Input and Output description

Where can I find useful information about Quantum ESPRESSO ?

Where can I find useful information about Quantum ESPRESSO ?

```
prompt > cd  $espresso_dir/Doc; ls *.html
```

```
INPUT_DOS.html          INPUT_PP.html       INPUT_PW.html
INPUT_BANDS.html        INPUT_GIPAW.html    INPUT_PROJWFC.html
INPUT_CPPP.html         INPUT_LD1.html      INPUT_PWCOND.html
INPUT_D3.html           INPUT_PH.html       INPUT_pw_export.html
```

In particular INPUT_PW.html contains a rather complete description of the input of PWscf.

Similarly INPUT_PP.html, INPUT_PH.html,... contain descriptions of post processing, phonon...

We will examine to some extent the input of PWscf

The input file for PWscf is structured in a number of NAMELISTS and INPUT_CARDS.

```
&NAMELIST1 ... /

&NAMELIST2 ... /

&NAMELIST3 ... /

INPUT_CARD1
  ....
  ....
INPUT_CARD2
  ....
  ....
```

NAMELISTS are a standard input construct in fortran90.

The use of NAMELISTS allows to specify the value of an input variable only when it is needed and to define default values for most variables that then need not be specified. Variable can be inserted in any order.

```
&NAMELIST
   needed_variable2=XX, needed_variable1=X,
   character_variable1='a suitable string'
 /
```

NAMELISTS are read in a specific order

NAMELISTS that are not required are ignored

INPUT_CARDS are specific of QuantumESPRESSO codes and are used to provide input data that are always needed and would be boring to specify with the `variable_name=variable_value` syntax used by NAMELIST.

INPUT_CARDS require data in specific order (which may depend on the situation and on the value of a card_format_specifier)

For instance:

```
INPUT_CARD   card_format_specifier
 data(1,1) data(1,2) data(1,3) ...
 data(2,1) data(2,2) data(2,3) ...
 data(3,1) data(3,2) data(3,3) ...
 ... ... ...
```

Logically independent INPUT_CARDS can be given in any order

There are three mandatory NAMELISTS in PWscf:

There are three mandatory NAMELISTS in PWscf:

```
&CONTROL    input variables that control the flux of the
            calculation and the amount of I/O on disk and
            on the screen.
```

There are three mandatory NAMELISTS in PWscf:

`&CONTROL`	`input variables that control the flux of the calculation and the amount of I/O on disk and on the screen.`
`&SYSTEM`	`input variables that specify the system under study.`

There are three mandatory NAMELISTS in PWscf:

`&CONTROL`	input variables that control the flux of the calculation and the amount of I/O on disk and on the screen.
`&SYSTEM`	input variables that specify the system under study.
`&ELECTRONS`	input variables that control the algorithms used to reach the self-consistent solution of KS equations for the electrons.

There are three additional NAMELISTS in PWscf that must be specified under certain circumstances:

There are three additional NAMELISTS in PWscf that must be specified under certain circumstances:

```
&IONS      needed when ATOMS MOVE! IGNORED otherwise !
           input variables that control ionic motion in
           molecular dynamics run or structural relaxation
```

There are three additional NAMELISTS in PWscf that must be specified under certain circumstances:

```
&IONS      needed when ATOMS MOVE! IGNORED otherwise !
           input variables that control ionic motion in
           molecular dynamics run or structural relaxation

&CELL      needed when CELL MOVES! IGNORED otherwise !
           input variables that control the cell-shape
           evolution in a variable-cell-shape MD or
           structural relaxation
```

There are three additional NAMELISTS in PWscf that must be specified under certain circumstances:

```
&IONS      needed when ATOMS MOVE! IGNORED otherwise !
           input variables that control ionic motion in
           molecular dynamics run or structural relaxation

&CELL      needed when CELL MOVES! IGNORED otherwise !
           input variables that control the cell-shape
           evolution in a variable-cell-shape MD or
           structural relaxation

&EE        needed when density counter charge corrections
           are used to solve the problem with open boundary
           conditions
```

