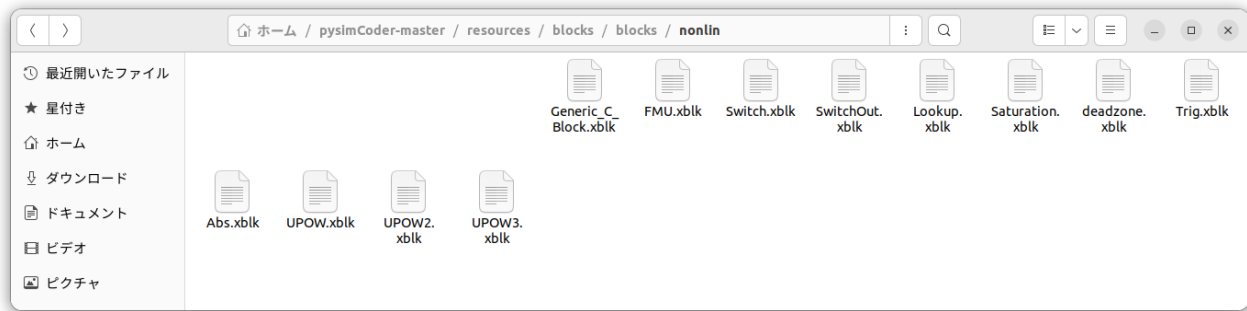


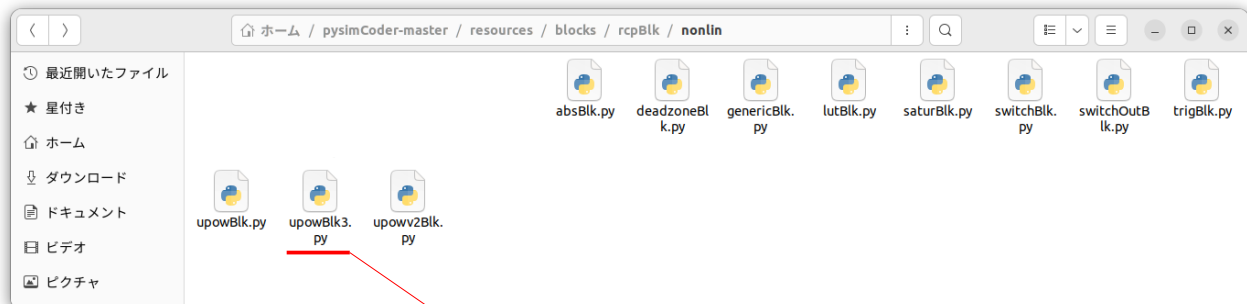
### 1.1 Save **UPOW3.xblk** in the following folder.



```
開く(O)  UPOW3.xblk
~/pysimCoder-master/resources/blocks/blocks/nonlin

1 {
2   "lib": "nonlin",
3   "name": "UPOWV3",
4   "ip": 1,
5   "op": 1,
6   "stin": 0,
7   "stout": 0,
8   "icon": "UV3",
9   "params": "upowBlk3",
10  "help": ""
11 }
12 ,
```

### 1.2 Save **upowBlk3.py** in the following folder.

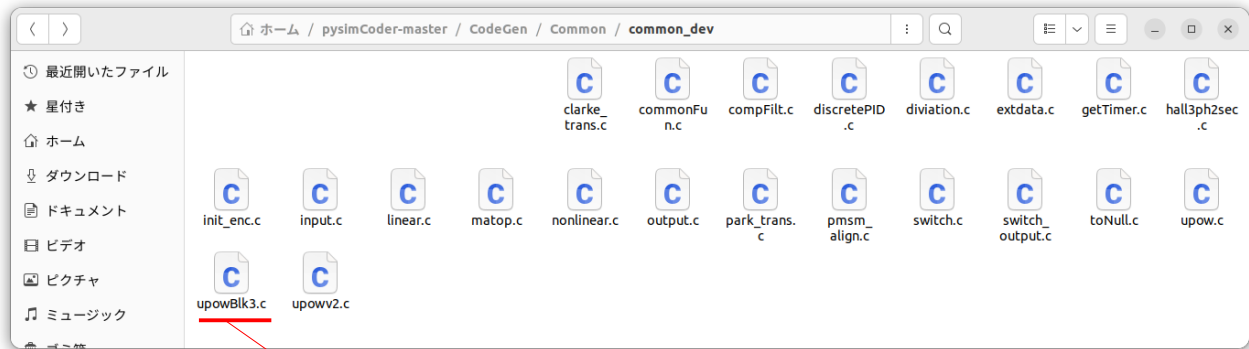


Add **import math etc.** to upowBlk3.py.

```
開く(O)  upowBlk3.py
~/pysimCoder-master/resources/blocks/rcpBlk/nonlin

1
2 from supsisin.RCPblk import RCPblk
3 from scipy import size
4
5 import math
6
7 def upowBlk3(pin, pout):
8     """
9
10    Call:  upowBlk3(pout)
11
12    Parameters
13    -----
14    pout: connected output port(s)
15
16    Returns
17    -----
18    blk: RCPblk
19
20    """
21
22    result = math.sqrt(25)
23    print(result)
24
25    blk = RCPblk('upowBlk3', pin, pout, [0,0], 0, [], [])
26    return blk
```

