### Coding Standards

The java pseudo code follows the Google Java Style.   
Source to Google Java Style:<https://google-styleguide.googlecode.com/svn/trunk/javaguide.html>

Php code used in this project follows the PSR-1 and PSR-2 coding standards.  
Source: <http://www.php-fig.org/psr/psr-1/> and <http://www.php-fig.org/psr/psr-2/>

##### Translating to pseudo java:

The java program starts by declaring the output variables. The names of the output variables will keep their original name, without spaces, in a camelCase form. The variable type will be determined from the Output table.

The inputs follow the same pattern.

Every states is represented as a function, keeping their name in the camelCase fashion, they will be all void functions due to the fact that they do not return anything.

Every state function will run preconditions if any, then check for specific input values using ***if*** statements, if an ***if*** statement is satisfied, there will be changes to the output values to match the next states output values and then the next state function is called according to the state transition diagram, if no ***if*** statement is satisfied the current function is recalled.

The program is always looping, consequence of no deadlocks in the state machine as proven by the uppaal model test.

Example: Initial -> Calibrate\_Sensor

So in this example the function initial is currently running, there are no preconditions to be checked, if the inputs have the desired value, in our case we check if the push button is pressed by the sorter, if so we will have the sorter moved down by activating the sorter motor via having the Hbridge0 variable set to 1. After this we call calibrateSensor function and if the if statement wasn’t satisfied we recall initial entering a loop.

##### Translating from java to php:

The translation will be natural, replacing every java parts with their php counter parts based on the coding standards above.