There are three mandatory INPUT_CARDS in PWscf

There are three mandatory INPUT_CARDS in PWscf

```
ATOMIC_SPECIES    name, mass and pseudopotential used for
                  each atomic species present in the system
```

There are three mandatory INPUT_CARDS in PWscf

```
ATOMIC_SPECIES    name, mass and pseudopotential used for
                  each atomic species present in the system

ATOMIC_POSITIONS  type and coordinates of each atom in the
                  unit cell
```

There are three mandatory INPUT_CARDS in PWscf

`ATOMIC_SPECIES`	`name, mass and pseudopotential used for each atomic species present in the system`
`ATOMIC_POSITIONS`	`type and coordinates of each atom in the unit cell`
`K_POINTS`	`coordinates and weights of the k-points used for BZ integration`

There are additional INPUT_CARDS in PWscf that must be specified only under certain circumstances.

They are

There are additional INPUT_CARDS in PWscf that must be specified only under certain circumstances.

They are

CELL_PARAMETERS

There are additional INPUT_CARDS in PWscf that must be specified only under certain circumstances.

They are

`CELL_PARAMETERS`

`OCCUPATIONS`

There are additional INPUT_CARDS in PWscf that must be specified only under certain circumstances.

They are

```
CELL_PARAMETERS

OCCUPATIONS

CLIMBING_IMAGES    (only for NEB calculations)
CONSTRAINTS        (only for constrained dynamics)
COLLECTIVE_VARS    (only for metadynamics)
```

The &CONTROL namelist

&CONTROL input variables that control the flux of the calculation and the amount of I/O on disk and on the screen.

FLUX : calculation

I/O : title, verbosity, iprint, outdir, prefix, pseudo_dir, tprnfor, tstress, disk_io, wf_collect

RESTART : restart_mode, max_seconds

MISC : dt, nstep, etot_conv_thr, forc_conv_thr, tefield, dipfield, lelfield, lberry

The &CONTROL namelist (FLUX)

```
calculation CHARACTER  (default = 'scf')
            a string describing the task to be performed:
            'scf', 'bands', 'nscf', 'relax', 'md',
            'vc-relax', 'vc-md', 'neb'
            (vc=variable-cell; 'phonon' is no longer used)
```

Input structure for a SCF run

```
&CONTROL ... /
&SYSTEM ... /
&ELECTRONS ... /
ATOMIC_SPECIES
ATOMIC_POSITIONS
K_POINTS
```

```
&CONTROL ... /
&SYSTEM ibrav=0 ... /
&ELECTRONS ... /
CELL_PARAMETERS
ATOMIC_SPECIES
ATOMIC_POSITIONS
K_POINTS
```

```
&CONTROL ... /
&SYSTEM ... /
&ELECTRONS occupations='fixed' ... /
OCCUPATIONS
ATOMIC_SPECIES
ATOMIC_POSITIONS
K_POINTS
```

Input structure for a RELAX / MD run

```
&CONTROL calculation='relax' ... /
&SYSTEM ... /
&ELECTRONS ... /
&IONS ... /
ATOMIC_SPECIES
ATOMIC_POSITIONS
K_POINTS

&CONTROL calculation='vc-relax' ... /
&SYSTEM ... /
&ELECTRONS ... /
&IONS ... /
&CELL ... /
ATOMIC_SPECIES
ATOMIC_POSITIONS
K_POINTS
```

An example

```
&control
    pseudo_dir = './',
    outdir='./tmp/',
    prefix='be0001'
    tprnfor = .true.
 /
 &system
    ibrav=4, celldm(1)=4.247, celldm(3)=16.0, nat=12, ntyp=1, nbnd=20,
    occupations='smearing', smearing='marzari-vanderbilt', degauss=0.05
    ecutwfc=22.0
 /
 &electrons
 /
ATOMIC_SPECIES
Be   1.0   Be.vbc2
ATOMIC_POSITIONS alat
Be   0.000000000  -0.288675135   4.359667099
Be   0.000000000   0.288675135   3.548485449
Be   0.000000000  -0.288675135   2.754655986
Be   0.000000000   0.288675135   1.965554700
Be   0.000000000   0.288675135   1.965554700
```

```
Be   0.000000000  -0.288675135   1.178901500
Be   0.000000000   0.288675135   0.392919700
Be   0.000000000  -0.288675135  -0.392919700
Be   0.000000000   0.288675135  -1.178901500
Be   0.000000000  -0.288675135  -1.965554700
Be   0.000000000   0.288675135  -2.754655986
Be   0.000000000  -0.288675135  -3.548485449
Be   0.000000000   0.288675135  -4.359667099
K_POINTS automatic
  15 15 1 0 0 0
```

