

Arduino_Serial_Communication_Library

Generated by Doxygen 1.8.1.2

Fri Nov 16 2012 17:32:02

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	1
2.1	File List	1
3	Class Documentation	1
3.1	SerialPacket Class Reference	1
3.1.1	Detailed Description	3
3.1.2	Constructor & Destructor Documentation	4
3.1.3	Member Function Documentation	4
3.1.4	Member Data Documentation	11
4	File Documentation	12
4.1	defines.h File Reference	12
4.1.1	Macro Definition Documentation	13
4.2	SerialPacket.cpp File Reference	13
4.3	SerialPacket.h File Reference	14

1 Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

SerialPacket	1
------------------------------	----------

2 File Index

2.1 File List

Here is a list of all files with brief descriptions:

defines.h	12
SerialPacket.cpp	13
SerialPacket.h	14

3 Class Documentation

3.1 SerialPacket Class Reference

```
#include <SerialPacket.h>
```

Collaboration diagram for SerialPacket:



Public Member Functions

- [SerialPacket](#) ()
[SerialPacket.cpp](#) - Library for sending sensor data packets over UART. For more information: variable declaration, changelog,... see [SerialPacket.h](#)
- void [begin](#) ()
Begin using default settings:
- void [begin](#) (long speed, uint8_t nodeID)
Begin using custom settings
- void [sendCommand](#) (uint8_t commandID, uint8_t payload)
Send a single command
- void [sendCommand](#) (uint8_t payload)
Send a single command, reuses commandID from previous packets
- void [sendCommandReply](#) (uint8_t commandID, uint8_t payload)

- Send a reply to a command*
 - void [sendDataRequest](#) (uint8_t sensorID, uint8_t payload)
- Request a single data value*
 - void [sendData](#) (uint8_t sensorID, uint8_t payload)
- Send a single data value*
 - void [sendData](#) (uint8_t sensorID, int16_t payload)
- Send a single data value*
 - void [sendData](#) (uint8_t payload)
- Send a single 8-bit data value (Arduino 'byte' type), reuses sensorID from previous packets*
 - void [sendData](#) (int16_t payload)
- Send a single 16-bit data value (Arduino 'int' type), reuses sensorID from previous packets*
 - void [sendDataArrayRequest](#) (uint8_t arrayID, uint8_t length)
- Request an array of data values*
 - void [sendDataArray](#) (uint8_t *dataArray, uint8_t length)
- Send multiple data samples in one packet by passing an array and its length*

Private Member Functions

- void [sendPacket](#) (uint8_t &payload)
 - Send out the actual 8-bit data packet (called from other 'send' functions)*
- void [sendPacket](#) (int16_t &payload)
 - Send out the actual 16-bit data packet (called from other 'send' functions)*
- void [setPacketType](#) (uint8_t type)
 - Set packet type*
- void [setCommandID](#) (uint8_t &commandID)
 - Set commandID*
- void [setSensorID](#) (uint8_t &sensorID)
 - Set sensorID*
- void [setNodeID](#) (uint8_t &nodeID)
 - Set nodeID*
- void [hexPrinting](#) (uint8_t &data)
 - HexPrinting: helper function to print data with a constant field width (2 hex values)*
- void [hexPrinting](#) (int16_t &data)
 - HexPrinting: helper function to print data with a constant field width (2 hex values)*

Private Attributes

- uint8_t [_packetType](#)
- uint8_t [_nodeID](#)
- uint8_t [_sensorID](#)
- uint8_t [_commandID](#)
- uint8_t [_parity](#)

3.1.1 Detailed Description

Definition at line 36 of file SerialPacket.h.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 SerialPacket::SerialPacket ()

[SerialPacket.cpp](#) - Library for sending sensor data packets over UART. For more information: variable declaration, changelog,... see [SerialPacket.h](#)

Constructor

Definition at line 31 of file [SerialPacket.cpp](#).

3.1.3 Member Function Documentation

3.1.3.1 void SerialPacket::begin ()

Begin using default settings:

- speed: 115200 baud
- nodeID: 0

Definition at line 40 of file [SerialPacket.cpp](#).

3.1.3.2 void SerialPacket::begin (long *speed*, uint8_t *nodeID*)

Begin using custom settings

Definition at line 48 of file [SerialPacket.cpp](#).

