# COMMUNICATION PROTOCOL
FOR
SL2FF EVO, SL4FF, SL4 EVO, SL4 DIA, SL4 WOOD, SL8TFF, SL4Q, SL4QN, SL4U, SL6Q, WP-CNC1, WP-CNC2, WP-CNC4, WPCNC-TURBO, HDH4000, HDMV, HDHS, RF300GS, STORAGE SYSTEM
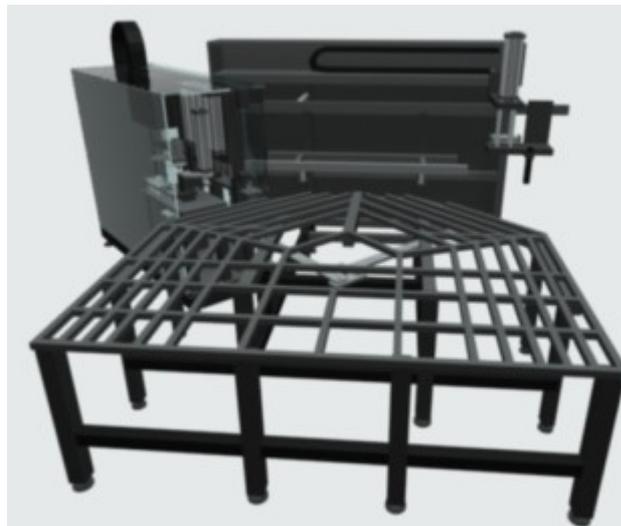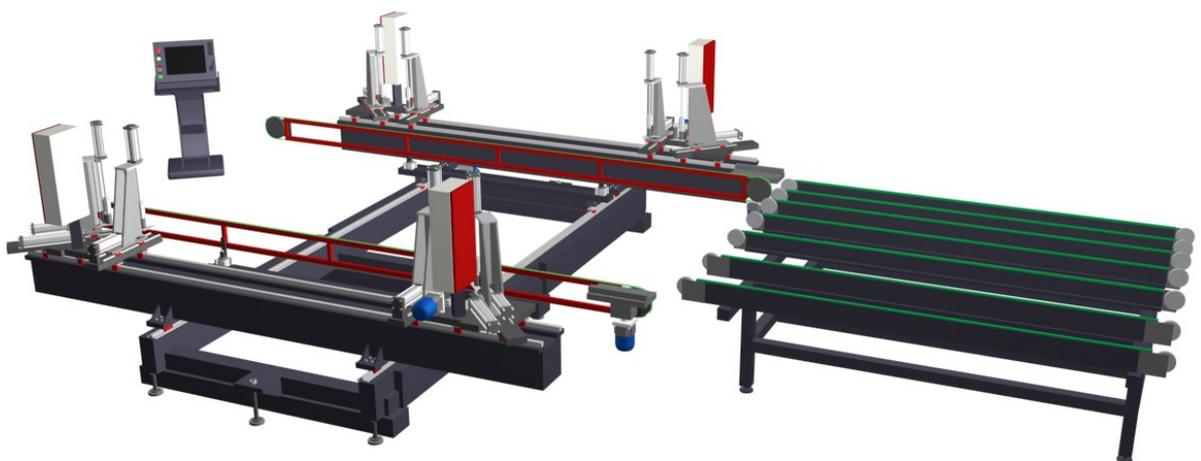
TECHNICAL SPECIFICATION

INDEX

1. WEL Protocol

1.1. File specification

This protocol is for welder SL2
FF EVO, SL4FF, SL4FF-EVO, SL4 DIA, SL4 WOOD, SL8TFF, SL4Q, SL4QN, SL4Q, SL6Q and corner cleaner WP-CNC1, WP-CNC2, WP-CNC4, WPCNC-Turbo, frame hinge drilling HDH4000, HDMV, sash hinge drilling HDHS, trimming RF300GS, shuttle/store SORAGE SYSTEM

All information for working lists is in this type file:

- The file name has extension "**.WEL**".
- The length of the file name is configurable, but after deciding what should be long, should stay that.
- The file name must appear in the first digit of the bar code to be read to identify the "square".
- The file format is ASCII text type, where the different machine programming blocks are identifiable.
- Each programming block is constituted by:
  o From 3 to 6 window acknowledgment rows (Usual are 4, corresponding to the four window sides).
    - These rows will be always and only found by the keyword "**CodeId**"
    - Each field, within a single row, will be separate from next, with the semicolon character "**;**" (;=ASCII 59).
    - These rows will finish by the characters semicolon "**;**" + Carriage-return "**CR**" (**;**=ASCII 59+ **CR**=ASCII 13).
  o One machine configuration row.
    - This row will be the first row that **don't** start with CodeId.
    - Each field, within the row, will be separate from next, with the semicolon character "**;**" (;= ASCII 59).
    - These rows will finish by the characters semicolon "**;**" + Carriage-return "**CR**" (**;**=ASCII 59+ **CR**=ASCII 13).
  o Two or more rows for window type and match between frame and sashes (optional).
    - These rows will be always and only found by the keyword "**FrameInfo**" and "**SashInfo**"
    - Each field, within a single row, will be separate from next, with the semicolon character "**;**" (;=ASCII 59).
    - These rows will finish by the characters semicolon "**;**" + Carriage-return "**CR**" (**;**=ASCII 59+ **CR**=ASCII 13).
  o Seven rows for window configuration:
    - These rows will be always and only found by the keywords:
      - "**Gasket**": gasket presence.
      - "**ColorInside**": indicates if the window inside is white or colored.
      - "**ColorOutside**": indicates if the window outside is white or colored.
      - "**Height**": window height or left side.
      - "**Height1**": window height or right side.
      - "**Width**": window width.

- • "**NoAngle**": angles number that the corner cleaner machine must clean and window type.
  - ▪ Each field, within the single row, will be separate from next, with the semicolon character "**;**" (**;**= ASCII 59).
  - ▪ These rows will finish by the characters semicolon "**;**" + Carriage-return "**CR**" (**;**=ASCII 59+ **CR**=ASCII 13).
- o One or more working configuration rows (optional).
  - ▪ These rows will be always and only found by the keyword "**Fab**".
  - ▪ Each field, within the single row, will be separate from next, with the semicolon character "**;**" (**;**= ASCII 59).
  - ▪ These rows will finish by the characters semicolon "**;**" + Carriage-return "**CR**" (**;**=ASCII 59+ **CR**=ASCII 13).
- o One or two mullion configuration row (optional).
  - ▪ This row will be always and only found by the keyword "**Mullion**".
  - ▪ Each field, within the single row, will be separate from next, with the semicolon character "**;**" (**;**= ASCII 59).
  - ▪ This row will finish by the characters semicolon "**;**" + Carriage-return "**CR**" (**;**=ASCII 59+ **CR**=ASCII 13)
- o One or more hinge hole configuration rows (optional only for HDH400).
  - ▪ These rows will be always and only found by the keyword "**Hinge**".
  - ▪ Each field, within the single row, will be separate from next, with the semicolon character "**;**" (**;**= ASCII 59).
  - ▪ These rows will finish by the characters semicolon "**;**" + Carriage-return "**CR**" (**;**=ASCII 59+ **CR**=ASCII 13).
- o Label print row (optional).
  - ▪ This row will be always and only found by the keyword "**LabId**".
  - ▪ This row will finish by the characters semicolon "**;**" + Carriage-return "**CR**" (**;**=ASCII 59+ **CR**=ASCII 13).
- o End block row.
  - ▪ This row will be always and only found by the keyword "**End**".
  - ▪ This row will finish by the characters semicolon "**;**" + Carriage-return "**CR**" (**;**=ASCII 59+ **CR**=ASCII 13).

A typical block is:

```
CodeId;06030045401;1;PG612;5615;0;;;;;
CodeId;06030046301;2;PG612;12800;100;;;;;
CodeId;06030011101;3;PG612;5615;;;;;;
CodeId;06030012201;4;PG612;12800;100;;;;
CodeId;06030012202;5;PD722;11200;;;;;;
PG612;06031000001;06032000001;13;1;603/2006;Graf Gynergy;Window 01;FRAME;Operatore01;Window;
SashInfo;0010002514;0010002515;0010002516;0010002517;11000;6000;
SashInfo;0010002518;0010002519;0010002520;0010002521;11000;6000;
Gasket;G;
ColorInside;WH;
ColorOutside;WH;
Height;12800;
Height1;12800;
Width;5615;
NoAngle;4;1;
Fab;CERN_ANT;1000;4;
Fab;CERN_ANT;11800;4;
Fab;CERN_ANT;1000;2;
Fab;CERN_ANT;11800;2;
```

```
Mullion;5000;1;
Hinge;CERN_TEL;2;3;
LabId;001;
End;
```

1.2. Blocks describe

We can indicate the following strings with a generic block:

1) **Window acknowledgment row**:
`CodeId;BarCode;ProfPos;ProfileCode;Length;Trim;Fin;CaliperLength;SideDescription;←`
2) **Machine configuration row**:
`ProfProg;SquaredId;WinID;PieceForWin;Box;Order;Customer;FreeComment;ProfileType;OpN;WinType;←`
3) **Window type configuration rows:**
`FrameInfo;FirstBarcode;SecondBarcode;ThirdBarcode;FourthBarcode;Height;Width;`
`SashInfo;FirstBarcode;SecondBarcode;ThirdBarcode;FourthBarcode;Height;Width;`
4) **Window configuration rows**:
```
                                    Gasket;TypeGasket;←
                                    ColorInside;TypeColorInside;←
                                    ColorOutside;TypeColorOutside;←
                                    Height;ValueHeight;←
                                    Height1;ValueHeight;←
                                    Width;ValueWidth;←
                                    NoAngle;QtyAngle;MWType;←
```
5) **Work configuration row:**   `Fab;WorkName;Posit;ProfPos;←`
6) **Mullion configuration row:** `Mullion;Posit;ProfPos;←`
7) **Hinge configuration row:**   `Hinge;HingeName;ProfPos;HingeQty;←`
8) **Label configuration row:**   `LabId;LabelNumber;←`
9) **End block row:**   `End;←`

Note: each row is separate by the character semicolon "**;**" + CARRIAGE-RETURN (ASCII59+ASCII13).

