Arduino Serial Communication Library

Generated by Doxygen 1.8.1.2

Thu Feb 7 2013 16:29:43

Contents

1 Todo List 1

2 Class Index 1

2.1 Class List . . . 1

3 File Index 1

3.1 File List . . . 1

4 Class Documentation 1

4.1 SerialPacket::packet Struct Reference . . . 2

4.1.1 Detailed Description . . . 2

4.1.2 Member Data Documentation . . . 2

4.2 SerialPacket Class Reference . . . 3

4.2.1 Detailed Description . . . 6

4.2.2 Constructor & Destructor Documentation . . . 6

4.2.3 Member Function Documentation . . . 6

4.2.4 Member Data Documentation . . . 15

5 File Documentation 16

5.1 defines.h File Reference . . . 17

5.1.1 Macro Definition Documentation . . . 17

5.2 SerialPacket.cpp File Reference . . . 19

5.2.1 Macro Definition Documentation . . . 20

5.3 SerialPacket.h File Reference . . . 20

5.4 SerialPacket_before check.cpp File Reference . . . 21

5.4.1 Macro Definition Documentation . . . 22

1 Todo List

Member SerialPacket::parseSerialData ()

: error handling: illegal payloads is now handled by char position (G2 -> 02, 2G -> 20, GG -> 00), requires hextodec reimplementation

: error handling: illegal payloads is now handled by char position (G2 -> 02, 2G -> 20, GG -> 00), requires hextodec reimplementation

Member SerialPacket::readSerialData ()

: used detect new packets (for command reply), needs to be merged with the (currenty non-existing packet validation code)

: used detect new packets (for command reply), needs to be merged with the (currenty non-existing packet validation code)

2 Class Index

2.1 Class List

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

SerialPacket::packet 2

SerialPacket 3

3 File Index

3.1 File List

Here is a list of all files with brief descriptions:

defines.h 17

SerialPacket.cpp 19

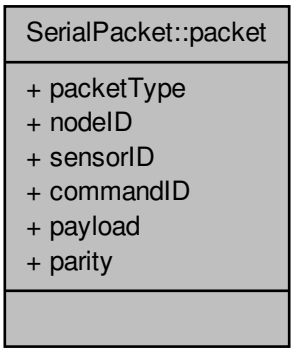
SerialPacket.h 20

SerialPacket_before check.cpp 21

4 Class Documentation

4.1 SerialPacket::packet Struct Reference

Collaboration diagram for SerialPacket::packet:



Public Attributes

- uint8_t packetType
- uint8_t nodeID
- uint8_t sensorID
- uint8_t commandID
- uint8_t payload

- uint8_t parity

4.1.1 Detailed Description

Definition at line 65 of file SerialPacket.h.

4.1.2 Member Data Documentation

4.1.2.1 uint8_t SerialPacket::packet::commandID

Definition at line 70 of file SerialPacket.h.

Referenced by SerialPacket::getCommandID(), SerialPacket::parseSerialData(), SerialPacket::printInfo(), and SerialPacket::SerialPacket().

4.1.2.2 uint8_t SerialPacket::packet::nodeID

Definition at line 68 of file SerialPacket.h.

Referenced by SerialPacket::parseSerialData(), SerialPacket::printInfo(), and SerialPacket::SerialPacket().

4.1.2.3 uint8_t SerialPacket::packet::packetType

Definition at line 67 of file SerialPacket.h.

Referenced by SerialPacket::parseSerialData(), SerialPacket::printInfo(), and SerialPacket::SerialPacket().

4.1.2.4 uint8_t SerialPacket::packet::parity

Definition at line 72 of file SerialPacket.h.

Referenced by SerialPacket::checkParity(), SerialPacket::parseSerialData(), SerialPacket::printInfo(), and SerialPacket::SerialPacket().

4.1.2.5 uint8_t SerialPacket::packet::payload

Definition at line 71 of file SerialPacket.h.

Referenced by SerialPacket::getPayload(), SerialPacket::parseSerialData(), SerialPacket::printInfo(), and SerialPacket::SerialPacket().

