

SoftwareSerial Library

The SoftwareSerial library allows serial communication on other digital pins of an Arduino board.

👤AUTHOR: **Arduino**

📅LAST REVISION: **09/14/2022, 03:09 PM**

The SoftwareSerial library allows serial communication on other digital pins of an Arduino board, using software to replicate the functionality (hence the name "SoftwareSerial"). It is possible to have multiple software serial ports with speeds up to 115200 bps. A parameter enables inverted signaling for devices which require that protocol.

The version of SoftwareSerial included in 1.0 and later is based on the [NewSoftSerial library](#) by 'Mikal Hart'.

To use this library:

COPY

```
1 #include <SoftwareSerial.h>
```

Limitations of This Library

SoftwareSerial library has the following known limitations:

- It cannot transmit and receive data at the same time.

- If using multiple software serial ports, only one can receive data at a time.

- Not all pins on the Mega and Mega 2560 boards support change interrupts, so only the following can be used for RX: 10, 11, 12, 13, 14, 15, 50, 51, 52, 53, A8 (62), A9 (63), A10 (64), A11 (65), A12 (66), A13 (67), A14 (68), A15 (69). Not all pins on the Leonardo and Micro boards support change interrupts, so only the following can be used for RX: 8, 9, 10, 11, 14 (MISO), 15 (SCK), 16 (MOSI).

- On Arduino or Genuino 101 boards the current maximum RX speed is 57600bps.

- On Arduino or Genuino 101 boards RX doesn't work on digital pin 13.

If your project requires simultaneous data flows, see Paul Stoffregen's [AltSoftSerial library](#).

Examples

[SoftwareSerial example](#): sometimes one serial port just isn't enough!

[Two port receive](#): Work with multiple software serial ports.

Methods

SoftwareSerial()

Create an instance of a SoftwareSerial object. Multiple SoftwareSerial objects may be created, however only one can be active at a given moment.

Syntax

[COPY](#)

```
1 SoftwareSerial(rxPin, txPin, inverse_logic)
```

Parameters

rxPin: the pin on which to receive serial data.

txPin: the pin on which to transmit serial data.

inverse_logic: used to invert the sense of incoming bits (the default is normal logic). If set, SoftwareSerial treats a LOW (0v on the pin, normally) on the RX pin as a 1-bit (the idle state) and a HIGH (5V on the pin, normally) as a 0-bit. It also affects the way that it writes to the TX pin. Default value is false.

Returns

None.

Example

[COPY](#)

```
1 #include <SoftwareSerial.h>
2
3 const byte rxPin = 2;
4 const byte txPin = 3;
5
6 // Set up a new SoftwareSerial object
```

```
7 SoftwareSerial mySerial (rxPin, txPin);
```

See also

- `available()`
- `begin()`
- `isListening()`
- `overflow()`
- `peek()`
- `read()`
- `print()`
- `println()`
- `listen()`
- `write()`

`available()`

Get the number of bytes (characters) available for reading from a software serial port. This is data that has already arrived and stored in the serial receive buffer.

Syntax

[COPY](#)

```
1 mySerial.available()
```

Parameters

None.

Returns

The number of bytes available to read.

Example

```
1  #include <SoftwareSerial.h>
2
3  #define rxPin 10
4  #define txPin 11
5
6  // Set up a new SoftwareSerial object
7  SoftwareSerial mySerial = SoftwareSerial(rxPin, txPin);
8
9  void setup() {
10     // Define pin modes for TX and RX
11     pinMode(rxPin, INPUT);
12     pinMode(txPin, OUTPUT);
13
14     // Set the baud rate for the SoftwareSerial object
15     mySerial.begin(9600);
16 }
17
18 void loop() {
19     if (mySerial.available() > 0) {
20         mySerial.read();
21     }
22 }
```

See also

[SoftwareSerial\(\)](#)

[begin\(\)](#)

[isListening\(\)](#)

[overflow\(\)](#)

[peek\(\)](#)

[read\(\)](#)

[print\(\)](#)

[println\(\)](#)

[listen\(\)](#)

[write\(\)](#)

begin()

Sets the speed (baud rate) for the serial communication. Supported baud rates are: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 31250, 38400, 57600, and

115200 bauds.