1) **Window acknowledgment row (to be repeated for all barcodes that make up the "square")**:

1.1) **Block identification:**
`CodeId;BarCode;ProfPos;ProfileCode;Length;Trim;Fin;CaliperLength;SideDescription;←`
- **Description**: Use the keyword "**CodeId**". This identifies the barcodes that make up a "square".
- **String type:** Alphabetical
  o **Acceptable strings:** "**CodeId**"
  Other values generate errors.

1.2) **Barcode:**
`CodeId;BarCode;ProfPos;ProfileCode;Length;Trim;Fin;CaliperLength;SideDescription;←`
- **Description**: Barcode belonging to the "square" to clean.
- **String type:** Alphanumerical

1.3) **Profile position (**opzional, only for the corner cleaners WP-CNC1, WP-CNC2 and WP-CNC4 with working system)**:**
`CodeId;BarCode;ProfPos;ProfileCode;Length;Trim;Fin;CaliperLength;SideDescription;←`
   o **Description**: Profile position in the "square". Using the convention that the internal window view is bottom on machine (in most cases, but there may be needs depending on

the shape of the profiles that require loading them backwards), the profile positions will have the following numbering:

➢ "**1**" for the bottom profile.
➢ "**2**" for the right profile.
➢ "**3**" for the up profile
➢ "**4**" for the left profile.
➢ "**5**" for the main mullion.
➢ "**X**" for all other, as mullions, sash reverse ledges, etc.
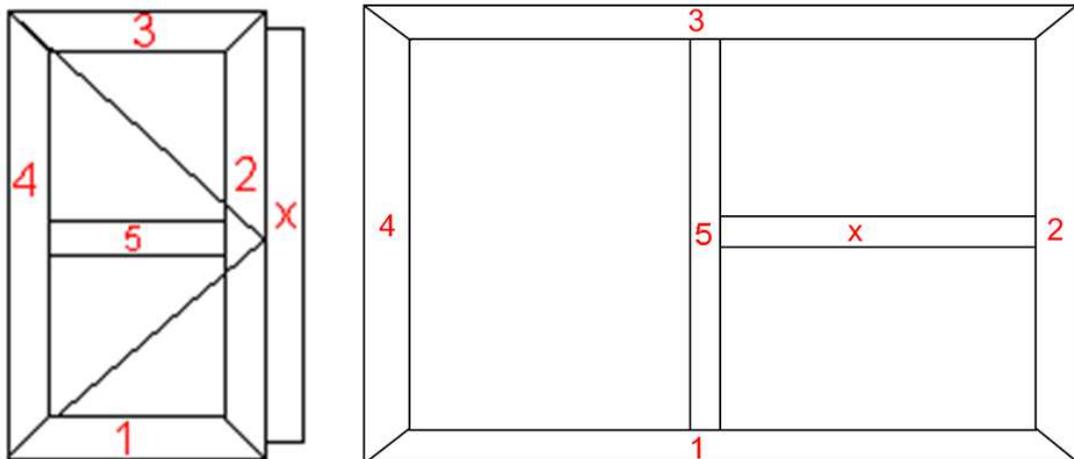
- **String type:** Numerical
  o **Acceptable strings: "1", "2", "3", "4", "5", "X".**
  Other values generate errors.

1.4) **Profile code:**

`CodeId;BarCode;ProfPos;`ProfileCode`;Length;Trim;Fin;CaliperLength;SideDescription;←`

- **Description**: Profile identification code.
- **String type:** Alphanumerical

1.5) **Profile length:**

`CodeId;BarCode;ProfPos;ProfileCode;`Length`;Trim;Fin;CaliperLength;SideDescription;←`

- **Description**: Profile length. Values in 1/10mm.
- **String type:** Numerical

1.6) **Trimming size** (optional)**:**

`CodeId;BarCode;ProfPos;ProfileCode;Length;`Trim`;Fin;CaliperLength;SideDescription;←`

- **Description** Trimming size, how much must remain from the stop. Values in 1/10mm.
  If you do not have to trim:
  o Leave empty field, or
  o Put 99, or
  o The dimension of complete fin, in this case must present the following field "Fin".
- **String type:** Numerical
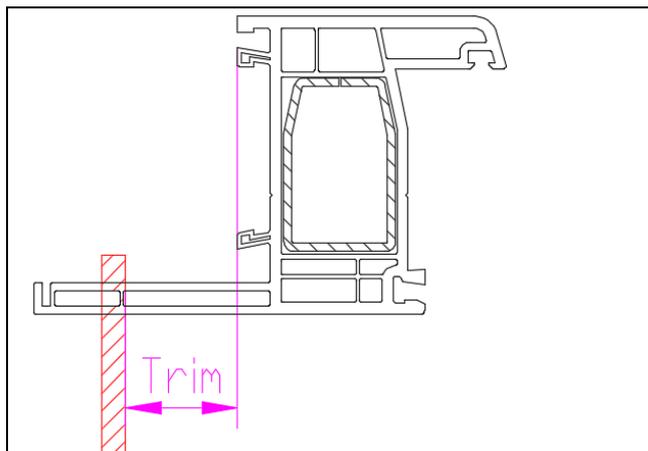
1.7) Fin size of Z-frame (optional):
`CodeId;BarCode;ProfPos;ProfileCode;Length;Trim;Fin;CaliperLength;SideDescription;←`

- **Description:** Fin size of Z-frame. Values in 1/10mm.
- **String type:** Numerical
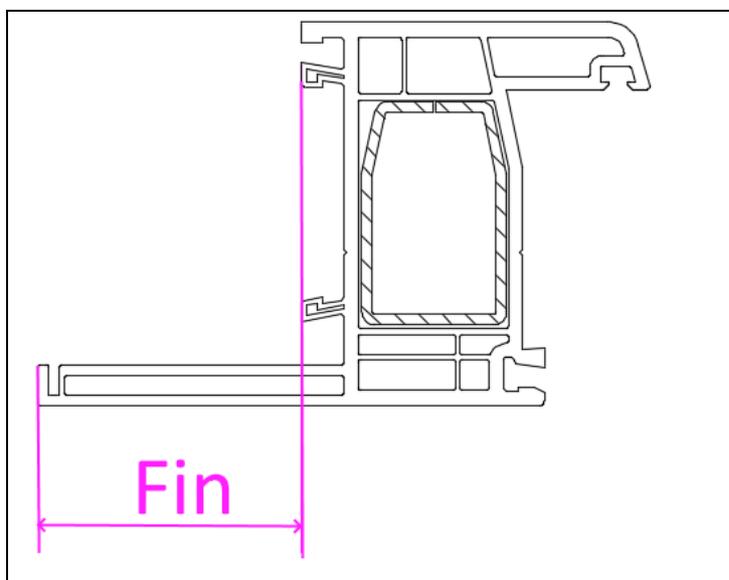
1.8) Combination of profiles code or dimension to put the "caliper" (optional):
`CodeId;BarCode;ProfPos;ProfileCode;Length;Trim;Fin;CaliperLength;SideDescription;←`

NOTA: It has double meaning depending on the type of machine

- For **SL2FF EVO:**
  - **Description:** Profile code, if the square consists of only one type of profile, or the combination of the profile codes, in the case of a square consisting of several types of profile.
  - **String type:** Alphanumeric
- For SL4FF of all type:
  - **Description:** Dimension to put the "caliper", see the following image. Use only for these window type. Values in 1/10mm.
  - **String type:** Numerical
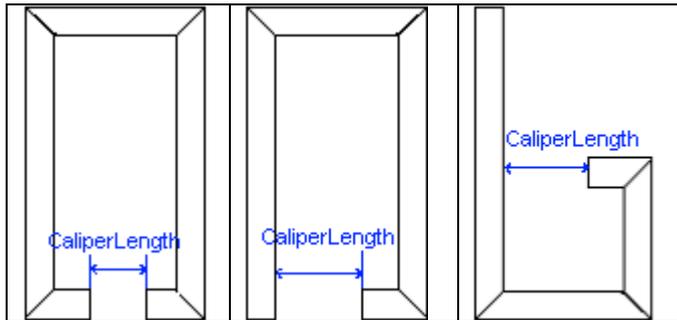
1.9) Description of square side (optional)**:**

`CodeId;BarCode;ProfPos;ProfileCode;Length;Trim;Fin;CaliperLength;SideDescription;←`

- **Description:** Description of square side. For EVO and WOOD version this information will shown on monitor
- **String type:** Alphanumerical

2) **Machine configuration row**:

2.1) Profile code or profile program in case of combinate profile:

`ProfProg;SquaredId;WinID;PieceForWin;Box;Order;Customer;FreeComment;ProfileType;OpN;WinType;←`

