25 Fax 052 - 235 25 00
**KELLER Gesellschaft für Druckmesstechnik mbH** Schwarzwaldstrasse 17 D-79798 Jestetten Tel. 07745 - 9214 - 0 Fax 07745 - 9214 - 50

KELLER

## SPECIFICATIONS

(BR: **Basic Range** of transducer) (FS: adjusted **Full Scale**)

| | **PRESSURE RANGES (BASIC RANGE). OVERPRESSURE. IN BAR** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **PR-21, PR-21PRO** | **-1** | **1** | **2** | **5** | **10** | **20** | | | | |
| **PAA-21, PAA-21PRO** | | **1** | **2** | **5** | **10** | **20** | | | | |
| **PA-21, PA-21PRO** | | | | **5** | **10** | **20** | **50** | **100** | **200** | **400** |
| Overpressure | -1 | 3 | 4 | 10 | 20 | 30 | 75 | 150 | 300 | 500 |
| Adjustment Range (PRO-Version) | | | | | | | | | | |
| Minimum: | -0,25 | 0,25 | 0,5 | 1,25 | 2,5 | 5 | 12 | 25 | 50 | 100 |
| Maximum: | -1 | 1,2 | 2,5 | 6 | 12 | 25 | 60 | 125 | 250 | 500 |

PR: Vented Gauge. Zero at atmospheric pressure PAA: Absolute. Zero at vacuum PA: Sealed Gauge. Zero at atmospheric pressure (at calibration day)

| | | | | |
|---|---|---|---|---|
| Signal Output | 4…20 mA | 0,5…4,5 Vdc | 0…10 Vdc | 0…20 mA |
| Supply Voltage | 8…28 Vdc | 5 V | 13…28 Vdc | 8…28 Vdc |
| Load Resistance | $R_{\Omega}$=(U - 8 V) / 0,02 A | > 5 kΩ | > 5 kΩ | $R_{\Omega}$=(U - 8 V) / 0,02 A |
| Current required | max. 25 mA | 5 mA max. | | max. 25 mA |
| Zero/Span Tolerance | ±0,5% BR (Basic Range) | ±0,5% BR | ±0,5% BR | ±0,5% BR |
| Configuration | 2 Wire | 3 Wire | | |
| Electrical Connection: | OUT/GND: Pin 1 / White | GND: Pin 1 / Green | | |
| mPm 393 or | | +OUT: Pin 2 / White | | |
| cable 2m | +Vcc: Pin 3 / Brown | +Vcc: Pin 3 / Brown | | |
| Linearity | ±0,2 %BR typ. / ±0,5%BR max. | | | |
| Total Error Band* +18…+22°C | | ±0,5%BR max. | ≤ 2 bar: ± 1%BR max. | |
| Total Error Band* 0…+50°C | | ±1,0%BR max. | ≤ 2 bar: ± 1,5%BR max. | |
| Total Error Band* -20…+80°C | | ±2,5%BR max. | ≤ 2 bar: ± 3%BR max. | |
| Tightening Effect on Zero | max. 6 mbar / 20 Nm ** (14,77 lbf/ft) | | | |

* Total error band includes linearity, hysteresis, repeatability, zero/span offsets and temperature effects.
** 20 Nm is a sufficient tightening moment for stable mounting. The moment can be reduced to 10 Nm with front-sided elastic seals.

CE

| | | |
|---|---|---|
| Storage - / Operating Temperature | -40...100°C / -20...80°C | |
| Compensated Temperature Range | 0...50°C (others on request) | |
| Temperature-Coefficients… | | |
| · Zero | typ. ≤ 0,1%BR / 10K | max. ≤ 0,2%BR / 10K |
| · Sensitivity | typ. ≤ 0,1% / 10K | max. ≤ 0,2% / 10K |
| Stability | ≤ 0,2%FS / year | |

| | |
|---|---|
| Electrical Connection | Series 21: mPm-Connector 393, incl. female connector or 2m screened cable<br>Series 21 PRO: Amphenol C91 (8-pin), incl. female connector |
| Dead Volume Change | < 0,1 $mm^3$ / 400 bar |
| Pressure Connection | G1/4" male, Viton©-Eolastic© -seal |
| Materials in Contact with Media | Stainless steel AISI 316 |
| Protection, CEI529 | mPm-plug or cable: IP65 Amphenol-plug: IP40 |
| Weight | ≈ 85 g |
| Insulation | > 100 MΩ / 50 V |
| Pressure Endurance | 10 Million Pressure Cycles 0…100 %FS at 25 °C |
| Vibration Endurance | 20 g, 20 to 5000 Hz |
| Shock Endurance | 20 g sinus 11 msec. |

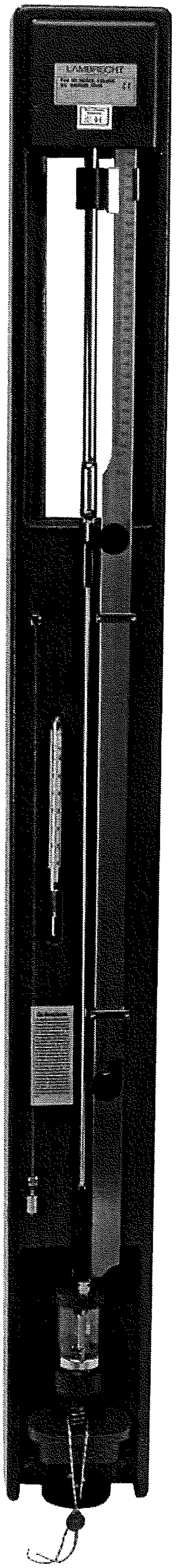
## OPTIONS

| | |
|---|---|
| Pressure Connection | G1/4" female, 7/16"-20-UNF male/female, 1/4"-18-NPT male, M12x1,5 male |
| Electrical Connection | Plug DIN 43650, others on request |
| Output signals | 1...5 V |
| Oil Filling | Halocarbon-oil for oxygen applications, olive-oil for food applications |



**KELLER AG für Druckmesstechnik** St. Gallerstrasse 119 CH-8404 Winterthur Tel. 052 - 235 25 25 Fax 052 - 235 25 00
**KELLER** **Gesellschaft für Druckmesstechnik mbH** Schwarzwaldstrasse 17 D-79798 Jestetten Tel. 07745 - 9214 - 0 Fax 07745 - 9214 - 50

**Lambrecht mercucry barometer datasheet**



# Mercury Barometer acc. to LAMBRECHT (604)

## 1. General remarks

The LAMBRECHT Mercury Barometer is a cistern barometer with adjustable scale.

Measuring the length of the mercury column, which holds the balance to the air pressure above the point of observation as against the vacuum, makes the measurement of the air pressure. For this reason the zero of the scale has to be adjusted to the level in the low cistern before each reading. The nonius and the scale can determine the length of the mercury column. After checking in our plant every barometer is locked and equipped with a seal.

ATTENTION: It is important to attend to item 3!

## 2. Choosing a proper place

The barometer should be hung up in a cold room which shows to the north, but not too close to the window. It has to be protected against rapid changes of temperature, direct heat radiation, e. g. sun or stove, and vibrations.

The read position above and the mercury level below must be sufficiently illuminated. When hanging up the instrument, the upper part of the scale should be at eye-level. By means of the plummet the instrument has to be aligned vertically.

## 3. Initial operation

The placing into operation has to be made very carefully. When the mercury barometer is not mounted in strict accordance to the following instructions, there will not be any guarantee.

During transport the barometer tube is completely with mercury. A Special seal prevents the departure of the mercury, as well as the penetration of air.

After erection of the barometer (in vertical position) the instrument can be unlocked after approx. 5 hours assimilation time.

- Removing of the seal.
- Opening of the stopper by clockwise rotation of the great knurling wheel (until limit). Then the mercury is sinking to the corresponding level.

As the smallest portion of air in the vacuum gauge decreases the barometer level, there is an air trap fused in the barometer (Bunten's peak). When there is air below the air trap in the tube, this has no influence to the measuring accuracy, provided that the whole mercury column is not interrupted (see fig. 1).

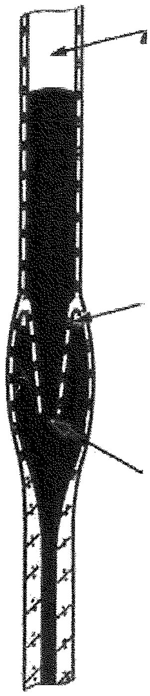



Fig.1

It is advisable, 3 - 4 hours before the first reading, to satisfy oneself that there is no air in the vacuum tube. For this reason the barometer must be inclined slowly from the vertical position (up to max. 40°) until the mercury fills the whole tube.

**ATTENTION**
**The instrument in unlocked position must not be inclined for more than 40° or in horizontal position and must not be turned upside down!**

When the mercury meets with a slight sound the upper part of the tube the space is a vacuum. This inspection should be made very carefully and should be repeated from time to time. If it is not possible to remove air blisters, which are beyond the air trap, or if there is air in the vacuum, it is necessary to return the instrument to LAMBRECHT for repair!

## 4. Transport

If the place of the barometer has to be changed, the lower end of the instrument must be moved to one side until the whole tube is filled with mercury.

The tube must then be locked by left-hand rotation of the knurled screw, whereat a small pressure to the screw at the same time, thus ensuring that the tube is firmly closed. The barometer may now be taken off the wall and packed.

## 5. Maintenance

Due to the unimpeded access of air, the mercury level in the lower cistern oxidizes in the course of the time. If the oxidation has reached that state where the point of the scale can no more be adjusted with security to the lower mercury level, it is recommended to remove the mercury from the cistern and to clean it.

The mercury cleaning has to be carried out by skilled staff only!

## 6. Measurement

A scale at the right side of the tube measures the length of the mercury column, which holds the balance to the present air pressure. The zero of the division coincides exactly with the point of the scale. Before each reading this point has to be adjusted to the mercury level in the cistern, so that the point and its reflected image will form a cross (X) (see fig. 2).

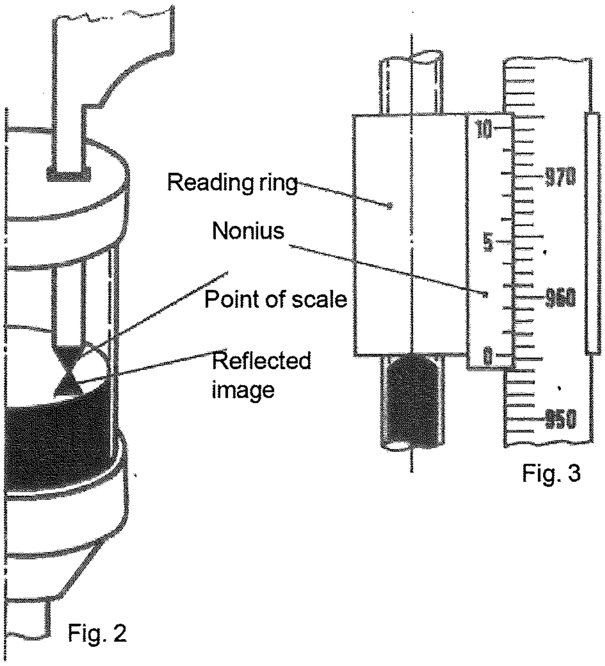



Fig. 3

Fig. 2

The knurled screw at the lower end of the scale makes the adjustment. The reading ring at the nonius, which surrounds the tube, is adjusted by the small knurled screw at the upper part of the scale in such a way that its lower edge is situated immediately above the meniscus of the mercury column.

The eye must then be in the same height with the reading ring and the meniscus of the mercury. The adjustment is correct when the edges of the metal ring before and behind the tube coincide touching the meniscus of the mercury tangentially (see fig. 3).

The last graduation line on the scale below the zero of the nonius division indicates the entire 1/1 mbar. For the reading of the 1/10 mbar serves the nonius. That graduation line of the nonius, which coincides with a graduation line of the main division, indicates the 1/10 mbar. They must be added to the entire 1/1 mbar lead on the main scale. The barometric reading in fig. 3 is 955.3 mbar.

For the determination of the height of the meniscus a second adjustment is necessary. The lower edge of the reading ring has to coincide with the line, at which the mercury touches the tube. The difference of the two readings is the height of the meniscus.

## 7. Correction of the raw barometric reading

The length of the mercury column ascertained in the above mentioned way depends apart from the air pressure on other influences, which do not admit a direct comparison of the individual measuring result. Therefore, various corrections have to be made for the raw barometric reading, which are described as follows:

### 7.1 Correction of temperature (table I)

The length of the mercury column depends on the density of the mercury and thus on its temperature. The temperature also influences the length of the scale. For this reason barometric readings are converted to a reference temperature of 0°C, i. e. the length must be determined, which the mercury column would have at a temperature of 0°C.

The following formula has teen taken as a basis for table I:

$$K_t = -b_t \cdot \frac{182 \cdot 10^{-6} \cdot t - 11 \cdot 10^{-6}(t-20)}{1 + \left[182 \cdot 10^{-6} \cdot t - 11 \cdot 10^{-6}(t-20)\right]}$$

Where: $K_t$ = temperature correction

$b_t$ = barometric reading at the temperature t [°C]

### 7.2 Correction of gravity (table IIa and IIb)

The length of the mercury column also depends on the acceleration due to gravity, which changes with the geographical latitude (table IIa) and the height above sea level (table IIb). Therefore the barometric reading is converted to the normal gravity.

(At sea level below 45° of latitude the gravitational acceleration amounts to 9.80616 m/s². The normal gravity or the standard value of the gravitational acceleration is 9.80665 m/s², which is reached at sea level at approx. 45°33').

The following formula has been taken as a basis for table IIa:

$$K_g\varphi = \left[\frac{9{,}80616}{9{,}80665}\left(1 - 0{,}002637\cos 2\varphi\right) - 1\right] \cdot b_0$$

Where:

$K_g\varphi$ = correction of gravity due to the geographical latitude $\varphi$°

$b_0$ = barometric reading reduced to 0°C

The following formula has been taken as a basis for table IIb:

$$K_{gH} = -0{,}195 \cdot 10^{-6} \cdot H \cdot b_0$$

Where:

$K_{gH}$ = correction of gravity due to the height H

$b_0$ = barometric reading reduced to °C

When using the barometer stationary it is recommended to register the interpolated values of the table for latitude $\varphi$ and the height H in the top column of the individual barometric readings $b_0$.

### 7.3 Correction of the capillary depression (table III)

Mercury does not moistened glass normally, a convex meniscus is formed in the barometric tube. Thereby the mercury height will be reduced. The correction values depend on the inside diameter of the tube and on the height of the meniscus. They are ascertained empirically by comparing them with a barometer free of depression.

The corresponding correction values for the mercury barometer of LAMBRECHT type are listed in table III.

### 7.4 Example of correction

For scale in mbar

| | | |
|---|---|---|
| Barometric reading $b_t$ | | 1000.00 mbar |
| Temperature t at the barometer: 20°C *(acc. to table I)* | - | 3.63 mbar |
| $b_0$ | = | 996.37 mbar |
| Geographical latitude of the place of observation: 55° *(acc. to table IIa)* | + | 0.85 mbar |
| Height H of the place of observation: 200 m *(acc. to table IIb)* | - | 0.04 mbar |
| Depression, height of the top: 1 mbar *(acc. to table III)* | + | 0.57 mbar |
| Real barometric reading at a height of 200 m | = | 997.75 mbar |

For scale mmHg

| | | |
|---|---|---|
| Barometric reading $b_t$ | | 750.00 mmHg |
| Temperature t at the barometer: 20°C *(acc. to table I)* | - | 2. 72 mmHg |
| $b_0$ | = | 747.28 mmHg |
| Geographical latitude of the place of observation: 55° *(acc. to table IIa)* | + | 0.64 mmHg |
| Height H of the place of observation: 200 m *(acc. to table IIb)* | - | 0.03 mmHg |
| Depression, height of the top: 1 mm *(acc. to table III)* | + | 0.56 mmHg |
| Real barometric reading at a height of 200 m | = | 748.45 mmHg |

## 8. Reduction of the pressure

from actual height to sea level, see table of reduction (table 4).

### (604) Mercury Barometer acc. to LAMBRECHT

| | |
|---|---|
| **Id-No.** | **00.06040.100 000** |
| Measuring elements: | Mercury column • liquid thermometer |
| Measuring ranges: | 840...1050 hPa • -15...+50°C |
| Accuracy: | ± 0.25 hPa at 20°C • ± 1°C |
| Resolution: | 0.1 hPa with vernier • 0.5°C |
| Scale: | 1 hPa • 1°C |
| Range of application: | Altitudes 0...1200 m • Temperatures -15...+50°C |
| Housing: | Aluminium · lacquer RAL 5009 (azure) |
| Dimensions: | 1030 x 105 x 75 mm inside Ø of the tube is 8 mm |
| Weight: | Approx. 5.5 kg |

Varieties:

| | |
|---|---|
| **00.06040.100 002** | **(604) Mercury Barometer acc. to LAMBRECHT** officially calibrated |
| **00.06070.100 000** | **(607) Mercury Baro-Vacuum Gauge** (on request) |

Enclosures:

- Correction tables I; IIa; IIb; III for mercury barometer according to LAMBRECHT
- Table 4 (reduction of atmospheric pressure)



**Wilh. LAMBRECHT GmbH**
**Friedländer Weg 65-67**
**37085 Göttingen**
**Germany**

Tel +49-(0)551-4958-0
Fax +49-(0)551-4958-312
E-Mail info@lambrecht.net
Internet www.lambrecht.net

LAMBRECHT
METEOROLOGICAL INSTRUMENTS GERMANY

# Korrektionstabellen
für Quecksilber-Barometer nach Lambrecht

# Correction Tables
for Lambrecht Mercury Barometers

# Tables de Correction
pour Baromètres à Mercure de Lambrecht

# Tablas de Corrección
para Barómetros de Mercurio según Lambrecht

LAMBRECHT
METEOROLOGICAL INSTRUMENTS GERMANY

BA 604 Tab. I

**Umrechnung der bei der Temperatur t (° C) abgelesenen Barometerstände $b_t$ (mm QS, mb oder hPa) auf die Bezugstemperatur 0 (° C)**

Conversion of the barometer reading $b_t$ (mm mercury, mb or hPa) taken at the temperature (° C) for the reference temperature of 0 (° C)

Récalculation des niveaux barométriques $b_t$ (mm mercure, mb ou hPa) lus à la température t (° C) sur la température de référence 0 (° C)

Conversión de las alturas barométricas $b_t$ (mm mercurio, mb o hPa) leídas con la temperatura t (° C) a la temperatura de referencia 0 (° C)

