- SDCard::init() condition:
    - Some operations required checking the file size of the SD Card before the function to read the file was called, therefore this function initialises the SD Card and its library, opens the text file, reads and saves the file size on setup.

SCENARIO	Expected OUTPUT	Pass/Fail	Comments	Tested By
SD Card has not been inserted	“Initialization failed”	Pass		Natalie
SPI bus has begun successfully	“SDCard Initialised”	Pass		Natalie

- SDCard::fileHasReachedSizeLimit() condition:
    - This function is used to check whether the SD Card text file has reached its acceptable size limit.

SCENARIO	Expected OUTPUT	Pass/Fail	Comments	Tested By
File size is under limit - 3,004	False	Pass		Natalie
File size is almost at limit (200 bytes left) - 7,741,678,351	True	Fail	We stored the file size variable as a 32 bit int (in case the size was too large), and divided the size by 10,000 to lose accuracy. However the test kept failing.	Natalie
File size is equal	True	Fail	We stored the	Natalie

to limit - 7,741,678,551			file size variable as a 32 bit int (in case the size was too large), and divided the size by 10,000 to lose accuracy. However the test kept failing.	

- SDCard::writeToLog(String data):
    - This function handles writing to the SD Card text file (log). It takes a string and prints it into the text file.

SCENARIO	Expected OUTPUT	Pass/Fail	Comments	Tested By
Writing data to log	True - The number of bytes written to the file is equal to the number of bytes of the parameter string	Pass		Natalie
No arguments given	Function expected one argument, none given	Pass		Natalie
Wrong argument type given e.g. 55	Function expected String argument, integer given	Pass		Natalie

- SDCard::testReadLog(String data):
    - This function ensures that the system is able to read values from the SD Card. The function should return the same string that was previously written by the writeToLog function.

SCENARIO	Expected OUTPUT	Pass/Fail	Comments	Tested By
A test string is given e.g. “test string” - this is compared with the last string written by the writeToLog function	“Reading: passed:” “test string”	Fail		Natalie

- Lorawan::join():
    - Attempt to join the network

SCENARIO	Expected OUTPUT	Pass/Fail	Comments	Tested By
Attempt to connect with the correct AppEui and AppKey set close to a Gateway	Lorawan has connected	Pass		Daniel
Attempt to connect with the correct AppEui and AppKey set far away from a gateway	Lorawan has connected	Fail	Needs a reachable gateway to connect	Daniel
Attempt to connect with the incorrect AppEui and AppKey set	Lorawan fails to join	Pass		Daniel

- Lorawan::provision():
    - Helper function that provisions with the appEui and appKey previously set on the Lorawan class

SCENARIO	Expected OUTPUT	Pass/Fail	Comments	Tested By
Attempt to provision with keys of correct length	Provisioned the keys successfully and returns true	Pass		Daniel
Attempt to provision with keys of incorrect length	Provision failed and returns false	Pass		Daniel

Lorawan::setSpreadFactor(uint8_t spreadFactor):
Lorawan::getSpreadFactor()

- Sets the spread factor to join and send transmissions with. The higher the spread factor the less that can be sent overall but more likely will be able to send succesfully. The min and max of spreadfactores is 7 and 12 respectively

SCENARIO	Expected OUTPUT	Pass/Fail	Comments	Tested By
Setting the default SpreadFactor of 7	SpreadFactor set to 7	Pass		Daniel
Setting the SpreadFactor to 9	SpreadFactor set to 9	Pass		Daniel
Setting an extremely low SpreadFactor of 0	SpreadFactor set to 7	Pass		Daniel
Setting an extremely high SpreadFactor of 200	SpreadFactor set to 12	Pass		Daniel

- Lorawan::sendReading(int16_t reading, uint8_t powerLevel):
    - Send a Reading Uplink through lorawan to the TTN App

SCENARIO	Expected OUTPUT	Pass/Fail	Comments	Tested By
Sending a reading of 3000 with a power level of 98 close to the gateway	Data is successfully received with a 1 returned from the function	Pass		Daniel
Sending a reading of 3000 with a power level of 98 away from the gateway	Data is successfully received with a 1 returned from the function	Fail	Was unable to transmit	Daniel

- Lorawan::sendStillAlive(uint8_t powerLevel):
    - Send a Still Here Uplink through lorawan to the TTN App

SCENARIO	Expected OUTPUT	Pass/Fail	Comments	Tested By
Sending a still here with a power level of 95 close to the gateway	Data is successfully received with a 1 returned from the function	Pass		Daniel
Sending a still here with a power level of 95 away from the gateway	Data is successfully received with a 1 returned from the function	Fail	Was unable to transmit	Daniel

- Lorawan::sendGenericError(uint8_t powerLevel):
    - Send a Generic Error Uplink through lorawan to the TTN App