- **Description**: Profile identification code. In the case of "square" composed of different profiles, it will be agreed a name, which is usually the name of the profiles that make up the "square" separated by "-" or "+".
- **String type:** Alphanumerical

2.2) "Squares" identification (optional):

`ProfProg;SquaredId;WinID;PieceForWin;Box;Order;Customer;FreeComment;ProfileType;OpN;WinType;←`

- **Description**: "Square" identification code. The code must to be unique in order list. If possible this code should start with the order number.
- **String type:** Alphanumerical

2.3) Window identification (optional):

`ProfProg;SquaredId;WinID;PieceForWin;Box;Order;Customer;FreeComment;ProfileType;OpN;WinType;←`

- **Description**: Window identification code. The code must to be unique in order list. If possible this code should start with the order number.
- **String type:** Alphanumerical

2.4) Window pieces number (optional):

`ProfProg;SquaredId;WinID;PieceForWin;Box;Order;Customer;FreeComment;ProfileType;OpN;WinType;←`

- **Description**: Total pieces window number.
- **String type:** Numeric

2.5) Box:

`ProfProg;SquaredId;WinID;PieceForWin;Box;Order;Customer;FreeComment;ProfileType;OpN;WinType;←`

- **Description**: Position where insert the clean "square".
- **String type:** Alphanumerical

2.6) Order:

`ProfProg;SquaredId;WinID;PieceForWin;Box;Order;Customer;FreeComment;ProfileType;OpN;WinType;←`

- **Description**: Description for the order acknowledge.
- **String type:** Alphanumerical

2.7) Customer:

`ProfProg;SquaredId;WinID;PieceForWin;Box;Order;Customer;FreeComment;ProfileType;OpN;WinType;←`

- **Description**: Description for the customer acknowledge.
- **String type:** Alphanumerical

2.8) Comment:

`ProfProg;SquaredId;WinID;PieceForWin;Box;Order;Customer;FreeComment;ProfileType;OpN;WinType;←`

- **Description**: Field to insert a comment.
- **String type:** Alphanumerical

2.9) **Profile type**:

`ProfProg;SquareId;WinID;PieceForWin;Box;Order;Customer;FreeComment;`**`ProfileType`**`;OpN;WinType;←`

- **Description**: Field to describe the profile type, i. e. sash or frame
- **String type:** Alphabetical
- **Acceptable strings:** "**SASH**" for the sashes, "**FRAME**" for the frames.

2.10) **Operator identify(optional)**:

`ProfProg;SquareId;WinID;PieceForWin;Box;Order;Customer;FreeComment;ProfileType;`**`OpN`**`;WinType;←`

- **Description**: Field to describe the operator.
- **String type:** Alphabetical

2.11) **Window Type(optional)**:

`ProfProg;SquareId;WinID;PieceForWin;Box;Order;Customer;FreeComment;ProfileType;OpN;`**`WinType`**`;←`

- **Description**: Field used to describe the type of window, that is, if it is a window, door or sliding.
- **String type:** Alphabetical
- **Acceptable strings:** "**WINDOW**" for the windows, "**DOOR**" for the doors, "**SLIDING**" for the sliding, "**WASISTAS**" for wasistas.

3) **Window type configuration rows:**

   3.1) **Block identification:**

   `FrameInfo``;FirstBarcode;SecondBarcode;ThirdBarcode;FourthBarcode;Height;Width;`
   `SashInfo``;FirstBarcode;SecondBarcode;ThirdBarcode;FourthBarcode;Height;Width;`

   - **Description**: Use the keyword "**FrameInfo**" and "**SashInfo**". This identifies the rows of windows type. The row "**FrameInfo**" can put in the block of sash and "**SashInfo**" can put in the block of frame and in other sashes.
   - **String type:** Alphabetical
     - **Acceptable strings:** "**FrameInfo**" and "**SashInfo**"
     Other values generate errors.

   3.2) **Barcode of square**:

   `FrameInfo;`**`FirstBarcode`**`;`**`SecondBarcode`**`;`**`ThirdBarcode`**`;`**`FourthBarcode`**`;Height;Width;`
   `SashInfo;`**`FirstBarcode`**`;`**`SecondBarcode`**`;`**`ThirdBarcode`**`;`**`FourthBarcode`**`;Height;Width;`

   - **Description**: Are four barcodes of square.
   - **String type:** Alphabetical

   3.3) **Height of square**:

   `FrameInfo;FirstBarcode;SecondBarcode;ThirdBarcode;FourthBarcode;`**`Height`**`;Width;`
   `SashInfo;FirstBarcode;SecondBarcode;ThirdBarcode;FourthBarcode;`**`Height`**`;Width;`

   - **Description**: Height of square. In 1/10mm.
   - **String type:** Numerical

   3.4) **Width of square**:

   `FrameInfo;FirstBarcode;SecondBarcode;ThirdBarcode;FourthBarcode;Height;`**`Width`**`;`
   `SashInfo;FirstBarcode;SecondBarcode;ThirdBarcode;FourthBarcode;Height;`**`Width`**`;`

   - **Description**: Width of square. In 1/10mm.
   - **String type:** Numerical

4)

4. **Window configuration rows:Gasket type configuration:**

   4.1.1) **Gasket identification:**

   `Gasket``;TypeGasket;←`

   - **Description**: Insert the keyword "**Gasket**".
   - **String type:** Alphabetical
     - **Acceptable string:** "**Gasket**"

Other values generate errors.

4.1.2) **Gasket presence**:

`Gasket;`<mark>`TypeGasket`</mark>`;←`

- **Describe**: Indicates the gasket presence. Enter "**G**" if the gasket is present, or "" or "" if the gasket is absent.
- **String type:** Alphanumerical
  o **Acceptable strings:**
    - "**G**" if the gasket is present.
    - "" or " " if the gasket isn't present.

4.2) **Inside color type configuration:**

4.2.1) **Inside color identifier**:

<mark>`ColorInside`</mark>`;TypeColorInside;←`

- **Description**: Insert the keyword "**ColorInside**".
- **String type:** Alphabetical
  o **Acceptable string:** "**ColorInside**"
  Other values generate errors.

4.2.2) **Inside color type**:

`ColorInside;`<mark>`TypeColorInside`</mark>`;←`

- **Description**: Insert:
    - "**WH**" or "**W**" if the window is white inside.
    - "**C**" or "**C**" if the window is peeled inside.
    - "**P**" or "**P**" if the window is colored inside.
- **String type:** Alphanumerical

4.3) **Outside color type configuration:**

4.3.1) **Outside colour identification**:

<mark>`ColorOutside`</mark>`;TypeColorOutside;←`

- **Description**: Insert the keyword "**ColorOutside**".
- **String type:** Alphabetical
  o **Acceptable string:** "**ColorOutside**"
  Other values generate errors.

4.3.2) **Outside colour type**:

`ColorOutside;`<mark>`TypeColorOutside`</mark>`;←`

- **Description**: Insert:
    - "**WH**" or "**W**" if the window is white outside.
    - "**C**" or "**C**" if the window is peeled outside.
    - "**P**" or "**P**" if the window is colored outside.
- **String type:** Alphanumerical

4.4) **Window height configuration:**

4.4.1) **Window height identifier left side**:

<mark>`Height`</mark>`;ValueHeight;←`

- **Description**: Insert the keyword "**Height**".
- **String type:** Alphabetical
  o **Acceptable string:** "**Height**"
  Other values generate errors.

4.4.2) **Window height left side**:

`Height;`<mark>`ValueHeight`</mark>`;←`

- **Description**: Insert the window height, or in the case of different height, put the left side height (see the following picture). Values in 1/10mm.
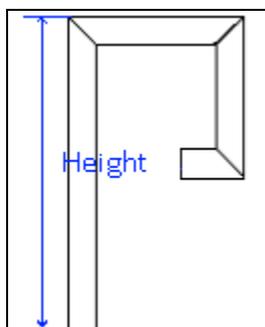
- o **NOTE**: If the welder is SL4FF:
  - ▪ The size of "square" must be a finished dimension, i.e. without burning.
  - ▪ If the "square" consists of different profiles, one must consider having the larger profile on all sides.
  - ▪ In the case of frames for door, i.e. with three sides, the size must not take into account any additional thresholds.
- **String type:** Numerical

4.4.3) **Window height identifier right side**:

<mark>`Height`</mark>`1;ValueHeight1;←`

- **Description**: Insert the keyword "**Height1**".
- **String type:** Alphabetical
- o **Acceptable string:** "**Height1**"

  Other values generate errors.

4.4.4) **Window height right side**:

`Height1;`<mark>`ValueHeight`</mark>`1;←`

- **Description**: Insert the window height, or in the case of different height, put the right side height (see the following picture). Values in 1/10mm.
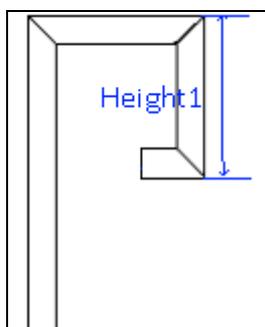
- o **NOTE**: If the welder is SL4FF:
  - ▪ The size of "square" must be a finished dimension, i.e. without burning.