4.1.2.6 uint8_t SerialPacket::packet::sensorID

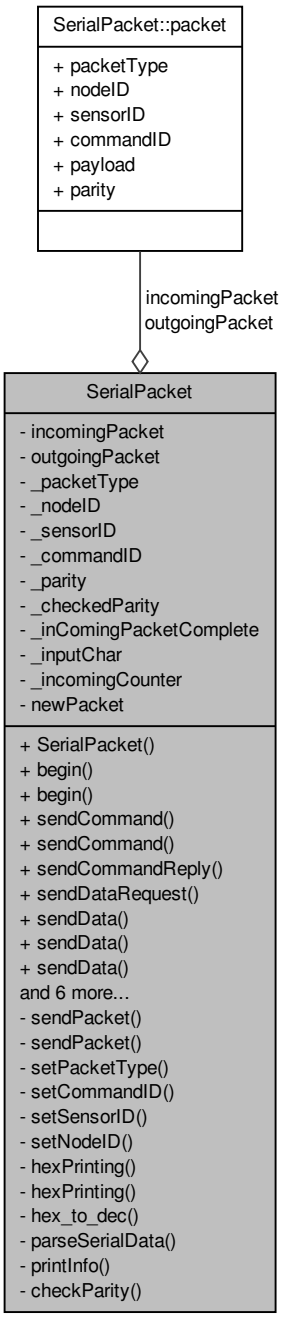
Definition at line 69 of file SerialPacket.h.

Referenced by SerialPacket::parseSerialData(), SerialPacket::printInfo(), and SerialPacket::SerialPacket().

4.2 SerialPacket Class Reference

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

Collaboration diagram for SerialPacket:



Classes

- struct packet

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
- boolean readSerialData ()

  Set readSerialData
- uint8_t getCommandID ()

  Get commandID
- uint8_t getPayload ()

  Get getPayload

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)

- uint8_t hex_to_dec (uint8_t in)
- boolean parseSerialData ()

  Set parseSerialData
- void printInfo ()

  printInfo:
- boolean checkParity ()

  Convert HEX to Decimal

Private Attributes

- struct SerialPacket::packet incomingPacket
- struct SerialPacket::packet outgoingPacket
- uint8_t _packetType
- uint8_t _nodeID
- uint8_t _sensorID
- uint8_t _commandID
- uint8_t _parity
- uint8_t _checkedParity
- boolean _inComingPacketComplete
- char _inputChar [20]
- uint8_t _incomingCounter
- boolean newPacket

4.2.1 Detailed Description

Definition at line 38 of file SerialPacket.h.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 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.

4.2.3 Member Function Documentation

4.2.3.1 void SerialPacket::begin ( )

Begin using default settings:

- speed: 115200 baud
- nodeID: 0

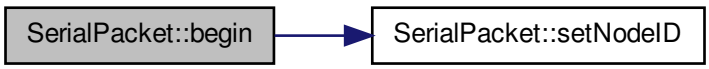
Definition at line 49 of file SerialPacket.cpp.

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

Begin using custom settings

Definition at line 57 of file SerialPacket.cpp.

Here is the call graph for this function:



4.2.3.3 boolean SerialPacket::checkParity ( ) [private]

Convert HEX to Decimal

Convert HEX to DecimalT12N00I00P1CQ0E

Definition at line 399 of file SerialPacket.cpp.

4.2.3.4 uint8_t SerialPacket::getCommandID ( )

Get commandID

Definition at line 412 of file SerialPacket.cpp.

4.2.3.5 uint8_t SerialPacket::getPayload ( )

Get getPayload

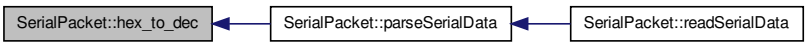
Definition at line 420 of file SerialPacket.cpp.

4.2.3.6 uint8_t SerialPacket::hex_to_dec (uint8_t *in*) [private]

Definition at line 365 of file SerialPacket.cpp.

Referenced by parseSerialData().

Here is the caller graph for this function:



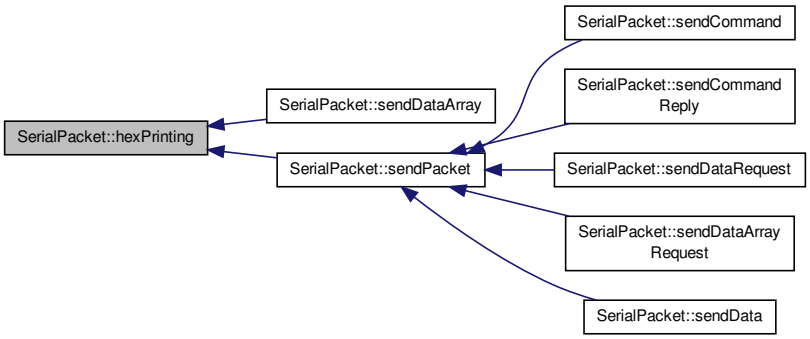
4.2.3.7 void SerialPacket::hexPrinting (uint8_t & *data*) [private]

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

Definition at line 245 of file SerialPacket.cpp.