Syntax

[COPY](#)

```
1 mySerial.begin(speed)
```

Parameters

speed: the desired baud rate (long). Supported baud rates are: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 31250, 38400, 57600, and 115200 bauds.

Returns

None.

Example

[COPY](#)

```
1  #include <SoftwareSerial.h>
2
3  #define rxPin 10
4  #define txPin 11
5
6  // Set up a new SoftwareSerial object
7  SoftwareSerial mySerial = SoftwareSerial(rxPin, txPin);
8
9  void setup() {
10     // Define pin modes for TX and RX
11     pinMode(rxPin, INPUT);
12     pinMode(txPin, OUTPUT);
13
14     // Set the baud rate for the SoftwareSerial object
15     mySerial.begin(9600);
16 }
17
18 void loop() {
19     // ...
20 }
```

See also

[SoftwareSerial\(\)](#)

[available\(\)](#)

isListening()

overflow()

peek()

read()

print()

println()

listen()

write()

isListening()

Tests to see if requested software serial object is actively listening.

Syntax

COPY

```
1 mySerial.isListening()
```

Parameters

None.

Returns

Boolean.

Example

COPY

```
1 #include <SoftwareSerial.h>
2
3 // Set up a new SoftwareSerial object with RX in digital pin 10 and TX in dig
4 SoftwareSerial portOne(10, 11);
5
6 void setup() {
7     // Set the baud rate for the Serial port
8     Serial.begin(9600);
9 }
```

```
10      // Set the baud rate for the SerialSoftware object
11      portOne.begin(9600);
12  }
13
14  void loop() {
15      if (portOne.isListening()) {
16          Serial.println("portOne is listening!");
17      }
18
19      // ...
```

See also

[SoftwareSerial\(\)](#)

[available\(\)](#)

[begin\(\)](#)

[overflow\(\)](#)

[peek\(\)](#)

[read\(\)](#)

[print\(\)](#)

[println\(\)](#)

[listen\(\)](#)

[write\(\)](#)

overflow()

Tests to see if a SoftwareSerial buffer overflow has occurred. Calling this function clears the overflow flag, meaning that subsequent calls will return false unless another byte of data has been received and discarded in the meantime. The SoftwareSerial buffer can hold up to 64 bytes.

Syntax

```
1  mySerial.overflow()
```

COPY

Parameters

None.

Returns

Boolean.

Example

[COPY](#)

```
1  #include <SoftwareSerial.h>
2
3  // Set up a new SoftwareSerial object with RX in digital pin 10 and TX in dig
4  SoftwareSerial portOne(10, 11);
5
6  void setup() {
7      // Set the baud rate for the Serial port
8      Serial.begin(9600);
9
10     // Set the baud rate for the SerialSoftware object
11     portOne.begin(9600);
12 }
13
14 void loop() {
15     if (portOne.overflow()) {
16         Serial.println("portOne overflow!");
17     }
18
19     // ...
```

See also

[SoftwareSerial\(\)](#)

[available\(\)](#)

[begin\(\)](#)

[isListening\(\)](#)

[peek\(\)](#)

[read\(\)](#)

[print\(\)](#)

`println()`

`listen()`

`write()`

`peek()`

Return a character that was received on the RX pin of the software serial port. Unlike `read()`, however, subsequent calls to this function will return the same character. Note that only one `SoftwareSerial` object can receive incoming data at a time (select which one with the `listen()` function).

Syntax

[COPY](#)

```
1 mySerial.peek()
```

Parameters

None.

Returns

The character read or -1 if none is available.

Example

[COPY](#)

```
1 #include <SoftwareSerial.h>
2
3 // Set up a new SoftwareSerial object with RX in digital pin 10 and TX in dig
4 SoftwareSerial mySerial(10, 11);
5
6 void setup() {
7     // Set the baud rate for the SerialSoftware object
8     mySerial.begin(9600);
9 }
10
11 void loop() {
12     char c = mySerial.peek();
13 }
```

See also

[SoftwareSerial\(\)](#)

[available\(\)](#)

[begin\(\)](#)

[isListening\(\)](#)

[overflow\(\)](#)

[read\(\)](#)

[print\(\)](#)

[println\(\)](#)

[listen\(\)](#)

[write\(\)](#)

read()

Return a character that was received on the RX pin of the SoftwareSerial object. Note that only one SoftwareSerial object can receive incoming data at a time (select which one with the listen() function).