> > ▪ If the "square" consists of different profiles, one must consider to have the larger profile on all sides.
> > ▪ In the case of frames for door, i.e. with three sides, the size must not take into account any additional thresholds.
> - **String type:** Numerical

4.5) **Window width configuration:**
  4.5.1) **Window width identifier**:
  `Width;ValueWidth;←`
  - **Description**: Insert the keyword "**Width**".
  - **String type:** Alphabetical
    o **Acceptable string:** "**Width**"
    Other values generate errors.
  4.5.2) **Window width**:
  `Width;ValueWidth;←`
  - **Description**: Insert the window width. Values in 1/10mm.
    o **NOTE**: If the welder is SL4FF:
      ▪ The size of "square" must be a finished dimension, i.e. without burning.
      ▪ If the "square" consists of different profiles, one must consider to have the larger profile on all sides.
  - **String type:** Numerical
4.6) **Angles number to clean configuration and window type:**
  4.6.1) **Angles number identifier:**
  `NoAngle;QtyAngle;MWType;←`
  - **Description**: Insert the keyword "**NoAngle**".
  - **String type:** Alphabetical
    o **Acceptable string:** "**NoAngle**"
    Other values generate errors.
  4.6.2) **Angles number**:
  `NoAngle;QtyAngle;MWType;←`
  - **Description**: Insert the angles number to welding or clean.
    o "**2**" for doors with threshold
    o "**3**" for special window with 3 angles to welding (optional)
    o "**4**" for standard square
  - **String type:** Numerical
    o **Acceptable string:** "**2**", "**3**", "**4**".
    Other values generate errors.
  4.6.3) **Window type**:
  `NoAngle;QtyAngle;MWType;←`
  - **Description**: Insert the number associated with the type of window as shown in the following image.
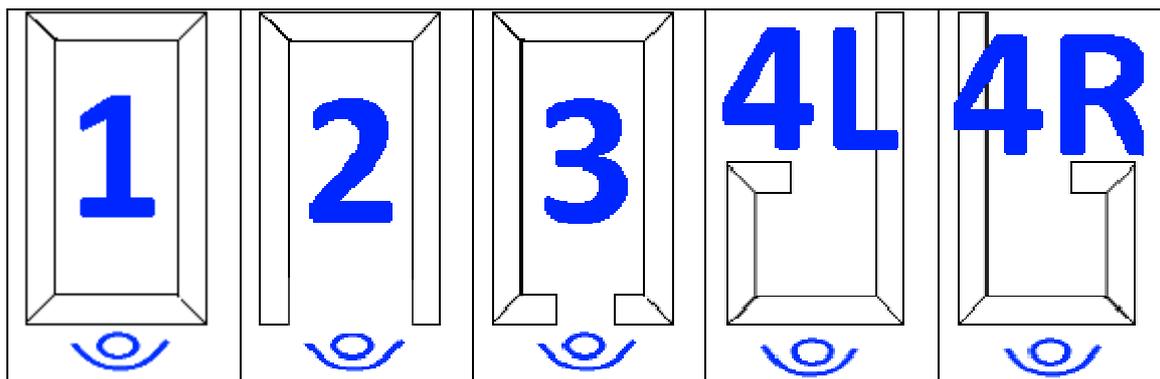
- o "**1**" for standard square.
- o "**2**" for doors with threshold.
- o "**3**" for special window like image.
- o "**4L**" for special window like image with left side more long.
- o "**4R**" for special window like image with right side more long.
  - **String type:** Alphanumeric
    - o **Acceptable string:** "**1**", "**2**", "**3**", "**4L**", "**4R**".
    - Other values generate errors.

5) **Working configuration rows** (optional, only for corner cleaners WP-CNC2 and WP-CNC4 with working system and HDHS machine for hinge on sash, HDMV machine for hinge on frama) **(to repeat for each working to be included on the "square"):**

5.1) **Working identification**:

`Fab;WorkName;Posit;ProfPos;←`

- **Description**: Insert the keyword "**Fab**". This keyword indicates the start of a new working configuration row.
- **String type:** Alphabetical
  - o **Acceptable string:** "**Fab**"
 - Other values generate errors.

5.2) **Working name**:

`Fab;WorkName;Posit;ProfPos;←`

- **Description**: Generic name for the working.
- **String type:** Alphanumerical
- **NOTE**: For the HDMV, for the "Wasistas" type windows the name of the upper hinge must be the left one for the type with the hinges at the top (side 3) and the right one for the type with the hinges at the bottom. This is because the window will be rotated 90° counterclockwise and side 4 will be at the bottom (see image) when it is loaded on the machine.
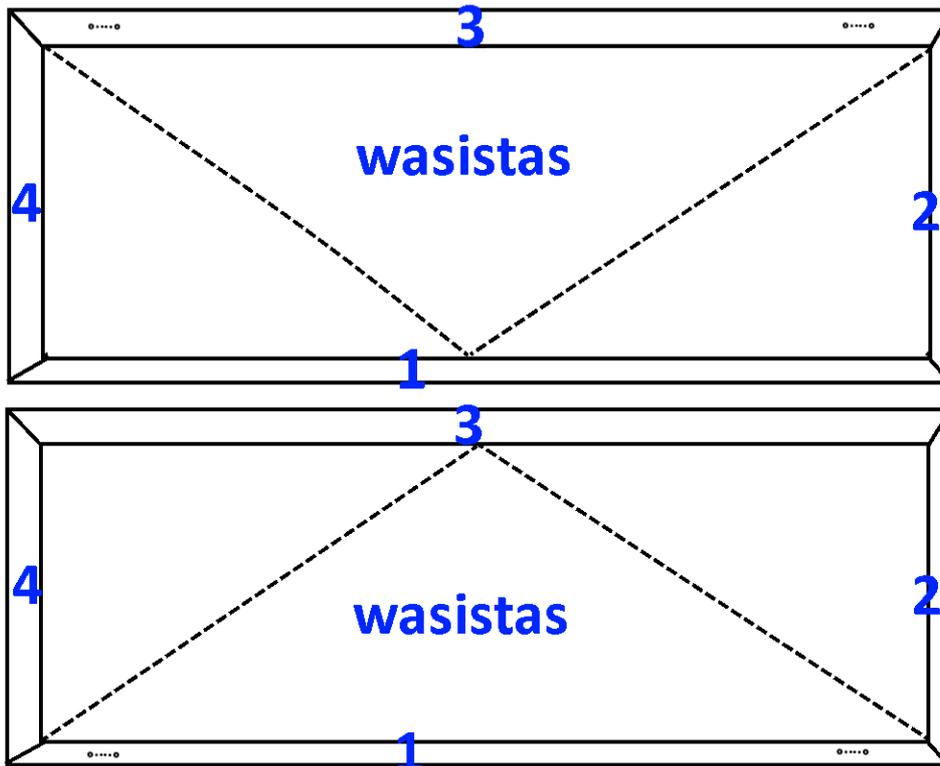
It will be loaded on the machine as shown in the following image.
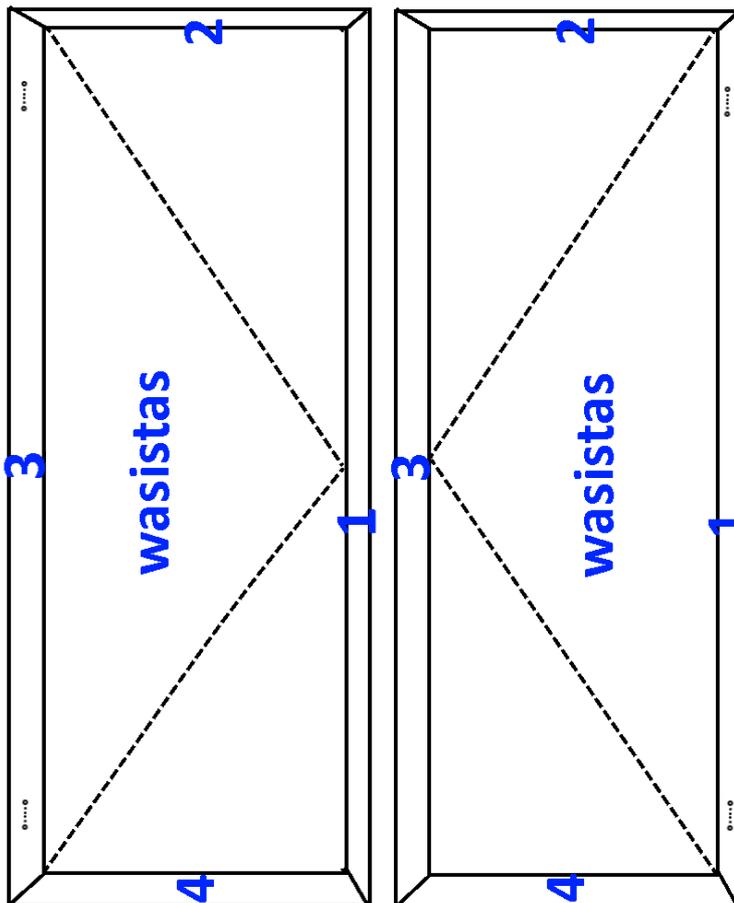
5.3) **Working position:**
`Fab;WorkName;`<mark>`Posit`</mark>`;ProfPos;←`
- **Description**:
  o This indicates the position where the working must be performed.
  o The value is in tenths of a millimeter (1/10mm). Example 123.5mm = "**1235**"
  o See paragraph "**Workings quotas**", for the rules by which to quote the working.
  o **String type:** Numerical
  o **Acceptable strings**: "**0**"..."**99999**" These values are permissible for the protocol. The real
   permissible values depended by machine type. See technical
    machine book.
  Others values generation errors.

