chipKIT™ MAXSonar Library Reference Manual

Overview

The MAXSonar library provides an interface to a MaxBotix ultrasonic detection module. The sensor uses an onboard microcontroller to obtain range data, which converts the data to analog, PWM, and serial signals. The serial data is sent in a packet of 5 ASCII characters.

Library Operation

Library Interface

The header file MAXSonar.h defines the interfaces to the MAXSonar library. The library is accessed via the methods and constants defined for the MAXSonar object class. To instantiate an MAXSonar object, simply include the library and instantiate an MAXSonar object.

Sensor Initialization

The MAXSonar sensor has an initial power on sequence that includes a calibration cycle. This sequence should run when the chipKIT board is powered on and the boot loader is running. However, it is suggested to include a 250ms delay, write the sensor's RX pin HIGH, and then delay for an additional 98 seconds. Each subsequent range reading begins with the RX pin being written HIGH using MAXSonar.begin(), and ends with the RX pin being written LOW using MAXSonar.end().

The MAXSonar sensor can also be operated in free-run mode. This requires a single call of MAXSonar.begin() to start, and results in a range reading being taken every ~50ms.

Enumerated Types

UNIT

Enumerates available metric and standard units that can be used with the library. This assigns of value of 0 to INCH, 1 to CM, and 2 to MM.

MODE

Enumerates available modes of communication between the chipKIT™ board and the MAXSonar sensor. This assigns a value of 0 to UART, a value of 1 to ANALOG, and a value of 2 to PULSEWIDTH.

MAXSonar Library Functions

Public Functions

uint8_t begin(MODE mode, HardwareSerial& serPin, uint8_t RXPin)

Parameters:
mode	The method to be used to access range data. For this instance, only UART is valid.
serPin	The serial port being used to communicate with the sensor. This uses the MPIDE Serial definitions (valid entries are Serial, Seral1, Serial2, etc. depending on the board).
RXPin	The digital pin number of the RX pin for the MAXSonar sensor.

Return Value:
0	Incorrect value of mode (mode != UART)
1	Setup was correctly implemented

Initializes a serial port to receive serial data from the sensor. Writes the RX pin HIGH to start the ranging and conversion process.

uint8_t begin(MODE mode, uint8_t RXPin)

Parameters:
mode	The method to be used to access range data. For this instance, UART is not valid.
RXPin	The digital pin number of the RX pin for the MAXSonar sensor.

Return Value:
0	Incorrect value of mode (mode = UART)
1	Setup was correctly implemented

Writes the RX pin HIGH to start the ranging and conversion process for analog and PWM methods of reading the sensor.

---

uint16_t getDistance(MODE mode, uint8_t pin, UNIT unit)

Parameters:

mode	The method to be used to access range data. For this instance, UART is not valid.
pin	The digital pin number of the AN or PW pin.
unit	The distance units the output should be formatted in.

Return Value:
The range detected in the specified units.

Reads available range data from the sensor via analog voltage or pulse width.

uint16_t getDistance(HardwareSerial& serPin, UNIT unit)

Parameters:

serPin	The serial port being used to communicate with the sensor. This uses the MPIDE Serial definitions (valid entries are Serial, Seral1, Serial2, etc. depending on the board).
unit	The distance units the output should be formatted in.

Return Value:
The range detected in the specified units.

Reads available range data from the sensor via a serial packet.

void end(HardwareSerial& serPin, uint8_t RXPin)

Parameters:

serPin	The serial port being used to communicate with the sensor. This uses the MPIDE Serial definitions (valid entries are Serial, Seral1, Serial2, etc. depending on the board).
RXPin	The digital pin number of the RX pin for the MAXSonar sensor.

Closes the serial port and writes the RX pin LOW to cease further conversions.

void end (uint8_t RXPin)

Parameters:

RXPin	The digital pin number of the RX pin for the MAXSonar sensor.

Writes the RX pin LOW to cease further conversions.

Private Functions

uint16_t getUnits (uint16_t data, UNIT units)

Parameters:
data   The unformatted range data (in inches) from the sensor.
unit   The distance units the output should be formatted in.

Converts data (originally in inches) to centimeters or millimeters. Defaults to inches if an invalid unit is specified.