Abgelesener Barometerstand $b_t$ in mm QS, mb oder hPa
Barometer reading $b_t$ in mm mercury, mb or hPa
Niveau barométrique $b_t$ en mm mercure, mb ou hPa
Altura barométrica $b_t$ en mm mercurio,mb o hPa

| t (° C) | 620 | 640 | 660 | 680 | 700 | 720 | 740 | 760 | 780 | 800 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0,14 | 0,14 | 0,15 | 0,15 | 0,15 | 0,16 | 0,16 | 0,17 | 0,17 | 0,18 |
| 1 | 0,24 | 0,25 | 0,26 | 0,27 | 0,27 | 0,28 | 0,29 | 0,30 | 0,30 | 0,31 |
| 2 | 0,35 | 0,36 | 0,37 | 0,38 | 0,39 | 0,40 | 0,42 | 0,43 | 0,44 | 0,45 |
| 3 | 0,45 | 0,47 | 0,48 | 0,50 | 0,51 | 0,53 | 0,54 | 0,56 | 0,57 | 0,59 |
| 4 | 0,56 | 0,58 | 0,60 | 0,61 | 0,63 | 0,65 | 0,67 | 0,69 | 0,70 | 0,72 |
| 5 | 0,67 | 0,69 | 0,71 | 0,73 | 0,75 | 0,77 | 0,79 | 0,82 | 0,84 | 0,86 |
| 6 | 0,77 | 0,80 | 0,82 | 0,85 | 0,87 | 0,90 | 0,92 | 0,95 | 0,97 | 1,00 |
| 7 | 0,88 | 0,91 | 0,93 | 0,96 | 0,99 | 1,02 | 1,05 | 1,08 | 1,10 | 1,13 |
| 8 | 0,98 | 1,01 | 1,05 | 1,08 | 1,11 | 1,14 | 1,17 | 1,20 | 1,24 | 1,27 |
| 9 | 1,09 | 1,12 | 1,16 | 1,19 | 1,23 | 1,26 | 1,30 | 1,33 | 1,37 | 1,40 |
| 10 | 1,19 | 1,23 | 1,27 | 1,31 | 1,35 | 1,39 | 1,43 | 1,46 | 1,50 | 1,54 |
| 11 | 1,30 | 1,34 | 1,38 | 1,43 | 1,47 | 1,51 | 1,55 | 1,59 | 1,64 | 1,68 |
| 12 | 1,41 | 1,45 | 1,50 | 1,54 | 1,59 | 1,63 | 1,68 | 1,72 | 1,77 | 1,81 |
| 13 | 1,51 | 1,56 | 1,61 | 1,66 | 1,71 | 1,75 | 1,80 | 1,85 | 1,90 | 1,95 |
| 14 | 1,62 | 1,67 | 1,72 | 1,77 | 1,83 | 1,88 | 1,93 | 1,98 | 2,03 | 2,09 |
| 15 | 1,72 | 1,78 | 1,83 | 1,89 | 1,94 | 2,00 | 2,06 | 2,11 | 2,17 | 2,22 |
| 16 | 1,83 | 1,89 | 1,95 | 2,00 | 2,06 | 2,12 | 2,18 | 2,24 | 2,30 | 2,36 |
| 17 | 1,93 | 2,00 | 2,06 | 2,12 | 2,18 | 2,24 | 2,31 | 2,37 | 2,43 | 2,49 |
| 18 | 2,04 | 2,10 | 2,17 | 2,24 | 2,30 | 2,37 | 2,43 | 2,50 | 2,56 | 2,63 |
| 19 | 2,14 | 2,21 | 2,28 | 2,35 | 2,42 | 2,49 | 2,56 | 2,63 | 2,70 | 2,77 |
| 20 | 2,25 | 2,32 | 2,39 | 2,47 | 2,54 | 2,61 | 2,68 | 2,76 | 2,83 | 2,90 |
| 21 | 2,35 | 2,43 | 2,51 | 2,58 | 2,66 | 2,73 | 2,81 | 2,89 | 2,96 | 3,04 |
| 22 | 2,46 | 2,54 | 2,62 | 2,70 | 2,78 | 2,86 | 2,93 | 3,01 | 3,09 | 3,17 |
| 23 | 2,56 | 2,65 | 2,73 | 2,81 | 2,90 | 2,98 | 3,06 | 3,14 | 3,23 | 3,31 |
| 24 | 2,67 | 2,76 | 2,84 | 2,93 | 3,01 | 3,10 | 3,19 | 3,27 | 3,36 | 3,44 |
| 25 | 2,77 | 2,86 | 2,95 | 3,04 | 3,13 | 3,22 | 3,31 | 3,40 | 3,49 | 3,58 |
| 26 | 2,88 | 2,97 | 3,07 | 3,16 | 3,25 | 3,34 | 3,44 | 3,53 | 3,62 | 3,72 |
| 27 | 2,98 | 3,08 | 3,18 | 3,27 | 3,37 | 3,47 | 3,56 | 3,66 | 3,75 | 3,85 |
| 28 | 3,09 | 3,19 | 3,29 | 3,39 | 3,49 | 3,59 | 3,69 | 3,79 | 3,89 | 3,99 |
| 29 | 3,19 | 3,30 | 3,40 | 3,50 | 3,61 | 3,71 | 3,81 | 3,92 | 4,02 | 4,12 |
| 30 | 3,29 | 3,40 | 3,51 | 3,61 | 3,72 | 3,82 | 3,93 | 4,04 | 4,14 | 4,25 |

**Die in der Tabelle angegebenen Korrektionswerte $K_t$ (in mm QS,mb oder hPa) sind vom Barometerstand abzuziehen**

The corrections $K_t$ (in mm Hg, mb or hPa) of this table should be deducted from the barometer reading.

Les corrections $K_t$ (en mm Hg, mb ou hPa) du tableau sont à déduire du niveau barométrique.

Las correcciones $K_t$ (en mm Hg, mb o hPa) de la tabla hay que deducirlas de la altura barométrica.

BA 604 Tab. I

**Umrechnung der bei der Temperatur t (° C) abgelesenen Barometerstände $b_t$ (mm QS, mb oder hPa) auf die Bezugstemperatur 0 (° C)**

Conversion of the barometer reading $b_t$ (mm mercury, mb or hPa) taken at the temperature (° C) for the reference temperature of 0 (° C)

Récalculation des niveaux barométriques $b_t$ (mm mercure, mb oú hPa) lus à la température t (° C) sur la température de référence 0 (°C )

Conversión de las alturas barométricas $b_t$ (mm mercurio, mb o hPa) leídas con la temperatura t (° C) a la temperatura de referencia 0 (° C)

| | Abgelesener Barometerstand $b_t$ in mm QS, mb oder hPa<br>Barometer reading $b_t$ in mm mercury, mb or hPa<br>Niveau barométrique $b_t$ en mm mercure, mb ou hPa<br>Altura barométrica $b_t$ en mm mercurio, mb o hPa | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| t (° C) | 840 | 860 | 880 | 900 | 920 | 940 | 960 | 980 | 1000 | 1020 | 1040 | 1060 |
| 0 | 0,18 | 0,19 | 0,19 | 0,20 | 0,20 | 0,21 | 0,21 | 0,22 | 0,22 | 0,22 | 0,23 | 0,23 |
| 1 | 0,33 | 0,34 | 0,34 | 0,35 | 0,36 | 0,37 | 0,38 | 0,38 | 0,39 | 0,40 | 0,41 | 0,41 |
| 2 | 0,47 | 0,48 | 0,49 | 0,51 | 0,52 | 0,53 | 0,54 | 0,55 | 0,56 | 0,57 | 0,58 | 0,60 |
| 3 | 0,62 | 0,63 | 0,64 | 0,66 | 0,67 | 0,69 | 0,70 | 0,72 | 0,73 | 0,75 | 0,76 | 0,78 |
| 4 | 0,76 | 0,78 | 0,79 | 0,81 | 0,83 | 0,85 | 0,87 | 0,89 | 0,90 | 0,92 | 0,94 | 0,96 |
| 5 | 0,90 | 0,92 | 0,94 | 0,97 | 0,99 | 1,01 | 1,03 | 1,05 | 1,07 | 1,10 | 1,12 | 1,14 |
| 6 | 1,05 | 1,07 | 1,10 | 1,12 | 1,14 | 1,17 | 1,19 | 1,22 | 1,24 | 1,27 | 1,29 | 1,32 |
| 7 | 1,19 | 1,22 | 1,25 | 1,27 | 1,30 | 1,33 | 1,36 | 1,39 | 1,41 | 1,44 | 1,47 | 1,50 |
| 8 | 1,33 | 1,36 | 1,40 | 1,43 | 1,46 | 1,49 | 1,52 | 1,55 | 1,59 | 1,62 | 1,65 | 1,68 |
| 9 | 1,47 | 1,51 | 1,55 | 1,58 | 1,62 | 1,65 | 1,69 | 1,72 | 1,76 | 1,79 | 1,83 | 1,86 |
| 10 | 1,62 | 1,66 | 1,70 | 1,73 | 1,77 | 1,81 | 1,85 | 1,89 | 1,93 | 1,96 | 2,00 | 2,04 |
| 11 | 1,76 | 1,80 | 1,85 | 1,89 | 1,93 | 1,97 | 2,01 | 2,05 | 2,10 | 2,14 | 2,18 | 2,22 |
| 12 | 1,90 | 1,95 | 1,99 | 2,04 | 2,09 | 2,13 | 2,18 | 2,22 | 2,27 | 2,31 | 2,36 | 2,40 |
| 13 | 2,05 | 2,10 | 2,14 | 2,19 | 2,24 | 2,29 | 2,34 | 2,39 | 2,44 | 2,49 | 2,53 | 2,58 |
| 14 | 2,19 | 2,24 | 2,29 | 2,35 | 2,40 | 2,45 | 2,50 | 2,56 | 2,61 | 2,66 | 2,71 | 2,76 |
| 15 | 2,33 | 2,39 | 2,44 | 2,50 | 2,56 | 2,61 | 2,67 | 2,72 | 2,78 | 2,83 | 2,89 | 2,94 |
| 16 | 2,48 | 2,53 | 2,59 | 2,65 | 2,71 | 2,77 | 2,83 | 2,89 | 2,95 | 3,01 | 3,07 | 3,12 |
| 17 | 2,62 | 2,68 | 2,74 | 2,81 | 2,87 | 2,93 | 2,99 | 3,05 | 3,12 | 3,18 | 3,24 | 3,30 |
| 18 | 2,76 | 2,83 | 2,89 | 2,96 | 3,02 | 3,09 | 3,16 | 3,22 | 3,29 | 3,35 | 3,42 | 3,48 |
| 19 | 2,90 | 2,97 | 3,04 | 3,11 | 3,18 | 3,25 | 3,32 | 3,39 | 3,46 | 3,53 | 3,60 | 3,66 |
| 20 | 3,05 | 3,12 | 3,19 | 3,26 | 3,34 | 3,41 | 3,48 | 3,55 | 3,63 | 3,70 | 3,77 | 3,84 |
| 21 | 3,19 | 3,27 | 3,34 | 3,42 | 3,49 | 3,57 | 3,64 | 3,72 | 3,80 | 3,87 | 3,95 | 4,02 |
| 22 | 3,33 | 3,41 | 3,49 | 3,57 | 3,65 | 3,73 | 3,81 | 3,89 | 3,97 | 4,05 | 4,12 | 4,20 |
| 23 | 3,47 | 3,56 | 3,64 | 3,72 | 3,80 | 3,89 | 3,97 | 4,05 | 4,14 | 4,22 | 4,30 | 4,38 |
| 24 | 3,62 | 3,70 | 3,79 | 3,87 | 3,96 | 4,05 | 4,13 | 4,22 | 4,31 | 4,39 | 4,48 | 4,56 |
| 25 | 3,76 | 3,85 | 3,94 | 4,03 | 4,12 | 4,21 | 4,30 | 4,39 | 4,47 | 4,56 | 4,65 | 4,74 |
| 26 | 3,90 | 3,99 | 4,09 | 4,18 | 4,27 | 4,37 | 4,46 | 4,55 | 4,64 | 4,74 | 4,83 | 4,92 |
| 27 | 4,04 | 4,14 | 4,24 | 4,33 | 4,43 | 4,52 | 4,62 | 4,72 | 4,81 | 4,91 | 5,01 | 5,10 |
| 28 | 4,19 | 4,29 | 4,39 | 4,48 | 4,58 | 4,68 | 4,78 | 4,88 | 4,98 | 5,08 | 5,18 | 5,28 |
| 29 | 4,33 | 4,43 | 4,53 | 4,64 | 4,74 | 4,84 | 4,95 | 5,05 | 5,15 | 5,26 | 5,36 | 5,46 |
| 30 | 4,46 | 4,57 | 4,67 | 4,78 | 4,89 | 4,99 | 5,10 | 5,21 | 5,31 | 5,42 | 5,52 | 5,63 |

**Die in der Tabelle angegebenen Korrektionswerte $K_t$ (in mm QS, mb oder hPa) sind vom Barometerstand abzuziehen**

The corrections $K_t$ (in mm Hg, mb or hPa) of this table should be deducted from the barometer reading.

Les corrections $K_t$ (en mm Hg, mb ou hPa) du tableau sont à déduire du niveau barométrique.

Las correcciones $K_t$ (en mm Hg, mb o hPa) de la tabla hay que deducirlas de la altura barométrica.

BA 604 Tab. II a

**Umrechnung der ermittelten Barometerstände $b_0$ (mm QS, mb oder hPa) auf Normalschwere $g_n$ = 9,80665 (m/s²)**
**a) wegen der geographischen Breite φ (°)**

Conversion of the barometer reading $b_0$ (mm Hg, mb or hPa) for normal gravity $g_n$ = 9,80665 (m/s²).
a) because of the geographical latitude φ (°).

Récalculation des niveaux barométriques $b_0$ (mm Hg, mb ou hPa) sur gravité normale $g_n$ = 9,80665 (m/s²).
a) à cause de la latitude géographique φ (°).

Conversión de las alturas barométricas $b_0$ (mm Hg, mb o hPa) en gravedad normal $g_n$ = 9,80665 (m/s²)
a) por latitud geográfica φ (°)

Barometerstand $b_0$ in mm QS, mb oder hPa
Barometer reading $b_0$ in mm of mercury, mb or hPa
Niveau barométrique $b_0$ en mm de mercure, mb ou hPa
Altura barométrica $b_0$ en mm de mercurio, mb o hPa

| φ (°) | 620 | 640 | 660 | 680 | 700 | 720 | 740 | 760 | 780 | 800 | 840 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | −1,67 | −1,72 | −1,77 | −1,83 | −1,88 | −1,93 | −1,99 | −2,04 | −2,10 | −2,15 | −2,26 |
| 5 | −1,64 | −1,69 | −1,75 | −1,80 | −1,85 | −1,91 | −1,96 | −2,01 | −2,06 | −2,12 | −2,22 |
| 10 | −1,57 | −1,62 | −1,67 | −1,72 | −1,77 | −1,82 | −1,87 | −1,92 | −1,97 | −2,02 | −2,12 |
| 15 | −1,45 | −1,49 | −1,54 | −1,59 | −1,63 | −1,68 | −1,73 | −1,77 | −1,82 | −1,87 | −1,96 |
| 20 | −1,28 | −1,32 | −1,37 | −1,41 | −1,45 | −1,49 | −1,53 | −1,57 | −1,61 | −1,66 | −1,74 |
| 25 | −1,08 | −1,12 | −1,15 | −1,19 | −1,22 | −1,26 | −1,29 | −1,33 | −1,36 | −1,40 | −1,47 |
| 30 | −0,85 | −0,88 | −0,90 | −0,93 | −0,96 | −0,99 | −1,01 | −1,04 | −1,07 | −1,10 | −1,15 |
| 35 | −0,59 | −0,61 | −0,63 | −0,65 | −0,67 | −0,69 | −0,70 | −0,72 | −0,74 | −0,76 | −0,80 |
| 40 | −0 31 | −0,33 | −0,34 | −0,35 | −0,36 | −0,37 | −0,38 | −0,39 | −0,40 | −0,41 | −0,43 |
| 45 | −0,03 | −0,03 | −0,03 | −0,03 | −0,03 | −0,04 | −0,04 | −0,04 | −0,04 | −0,04 | −0,04 |
| 50 | +0,25 | +0,26 | +0,27 | +0,28 | +0,29 | +0,29 | +0,30 | +0,31 | +0,32 | +0,33 | +0,34 |
| 55 | +0,53 | +0,55 | +0,56 | +0,58 | +0,60 | +0,61 | +0,63 | +0,65 | +0,66 | +0,68 | +0,72 |
| 60 | +0,79 | +0,81 | +0,84 | +0,86 | +0,89 | +0,91 | +0,94 | +0,96 | +0,99 | +1,02 | +1,07 |
| 65 | +1,02 | +1,05 | +1,09 | +1,12 | +1,15 | +1,18 | +1,22 | +1,25 | +1,28 | +1,32 | +1,38 |
| 70 | +1,23 | +1,26 | +1,30 | +1,34 | +1,38 | +1,42 | +1,46 | +1,50 | +1,54 | +1,58 | +1,65 |
| 75 | +1,39 | +1,43 | +1,47 | +1,52 | +1,56 | +1,61 | +1,65 | +1,70 | +1,74 | +1,79 | +1,88 |
| 80 | +1,51 | +1,55 | +1,60 | +1,65 | +1,70 | +1,75 | +1,80 | +1,85 | +1,89 | +1,94 | +2,04 |
| 85 | +1,58 | +1,63 | +1,68 | +1,73 | +1,78 | +1,83 | +1,88 | +1,94 | +1,99 | +2,04 | +2,14 |
| 90 | +1,60 | +1,66 | +1,71 | +1,76 | +1,81 | +1,86 | +1,91 | +1,97 | +2,02 | +2,07 | +2,17 |

**Die in der Tabelle angegebenen Korrektionswerte $K_{g\varphi}$ (in mm QS, mb oder hPa) sind im Bereich der Breitengrade von 0 bis ≈45° 33′ vom Barometerstand abzuziehen, im Bereich der Breitengrade von ≈45° 33′ bis 90° zum Barometerstand zuzuzählen.**

The corrections of the table $K_{g\varphi}$ (in mm Hg, mb or. hPa) should be deducted from the barometer reading in case of latitudes from 0 to ≈45° 33′, in case of latitudes from ≈45° 33′ to 90° they are to be added.

Les corrections $K_{g\varphi}$ (en mm Hg, mb ou hPa) du tableau sont à déduire du niveau barométrique pour des latitudes entre 0 et ≈45° 33′, pour des latitudes entre ≈45° 33′ à 90° elles sont à ajouter.

Las correcciones $K_{g\varphi}$ (en mm Hg, mb o hPa) de la tabla hay que deducirlas de la altura barométrica en latitudes de 0 a ≈45° 33′ y añadirlas en latidudes de ≈45° 33′ a 90°.

**Umrechnung der ermittelten Barometerstände $b_0$ (mm QS, mb oder hPa) auf Normalschwere $g_n$ = 9,80665 (m/s²)**
**a) wegen der geographischen Breite φ (°)**

Conversion of the barometer reading $b_0$ (mm Hg, mb or hPa) for normal gravity $g_n$ = 9,80665 (m/s²).
a) because of the geographical latitude φ (°).

Récalculation des niveaux barométriques $b_0$ (mm Hg, mb ou hPa) sur gravité normale $g_n$ = 9,80665 (m/s²).
a) à cause de la latitude géographique φ (°).