Here is the call graph for this function:



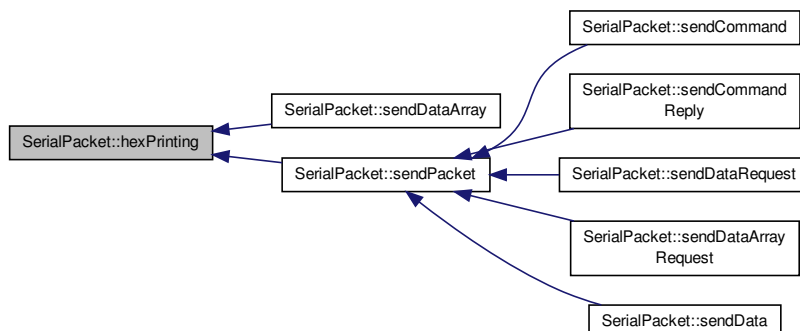
3.1.3.3 void SerialPacket::hexPrinting (uint8_t & *data*) [private]

HexPrinting: helper function to print data with a constant field width (2 hex values)

Definition at line 239 of file [SerialPacket.cpp](#).

Referenced by [sendDataArray\(\)](#), and [sendPacket\(\)](#).

Here is the caller graph for this function:



3.1.3.4 void SerialPacket::hexPrinting (int16_t & data) [private]

HexPrinting: helper function to print data with a constant field width (2 hex values)

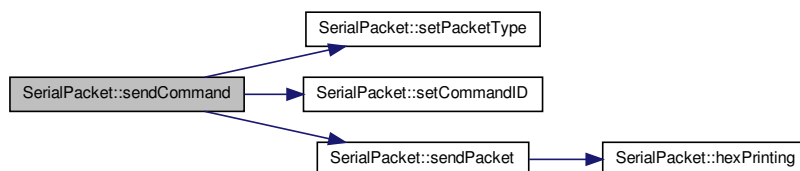
Definition at line 251 of file SerialPacket.cpp.

3.1.3.5 void SerialPacket::sendCommand (uint8_t commandID, uint8_t payload)

Send a single command

Definition at line 57 of file SerialPacket.cpp.

Here is the call graph for this function:

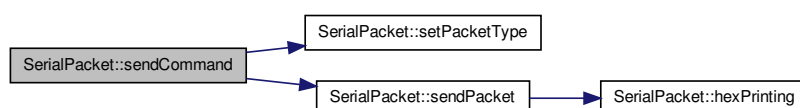


3.1.3.6 void SerialPacket::sendCommand (uint8_t payload)

Send a single command, reuses commandID from previous packets

Definition at line 67 of file SerialPacket.cpp.

Here is the call graph for this function:

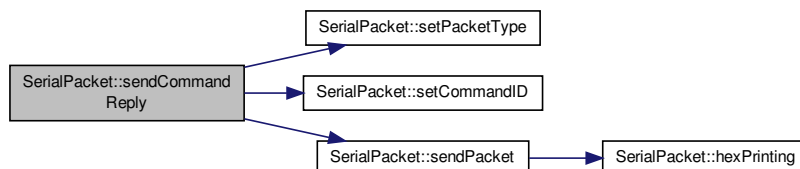


3.1.3.7 void SerialPacket::sendCommandReply (uint8_t *commandID*, uint8_t *payload*)

Send a reply to a command

Definition at line 76 of file SerialPacket.cpp.

Here is the call graph for this function:

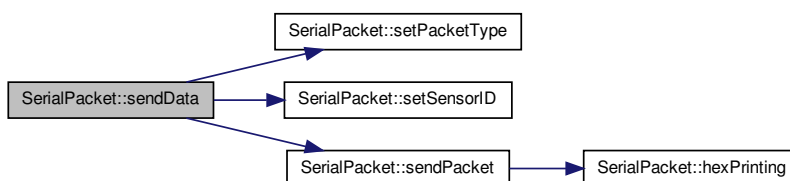


3.1.3.8 void SerialPacket::sendData (uint8_t *sensorID*, uint8_t *payload*)

Send a single data value

Definition at line 106 of file SerialPacket.cpp.

