VL53L0X Library

This library is used for communication between the eYFi-Mega board and the VL53L0X ToF sensors using the I2C protocol. Two pins (SDA and SCL) are required for interfacing.

The following functions can be used to initialize VL53L0X and get readings from it in “mm”:

- **void setAddress (uint8_t new_addr)**
  - Description: Used to configure I2C address of sensor connected
  - Parameters:
    - ◆ **new_addr:** 8-bit address of the I2C device (with LSB = 0)
  - Returns: void

- **uint8_t getAddress (void)**
  - Description: Used to get the current I2C address
  - Parameters: None
  - Returns: the current I2C address

- **bool initVL53L0X (bool io_2v8)**
  - Description: Initializes and configures the sensor. By default, the sensor is configured to work in 1V8 mode, but we need to configure it to work in 2V8 mode. You can refer to the datasheet for more explanation of these modes.
  - Parameters:
    - ◆ **io_2v8:** if 1, then sensor is configured in 2V8 mode

      if 0, then sensor is left in 1V8 mode
  - Returns: **1** if the initialization completed successfully

- **void startContinuous (uint32_t period_ms)**
  - Description: Starts continuous ranging measurements. If the argument **period_ms** is 0, continuous back-to-back mode is used (the sensor takes measurements as often as possible); otherwise, continuous timed mode is used, with the given inter-measurement period in milliseconds determining how often the sensor takes a measurement.

- Parameters:
  - **period_ms:** the specified inter-measurement period in milliseconds determining how often the sensor takes a measurement.
- Returns: void

- **void stopContinuous (void)**
  - Description: Stops the continuous measurements.
  - Parameters: None
  - Returns: void

- **bool setMeasurementTimingBudget (uint32_t budget_us)**
  - Description: Set the measurement timing budget in microseconds, which is the time allowed for one measurement. A longer timing budget allows for more accurate measurements. Increasing the budget by a factor of **N** decreases the range measurement standard deviation by a factor of **sqrt(N)**. Defaults to about **33 milliseconds**; the minimum is **20 ms**.
  - Parameters:
    - **budget_us:** measurement timing budget in microseconds
  - Returns: **true** if there is room to set the timing budget asked;

    false if there is no room to set the timing budget asked for

- **uint32_t getMeasurementTimingBudget (void)**
  - Description: Used to get the current measurement timing budget in microseconds
  - Parameters: None
  - Returns: void

- **uint16_t readRangeSingleMillimeters (statInfo_t *extraStats)**
  - Description: Performs a single-shot range measurement and returns the reading in millimeters
  - Parameters:
    - **extraStats:** provides additional info for the measurement, set it to **0** if not needed.
  - Returns: the range reading in millimeters, if Timeout occurs then it will return 65535.

- **uint16_t readRangeContinuousMillimeters (statInfo_t *extraStats)**
  - Description: Used to get range reading in millimeters when continuous mode is active.
  - Parameters:
    - **extraStats:** provides additional info for the measurement, set it to **0** if not needed.
  - Returns: the range reading in millimeters, if Timeout occurs then it will return 65535.