Start code as (for instance):

```
prompt> $espresso_dir/bin/pw.x < pw.in > pw.out
```

Alternative syntax (useful on some parallel machines):

```
prompt> $espresso_dir/bin/pw.x -inp pw.in > pw.out
```

The output

```
Program PWSCF     v.4.1    starts on 20Sep2009 at 16:19:46

This program is part of the open-source Quantum ESPRESSO suite
for quantum simulation of materials; please acknowledge
    "P. Giannozzi et al., J. Phys.:Condens. Matter 21 395502 (2009);
     URL http://www.quantum-espresso.org",
in publications or presentations arising from this work. More details:
http://www.quantum-espresso.org/wiki/index.php/Citing_Quantum-ESPRESSO

Current dimensions of program PWSCF are:
Max number of different atomic species (ntypx) = 10
Max number of k-points (npk) =  40000
Max angular momentum in pseudopotentials (lmaxx) =  3
Waiting for input...
```

```
bravais-lattice index     =            4
lattice parameter (a_0)   =       4.2470  a.u.
unit-cell volume          =    1061.4448 (a.u.)^3
number of atoms/cell      =           12
number of atomic types    =            1
number of electrons       =        24.00
number of Kohn-Sham states=           20
kinetic-energy cutoff     =      22.0000  Ry
charge density cutoff     =      88.0000  Ry
convergence threshold     =      1.0E-06
g beta                 =       0.7000
number of iterations used =            8  plain     mixing
Exchange-correlation      = PZ (1100)

celldm(1)=   4.247000  celldm(2)=   0.000000  celldm(3)=  16.000000
celldm(4)=   0.000000  celldm(5)=   0.000000  celldm(6)=   0.000000

crystal axes: (cart. coord. in units of a_0)
          a(1) = (  1.000000  0.000000  0.000000 )
          a(2) = ( -0.500000  0.866025  0.000000 )
          a(3) = (  0.000000  0.000000 16.000000 )
```

```
reciprocal axes: (cart. coord. in units 2 pi/a_0)
          b(1) = (  1.000000  0.577350 -0.000000 )
          b(2) = (  0.000000  1.154701  0.000000 )
          b(3) = (  0.000000 -0.000000  0.062500 )


PseudoPot. # 1 for Be read from file Be.vbc2
Pseudo is Norm-conserving + core correction, Zval =  2.0
From published tables, or generated by old code (analytical format)
Using radial grid of  153 points,  1 beta functions with:
           l(1) =   0

atomic species   valence    mass     pseudopotential
   Be              2.00     1.00000     Be( 1.00)

12 Sym.Ops. (with inversion)
```