SCENARIO	Expected OUTPUT	Pass/Fail	Comments	Tested By
Sending a generic error with a power level of 95 close to the gateway	Data is successfully received with a 1 returned from the function	Pass		Daniel
Sending a generic error with a power level of 95 away from the gateway	Data is successfully received with a 1 returned from the function	Fail	Was unable to transmit	Daniel

- Lorawan::sendMicrocontrollerError(uint8_t powerLevel):
    - Send a Microcontroller Error Uplink through lorawan to the TTN App

SCENARIO	Expected OUTPUT	Pass/Fail	Comments	Tested By
Sending a microcontroller error with a power level of 86 close to the gateway	Data is successfully received with a 1 returned from the function	Pass		Daniel
Sending a microcontroller error with a power level of 86 away from the gateway	Data is successfully received with a 1 returned from the function	Fail	Was unable to transmit	Daniel

- Lorawan::sendSensorError(uint8_t powerLevel):
    - Send a Sensor Error Uplink through lorawan to the TTN App

SCENARIO	Expected OUTPUT	Pass/Fail	Comments	Tested By
Sending a sensor error with a power level of 72 close to the gateway	Data is successfully received with a 1 returned from the function	Pass		Daniel
Sending a sensorerror with a power level of 72 away from the gateway	Data is successfully received with a 1 returned from the function	Fail	Was unable to transmit	Daniel

- Lorawan::sendBatteryError(uint8_t powerLevel):
    - Send a Battery Error Uplink through lorawan to the TTN App

SCENARIO	Expected OUTPUT	Pass/Fail	Comments	Tested By
Sending a battery error with a power level of 56 close to the gateway	Data is successfully received with a 1 returned from the function	Pass		Daniel
Sending a battery error with a power level of 56 away from the gateway	Data is successfully received with a 1 returned from the function	Fail	Was unable to transmit	Daniel

- Lorawan::sendStorageError(uint8_t powerLevel):
    - Send a Storage Error Uplink through lorawan to the TTN App

SCENARIO	Expected OUTPUT	Pass/Fail	Comments	Tested By
Sending a storage error with a power level of 43 close to the gateway	Data is successfully received with a 1 returned from the function	Pass		Daniel
Sending a storage error with a power level of 42 away from the gateway	Data is successfully received with a 1 returned from the function	Fail	Was unable to transmit	Daniel

- Sensor::getCurrentMeasurement():
    - Use ultrasonic sensor to calculate the current river depth level in mm.

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Get measurement whilst sensor is at minimum distance (< 300mm)	Value between 4710mm and 4690mm	Pass		Dharius
Get measurement whilst sensor is below possible river depth	0mm (Should handle negative value and adjust to 0)	Pass	Failed initially. Added fix.	Dharius
Get measurement	Between 1990mm and	Pass		Dharius

where sensor is out of range of river top (> 5000mm)	2010mm			

- Sensor::isCurrentWorthSending(int16_t currentMeasurement):
    - Check if the current measurement is different enough from the last one sent (worth sending?).

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Current measurement is within the range difference threshold of last measurement sent	False	Pass		Dharius
Current measurement is outside of the range threshold of last measurement sent	True	Pass		Dharius

- Processor::getBatteryVoltage():
    - Calculates and returns current battery voltage value.
    - Gets an average of 5 readings over 5 seconds.

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Battery is	Between 4.1v	Pass		Dharius

plugged in and fully charged	and 4.3v			
Battery is plugged in and mostly charged	Between 3.7v and 4v	Pass		Dharius
Battery is plugged in (not fully charged) along with USB power	4.2v (full power directly from USB power supply)	Pass	In this scenario we are expecting the output to be the voltage supplied from the USB (ignoring the battery)	Dharius

- Processor::getBatteryVoltageByte():
    - Converts battery voltage to a byte to be sent to the API .
    - Done to save TTN bandwidth (So we don't need to send a full floating point number).

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Battery is plugged in and fully charged	Between 90 and 110	Pass		Dharius
Battery is plugged in and mostly charged	Between 50 and 80	Pass		Dharius
Battery is plugged in (not fully charged) along with USB power	100 (full power directly from USB power supply)	Pass		Dharius

- Processor::getEstimatedPowerLevel():
    - Calculates and returns estimated battery percentage (powerlevel/capacity) based on voltage and mAh of the battery.
    - Expected default battery:
    - 4.2v max (and 3.2v min cut off)
    - 700mAh

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Call while battery is unplugged (USB Power)	4.15v 95%	Pass		Dharius
Call with battery plugged in	Voltage with percentage value to match	Pass		Dharius

- Processor::recalibrateSensor():
    - Used to calibrate the sensor as done in initial setup.
    - Allows the engineer to input an initial depth for the river, for which the distance from sensor to river bed can be calculated.

