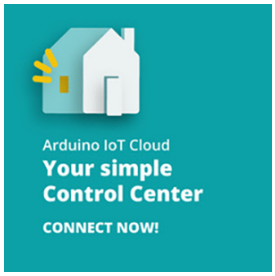
LANGUAGE
FUNCTIONS
VARIABLES
STRUCTURE
LIBRARIES
IOT CLOUD API
GLOSSARY



Last Revision: 2019/09/03

Last Build: 2021/10/13

EDIT THIS PAGE

Serial.write()

Description

Writes binary data to the serial port. This data is sent as a byte or series of bytes; to send the characters representing the digits of a number use the print() function instead.

Syntax

```
Serial.write(val)
Serial.write(str)
Serial.write(buf, len)
```

Parameters

`Serial`: serial port object. See the list of available serial ports for each board on the Serial main page.

`val`: a value to send as a single byte.

`str`: a string to send as a series of bytes.

`buf`: an array to send as a series of bytes.

`len`: the number of bytes to be sent from the array.

Returns

`write()` will return the number of bytes written, though reading that number is optional. Data type: `size_`

Example Code

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.write(45); // send a byte with the value 45

  int bytesSent = Serial.write("hello");  //send the string "hello" and return the length of the string
}
```

Notes and Warnings

As of Arduino IDE 1.0, serial transmission is asynchronous. If there is enough empty space in the transmi buffer, `Serial.write()` will return before any characters are transmitted over serial. If the transmit buffer full then `Serial.write()` will block until there is enough space in the buffer. To avoid blocking calls to `Serial.write()`, you can first check the amount of free space in the transmit buffer using availableForWri