5.4) **Profile position**:
`Fab;WorkName;Posit;`<mark>`ProfPos`</mark>`;←`
  o **Describe**: Profile position inside the square. Using the convention that the following image
    show the internal view of the frame, the positions of the profiles, will have the following
    numbering:
  ➢ "**1**" for the bottom profile.
  ➢ "**2**" for the right profile.
  ➢ "**3**" for the up profile
  ➢ "**4**" for the left profile.
  ➢ "**5**" for the mullion.
  ➢ For the HDMV you can also have them:
    o "**5R**" for the first traverse starting from the left if the opening of the window is on
      the right.
    o "**5L**" for the first traverse starting from the left if the opening of the window is on
      the left.
    o "**6R**" for the second traverse starting from the left if the opening of the window is
      on the right.
    o "**6L**" for the second traverse starting from the left if the opening of the window is
      on the left.
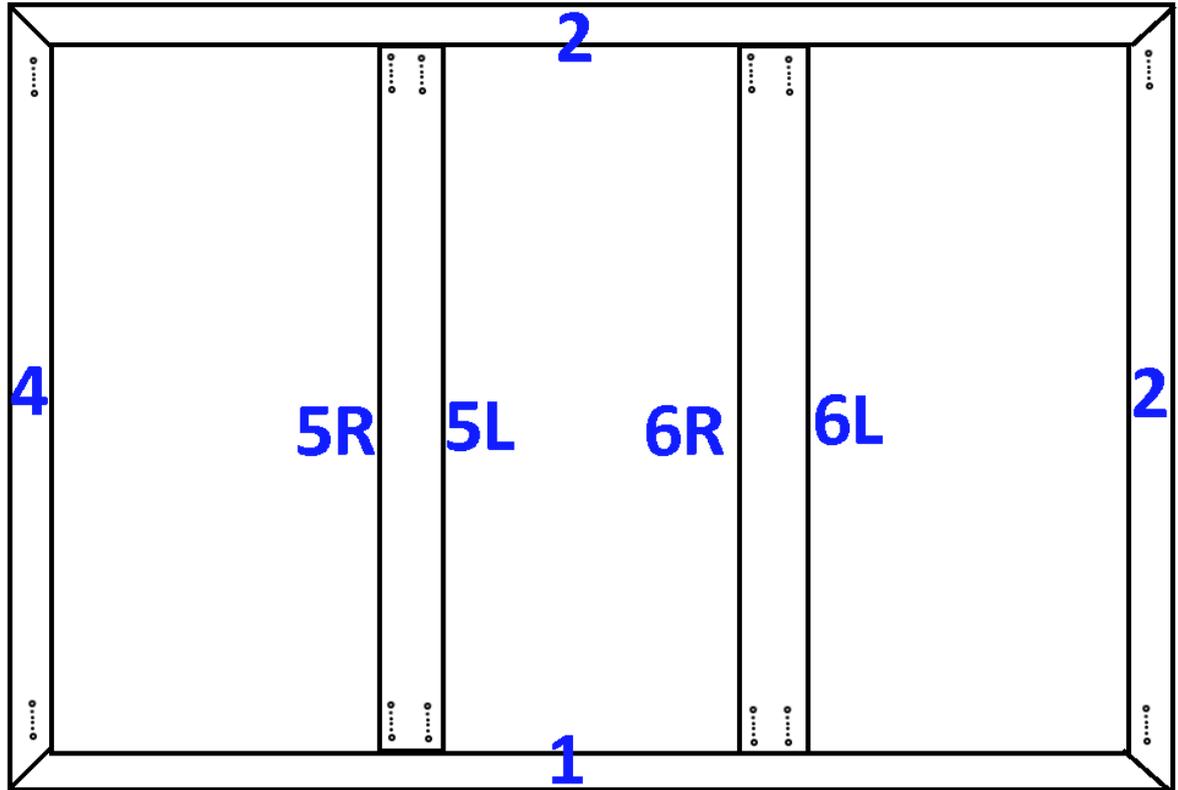
- **String type:** Numerical
  - o **Acceptable strings:** "**1**", "**2**", "**3**", "**4**", "**5**", for HDMV also "**5R**", "**5L**", "**6R**", "**6L**". Other values generate errors.

6) **Mullion configuration row** (optional, only for SL6Q and SL8TFF, not enter with the other machines) **(to repeat for each mullion to be included on the "square"):**

   6.1) **Mullion identification**:

`Mullion;Posit;ProfPos;←`

- **Description**: Insert the keyword "**Mullion**". This keyword indicates the start of a new working configuration row.
- **String type:** Alphabetical
  - o **Acceptable string:** "**Mullion**"
  - Other values generate errors.

   5.2) **Mullion positioning:**

`Mullion;Posit;ProfPos;←`

- **Description**:
  - o Distance from the mullion center, outside the profile:
    - ▪ below, if the mullion is in horizontal position
    - ▪ to left, if the mullion is in vertical position
  - o The value is in tenths of a millimeter (1/10mm). Example 123.5mm = "**1235**".
  - o If the mullion is absent, it is essential to enter "**0**".
- **String type**: numerical
  - o **Acceptable strings:** "**0**"… "**99999**". These values are those which are acceptable for the protocol. Those acceptable actually depend on the machine type. Refer to the machine technical data.
  - Other values generate errors.

5.3) **Mullion position:**

`Mullion;Posit;ProfPos;←`

- **Description:** Profile position inside the square.
  - ➢ "**0**" if the mullion is absent.
  - ➢ "**1**" if the mullion is in horizontal position.
  - ➢ "**2**" if the mullion is in vertical position.
- **String type:** Numerical
  - o **Acceptable strings:** "**0**", "**1**", "**2**".

  Other values generate errors.

7) **Hinge drills configuration rows** (optional, only for HDH400) **(to repeat for each side of "square" where do the hinge drill):**

7.1) **Hinge drill identification**:

`Hinge;HingeName;ProfPos;HingeQty;←`

- **Description**: Insert the keyword "**Hinge**". This keyword indicates the start of a new hinge drill configuration row.
- **String type:** Alphabetical
  - o **Acceptable string:** "**Hinge**"

  Other values generate errors.

7.2) **Hinge drill name**:

`Hinge;HingeName;ProfPos;HingeQty;←`

- **Description**: Generic name for the hinge drill.
- **String type:** Numerical
  - o **String type:** Alphanumerical

7.3) **Profile position**:

`Hinge;HingeName;ProfPos;HingeQty;←`

- o **Describe**: Profile position inside the square. Using the convention that the internal view of the frame is on top of the machine, the positions of the profiles, will have the following numbering:
  - ➢ "**2**" for the right profile.
  - ➢ "**4**" for the left profile.
- **String type:** Numerical
  - o **Acceptable strings:** "**2**", "**4**".

  Other values generate errors.

7.4) **How many hinge, to do**:

`Hinge;HingeName;ProfPos;HingeQty;←`

- o **Describe**: Indicate how many and where do the hinges:
  - ➢ "**1**" only upper hinge.
  - ➢ "**2**" only bottom hinge.
  - ➢ "**3**" both.
- **String type:** Numerical
  - o **Acceptable strings:** "**1**", "**2**", "**3**".

  Other values generate errors.

8) **Label print configuration rows** (optional)**:**

8.1) **Hinge drill identification**:

`LabId;LabelNumber;←`

- **Description**: Insert the keyword "**LabId**". This keyword indicates the start of a new label print configuration row.
- **String type:** Alphabetical

o **Acceptable string:** "**LabId**"
Other values generate errors.
8.2) **Label print number**:
`LabId;``LabelNumber``;←`
- **Description**: Number identification of label to print.
- **String type:** Numerical
9) **End block row**:
9.1) **End block configuration:**
`End``;←`
- **Description**: Insert the keyword "**End**"
- **String type:** Alphabetical
  o **Acceptable string:** "**End**"
Other values generate errors.

1.3.   Resume table

The next table (Table 16) describes in shortly mode the various records.

String	Description	Type*	Note
Wel Protocol			
1) Window acknowledgment rows(Repeatable up to 4 times): **CodeId;BarCode;ProfPos;ProfileCode;Length;Trim;**←			
CodeId	Block identifier	A	Use keyword **"CodeId"**. Identifies the codes that make up a "square"
BarCode1.. BarCode4	Barcode	AN	Barcode that belongs to the "square"
ProfPos °P	Profile position	N	Profile position inside the "square": ➢ **"1"** for bottom profile. ➢ **"2"** for right profile. ➢ **"3"** for up profile. ➢ **"4"** for left profile. ➢ "**5**" for mullion. ➢ **"X"** for all other, as mullions, sash reverse ledges, etc.
ProfileCode (optional)	Profile code	AN	Profile identification code
Length (optional)	Profile length	N	Profile length. Value in 1/10mm.
Trim (optional)	Trimming size	N	Trimming size, how much must remain from the stop. Values in 1/10mm. If you do not have to trim, leave empty field.
Fin (optional)	Fin size of Z-frame	N	Fin size of Z-frame. Values in 1/10mm.
CaliperLength (optional)	Combination of profile code OR Dimension to put the "caliper"	AN/N	Profile code, if the square consists of only one type of profile, or the combination of the profile codes, in the case of a square consisting of several types of profile. OR Dimension to put the "caliper", see the upper image. Use only for these window type. Values in 1/10mm.
SideDescription (optional)	Description of square side	AN	Description of square side. For EVO and WOOD version this information will shown on monitor
2) Machine configuration row: **ProfProg;SquareId;WinID;PieceForWin;Box;Order;Customer;FreeComment;ProfileType;OpN;WinType;**←			
ProfProg	Profile code	AN	Profile identification code. In the case of "square" composed of different profiles, it will be agreed a name, which is usually the name of the profiles that make up the "square" separated by "-" or "+".
SquareId (optional)	"Squares" identifier	AN	"Square" identification code. The code must to be unique in the order list. If possible this code should start with the order code.
WinID (optional)	Windows identifier	AN	Window identification code. The code must to be unique in order list. If possible this code should start with the order code.

