

# SerialRx

This is a CFunction module which can add up to 8 serial input ports to the Micromite MkII running MMBasic 4.6 or later. The SerialRx module has the following features:

- It is a receive only function (a SerialTx function is separately available).
- Will receive at any baudrate from 110 bps to 38400 bps.
- Will work with up to four I/O pins on the 28-pin Micromite and eight on the 44-pin device.
- Can terminate the receive function based on the number of characters received, a timeout or one or more terminating characters.

## Adding the Function to MMBasic

To add the SerialRx function to MMBasic you must insert the following code somewhere in your BASIC program (you can use copy and paste from this document). The exact spot is not important but at the end of the program is typical.

```
CFunction SerialRx 0000001C
 40024800 00442021 40024800 0044102b 1440ffffd 00000000 03e00008 00000000
 3c03bf81 8c65f000 3c02003d 24420900 7ca51400 70a23002 8c63f000 3c040393
 34848700 7c6316c0 00c41021 00621007 3c03029f 24636300 50430003 3c0202dc
 03e00008 00000000 03e00008 24426c00 27bdfc0 afa70014 afbf003c afb5002c
 afb40028 afb20020 afb1001c afa60048 00808821 afa50010 afbe0038 afb70034
 afb60030 afb30024 afb00018 0411FFDC 8fb50054 8fa50010 3c141062 8cb20000
 26944dd3 00129040 0052001b 024001f4 0411FFD3 00009012 00540019 8fa70014
 0000a010 8ce30000 0014a1c2 2652ffffb 7283a002 00001021 40824800 3c10bf88
 8e240000 0012b842 0000b021 0200f021 24130008 8e026020 00821006 30420001
 50400014 02e02021 40024800 0054102b 1440fff8 8fbf003c 24040001 00002821
 00801021 00a01821 8fbe0038 8fb70034 8fb60030 8fb5002c 8fb40028 8fb30024
 8fb20020 8fb1001c 8fb00018 03e00008 27bd0040 0411FFA2 00000000 8fc26020
 8e240000 00821006 30420001 1440ffe1 00001821 02402021 afa20010 0411FF98
 afa30014 8e056020 8e240000 8fa20010 00852006 30840001 8fa30014 00442004
 00641825 24420001 1453ffff2 306300ff 02402021 0411FF8A afa30014 3c02bf88
 8c426020 8e240000 00821006 30420001 1040ffc8 8fa30014 8fa40048 26d60001
 00961021 a0430000 a0960000 8fa20050 10400005 00000000 8c420000 02c2102b
 10400027 24040002 12a00011 2ac200ff 82a60000 18c0000e 00000000 82a20001
 1062000f 00001021 10000005 24420001 80a40001 1083000b 24040003 24420001
 0046202a 1480fffa 02a22821 2ac200ff 50400013 24040002 1000ffa6 8e240000
 24040003 00002821 8fbf003c 00801021 00a01821 8fbe0038 8fb70034 8fb60030
 8fb5002c 8fb40028 8fb30024 8fb20020 8fb1001c 8fb00018 03e00008 27bd0040
 1000fff1 00002821
End CFunction
```

## Parameters

The SerialTx function (created by adding the above code) takes four to six parameters:

`r = SerialTx( port, baud-rate, string$, timeout, maxchars, termchars)`

Where *port* is a code for the I/O pin (see below for how to map this to an I/O pin)  
*baud-rate* is the desired transmit speed  
*string\$* is the string where the received characters will be stored.  
*timeout* is the maximum time (in milliseconds) to wait before returning with whatever has been received.  
*maxchars* is the number of characters to wait for (optional)  
*termchars* is a string and capture will be terminated if a character received matches any character in the string (optional).

Notes:

- The maximum workable *baud-rate* ranges from 38400 at 40MHz to 9600 at 10MHz.
- Both *maxchars* and *termchars* are optional and can be omitted if not required.

The return value will be the number 1, 2 or 3 where:

- 1 a timeout occurred and *string\$* contains the characters received up to then.
- 2 the number of characters specified in *maxchars* had been received.
- 3 the last character received matched a character in the argument *termchars*.

The *port* argument is used to select the input pin for the received data. Use this table to determine the value of *port*:

<i>port</i>	I/O pin on a 28-pin chip	I/O pin on a 44-pin chip
0	2	19
1	3	20
2	9	30
3	10	31
7		13
8		32
9		35
10		12

## Using the Function

To receive a string from a serial port using SerialRx you must first prepare the I/O pin that you will use (this example uses I/O pin 30 on the 44-pin Micromite):

```
setpin 30, din ' set the pin as an input
```

Once you have done this you can receive data using the SerialRx function. For example:

```
r = SerialRx( 2, 19200, s$, 1000 )
```

This will wait for one second (1000mS) and return with whatever was received during that time. The received characters will be saved to the string *s\$*.

As another example, the following will wait for ten seconds or five characters (whichever occurred first) and print the received data:

```
setpin 30, din ' set the pin as an input
r = SerialRx( 2, 19200, s$, 10000 , 5)
print s$
```

This example will wait for thirty seconds, or for 200 characters, or for a carriage return character (whichever occurred first) and print the received data:

```
setpin 30, din ' set the pin as an input
r = SerialRx( 2, 19200, s$, 30000 , 200, chr$(13))
print s$
```

You can receive data from as many I/O pins as you want simply by changing *port* and *baud-rate* as required every time the function is used. But note that the function will only return characters received while the function is waiting for input.