

EES SOFTWARE CHEATSHEET

LIBRARY FILE STRUCTURE

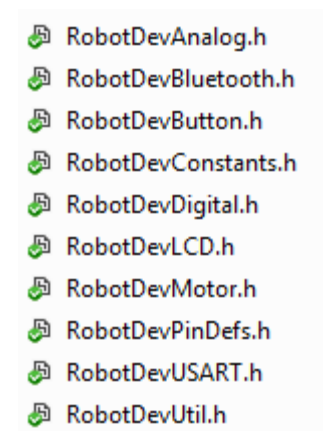
Specific projects are placed under the 'EES' folder, if you are creating a new project please use a clear and concise name using [upper camel case](#), you can create subfolders within the project folders if you really need to.

All header files are contained in the project folder.

Folder Example:



Files inside the RobotDev folder:



FILE NAMING CONVENTIONS

All library code is stored in header files, as they are easier to update and maintain whilst giving the ability to be included in other source files. All headers are prefixed with their specific project acronym, for example a header file in libRobotDev is prefixed with RobotDev then with a short word that shows what the header file is for.

See example above

FUNCTION CONVENTIONS

ALL functions are prefixed with their appropriate project acronym, and if possible a descriptive word, for example a function in libRobotDev in RobotDevAnalog.h that reads an analog value from a specified pin would be `RDAnalogRead()`, or another function in the same header that initializes the ADC would be `RDAnalogInit()`.

All functions' starting parenthesis are on the same line as the function declaration.

Example:

```
void RDAnalogInit(unsigned char prescaler){...  
}
```

FILE HEADERS

All files must be prefixed with the standard header:

```
/* Project Name
 * Filename
 * Purpose:
 * Created:
 * Author(s):
 * Status:
 */
```

An example of a standard header in a libRobotDev header file:

```
/* libRobotDev
 * RobotDevAnalog.h
 * Purpose: Abstracts all micro analog functions
 * Created: 25/07/2014
 * Author(s): Jerry Luck
 * Status: UNTESTED
 */
```

If you contribute to a file you are expected to place your name in the authors section of the header, this gives you accountability for the code you added/modified. **Only senior members** may change the status field of the header as we only want robust and optimal code that actually works in the field, which may take time to test properly.