Conversión de las alturas barométricas $b_0$ (mm Hg, mb o hPa) en gravedad normal $g_n$ = 9,80665 (m/s²)
a) por latitud geogràfica φ (°)

| | Barometerstand $b_0$ in mm QS, mb oder hPa<br>Barometer reading $b_0$ in mm of mercury, mb or hPa<br>Niveau barométrique $b_0$ en mm de mercure, mb ou hPa<br>Altura barométrica $b_0$ en mm de mercurio, mb o hPa | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| φ (°) | 860 | 880 | 900 | 920 | 940 | 960 | 980 | 1000 | 1020 | 1040 | 1060 |
| 0 | −2,31 | −2,36 | −2,42 | −2,47 | −2,53 | −2,58 | −2,63 | −2,69 | −2,74 | −2,79 | −2,85 |
| 5 | −2,28 | −2,33 | −2,38 | −2,44 | −2,49 | −2,54 | −2,59 | −2,65 | −2,70 | −2,75 | −2,81 |
| 10 | −2,15 | −2,22 | −2,28 | −2,33 | −2,38 | −2,43 | −2,48 | −2,53 | −2,58 | −2,63 | −2,68 |
| 15 | −2,01 | −2,05 | −2,10 | −2,15 | −2,20 | −2,24 | −2,29 | −2,33 | −2,38 | −2,43 | −2,47 |
| 20 | −1,78 | −1,82 | −1,86 | −1,90 | −1,95 | −1,99 | −2,03 | −2,07 | −2,10 | −2,15 | −2,19 |
| 25 | −1,50 | −1,54 | −1,57 | −1,61 | −1,64 | −1,68 | −1,71 | −1,75 | −1,78 | −1,81 | −1,85 |
| 30 | −1,18 | −1,20 | −1,23 | −1,26 | −1,29 | −1,31 | −1,34 | −1,37 | −1,40 | −1,42 | −1,45 |
| 35 | −0,82 | −0,84 | −0,86 | −0,88 | −0,89 | −0,91 | −0,93 | −0,95 | −0,97 | −0,99 | −1,01 |
| 40 | −0,44 | −0,45 | −0,46 | −0,47 | −0,48 | −0,49 | −0,50 | −0,51 | −0,52 | −0,53 | −0,54 |
| 45 | −0,04 | −0,04 | −0,05 | −0,05 | −0,05 | −0,05 | −0,05 | −0,05 | −0,05 | −0,05 | −0,05 |
| 50 | +0,35 | +0,36 | +0,37 | +0,38 | +0,38 | +0,39 | +0,40 | +0,41 | +0,42 | +0,42 | +0,43 |
| 55 | +0,73 | +0,75 | +0,77 | +0,78 | +0,80 | +0,82 | +0,83 | +0,85 | +0,87 | +0,89 | +0,90 |
| 60 | +1,09 | +1,12 | +1,14 | +1,17 | +1,19 | +1,22 | +1,24 | +1,27 | +1,29 | +1,32 | +1,35 |
| 65 | +1,42 | +1,45 | +1,48 | +1,51 | +1,55 | +1,58 | +1,61 | +1,65 | +1,68 | +1,71 | +1,74 |
| 70 | +1,69 | +1,73 | +1,77 | +1,81 | +1,85 | +1,89 | +1,93 | +1,97 | +2,01 | +2,05 | +2,09 |
| 75 | +1,92 | +1,97 | +2,01 | +2,06 | +2,10 | +2,14 | +2,19 | +2,23 | +2,28 | +2,32 | +2,37 |
| 80 | +2,09 | +2,14 | +2,19 | +2,23 | +2,28 | +2,33 | +2,38 | +2,43 | +2,48 | +2,52 | +2,57 |
| 85 | +2,19 | +2,24 | +2,29 | +2,34 | +2,39 | +2,44 | +2,50 | +2,55 | +2,60 | +2,65 | +2,70 |
| 90 | +2,22 | +2,28 | +2,33 | +2,38 | +2,43 | +2,48 | +2,54 | +2,59 | +2,64 | +2,69 | +2,74 |

**Die in der Tabelle angegebenen Korrektionswerte $K_{g\varphi}$ (in mm QS, mb oder hPa) sind im Bereich der Breitengrade von 0 bis ≈45° 33′ vom Barometerstand abzuziehen, im Bereich der Breitengrade von ≈45° 33′ bis 90° zum Barometerstand zuzuzählen.**

The corrections of the table $K_{g\varphi}$ (in mm Hg, mb or hPa) should be deducted from the barometer reading in case of latitudes from 0 to ≈45° 33′, in case of latitudes from ≈45° 33′ to 90° they are to be added.

Les corrections $K_{g\varphi}$ (en mm Hg, mb ou hPa) du tableau sont à dé duire du niveau barométrique pour des latitudes entre 0 et ≈45° 33′, pour des latitudes entre ≈45° 33′ à 90° elles sont à ajouter.

Las correcciones $K_{g\varphi}$ (en mm Hg, mb o hPa) de la tabla hay que deducirlas de la altura barométrica en latitudes de 0 a ≈45° 33′ y añadirlas en latitudes de ≈45° 33′ a 90°.

BA 604 Tab. II b

**Umrechnung der ermittelten Barometerstände $b_0$ (mm QS, mb oder hPa) auf Normalschwere $g_n = 9{,}80665$ (m/s²)**
**b) wegen der Höhe H (m) über dem Meeresspiegel**

Conversion of the barometer reading $b_0$ (mm Hg, mb or hPa) for normal gravity $g_n = 9{,}80665$ (m/s²).
b) because of the height H (m) above sea-level

Récalculation des niveaux barométriques $b_0$ (mm Hg, mb ou hPa) sur gravité normale $g_n = 9{,}80665$ (m/s²).
b) à cause de la hauteur H (m) au-dessus du viveau de la mer.

Conversión de las alturas barométricas $b_0$ (mm Hg, mb o hPa) a la gravedad normal $g_n = 9{,}80665$ (m/s²)
b) a causa de la altura H (m) sobre el nivel del mar

Barometerstand $b_0$ in mm QS, mb oder hPa
Barometer reading $b_0$ in mm of mercury, mb or hPa
Niveau barométrique $b_0$ en mm de mercure, mb ou hPa
Altura barométrica $b_0$ en mm de mercurio, mb o hPa

| H (m) | 600 | 650 | 700 | 750 | 800 | 850 | 900 | 950 | 1000 | 1050 |
|---|---|---|---|---|---|---|---|---|---|---|
| 100 | 0,01 | 0,01 | 0,01 | 0,01 | 0,02 | 0,02 | 0,02 | 0,02 | 0,02 | 0,02 |
| 200 | 0,02 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 | 0,04 | 0,04 | 0,04 | 0,04 |
| 300 | 0,04 | 0,04 | 0,04 | 0,04 | 0,05 | 0,05 | 0,05 | 0,06 | 0,06 | 0,06 |
| 400 | 0,05 | 0,05 | 0,05 | 0,06 | 0,06 | 0,07 | 0,07 | 0,07 | 0,08 | 0,08 |
| 500 | 0,06 | 0,06 | 0,07 | 0,07 | 0,08 | 0,08 | 0,09 | 0,09 | 0,10 | 0,10 |
| 600 | 0,07 | 0,08 | 0,08 | 0,09 | 0,09 | 0,10 | 0,11 | 0,11 | 0,12 | 0,12 |
| 700 | 0,08 | 0,09 | 0,10 | 0,10 | 0,11 | 0,12 | 0,12 | 0,13 | 0,14 | 0,14 |
| 800 | 0,09 | 0,10 | 0,11 | 0,12 | 0,12 | 0,13 | 0,14 | 0,15 | 0,16 | 0,16 |
| 900 | 0,11 | 0,11 | 0,12 | 0,13 | 0,14 | 0,15 | 0,16 | 0,17 | 0,18 | 0,18 |
| 1000 | 0,12 | 0,13 | 0,14 | 0,15 | 0,16 | 0,17 | 0,18 | 0,19 | 0,20 | 0,20 |
| 1100 | 0,13 | 0,14 | 0,15 | 0,16 | 0,17 | 0,18 | 0,19 | 0,20 | 0,21 | 0,23 |
| 1200 | 0,14 | 0,15 | 0,16 | 0,18 | 0,19 | 0,20 | 0,21 | 0,22 | 0,23 | 0,25 |
| 1300 | 0,15 | 0,16 | 0,18 | 0,19 | 0,20 | 0,22 | 0,23 | 0,24 | 0,25 | 0,27 |
| 1400 | 0,16 | 0,18 | 0,19 | 0,20 | 0,22 | 0,23 | 0,25 | 0,26 | 0,27 | 0,29 |
| 1500 | 0,18 | 0,19 | 0,20 | 0,22 | 0,23 | 0,25 | 0,26 | 0,28 | 0,29 | 0,31 |
| 1600 | 0,19 | 0,20 | 0,22 | 0,23 | 0,25 | 0,27 | 0,28 | 0,30 | 0,31 | 0,33 |
| 1700 | 0,20 | 0,22 | 0,23 | 0,25 | 0,27 | 0,28 | 0,30 | 0,31 | 0,33 | 0,35 |
| 1800 | 0,21 | 0,23 | 0,25 | 0,26 | 0,28 | 0,30 | 0,32 | 0,33 | 0,35 | 0,37 |
| 1900 | 0,22 | 0,24 | 0,26 | 0,28 | 0,30 | 0,31 | 0,33 | 0,35 | 0,37 | 0,39 |
| 2000 | 0,23 | 0,25 | 0,27 | 0,29 | 0,31 | 0,33 | 0,35 | 0,37 | 0,39 | 0,41 |
| 2500 | 0,29 | 0,32 | 0,34 | 0,37 | 0,39 | 0,41 | 0,44 | 0,46 | 0,49 | 0,51 |
| 3000 | 0,35 | 0,38 | 0,41 | 0,44 | 0,47 | 0,50 | 0,53 | 0,56 | 0,59 | 0,61 |
| 3500 | 0,41 | 0,44 | 0,48 | 0,51 | 0,55 | 0,58 | 0,61 | 0,65 | 0,68 | 0,72 |
| 4000 | 0,47 | 0,51 | 0,55 | 0,59 | 0,62 | 0,66 | 0,70 | 0,74 | 0,78 | 0,82 |

**Die in der Tabelle angegebenen Korrektionswerte $K_{g\varphi}$ (in mm QS, mb oder hPa) sind vom Barometerstand abzuziehen.**

The corrections of the table $K_{g\varphi}$ (in mm of mercury, mb or hPa) should be deducted from the barometer reading.

Les corrections du tableau $K_{g\varphi}$ (en mm de mercure, mb ou hPa) sont à déduire du niveau barométrique.

Las correcciones $K_{g\varphi}$ (en mm de mercurio, mb o hPa) de la tabla hay que deducirlas de la altura barométrca.

BA 604 Tab. III

**Korrektion der Barometerstände (mm QS, mb oder hPa) infolge der Kapillardepression.**

Correction of the Barometer Readings (mm Hg, mb or hPa) pursuant to the Capillary Depression.

Correction de la hauteur capillaire (mm Hg, mb ou hPa) en conséquence de la dépression capillaire.

Corrección de las alturas barométricas (mm Hg, mb o hPa) a causa de la depresión capilar.

| | Kuppenhöhe (für 8mm Rohr-Ø) in mm, mb oder hPa<br>Height of the top (for 8mm. tube-Ø) in mm, mb or hPa<br>Hauteur de la courbure (pour 8mm. Ø de tuyau) en mm, mb ou hPa<br>Altura cumbre (para 8mm. Ø de tubo) en mm, mb o hPa | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 1,0 | 1,1 | 1,2 | 1,3 | 1,4 |
| $K_k$ (mm QS) | 0,24 | 0,29 | 0,35 | 0,41 | 0,46 | 0,51 | 0,56 | 0,60 | 0,64 | 0,68 | 0,71 |
| $K_k$ (mb, hPa) | 0,24 | 0,30 | 0,36 | 0,41 | 0,47 | 0,52 | 0,57 | 0,63 | 0,68 | 0,73 | 0,77 |

| | Kuppenhöhe (für 8mm Rohr-Ø) in mm, mb oder hPa<br>Height of the top (for 8mm. tube-Ø) in mm, mb or hPa<br>Hauteur de la courbure (pour 8mm. Ø de tuyau) en mm, mb ou hPa<br>Altura cumbre (para 8mm. Ø de tubo) en mm, mb o hPa | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 2,0 | 2,1 | 2,2 | 2,3 | 2,4 |
| $K_k$ (mm QS) | 0,74 | 0,77 | 0,80 | 0,82 | | | | | | |
| $K_k$ (mb, hPa) | 0,81 | 0,85 | 0,89 | 0,93 | 0,96 | 0,99 | 1,02 | 1,05 | 1,07 | 1,09 |

**Die in der Tabelle angegebenen Korrektionswerte $K_k$ (in mm QS, mb oder hPa) sind zum Barometerstand zuzuzählen.**

The corrections $K_k$ (in mm Hg, mb or hPa) of the table should be added to the barometer reading.

Les corrections $K_k$ (en mm Hg, mb ou hPa) du tableau sont à ajouter au niveau barométrique.

Las correcciones $K_k$ (en mm Hg, mb o hPa) de la tabla hay que añadirlas a la altura barométrica.

**Differential pressure datasheet**

**Pressure Transmitter DS 1 and DS 2 for small pressure, vacuum and differential pressure** | **DS 1 DS 2**

# Pressure Transmitter DS 1 and DS 2 for small pressure, vacuum and differential pressure

**Ranges from 0/2.5 mbar to 0/1000 mbar**
**Analogue Output 0-10 V or 4-20 mA**


- Supervision of airblowers and airfilters
- Supervision of liquidlevels
- Controlling of aircurrents
- Pressure controlling in clean rooms
- Medical engineerings

## Description:

The pressure transmitter DS 1 and DS 2 are measuring positive and negative overpressure and differential pressure. An absolute pressure type is also available. The piezoresistive sensor gives a high reliability and precision. The type DS 2 (low cost version) is designed for applications with less requirements regarding the temperature-stability.

The DS 1 and DS 2 have a very robust aluminium housing, this guarantees also very good EMV-properties.

An individual inspection certificate is included in standard supply.

## Available Types:

| Type | Output Signal |
|---|---|
| DS 1-010 | 0-10 V (3-wire) |
| DS 1-420 | 4-20 mA (2-wire) |
| DS 2-010 | 0-10 V (3-wire) Low Cost Version |

## Pressure Ranges (Type DS 1):

| Pressure Range (mbar) | Pressure Range (kPa) | Over-pressure (mbar) | Linear error max. | Temp. error (max) 0-50°C | longtime Stability | Repeat-ability | Response time |
|---|---|---|---|---|---|---|---|
| 0 - 2.5 | 0 - 0.25 | 250 | ± 1% FS | ± 2% FS | 2% / year | 0.3% FS | 1 ms |
| 0 - 5 | 0 - 0.5 | | | | | | |
| 0 - 10 | 0 - 1 | 375 | | ± 1% FS | 1% / year | 0.2% FS | |
| 0 - 25 | 0 - 2.5 | | | ± 0.7% FS | 0.1%/year | 0.1% FS | |
| 0 - 100 | 0 - 10 | 750 | ± 0.6% FS | | | | |
| 0 - 250 | 0 - 25 | 1125 | | | | | |
| 0 - 500 | 0 - 50 | triple | | | | | |
| 0 - 1000 | 0 - 100 | | | | | | |
| 700-1100 **absolute** | 70-110 **absolute** | triple | ±0.9 mbar | ±2.3 mbar | 0.1%/year | 0.1% FS | 1 ms |

**DRUCK & TEMPERATUR Leitenberger GmbH**
Bahnhofstr. 33 • D-72138 Kirchentellinsfurt • Germany
Tel.: 0 71 21 - 9 09 20 - 0 • Fax: 0 71 21 - 9 09 20 - 99
E-Mail: dt-info@leitenberger.de • http://www.leitenberger.de

TÜV CERT DIN EN ISO 9001

**Pressure Transmitter DS 1 and DS 2 for small pressure, vacuum and differential pressure** **DS 1 DS 2**

## Technical Data (Type DS):

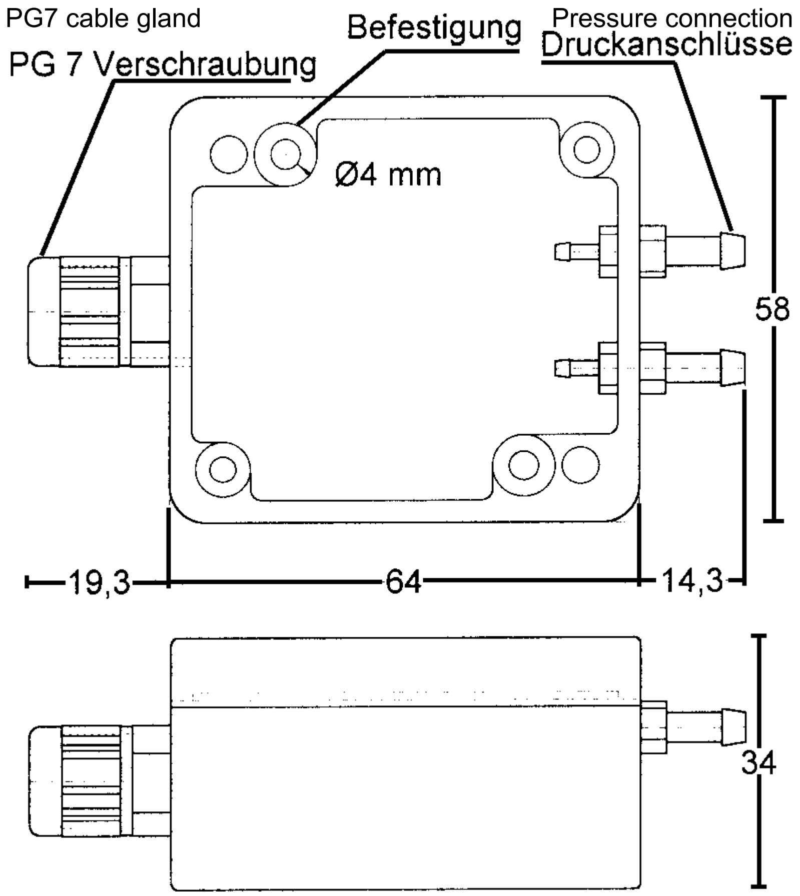
| | | | |
|---|---|---|---|
| Operating temperature: | -20°C to +70°C | | |
| Hysteresis: | 0.1% | | |
| Compatible media: | Air, non-corrosive gases | | |
| Output signals: | Type DS 1-010: 0-10 V | RL ≥ 2 kΩ | Power supply: 15...24 VDC |
| | Type DS 1-420: 4-20 mA | RL ≤ 400 Ω | Power supply: 10...30 VDC |
| Electrical connection: | Depht clamps for 0.14-1.5 mm² | | |
| Cable thread joint: | PG 7 | | |
| Pressure connections: | 2 connections for hose with 4 mm inner diameter | | |
| Weight: | appr. 160 g | | |
| Protection: | IP 65 | | |

## Wiring:

| | | |
|---|---|---|
| Type DS 1-010: | printclamp 1: | +15...+24 VDC |
| | printclamp 2: | output 0-10 V |
| | printclamp 3: | GND |
| Type DS 2-010: | printclamp 1: | +15...+24 VDC |
| | printclamp 2: | GND (Masse) |
| | printclamp 3: | output 0-10 V |

Type DS 1-420: Because of a special additional circuit, the pressure transmitter cannot be damaged by wrong connection. Both pins are exchangable

## Dimension:




**DRUCK & TEMPERATUR Leitenberger GmbH**
Bahnhofstr. 33 • D-72138 Kirchentellinsfurt • Germany
Tel.: 0 71 21 - 9 09 20 - 0 • Fax: 0 71 21 - 9 09 20 - 99
E-Mail: dt-info@leitenberger.de • http://www.leitenberger.de

DIN EN ISO 9001

**Balance operators manual**

# AEROLAB

3-component Sting Balance, Model Positioning System and Data Acquisition System

# Operation Manual

for

Escola Tècnica Superior d'Enginyeries Industrial i Aeronàutica de Terrassa

UNIVERSITAT POLITÈCNICA DE CATALUNYA

April 2010

## Contents


Introduction ..... 3
- 3-component "Sting" Balance ..... 3
- Model Positioning System ..... 4
- Data Acquisition (DAQ) System ..... 4

Specifications and Warnings ..... 4
- 3-component "Sting" Balance ..... 4
- Model Positioning System (MPS) ..... 6
- Data Acquisition (DAQ) System ..... 6

Software ..... 6
Model Installation ..... 8
- Warning ..... 8
- Model Mounting on Sting Balance ..... 8
- Model Mounting Socket ..... 10

Operation ..... 11
- Sting Balance Installation and Removal ..... 11
- Sting Balance Configuration Changes ..... 12
- DAQ Executable (.exe) Application ..... 14

Pitch Angle Zero Set ..... 18
Contact AEROLAB ..... 22
Warranty ..... 22
Balance Calibration Results ..... 23
- Normal Force Response (Normal) ..... 24
- Axial Force Response (Normal) ..... 25
- Pitching Moment Response (Normal) ..... 26
- Side Force Response (Side) ..... 27
- Axial Force 2 Response (Side) ..... 28
- Yawing Moment Response (Side) ..... 29
- Calibration Matrices ..... 30

Figures ..... 31
- Sting Balance Plug Pin-out ..... 31

# Introduction

In 2009, UPC ordered a 3-component "sting" internal Force/Moment balance, a top-mounted manually-operated (pitch angle) model positioning system and a custom-made Data Acquisition (DAQ) System from AEROLAB LLC, Laurel, Maryland, U.S.A.

## 3-component "Sting" Balance

The balance is made of two highly-machined pieces of hardened stainless steel:
- The 3/8-inch (9.525 mm) diameter "sting" and,
- the parallelogram base.

High-quality, professionally-applied strain gages allow the balance to sense two Forces and one Moment – Axial Force, Normal Force, and Pitching Moment. The balance can be quickly reconfigured to allow sensing of Axial Force, Side Force and Yawing Moment. As such, the device is effectively a 5-component balance.

