



DESCRIPTIVO ESTÁNDAR MINOMIS. PALETIZADO Y DESPALETIZADO

Contenidos:

DESCRIPTIVO ESTÁNDAR MINOMIS. PALETIZADO Y DESPALETIZADO.....	1
MINOMI STANDARD RENAULT.	2
CONDICIONES INICIALES	2
PREMISAS MINOMI	2
CODIFICACIÓN CARROS MINOMI.....	2
RECONOCIMIENTO DE CONTENEDOR.....	2
GESTIÓN DE FILAS Y POSICIONES MINOMI y/o CONTENEDOR.....	2
GESTIÓN DE ENVÍOS DE CÓDIGO DE ROBOT	3
ALARMA FIN DE FILA	4
FIN DE FILA	4
DEFECTO BÚSQUEDA	4
MINOMI STANDARD RENAULT. DESPALETIZADO	6
DETECCIONES EN GARRA	6
INTERCAMBIO DE SEÑALES ROBOT-PLC:	6
DESCRIPCIÓN DEL FUNCIONAMIENTO.....	6
VARIABLES ÚNICAS DEL MINOMI STANDARD.....	8
MINOMI STANDARD RENAULT. PALETIZADO	9
DETECCIONES EN GARRA	9
INTERCAMBIO DE SEÑALES	9
DESCRIPCIÓN DEL FUNCIONAMIENTO.....	9
VARIABLES ÚNICAS DEL MINOMI STANDARD. Paletizado	10



MINOMI STANDARD RENAULT.

CONDICIONES INICIALES

PREMISAS MINOMI

Cada “hueco” MINOMI es considerado como una zona de trabajo (Código de proceso).

Cada “hueco” MINOMI puede albergar un máximo de 7 carros (Codificación binaria mediante levas y detectores o Codificación en Balogh o RFID).

Cada Carro debe tener una codificación univoca, (A través de detección de levas o datos escritos en Balogh o RFID).

En una misma instalación, pueden existir varios “huecos” MINOMI con la misma funcionalidad.

CODIFICACIÓN CARROS MINOMI

El Nº de carro será controlado en todo momento.

Ante la presencia de un carro desconocido, la instalación no permitirá su validación y mostrará el mensaje correspondiente por IHM+SMPLOC.

Ante la presencia de un carro conocido y validado, se enviará al robot un código de ciclo específico por Nº de CARRO.

Si en un mismo carro se cargan diferentes diversidades, al número de carro se sumara un valor fijo que determinará el código de robot para cada carro y diversidad. Ejemplo:

- Carro nº3 div HFE = 3 + 0
- Carro nº6 div BFB = 6 + 20

RECONOCIMIENTO DE CONTENEDOR

El tipo de contenedor (por diversidad) será controlado en todo momento.

La correcta colocación del contenedor en el “hueco” (al menos dos detectores de posición) así como el estado de los peines será controlada en todo momento. (Nº de detectores a precisar en función del diseño del contenedor). En caso de un contenerizado se comprobará que el contenedor no contiene ninguna pieza. La detección errónea de peines o la detección de pieza en un contenerizado, generara un defecto en IHM y SMP además de impedir la evolución del robot.

Ante la presencia de un contenedor desconocido, la instalación no permitirá su validación y mostrará el mensaje correspondiente por IHM+SMPLOC.

GESTIÓN DE FILAS Y POSICIONES MINOMI y/o CONTENEDOR

Puesta a 0 de todas las memorias de gestión filas y/o posición:

- Si pérdida de seguridad minomi y/o ausencia carro.
- Si ausencia contenedor.



Para el set/reset de cualquier memoria referente a la gestión de filas:

- Es necesario tener la seguridad minomi ok.
- Es necesario tener la info contenedor ok. (En posición y Ctrl de Diversidad)

Cada vez que introducimos un nuevo carro y/o contenedor:

- Set NºFila $n=1$ + Init Fila (y POS = 1 en caso de contenerizado).
- La posición será dada por los datos de SIPTOL en contenerizados al FILM.

Con NºFila n y Flanco Ascendente Fin de Fila:

- Set NºFila $n+1$ y POS = 1, (La posición será dada por los datos de SIPTOL en contenerizados al FILM)
- Init Fila.
- Reset resto de Nº Filas.