Here is the call graph for this function:

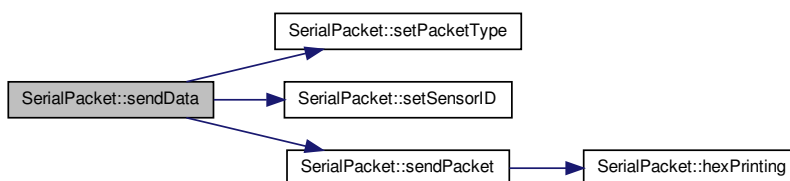


3.1.3.9 void SerialPacket::sendData (uint8_t *sensorID*, int16_t *payload*)

Send a single data value

Definition at line 116 of file SerialPacket.cpp.

Here is the call graph for this function:

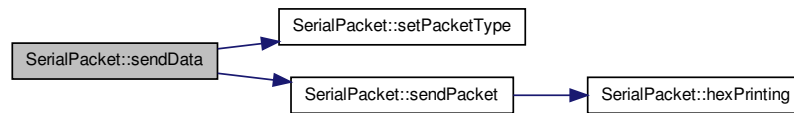


3.1.3.10 void SerialPacket::sendData (uint8_t *payload*)

Send a single 8-bit data value (Arduino 'byte' type), reuses sensorID from previous packets

Definition at line 126 of file SerialPacket.cpp.

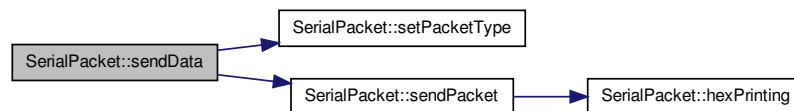
Here is the call graph for this function:

**3.1.3.11 void SerialPacket::sendData (int16_t *payload*)**

Send a single 16-bit data value (Arduino 'int' type), reuses sensorID from previous packets

Definition at line 135 of file SerialPacket.cpp.

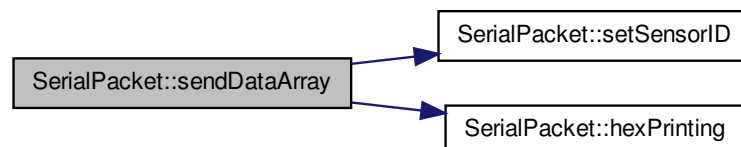
Here is the call graph for this function:

**3.1.3.12 void SerialPacket::sendDataArray (uint8_t * *dataArray*, uint8_t *length*)**

Send multiple data samples in one packet by passing an array and its length

Definition at line 198 of file SerialPacket.cpp.

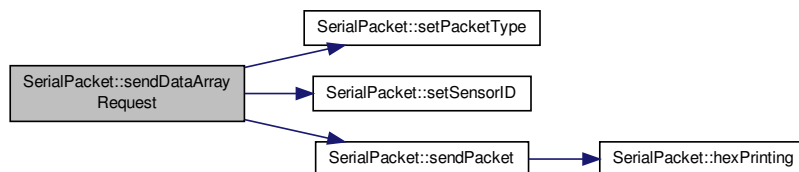
Here is the call graph for this function:

**3.1.3.13 void SerialPacket::sendDataArrayRequest (uint8_t *arrayID*, uint8_t *payload*)**

Request an array of data values

Definition at line 96 of file SerialPacket.cpp.

Here is the call graph for this function:

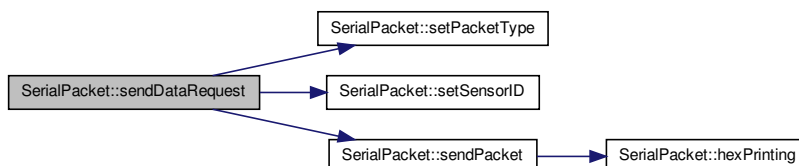


3.1.3.14 void SerialPacket::sendDataRequest (uint8_t sensorID, uint8_t payload)

Request a single data value

Definition at line 86 of file `SerialPacket.cpp`.