Referenced by sendDataArray(), and sendPacket().

Here is the caller graph for this function:



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

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

Definition at line 256 of file SerialPacket.cpp.

4.2.3.9 boolean SerialPacket::parseSerialData ( ) [private]

Set parseSerialData

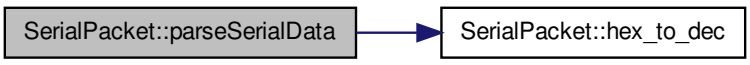
Todo : error handling: illegal payloads is now handled by char position (G2 -> 02, 2G -> 20, GG -> 00), requires hextodec reimplementation

Todo : error handling: illegal payloads is now handled by char position (G2 -> 02, 2G -> 20, GG -> 00), requires hextodec reimplementation

Definition at line 315 of file SerialPacket.cpp.

Referenced by readSerialData().

Here is the call graph for this function:



Here is the caller graph for this function:

4.2.3.10 void SerialPacket::printInfo ( ) [private]

printInfo:

Definition at line 376 of file SerialPacket.cpp.

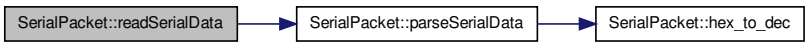
4.2.3.11 boolean SerialPacket::readSerialData ( )

Set readSerialData

Todo : used detect new packets (for command reply), needs to be merged with the (currenty non-existing packet validation code)

Definition at line 289 of file SerialPacket.cpp.

Here is the call graph for this function:

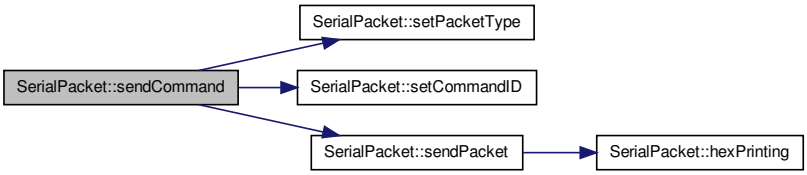


4.2.3.12 void SerialPacket::sendCommand (uint8_t *commandID,* uint8_t *payload*)

Send a single command

Definition at line 68 of file SerialPacket.cpp.

Here is the call graph for this function:

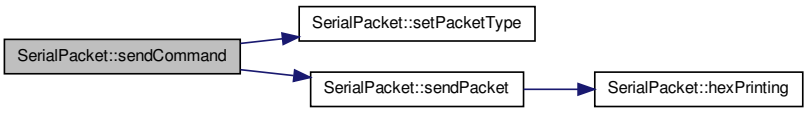


4.2.3.13 void SerialPacket::sendCommand (uint8_t *payload*)

Send a single command, reuses commandID from previous packets

Definition at line 78 of file SerialPacket.cpp.

Here is the call graph for this function:

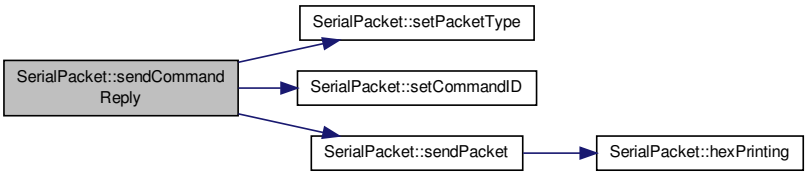


4.2.3.14 void SerialPacket::sendCommandReply (uint8_t *commandID,* uint8_t *payload*)

Send a reply to a command

Definition at line 87 of file SerialPacket.cpp.

Here is the call graph for this function:



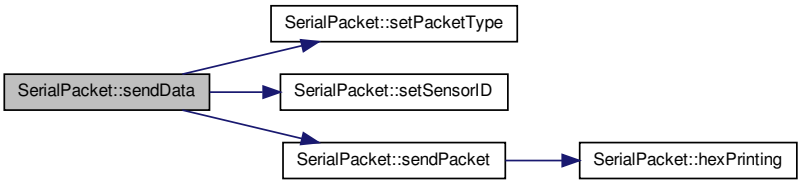
4.2.3.15 void SerialPacket::sendData (uint8_t *sensorID,* uint8_t *payload*)

Send a single data value

T12N00I00P00Q12

Definition at line 117 of file SerialPacket.cpp.