Wel Protocol			
String	Description	Type*	Note
PieceForWin (optional)	Pieces number for window	N	Total window pieces number.
Box	Box	AN	Box where to insert the clean "square".
Order	Order	AN	Order acknowledge description.
Customer	Customer	AN	Customer acknowledge description.
FreeComment	Comment	AN	Field used to insert a comment.
Table 16.1			
ProfileType (optional)	Profile type	A	Field to describe profile type. Use **"SASH"** for the sashes and **"FRAME"** for the frames.
OpN (optional)	Operator identify	AN	Field used to identify the operator.
WinType (optional)	Window type identify	AN	Field to identify the window type. Use **"WINDOW"** for the windows, **"DOOR"** for the doors, **"SLIDING"** for the sliding.
3) Window type configuration rows:			
FrameInfo;FirstBarcode;SecondBarcode;ThirdBarcode;FourthBarcode;Height;Width; SashInfo;FirstBarcode;SecondBarcode;ThirdBarcode;FourthBarcode;Height;Width;			
FrameInfo SashInfo	Block identifier	A	Use the keyword **"FrameInfo"** and **"SashInfo"**. This identifies the rows of windows type. The row **"FrameInfo"** can put in the block of sash and **"SashInfo"** can put in the block of frame and in other sashes.
FirstBarcode SecondBarcode ThirdBarcode FourthBarcode	Barcode	AN	Are four barcodes of square.
Height	Height of square	N	Height of square. In 1/10mm.
Width	Width of square	N	Width of square. In 1/10mm.
4) Window configuration rows:			
4.1) Gasket type configuration: **Gasket;TypeGasket;**			
Gasket	Gasket identifier	A	Use the keyword **"Gasket"**. Identifies the gasket type
TypeGasket	Gasket presence	AN	Use: ▪ **"G"** if gasket is present ▪ **""** or " if gasket isn't present.
4.2) Inside color type configuration: **ColorInside;TypeColorInside;**←			
ColorInside	Inside color identifier	A	Use the keyword **"ColorInside"**. Identifies the inside type color
TypeColor Inside	Internal color type	AN	Use: ▪ **"WH"** or **"W"** if the window is white inside. ▪ **"C"** or **"C"** if the window is peeled inside. ▪ **"P"** or **"P"** if the window is colored inside.
4.3) Outside color type configuration: **ColorOutside;TypeColorOutside;**←			

Wel Protocol			
String	**Description**	**Type***	**Note**
ColorOutside	Outside color identifier	A	Use the keyword **"ColorOutside"**. Identifies the outside type color
TypeColor Outside	Outside color type	AN	Use: ▪ **"WH"** or **"W"** if the window is white outside. ▪ **"C"** or **"C"** if the window is peeled outside. ▪ **"P"** or **"P"** if the window is colored outside.
4.4) Window height configuration left side: **Height;ValueHeight;**←			
Height	Window height identifier for left side	A	Use the keyword **"Height"**. Identifies the window height for left
ValueHeight	Window height for left side	N	Window height for left side. Values in 1/10mm.
4.5) Window height configuration right side: **Height1;ValueHeight1;**←			
Height1	Window height identifier for right side	A	Use the keyword **"Height1"**. Identifies the window height for right
ValueHeight1	Window height for right side	N	Window height for right side. Values in 1/10mm.
Table 16.2			
4.6) Window width configuration: **Width;ValueWidth;**←			
Width	Window width identification	A	Use the keyword **"Width"**. Identifies the window width
ValueWidth	Window width	N	Window width. Values in 1/10mm.
4.7) Angles number to clean configuration: **NoAngle;QtyAngle;**←			
NoAngle	Angles number identification	A	Use the keyword **"NoAngle"**. Identifies the angles number to clean or welding 2, 3, 4.
QtyAngle	Angles number	N	Insert the angles number to clean or welding.
MWType	Window type	AN	Insert the number associated with the type of window as shown in the upper image.
5) Working configuration rows ᵒᵖ (to repeat for each working to be included on the "square"): **Fab;WorkName;Posit;ProfPos;**←			
Fab	Working identification	A	Use the keyword **"Fab"**. This keyword indicates the start of a workings new configuration row.
WorkName	Working name	AN	Generic name for the working.
Posit	Working position	N	▪ This indicates the position where the working must to be performed. ▪ Value in tenths of a millimeter (1/10mm) ▪ Example 123.5mm = **"1235"**.

Wel Protocol			
String	**Description**	**Type***	**Note**
ProfPos	Profile position	N	Profile position inside the "square". Using the convention that the window inside view is on top of the machine, the profiles positions will have the following numbering: ➢ **"1"** for bottom profile. ➢ **"2"** for right profile. ➢ **"3"** for up profile. ➢ **"4"** for left profile. ➢ **"5"** for the mullion.
6) Mullion configuration row ^{op1}: **Mullion;Posit;ProfPos;←**			
Mullion	Mullion identification	A	Use the keyword **"Mullion"**. This keyword indicates the start of a mullion condiguration row.
Posit	Mullion positioning	N	o Distance from the mullion center, outside the profile: ▪ below, if the mullion is in vertical position, ▪ to left, if the mullion is in horizontal position. o Value in tenths of a millimeter (1/10mm). o Exemple 123.5mm = **"1235"**. o If the mullion in absent is essential to insert **"0"**.
Table 16.3			
ProfPos	Mullion position	N	Profile position inside the square: ➢ **"0"** if the mullion is absent. ➢ **"1"** if the mullion is in horizontal position. ➢ **"2"** if the mullion is in vertical position.
7) Hinge drilling configuration rows ^{op2} (to repeat for each side of "square" where do the hinge): **Hinge;HingeName;ProfPos;HingeQty;←**			
Hinge	Hinge drilling identification	A	Use the keyword **"Hinge"**. This keyword indicates the start of a hinge drilling new configuration row.
HingeName	Hinge name	AN	Generic name for the hinge.
ProfPos	Profile position	N	Profile position inside the "square". Using the convention that the window inside view is on top of the machine, the profiles positions will have the following numbering: ➢ **"2"** for right profile. ➢ **"4"** for left profile.
HingeQty	How many hinge to do.	N	How many hinges to do: ➢ **"1"** only upper hinge. ➢ **"2"** only bottom hinge. ➢ **"3"** both hinges.
8) Label print configuration rows (optional): **LabId;LabelNumber;←**			
LabId	Label identification	A	Use the keyword **"LabId"**. Identifies the label print row.

Wel Protocol			
String	**Description**	**Type***	**Note**
LabelNumber	Number identification	N	Number identification of label to print
9) Block end configuration: **End;**←			
End	End block	A	Use the keyword **"End"**. Identifies the block end.
Table 16.4			

Note: *   Type:
         • AN = Alphanumerical
         • N = Numerical
         • A = Alphabetical
   op: optional, only for corner cleaners WP-CNC2 and WP-CNC4 with working system.
   op¹: optional, only for welders SL6Q, not enter with the other machines.
   Op²: optional, only for HDH4000 (hinge drilling machine), not enter with the other machines.			

1.4. Examples

With reference to the technical data in the [Appendix A](#), Following are three protocol examples:
 1) File "**00001.Wel**", for two sashes window
 2) File "**00002.Wel**", for a door.
 3) File "**00003.Wel**", for a window with a fixed sash and an openable one, with a mullion on the frame and one on the openable door. This example is referred to six heads welder SL6Q.
 4) File "**00004.Wel**", for two sashes window. Example for hinge drilling machine for frame (HDH4000)

1) File "**00001.Wel**":

```
CodeId;000010000001;1;1505;14000;0;;;;;
CodeId;000010000002;2;1505;13000;100;;;;
CodeId;000010000003;3;1505;14000;;;;;;
CodeId;000010000004;4;1505;13000;100;;;;
1505;0001000101;0001010001;12;1;Graf00001;Graf Gynergy;Room1;FRAME;Op01;WINDOW;
SashInfo;000010000005;000010000006;000010000007;000010000008;11000;6000;
SashInfo;000010000009;000010000010;000010000011;000010000012;11000;6000;
Gasket;G;
ColorInside;C;
ColorOutside;C;
Height;13000;
Height1;13000;
Width;14000;
NoAngle;4;1;
Fab;HINGE_FRAME;1325;2;
Fab;HINGE_FRAME;11725;2;
Fab;HINGE_FRAME;1325;4;
Fab;HINGE_FRAME;11725;4;
LabId;001;
End;
CodeId;000010000005;1;1601;6000;;;;;;
CodeId;000010000006;2;1601;11000;;;;;;
CodeId;000010000007;3;1601;6000;;;;;
CodeId;000010000008;4;1601;11000;;;;;;
1601;0001000201;0001010001;12;2;Graf00001;Graf Gynergy;Room1;SASH;Op01;WINDOW;
FrameInfo;000010000001;000010000002;000010000003;000010000004;13000;14000;
SashInfo;000010000009;000010000010;000010000011;000010000012;6000;6000;
Gasket;G;
ColorInside;WH;
ColorOutside;WH;
Height;11000;
Height1;11000;
Width;6000;
NoAngle;4;1;
LabId;002;
End;
CodeId;000010000009;1;1601;6000;;;;;;
CodeId;000010000010;2;1601;11000;;;;;;
CodeId;000010000011;3;1601;6000;;;;;
CodeId;000010000012;4;1601;11000;;;;;;
1601;0001000301;0001010001;12;3;Graf00001;Graf Gynergy;Room1;SASH;Op01;WINDOW;
FrameInfo;000010000001;000010000002;000010000003;000010000004;13000;14000;
```

```
SashInfo;000010000005;000010000006;000010000007;000010000008;6000;6000;

Gasket;G;
ColorInside;WH;
ColorOutside;WH;
Height;11000;
Height1;11000;
Width;6000;
NoAngle;4;1;
LabId;003;
End;
```

