lidar-projekti Documentation

Release 1.0

lidar-projekti

Aug 02, 2023

CONTENTS

An Arduino project that measures distances using a LiDAR sensor.

CHAPTER

ONE

FEATURES

- Reads LiDAR data via UART
- Stores to and reads measurement unit configuration from EEPROM
- Unit can be changed via a button
- **An RGB LED displays the current measurement unit**
  - Green: centimetres
  - Yellow: millimetres
  - Red: Pixhawk-drone internal units
- [UNIMPLEMENTED] Measures battery level via voltage tracking
- Sends output to external USART

CHAPTER

TWO

HARDWARE REQUIREMENTS

- Arduino Every (x1)
- TFMini Plus -LiDAR sensor (x1)
- Push button (x1)
- RGB LED or equivalent (x1)
- Wires

For the pin layout, please see the source code or documentation for the pin numbers.

CHAPTER

THREE

INSTALLATION

You can either run the source code via AVRdude yourself, or build it with PlatformIO. For the latter, a configuration file is provided within the repository itself for your convenience.

CHAPTER

FOUR

DOCUMENTATION

An Arduino Every program.

This program reads data from a LiDAR sensor and outputs distances to objects.

Type Aliases

Rust-like aliases for numeric types

typedef uint8_t **u8**

Unsigned 8-bit integer.

typedef uint16_t **u16**

Unsigned 16-bit integer.

typedef uint32_t **u32**

Unsigned 32-bit integer.

typedef uint64_t **u64**

Unsigned 64-bit integer.

typedef int8_t **i8**

Signed 8-bit integer.

typedef int16_t **i16**

Signed 16-bit integer.

typedef int32_t **i32**

Signed 32-bit integer.

typedef int64_t **i64**

Signed 64-bit integer.

Enums

enum **unit_mode**

Represents the different modes for measuremement units supported by the LiDAR sensor.

Used to store the mode in use in the device memory.

Values:

enumerator **LIDAR_CM**

Measurements in centimetres.

enumerator **LIDAR_MM**

Measurements in millimetres.

enumerator **LIDAR_PIXHAWK**

Measurements in Pixhawk drone units.

Functions

SoftwareSerial **soft_serial**(*LIDAR_RX*, *LIDAR_TX*)

Handles serial traffic with the LiDAR sensor.

void **RGB_colour**(*u8* red_light_value = 0, *u8* green_light_value = 0, *u8* blue_light_value = 0)

Sets the colour and brightness of the RGB LED based on the colour values of the parameters.

By mixing the colours, different colours can be created.

Yellow = red + green

Purple = red + blue

Cyan = green + blue

White = red + green + blue

Different concentrations produce different shades.

Parameters

- **red_light_value** – Controls the amount of red light (0-255)
- **green_light_value** – Controls the amount of green light (0-255)
- **blue_light_value** – Controls the amount of blue light (0-255)

void **set_LED**(*unit_mode* mode)

Changes the colour of the RGB LED to fixed colours based on the current LiDAR unit.

Parameters

mode – The current LiDAR unit mode

void **set_lidar_output_format**(*unit_mode* mode)

Sets the lidar output format.

Parameters

mode – The current LiDAR unit mode

void **cycle_settings()**

Changes the LiDAR unit mode.

The function cycles over all the modes available, picking the next one in the sequence, then changes the config mode to that. Afterwards it changes the RGB LED to show the new setting and stores this setting into the EEPROM.

void **read_battery_level()**

Reads the battery level and shows it via LED.

void **setup()**

Sets up the register, serial traffic, LiDAR configuration, and EEPROM config.

void **loop()**

The main program loop.

Variables

const int **LIDAR_RX** = 2

Lidar sensor data read pin, D2.

const int **LIDAR_TX** = 3

Lidar sensor data write pin, D3.

const int **BUTTON_PIN** = 4

Pin assigned for the button that controls measurement precision, D4.

const int **LED_PIN_RED** = 9

PWM pin used to control the red LED, D9.

const int **LED_PIN_GREEN** = 10

PWM pin used to control the green LED, D10.

const int **LED_PIN_BLUE** = 11

PWM pin used to control the blue LED, D11.

const int **BATTERY_VOLTAGE_PIN** = 14

ADC pin used to read battery voltage, A0 (D14) !!!!!CHECK PIN!!!!!

const *u8* **UNIT_MODE_CONFIG_ADDRESS** = 0

EEPROM address where LiDAR unit mode configuration is stored.

const long **LIDAR_UART_BAUDRATE** = 115200

LiDAR serial output frequency.

const long **LOG_SERIAL_BAUDRATE** = 115200

Serial logging output frequency.

unit_mode **config_mode**

Stores the LiDAR measurement unit.

int **button_state** = 0

Stores the current state of the button (up/down, 0/1)

int **previous_state** = {0}

Stores the previous state of the button (up/down, 0/1)

TFminiPlus **lidar**

Needed to configure the LiDAR sensor.

This documentation was built using ArduinoDocs.

INDEX