Here is the call graph for this function:

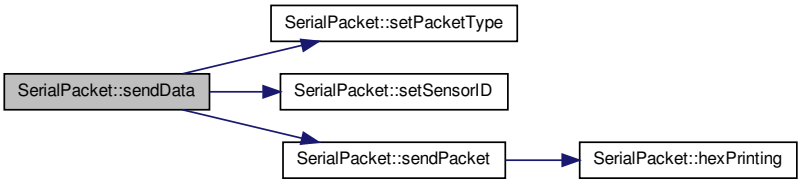


4.2.3.16 void SerialPacket::sendData (uint8_t *sensorID,* int16_t *payload*)

Send a single data value

Definition at line 127 of file SerialPacket.cpp.

Here is the call graph for this function:

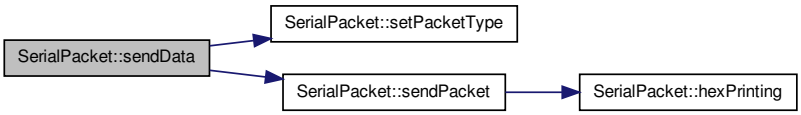


4.2.3.17 void SerialPacket::sendData (uint8_t *payload*)

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

Definition at line 137 of file SerialPacket.cpp.

Here is the call graph for this function:

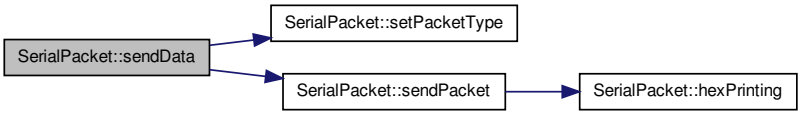


4.2.3.18 void SerialPacket::sendData (int16_t *payload*)

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

Definition at line 146 of file SerialPacket.cpp.

Here is the call graph for this function:



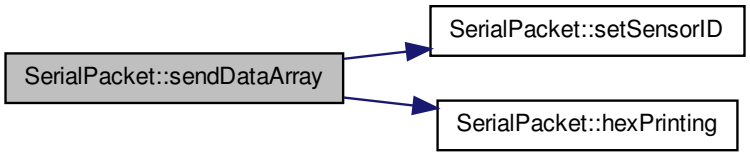
4.2.3.19 void SerialPacket::sendDataArray (uint8_t * *dataArray*, uint8_t *length*)

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

T12N00I00P1CQ0E

Definition at line 205 of file SerialPacket.cpp.

Here is the call graph for this function:

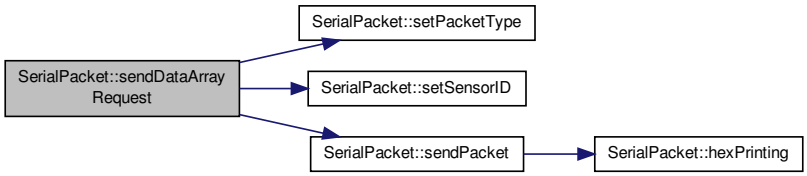


4.2.3.20 void SerialPacket::sendDataArrayRequest (uint8_t *arrayID*, uint8_t *payload*)

Request an array of data values

Definition at line 107 of file SerialPacket.cpp.

Here is the call graph for this function:

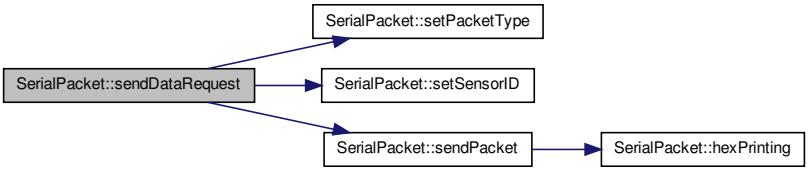


4.2.3.21 void SerialPacket::sendDataRequest (uint8_t *sensorID,* uint8_t *payload*)

Request a single data value

Definition at line 97 of file SerialPacket.cpp.

