

```

1  """
2  Embedded Python Blocks:
3
4  Each time this file is saved, GRC will instantiate the first class it finds
5  to get ports and parameters of your block. The arguments to __init__ will
6  be the parameters. All of them are required to have default values!
7  """
8  # SE IMPORTAN LAS LIBRERIAS PARA LA REALIZACION DEL PROCESO.
9  import numpy as np
10 from gnuradio import gr
11
12
13 class blk(gr.sync_block): # other base classes are basic_block, decim_block, interp_block
14     """Embedded Python Block example - a simple multiply const"""
15
16     def __init__(self, example_param=1.0): # only default arguments here
17         """arguments to this function show up as parameters in GRC"""
18         gr.sync_block.__init__(
19             self,
20             # EN ESTE APARTADO SE TIENEN LOS PARAMETROS DE LA CAJA PARA TRABAJAR
21             name='Embedded Python Block', # EN ESTA LINEA SE LA ASIGNA NOMBRE A LA CAJA
22             in_sig=[np.complex64], # EN ESTA LINEA SE OBTIENE LA SEÑAL DE ENTRADA
23             out_sig=[np.complex64] # EN ESTA LINEA SE OBTIENE LA SEÑAL DE SALIDA
24         )
25         # if an attribute with the same name as a parameter is found,
26         # a callback is registered (properties work, too).
27         self.example_param = example_param # EN ESTA LINEA SE OBTIENE EL PARAMETRO DE GANANCIA
28         # O CONSTANTE QUE SE MULTIPLICA POR LA SEÑAL DE ENTRADA
29
30     def work(self, input_items, output_items):
31         """AQUI SE REALIZA LA ACCION LA CAUL ES MULTIPLICAR LA SEÑAL DE ENTRADA POR LA CONSTANTE DADA POR LA CAJA"""
32         output_items[0][:] = input_items[0] * self.example_param
33         # SE RETORNA LA SEÑAL DE SALIDA.
34         return len(output_items[0])
35

```