Con Nº Última Fila y Flanco Ascendente Fin de Fila:

- Set Fin de Carro y/o Contenedor. (Aviso Operario IHMP+SMPLOC.)
- Reset resto de Nº Filas.

GESTIÓN DE ENVÍOS DE CÓDIGO DE ROBOT

Ante la presencia de un carro y/o contenedor conocido y validado, cuando se cumplan el resto de condiciones de la instalación, se enviará al robot:

- Un código de ciclo específico por Nº de CARRO en función de la diversidad, si tuviese varias diversidades a descontenerizar o contenerizar sobre el mismo carro.
- El robot responderá con un código de proceso en función del “hueco” y la posición (si es un contenerizado) sobre el que vaya a trabajar.

Gracias a este código de proceso, el autómatas podrá enviar al robot los datos actualizado correspondientes al “hueco” sobre el que el Robot va a trabajar (Nºde Fila, Inicio_Fila)

- La información Inicio Fila le indicará al robot que debe buscar la primera pieza de la fila en cuestión (búsqueda lenta). Una vez el robot haya cogido la primera pieza de la fila, enviará al autómatas la información ACK Init Fila (Enterado de inicio de fila). Esta información será utilizada por el autómatas para resetear la info Init Fila.
La información Inicio Fila la utilizará el robot en caso de contenerizado para resetear las señales de “alarma fin fila” y de “fin fila” y para dar información al PLC de que se ha enterado de que tenemos una nueva fila a través del envío de la señal “ACK_INI_FILA”.
- El código de fila se compondrá de la fila en la que tiene que trabajar el robot + la posición (en caso de contenerizado) o solo la fila en el caso de



ALARMA FIN DE FILA

Si estamos trabajando en la última fila y el robot nos envía la información “Alarma Fin de Fila” mostramos aviso operario por IHMP.

FIN DE FILA

Cuando el robot ha cogido la última pieza, nos dará la información Fin Fila que nos servirá para cambiar de fila. Si estamos trabajando en la última fila y el robot nos envía la info “Fin de Fila” activamos la memoria Fin de carro y/o contenedor.

En un contenerizado, cuando el robot ha dejado en la última posición envía al PLC la información de fin de fila. El autómatas cambiará el número de fila o si es la última fila a llenar, gestionará el fin de carro.

DEFECTO BÚSQUEDA

Defecto enviado por el robot tras una búsqueda fallida.

Ante este defecto, tenemos dos opciones:

- 1- Repetir el ciclo de búsqueda desde IHMP.
 - a. Con una primera pulsación en el botón “reinicio ciclo” se enviará un Evento al robot (previo envío de una ORDEN del robot “en posición Reinicio ciclo”. El operario deberá cerciorarse de que el robot puede volver a rebouclage y después pulsar el botón del IHMP.
 - b. El robot se va a rebouclage y esperará código. El operario si tiene que entrar a colocar piezas o realizar alguna operación, lo hará y posteriormente cuando todo esté en orden pulsará de nuevo el botón “reinicio ciclo” con lo que se enviará de nuevo código al robot.
 - c. El envío de reinicio ciclo al robot implica la activación de la información Inicio Fila para que el robot inicie la búsqueda como si se tratara de la primera pieza sin cambiar el Nº Fila. (depende del tipo de MINOMI y/o CONTENEDOR).
- 2- Abandonar el carro desde IHMP.



MDP - DMP V3.3.1

Defecto a leer Consigna a leer Información a leer

Retorno

Minomi V1	
Estado Minomi	
Nº de Carro	2
Nº de File	2
Nº de Piezas	31
Alarma cambio de carro	
Carga carro	
DPAS1	DPAS2
DPAS3	DPOH1
DPOH2	DCV1
DCV2	
Acción sobre el Minomi	
Dejar carro	
Autorización de evacuación de carro	
MAJ nombre pieza carro	Autorización MAJ Número de piezas
	0
Defecto Minomi	
Codigo de error	
	0
Reinicio Ciclo	
Etiquetado	0
Consigna	0

Anterior

Siguiente

Ayuda al reinicio Modos de Marcha Comandos manuales SMPLOC Sinóptico de isla Explotación Mantenimiento Procedimiento

Inicio Smploc2.14.2 - [Table... SIMATIC Manager - [381... MDP - DMP V3.3.1 KUKA Virtual Remote Pen... KOPAWUP - [9FB10...