Here is the call graph for this function:



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

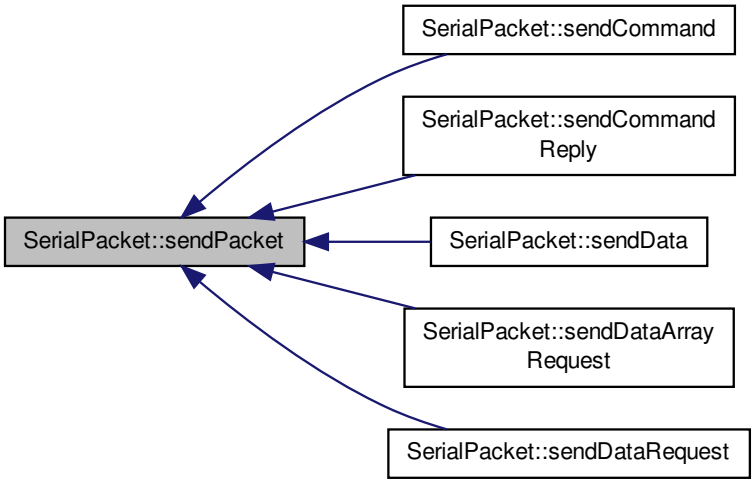
Send out the actual 8-bit data packet (called from other 'send' functions)

Definition at line 155 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:



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

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

Definition at line 180 of file SerialPacket.cpp.

Here is the call graph for this function:

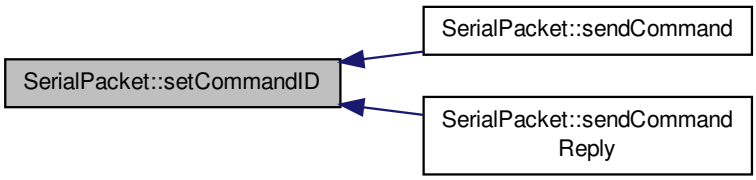
4.2.3.24 void SerialPacket::setCommandID (uint8_t & *commandID*) [private]

Set commandID

Definition at line 229 of file SerialPacket.cpp.

Referenced by sendCommand(), and sendCommandReply().

Here is the caller graph for this function:



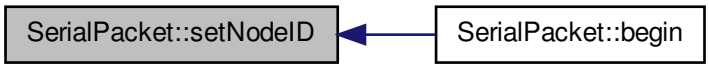
4.2.3.25 void SerialPacket::setNodeID (uint8_t & *nodeID*) [private]

Set nodeID

Definition at line 273 of file SerialPacket.cpp.

Referenced by begin().

Here is the caller graph for this function:



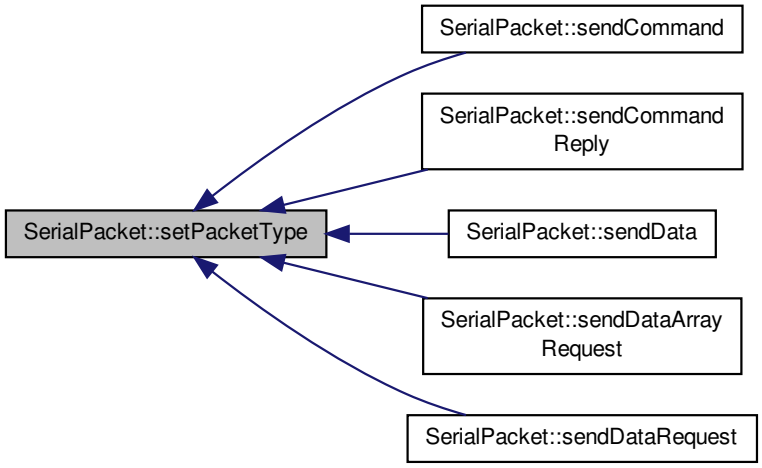
4.2.3.26 void SerialPacket::setPacketType (uint8_t *type*) [private]

Set packet type

Definition at line 237 of file SerialPacket.cpp.

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

Here is the caller graph for this function:



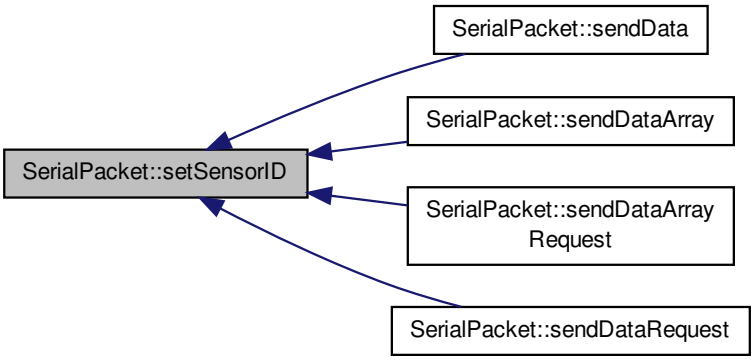
4.2.3.27 void SerialPacket::setSensorID (uint8_t & *sensorID*) [private]

Set sensorID

Definition at line 281 of file SerialPacket.cpp.

