

## Arduino\_Serial\_Communication\_Library

Generated by Doxygen 1.8.1.2

Wed Nov 14 2012 19:29:17

## Contents

<b>1</b>	<b><a href="#">Class Index</a></b>	<b>1</b>
1.1	<a href="#">Class List</a>	1
<b>2</b>	<b><a href="#">File Index</a></b>	<b>1</b>
2.1	<a href="#">File List</a>	1
<b>3</b>	<b><a href="#">Class Documentation</a></b>	<b>1</b>
3.1	<a href="#">SerialPacket Class Reference</a>	1
3.1.1	<a href="#">Detailed Description</a>	3
3.1.2	<a href="#">Constructor &amp; Destructor Documentation</a>	3
3.1.3	<a href="#">Member Function Documentation</a>	3
3.1.4	<a href="#">Member Data Documentation</a>	9
<b>4</b>	<b><a href="#">File Documentation</a></b>	<b>9</b>
4.1	<a href="#">defines.h File Reference</a>	10
4.1.1	<a href="#">Macro Definition Documentation</a>	10
4.2	<a href="#">SerialPacket.cpp File Reference</a>	11
4.3	<a href="#">SerialPacket.h File Reference</a>	11

## 1 Class Index

### 1.1 Class List

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

<a href="#">SerialPacket</a>	<b>1</b>
------------------------------	----------

## 2 File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

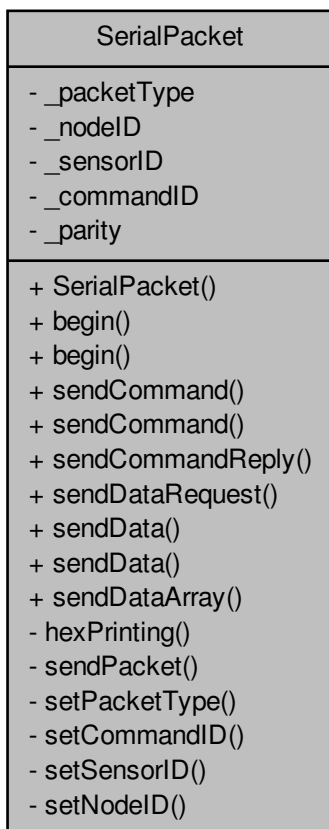
<a href="#">defines.h</a>	<b>10</b>
<a href="#">SerialPacket.cpp</a>	<b>11</b>
<a href="#">SerialPacket.h</a>	<b>11</b>

## 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 in a packet*
- void [sendCommand](#) (uint8\_t payload)  
*Send a single command in a packet, reuses commandID from previous packets*
- void [sendCommandReply](#) (uint8\_t commandID, uint8\_t payload)  
*Send a single command in a packet, reuses commandID from previous packets*
- void [sendDataRequest](#) (uint8\_t sensorID, uint8\_t payload)  
*Send a single data value in a packet*
- void [sendData](#) (uint8\_t sensorID, uint8\_t payload)  
*Send a single data value in a packet*

- void [sendData](#) (uint8\_t payload)  
*Send a single data value in a packet, //reuses sensorID from previous packets*
- 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 [hexPrinting](#) (uint8\_t data)  
*HexPrinting: helper function to print data with a constant field width (2 hex values)*
- void [sendPacket](#) (uint8\_t payload)  
*Send a single sample in a packet*
- 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*

#### 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 33 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

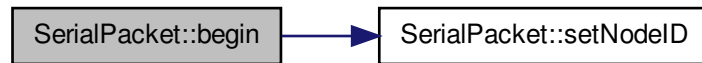
Definition at line 40 of file SerialPacket.cpp.