2) File "**00002.Wel**":

```
CodeId;000020000001;2;1505;24000;;;;;
CodeId;000020000002;3;1505;13000;;;;;
CodeId;000020000003;4;1505;24000;;;;;
1505;0002000101;0002010001;7;1;Graf00002;Graf Gynergy;Room2;FRAME;Op01;DOOR;
Gasket;G;
ColorInside;C;
ColorOutside;C;
Height;24000;
Height1;24000;
Width;13000;
NoAngle;2;2;
LabId;001;
End;
CodeId;000020000004;1;1601;11000;;;;;
CodeId;000020000005;2;1601;23000;;;;;
CodeId;000020000006;3;1601;11000;;;;;
CodeId;000020000007;4;1601;23000;;;;;
1601;0002000201;0002010001;7;2;Graf00002;Graf Gynergy;Room2;SASH;Op01;DOOR;
Gasket;G;
ColorInside;WH;
ColorOutside;WH;
Height;23000;
Height1;23000;
Width;11000;
NoAngle;4;1;
LabId;002;
End;
```

3) File "**00003.Wel**":

```
CodeId;000030000001;1;1505;14000;200;;;;
CodeId;000030000002;2;1505;13000;;;;;
CodeId;000030000003;3;1505;14000;;;;;
CodeId;000030000004;4;1505;13000;;;;;
CodeId;000030000005;5;1702;12000;;;;;
1505;0003000101;0003010001;10;1;Graf00003;Graf Gynergy;Room3;FRAME;Op01;WINDOW;
Gasket;G;
ColorInside;C;
ColorOutside;C;
Height;13000;
Height1;13000;
Width;14000;
NoAngle;6;1;
```

```
Mullion;7000;3;
LabId;001;
End;
CodeId;000030000006;1;1601;6000;;;;;
CodeId;000030000007;2;1601;11000;;;;;
CodeId;000030000008;3;1601;6000;;;;;
CodeId;000030000009;4;1601;11000;;;;;
CodeId;000030000010;5;1702;5000;;;;;
1601;0003000201;0003010001;10;2;Graf00003;Graf Gynergy;Room3;SASH;Op01;WINDOW;
Gasket;G;
ColorInside;WH;
ColorOutside;WH;
Height;11000;
Height1;11000;
Width;6000;
NoAngle;6;1;
Mullion;3830;2;
LabId;002;
End;
```

4) File "**00004.Wel**":
```
CodeId;000040000001;1;1505;14000;300;;;;
CodeId;000040000002;2;1505;13000;150;;;;
CodeId;000040000003;3;1505;14000;;;;;
CodeId;000010000004;4;1505;13000;150;;;;
1505;0004000101;0004010001;12;1;Graf00001;Graf Gynergy;Room1;FRAME;Op01;WINDOW;
Gasket;G;
ColorInside;C;
ColorOutside;C;
Height;13000;
Height1;13000;
Width;14000;
NoAngle;4;1;
LabId;001;
Hinge;HINGE_FRAME;2;3;
Hinge;HINGE_FRAME;4;3;
End;
CodeId;000040000005;1;1601;6000;;;;;
CodeId;000040000006;2;1601;11000;;;;;
CodeId;000040000007;3;1601;6000;;;;;
CodeId;000040000008;4;1601;11000;;;;;
1601;0004000201;0004010001;12;2;Graf00001;Graf Gynergy;Room1;SASH;Op01;WINDOW;
Gasket;G;
ColorInside;WH;
ColorOutside;WH;
Height;11000;
Height1;11000;
Width;6000;
NoAngle;4;1;
LabId;002;
End;
CodeId;000040000009;1;1601;6000;;;;;
CodeId;000040000010;2;1601;11000;;;;;
CodeId;000040000011;3;1601;6000;;;;;
CodeId;000040000012;4;1601;11000;;;;;
1601;0004000301;0004010001;12;3;Graf00001;Graf Gynergy;Room1;SASH;Op01;WINDOW;
Gasket;G;
ColorInside;WH;
```

```
ColorOutside;WH;
Height;11000;
Height1;11000;
Width;6000;
NoAngle;4;1;
LabId;003;
End;
```

2. Workings quotas

2.1. Rules for quoting workings.

With reference to the following picture, considering the window inside view , the rules for quoting workings are the following (arrow point of origin)
- For left profile is in up position.
- For bottom profile is in left position.
- For right profile is in bottom.
- For up profile is in right position.

By synthesizing the rule, we must start from the left end at the top and go anticlockwise. From these points, you need to give the workings quotas (L_1, L_2, L_3, L_4 in the picture).
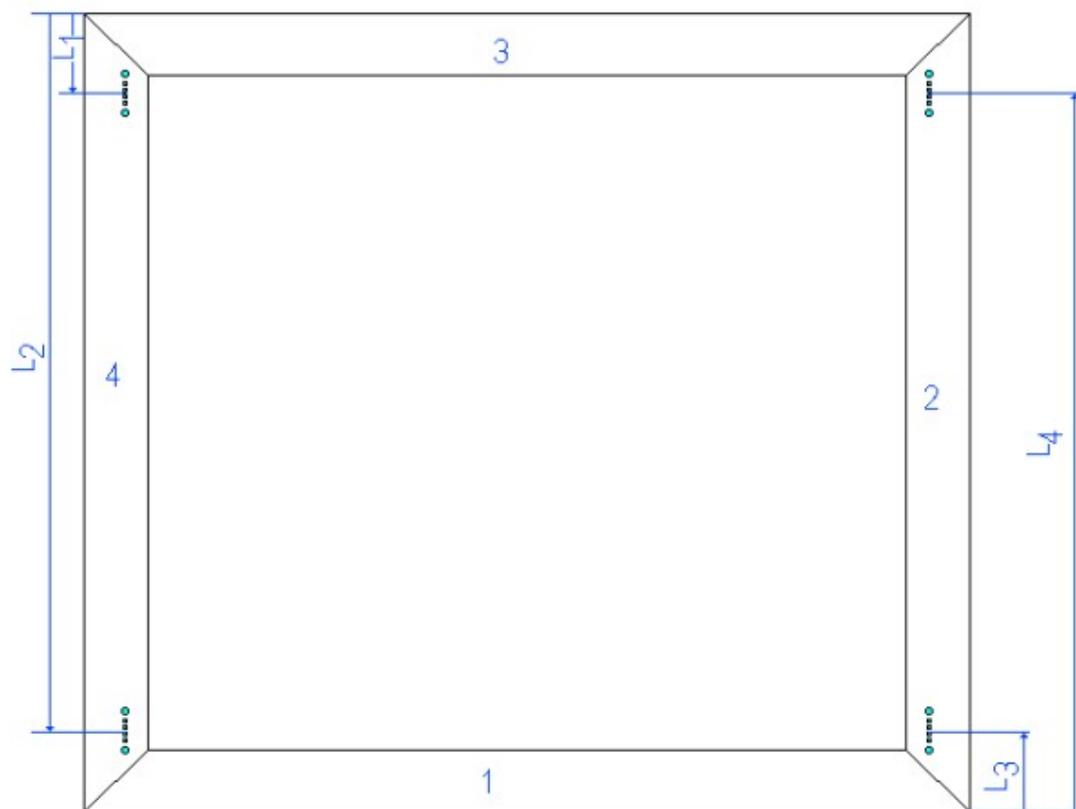
3. Appendix A

## 3.1. Cutting technical informations for two sashes window

In picture A.1, see cutting technical informations for two sashes window.
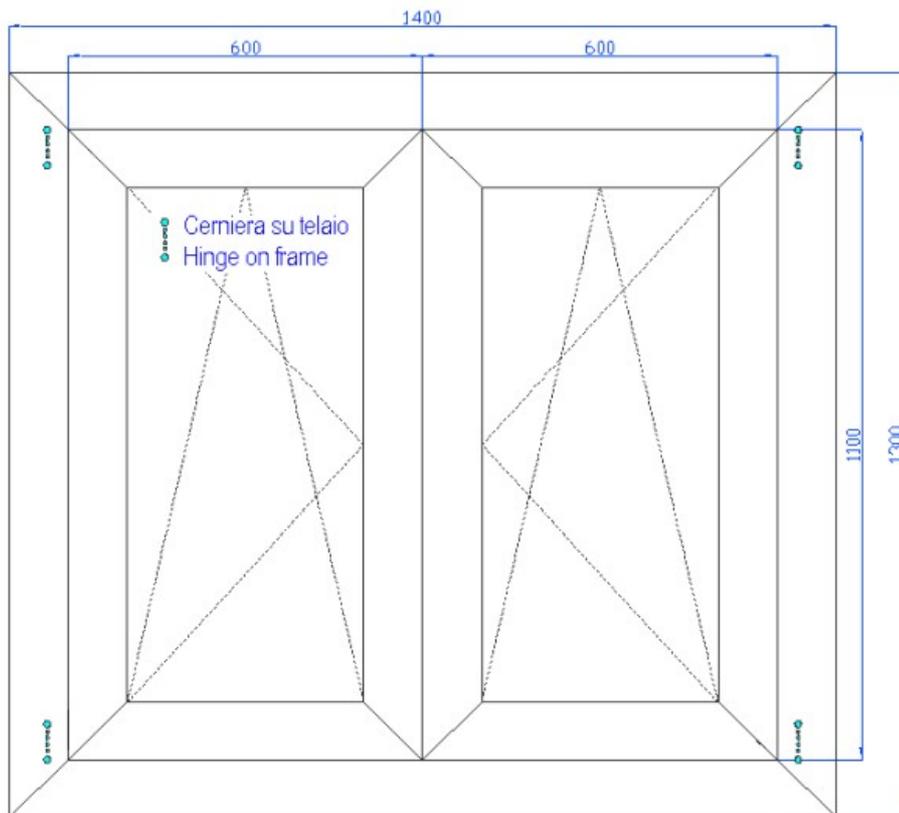
At these dimensions, must be added the burns (for the examples we use a 2.5mm burn).