Here is the call graph for this function:



3.1.3.15 void SerialPacket::sendPacket (uint8_t & payload) [private]

Send out the actual 8-bit data packet (called from other 'send' functions)

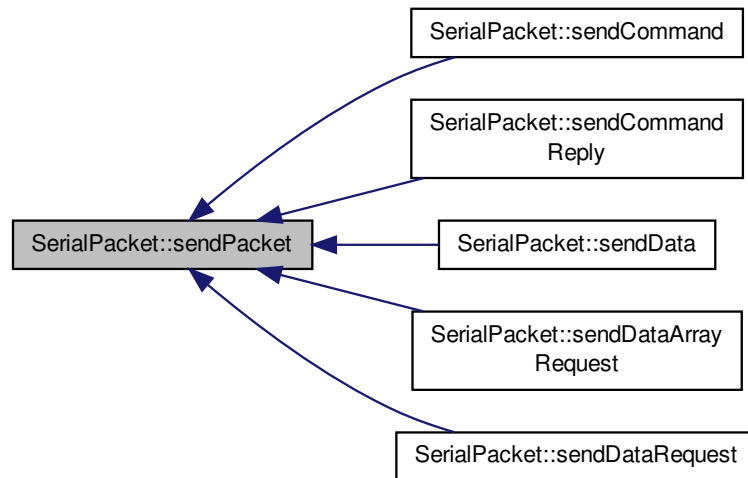
Definition at line 144 of file `SerialPacket.cpp`.

Referenced by `sendCommand()`, `sendCommandReply()`, `sendData()`, `sendDataArrayRequest()`, and `sendDataRequest()`.

Here is the call graph for this function:



Here is the caller graph for this function:



3.1.3.16 void SerialPacket::sendPacket (int16_t & payload) [private]

Send out the actual 16-bit data packet (called from other 'send' functions)

Definition at line 171 of file `SerialPacket.cpp`.

Here is the call graph for this function:



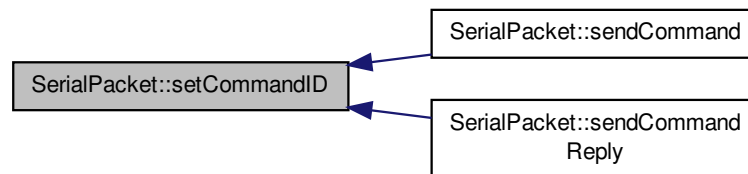
3.1.3.17 void SerialPacket::setCommandID (uint8_t & commandID) [private]

Set commandID

Definition at line 223 of file `SerialPacket.cpp`.

Referenced by `sendCommand()`, and `sendCommandReply()`.

Here is the caller graph for this function:



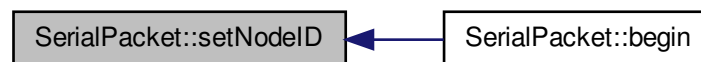
3.1.3.18 void SerialPacket::setNodeID (uint8_t & nodeID) [private]

Set nodeID

Definition at line 271 of file SerialPacket.cpp.

Referenced by `begin()`.

Here is the caller graph for this function:



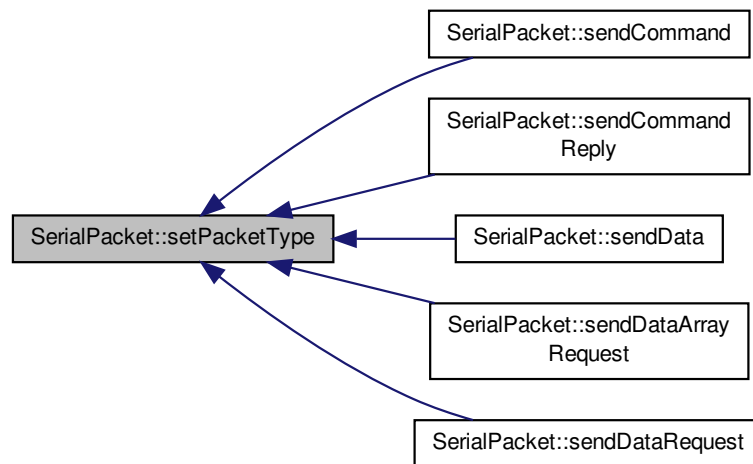
3.1.3.19 void SerialPacket::setPacketType (uint8_t type) [private]

Set packet type

Definition at line 231 of file SerialPacket.cpp.

Referenced by `sendCommand()`, `sendCommandReply()`, `sendData()`, `sendDataArrayRequest()`, and `sendDataRequest()`.

