

The `dwt_readcarrierintegrator()` API function reads the receiver carrier integrator value and returns it as a 32-bit signed value. The receive carrier integrator value is valid at the end of reception of a frame, (and before the receiver is re-enabled). It reflects the frequency offset of the remote transmitter with respect to the local receive clock. A positive carrier integrator value means that the local receive clock is running faster than that of the remote transmitter device.

Parameters :

none

Return Parameters :

type	Description
int32	Receiver carrier integrator value

Notes :

This `dwt_readcarrierintegrator()` API may be called after receiving a frame to determine the clock offset of the remote transmitter the sent the frame. The receive frame should be valid (i.e. with good CRC) otherwise the clock offset information may be incorrect. The following constants are defined to allow the returned carrier integrator be converted to a frequency offset in Hertz (which depends on the data rate, 110Kb/s is different to the rest), and from that to a clock offset in PPM (which depends on the channel centre frequency): `FREQ_OFFSET_MULTIPLIER`, `FREQ_OFFSET_MULTIPLIER_110KB`, `HERTZ_TO_PPM_MULTIPLIER_CHAN_1`, `HERTZ_TO_PPM_MULTIPLIER_CHAN_2`, `HERTZ_TO_PPM_MULTIPLIER_CHAN_3` and `HERTZ_TO_PPM_MULTIPLIER_CHAN_5`.

The `HERTZ_TO_PPM_xxx` multipliers are negative quantities, so when the resultant clock offsets are positive it means that the local receiver's clock is running slower than that of the remote transmitter.

Example code:

```
int32 ci ;
float clockOffsetHertz ;
float clockOffsetPPM ;

ci = dwt_readcarrierintegrator() ; // Read carrier integrator value

// at 110 kb/s data rate convert carrier integrator to clock offset in Hz.
clockOffsetHertz = ci * FREQ_OFFSET_MULTIPLIER_110KB ;

// On channel 5 convert this to clock offset in PPM.
clockOffsetPPM = clockOffsetHertz * and HERTZ_TO_PPM_MULTIPLIER_CHAN_5 ;
```

NB: Please also refer to simple example 6: single-sided two-way ranging (SS TWR) where the [initiator](#) end (since driver version 4.0.6) uses the carrier integrator to correct the range estimate calculation for the clock offset of the remote responder node.

5.77 SPI driver functions

These functions are platform specific SPI read and write functions, external to the DW1000 driver code, used by the device driver to send and receive data over the SPI interface to and from the DW1000. The DW1000 device driver abstracts the target SPI device by calling it through generic functions `writetospi()` and `readfromspi()`. In porting the DW1000 device driver, to different target