QE-emacs-modes User's Guide (v.6.2)

Contents

1 Introduction

This guide covers the usage of `QE-emacs-modes` package (aka `QE-modes`): an open-source collection of Emacs major-modes for making the editing of Quantum ESPRESSO (QE) input files somewhat easier and more comfortable with Emacs editor. The package provides syntax highlighting (see Figure 1), basic auto-indentation, and several utility commands.

2 Terms of use

`QE-modes` is free software, released under the GNU General Public License. See: `http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt`, or the file `License` in the Quantum ESPRESSO distribution.

The `QE-modes` package was written by Anton Kokalj. The implementation of `QE-modes` was made possible by several useful and helpful resources that are gratefully acknowledged, in particular: *Mode Tutorial* of Scott Andrew Borton (`https://www.emacswiki.org/emacs/ModeTutorial` for indentation code), *Derived*

Mode and *Sample Mode* pages (`https://www.emacswiki.org/emacs/DerivedMode`, `https://www.emacswiki.org/emacs/SampleMode`) as well as the very useful resources of Xah Lee (`http://ergoemacs.org/emacs/elisp_syntax_coloring.html`). Last but not the least Sebastijan Peljhan is acknowledged for his work on `xsf-mode` that inspired the idea of writing the `QE-modes`.
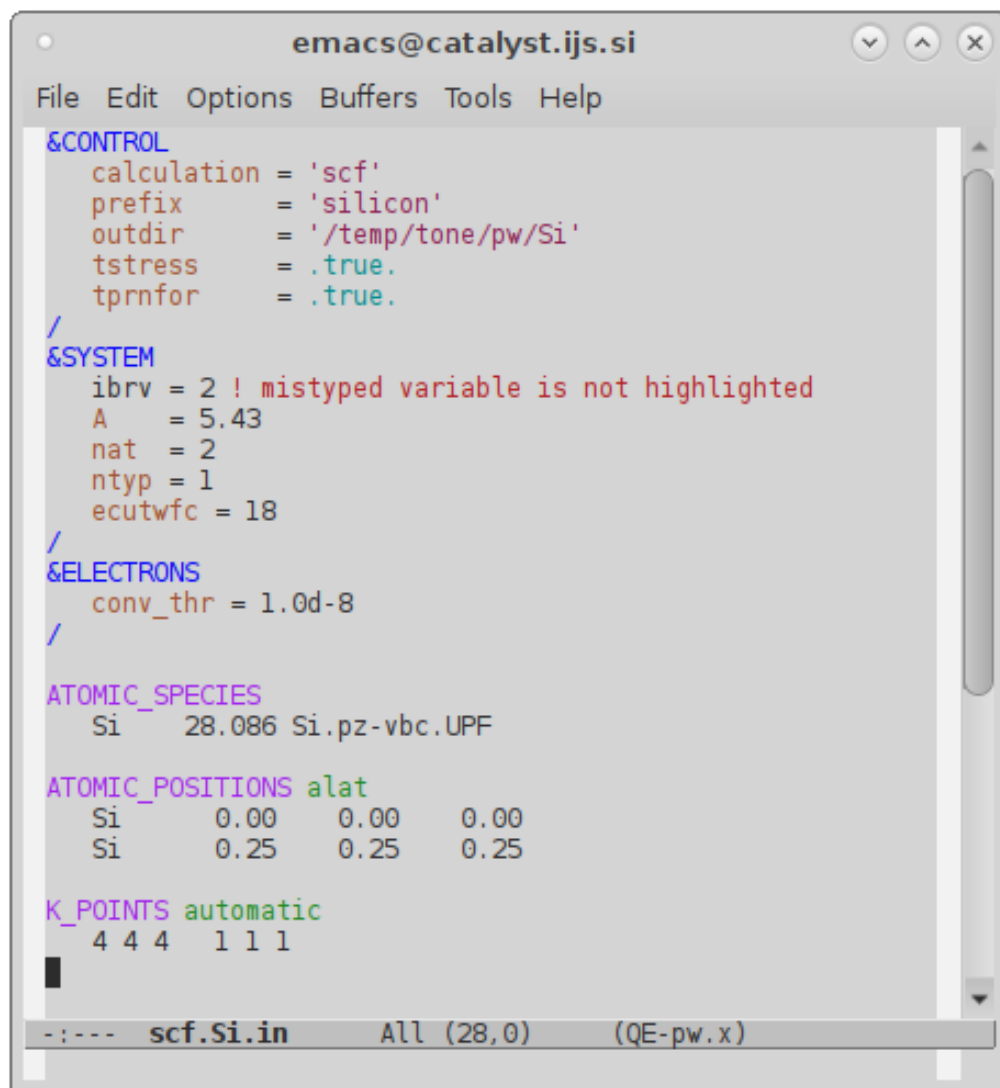

Figure 1: A `pw.x` input file opened with `pw-mode` in Emacs. Note the highlighted elements: namelists and their variables (blue and brown), cards and their flags (purple and green), comments (red), string and logical variable values (burgundy and cyan, respectively). Note that mistyped variable (i.e. `ibrv` instead of `ibrav`) is not highlighted.