Cartesian axes

site n.	atom		positions (a_0 units)		
1	Be	tau(1) = (	0.0000000	-0.2886751	4.3596671)
2	Be	tau(2) = (	0.0000000	0.2886751	3.5484854)
3	Be	tau(3) = (	0.0000000	-0.2886751	2.7546560)
4	Be	tau(4) = (	0.0000000	0.2886751	1.9655547)
5	Be	tau(5) = (	0.0000000	-0.2886751	1.1789015)
6	Be	tau(6) = (	0.0000000	0.2886751	0.3929197)
7	Be	tau(7) = (	0.0000000	-0.2886751	-0.3929197)
8	Be	tau(8) = (	0.0000000	0.2886751	-1.1789015)
9	Be	tau(9) = (	0.0000000	-0.2886751	-1.9655547)
10	Be	tau(10) = (	0.0000000	0.2886751	-2.7546560)
11	Be	tau(11) = (	0.0000000	-0.2886751	-3.5484854)
12	Be	tau(12) = (	0.0000000	0.2886751	-4.3596671)

```
number of k points=   27  gaussian broad. (Ry)=  0.0500     ngauss =
                  cart. coord. in units 2pi/a_0
   k(    1) = (   0.0000000   0.0000000   0.0000000), wk =   0.0088889
   k(    2) = (   0.0000000   0.0769800   0.0000000), wk =   0.0533333
   k(    3) = (   0.0000000   0.1539601   0.0000000), wk =   0.0533333
   k(    4) = (   0.0000000   0.2309401   0.0000000), wk =   0.0533333
   k(    5) = (   0.0000000   0.3079201   0.0000000), wk =   0.0533333
   k(    6) = (   0.0000000   0.3849002   0.0000000), wk =   0.0533333
   k(    7) = (   0.0000000   0.4618802   0.0000000), wk =   0.0533333
   k(    8) = (   0.0000000   0.5388603   0.0000000), wk =   0.0533333
   k(    9) = (   0.0666667   0.1154701   0.0000000), wk =   0.0533333
   k(   10) = (   0.0666667   0.1924501   0.0000000), wk =   0.1066667
   k(   11) = (   0.0666667   0.2694301   0.0000000), wk =   0.1066667
   k(   12) = (   0.0666667   0.3464102   0.0000000), wk =   0.1066667
   k(   13) = (   0.0666667   0.4233902   0.0000000), wk =   0.1066667
   k(   14) = (   0.0666667   0.5003702   0.0000000), wk =   0.1066667
   k(   15) = (   0.0666667   0.5773503   0.0000000), wk =   0.0533333
   k(   16) = (   0.1333333   0.2309401   0.0000000), wk =   0.0533333
   k(   17) = (   0.1333333   0.3079201   0.0000000), wk =   0.1066667
   k(   18) = (   0.1333333   0.3849002   0.0000000), wk =   0.1066667
        ... .... ... ....
```

```
G cutoff =   40.2057  (  14795 G-vectors)     FFT grid: ( 16, 16,216)

Largest allocated arrays     est. size (Mb)     dimensions
   Kohn-Sham Wavefunctions         0.58 Mb     (   1899,  20)
   NL pseudopotentials             0.35 Mb     (   1899,  12)
   Each V/rho on FFT grid          0.84 Mb     (  55296)
   Each G-vector array             0.11 Mb     (  14795)
   G-vector shells                 0.01 Mb     (    943)
Largest temporary arrays     est. size (Mb)     dimensions
   Auxiliary wavefunctions         2.32 Mb     (   1899,  80)
   Each subspace H/S matrix        0.10 Mb     (     80,  80)
   Each <psi_i|beta_j> matrix      0.00 Mb     (     12,  20)
   Arrays for rho mixing           6.75 Mb     (  55296,   8)

Check: negative/imaginary core charge=   -0.000003    0.000000

Initial potential from superposition of free atoms
Check: negative starting charge=   -0.001695

starting charge   23.99904, renormalised to   24.00000
```

```
negative rho (up, down):  0.169E-02 0.000E+00
Starting wfc are   12 atomic +    8 random wfc

total cpu time spent up to now is      4.56 secs

Self-consistent Calculation

iteration #  1     ecut=    22.00 Ry     beta=0.70
Davidson diagonalization with overlap
ethr =  1.00E-02,  avg # of iterations =  7.9

negative rho (up, down):  0.465E-03 0.000E+00

total cpu time spent up to now is     36.93 secs

total energy              =     -29.25526792 Ry
Harris-Foulkes estimate   =     -29.58353697 Ry
estimated scf accuracy    <       0.39433819 Ry

... ... ... ...
```

```
iteration # 14     ecut=    22.00 Ry     beta=0.70
Davidson diagonalization with overlap
ethr =  6.75E-09,  avg # of iterations =  3.0

total cpu time spent up to now is    243.76 secs

End of self-consistent calculation

     k = 0.0000 0.0000 0.0000 (  1883 PWs)   bands (ev):

8.7542  -8.4238  -8.0330  -7.5817  -7.0563  -6.4469  -5.7471  -4.9601
4.1001  -3.2132  -2.4661  -0.2226  -0.1978   4.3114   5.4068   6.5157
7.1528   7.7886   7.8789   9.1487

     .... .... .... ....

     k = 0.3333 0.5774 0.0000 (  1899 PWs)   bands (ev):

0.0424   0.4457   0.9310   1.2241   1.2241   1.3392   1.3392   1.4432
1.4432   1.4989   1.6419   1.6419   2.0181   2.0181   2.1641   2.9349
3.3634   3.3634   3.8134   4.7957
```

```
the Fermi energy is     2.4382 ev

!    total energy              =     -29.53449845 Ry
Harris-Foulkes estimate        =     -29.53449871 Ry
estimated scf accuracy         <       0.00000030 Ry

The total energy is the sum of the following terms:

one-electron contribution =    -847.54068683 Ry
hartree contribution      =     431.26799021 Ry
xc contribution           =     -16.79608807 Ry
ewald contribution        =     403.53336936 Ry
smearing contrib. (-TS)   =       0.00091689 Ry

convergence has been achieved in  14 iterations
```