### 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:



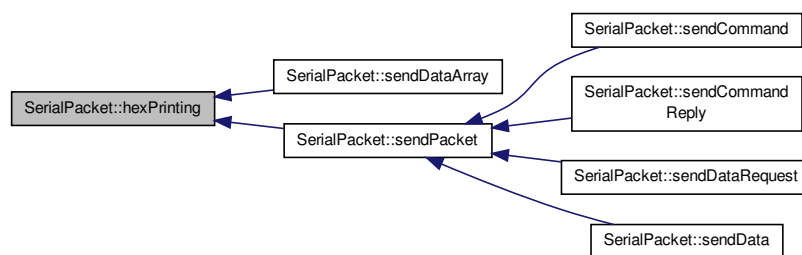
### 3.1.3.3 void SerialPacket::hexPrinting ( uint8\_t *data* ) [private]

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

Definition at line 183 of file SerialPacket.cpp.

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

Here is the caller graph for this function:

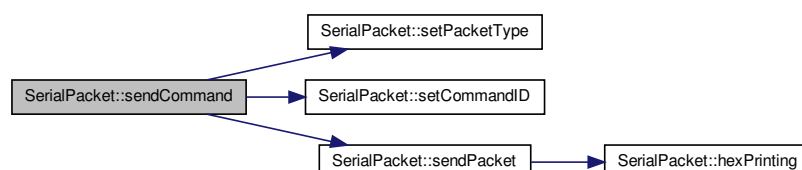


### 3.1.3.4 void SerialPacket::sendCommand ( uint8\_t *commandID*, uint8\_t *payload* )

Send a single command in a packet

Definition at line 57 of file SerialPacket.cpp.

Here is the call graph for this function:

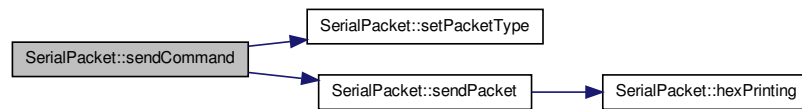


### 3.1.3.5 void SerialPacket::sendCommand ( uint8\_t *payload* )

Send a single command in a packet, reuses commandID from previous packets

Definition at line 67 of file SerialPacket.cpp.

Here is the call graph for this function:

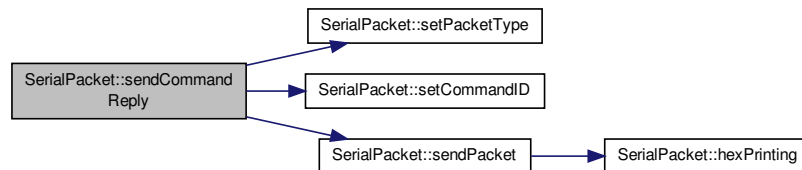


### 3.1.3.6 void SerialPacket::sendCommandReply ( uint8\_t *commandID*, uint8\_t *payload* )

Send a single command in a packet, reuses commandID from previous packets

Definition at line 76 of file SerialPacket.cpp.

Here is the call graph for this function:

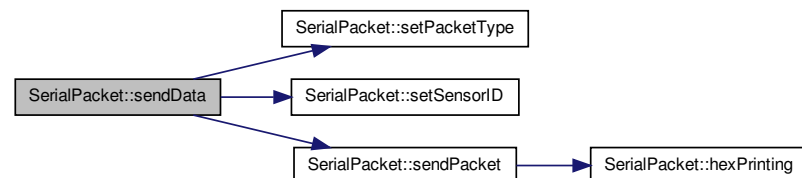


### 3.1.3.7 void SerialPacket::sendData ( uint8\_t *sensorID*, uint8\_t *payload* )

Send a single data value in a packet

Definition at line 96 of file SerialPacket.cpp.

Here is the call graph for this function:

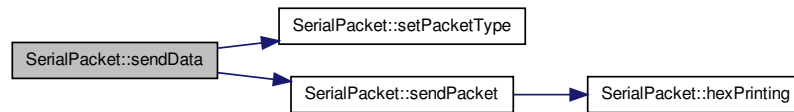


### 3.1.3.8 void SerialPacket::sendData ( uint8\_t *payload* )

Send a single data value in a packet, //reuses sensorID from previous packets

Definition at line 106 of file SerialPacket.cpp.