3   Installation

The installation of `QE-modes` package consists of two parts: (i) installing the package itself and (ii) informing Emacs about it by editing the `user-init-file` (typically `$HOME/.emacs`).

3.1 Installing the `QE-modes` package

Once the `QE-modes-6.2.tar.gz` archive is unpacked and you are located in its root directory, the installation is trivial. Simply copy the whole `qe-modes` subdirectory to appropriate place. To facilitate this copying on Unix-like operating systems, one can use:

```
./install.sh
```

which will install the package in the `qe-modes` subdirectory of the `$HOME/.emacs.d/` directory. If you prefer to install it into other directory, then use:

```
prefix=where-to-install ./install.sh
```

which will install the package in the `qe-modes` subdirectory of *where-to-install* directory.

3.2 Editing the `user-init-file` file

An example snippet of `user-init-file` for `QE-modes` is provided by the `qe-modes.emacs` file in the package root directory and the text below explains it. In particular, to use the installed `QE-modes` in Emacs, add the following to your `user-init-file` (e.g. `$HOME/.emacs`):

```
;; make sure package is visible to emacs (if needed)
(add-to-list 'load-path "/full/path/name/of/qe-modes")

;; load the package
(require 'qe-modes)
```

where */full/path/name/of* is the directory where the `qe-modes` are installed (either the `$HOME/.emacs.d/` or the above *where-to-install*).

Furthermore, we can specify some filename patterns so that Emacs will automatically recognize from the filename if it is some variant of the QUANTUM ESPRESSO input file. Say that we use the `.in` extension for the QUANTUM ESPRESSO input files in general and more specifically, the `pw.`, `scf.`, `relax.`, and `vc-relax.` prefixes for the `pw.x` input files and `neb.`, `ph.`, and `pp.` prefixes for the `neb.x`, `ph.x`, and `pp.x` input files. These filename recognitions can be achieved by:

```
;; automatically open the *.in files with generic QE mode
(add-to-list 'auto-mode-alist  '("\\.in\\'" . qe-mode))

;; automatically open the pw*.in, scf*.in, relax*in, vc-relax*.in files
;; with pw.x mode
(add-to-list 'auto-mode-alist  '("pw.*\\.in\\'" . pw-mode))
(add-to-list 'auto-mode-alist  '("scf.*\\.in\\'" . pw-mode))
(add-to-list 'auto-mode-alist  '("relax.*\\.in\\'" . pw-mode))
(add-to-list 'auto-mode-alist  '("vc-relax.*\\.in\\'" . pw-mode))

;; automatically open the neb*.in files with neb.x mode
(add-to-list 'auto-mode-alist  '("neb.*\\.in\\'" . neb-mode))

;; automatically open the ph*.in files with ph.x mode
(add-to-list 'auto-mode-alist  '("ph.*\\.in\\'" . ph-mode))
```

```
;; automatically open the pp*.in files with pp.x mode
(add-to-list 'auto-mode-alist  '("pp.*\\.in\\'" . pp-mode))
```

Beware that the more general `*.in` pattern for the generic `qe-mode`[1] should be specified first or else any `*.in` file will be recognized as generic QE input file.

For those who are fans of regular-expressions, the above four lines for `pw-mode` can be expressed by the following one-liner:

```
(add-to-list 'auto-mode-alist '("\\(pw\\|scf\\|\\(?:vc-\\)?relax\\).*\\.in\\'" . pw-mode))
```

Once the package is installed according to the above instructions, we are ready to use it. Let us, for the sake of example, open an existing `pw.x` input file whose name does not match the above specified filename pattern for the `pw-mode`. In such cases we can load the mode with `M-x pw-mode` command and we will get the content of the file highlighted as in Figure 1.