This unique AEROLAB design incorporates several features to protect the delicate subcomponents:
- A machined sleeve protects the strain gages and wires of the cylindrical sting.
- A machined guard protects the strain gages and wires of the Axial Force parallelogram base.
- A travel stop in the Axial Force parallelogram protects the balance against damage from abnormally large Forces in the Axial direction (as would result from mishandling such as bumps, pulls or pushes).

A machined bar is supplied with the balance for calibration purposes. This bar is unique to the balance and must not be misplaced – no other bar will function properly.

The balance requires a data acquisition system to operate.

The balance should never be disassembled. Contact AEROLAB with all questions and concerns.

### Model Positioning System

The supplied model positioning system (MPS) was designed specifically for the 3-component balance. A manually-operated gearbox allows fine adjustment of pitch angle. A potentiometer is used to sense pitch angle for the supplied data acquisition system (DAQ). The system was designed to allow pitch angle adjustment about a fixed point in the test section. The fixed point nearly coincides with the forward-most end (tip) of the balance sting. Careful design of a test model will ensure the model remains centered in the wind tunnel airflow throughout a pitch sweep.

### Data Acquisition (DAQ) System

The supplied DAQ was designed and assembled specifically for UPC. It employs a Dell desktop computer, National Instruments hardware (SCXI-1000 chassis, SCXI-1314 terminal block, PCI 6221 data acquisition card) and a LabVIEW application.

The DAQ excites the balance and MPS potentiometer. It then applies calibration matrices and displays the output in your choice of SI or American units. Data can be sampled in a number of ways and saved in a tab-delimited text file for use with many common applications such as MatLab and Microsoft Excel.

## Specifications and Warnings

### 3-component "Sting" Balance

The balance has specific load limits. Do not exceed these limits or permanent damage will result.

| | | | |
|---|---|---|---|
| Normal Force (Side Force): | 25 | lbs. | (111.2 N) |
| Axial Force: | 10 | lbs. | (44.5 N) |
| Pitching Moment (Yawing Moment): | 50 | inch-lbs. | (5.65 N-m) |

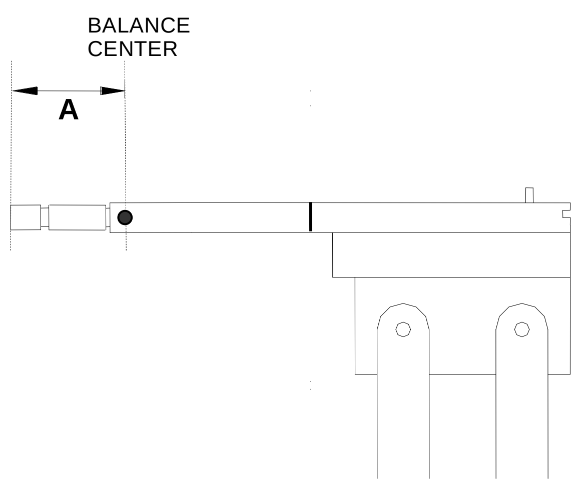
Combined Loading – <u>The balance was not designed for high combined loads.</u> Rather, it was designed for increased sensitivity. As such, deflection of the balance must be considered. If deflection is anticipated, make certain the model mounting fixture does not contact the balance at any point. Additionally, consider deflection and the possibility of balance contact (fouling) in post-test data reduction and analysis. If deflection is found to be a problem, redesign your test accordingly.

The balance is a delicate instrument. It should never be overloaded or handled roughly. The wires within the black protective tubing are extremely thin and fragile. Never pull or twist the tubing.

The DAQ excites the strain gages as follows:
Normal (Side) Force: 5VDC
Axial Force: 9VDC
Pitching (Yawing) Moment: 5VDC

In the event this balance is used with a different DAQ, never exceed these Voltages.

The electrical center of the balance (the longitudinal point about which the balance measures Pitching Moment) is located as follows. In the event the balance gages are replaced, this measurement will change. Additionally, a new calibration bar would need to be machined.




Dimension "A" is 1.126 inches (2.860 cm) for balance PGB-10-07.

Test models must be mounted to the sting as described in "Model Installation" on page 8. Incorrectly mounted models can lead to permanent damage. Please follow the provided directions carefully.

### Model Positioning System (MPS)

The MPS is made largely of anodized 6061 aluminum. All fasteners are (standard American) Unified screw series.

The pitch angle adjustment range is approximately +/- 20°. Never exceed this range or permanent damage can result to the MPS or the balance wires.

The MPS was deigned to be fastened to the bottom of an existing test section yaw table. Two threaded 10-32 screw holes have been tapped in the top plate for this purpose. If 10-32 screws are not available in your area, contact AEROLAB.

### Data Acquisition (DAQ) System

All elements of the DAQ operate on 220VAC 50 Hz. Standard Spanish type F plugs are provided.

The Dell desktop computer is an Optiplex 360 model. Specifications for the computer are available from Dell or from the computer.

The supplied DAQ application is an executable LabVIEW file. No license is required. The source code is not supplied. Modifications to the DAQ application are possible at additional cost. Contact AEROLAB for details.

## Software

The provided executable DAQ application was written in LabVIEW. No license is required. The source code is not included with the system.

The custom-made UPC Data Acquisition (DAQ) system provides a first-order compensation for balance interactions. The balance was thoroughly tested and calibrated at AEROLAB prior to shipment. The interaction matrix is located in the "interaction.ini" file. The matrix should never be changed unless a thorough calibration by a qualified technician deems it necessary.

The MPS pitch angle potentiometer was thoroughly tested and calibrated at AEROLAB prior to delivery. The Voltage/position correlation is located in the (provided) Measurements and Automation (MAX) software. The correlation should never be changed unless a thorough calibration by a qualified technician deems it necessary.

The software monitors four (4) channels: Normal Force, Axial Force, Pitching Moment and pitch angle. The National Instruments (NI) data acquisition card (PCI 6221) is set to sample at a rate of 10 Kilohertz. The channel information is not multiplexed. 100 voltage samples are collected per acquisition cycle per channel, which can be nominally delayed using the program interface. The means and standard deviations are scaled through the calibration data/MAX hardware and reported to the Operator (interface display) as well as recorded in the data file.

The selectable delay is not real-time. Rather, it is a minimum delay between acquisition cycles. The selected delay is also the interface display refresh rate.

The bridge voltage and conditioning are provided through the SCXI 1520 module (National Instruments).

All loop output is calculated in U.S. units. If SI units are selected, the U.S. units are converted as follows:

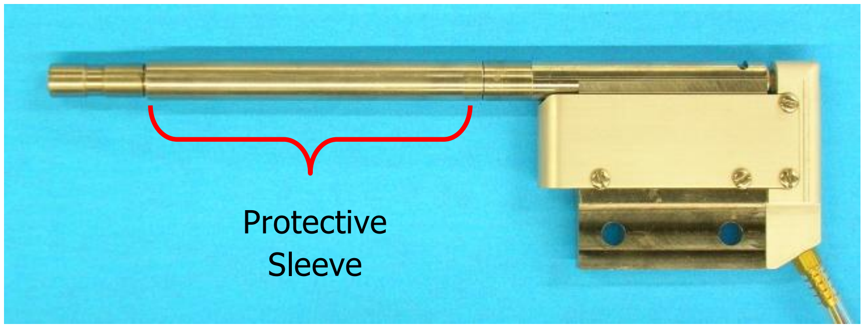
- 4.44822 N/$lb_f$
- 0.112985 N-m/in-$lb_f$

Note – The data file will be written in only one unit system – the unit system selected when the **Save to File** button is pressed.

The **Take Data** button adds data to a buffer (cache). Buffer data is not written to file until the **Save to File** button has been pressed. The buffer will be emptied after each save.

# Model Installation

## Warning

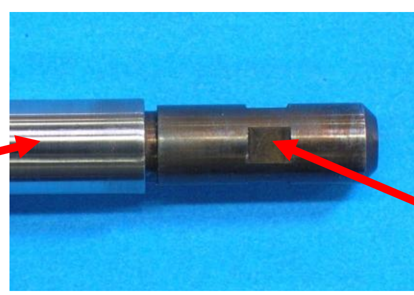



**Warning:** The sting balance is provided with a protective sleeve. The sleeve protects the balance's delicate strain gages and wires. This sleeve should never be removed. The NUMBER ONE cause of balance damage is incorrectly mounted models. NEVER MOUNT MODELS TO THE PROTECTIVE SLEEVE!!

## Model Mounting on Sting Balance

Note: There are four "flats" machined into the tip of the sting balance. These flats serve an important purpose – they are locations on the sting where all model mounting setscrews are to be tightened. Tightening a setscrew on any other surface will potentially make model installation and removal difficult or impossible! DO NOT tighten setscrews outside of the setscrew flats.

Protective sleeve – models should never contact this at any point.



setscrew flat

Typically, sting-mounted models (drag models, airplane models, etc) are held in place with a setscrew.

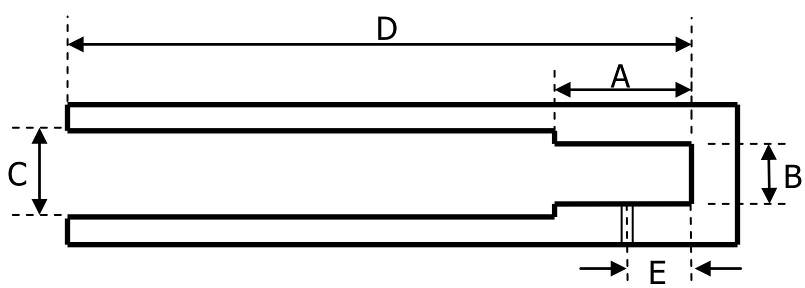
AEROLAB Drag Model Set (not included)

NOTE: The weight of the model mounted on the balance affects all three component readings as a function of angle of attack. For accurate test results, it is necessary to perform a wind-off pitch sweep to evaluate gravitational corrections for each individual component over the planned pitch angle range.

## Model Mounting Socket

To mount a test model on the sting, it must be equipped with a cylindrical socket. As an example, the socket should be designed similar to the calibration bar - the forward portion of the socket should be reamed 0.375 inches (9.525 mm) in diameter for a snug fit and the rear portion of the socket must be 5/8 inches (15.875 mm) diameter or more to provide ample clearance (to avoid interference with any other part of the sting when being deflected under load). The position of the socket within the model should be determined relative to a reference point on the sting - this way, the moment transfer distance can be determined.  The position of the mounting setscrew should be determined when the sting is fully inserted into the socket ("bottomed out", so to speak). THE SETSCREW LOCATION MUST NOT ALLOW THE SETCREW TO BE TIGHTENED IN ANY LOCATION OTHER THAN ON ONE OF THE FOUR SETSCREW FLATS (see "Model Mounting on Sting Balance" on page 8 for a description).  Contact AEROLAB for more information.

- length A should be 0.900 inches (22.86mm)
- diameter B should be 0.375 inches (9.525mm)
- diameter C should be 0.625 inches (15.875mm)
- length D must not exceed 5.0 inches (127mm)
- length E is the distance between the end of the socket and the *center* of the set screw.  E must be 0.50 inches (12.7mm).  Use #8 set screws or smaller.



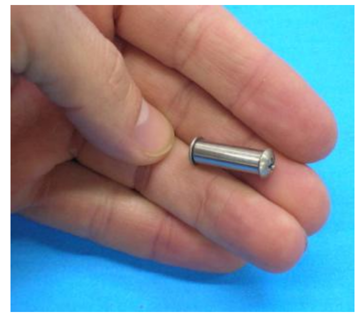

Custom model mounting socket

# Operation

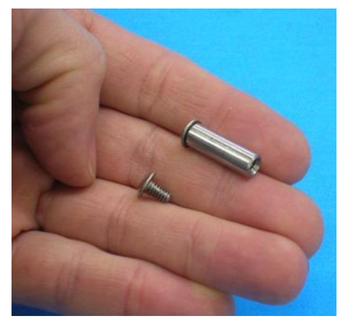
## Sting Balance Installation and Removal

The 3-component balance is mounted to the MPS with two machined pins. The pins have threaded 6-32 center holes and are *loosely* retained by two Truss Head screws.

Installation:

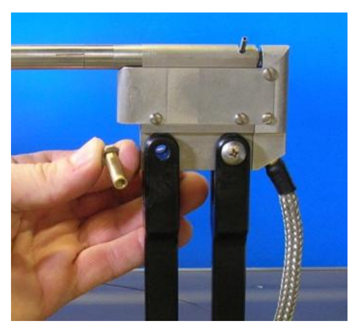


1. Identify the mounting pins (2).



2. Using two Philip head screwdrivers, remove a screw from each pin. Allow one screw to remain in each pin.

3. Position the balance between the vertical ends of the MPS and align the pin holes.



4. Slide the pins through the holes.

5. Install the two Truss Head screws you removed in step #2.
6. Using two Philip head screwdrivers gently tighten the screws.
7. Plug the balance to the NI SCXI-1000 chassis.

Notes:

- The pins were machined to a very close tolerance. They are slightly snug in the balance and MPS holes. Do not force them! Do not hammer them into position!!
- A balance plug pin-out is provided in the Figures section of this manual.

Removal:

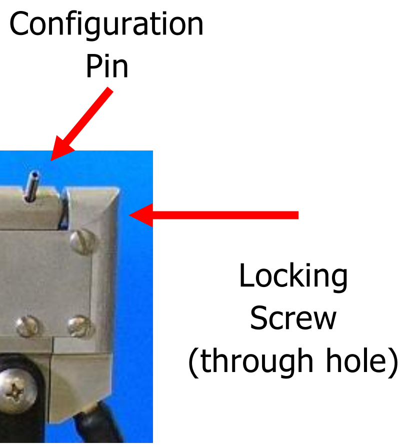
Removal is the opposite of installation.

## Sting Balance Configuration Changes

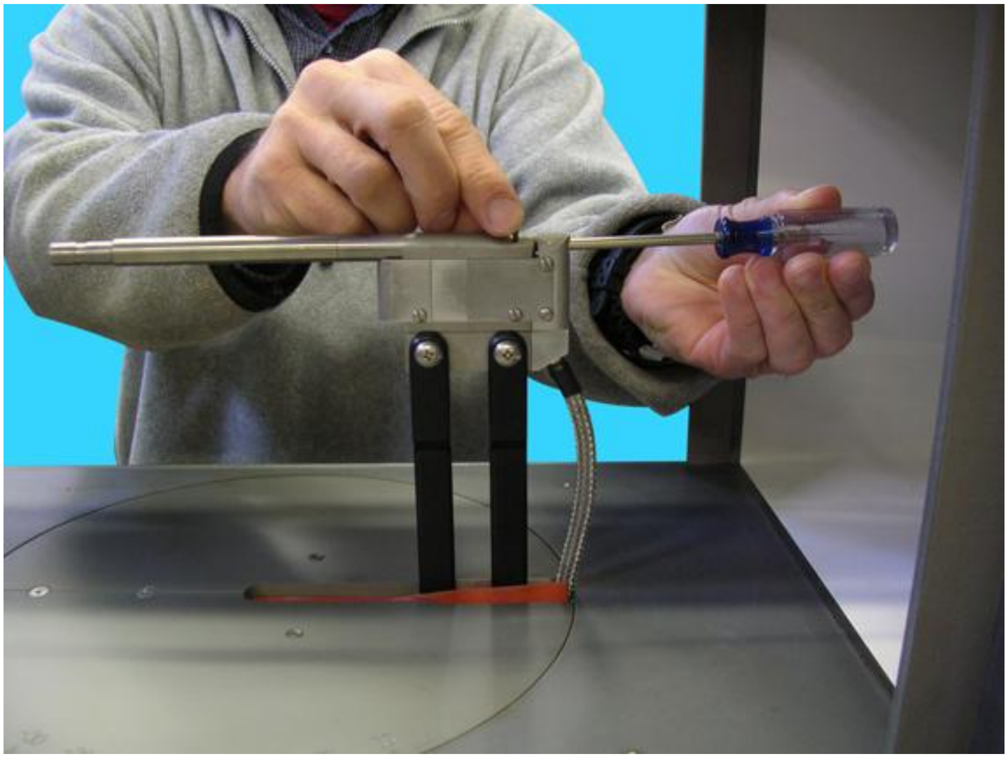
The balance can be quickly reconfigured to measure either Normal/Axial/Pitching or Side/Axial/Yawing.

To do so:

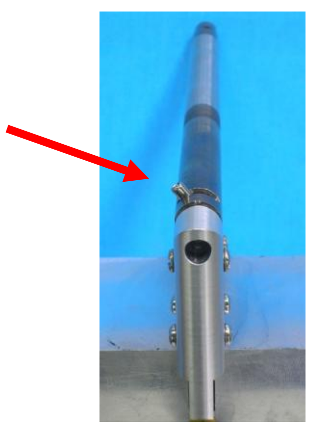
1. Locate the configuration pin.
2. Locate the configuration locking screw (Philips head).




3. Using a Philip head screwdriver, loosen the configuration locking screw (counter-clockwise). Caution – Loosen the locking screw only ½ turn! Loosening the screw more than ½ turn can cause permanent damage to the balance. The cylindrical "sting" portion of the balance is now free to rotate 90° within the parallelogram base.



4. Using finger pressure only, rotate the configuration pin to the opposite end of the slot and hold.
5. While holding finger pressure on the configuration pin, gently tighten the locking screw.

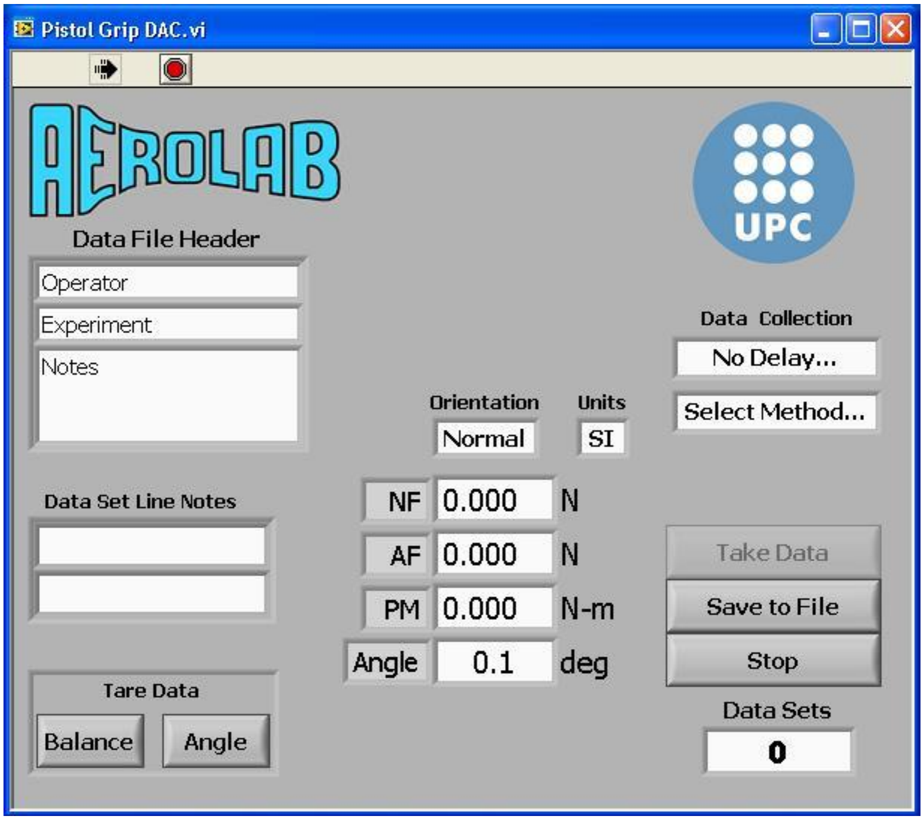


Configured for Normal/Axial/Pitching

Configured for Side/Axial/Yawing

**Note:** The pin *must be* rotated to its stop in either direction. Intermediate positions would invalidate the factory calibration because the strain gages would not be in proper orientation.

## DAQ Executable (.exe) Application




1. Connect the NI SCXI-1000 chassis and computer to power.
2. Connect the sting balance and the MPS potentiometer to the NI chassis.
3. Switch ON the SCXI-1000 chassis.