3.2.   Cutting technical informations for door

In picture A.2, see cutting technical informations for door.
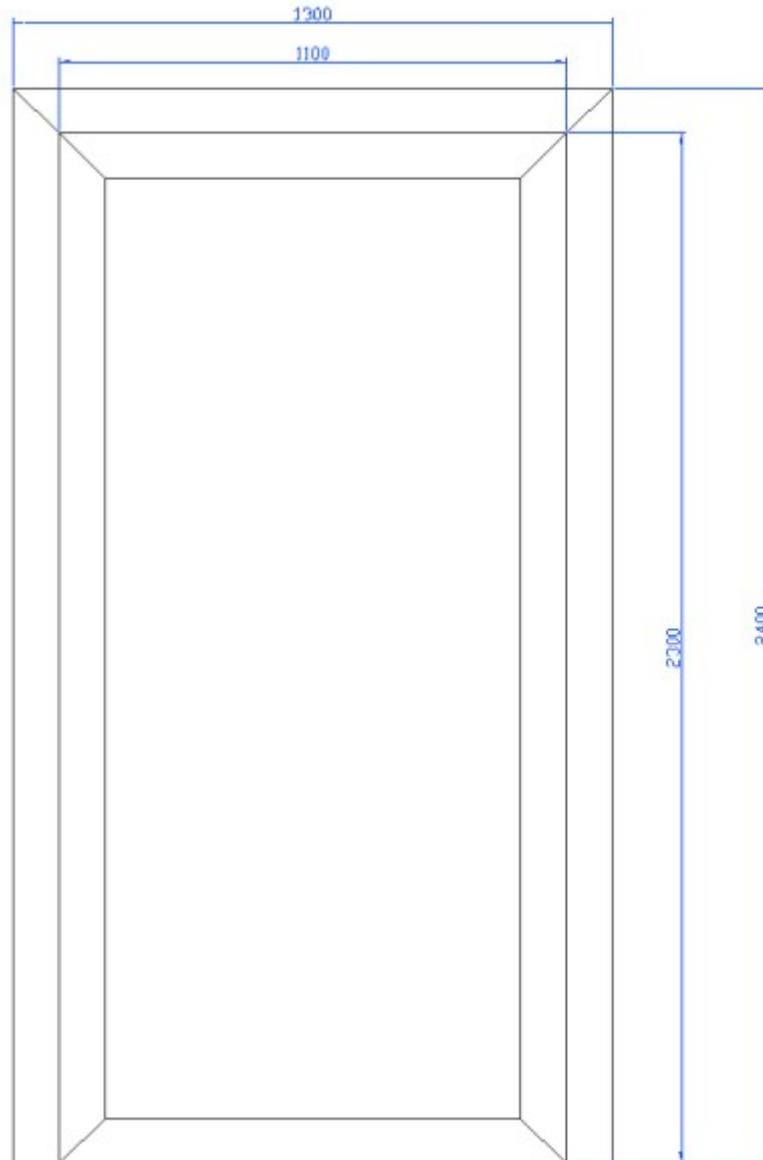
At these dimensions, must be added the burns (for the examples we use a 2.5mm burn).

## 3.3. Cutting information's for a window with a fixed and one sash

In picture A.3, see cutting technical informations for a window with a fixed sash and an openable one, with a mullion on the frame and one on the openable sash. Example for six heads welding SL6Q.
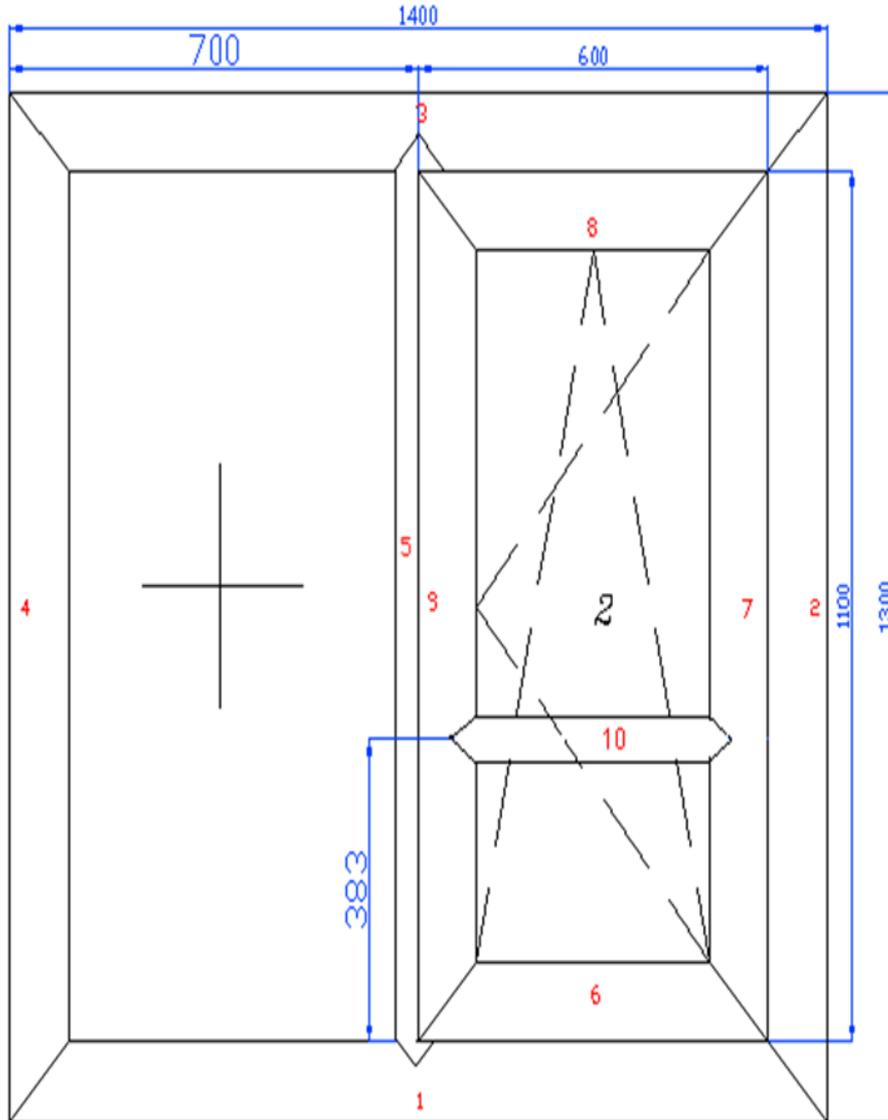
At these dimensions, see cutting technical informations that the customer provides on how to distribute the workings (hinges) for a two sashes window.

3.4. Workings (hinges) technical specifications for two sashes window

At these dimensions, see cutting technical informations that the customer provides on how to distribute the workings (hinges) for a two sashes window..
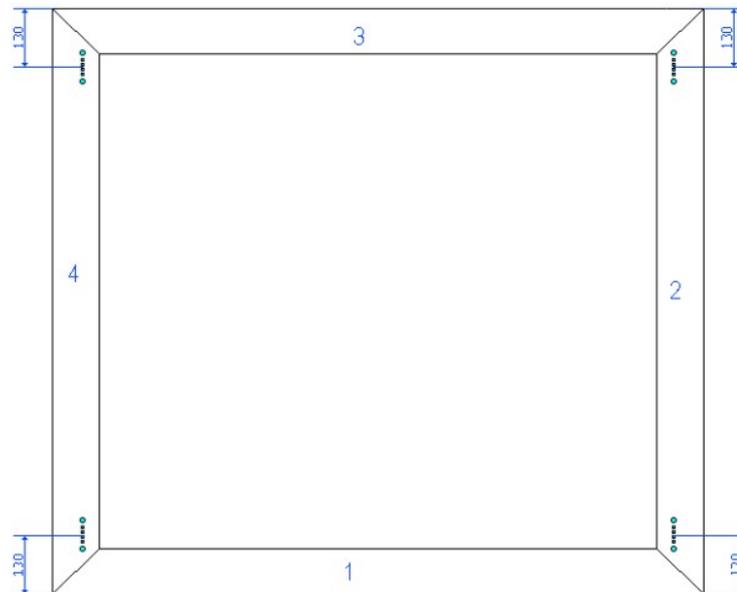
In the following picture, you see how to provide the quotas to the machine, by the program management office. for a two sashes window. At these dimensions, must be added the burns (for the examples we use a 2.5mm burn).