MINOMI STANDARD RENAULT. DESPALETIZADO

DETECCIONES EN GARRA

DETECTOR LÁSER: En lógica negativa. A 1 si no detecta. El detector preconizado para esta aplicación es láser LEUZE (ref. ODSL 9/C6-450-S12)

DETECTOR(ES) INDUCTIVO(S): En lógica positiva. A 0 si no detecta.

En el caso de robots con Interbus, tanto los estos detectores inductivos como el láser estarán cableados a las entradas rápidas del robot.

INTERCAMBIO DE SEÑALES ROBOT-PLC:

ENTRADAS AL ROBOT

Con el código de programa ya sabemos el modelo y el carro.

SIGNAL COD_FILA \$IN[225] TO \$IN[234] ; 10 BITS. Nos indica el Número de fila (palabra 5)

SIGNAL INICIO_FILA \$IN[240]; Nos indica que es un inicio de fila (palabra 5)

SALIDAS DEL ROBOT

SIGNAL COD_PROCESS \$OUT[225] TO \$OUT[232] ; Rango para el código de proceso (palabra 5)

SIGNAL ACK_INI_FILA \$OUT[237] TO \$OUT[237] ; Enterado inicio fila (palabra 5)

SIGNAL DEF_BUSQUEDA \$OUT[238] TO \$OUT[238] ; Defecto busqueda (palabra 5)

SIGNAL Alarma_Fin_Fila_X \$OUT[239] TO \$OUT[239] ; Activar alarma fin de fila (palabra 5)

SIGNAL Fin_Fila_X \$OUT[240] TO \$OUT[240] ; Activar fin de fila (palabra 5)

*****NOTA. EN ROBOTS CON MÁSTICO, ESTÁS SEÑALES ESTARÁN EN LA PALABRA 9. LA PALABRA 5 ES LA PALABRA DE LAS ORDENES Y EVENTOS ÚTIL. EN EL CASO DE ROBOT CON MÁSTICO SE UTILIZAN ORDENES ÚTIL PARA EL COMANDO DE LA APERTURA Y CIERRE DE LAS PISTOLAS. (Ejemplo V0 de SB1)**

DESCRIPCION DEL FUNCIONAMIENTO

Se va a generar un código de programa para cada modelo o nicho y cada carro. Si hay 10 carros se crearán 10 códigos de programa para HFE, 10 para BFB y 10 para KFB. Puede darse el caso de que haya varios minomis o nichos de la misma diversidad que utilicen los mismos carros indistintamente. Por ejemplo, suponemos el caso de que en el minomi X1 y X2 se utilicen para el modelo HFE y que existan 5 carros "iguales". Crearemos 5 códigos HFE para el X1 y otros 5 para el X2, con el fin de poder independizar las posiciones de cogida en cada uno de los carros con un programa único para cada uno de los minomis.

Cada uno de los carros de cada modelo tiene grabados un punto de inicio_fila y otro de fin_fila para cada una de las filas. De esta manera no dependeremos tanto del ajuste mecánico del carro y podremos coger las piezas y hacer correcciones para cada carro en concreto.



A través de número de programa recibido sabemos el modelo a despaletizar. Por ejemplo, si HFE en X1 del ejemplo anterior es el modelo 1, asignaremos a la variable MODELO el valor 1.

CARRO. El número de programa nos dice el número de carro introducido en la instalación. Por ejemplo, si el PLC nos envía el código de programa 22, restaremos 22-20 y sabremos que el carro introducido es el 2. En el caso del código 45 será el 5.

El PLC nos enviará la fila de la que debemos despaletizar, a través de un conjunto de 10 bits llamado COD_FILA. Al recibir el código de programa, grabaremos la variable interna FILA=COD_FILA para poder trabajar en todo momento con ella.

Con estas tres variables, MODELO, CARRO y FILA, buscamos los puntos guardados en las matrices de datos XINI_FILA y XFIN_FILA del \$config.dat para ejecutar los puntos de búsqueda independientes de cada uno de los carros y de cada modelo.

Al, comenzar el ciclo de búsqueda del robot, se verifica que los detectores están en reposo están OK y pasamos a iniciar la búsqueda.

