class UART – duplex serial communication bus

UART implements the standard UART/USART duplex serial communications protocol. At the physical level it consists of 2 lines: RX and TX. The unit of communication is a character (not to be confused with a string character) which can be 8 or 9 bits wide.

UART objects can be created and initialised using:

```
from machine import UART

uart = UART(1, 9600)                         # init with given baudrate
uart.init(9600, bits=8, parity=None, stop=1) # init with given parameters
```

Supported parameters differ on a board:

Pyboard: Bits can be 7, 8 or 9. Stop can be 1 or 2. With *parity=None*, only 8 and 9 bits are supported. With parity enabled, only 7 and 8 bits are supported.

WiPy/CC3200: Bits can be 5, 6, 7, 8. Stop can be 1 or 2.

A UART object acts like a `stream` object and reading and writing is done using the standard stream methods:

```
uart.read(10)       # read 10 characters, returns a bytes object
uart.read()         # read all available characters
uart.readline()     # read a line
uart.readinto(buf)  # read and store into the given buffer
uart.write('abc')   # write the 3 characters
```

Constructors

class **machine.UART**(*id, ...*)

Construct a UART object of the given id.

Methods

UART.init(*baudrate=9600, bits=8, parity=None, stop=1, *, ...*)

Initialise the UART bus with the given parameters:

- *baudrate* is the clock rate.
- *bits* is the number of bits per character, 7, 8 or 9.
- *parity* is the parity, `None`, 0 (even) or 1 (odd).
- *stop* is the number of stop bits, 1 or 2.

Additional keyword-only parameters that may be supported by a port are:

- *tx* specifies the TX pin to use.
- *rx* specifies the RX pin to use.
- *txbuf* specifies the length in characters of the TX buffer.
- *rxbuf* specifies the length in characters of the RX buffer.

On the WiPy only the following keyword-only parameter is supported:

- *pins* is a 4 or 2 item list indicating the TX, RX, RTS and CTS pins (in that order). Any of the pins can be None if one wants the UART to operate with limited functionality. If the RTS pin is given the the RX pin must be given as well. The same applies to CTS. When no pins are given, then the default set of TX and RX pins is taken, and hardware flow control will be disabled. If *pins* is `None`, no pin assignment will be made.

UART.deinit()

Turn off the UART bus.

UART.any()

Returns an integer counting the number of characters that can be read without blocking. It will return 0 if there are no characters available and a positive number if there are characters. The method may return 1 even if there is more than one character available for reading.

For more sophisticated querying of available characters use select.poll:

```
poll = select.poll()
poll.register(uart, select.POLLIN)
poll.poll(timeout)
```

UART.read([*nbytes*])

Read characters. If `nbytes` is specified then read at most that many bytes, otherwise read as much data as possible.

Return value: a bytes object containing the bytes read in. Returns `None` on timeout.

UART.readinto(*buf* [, *nbytes*])

Read bytes into the `buf`. If `nbytes` is specified then read at most that many bytes. Otherwise, read at most `len(buf)` bytes.

Return value: number of bytes read and stored into `buf` or `None` on timeout.

UART.readline()

Read a line, ending in a newline character.

Return value: the line read or `None` on timeout.

UART.write(*buf*)

Write the buffer of bytes to the bus.

Return value: number of bytes written or `None` on timeout.

UART.sendbreak()

Send a break condition on the bus. This drives the bus low for a duration longer than required for a normal transmission of a character.

UART.irq(*trigger, priority=1, handler=None, wake=machine.IDLE*)

Create a callback to be triggered when data is received on the UART.

- *trigger* can only be `UART.RX_ANY`
- *priority* level of the interrupt. Can take values in the range 1-7. Higher values represent higher priorities.
- *handler* an optional function to be called when new characters arrive.
- *wake* can only be `machine.IDLE`.

> **Note**
>
> The handler will be called whenever any of the following two conditions are met:
>
> - 8 new characters have been received.
> - At least 1 new character is waiting in the Rx buffer and the Rx line has been silent for the duration of 1 complete frame.
>
> This means that when the handler function is called there will be between 1 to 8 characters waiting.

Returns an irq object.

Availability: WiPy.

Constants

UART.RX_ANY

IRQ trigger sources

Availability: WiPy.