### 1.3 Save `upowBlk3.c` in the following folder.

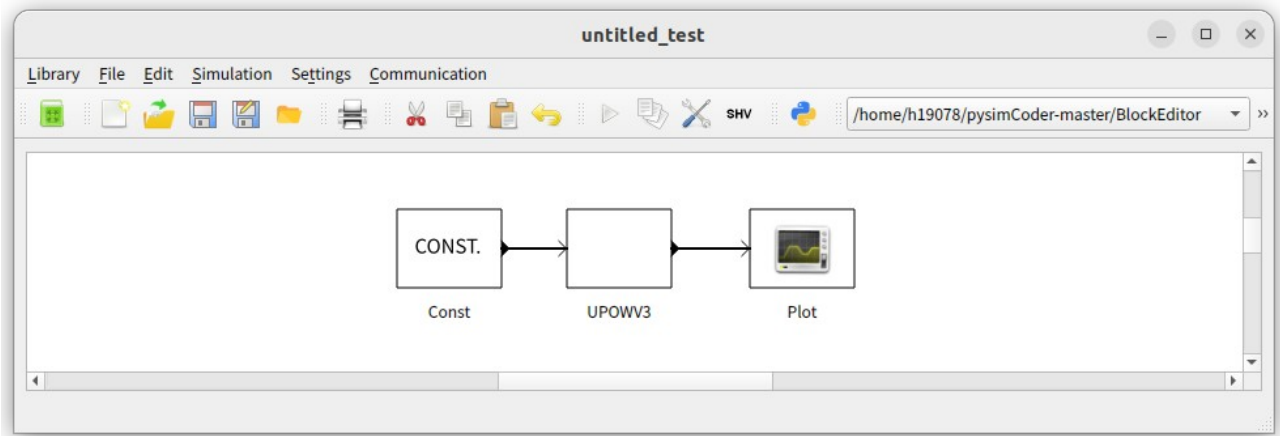


```
開く(O)  upowBlk3.c  保存(S)
~/pysimCoder-master/CodeGen/Common/common_dev

19 #include <pyblock.h>
20
21 static void init(python_block *block)
22 {
23     /* double * realPar = block->realPar; */
24     /* int * intPar = block->intPar; */
25     /* double *y = block->y[0]; */
26     /* double *u = block->u[0]; */
27
28 }
29
30 static void inout(python_block *block)
31 {
32     /* double * realPar = block->realPar; */
33     /* int * intPar = block->intPar; */
34     double *u = block->u[0];
35     double *y = block->y[0];
36     y[0] = u[0];
37 }
38
39
40 static void update(python_block *block)
41 {
42     /* double * realPar = block->realPar; */
43     /* int * intPar = block->intPar; */
44     /* double *y = block->y[0]; */
45     /* double *u = block->u[0]; */
46
47 }
48
49 static void end(python_block *block)
50 {
51     /* double * realPar = block->realPar; */
52     /* int * intPar = block->intPar; */
53     /* double *y = block->y[0]; */
54     /* double *u = block->u[0]; */
55 }
56
57
58 void upowBlk3(int flag, python_block *block)
59 {
60     if (flag==CG_OUT){          /* get input */
61         inout(block);
62     }
63     else if (flag == CG_STUPD){
64         update(block);
65     }
66     else if (flag==CG_END){    /* termination */
67         end(block);
68     }
69     else if (flag ==CG_INIT){  /* initialisation */
70         init(block);
71     }
72 }
73
```

C v タブ幅: 8 v (31行、2列) [挿入]

2.1 Run untitled\_test.dgm as shown below.



And then, 5 of 1.2's "math.sqrt(25)" is output as shown below.

```
h19078@h19078: ~/pysimCoder-master/BlockEditor
h19078@h19078:~/pysimCoder-master/BlockEditor$ python3 pysimCoder.py
Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=
wayland to run on Wayland anyway.
5.0
cp /home/h19078/pysimCoder-master/CodeGen/src/linux_main.c .
gcc -g -O2 -I/home/h19078/pysimCoder-master/CodeGen/tos1a/includes -I/hom
e/h19078/pysimCoder-master/CodeGen/arduinoFirmata/includes -I/home/h19078
/pysimCoder-master/CodeGen/LinuxRT/include -I/home/h19078/pysimCoder-mast
er/CodeGen/Common/include -DMODEL=untitled_test -c -o linux_main.o lin
ux_main.c
gcc -g -O2 -I/home/h19078/pysimCoder-master/CodeGen/tos1a/includes -I/hom
e/h19078/pysimCoder-master/CodeGen/arduinoFirmata/includes -I/home/h19078
/pysimCoder-master/CodeGen/LinuxRT/include -I/home/h19078/pysimCoder-mast
er/CodeGen/Common/include -DMODEL=untitled_test -c -o untitled_test.o
untitled_test.c
gcc -static -o ../untitled_test linux_main.o untitled_test.o /home/h190
78/pysimCoder-master/CodeGen/LinuxRT/lib/libpyblk.a /home/h19078/pysimCo
der-master/CodeGen/arduinoFirmata/lib/firmatapyblk.a -lrt -lpthread -lm
### Created executable: untitled_test
QCoreApplication::exec: The event loop is already running
```

3.1 Let me ask you a question here. Is it possible to output "math.sqrt(25)" to "return blk"? This is because I want to create a block only with the Python program of upowBlk3.py without modifying the C program of upowBlk3.c.

```
upowBlk3.py
~/pysimCoder-master/resources/blocks/rcpBlk/nonlin

1
2 from supsisin.RCPblk import RCPblk
3 from scipy import size
4
5 import math
6
7 def upowBlk3(pin, pout):
8     """
9
10    Call:   upowBlk3(pout)
11
12    Parameters
13    -----
14    pout:  connected output port(s)
15
16    Returns
17    -----
18    blk:  RCPblk
19
20    """
21
22    result = math.sqrt(25)
23    print(result)
24
25    blk = RCPblk('upowBlk3', pin, pout, [0,0], 0, [], [])
26    return blk
```