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Input same river depth as before, but move sensor lower	Distance to river bed decreased based on new height	Pass		Dharius
Input same river depth as before, but move sensor higher	Distance to river bed increased based on new height	Pass		Dharius
Input new river depth and leave sensor where it is	Distance to river bed remains the same as before (based	Pass		Dharius

	on new river depth being higher or lower than before)			

- Processor::triggerClearFlash():
    - Clear the device settings variables stored in the flash storage.

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Clear flash when no variables are stored	setupDone_FlashStore boolean variable remains false	Pass		Dharius
Clear flash when device settings variables are stored	setupDone_FlashStore boolean variable remains is set from true to false	Pass		Dharius

- Processor::adjustAppEui(String newAppEui):
    - Adjust the application's Eui value

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Input same value as before	EUI remains the same	Pass		Dharius
Input new correct format EUI value	EUI is changed to new value in class variable and flash	Pass		Dharius

	storage			
Input EUI with an incorrect format (just an integer or only 3 characters long)	Output error message and ask to enter a valid EUI	Fail	Have not implemented handling of incorrect string formats (Need to add EUI regex expectations to check against the string)	Dharius

- Processor::adjustAppKey(String newAppKey):
    - Adjust the application's key

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Input same value as before	AppKey remains the same	Pass		Dharius
Input new correct format AppKey value	AppKey is changed to new value in class variable and flash storage	Pass		Dharius
Input AppKey with an incorrect format (just an integer or only 3 characters long)	Output error message and ask to enter a valid EUI	Fail	Have not implemented handling of incorrect string formats (Need to add AppKey regex expectations to check against the string)	Dharius

- Processor::adjustARModeDelay(uint32_t newDelayPeriod):
    - Set delay period to be used whilst the device is in AR (Accelerated Readings) Mode.

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Increase delay period	Delay period is increased for class variable and flash storage variable	Pass		Dharius
Decrease delay period	Delay period is decreased for class variable and flash storage variable	Pass		Dharius
Input the same delay period	Delay period should remain the same	Pass		Dharius
Input value larger than uint32 can handle	Set delay to highest possible value	Fail	No handling for integers outside of range (this should ideally be handled before the function is called)	Dharius

- Processor::adjustARModeThreshold(int16_t newActivationThreshold)
    - Set max river depth threshold to trigger AR Mode (Measured in: mm).

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Increase threshold	Threshold is increased for class variable and flash storage variable	Pass		Dharius
Decrease threshold	Threshold is decreased for class variable and flash storage variable	Pass		Dharius
Input the same threshold	Threshold should remain the same	Pass		
Input value larger than uint32 can handle	Set threshold to highest possible value	Fail	No handling for integers outside of range (this should ideally be handled before the function is called)	

- Processor::activateOrDeactivateARMode()
    - Swaps delayPeriod and delayPeriodARMode variables to activate/deactivate Accelerated Readings mode.

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Called when AR Mode is off	Delay Periods are swapped	Pass		Dharius

	and AR Mode is turned on			
Called when AR Mode is on	Delay Periods are swapped and AR Mode is turned off	Pass		Dharius

- Processor::adjustIgnoreThreshold(int16_t newIgnoreThreshold)
    - Set minimum river depth threshold (anything below this value will be ignored, and considered "not worth" sending) (mm).

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Increase threshold	Threshold is increased for class variable and flash storage variable	Pass		Dharius
Decrease threshold	Threshold is decreased for class variable and flash storage variable	Pass		Dharius
Input the same threshold	Threshold should remain the same	Pass		Dharius
Input value larger than int16 can handle	Set threshold to highest possible value	Fail	No handling for integers outside of range (this should ideally be handled before the function is called)	Dharius

- Processor::delayWithPeriod()
    - Makes the entire device sleep for a specified period of time.
    - Used to sleep in between taking river depth measurements.

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Delay for specific short amount of time	Device pauses for exact time expected	Pass		Dharius
Delay for specific longer amount of time	Device pauses for exact time expected	Pass		Dharius

- Processor::changeMeasurementPeriod(uint32_t milliseconds)
    - Set the delay period between each measurement to be taken.

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Increase delay period	Delay period is increased for class variable and flash storage variable	Pass		Dharius
Decrease delay period	Delay period is decreased for class variable and flash storage variable	Pass		Dharius
Input the same delay period	Delay period should remain the same	Pass		Dharius
Input value larger than uint32 can handle	Set delay to highest possible value	Fail	No handling for integers outside of range (this should ideally	Dharius

			be handled before the function is called)	

- Processor::printMeasurementToSDLog(int16_t measurement)
    - Call SDCard class function to store the last measurement sent to a log on the SD card.

SCENARIO	Expected OUTCOME	Pass/Fail	Comments	Tested By
Valid integer value is passed	Integer is converted to a string and printed to log	Pass		Dharius