

Ayudantía N°3

1s/2019

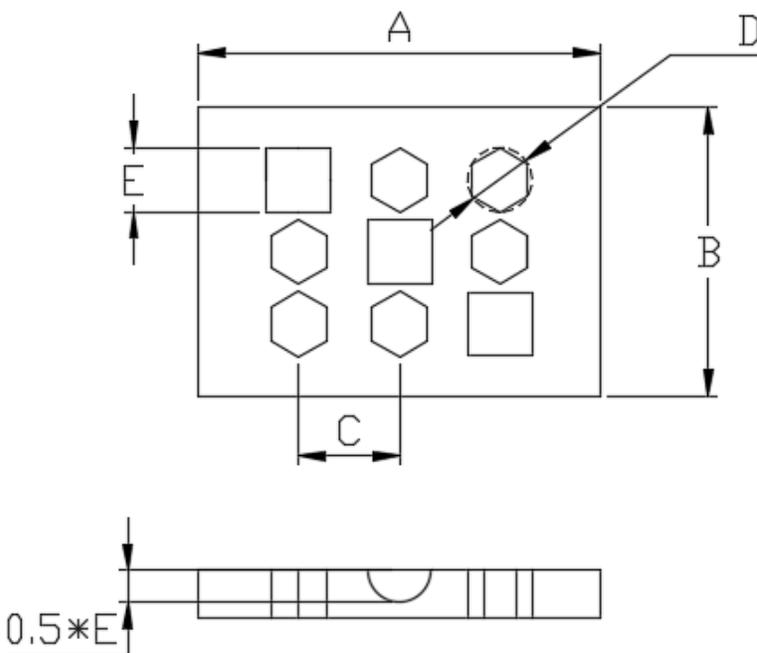
Problema: Se tiene una pieza rectangular que contiene una matriz dentro de ella. Esta matriz tiene la característica de que es cuadrada y que su diagonal es distinta de los demás componentes. Los elementos de la diagonal serán semicilindros, mientras que los bloques no pertenecientes a ésta son hexágonos.

Por otro lado se tiene la particularidad de que la pieza posee dimensiones variables, esto quiere decir que las longitudes a trabajar las da el usuario, por lo cual se pide crear un programa en Código parametrizado para poder variar las dimensiones, utilizando el comando G65.

Finalmente se pide que se verifique si el problema es realmente posible de realizarse, ya que las dimensiones las da un usuario el cual puede cometer errores.

En la siguiente imagen se adjunta el plano de la pieza.

Figura 1:



Donde los comandos a ingresar en el G65 deberán ser de la siguiente manera:

G65 A__ . B__ . C__ . D__ . E__ . F__ . ;

Donde F es el valor de la dimension de la matriz a utilizar.

Solución:

(Programa principal)

O2000;

S2000 M03;

F50;

G00 X0. Y0. Z100.;

G65 P2001 A250. B180. C62.5 D38. E40. F3.;

M05;

M30;

(Sub programa G65)

O2001;

#101=#1; (A)

#102=#2; (B)

#103=#3; (C)

#104=#7; (D)

#105=#8; (E)

#106=#9; (F)

(Condiciones que estudian factibilidad de la pieza)

IF [#103*#106-1.]+#105/2.+#104/2.] GT #101] GOTO N100; (ver que las perforaciones sean menores que el lado A)

IF [#103*#106-1.]+#105/2.+#104/2.] GT #102] GOTO N100; (ver que las perforaciones sean menores que el lado B)

IF [#103 LT #104] GOTO N100; (Ver la distancia entre hexágonos sea mas grande que la distancia C)

IF [#103 LT [#104/2.+#105/2.]] GOTO N100; (Ver la distancia entre hexagono y semicilindro sea mas grande que la distancia C)

(Empezar con la matriz)

G01 X0. Y0.;

X[-#103*#106/2.-0.5] Y[#103*#106/2.-0.5]]; (Centro de la figura superior izquierda)

Z0.;

(Diagonal)

#110=1.:(contador)

G91;

WHILE [#110 LE #106] DO 1; (Recorre la diagonal)

#111=1.; (Contador)

G01 X[#105/2.] Y[#105/2.];

WHILE [#111 LE #105] DO 2; (Realiza los semicilindros)

G18; (Cambio de plano a XZ)

```
G03 X[-#105] I[-#105/2.];  
G01;  
#111=[#111+1.];  
G17; (Vuelvo al plano XZ)  
Y[-1.];  
X[#105];  
END 2;  
#110=[#110+1.];  
G01 X[-#105/2.] Y[#105/2.];  
Z10.;  
X[#103] Y[-#103];  
Z-10.;  
END 1;  
G90;  
Z10.;  
X0. Y0.;  
X[-#103*[#106/2.-0.5]] Y[#103*[#106/2.-0.5]]; (Posicion inicial)  
  
(Resto de la matriz)  
  
#112=1.; (Contador en Y)  
G91; (Volvemos a ocupar coordenadas relativas)  
WHILE [#112 LE #106] DO 1; (While que recorre eje Y)
```

```
G01;
#113=1.; (Contador en X)
WHILE [#113 LE #106] DO 2; (While que recorre el eje X)
#115=[360./6.] (Desfase angular)
#116=[210.]; (Contador angular)
G01 Y[[#104/2.]*SIN[90.]]; (Posicion inicial de las poligonos)
Z-10.;(Bajamos en Z cero, desde Z10, no olvidar re relativas)
WHILE [#116 LE [510.]] DO 3; (Ciclo para hacer el hexágono)
IF [#112 EQ #113] GOTO N10;
G01 X[[#104/2.]*COS[#116]] Y[[#104/2.]*SIN[#116]];
N10 #116=[#116+#115];(Variamos el ángulo)
END 3;
Y[-#104/2.]*SIN[90.]]; (Volver al centro)
Z10.;
#113=[#113+1.];
X[#103];
END 2;
#112=[#112+1.];
Y[-#103];
X[-#103*#106]];
END 1;
X0. Y0.;
```

(Contorno)

G90.

X[#101/2.] Y[#102/2.];

X[-#101/2.];

Y[-#102/2.];

X[#101/2.];

Y[#102/2.];

Z30.;

X0. Y0.;

N100 M99;