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

Here is the caller graph for this function:



4.2.4 Member Data Documentation

4.2.4.1 uint8_t SerialPacket::_checkedParity [private]

Definition at line 80 of file SerialPacket.h.

Referenced by checkParity(), parseSerialData(), and printInfo().

4.2.4.2 uint8_t SerialPacket::_commandID [private]

Definition at line 78 of file SerialPacket.h.

Referenced by sendPacket(), and setCommandID().

4.2.4.3 uint8_t SerialPacket::_incomingCounter [private]

Definition at line 84 of file SerialPacket.h.

Referenced by readSerialData(), and SerialPacket().

4.2.4.4 boolean SerialPacket::_inComingPacketComplete [private]

Definition at line 82 of file SerialPacket.h.

Referenced by SerialPacket().

4.2.4.5 char SerialPacket::_inputChar[20] [private]

Definition at line 83 of file SerialPacket.h.

Referenced by parseSerialData(), and readSerialData().

4.2.4.6 uint8_t SerialPacket::_nodeID [private]

Definition at line 76 of file SerialPacket.h.

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

4.2.4.7 uint8_t SerialPacket::_packetType [private]

Definition at line 75 of file SerialPacket.h.

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

4.2.4.8 uint8_t SerialPacket::_parity [private]

Definition at line 79 of file SerialPacket.h.

Referenced by sendDataArray(), and sendPacket().

4.2.4.9 uint8_t SerialPacket::_sensorID [private]

Definition at line 77 of file SerialPacket.h.

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

4.2.4.10 struct SerialPacket::packet SerialPacket::incomingPacket [private]

Referenced by checkParity(), getCommandID(), getPayload(), parseSerialData(), printInfo(), and SerialPacket().

4.2.4.11 boolean SerialPacket::newPacket [private]

Definition at line 98 of file SerialPacket.h.

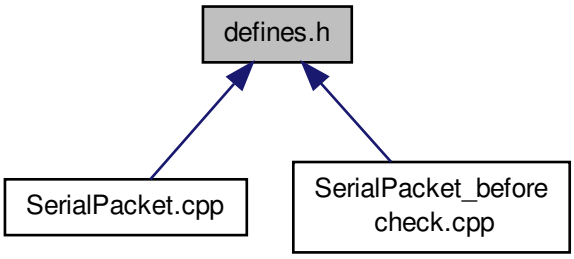
Referenced by parseSerialData(), readSerialData(), and SerialPacket().

4.2.4.12 struct SerialPacket::packet SerialPacket::outgoingPacket [private]

5 File Documentation

5.1 defines.h File Reference

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



Macros

- #define SERIAL_ASCII
- #define DEFAULT_BAUDRATE 115200
- #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
- #define TEMPERATURE 0x10

  Sensor Types:
- #define HUMIDITY 0x11
- #define DISTANCE 0x30
- #define MOTORSTATUS 0x50
- #define STOP_MOTOR_A 0x10

  Command IDs.
- #define START_MOTOR_A 0x11
- #define SET_SPEED_MOTOR_A 0x12
- #define BRAKE_MOTOR_A 0x13
- #define STOP_MOTOR_B 0x15
- #define START_MOTOR_B 0x16
- #define SET_SPEED_MOTOR_B 0x17
- #define BRAKE_MOTOR_B 0x18

5.1.1 Macro Definition Documentation

5.1.1.1 #define BRAKE_MOTOR_A 0x13

Definition at line 34 of file defines.h.

5.1.1.2 #define BRAKE_MOTOR_B 0x18

Definition at line 39 of file defines.h.

5.1.1.3 #define COMMAND 0x01

Definition at line 11 of file defines.h.

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

5.1.1.4 #define COMMAND_REPLY 0x02

Definition at line 12 of file defines.h.

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

5.1.1.5 #define DATA_ARRAY 0x22

Definition at line 19 of file defines.h.

5.1.1.6 #define DATA_ARRAY_REQUEST 0x21

Definition at line 18 of file defines.h.

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

5.1.1.7 #define DATA_BYTE 0x12

Definition at line 15 of file defines.h.

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

5.1.1.8 #define DATA_INT 0x13

Definition at line 16 of file defines.h.

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

5.1.1.9 #define DATA_REQUEST 0x11

Definition at line 14 of file defines.h.

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

5.1.1.10 #define DEFAULT_BAUDRATE 115200

Definition at line 9 of file defines.h.

Referenced by SerialPacket::begin().

5.1.1.11 #define DISTANCE 0x30

Definition at line 26 of file defines.h.