4. Start the computer.
5. Double click the DAQ icon on the desktop.  The application will begin to run immediately.
6. Allow 30 minutes for warm-up.  This will ensure stable operation of the balance and electronic hardware.
7. If you wish, enter information in the **Data File Header** fields.  This information will be entered as a header to the data file – only once for each

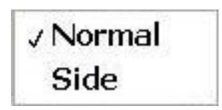
saved data file. Note – The **Data File Header** is not written to the data file until the **Save to File** button has been push. As such, information in the fields is ignored until the **Save to File** button is pushed.

8. If you wish, enter information in the **Data Set Line Notes** fields. The Operator can enter up to two notes for each data set line.

   **Notes** - Each data set line contains 11 items: balance orientation (Normal or Side), line notes (2), balance outputs (3), pitch angle, balance output standard deviations (3) and pitch angle standard deviation.
   - Line notes are written to each data set every time the **Take Data** button is pressed.

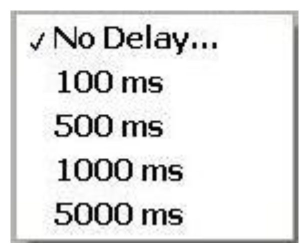
9. Using standard, acceptable wind tunnel testing practices, tare the balance and MPS as necessary. Warning – <u>Do not overload the balance</u>! Doing so can cause permanent damage! Damage caused by overloading is not covered by the warranty.

10. Select the balance orientation. To do so, click in the **Orientation** field and select the current balance orientation – Normal (Normal/Axial/PM) or Side (Side/Axial/YM).




11. Select the preferred unit system – SI or US (American). To do so, click the **Units** field and select.


12. Select the data collection delay. To do so, click the upper field. You will be given 4 choices. If “No Delay” is selected, output will be displayed and recorded as fast as the application and hardware are able to loop.




Note – The interface display will also be refreshed at the rate you select in this field.

13. Select the collection method. To do so, click the lower field.


You have three choices:

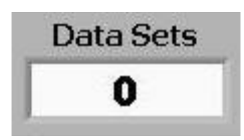
Note – The **Take Data** button will be gray and inactive until a recording method is selected.


- Stream Data – This choice will continuously add data to the buffer until stopped. To start streaming data, click the **Take Data** button. To stop streaming data, click the **Take Data** button again.

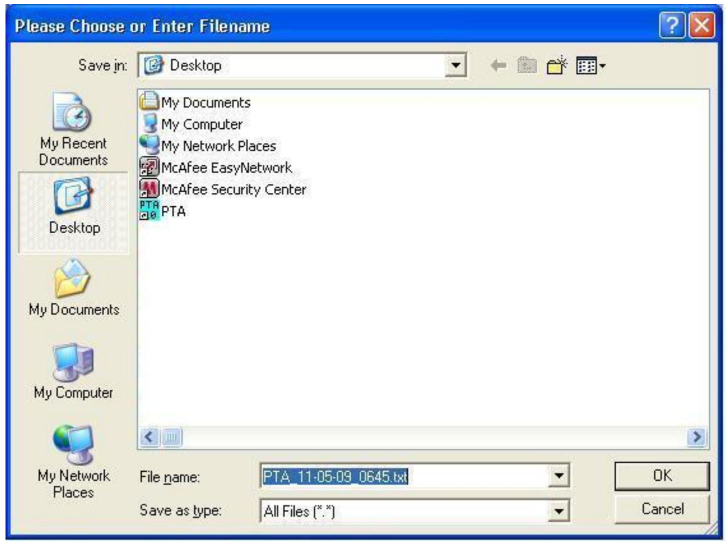

- Sample Data – This choice allows the Operator to select the number of data sets to be recorded. Upon selection of this choice, a Samples window will appear. To enter the number of samples you wish to be taken, click inside the **Samples** window and enter the value. Click **Take Data** to begin taking data.


- Snapshot Data – This choice allows data to be taken only once each time the **Take Data** button is pressed.

  Note – Every time a data set is added to the data buffer, the **Data Sets** window will increment by one. The Operator cannot make selections in this window directly – only indirectly by taking more data.



14. To save data, press the **Save to File** button. The following window will appear.

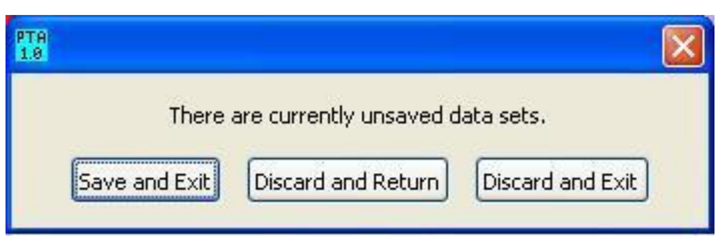




Select a file name and destination. By default, the destination will be the **desktop** and the name will be DAQ_date_time.txt. Click OK to save the data currently held in the data buffer.

Note - Saving Data will clear the data buffer and reset the Sets Total window to zero.

15. Exit the program. To do so, click the **Stop** button.


Note – If the **Data Sets** window (data buffer) is non-zero when the **Stop** button is pressed, the following warning will appear.



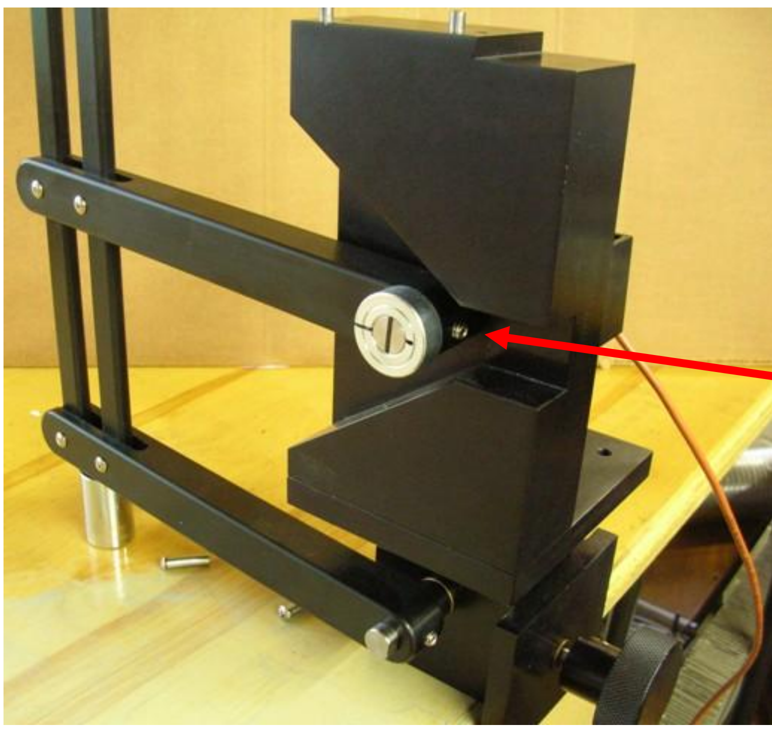

Select from the three options carefully. If you choose to discard data, it will be **lost and not recoverable**.

# Pitch Angle Zero Set

The potentiometer on the MPS was calibrated at AEROLAB. No further calibration is necessary. If, however, an adjustment to the zero position is necessary, this section describes the correct procedure. The zero position was set at AEROLAB prior to shipment with the balance parallel to the top surface (mounting surface) of the MPS.

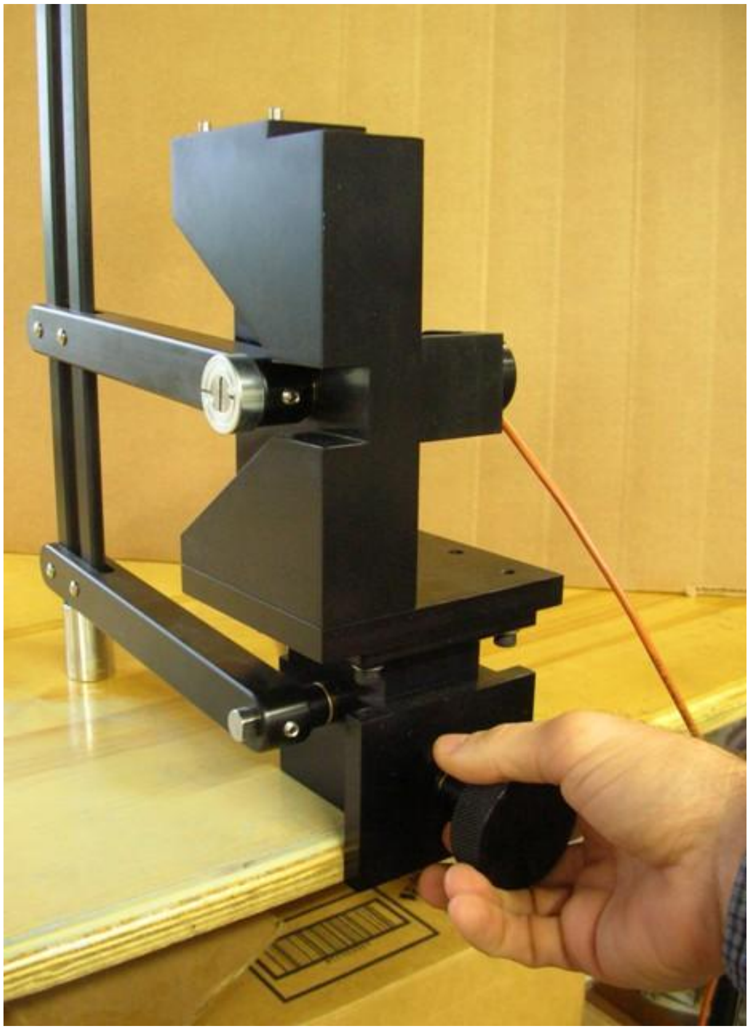
To adjust the zero set:

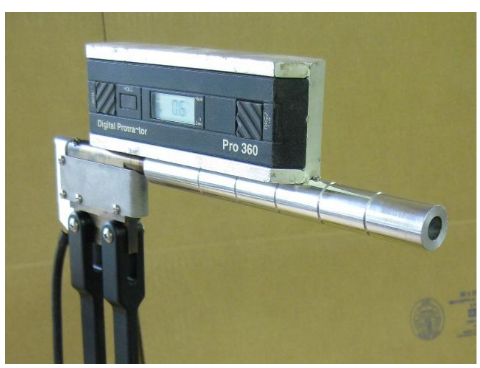
1. Start the DAQ application.
2. Locate the potentiometer shaft set screw.



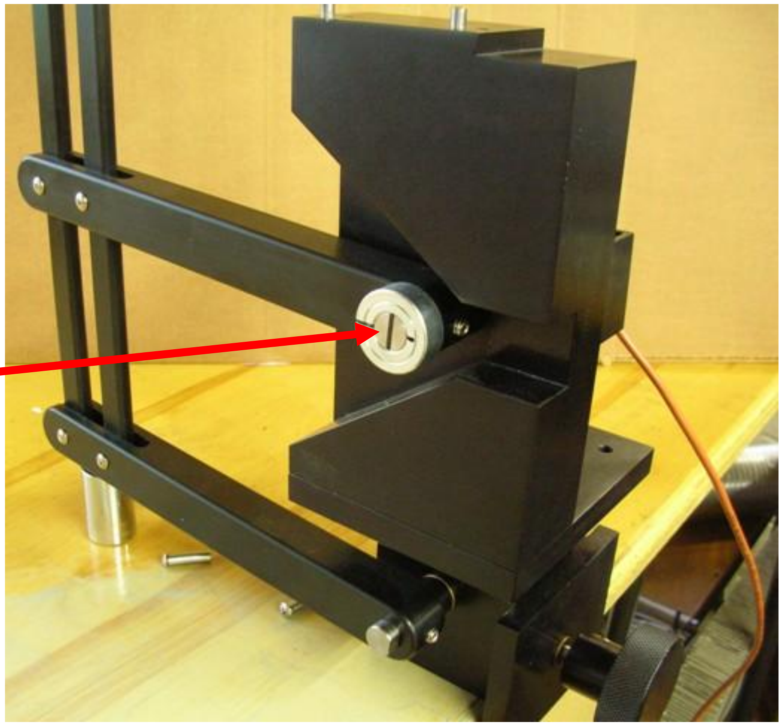

3. Loosen the set screw.

4. Using the hand knob, position the balance as you require. Note – If you plan to use a digital inclinometer, the calibration bar is a good reference surface.



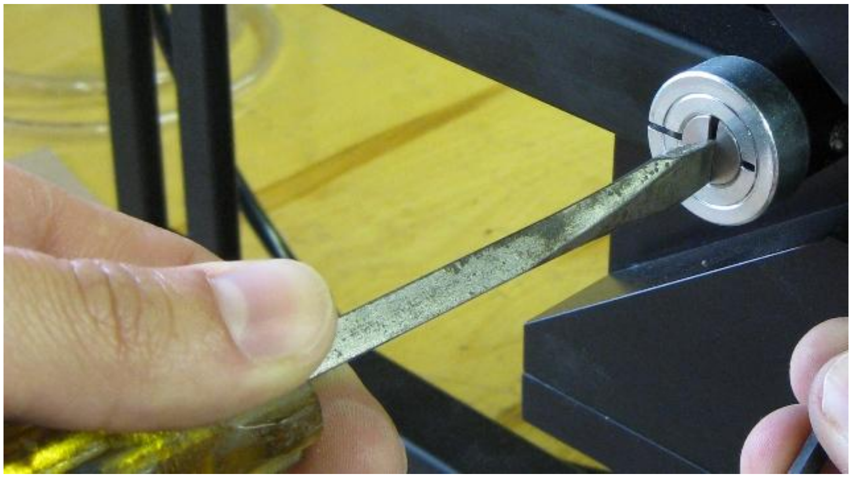


5. Locate the potentiometer adjustment slot.



Potentiometer
Adjustment Slot

6. While watching the DAQ pitch angle display, use a flat-blade screwdriver to adjust the potentiometer shaft to your needs.



7. Gently tighten the potentiometer shaft set screw when finished.

## Contact AEROLAB

AEROLAB LLC
9580 Washington Boulevard
Laurel, Maryland 20723 U.S.A.

(Telephone) 301-776-6585
(Facsimile) 301-776-2892

(Internet) http://www.aerolab.com

(e-mail) aerolab@aerolab.com

## Warranty

AEROLAB guarantees all products to be free of defects for a period of one year following purchase. Warranty claims must be returned to AEROLAB, postage-paid. AEROLAB will repair or replace defective components or systems free of charge. The warranty does not cover misuse or neglect. Read all instructions carefully. Contact AEROLAB with any questions prior to use.

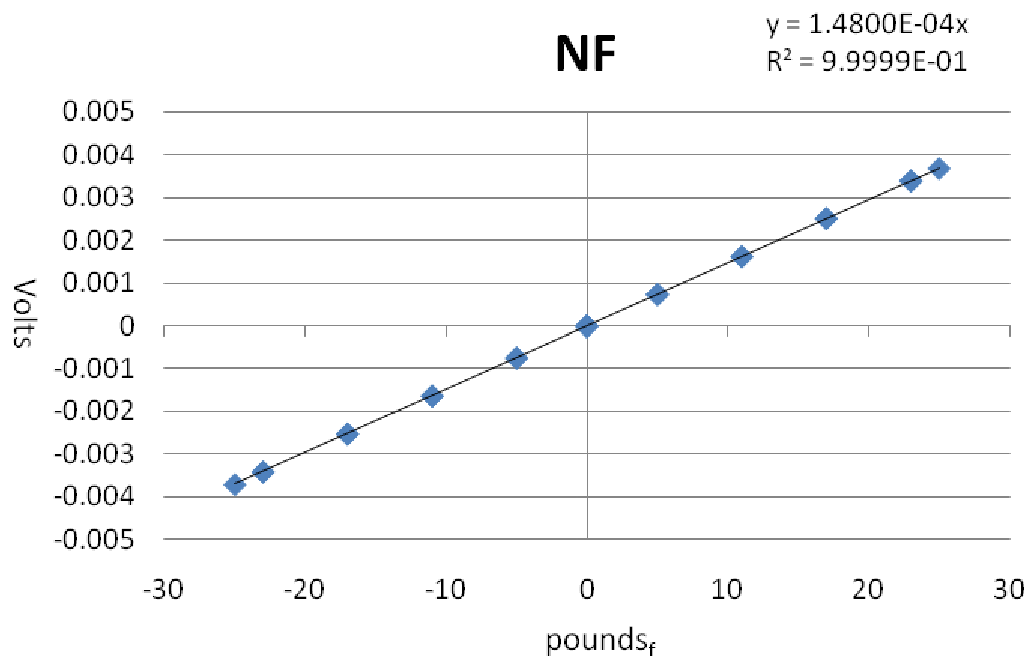
# Balance Calibration Results

Notes:

- "Normal" results are with the sting configured in the Normal position. (See "Sting Balance Configuration Changes" on page 12).
- "Side" results are with the sting configured in the Side position.
- All loadings are in $pounds_f$ (1 $pound_f$ = 4.4482 Newton) or $inch\text{-}pounds_f$ (1 $inch\text{-}pound_f$ = 11.2985 Newton centimeter).
- All Voltages are in Volts except where otherwise noted.
- Six single-Force/Moment loadings were performed. The loadings were Normal Force (NF), Axial Force (AF), Pitching Moment (PM), Side Force (SF), Axial Force (AF2), Yawing Moment (YM).
- The charts have been corrected for offset.
- The UPC sting balance is serial number PGB-10-07.