Here is the caller graph for this function:



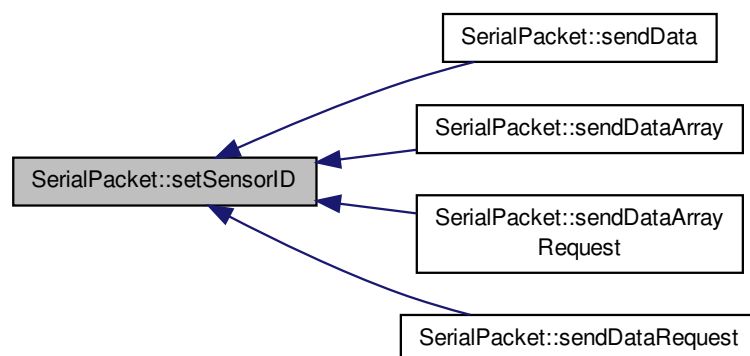
3.1.3.20 void SerialPacket::setSensorID (uint8_t & sensorID) [private]

Set sensorID

Definition at line 279 of file SerialPacket.cpp.

Referenced by `sendData()`, `sendDataArray()`, `sendDataArrayRequest()`, and `sendDataRequest()`.

Here is the caller graph for this function:



3.1.4 Member Data Documentation

3.1.4.1 uint8_t SerialPacket::commandID [private]

Definition at line 62 of file SerialPacket.h.

Referenced by `sendPacket()`, and `setCommandID()`.

3.1.4.2 `uint8_t SerialPacket::_nodeID` [private]

Definition at line 60 of file `SerialPacket.h`.

Referenced by `sendDataArray()`, `sendPacket()`, and `setNodeID()`.

3.1.4.3 `uint8_t SerialPacket::_packetType` [private]

Definition at line 59 of file `SerialPacket.h`.

Referenced by `sendDataArray()`, `sendPacket()`, and `setPacketType()`.

3.1.4.4 `uint8_t SerialPacket::_parity` [private]

Definition at line 63 of file `SerialPacket.h`.

Referenced by `sendDataArray()`, and `sendPacket()`.

3.1.4.5 `uint8_t SerialPacket::_sensorID` [private]

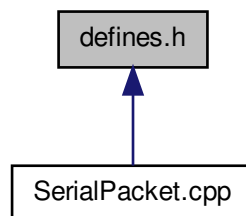
Definition at line 61 of file `SerialPacket.h`.

Referenced by `sendDataArray()`, `sendPacket()`, and `setSensorID()`.

4 File Documentation

4.1 `defines.h` File Reference

This graph shows which files directly or indirectly include this file:



Macros

- `#define SERIAL_ASCII`
- `#define COMMAND 0x01`
- `#define COMMAND_REPLY 0x02`
- `#define DATA_REQUEST 0x11`
- `#define DATA_BYTE 0x12`
- `#define DATA_INT 0x13`
- `#define DATA_ARRAY_REQUEST 0x21`
- `#define DATA_ARRAY 0x22`

4.1.1 Macro Definition Documentation

4.1.1.1 #define COMMAND 0x01

Definition at line 6 of file defines.h.

Referenced by SerialPacket::sendCommand(), and SerialPacket::sendPacket().

4.1.1.2 #define COMMAND_REPLY 0x02

Definition at line 7 of file defines.h.

Referenced by SerialPacket::sendCommandReply(), and SerialPacket::sendPacket().

4.1.1.3 #define DATA_ARRAY 0x22

Definition at line 14 of file defines.h.

4.1.1.4 #define DATA_ARRAY_REQUEST 0x21

Definition at line 13 of file defines.h.

Referenced by SerialPacket::sendDataArrayRequest(), and SerialPacket::sendPacket().

4.1.1.5 #define DATA_BYTE 0x12

Definition at line 10 of file defines.h.

Referenced by SerialPacket::sendData(), and SerialPacket::sendPacket().

4.1.1.6 #define DATA_INT 0x13

Definition at line 11 of file defines.h.

Referenced by SerialPacket::sendData(), and SerialPacket::sendPacket().

4.1.1.7 #define DATA_REQUEST 0x11

Definition at line 9 of file defines.h.

Referenced by SerialPacket::sendDataRequest(), and SerialPacket::sendPacket().

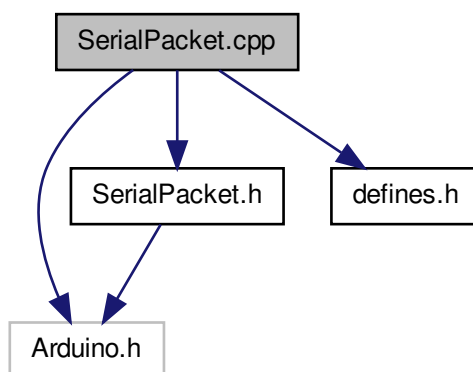
4.1.1.8 #define SERIAL_ASCII

Definition at line 3 of file defines.h.