5.1.1.12 #define HUMIDITY 0x11

Definition at line 24 of file defines.h.

5.1.1.13 #define MOTORSTATUS 0x50

Definition at line 27 of file defines.h.

5.1.1.14 #define SERIAL_ASCII

Definition at line 3 of file defines.h.

5.1.1.15 #define SET_SPEED_MOTOR_A 0x12

Definition at line 33 of file defines.h.

5.1.1.16 #define SET_SPEED_MOTOR_B 0x17

Definition at line 38 of file defines.h.

5.1.1.17 #define START_MOTOR_A 0x11

Definition at line 32 of file defines.h.

5.1.1.18 #define START_MOTOR_B 0x16

Definition at line 37 of file defines.h.

5.1.1.19 #define STOP_MOTOR_A 0x10

Command IDs.

Definition at line 31 of file defines.h.

5.1.1.20 #define STOP_MOTOR_B 0x15

Definition at line 36 of file defines.h.

5.1.1.21 #define TEMPERATURE 0x10

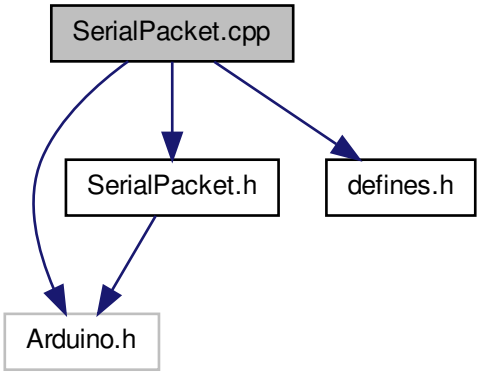
Sensor Types:

Definition at line 23 of file defines.h.

5.2 SerialPacket.cpp File Reference

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

Include dependency graph for SerialPacket.cpp:



Macros

- #define HEX_DEC_ERROR 42

  Convert HEX to Decimal
- #define HEX_DEC_ERROR 42

  Convert HEX to Decimal

5.2.1 Macro Definition Documentation

5.2.1.1 #define HEX_DEC_ERROR 42

Convert HEX to Decimal

Definition at line 364 of file SerialPacket.cpp.

Referenced by SerialPacket::hex_to_dec().

5.2.1.2 #define HEX_DEC_ERROR 42

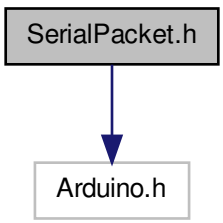
Convert HEX to Decimal

Definition at line 364 of file SerialPacket.cpp.

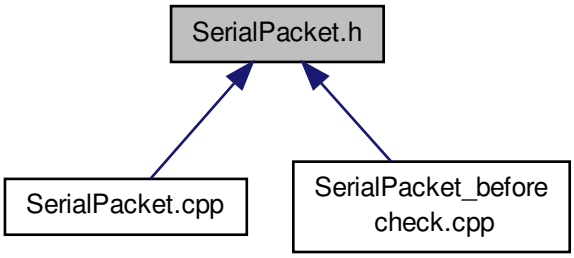
5.3 SerialPacket.h File Reference

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

Include dependency graph for SerialPacket.h:



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



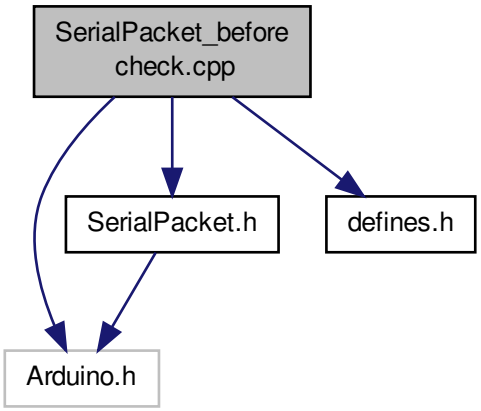
Classes

- class SerialPacket
- struct SerialPacket::packet

5.4 SerialPacket_before check.cpp File Reference

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

Include dependency graph for SerialPacket_before check.cpp:



Macros

- #define HEX_DEC_ERROR 0

  Convert HEX to Decimal

5.4.1 Macro Definition Documentation

5.4.1.1 #define HEX_DEC_ERROR 0

Convert HEX to Decimal

Definition at line 347 of file SerialPacket_before check.cpp.