Here is the call graph for this function:

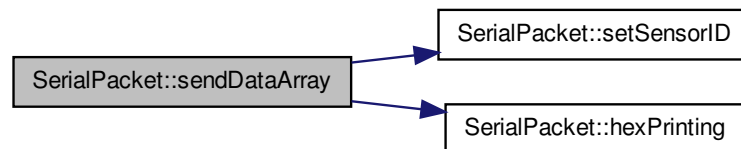


### 3.1.3.9 void SerialPacket::sendDataArray ( uint8\_t \* *dataArray*, uint8\_t *length* )

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

Definition at line 142 of file SerialPacket.cpp.

Here is the call graph for this function:

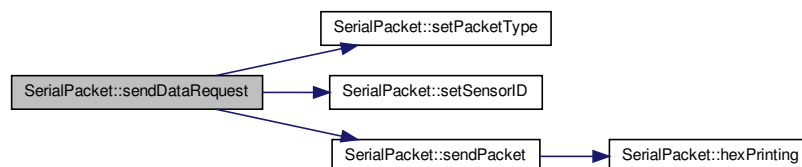


### 3.1.3.10 void SerialPacket::sendDataRequest ( uint8\_t *sensorID*, uint8\_t *payload* )

Send a single data value in a packet

Definition at line 86 of file SerialPacket.cpp.

Here is the call graph for this function:



### 3.1.3.11 void SerialPacket::sendPacket ( uint8\_t *payload* ) [private]

Send a single sample in a packet

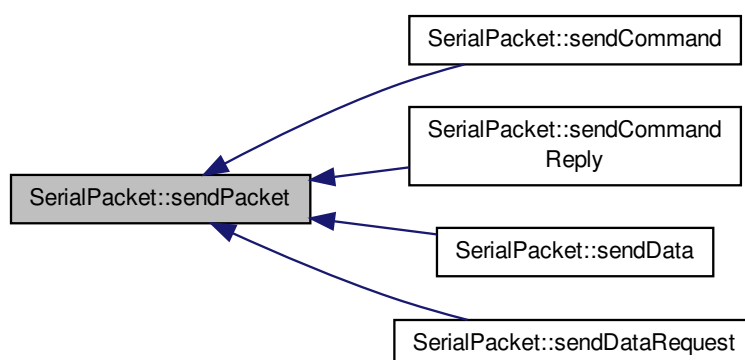
Definition at line 115 of file SerialPacket.cpp.

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

Here is the call graph for this function:



Here is the caller graph for this function:



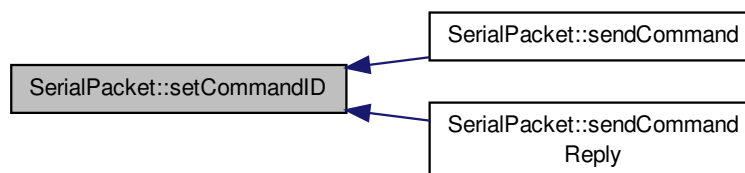
#### 3.1.3.12 void SerialPacket::setCommandID ( uint8\_t *commandID* ) [private]

Set commandID

Definition at line 167 of file `SerialPacket.cpp`.

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

Here is the caller graph for this function:



#### 3.1.3.13 void SerialPacket::setNodeID ( uint8\_t *nodeID* ) [private]

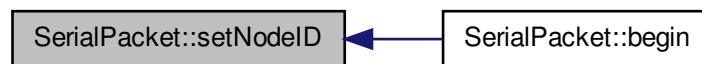
Set nodeID



Definition at line 194 of file SerialPacket.cpp.

Referenced by begin().

