**RPi::Serial** - Basic read/write interface to a serial port

**SYNOPSIS**

use RPi::Serial;

my $dev = "/dev/ttyAMA0";

my $baud = 115200;

my $ser = RPi::Serial->new($dev, $baud);

$ser->putc(5);

$ser->puts("hello, world!");

my $char = $ser->getc;

my $num\_bytes = 12;

my $str = $ser->gets($num\_bytes);

$ser->flush;

my $bytes\_available = $ser->avail;

$ser->close;

**METHODS**

**new**($device, $baud);

Opens the specified serial port at the specified baud rate, and returns a new RPi::Serial object.

Parameters:

$device Mandatory, String: The serial device to open (eg: "/dev/ttyAMA0").

$baud Mandatory, Integer: A valud baud rate to use (eg: 115200).

**close**

Closes an already open serial device.

**avail**

Returns the number of bytes waiting to be read if any.

**flush**

Flush any data currently in the serial buffer.

**fd**

Returns the ioctl file descriptor for the current serial object.

**getc**

Retrieve a single character from the serial port.

**gets**($num\_bytes)

Read a specified number of bytes into a string.

Parameters:

$num\_bytes

Mandatory, Integer; The number of bytes to read. If this number is larger than what is available to be read, a 10 second timeout will briefly hand your application.

**putc**($char)

Writes a single character to the serial device.

Parameters:

$char Mandatory, Unsigned Char: The character to write to the port.

**puts**($string)

Write a character string to the serial device.

Parameters:

$string Mandatory, String: Whatever you want to write to the serial line.