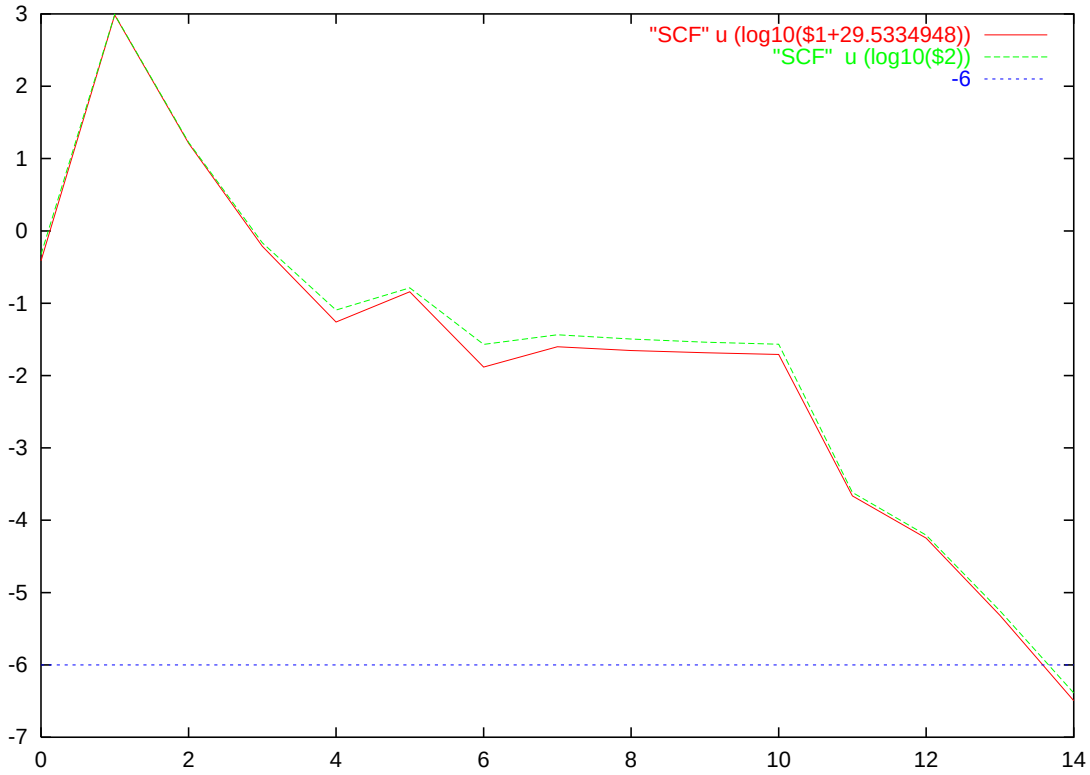
```
Forces acting on atoms (Ry/au):

atom    1 type  1   force =     0.00000000    0.00000000   -0.00030967
atom    2 type  1   force =     0.00000000    0.00000000   -0.00017252
atom    3 type  1   force =     0.00000000    0.00000000    0.00106407
atom    4 type  1   force =     0.00000000    0.00000000    0.00055948
atom    5 type  1   force =     0.00000000   -0.00000000    0.00032532
atom    6 type  1   force =     0.00000000   -0.00000000   -0.00011570
atom    7 type  1   force =     0.00000000    0.00000000    0.00011570
atom    8 type  1   force =     0.00000000    0.00000000   -0.00032532
atom    9 type  1   force =     0.00000000   -0.00000000   -0.00055948
atom   10 type  1   force =     0.00000000   -0.00000000   -0.00106407
atom   11 type  1   force =     0.00000000   -0.00000000    0.00017252
atom   12 type  1   force =     0.00000000   -0.00000000    0.00030967

Total force =     0.001839     Total SCF correction =     0.000950

Writing output data file be0001.save
```

```
PWSCF        :  4m 4.55s CPU time,     4m20.75s wall time

init_run     :     4.55s CPU
electrons    :   239.20s CPU
forces       :     0.60s CPU

Called by init_run:
wfcinit      :     4.39s CPU
potinit      :     0.05s CPU

Called by electrons:
c_bands      :   210.23s CPU (      14 calls,  15.016 s avg)
sum_band     :    28.02s CPU (      14 calls,   2.002 s avg)
v_of_rho     :     0.40s CPU (      15 calls,   0.027 s avg)
mix_rho      :     0.20s CPU (      14 calls,   0.014 s avg)

Called by c_bands:
init_us_2    :     0.83s CPU (     810 calls,   0.001 s avg)
cegterg      :   209.68s CPU (     378 calls,   0.555 s avg)
```

```
Called by *egterg:
h_psi         :   163.66s CPU (    1688 calls,   0.097 s avg)
