Serial.print()

Description

Prints data to the serial port as human-readable ASCII text. This command can take many forms. Numbers are printed using an ASCII character for each digit. Floats are similarly printed as ASCII digits, defaulting to two decimal places. Bytes are sent as a single character. Characters and strings are sent as is. For example-

- `Serial.print(78)` gives "78"
- `Serial.print(1.23456)` gives "1.23"
- `Serial.print('N')` gives "N"
- `Serial.print("Hello world.")` gives "Hello world."

An optional second parameter specifies the base (format) to use; permitted values are `BIN(binary, or base 2)`, `OCT(octal, or base 8)`, `DEC(decimal, or base 10)`, `HEX(hexadecimal, or base 16)`. For floating point numbers, this parameter specifies the number of decimal places to use. For example-

- `Serial.print(78, BIN)` gives "1001110"
- `Serial.print(78, OCT)` gives "116"
- `Serial.print(78, DEC)` gives "78"
- `Serial.print(78, HEX)` gives "4E"
- `Serial.print(1.23456, 0)` gives "1"
- `Serial.print(1.23456, 2)` gives "1.23"
- `Serial.print(1.23456, 4)` gives "1.2345"

You can pass flash-memory based strings to `Serial.print()` by wrapping them with F(). For example:

```
Serial.print(F("Hello World"))
```

To send data without conversion to its representation as characters, use Serial.writ

Syntax

```
Serial.print(val)
Serial.print(val, format)
```

Serial main page.

`val`: the value to print. Allowed data types: any data type.

Returns

`print()` returns the number of bytes written, though reading that number is option

Data type: `size_t`.

Example Code

```
/*
  Uses a for loop to print numbers in various formats.
*/
void setup() {
  Serial.begin(9600); // open the serial port at 9600 bps:
}

void loop() {
  // print labels
  Serial.print("NO FORMAT");  // prints a label
  Serial.print("\t");         // prints a tab

  Serial.print("DEC");
  Serial.print("\t");

  Serial.print("HEX");
  Serial.print("\t");

  Serial.print("OCT");
  Serial.print("\t");

  Serial.print("BIN");
  Serial.println();         // carriage return after the last label

  for (int x = 0; x < 64; x++) { // only part of the ASCII chart, change to suit
    // print it out in many formats:
    Serial.print(x);       // print as an ASCII-encoded decimal - same as "DEC"
    Serial.print("\t\t");  // prints two tabs to accomodate the label lenght

    Serial.print(x, DEC);  // print as an ASCII-encoded decimal
    Serial.print("\t");    // prints a tab

    Serial.print(x, HEX);  // print as an ASCII-encoded hexadecimal
    Serial.print("\t");    // prints a tab

    Serial.print(x, OCT);  // print as an ASCII-encoded octal
    Serial.print("\t");    // prints a tab

    Serial.println(x, BIN);  // print as an ASCII-encoded binary
    // then adds the carriage return with "println"
    delay(200);            // delay 200 milliseconds
  }
  Serial.println();        // prints another carriage return
}
```

Notes and Warnings

For information on the asyncronicity of *`Serial`*`.print()`, see the Notes and Warnings section of the Serial.write() reference page.

See also

LANGUAGE begin()

LANGUAGE
FUNCTIONS
VARIABLES
STRUCTURE
LIBRARIES
IOT CLOUD API
GLOSSARY

Last Revision: 2019/12/09

Last Build: 2021/04/15

EDIT THIS PAGE

LANGUAGE peek()

LANGUAGE flush()

LANGUAGE println()

LANGUAGE write()

LANGUAGE SerialEvent()

LANGUAGE Memory

Back to top