La búsqueda se realiza desde una posición a inicial a una final (INI_FILA y FIN_FILA. aprendidas previamente). La posición inicial se guarda en la primera pieza a coger, y la final en la última de cada fila. Estas posición están almacenadas en unas matrices INI_FILA y FIN_FILA dentro del \$config.dat. Se parametrizan a través de las variables MODELO, CARRO y fila. La posición desde la que el robot comienza a buscar las piezas será una posición desplazada respecto de la posición ini_fila lo suficiente para que nuestra célula láser (LEUZE) esté aún a 1.

La búsqueda de cada una de las piezas, la realizamos con 2 interrupciones, una para el láser y otra para los detectores.

¿Qué hacemos con el láser? Una vez detectado, lo que hacemos es poner el robot a una velocidad que nos asegure que el robot parará a tiempo cuando detecte con los detectores inductivos al tocar la pieza sin desplazarla ni deformarla. Y además, el momento en el que se desencadena la interrupción originada por dicho láser, se va a calcular una posición de colisión, que será la posición máxima hasta la que dejaremos que el robot avance con el fin de si se produce un fallo de detección de la pieza por parte de los detectores inductivos, no se produzcan deterioros de piezas o material de la garra. La Posición de colision = posición actual \pm ajuste del láser para ese modelo \pm offset de colisión. (+ O – EN FUNCIÓN DE NUESTRA BASE):

- Posición actual: Posición de interrupción
- Ajuste_laser[modelo]. Variable declarada en \$config.dat y parametrizada por modelo. Será la distancia física a la pieza desde los apoyos de la garra en el momento de la interrupción.
- Offset_colisión[modelo]. La distancia a mayores que queremos que el robot entre a buscar la pieza

Esta posición de colisión la vamos supervisando todo el tiempo que esté el robot en búsqueda en el módulo prj_sps (Se activará la supervisión mientras la variable BUSQUEDA_ACTIVA de tipo BOOL Y declarada en el \$config.dat esté activa), con el fin de testear si llegamos físicamente a esta posición para abortar la búsqueda y evitar colisiones.



La segunda interrupción se desencadena o bien mediante la detección de los detectores inductivos o por alcanzar la posición de colisión (testeada en prj_sps). Una vez que se cumple una de las dos condiciones, pasamos a coger pieza (cuando no hay colisión) o a salir sin pieza (cuando hay colisión). Cuando se llega a la posición de colisión sin detectar, se genera un DEFECTO DE BÚSQUEDA. El robot enviará la señal correspondiente al PLC, saldrá si los detectores de presencia de pieza no están activos. Fuera de la fila donde se encuentre trabajando, enviará una ORDEN (“en posición de reinicio ciclo”) y esperará el EVENTO (“REINICIO DE CICLO”) antes de volver a rebouclage con el fin de evitar colisionar con piezas atrapadas en la garra y volver que el operario coloque las piezas del minomi correctamente.

La primera pieza de una fila siempre se busca una velocidad tal que nos asegure el paro a través de los detectores inductivos sin provocar deterioros en piezas o garras. El PLC nos avisará a través de la señal INICIO_FILA para poder resetear algunas señales, dar el retorno de que hemos visto ese inicio de fila y colocar nuestra posición de colisión originalmente como la posición fin de fila. Por este motivo la búsqueda después de un INICIO_FILA es más lenta.

En los siguientes ciclos, se calcula la posición de colisión cómo la “posición de la última pieza encontrada” \pm “offset pieza [modelo]” que es la distancia entre piezas de ese modelo. El robot irá al 100% hasta llegar a la posición donde encontró la pieza anterior, o (si ocurre antes) hasta detectar con la célula láser Leuze. En este momento calculará de nuevo la posición de colisión con la ecuación mencionada anteriormente: Posición de colisión = posición actual \pm ajuste del láser para ese modelo \pm offset de colisión.

VARIABLES UNICAS DEL MINOMI STANDARD

Usar estas variables SOLO en Minomi Standard, NO mezclar con otros sistemas.

MODELO: Es el nº del modelo: 1 X95, 2 HFE, 3 BFB, 4 KFB. Si tuviéramos una misma diversidad en varios minomis, habría que crear más modelos.

CARRO: Es el nº de carro. Cada modelo puede tener 10 carros

FILA: Es el nº de fila. Cada carro puede tener 10 filas.

Offset_colision[4]-> Es el offset colisión para cada modelo.

Offset_pieza[4]-> Es la distancia entre 2 piezas en el minomi de cada modelo.