4   Usage

4.1   Available modes defined by `qe-modes`

The `QE-modes` package contains a generic `qe-mode` and the following specific modes: `pw-mode`, `neb-mode`, `cp-mode`, `ph-mode`, `ld1-mode`, and `pp-mode`. The difference between them is only in the extent of the syntax highlighting and auto-indentation. Namely, these modes recognize and highlight namelists (and their variables) and cards (and their options/flags) that they know about. The generic `qe-mode` is aware of all of them for all those QUANTUM ESPRESSO programs that have explicit documentation in the form of `INPUT_PROG.html` files (where *PROG* typically stands for the uppercase name of the program). In contrast, a given specific mode is aware only of namelists, variables, cards, and options of the corresponding program.

4.2   Commands

The `QE-modes` package provides the following commands:

- `M-X mode-mode`
  toggles the respective mode, where *mode* is one of qe, pw, neb, cp, ph, ld1, or pp

- `M-x indent-region` or `C-M-\`
  indents region according to `qe-modes` rules, i.e., namelist and card names are left aligned to the first column, while their content is indented by `qe-indent` spaces to the right (see Figure 1; default value of `qe-indent` is 3)

- `M-x prog-insert-template`
  inserts a respective input file template (see Figure 2); this command may not be defined for all the *prog*s; currently supported *prog*s are: pw, pp, neb, ph, dynmat, ld1, projwfc, dos, and bands.

- `M-x prog-NAMELIST`
  inserts a blank namelist section named *NAMELIST*

---

[1]Please note the difference between `qe-modes` and `qe-mode`: the first implies the whole package, whereas the second means the generic QE mode, which is only one among the available modes in the `qe-modes` package.

Figure 2: The result of executing the `M-x pp-insert-template` command, which insert a template for the `pp.x` input file into the current buffer.

- `M-x` *prog–CARD*
    inserts a blank card section named *CARD*

- `M-x` *prog–variable*
    inserts a namelist variable named *variable*

The above italicized words have the following meaning:

- *prog* stands for the lowercase name of respective program without the `.x` suffix (i.e. it is the lowercase variant of the *PROG* in the respective `INPUT_PROG.html` filename)

- *NAMELIST* is the uppercase name for a given Fortran namelist

- *CARD* is the uppercase name for a given card

- *variable* is the lowercase name for a given namelist' variable

Note that in the above commands the spelling of namelist and card names (*NAMELIST* and *CARD*) are intentionally made uppercase as to differentiate them from the names of variables

which are intentionally made lowercase.[2]

4.3 Auto-completion mechanism

It may at first seem that the above described commands are not a big deal. But given that QUANTUM ESPRESSO contains hundreds of variables it is difficult to remember the precise spelling for all of them. It is here where these commands becomes useful due to Emacs auto-completion mechanism. For example, typing a space or tab after `M-x pw-C` prints all the namelists and cards that starts with letter "C", i.e.:

```
Possible completions are:
pw-CELL                         pw-CELL_PARAMETERS
pw-CONSTRAINTS                  pw-CONTROL
```

whereas typing a space or tab after `M-x pw-c` prints all the `pw.x` variables that starts with letter "c", i.e.:

```
Possible completions are:
pw-c                            pw-calculation
pw-cell_dofree                  pw-cell_dynamics
pw-cell_factor                  pw-celldm
pw-constrained_magnetization    pw-conv_thr
pw-conv_thr_init                pw-conv_thr_multi
pw-cosab                        pw-cosac
pw-cosbc
```

From this list we can see that there is only one variable that starts with "ca", hence typing `M-x pw-ca[space][return]`, where `[space][return]` stands for space and return keys, prints at the point position of the current buffer:

```
calculation = ''
```

4.4 Controlling indentation

The basic indentation offset in `qe-modes` is 3. It is controlled by `qe-indent` variable. Hence if you want to change it, add the following into your `user-init-file` (e.g. `$HOME/.emacs`):

```
(setq qe-indent myOffset)
```

where *myOffset* is the integer value of the offset of your choice. For no indentation, set the `qe-indent` to 0 (this implies that auto-indentation will make all lines non-indented).

To disable the auto-indentation for a given mode (are you really sure you want to do this), add the following into your `user-init-file`:

```
(add-hook 'mode-mode (lambda () (setq indent-line-function 'indent-relative)))
```

where *mode* is qe, pw, neb, cp, ph, ld1, or pp.

---

[2]Note that in QUANTUM ESPRESSO the namelist and variable names are case-insensitive, while card names are case-sensitive.

4.5   Note to `Vi` users

A simple way to get a `QE-modes` aware `Vi`-compatible editor is to use the `Evil` package – an extensible **vi l**ayer for Emacs (`https://bitbucket.org/lyro/evil/wiki/Home`). With the `Evil` mode enabled, Emacs will behave like the `Vi` editor, but with the `QE-modes` support.