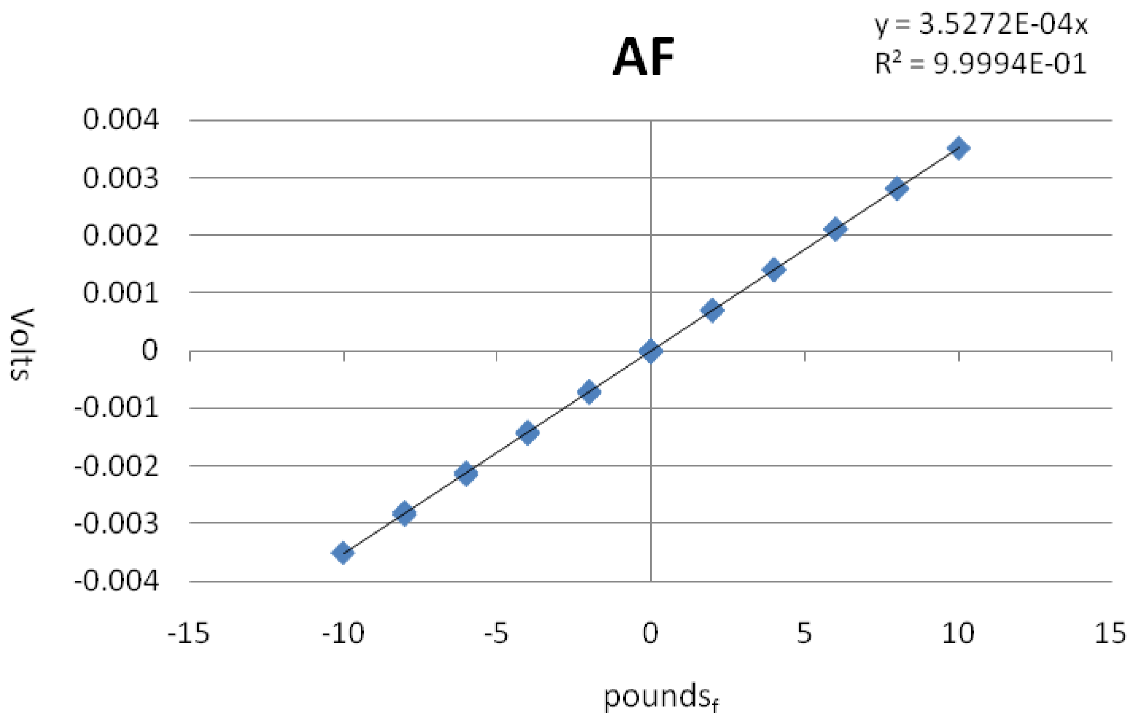
## Normal Force Response (Normal)

Normal Force response to Normal Force Application: $1.4800 \times 10^{-4}$ Volt/pound$_f$



## Axial Force Response (Normal)

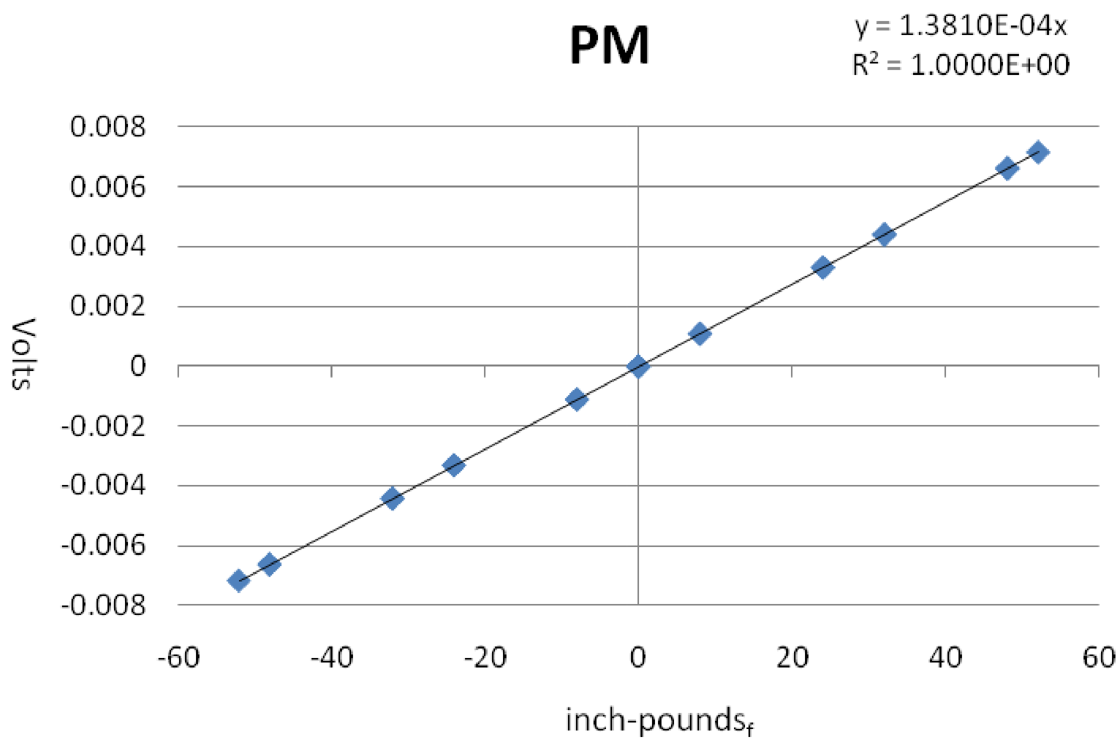
Axial Force response to Axial Force Application: $3.5272 \times 10^{-4}$ Volt/pound$_f$



## Pitching Moment Response (Normal)

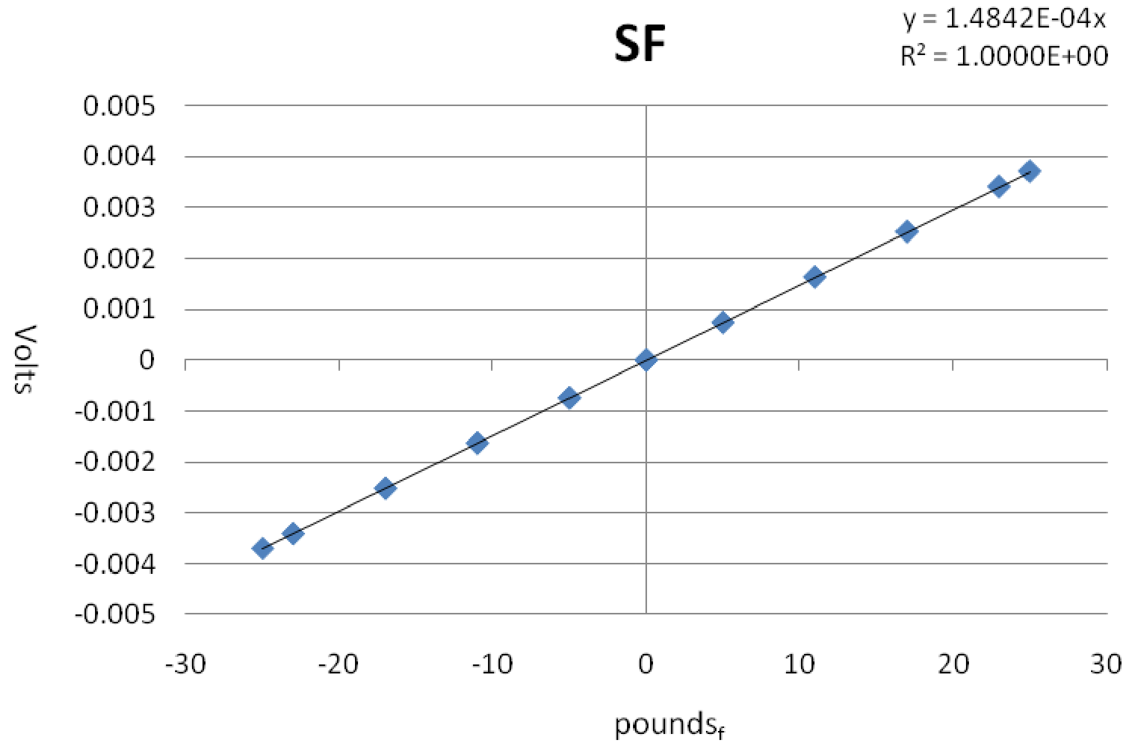
Pitching Moment response to Pitching Moment application:

$1.3810 \times 10^{-4}$ Volt/inch-pound$_f$



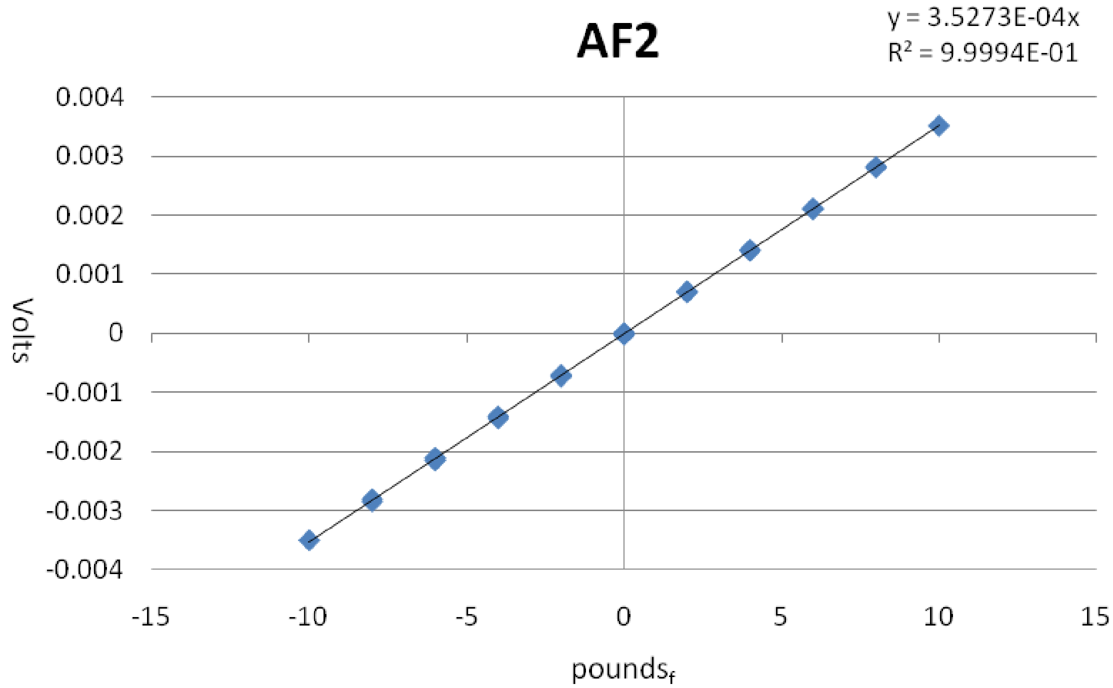
## Side Force Response (Side)

Side Force response to Side Force application: $1.4842 \times 10^{-4}$ Volt/pound$_f$



## Axial Force 2 Response (Side)

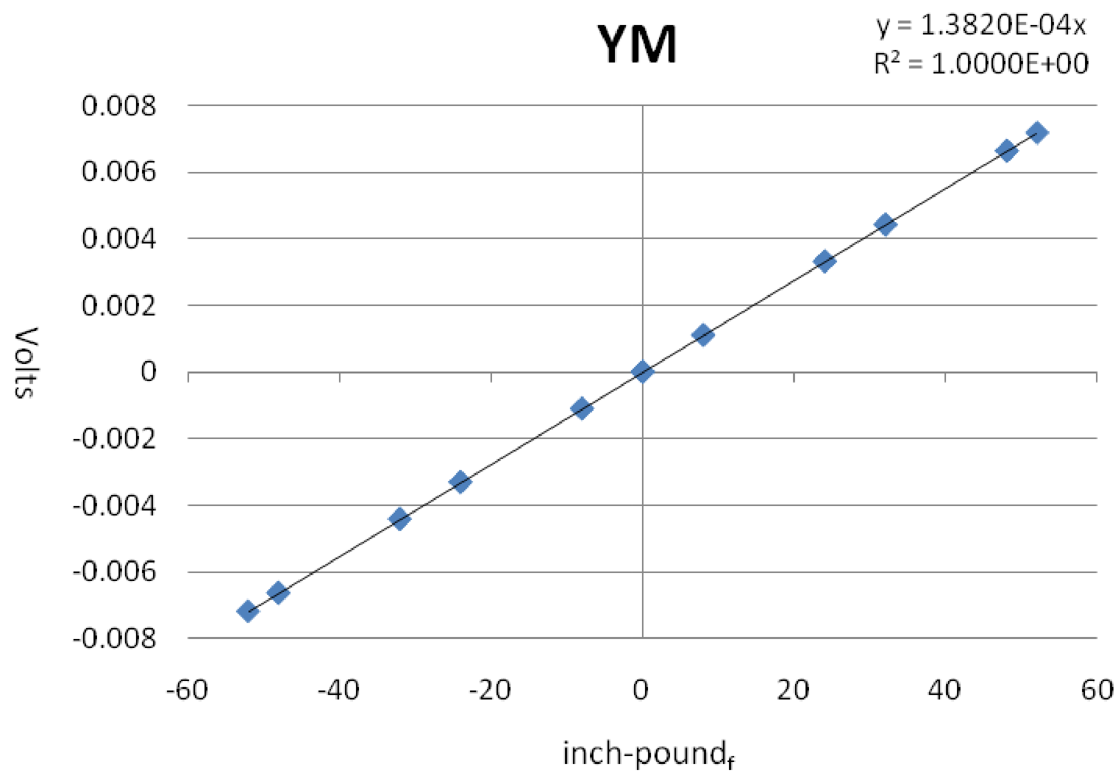
Axial Force 2 response to Axial Force 2 application: $3.5273 \times 10^{-4}$ Volt/pound$_f$



## Yawing Moment Response (Side)

Yawing Moment response to Yawing Moment application:

$1.3820 \times 10^{-4}$ Volt/inch-pound$_f$



## Calibration Matrices

| | | | | |
|---|---|---|---|---|
| Normal | C1_Inv | | | |
| | | 6.7560E+03 | -5.2869E+00 | 7.4421E+01 |
| | | 2.2505E+02 | 2.8335E+03 | 5.3566E+01 |
| | | -6.1646E+01 | -2.0545E+02 | 7.2273E+03 |
| | C1_Inv_C2 | | | |
| | | -3.4372E-04 | -1.2143E-04 | -4.3978E-04 |
| | | -8.4824E-03 | 1.9610E-05 | -4.6061E-04 |
| | | 1.5793E-03 | -3.6661E-02 | 1.2794E-04 |
| Side | C1_Inv | | | |
| | | 6.7432E+03 | 1.2594E+01 | 6.7554E+01 |
| | | 1.4484E+02 | 2.8351E+03 | 2.4817E+01 |
| | | -5.8909E+01 | -6.8244E+01 | 7.2282E+03 |
| | C1_Inv_C2 | | | |
| | | 9.1728E-04 | 3.0462E-03 | 6.3660E-04 |
| | | -1.8008E-04 | 1.1795E-03 | 1.4184E-04 |
| | | -4.2961E-04 | 2.0698E-02 | -2.8132E-04 |

Note:

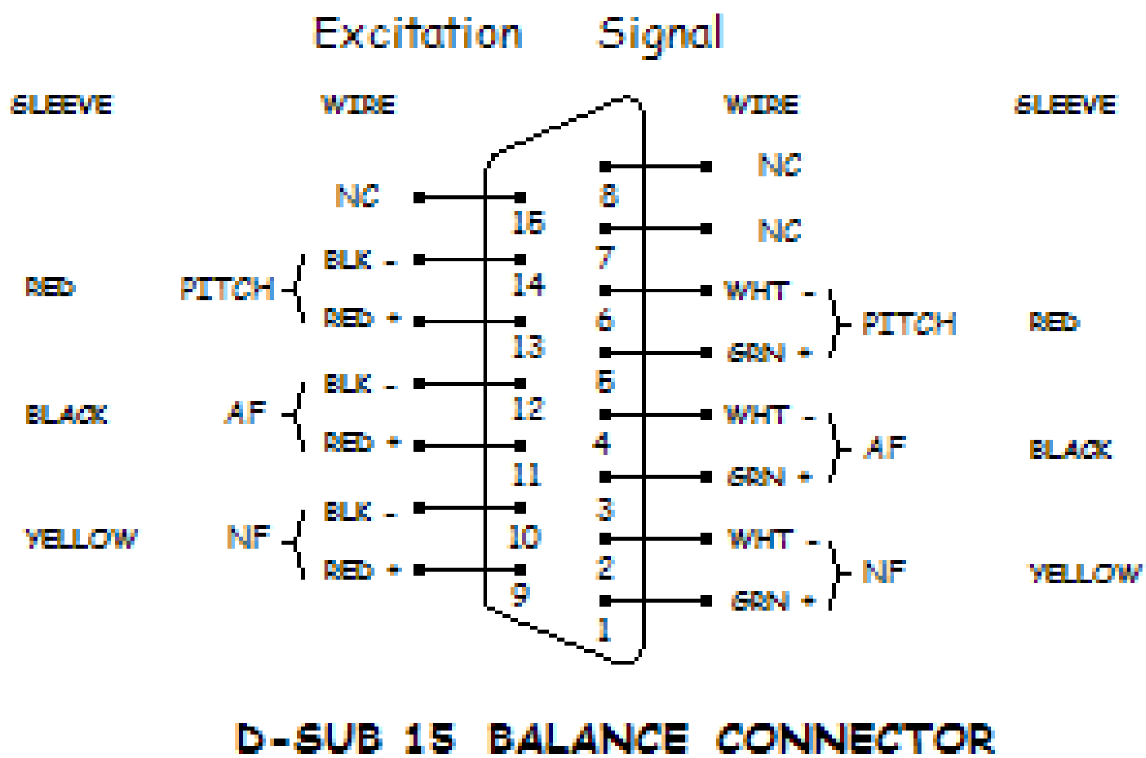
The calibration was performed following American Institute of Aeronautics and Astronautics (AIAA) standards. The nomenclature used in the above matrices follows the same standards. The AIAA published these standards in:

"Calibration and Use of Internal Strain-Gage Balances with Application to Wind Tunnel Testing" publication R-091-2003.

This document is available directly from the Institute at http://www.aiaa.org/

# Figures

## Sting Balance Plug Pin-out




Notes:

- The balance wires are grouped by three short pieces of heat-shrink tubing: Red for Pitching Moment, Black for Axial Force and Yellow for Normal Force.
- PITCH = Pitching Moment
- AF = Axial Force
- NF = Normal Force
- NC = no connection

**Differential pressure sensor calibration procedure**

# Práctica de laboratorio Nº 2: Calibración de un transductor de presión

## 1. Objetivos de la práctica

Mediante la realización de esta práctica se pretende que el alumno:

- Poniendo en práctica los conocimientos adquiridos en clase calibre un transductor de presión en un rango de trabajo determinado.
- Lleve a cabo un procedimiento típico de estimación de los errores que afectan la calibración.
- Se familiarice con el montaje experimental, con el conexionado de los instrumentos, con sus especificaciones técnicas más relevantes y con las complicaciones prácticas que muchas veces se presentan a la hora de realizar experimentos.

## 2. Breve descripción del montaje experimental

El montaje experimental se compone de una bomba manual de baja presión que se conecta, por un lado, a un micromanómetro de columna que será considerado como instrumento de referencia y, por otro lado, al transductor de presión cuya curva de calibración se desea obtener. El micromanómetro es marca Lambrecht, modelo 655M16, el fluido de trabajo es alcohol etílico ($\rho_{ref}$ = 0.8 g/cm$^3$ @ 20ºC) y su columna se coloca en posición vertical ($\beta$=90º) con el objetivo de aprovechar su rango máximo de trabajo (0-1600 Pa). El transductor escogido es un transductor diferencial de baja presión SETRA 239 de rango 0-0.5 psid. La lectura de la tensión de salida del transductor (0-10 vdc) se realiza mediante un sistema de adquisición de datos (NI PCI6259 16 bits) mientras que la lectura del micro-manómetro se lleva a cabo utilizando su escala graduada.

## 3. Procedimiento experimental a seguir

1. En primer lugar deberá verificarse que todos los instrumentos se encuentren correctamente conectados y en condiciones de ser utilizados. Es importante corroborar el **correcto nivelado** del micromanómetro y **que la columna de líquido indique cero** cuando no existe presión en el sistema (válvula de venteo abierta).

2. Antes de comenzar con las mediciones se deberá medir la temperatura ambiente. Este dato será necesario a fin de corregir la densidad del alcohol.
3. Con el sistema en condiciones de presión diferencial nula se tomará la lectura del primer cero del transductor. El sistema de adquisición de datos (rutina en labVIEW®) será ajustado para tomar 50 muestras con una frecuencia de 100 Hz.
4. A continuación se cierra la válvula de venteo y se realizan las siguientes mediciones en sentido ascendente (en unidades de escala): **50, 100, 200, 400, 800, 1200, 1600** y a continuación se miden en sentido descendente: **1200, 800, 400, 200, 100 y 50**. En cada punto de medición, en primer lugar deberá ajustarse la presión requerida, luego se tomará la lectura del manómetro (referencia) y a continuación se ejecutará la adquisición de datos de salida del transductor.
5. Finalmente se abre la válvula de venteo y se toma el segundo cero del transductor.
6. A continuación se tomará nota de las especificaciones técnicas del micromanómetro y del transductor y de todos aquellos datos de los instrumentos y del montaje que son necesarios para el post-proceso de las mediciones.

**Nota 1**: el sistema de adquisición de datos proveerá para cada lectura el valor medio y la desviación estándar. Se recomienda, durante el ensayo, tomar nota únicamente de la presión de referencia que corresponde a cada punto de medición. Luego, todos los datos podrán ser volcados en una misma planilla de cálculo para su procesamiento.

**Nota 2**: es una práctica recomendable y muy usual que de manera previa a la realización del ensayo se confeccione una planilla que permita tomar nota de todos aquellos datos de importancia que se consideren necesarios para registrar adecuadamente el experimento. La confección de dicha planilla quedará a criterio de cada grupo y no se incluirá en el informe.

## 4. Post-proceso de los datos

En primer lugar se deberá corregir la lectura del micromanómetro por cambios en la densidad debidos a temperatura. Con las presiones de referencia corregidas por temperatura, se deberá calcular la incerteza RSS en la medición del manómetro debido a la resolución de escala y la exactitud provista por el fabricante que denominaremos $\delta u^{man}$. Esta incerteza luego se trasladará a los resultados de la calibración.