Here is the caller graph for this function:



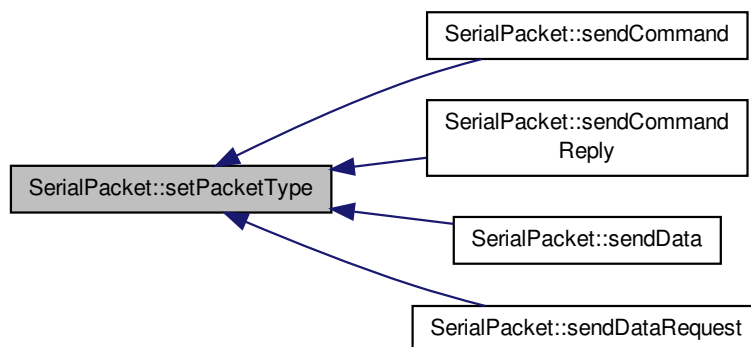
#### 3.1.3.14 void SerialPacket::setPacketType ( uint8\_t type ) [private]

Set packet type

Definition at line 175 of file SerialPacket.cpp.

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

Here is the caller graph for this function:



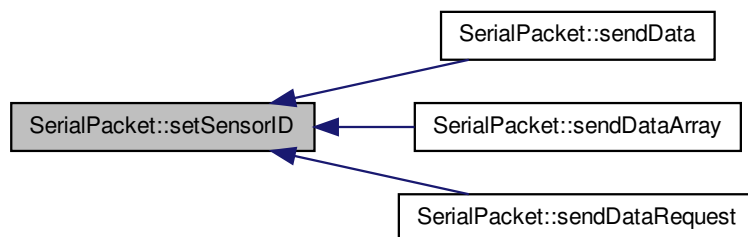
#### 3.1.3.15 void SerialPacket::setSensorID ( uint8\_t sensorID ) [private]

Set sensorID

Definition at line 202 of file SerialPacket.cpp.

Referenced by sendData(), sendDataArray(), 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 56 of file `SerialPacket.h`.

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

#### 3.1.4.2 `uint8_t SerialPacket::_nodeID` [private]

Definition at line 54 of file `SerialPacket.h`.

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

#### 3.1.4.3 `uint8_t SerialPacket::_packetType` [private]

Definition at line 53 of file `SerialPacket.h`.

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

#### 3.1.4.4 `uint8_t SerialPacket::_parity` [private]

Definition at line 57 of file `SerialPacket.h`.

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

#### 3.1.4.5 `uint8_t SerialPacket::_sensorID` [private]

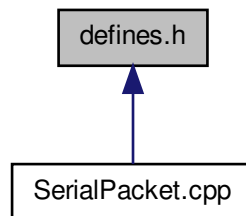
Definition at line 55 of file `SerialPacket.h`.

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 1`
- `#define COMMAND_REPLY 2`
- `#define DATA_REQUEST 3`
- `#define DATA 4`
- `#define DATA_ARRAY 5`

#### 4.1.1 Macro Definition Documentation

##### 4.1.1.1 `#define COMMAND 1`

Definition at line 6 of file defines.h.

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

##### 4.1.1.2 `#define COMMAND_REPLY 2`

Definition at line 7 of file defines.h.

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

##### 4.1.1.3 `#define DATA 4`

Definition at line 9 of file defines.h.

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

##### 4.1.1.4 `#define DATA_ARRAY 5`

Definition at line 10 of file defines.h.

##### 4.1.1.5 `#define DATA_REQUEST 3`

Definition at line 8 of file defines.h.

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

## 4.1.1.6 #define SERIAL\_ASCII

Definition at line 3 of file defines.h.