Pieza_Encontrada[4]-> Metemos el valor de la última posición encontrada de ese modelo.

colision_pieza=FALSE-> Si hay colisión se pone a true para testear esta señal y enviar defectos al PLC.

xPos_Colision-> Es la posición de colisión calculada durante la búsqueda.

Busqueda_activa=FALSE-> Cuando entramos en búsqueda la ponemos a true. Se testea en prj_sps para mantener la velocidad al valor deseado sin poderla modificar externamente, para verificar si llegamos a la posición de colisión y para informar al PLC con las señales Alarma_fin_fila y fin_fila

FINFILA=FALSE-> Variable interna que nos indica que hemos llegado al fin de fila



REAL PVM=100.0 ; Velocidad de búsqueda MINOMI

REAL VMA=100.0 ; Velocidad anterior a búsqueda MINOMI que se recuperará una vez acabada la búsqueda

BUSCA_OK=TRUE-> Si la búsqueda ha sido positiva se pone a true

xini_fila[4,10,10] y xfin_fila[4,10,10] -> Posición inicial/final de búsqueda [MODELO, CARRO, FILA]

MINOMI STANDARD RENAULT. PALETIZADO

DETECCIONES EN GARRA

La garra está equipada con los detectores de presencia pieza necesarios, al igual que cualquier garra de manipulación. No necesitamos detectores específicos, tipo laser ya que no se buscan piezas.

INTERCAMBIO DE SEÑALES

SEÑALES DEL AUTÓMATA:

ENTRADAS AL ROBOT

Con el código de programa ya sabemos el modelo y el carro.

SIGNAL COD_FILA \$IN[225] TO \$IN[234] ; 10 BITS. Nos indica el Numero de fila + posición (palabra 5)

SIGNAL INICIO_FILA \$IN[240]; Nos indica que es un inicio de fila (palabra 5)

SALIDAS DEL ROBOT

SIGNAL COD_PROCESS \$OUT[225] TO \$OUT[232] ; Rango para el código de proceso (palabra 5)

SIGNAL ACK_INI_FILA \$OUT[237] TO \$OUT[237] ; Enterado inicio fila (palabra 5)

SIGNAL Alarma_Fin_Fila_X \$OUT[239] TO \$OUT[239] ; Activar alarma fin de fila (palabra 5)

SIGNAL Fin_Fila_X \$OUT[240] TO \$OUT[240] ; Activar fin de fila (palabra 5)

DESCRIPCION DEL FUNCIONAMIENTO

Se va a generar un código de programa para cada modelo (diversidad) y cada carro. Si hay 10 carros se crearán 10 códigos de programa para HFE, 10 para BFB y 10 para KFB. Estos códigos de programa nos valen para el paletizado en todos los minomis (nichos), ya que la decisión de paletizar en uno u otro se realiza a través de un evento (aguillage)

El número de carro (Variable CARRO) vendrá determinado por el código de programa recibido, este código se construye con el número de carro + la diversidad de la pieza a contenerizar o paletizar.

En función del código de programa escribimos el valor de la variable MODELO (Un valor para cada diversidad de cada nicho o minomi)



En función del evento recibido en el aguijaje, escribimos el valor de la variable NICHOS. Un valor para cada uno de los nichos o minomis.

El cálculo de la fila se realiza a través de la subrutina “Calculo_Posicion_y_Fila” que está dentro del módulo “Minomi”. A ella se le pasa el valor de COD_FILA. Este valor es el resultado en decimal de la combinación de bits que nos envía el PLC. El PLC nos envía la fila y la posición en la cual debemos paletizar. Por ejemplo, para una pieza a dejar en la fila 1 y posición 3, el autómatas nos escribirá en COD_FILA el valor 103. El robot, a través de la subrutina “Calculo_Posicion_y_Fila” separará el valor de la fila y el de la posición para utilizarlos posteriormente:

- La fila nos sirve para buscar en la matriz de posiciones el ini_fila que corresponda a ese modelo y ese carro y esa fila.
- Con el cálculo de la posición actualizamos el valor de la variable CONTADOR[NICHOS] de manera que podemos parametrizar la posición de dejada realizando una multiplicación del espacio entre piezas (offset_pieza[modelo]) por el valor de dicho contador

Por último, se devuelve un valor al PLC a través del grupo de salidas denominado COD_PROCESS. Este valor será el valor de la variable CONTADOR[nichos]. El PLC necesita este dato para realizar una comprobación de que vamos a dejar la pieza en la posición correcta y validar el retroceso de dejada del robot.

