### Power Sensing

#### Booster Alarm

* 1. This library shall provide a function to raise various Booster Alarms based on power readings.
  2. Null pointers as parameters shall not be allowed.
  3. If the input power is below the low input power threshold, an alarm to indicate this situation shall be raised.
  4. If the input power is above the high input power threshold, an alarm to indicate this situation shall be raised.
  5. If the output power is above the high input power threshold, an alarm to indicate this situation shall be raised.
  6. If the output power is below the low input power threshold, an alarm to indicate this situation shall be raised.
  7. If there is a loss of signal (LOS) situation, an alarm indicating a LOS situation shall be raised.
  8. If there is either a LOS situation, an input power above the high input power threshold or output power above the high output power threshold, or any combination of the three, a critical alarm shall be raised.
  9. If the low output limit is greater than or equal to the high output limit, this indicates an issue with the CAN message; an alarm indicating an invalid CAN message shall be raised.
  10. If the low input limit is greater than or equal to the high input limit, this indicates an issue in the CAN message; an alarm indicating an invalid CAN message shall be raised.
  11. Input power limits are constrained by a low and a high value, representing the maximum and the minimum values that these limits can take. Both high and low input power limits must remain within their respective bounds, or an alarm indicating an invalid CAN message shall be raised. The limits can equal their bounds.
  12. Output power limits are constrained by a low and a high value, representing the maximum and the minimum values that these limits can take. Both high and low output power limits must remain within their respective bounds, or an alarm indicating an invalid CAN message shall be raised. The limits can equal their bounds.

#### LNA Alarm

* 1. This library shall provide a function to raise various LNA Alarms based on power readings.
  2. Null pointers as parameters shall not be allowed.
  3. If the input power is below the low input power threshold, an alarm to indicate this situation shall be raised.
  4. If the input power is above the high input power threshold, an alarm to indicate this situation shall be raised.
  5. If the output power is above the high input power threshold, an alarm to indicate this situation shall be raised.
  6. If the output power is below the low input power threshold, an alarm to indicate this situation shall be raised.
  7. If there is an input power above the high input power threshold or output power above the high output power threshold, a critical alarm shall be raised.
  8. If the low input limit is greater than or equal to the high input limit, this indicates an issue in the CAN message; an alarm indicating an invalid CAN message shall be raised.
  9. If the low output limit is greater than or equal to the high output limit, this indicates an issue in the CAN message; an alarm indicating an invalid CAN message shall be raised.
  10. Input power limits are constrained by a low and a high value, representing the maximum and the minimum values that these limits can take. Both high and low input power limits must remain within their respective bounds, or an alarm indicating an invalid CAN message shall be raised. The limits can equal their bounds.
  11. Output power limits are constrained by a low and a high value, representing the maximum and the minimum values that these limits can take. Both high and low output power limits must remain within their respective bounds, or an alarm indicating an invalid CAN message shall be raised. The limits can equal their bounds.

#### Power Sensor Update

3.1 This library shall provide a function to update power sensing critical alarm statuses.

3.2 This function shall call an externally implemented function (from another module) to obtain the offset values for the specific channels. If this function returns a non-success error message, this shall be reflected in the outer function.

3.3 This function shall call an externally implemented function (from another module) to read the values of power for each channel (the function shall be executed once per channel). If this function, at any point, returns a non-success error message this shall be reflected in the outer function.

3.4 This function shall call an externally implemented function (from the ReadPowerIn2Channels module) to determine, based on the value of the power read, whether to take power readings from channel LOW or HIGH. If this function returns a non-success error message, this shall be reflected in the outer function.

3.5 This function shall call an externally implemented function (from the Power Sensors Booster Alarm module) that will determine whether or not to raise various booster alarm flags based on power readings. If a critical alarm is raised, a separate function, also externally implemented, indicating that a critical error has been detected in the booster shall be executed.

3.6 This function shall call a function (from the Power Sensors LNA Alarm module) that will determine whether or not to raise various LNA alarm flags based on power readings. If a critical alarm is raised, a separate function, also externally implemented, indicating that a critical error has been detected in the LNA shall be raised.

3.7 If the function mentioned in Requirement 3.5 returns a non-success error message, this shall be reflected in the outer function. This can only occur if the LOS settings are mis-configured, as the function from the Power Sensors Booster Alarm module checks for a LOS situation.

3.8 The LNA and the booster will each have a channel selection bit, enabling differentiation between the two possible input wavelengths.