Index

_checkedParity
SerialPacket, 15
_commandID
SerialPacket, 15
_inComingPacketComplete
SerialPacket, 16
_incomingCounter
SerialPacket, 16
_inputChar
SerialPacket, 16
_nodeID
SerialPacket, 16
_packetType
SerialPacket, 16
_parity
SerialPacket, 16
_sensorID
SerialPacket, 16

BRAKE_MOTOR_A
defines.h, 17
BRAKE_MOTOR_B
defines.h, 17
begin
SerialPacket, 6

COMMAND
defines.h, 17
COMMAND_REPLY
defines.h, 18
checkParity
SerialPacket, 7
commandID
SerialPacket::packet, 2

DATA_ARRAY
defines.h, 18
DATA_ARRAY_REQUEST
defines.h, 18
DATA_BYTE
defines.h, 18
DATA_INT
defines.h, 18
DATA_REQUEST
defines.h, 18
DEFAULT_BAUDRATE
defines.h, 18
DISTANCE
defines.h, 18
defines.h, 17
BRAKE_MOTOR_A, 17
BRAKE_MOTOR_B, 17
COMMAND, 17
COMMAND_REPLY, 18
DATA_ARRAY, 18
DATA_ARRAY_REQUEST, 18
DATA_BYTE, 18
DATA_INT, 18
DATA_REQUEST, 18
DEFAULT_BAUDRATE, 18
DISTANCE, 18
HUMIDITY, 18
MOTORSTATUS, 18
SERIAL_ASCII, 18
SET_SPEED_MOTOR_A, 18
SET_SPEED_MOTOR_B, 19
START_MOTOR_A, 19
START_MOTOR_B, 19
STOP_MOTOR_A, 19
STOP_MOTOR_B, 19
TEMPERATURE, 19

getCommandID
SerialPacket, 7
getPayload
SerialPacket, 7

HEX_DEC_ERROR
SerialPacket.cpp, 20
SerialPacket_before check.cpp, 22
HUMIDITY
defines.h, 18
hex_to_dec
SerialPacket, 7
hexPrinting
SerialPacket, 7, 8

incomingPacket
SerialPacket, 16

MOTORSTATUS
defines.h, 18

newPacket
SerialPacket, 16
nodeID
SerialPacket::packet, 2

outgoingPacket
SerialPacket, 16

packetType
SerialPacket::packet, 2
parity
SerialPacket::packet, 2
parseSerialData
SerialPacket, 8
payload
SerialPacket::packet, 3
printInfo
SerialPacket, 9

readSerialData

SerialPacket, 9

SERIAL_ASCII
defines.h, 18
SET_SPEED_MOTOR_A
defines.h, 18
SET_SPEED_MOTOR_B
defines.h, 19
START_MOTOR_A
defines.h, 19
START_MOTOR_B
defines.h, 19
STOP_MOTOR_A
defines.h, 19
STOP_MOTOR_B
defines.h, 19
sendCommand
SerialPacket, 9
sendCommandReply
SerialPacket, 9
sendData
SerialPacket, 10, 11
sendDataArray
SerialPacket, 11
sendDataArrayRequest
SerialPacket, 11
sendDataRequest
SerialPacket, 12
sendPacket
SerialPacket, 12, 13
sensorID
SerialPacket::packet, 3
SerialPacket, 3
_checkedParity, 15
_commandID, 15
_inComingPacketComplete, 16
_incomingCounter, 16
_inputChar, 16
_nodeID, 16
_packetType, 16
_parity, 16
_sensorID, 16
begin, 6
checkParity, 7
getCommandID, 7
getPayload, 7
hex_to_dec, 7
hexPrinting, 7, 8
incomingPacket, 16
newPacket, 16
outgoingPacket, 16
parseSerialData, 8
printInfo, 9
readSerialData, 9
sendCommand, 9
sendCommandReply, 9
sendData, 10, 11
sendDataArray, 11
sendDataArrayRequest, 11
sendDataRequest, 12
sendPacket, 12, 13
SerialPacket, 6
SerialPacket, 6
setCommandID, 13
setNodeID, 14
setPacketType, 14
setSensorID, 15
SerialPacket.cpp, 19
HEX_DEC_ERROR, 20
SerialPacket.h, 20
SerialPacket::packet, 2
commandID, 2
nodeID, 2
packetType, 2
parity, 2
payload, 3
sensorID, 3
SerialPacket_before check.cpp, 21
HEX_DEC_ERROR, 22
setCommandID
SerialPacket, 13
setNodeID
SerialPacket, 14
setPacketType
SerialPacket, 14
setSensorID
SerialPacket, 15

TEMPERATURE
defines.h, 19