Syntax

[COPY](#)

```
1 mySerial.read()
```

Parameters

None.

Returns

The character read or -1 if none is available.

Example

[COPY](#)

```
1  #include <SoftwareSerial.h>
2
3  // Set up a new SoftwareSerial object with RX in digital pin 10 and TX in dig
4  SoftwareSerial mySerial(10, 11);
5
6  void setup() {
7      // Set the baud rate for the SerialSoftware object
8      mySerial.begin(9600);
9  }
10
11 void loop() {
12     char c = mySerial.read();
13 }
```

See also

[SoftwareSerial\(\)](#)

[available\(\)](#)

[begin\(\)](#)

[isListening\(\)](#)

[overflow\(\)](#)

[peek\(\)](#)

[print\(\)](#)

[println\(\)](#)

[listen\(\)](#)

[write\(\)](#)

print()

Prints data to the transmit pin of the SoftwareSerial object. Works the same as the `Serial.print()` function.

Syntax

```
1  mySerial.print(val)
```

COPY

Parameters

val: the value to print.

Returns

The number of bytes written (reading this number is optional).

Example

[COPY](#)

```
1  #include <SoftwareSerial.h>
2
3  // Set up a new SoftwareSerial object with RX in digital pin 10 and TX in d
4  SoftwareSerial mySerial(10, 11);
5
6  int analogValue;
7
8  void setup() {
9      // Set the baud rate for the SerialSoftware object
10     mySerial.begin(9600);
11 }
12
13 void loop() {
14     // Read the analog value on pin A0
15     analogValue = analogRead(A0);
16
17     // Print analogValue in the Serial Monitor in many formats:
18     mySerial.print(analogValue);           // Print as an ASCII-encoded decim
19     mySerial.print("\t");                  // Print a tab character
20     mySerial.print(analogValue, DEC);      // Print as an ASCII-encoded decim
21     mySerial.print("\t");                  // Print a tab character
22     mySerial.print(analogValue, HEX);      // Print as an ASCII-encoded hexad
23     mySerial.print("\t");                  // Print a tab character
24     mySerial.print(analogValue, OCT);      // Print as an ASCII-encoded octal
25     mySerial.print("\t");                  // Print a tab character
26     mySerial.print(analogValue, BIN);      // Print as an ASCII-encoded binar
27     mySerial.print("\t");                  // Print a tab character
28     mySerial.print(analogValue/4, BYTE);   // Print as a raw byte value (divi
29                                           // value in 4 because analogRead()
30                                           // from 0 to 1023, but a byte can
31
32     mySerial.print("\t");                  // Print a tab character
33     mySerial.println();                    // Print a line feed character
34
35     // Pause for 10 milliseconds before the next reading
```

See also

SoftwareSerial()

available()

begin()

isListening()

overflow()

peek()

read()

print()

println()

listen()

write()

println()

Prints data to the transmit pin of the SoftwareSerial object followed by a carriage return and line feed. Works the same as the Serial.println() function.

Syntax

```
1 mySerial.println(val)
```

COPY

Parameters

val: the value to print.

Returns

The number of bytes written (reading this number is optional).