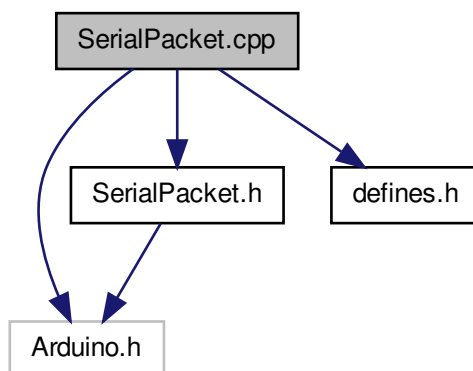
## 4.2 SerialPacket.cpp File Reference

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

```
#include "SerialPacket.h"
```

```
#include "defines.h"
```

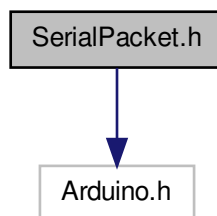
Include dependency graph for SerialPacket.cpp:



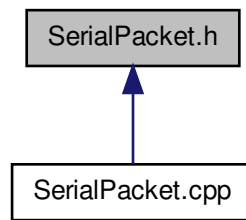
## 4.3 SerialPacket.h File Reference

```
#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](#), [9](#)
  - [\\_nodeID](#)  
[SerialPacket](#), [9](#)
  - [\\_packetType](#)  
[SerialPacket](#), [9](#)
  - [\\_parity](#)  
[SerialPacket](#), [9](#)
  - [\\_sensorID](#)  
[SerialPacket](#), [9](#)
- [begin](#)  
[SerialPacket](#), [3](#)
- [COMMAND](#)  
[defines.h](#), [10](#)
- [COMMAND\\_REPLY](#)  
[defines.h](#), [10](#)
- [DATA](#)  
[defines.h](#), [10](#)
- [DATA\\_ARRAY](#)  
[defines.h](#), [10](#)
- [DATA\\_REQUEST](#)  
[defines.h](#), [10](#)
- [defines.h](#), [10](#)
  - [COMMAND](#), [10](#)
  - [COMMAND\\_REPLY](#), [10](#)
  - [DATA](#), [10](#)
  - [DATA\\_ARRAY](#), [10](#)
  - [DATA\\_REQUEST](#), [10](#)
  - [SERIAL\\_ASCII](#), [10](#)
- [hexPrinting](#)  
[SerialPacket](#), [4](#)
- [SERIAL\\_ASCII](#)  
[defines.h](#), [10](#)
- [sendCommand](#)  
[SerialPacket](#), [4](#)
- [sendCommandReply](#)  
[SerialPacket](#), [5](#)
- [sendData](#)  
[SerialPacket](#), [5](#)
- [senddataArray](#)  
[SerialPacket](#), [6](#)
- [sendDataRequest](#)  
[SerialPacket](#), [6](#)
- [sendPacket](#)  
[SerialPacket](#), [6](#)
- [SerialPacket](#), [1](#)
  - [\\_commandID](#), [9](#)
  - [\\_nodeID](#), [9](#)
  - [\\_packetType](#), [9](#)
  - [\\_parity](#), [9](#)
  - [\\_sensorID](#), [9](#)
  - [begin](#), [3](#)
  - [hexPrinting](#), [4](#)
  - [sendCommand](#), [4](#)
  - [sendCommandReply](#), [5](#)
  - [sendData](#), [5](#)
  - [senddataArray](#), [6](#)
  - [sendDataRequest](#), [6](#)
  - [sendPacket](#), [6](#)
  - [SerialPacket](#), [3](#)
  - [SerialPacket](#), [3](#)
  - [setCommandID](#), [7](#)
  - [setNodeID](#), [7](#)
  - [setPacketType](#), [6](#)
  - [setSensorID](#), [8](#)
  - [SerialPacket.cpp](#), [11](#)
  - [SerialPacket.h](#), [11](#)
  - [setCommandID](#)  
[SerialPacket](#), [7](#)
  - [setNodeID](#)  
[SerialPacket](#), [7](#)
  - [setPacketType](#)  
[SerialPacket](#), [8](#)
  - [setSensorID](#)  
[SerialPacket](#), [8](#)