4.2 SerialPacket.cpp File Reference

```
#include <Arduino.h>
#include "SerialPacket.h"
#include "defines.h"
```

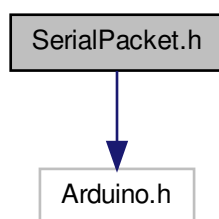
Include dependency graph for SerialPacket.cpp:



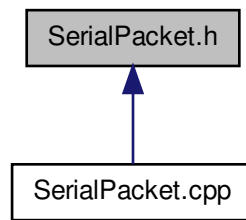
4.3 SerialPacket.h File Reference

```
#include <Arduino.h>
```

Include dependency graph for SerialPacket.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [SerialPacket](#)

Index

- [_commandID](#)
[SerialPacket](#), [11](#)
 - [_nodeID](#)
[SerialPacket](#), [11](#)
 - [_packetType](#)
[SerialPacket](#), [12](#)
 - [_parity](#)
[SerialPacket](#), [12](#)
 - [_sensorID](#)
[SerialPacket](#), [12](#)
- [begin](#)
[SerialPacket](#), [4](#)
- [COMMAND](#)
[defines.h](#), [13](#)
- [COMMAND_REPLY](#)
[defines.h](#), [13](#)
- [DATA_ARRAY](#)
[defines.h](#), [13](#)
- [DATA_ARRAY_REQUEST](#)
[defines.h](#), [13](#)
- [DATA_BYTE](#)
[defines.h](#), [13](#)
- [DATA_INT](#)
[defines.h](#), [13](#)
- [DATA_REQUEST](#)
[defines.h](#), [13](#)
- [defines.h](#), [12](#)
 - [COMMAND](#), [13](#)
 - [COMMAND_REPLY](#), [13](#)
 - [DATA_ARRAY](#), [13](#)
 - [DATA_ARRAY_REQUEST](#), [13](#)
 - [DATA_BYTE](#), [13](#)
 - [DATA_INT](#), [13](#)
 - [DATA_REQUEST](#), [13](#)
 - [SERIAL_ASCII](#), [13](#)
- [hexPrinting](#)
[SerialPacket](#), [4, 5](#)
- [SERIAL_ASCII](#)
[defines.h](#), [13](#)
- [sendCommand](#)
[SerialPacket](#), [5](#)
- [sendCommandReply](#)
[SerialPacket](#), [5](#)
- [sendData](#)
[SerialPacket](#), [6, 7](#)
- [sendDataArray](#)
[SerialPacket](#), [7](#)
- [sendDataArrayRequest](#)
[SerialPacket](#), [7](#)
- [sendDataRequest](#)
[SerialPacket](#), [8](#)
- [sendPacket](#)
[SerialPacket](#), [8, 9](#)
- [SerialPacket](#), [1](#)
 - [_commandID](#), [11](#)
 - [_nodeID](#), [11](#)
 - [_packetType](#), [12](#)
 - [_parity](#), [12](#)
 - [_sensorID](#), [12](#)
 - [begin](#), [4](#)
 - [hexPrinting](#), [4, 5](#)
 - [sendCommand](#), [5](#)
 - [sendCommandReply](#), [5](#)
 - [sendData](#), [6, 7](#)
 - [sendDataArray](#), [7](#)
 - [sendDataArrayRequest](#), [7](#)
 - [sendDataRequest](#), [8](#)
 - [sendPacket](#), [8, 9](#)
 - [SerialPacket](#), [4](#)
 - [SerialPacket](#), [4](#)
 - [setCommandID](#), [9](#)
 - [setNodeID](#), [10](#)
 - [setPacketType](#), [10](#)
 - [setSensorID](#), [11](#)
- [SerialPacket.cpp](#), [13](#)
- [SerialPacket.h](#), [14](#)
- [setCommandID](#)
[SerialPacket](#), [9](#)
- [setNodeID](#)
[SerialPacket](#), [10](#)
- [setPacketType](#)
[SerialPacket](#), [10](#)
- [setSensorID](#)
[SerialPacket](#), [11](#)