**3.2.3 Battery Management System API**

Header File: bmucell.h

Implementation: bmucell.cpp

**Public Methods**

|  |
| --- |
| **static Cell::setup()**  Configures pins used by all Cells. Must be called before all methods except the constructor. |
| **Cell::Cell(unsigned int cellNumber)**  Constructs a Cell and configures pins used by the specific Cell indicated by cellNumber. The mapping of cellNumber to the required pins and the data acquisition procedure is implementation specific. |
| **void Cell::update()**  Updates all measurements for the specific Cell. Does not identify or handle error conditions. |
| **bool Cell::checkErrors()**  Identifies and handles error conditions for the specific Cell. This method should be called soon after **Cell::update()** is called, in order to use the most recent measurements.  The return value is true if an error condition is detected. |
| **void Cell::logData(Stream& stream)**  Logs the measurements (temperature, voltage, current in, and current out) to a specified Stream. |
| **void Cell::logErrors(Stream& stream, bool verbose)**  Logs the error condition to a specified Stream. If verbose is true, then the errors will be displayed in English; otherwise the error conditions will be logged hexidecimal form. A lookup table to determine error conditions is in Appendix n. |
| **double Cell::getTemperature()**  Gets the most recently read temperature, in degrees Celsius. |
| **double Cell::getVoltage()**  Gets the most recently read voltage, in volts. |
| **double Cell::getCurrentIn()**  Gets the most recently current entering the cell, in amperes. Note that this value is considered invalid when error conditions prevent charging, but in practice will be close to zero. |
| **double Cell::getCurrentOut()**  Gets the most recently current exiting the cell, in amperes. Note that this value is considered invalid when error conditions prevent discharging, but in practice will be close to zero. |