Example

```
1 #include <SoftwareSerial.h>
2
```

COPY

```

3 // Set up a new SoftwareSerial object with RX in digital pin 10 and TX in d
4 SoftwareSerial mySerial(10, 11);
5
6 int analogValue;
7
8 void setup() {
9     // Set the baud rate for the SerialSoftware object
10    mySerial.begin(9600);
11 }
12
13 void loop() {
14     // Read the analog value on pin A0
15     analogValue = analogRead(A0);
16
17     // Print analogValue in the Serial Monitor in many formats:
18     mySerial.print(analogValue);           // Print as an ASCII-encoded decim
19     mySerial.print("\t");                  // Print a tab character
20     mySerial.print(analogValue, DEC);      // Print as an ASCII-encoded decim
21     mySerial.print("\t");                  // Print a tab character
22     mySerial.print(analogValue, HEX);      // Print as an ASCII-encoded hexad
23     mySerial.print("\t");                  // Print a tab character
24     mySerial.print(analogValue, OCT);      // Print as an ASCII-encoded octal
25     mySerial.print("\t");                  // Print a tab character
26     mySerial.print(analogValue, BIN);      // Print as an ASCII-encoded binar
27     mySerial.print("\t");                  // Print a tab character
28     mySerial.print(analogValue/4, BYTE);   // Print as a raw byte value (divi
29                                           // value in 4 because analogRead()
30                                           // from 0 to 1023, but a byte can
31
32     mySerial.print("\t");                  // Print a tab character
33     mySerial.println();                    // Print a line feed character
34
35     // Delay 500ms before reading the next value
36 }

```

See also

[SoftwareSerial\(\)](#)

[available\(\)](#)

[begin\(\)](#)

[isListening\(\)](#)

[overflow\(\)](#)

[peek\(\)](#)

[read\(\)](#)

[print\(\)](#)

[listen\(\)](#)

`write()`

`listen()`

Enables the selected SoftwareSerial object to listen. Only one SoftwareSerial object can listen at a time; data that arrives for other ports will be discarded. Any data already received is discarded during the call to `listen()` function (unless the given instance is already listening).

Syntax

[COPY](#)

```
1 mySerial.listen()
```

Parameters

None.

Returns

Returns true if it replaces another.

Example

[COPY](#)

```
1 #include <SoftwareSerial.h>
2
3 // Set up a new SoftwareSerial object with RX in digital pin 10 and TX in dig
4 SoftwareSerial portOne(10, 11);
5
6 // Set up a new SoftwareSerial object with RX in digital pin 8 and TX in digi
7 SoftwareSerial portTwo(8, 9);
8
9 void setup() {
10     // Set the baud rate for the Serial object
11     Serial.begin(9600);
12
13     // Set the baud rate for the SerialSoftware objects
14     portOne.begin(9600);
15     portTwo.begin(9600);
16 }
17
18 void loop() {
```

```
19 // Enable SoftwareSerial object to listen
20 portOne.listen();
21
22 if (portOne.isListening()) {
23     Serial.println("portOne is listening!");
24 } else {
25     Serial.println("portOne is not listening!");
26 }
27
28 if (portTwo.isListening()) {
29     Serial.println("portTwo is listening!");
30 } else {
31     Serial.println("portTwo is not listening!");
32 }
33 }
```

See also

[SoftwareSerial\(\)](#)

[available\(\)](#)

[begin\(\)](#)

[isListening\(\)](#)

[overflow\(\)](#)

[peek\(\)](#)

[read\(\)](#)

[print\(\)](#)

[println\(\)](#)

[write\(\)](#)

write()

Prints data to the transmit pin of the SoftwareSerial object as raw bytes. Works the same as the `Serial.write()` function.

Syntax

```
1 mySerial.write(val)
```

COPY

Parameters

val: the binary value to print.

Returns

The number of bytes written (reading this number is optional).

Example

[COPY](#)

```
1  #include <SoftwareSerial.h>
2
3  // Set up a new SoftwareSerial object with RX in digital pin 10 and TX in dig
4  SoftwareSerial mySerial(10, 11);
5
6  void setup() {
7      // Set the baud rate for the SerialSoftware object
8      mySerial.begin(9600);
9  }
10
11 void loop() {
12     // Send a byte with the value 45
13     mySerial.write(45);
14
15     //Send the string "hello" and return the length of the string.
16     int bytesSent = mySerial.write("hello");
17 }
```

See also

[SoftwareSerial\(\)](#)

[available\(\)](#)

[begin\(\)](#)

[isListening\(\)](#)

[overflow\(\)](#)

[peek\(\)](#)

[read\(\)](#)

[print\(\)](#)

[println\(\)](#)

listen()