Este tipo de paletizado, tiene las particularidades siguientes:

- En cada carro se pueden paletizar n piezas de n diversidades distintas siempre y cuando la mecánica lo permita.
- Es el PLC quien indica al robot la posición en la cual debe de dejar cada pieza:
 - En el caso de paletizados al Film, de esta forma se pueden generar huecos si así lo demanda el sistema superior (SIPTOL). Se escribirán los datos de cada una de las piezas en una tarjeta balogh (Interbús) o RFID (profinet)
 - En el caso de paletizados que no sean al Film, el funcionamiento será el mismo con la salvedad de que no tenemos que paletizar bajo la demanda de SIPTOL y tampoco escribir los datos de las piezas en las tarjetas balogh ni RFID.

VARIABLES UNICAS DEL MINOMI STANDARD. Paletizado

Para la función de paletizado, tendremos declaradas las siguientes variables en el \$config.dat:

Offset_pieza[10]; Array de 10 posiciones, que almacena para cada modelo ([1], [2], ... [10]) el espaciado entre piezas dentro del carro (minomi)

CONTADOR [5]; Array de 5 posiciones en las que se almacena el contador de cada uno de los minomis (huecos o nichos)

Variables de tipo entero:

- MODELO: Nos indicará el modelo a paletizar
- CARRO: Número de carro en el que paletizamos
- FILA: Número de fila en la que dejamos la pieza



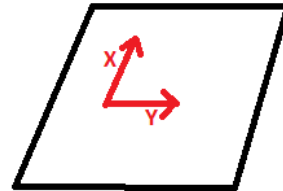
- NICHOS: Minomi, hueco.

POSICIONES DE INICIO FILA Y FIN DE FILA:

- `xini_fila[10,15,3]` ; Matriz de 3 posiciones en las que se almacena la posición de inicio de fila. Esta posición corresponde con la primera posición que el robot paletiza en cada fila. Cada posición de la matriz, corresponde por este orden al modelo, el carro y la fila.
- `xfin_fila[10,15,3]` ; Matriz de 3 posiciones en las que se almacena la posición de fin de fila. Esta posición corresponde con la última posición que el robot paletiza de cada una de las filas. Cada posición de la matriz, corresponde por este orden al modelo, el carro y la fila.
- Estas posiciones de inicio fila y fin de fila se grabarán a través del módulo “grabar_pos”. Tendremos que introducir el valor de las variables MODELO, CARRO y FILA. Dentro del módulo tenemos información del modelo para introducir el valor. El carro será el número de carro que queramos grabar, y la fila será la fila a grabar.

VARIABLES PARA APLICAR OFFSET PARA CORREGIR DESVIACIONES DE LAS BASES.

```
DECL REAL offset_YP91=0.0 ; Offset Y minomi P91
DECL REAL offset_ZP91=0.0 ; Offset Z minomi P91
DECL REAL offset_YP92=0.0 ; Offset Y minomi P92
DECL REAL offset_ZP92=0.0 ; Offset Z minomi P92
```



Como se ha explicado en el apartado anterior, el paletizado de cada fila se realiza a través de posiciones parametrizadas en función de la posición “`ini_fila[modelo,carro,fila]`”. Se dejará la primera pieza y las siguientes tendrán la misma posición que la primera desplazadas el inter espacio (`offset_pieza[modelo]`) que corresponda.

Para los robots de este ejemplo tomado del contenerizado de pieles en las instalaciones DCD3 y DCG3, la base se ha grabado como se muestra en la figura anterior. La X es el avance en el carro.

En el caso de que al avanzar en las posiciones de las dejadas, tengamos derivas en las coordenadas perpendiculares a la de avance en la fila (es decir, en el ejemplo “Y” o “Z”) utilizaremos las variables `offset_Y`, `offset_Z` de siguiente modo:

- `Offset_Y` será el resultado de la resta de `ini_fila.y – fin_fila.y` dividido por el número de piezas que caben en una fila, para calcular el avance de la Y para cada pieza.
- `Offset_Z` será el resultado de la resta de `ini_fila.y – fin_fila.y` dividido por el número de piezas que caben en una fila, para calcular el avance de la Z para cada pieza.