A continuación se obtendrá la función de calibración $P_{fit} = f(V_{out})$ mediante un recta de regresión lineal entre la presión de referencia y la tensión de salida del transductor. Se trabajará con **tensiones netas** y la decisión de adoptar corrección de cero promedio o lineal dependerá del criterio de cada grupo de trabajo. Luego se estimará las incerteza RSS en la curva de calibración del transductor ($\delta u^{trans}$ en Pa) teniendo en cuenta:

- La incerteza de ajuste de la función de calibración ($\|\cdot\|_\infty$ de los residuos).
- Incertezas debidas a no-linealidad, histéresis y repetibilidad del transductor de presión[1].
- Incerteza debida a la resolución de la tarjeta de adquisición de datos.
- Incerteza en la estimación de las tensiones medias (CI 95%). Se tomará la máxima de los valores obtenidos.

De manera conjunta a la estimación de incertezas se deberá presentar una gráfica con la curva de calibración (puntos medidos + curva de regresión lineal, incluyendo la ecuación y el valor de determinación) y otra con la distribución de residuos (con sus signos).

Finalmente se pide expresar la incerteza total $\delta u^{trans}$ obtenida de calibración en %FSS del rango de presiones medidas. Adicionalmente se añadirá a $\delta u^{trans}$ la incerteza en la medición del manómetro $\delta u^{man}$ y se calcularán los mismos errores relativos[2]. Se pide extraer de estos resultados algunas conclusiones en cuanto a la importancia relativa de las incertezas tratadas en el experimento.

## 5. Preparación del informe

El informe deberá ser preparado siguiendo el formato establecido en clase (cf. guía para la preparación del informe de laboratorio). En la sección 'Resultados' deberá incluirse todos los requerimientos expuestos anteriormente en la Sección 4. El informe se entregará en soporte papel y **no podrá superar las 10 páginas de extensión.** La **fecha límite de entrega** será el día **9 de Abril de 2014 (grupos 1.∗**) y el día **10 de Abril de 2014 (grupos 3.∗)**. Las hojas **deberán imprimirse por ambas caras** y no se aceptarán informes impresos a simple faz (ahorremos papel… y peso innecesario a los profesores!).

[1] Nótese que en este caso el fabricante los expresa por separado pero también combinados en la exactitud.
[2] Esta incerteza afecta a la curva de calibración obtenida.

**Wind tunnel calibration procedure**

# Práctica de laboratorio Nº 3: Calibración de la presión dinámica y estática en la cámara de ensayos del túnel de viento del Laboratorio de Aeronáutica

## 1. Objetivos de la práctica

Mediante la realización de esta práctica se pretende que el alumno:

- Realice la calibración de la presión dinámica en una sección de la cámara de ensayos del túnel de viento del Laboratorio de Aeronáutica.
- Lleve a cabo un procedimiento típico de estimación de incertezas a fin de evaluar la influencia en el resultado final de los distintos componentes de la cadena de medición.
- Se familiarice con el túnel de viento en cuestión, con el montaje experimental, con el conexionado de los instrumentos y sus especificaciones técnicas más relevantes.

## 2. Descripción del montaje experimental

La calibración de la presión dinámica y estática en la cámara de ensayos, en función de la caída de presión estática en la tobera, es un dato importante para la operación del túnel de viento que evita la necesidad de introducir en la sección de trabajo instrumentos adicionales para medir la velocidad durante los ensayos (siempre y cuando el factor de bloqueo se mantenga dentro de límites aceptables). La caída de presión estática en la tobera es fácil de medir y generalmente se realiza mediante un montaje fijo en el túnel de viento.

Para la realización de este ensayo la tobera del túnel de viento se encuentra instrumentada con dos anillos de tomas estáticas, uno a la entrada y otro a la salida. En cada anillo, las tomas estáticas interconectadas miden la presión estática promedio en la sección. Durante la calibración, la presión dinámica del flujo en la cámara de ensayos ($q^C$) se mide mediante una sonda pitot-estática[1] tipo Prandtl posicionada sobre el eje longitudinal de la cámara[2]. La diferencia de presión estática en la tobera ($\Delta p_s^T$) se mide mediante un manómetro de columna

[1] Ver esquema al final del documento.
[2] Debido a la presencia de perturbaciones espaciales en el flujo y a posibles imperfecciones de construcción/montaje del túnel de viento, la presión dinámica no será constante en la cámara de ensayos. Si bien el procedimiento correcto es realizar una exploración en distintas secciones de la cámara para determinar la variación de las propiedades del flujo, en esta práctica (y a modo de ejemplo) tomaremos un sólo punto de medición como representativo de la presión dinámica en la sección.

de agua de tubo en 'U' mientras que la diferencia de presión en la sonda pitot-estática se obtiene mediante un transductor de presión diferencial SETRA 239 cuyo rango es de 0-0.5 psid (calibrado en la Práctica de Laboratorio Nº 2). La lectura de tensión de salida del transductor se realiza mediante un multímetro ISO-TECH IDM 203. Completan el instrumental un sensor de temperatura TESTO 110 y un barómetro digital TESTO 511.

## 3. Procedimiento experimental a seguir

### Parte 1: medición de $q^C = f(\Delta p_s^T)$

1. Previo a la realización del ensayo deberá verificarse que todos los instrumentos se encuentren correctamente conectados y en condiciones de ser utilizados.
2. Utilizando el barómetro se medirá la presión atmosférica ($P_{atm}$) y se asumirá que la misma no varía durante el ensayo. Además, se apuntará el ángulo de inclinación de la columna de líquido del manómetro de tubo en 'U' y se medirá la temperatura ambiente, necesaria para corregir la densidad del agua del manómetro.
3. En condiciones *wind-off* se tomará la lectura del primer *cero* del transductor de presión conectado a la sonda pitot-estática (se recomienda trabajar con una resolución en el display del multímetro de 1/100 vdc).
4. Luego se pondrá en marcha el ventilador del túnel de viento y se medirá la presión dinámica en la sección de ensayos ($q^C$) para diferencias de presión estática en la tobera $\Delta p_s^T / 2$ = 20, 40, 80 y 160 mm$H_2O$. Para la máxima velocidad de ensayo se deberá medir la temperatura. Para ello se posicionará la sonda de temperatura a la entrada de la toma de aire del túnel y supondremos que dicha temperatura es igual a la temperatura de la cámara de ensayos.
5. A continuación, en condiciones *wind-off*, se tomará el segundo *cero* del transductor.

### Parte 2: medición de $\Delta p_s^C = f(\Delta p_s^T)$

1. Una vez conocida la variación de presión dinámica en la cámara, resulta necesario conocer la presión estática para poder determinar la densidad del fluido y mediante esta última la velocidad. Para realizar esta medición, se desconectará la toma total de la sonda pitot-estática de manera tal que la **toma +** del transductor quede abierta a la atmósfera. Con ello podremos medir $\Delta p_s^C = p_{atm} - p_s^C$ y, de manera simultánea, la

caída de presión estática en la tobera $\Delta p_s^T$ mediante el manómetro en 'U'. Para esta configuración, se repetirán los puntos (3)-(5)[3].

2. A continuación se tomará nota de las especificaciones técnicas del transductor y de todos aquellos datos del túnel de viento, de los instrumentos y del montaje experimental que son necesarios para el post-proceso de las mediciones.

Nota 1: es una práctica muy recomendable que de manera previa a la realización del ensayo se confeccione una planilla que permita tomar nota de todos aquellos datos necesarios para registrar adecuadamente el experimento. También debe incluirse en dicha planilla toda aquella información adicional que se considere necesaria para el posterior procesamiento de las mediciones. La confección de la planilla quedará a criterio de cada grupo.

## 4. Post-proceso de los datos

### 4.1 Obtención de las curvas de calibración

Como resultado del ensayo se obtendrán dos conjuntos de mediciones: $q^C = f_1(\Delta p_s^T)$ y $\Delta p_s^C = f_2(\Delta p_s^T)$. Los valores correspondientes a $q^C$ y a $\Delta p_s^C$ serán leídos en voltios de salida del transductor, por lo cual será necesario multiplicarlos por la sensibilidad del mismo obtenida de calibración[4] (utilizaremos tensiones netas). La densidad del agua del manómetro se calculará en base a la temperatura ambiente medida durante el ensayo. Una vez convertidos los datos a unidades coherentes (utilizaremos Pa), se deberán ajustar ambos conjuntos de mediciones mediante curvas de regresión lineal para obtener las correspondientes sensibilidades $\partial q^C / \partial \Delta p_s^T$ y $\partial \Delta p_s^C / \partial \Delta p_s^T$. Conocidas las curvas de calibración se obtendrá a partir de las mismas la velocidad en la cámara de ensayos en función de la caída de presión estática en la tobera $V^C = f(\Delta p_s^T)$ de la siguiente manera

$$V^C = \sqrt{\frac{2\,q^C}{\rho^C}} = \sqrt{\frac{2\,q^C}{p_s^C} RT^C} = \sqrt{\frac{2\,f_1(\Delta p_s^T)}{P_{atm} - f_2(\Delta p_s^T)} RT^C} \quad (1)$$

donde $P_{atm}$ y $T^C$ son las medidas durante el ensayo.

[3] Cabe remarcar que la presión estática manométrica en la cámara será aproximadamente igual a la presión dinámica ya que las pérdidas de presión total son pequeñas.
[4] Trabajo de Laboratorio Nº 2 (asumimos que la linealidad del transductor se mantiene en todo el rango de medidas tomadas en el ensayo).

Se deberán presentar las gráficas $q^C = f(\Delta p_s^T)$ y $\Delta p_s^C = f(\Delta p_s^T)$ con su correspondiente regresión lineal, incluyendo la ecuación de ajuste y el coeficiente de determinación. Además se presentará una gráfica con $V^C = f(\Delta p_s^T)$ (Ec. (1)) adoptando unidades de presión en Pa y de velocidad en m/s. Sobre esta curva aproximada se graficarán los puntos medidos durante el ensayo. Deberá indicarse en la gráfica la presión atmosférica y temperatura de ensayo para las cuales se obtiene la curva de velocidad.

### 4.2 Análisis de incertezas

1. En primer lugar se calcularán las incertezas totales (RSS) $\delta q^C$ y $\Delta p_s^C$ correspondientes a las curvas $q^C = f(\Delta p_s^T)$ y $\Delta p_s^C = f(\Delta p_s^T)$ respectivamente (en Pa). Para ello se tendrán en cuenta el error de calibración del transductor (obtenido en la Práctica de Laboratorio Nº 2), el error en la lectura del multímetro (accuracy), el error en la lectura del manómetro de tubo en 'U' (únicamente resolución) y finalmente los errores de ajuste de las respectivas funciones de calibración obtenidas ($\|\cdot\|_\infty$ de los residuos).

2. Para cada velocidad de ensayo se determinarán las incertezas totales $\delta V_i$ teniendo en cuenta las incertezas en las funciones de calibración $\delta q^C$ y $\Delta p_s^C$ anteriormente comentadas así como también la incerteza en la temperatura y presión atmosférica medidas durante el ensayo (cf. example 3, M2_2-Uncertainties, pag. 26). Para la incerteza en la temperatura y la presión atmosférica se tendrán en cuenta la exactitud y resolución de los instrumentos[5].

Las incertezas calculadas para cada una de las velocidades medidas deberán añadirse a la gráfica $V^C = f(\Delta p_s^T)$ mediante barras de error simétricas[6]. Adicionalmente, teniendo en cuenta las sensibilidades calculadas, se pide extraer algunas conclusiones en cuanto a las distintas influencias en el resultado final debidas a los elementos de la cadena de medición.

## 5. Preparación del informe

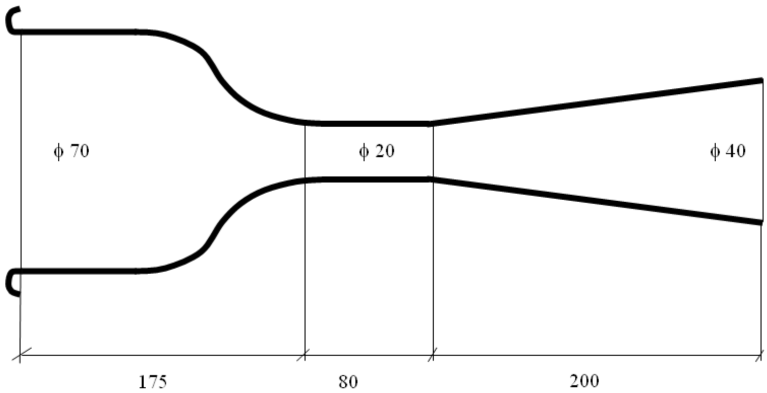
El informe deberá ser preparado siguiendo el formato establecido en clase (cf. guía para la preparación del informe de laboratorio). En la sección 'Resultados' deberá incluirse todos los requerimientos expuestos anteriormente en la Sección 4. El informe se entregará en soporte

[5] Ver hojas de datos de los respectivos instrumentos.
[6] Ver: http://www.ncsu.edu/labwrite/res/gt/gt-stat-home.html (se recomienda dedicar un poco de tiempo a navegar por este sitio web ya que contiene mucha información de utilidad referida al trabajo de laboratorio y a la preparación de informes).

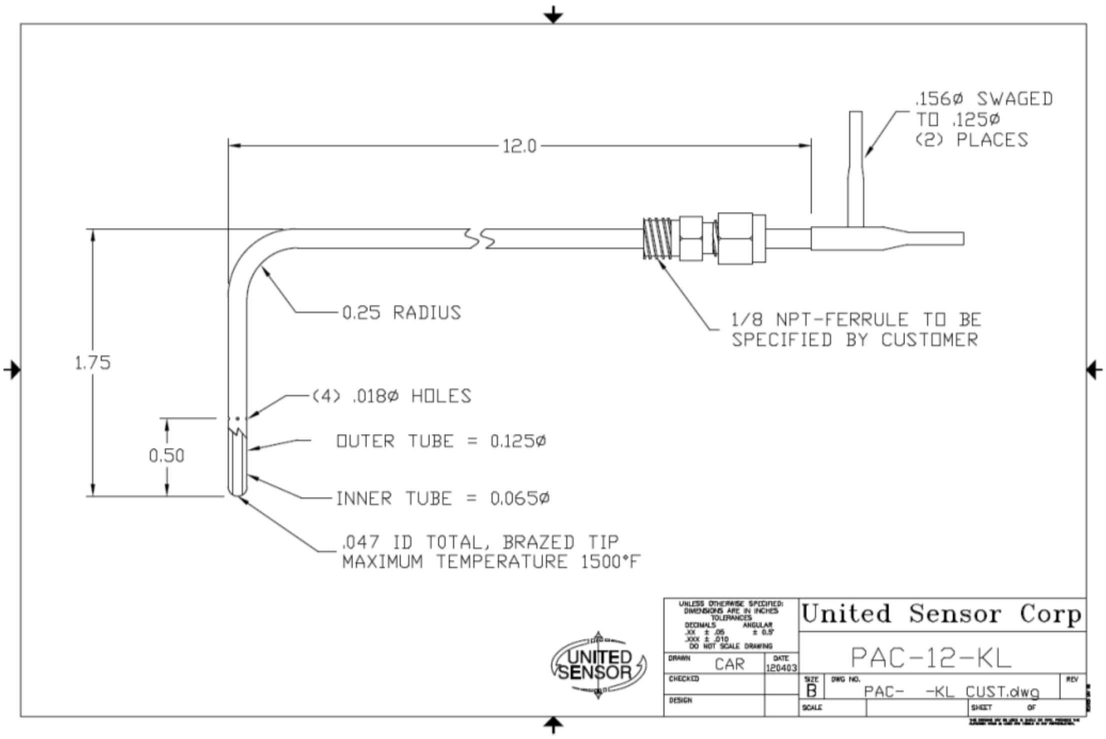
papel y **no podrá superar las 11 páginas de extensión.** La **fecha límite de entrega** será el día **7 de Mayo 2014 (grupos 1.***) y día **8 de Mayo de 2014** (**grupos 3.***). Las hojas **deberán imprimirse por ambas caras** y no se aceptarán informes impresos a simple faz (ahorremos papel… y peso innecesario a los profesores!).

---

**Esquema del túnel de viento del Laboratorio de Aeronáutica (con cámara de sección circular, unidades en cm)**




**Sonda pitot-estática tipo Prandtl a utilizar en el ensayo**



# D Further information

## D.1 Arduino UNO [1]

First of all the Arduino UNO board, which can be seen in Figure D.1, was studied carefully.

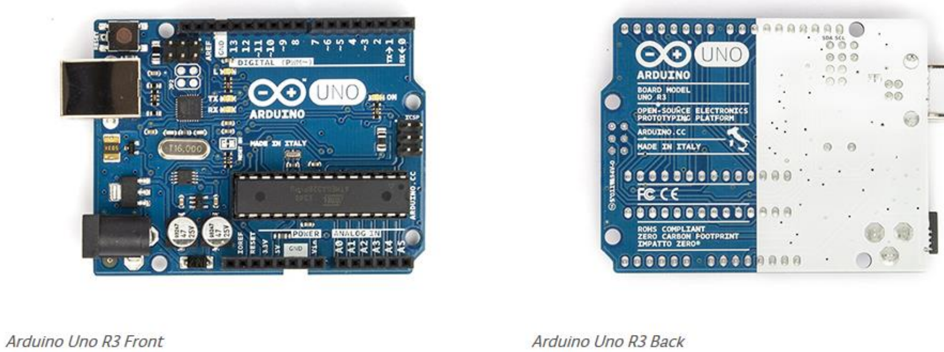



Figure D.1. Arduino UNO R3 version board Front and Back views [1]

The Arduino Uno is a microcontroller board based on the ATmega328 chipset. It has 14 digital input/output pins of which 6 can be used as PWM outputs; this feature would allow the Arduino to send the control signal to the fan of the wind tunnel, when properly conditioned thanks to a low-pass filter. It also has 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. The analog inputs have analog to digital conversion, so that when the signal from the sensors (analog) is received it translates it to understandable language for the microcontroller. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 programmed as a USB-to-serial converter

The Revision 3, which was the one studied in the decision making process, has the following features:

- Pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V.

- Stronger RESET circuit.

- Atmega 16U2 replaces the 8U2 from previous versions, which means that flash memory has been doubled, not specially for sketches, but it may be used for both the USB-to-Serial conversion and the USB Host connection to the Android phone.

”Uno” means one in Italian, where it comes from, and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 are the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; it is an evolution from the previous Arduino Duemilanove board, which is very similar but older, and with so less up-to-date.

Just extracted from Arduino's official website [1] its main characteristics are the ones that follow in Table D.1.

| **Arduino UNO R3** | |
|---|---|
| **Microcontroller** | ATmega328 |
| **Operating Voltage** | 5V |
| **Input Voltage (recommended)** | 7-12V |
| **Input Voltage (limits)** | 6-20V |
| **Digital I/O Pins** | 14 (of which 6 provide PWM output) |
| **Analog Input Pins** | 6 |
| **DC Current per I/O Pin** | 40 mA |
| **DC Current for 3.3V Pin** | 50 mA |
| **Flash Memory** | 32 KB (ATmega328) of which 0.5 KB used by boot loader |
| **SRAM** | 2 KB (ATmega328) |
| **EEPROM** | 1 KB (ATmega328) |
| **Clock Speed** | 16 MHz |
| **Length** | 68.6 mm |
| **Width** | 53.4 mm |
| **Weight** | 25 g |

Table D.1. Arduino main characteristics

### ATmega 328

The high-performance Atmel 8-bit AVR RISC-based microcontroller combines 32KB ISP flash memory with read-while-write capabilities, 1KB EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal

oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts.
By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed. [2]

### Power

The Arduino Uno board can be powered by two different ways, via the USB connection or with an external power supply. The power source is selected automatically by the board once it is connected.
The external power supply can come either from an AC-to-DC adapter or a battery, so, in fact there are three possible ways to power it. The adapter can be connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the GND (negative pole) and Vin (positive pole) pin headers of the POWER connector.
The board can operate on an external supply of 6 to 20 volts. However, if the power supply is less than 7V, the 5V pin may supply less than five volts and the board may be unstable. On the other hand, if using more than 12V, the voltage regulator may overheat and damage the board. So, the recommended range is 7 to 12 volts.
The power pins are as follows:
**VIN:** It is the input voltage pin to the Arduino board when it's using an external power source. One can supply voltage through this pin, or, if supplying voltage via the power jack, access to 5V through this pin, as an out source.
**5V**: This pin outputs a regulated 5V from the regulator on the board. Supplying voltage via the 5V pin bypasses the regulator, and can damage the board, so the user has to be careful at the time cable connections are made.
**3V3:** Similarly to the previous one, it can supply 3.3 volt generated by the on-board regulator. Maximum current draw here is 50 mA. As well as explained before, supplying voltage via the 3.3V pin bypasses the regulator, and can damage the board, so the user has to be careful at the time cable connections are made.
**GND.** Ground pins. Arduino UNO has 3 clearly identified GND pins.
**IOREF.** This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

### Memory

In this aspect, the ATmega328 features 32 KB, where 0.5 KB are used for the boot loader, and cannot be used for programming purposes. It also has 2 KB of SRAM and 1 KB of EEPROM. The electronic erasable programmable rom can be read and written with the Arduino EEPROM library.