g_psi         :     1.69s CPU (    1283 calls,   0.001 s avg)
cdiaghg       :     7.81s CPU (    1661 calls,   0.005 s avg)

Called by h_psi:
add_vuspsi    :     3.04s CPU (    1688 calls,   0.002 s avg)

General routines
calbec        :     3.76s CPU (    1715 calls,   0.002 s avg)
cft3          :     0.19s CPU (      49 calls,   0.004 s avg)
cft3s         :   169.46s CPU (   53864 calls,   0.003 s avg)
davcio        :     0.03s CPU (    1188 calls,   0.000 s avg)
```

```
prompt>  grep -e 'total energy' -e ' scf ' pw.out | \
         awk '/l e/{e=$(NF-1)}/ scf /{print e, $(NF-1)}'

-29.25526792 0.39433819
-18.34331063 667.85650410
-28.76713788 26.02680590
-29.51328737 0.34710555
-29.53372054 0.09027705
-29.54098991 0.10848232
-29.54224824 0.02339560
-29.54094557 0.02753465
-29.53957917 0.02582753
-29.53811930 0.02456945
-29.53351841 0.02000542
-29.53446102 0.00068346
-29.53449785 0.00000162
-29.53449845 0.00000030
```

```
prompt>  grep -e 'total energy' -e ' scf ' pw.out | \
         awk '/l e/{e=$(NF-1)}/ scf /{print e, $(NF-1)}' > SCF
```



Where can I find some useful information about PWscf ?

```
prompt > ls  $espresso_dir/Doc/
```

In particular INPUT_PW.html contains a rather complete description of the input of PWscf.

Similarly INPUT_PP.html, INPUT_PH.html,... contain descriptions of post processing, phonon...

```
prompt > ls  $espresso_dir/examples/
```

This directory contains a number of example scripts that illustrate (some) of the features implemented in PWscf and related codes.

There is a GUI for PWscf and the other codes in the package.
It can be used in order to have on-line help and to prepare well-formed input files.

When everything else fail read the manual at: Doc/user_guide.pdf
or online at http//www.quantum-espresso.org/wiki

THE END