### Inputs and outputs

First of all, the pins have to be divided, as they are in the board, in analog and digital for their analysis.
Each of the 14 digital pins on the Uno can be used as an input or output, using the functions pinMode(),to select input or output; digitalWrite(), to write 1 or 0 in the output pin assigned; and digitalRead(), to read digital values received in that pin. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some of these 14 pins have specialized functions:
**Serial: 0 (RX) and 1 (TX)**. Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
**External Interrupts: 2 and 3**. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
**PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function. These pins are really useful because, even they are digital outputs, the pulse width modulation signal can be set up to get a constant DC signal with the proper circuit. This will help to control the Altivar 31, and with so the fan velocity.
**SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library, but they are not going to be used for this project
**LED: 13**. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
The Uno has also 6 analog inputs, (no analog output) labelled A0 through A5, each of which provide 10 bits of resolution, which mean from 0 to 1023 values. By default they measure from ground to 5 volts, though is it possible to change (reduce) the upper end of their range using the AREF pin and the analogReference() function, always keeping in mind that the chipset working voltage is 5V.

There are a couple of other pins on the board:
**AREF**. Reference voltage for the analog inputs.
**Reset**. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

### Reset

The Arduino Uno board apart from the physical button to reset the software running on it, the board is designed in a way that allows it to be reset by software running on a connected computer so that there is no need to press the button before an upload from the computer.
One of the hardware flow control lines (DTR) of the ATmega is connected to the reset line via a 100 nanofarad capacitor. When this line is asserted (taken

low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout.

### USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects the user's computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

### Programing (software)

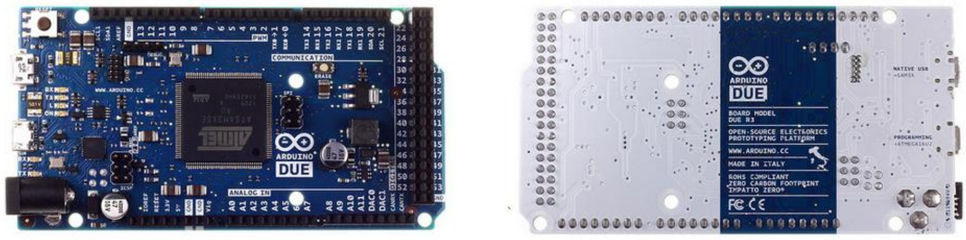
All Arduino boards share the same GUI to write code. This means that once Arduino is programmed the user should select the exact model he is using from the Tools ¿ Board menu (according to the microcontroller), and then the communication port (Usually for windows COMX, where X I can be 1, 2, 3 …)
As stated above, the chipset on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer.
The bootloader and chipset firmware source code are available and can be both reprogrammed and modified.
Something that one has to take into account when programming is that the code has two differentiated parts, the "setup" part which is mainly used to define constants, pins, etc. And the "loop" part, where there are the main statements, and whose code will be continuously repeating until the power supply is cut out.

### Price

The Arduino UNO board costs approximately 22€ [3] if purchased from the official distributors although it is possible to found copies on the network for half or even a third of this price. The price is not going to be really an issue on this project, since the ETSEIAT have already some boards in property of both Uno and DUE.

## D.2 Arduino DUE [4]

Afterwards, the Arduino Due board, which can be seen in Figure D.2 , was analysed as deeply as the previous one.



Arduino Due Front    Arduino Due Back

Figure D.2. Arduino DUE board Front and Back views [4]

The Arduino Due is a microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 CPU. It is the first Arduino board based on a 32-bit ARM core microcontroller. It has 54 digital input/output pins of which 12 can be used as PWM outputs, 12 analog inputs, 4 hardware serial ports, an 84 MHz clock, an USB OTG capable connection, 2 DAC (digital to analog), 2 TWI, a power jack, an SPI header, a JTAG header, a reset button and an erase button. All this equipment placed in a PCB with just 4 and 2.1 inches of length and width respectively.
The Arduino Due is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove this means that some of the digital pins, analog inputs 0 to 5, the power header, and "ICSP" (SPI) header are all in equivalent locations. This provides the Due with high versatility. However, the user has to be very careful when combining different boards since, unlike other Arduino boards, the Arduino Due board runs at 3.3V, meaning that the maximum voltage that the I/O pins can tolerate is 3.3V. If a higher voltage were supplied to an I/O pin, like 5V, the board could get damaged.
As well as the Uno, this board contains everything needed to support the microcontroller; simply connect it to a computer with a micro-USB cable or power it with an AC-to-DC adapter or battery to get started. The Due is compatible with all Arduino shields that work at 3.3V and are compliant with the 1.0 Arduino pinout. The IOREF pin allows an attached shield with the proper configuration to adapt to the voltage provided by the board. This enables shield compatibility with a 3.3V board like the Due and AVR-based boards which operate at 5V.

Just extracted from Arduino's official website its main characteristics are the ones that follow in Table D.**2**.

| Arduino DUE | |
|---|---|
| **Microcontroller** | AT91SAM3X8E |
| **Operating Voltage** | 3.3V |
| **Input Voltage (recommended)** | 7-12V |
| **Input Voltage (limits)** | 6-16V |
| **Digital I/O Pins** | 54 (of which 12 provide PWM output) |
| **Analog Input Pins** | 12 |
| **Analog Outputs Pins** | 2 (DAC) |
| **Total DC Output Current on all I/O lines** | 130 mA |
| **DC Current for 3.3V Pin** | 800 mA |
| **DC Current for 5V Pin** | 800 mA |
| **Flash Memory** | 512 KB all available for the user applications |
| **SRAM** | 96 KB (two banks: 64KB and 32KB) |
| **Clock Speed** | 84 MHz |
| **Length** | 101.52 mm |
| **Width** | 53.3 mm |
| **Weight** | 36 g |

Table D.2.Arduino Due summary of characteristics

**AT91SAM3X8E- 32-bit ARM core**

The Due board is provided with a 32-bit ARM core that can outperform typical 8-bit microcontroller boards (like the Uno). The most significant differences are:

- A 32-bit core, that allows operations on 4 bytes wide data within a single CPU clock.
- CPU Clock at 84 MHz.
- 96 Kbytes of SRAM.
- 512 Kbytes of Flash memory for code.
- A DMA controller, which can relieve the CPU from doing memory intensive tasks.

Based on the ARMff Cortexff-M3 processor, the Atmelff — SMART SAM3X8E runs at 84MHz and features 512KB of flash memory in 2 x 256KB banks and 100KB of SRAM in 64KB +32KB banks, with an additional 4KB as NFC (NAND Flash controller) SRAM. Its highly-integrated peripheral set for connectivity and communication includes Ethernet, dual CAN, High Speed USB MiniHost and device with on-chip PHY, high-speed SD/SDIO/MMC, and multiple USARTs, SPIs, TWIs (I2C), and one I2S. The SAM3X8E also features a 12-bit ADC/DAC, temperature sensor, 32-bit timers, PWM timer and RTC. The 16-bit external bus interface supports SRAM, PSRAM, NOR and NAND Flash with error code correction. The device operates from 1.62V to 3.6V and is available in 144-pin QFP and BGA packages. [5]

### Power

By the same token, the Due PCB can be powered via the USB connector or with an external power supply, which makes no difference with the Uno.
The external power (AC/DC adapter and with pin headers) and the USB supply work in the same way that it has been explained in 0-Power, and although the board can operate on an external supply of 6 to 20 volts. The recommended range is 7 to 12 volts, in order not to damage the board nor have power problems.

The power pins are as follows, and almost identic to the Uno's:
**VIN:** The input voltage to the Arduino board when it's using an external power source
**5V:** This pin outputs a regulated 5V from the regulator on the board.
**3.3V:** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 800 mA. This regulator also provides the power supply to the SAM3X microcontroller.
**GND:** Ground pins.
**IOREF:** This pin on the Arduino board provides the voltage reference with which the microcontroller operates. It is very useful to adapt this PCB, which operates at 3.3V, to other shields or boards.

### Memory

The SAM3X has two blocks of 256 KB, leading to 512KB, of flash memory for storing code. The boot loader is preburned in factory from Atmel and is stored in a dedicated ROM memory, so it doesn't take up space for the user-made code as it happened with the UNO.
The available SRAM is 96 KB in two contiguous banks of 64 KB and 32 KB.
It is possible to erase the Flash memory of the SAM3X with the on-board erase button. This will remove the currently loaded sketch from the MCU. This is an extra feature that it is not equipped in the Uno, which can obviously useful in some cases but not specially to achieve this project's goal.

### Input and Output

**Digital I/O: pins from 0 to 53**
Each of the 54 digital pins on the Due can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 3.3 volts. Each pin can provide (source) a current of 3 mA or 15 mA, depending on the pin, or receive (sink) a current of 6 mA or 9 mA, depending on the pin. They also have an internal pull-up resistor (disconnected by default) of 100 kOhm. In addition, some pins have specialized functions:
**Serial: 0 (RX) and 1 (TX)**
**Serial 1: 19 (RX) and 18 (TX)**
**Serial 2: 17 (RX) and 16 (TX)**
**Serial 3: 15 (RX) and 14 (TX)**

Used to receive (RX) and transmit (TX) TTL serial data (with 3.3 V level). Pins 0 and 1 are connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.

**PWM: Pins 2 to 13**

Provide 8-bit PWM output with the analogWrite() function. The resolution of the PWM can be changed with the analogWriteResolution() function.

**SPI:** SPI header (ICSP header on other Arduino boards)

These pins support SPI communication using the SPI library. The SPI pins are broken out on the central 6-pin header, which is physically compatible with the Uno, Leonardo and Mega2560. The SPI header can be used only to communicate with other SPI devices, not for programming the SAM3X with the In-Circuit-Serial-Programming technique. The SPI of the Due has also advanced features that can be used with the Extended SPI methods for Due.

**CAN:** CANRX and CANTX

These pins support the CAN communication protocol but are not yet supported by Arduino APIs.

**"L" LED: 13**

There is a built-in LED connected to digital pin 13. When the pin is HIGH, the LED is on, when the pin is LOW, it's off. It is also possible to dim the LED because the digital pin 13 is also a PWM output.

**TWI 1: 20 (SDA) and 21 (SCL)**

**TWI 2: SDA1 and SCL1.**

Support TWI communication using the Wire library. SDA1 and SCL1 can be controlled using the Wire1 class provided by the Wire library. While SDA and SCL have internal pull-up resistors, SDA1 and SCL1 have not. Adding two pull-up resistors on SDA1 and SCL1 lines is required for using Wire1.

**Analog Inputs: pins from A0 to A11**

The Due has 12 analog inputs, each of which can provide 12 bits of resolution (i.e. 4096 different values). By default, the resolution of the readings is set at 10 bits, for compatibility with other Arduino boards. It is possible to change the resolution of the ADC with analogReadResolution(). The Due's analog inputs pins measure from ground to a maximum value of 3.3V. Applying more than 3.3V on the Due's pins will damage the SAM3X chip. The analogReference() function is ignored on the Due.

The AREF pin is connected to the SAM3X analog reference pin through a resistor bridge. To use the AREF pin, resistor BR1 must be unsoldered from the PCB.

**DAC1 and DAC2**

These pins provides true analog outputs with 12-bits resolution (4096 levels) with the analogWrite() function. These pins can be used to create an audio output using the Audio library.

Other pins on the board:
**AREF**
Reference voltage for the analog inputs. Used with analogReference().
**Reset**
Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

### USB Overcurrent Protection

The Arduino Due has the same system explained above for the Uno in 0 as an USB Overcurrent Protection method.

### Programming

Equally to the Arduino Uno board, the Due can be programmed with the Arduino tool, and the procedure would be the same, except when the user has to select which board is he using, which is obvious.
Although it is not going to be a problem when programming, uploading sketches to the SAM3X is different than the AVR microcontrollers found in other Arduino boards because the flash memory needs to be erased before being re-programmed. So, the upload to the chip is managed by ROM.
In contrast with the Uno board, it has two USB ports (see Figure.3) and either of them can be used for programming the board, though it is recommended to use the Programming port due to the way the erasing of the chip is handled. Here the reset function is done automatically at the time of uploading the code with a “Hard Reset” or a “Soft Reset” depending on the USB port selected.

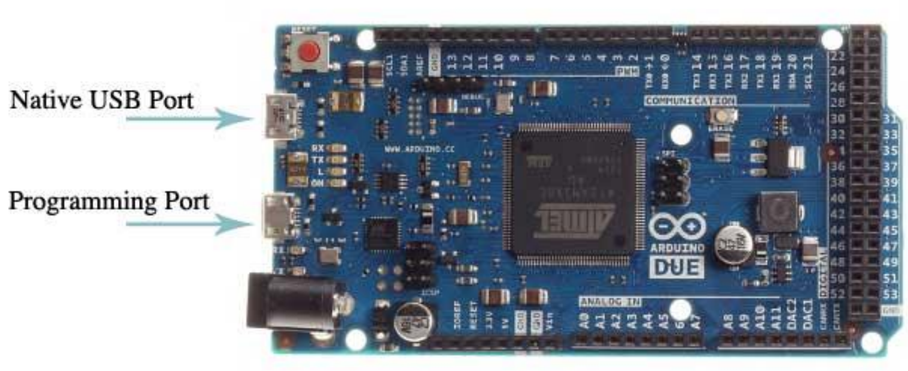



Figure D.3. Arduino DUE, USB Ports

In the same way as the UNO, the firmware comes preburned but it can also be modified or rewritten. The ATmega16U2 firmware source code is available in the Arduino repository. You can use the ISP header with an external programmer and overwrite the DFU boot loader (out of this project’s scope).

**Price**

The Arduino DUE board costs approximately 43€ [6] if purchased from the official distributors although it is possible to found copies on the network for half or even a third of this price. However, it is only a reference because; the university already has some boards bought.
When writing this paper, one has to mention that buying official boards not only contributes to the original company to continue updating their products or develop new ones but also helps all the support for users they provide

## D.3 Arduino Shields

The shields that were taken into consideration at the time off chossing the connection method between Arduino and the computer where those that follow:

- Wireless SD shield: The Wireless SD shield allows an Arduino board to communicate wirelessly using a wireless module. It is based on the Xbee or ZigBee modules, but can use any module with the same footprint. The module can communicate up to 100 feet indoors or 300 feet outdoors (with line-of-sight). It can be used as a serial/usb replacement. The shields breaks out each of the Xbee's pins to a through-hole solder pad. Included on board there is a SD card slot. The Xbee 802.15.4 modules support point-to-point or multipoint communication without routing, so they are ready out of the box to establish peer communication between two radios without any configuration. The ZigBee protocol allows the user to set up a radio link between the modules that are distant from each other by sending messages through multiple routers from endpoint to endpoint. They can also make networks in which the endpoints save power by sleeping, but don't lose messages when asleep. [7] [8]

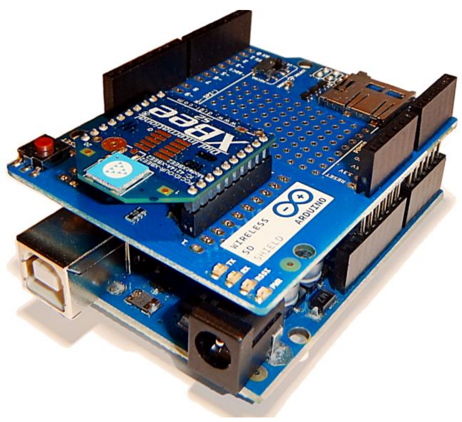


Figure D.4 Wireless shield with XBee module [9]

- Wifi Shield: The Arduino WiFi Shield connects Arduino to the internet wirelessly using the 802.11 wireless specification (WiFi). It is based on the HDG204 Wireless LAN 802.11b/g System in-Package. The WiFI shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top. The WiFi Shield can connect to wireless networks which operate according to the 802.11b and 802.11g specifications. There is also an onboard micro-SD card slot, which can be used to store files for serving over the network. [10]

- The Arduino Ethernet Shield also allows an Arduino board to connect to the internet but in this case through wire communication. It is based on the Wiznet W5100 ethernet chip. Like the previous one, the ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the, this keeps the pin layout intact and allows another shield to be stacked on top. The Ethernet Shield has a standard RJ-45 connection, with an integrated line transformer and Power over Ethernet enabled. [11]

There are, obviously, a lot more shields to stack to Arduino [12] [13], however these ones were object of study because they offer different possibilities to connect the Arduino to any computer.
Despite these possibilities, for the sake of simplicity, none of the shields was chosen. Two of them allowed to connect the Arduino directly to the internet , allowing to save data automatically and acces it from the cloud, for example. But there was no need to install this system as a computer with internet connecton will be near the Arduino, and it is necessary to power the board. The other meant having to use two devices one for sending the radio signals and other working as a receiver in the workstation. This was discarded for the same reason, the computer will be near the board and a USB cable is cheaper and easier to configure.

Moreover a link to the programming code and the excels is available:
https://www.dropbox.com/sh/9p0b55yaxhe9slk/AAAwSAv_pl3__l15DMtSRt9wa?dl=0

## Bibliography


1] "ArduinoUNO-Arduino.cc," March 2015. [Online]. Available: http://www.arduino.cc/en/main/arduinoBoardUno.

2] "ATmega328-Atmel.com," March 2015. [Online]. Available: http://www.atmel.com/devices/atmega328.aspx.

3] "Arduino UNO-Ondaradio," March 2015. [Online]. Available: http://www.ondaradio.es/producto/a000099-rohs-arduino-uno-a000099-rev3-whit-long-pins-atmega328-45200.aspx.

4] "Arduino Due Board-Arduino.cc," March 2015. [Online]. Available: http://www.arduino.cc/en/Main/arduinoBoardDue.

5] "Atmel," May 2015. [Online]. Available: http://www.atmel.com/devices/sam3x8e.aspx.

6] "Arduino DUE-Ondaradio," 2015 February. [Online]. Available: http://www.ondaradio.es/producto/a000056-rohs-arduino-due-without-headers-32bit-atmel-sam3x8e-arm-44587.aspx.

7] "Wireless Shield - Arduino.cc," [Online]. Available: http://www.arduino.cc/en/Main/ArduinoWirelessShield. [Accessed March 2015].

8] "XBee Shield- Arduino," [Online]. Available: http://www.arduino.cc/en/Main/ArduinoXbeeShield. [Accessed March 2015].

9] "Wiriless Shield-Arduino.cc," [Online]. Available: http://www.arduino.cc/en/Guide/ArduinoWirelessShieldS2. [Accessed March 2015].

10] "Wifi Shield-Arduino.cc," [Online]. Available: http://www.arduino.cc/en/Main/ArduinoWiFiShield. [Accessed March 2015].

11] "Ethernet Shield-Arduino.cc," [Online]. Available: http://www.arduino.cc/en/Main/ArduinoEthernetShield. [Accessed March 2015].

12] "Similar Shields- Arduino playrground," [Online]. Available: http://playground.arduino.cc/Main/SimilarBoards. [Accessed March 2015].

13] "Arduino Shields - Arduino.cc," [Online]. Available: http://www.arduino.cc/en/Main/ArduinoShields. [Accessed March 2015].