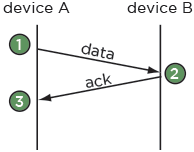
Functional description

# Wireless communication

The Pozyx system provides single-hop wireless communication between neighboring pozyx devices by making use of the ultra-wideband signals. Every device in the network is uniquely identified by its 16-bit network id stored in the POZYX\_NETWORK\_ID register. To make wireless communication possible between two devices, they must have the same UWB settings and the receiver must be turned on.

With Pozyx, it is possible to send custom data bytes or to wirelessly interact with the registers of a remote device, allowing for remote control of that device. This will be explained in the following subsections.

## Sending data

Transmission of data is initiated by device A by calling the register function POZYX\_TX\_SEND with parameters the network id of device B and option REG\_DATA. The transmission process is outlined in the figure on the right.

Step 1

Device A sends the data in the TX buffer (accessible through POZYX\_TX\_DATA) to device B.

Step 2

Upon reception of the data, device B will

place the data in its RX buffer (accessible through POZYX\_RX\_DATA),

place the length of the received data in the register POZYX\_RX\_DATA\_LEN,

place the network id of device A in POZYX\_RX\_NETWORK\_ID,

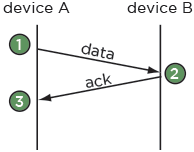
and mark the RX\_DATA bit in POZYX\_INT\_STATUS. If the interrupts are enabled, device B will also generate an interrupt.

Finally, device B will send a wireless acknowledgment to device A saying that it successfully received the data.

Step 3

Device A receives the acknowledgment (this does not fire an interrupt).

## Remote register write

Performing a remote register write is initiated by device A by calling the register function POZYX\_TX\_SEND with parameters the network id of device B and option REG\_WRITE. The transmission process is outlined in the figure on the right.

Step 1

Device A sends the data in the TX buffer (accessible through POZYX\_TX\_DATA) to device B. For a register write, the data packet that is transmitted must have the following content.

byte number Description

byte 0 Register address. The register address to start writing to.

byte 1 data byte 0 (will be written at the register address)

byte 2 data byte 1 (will be written at the register address + 1)

... ...

byte 99 data byte 98 (will be written at the register address + 98)

Step 2

Upon reception of the data, device B will perform the requested register write operation.

Finally, device B will send a wireless acknowledgment to device A saying that it successfully received the data.

Step 3

Device A receives the acknowledgment (this does not fire an interrupt).

## https://www.pozyx.io/assets/images/docs/sending_data.pngRemote register read

Performing a remote register read is initiated by device A by calling the register function POZYX\_TX\_SEND with parameters the network id of device B and option REG\_READ. The transmission process is outlined in the figure on the right.

Step 1

Device A sends the data in the TX buffer (accessible through POZYX\_TX\_DATA) to device B. For a register read, the data packet that is transmitted must have the following content.

byte number Description

byte 0 Register address. The register address to start reading from.

byte 1 Read length. The number of bytes to read in total.

Step 2

Upon reception of the data, device B will process the register read command and send a data acknowledgment to device A that contains the requested register data.

Step 3

Upon reception of the data acknowledgment, device A will

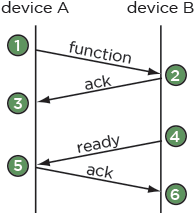
place the data in its RX buffer (accessible through POZYX\_RX\_DATA),

place the length of the received data in the register POZYX\_RX\_DATA\_LEN,

place the network id of device A in POZYX\_RX\_NETWORK\_ID,

and mark the RX\_DATA bit in POZYX\_INT\_STATUS. If the interrupts are enabled, device A will also generate an interrupt.

## Remote function call

Performing a remote register function is initiated by device A by calling the register function POZYX\_TX\_SEND with parameters the network id of device B and option REG\_FUNC. The transmission process is outlined in the figure on the right.

Step 1

Device A sends the data in the TX buffer (accessible through POZYX\_TX\_DATA) to device B. For a register function call, the data packet that is transmitted must have the following content.

byte number Description

byte 0 Register address. The register address of the function

byte 1 function parameter byte 0

byte 2 function parameter byte 1

... ...

byte 99 function parameter byte 98

Step 2

Upon reception of the function request, device B will process the register function command and send a data acknowledgment to device A that contains the function return data.

Step 3

Upon reception of the data acknowledgment, device A will

place the data in its RX buffer (accessible through POZYX\_RX\_DATA),

place the length of the received data in the register POZYX\_RX\_DATA\_LEN,

place the network id of device A in POZYX\_RX\_NETWORK\_ID,

and mark the RX\_DATA bit in POZYX\_INT\_STATUS. If the interrupts are enabled, device A will also generate an interrupt.

Step 4 (optional)

For certain remote function calls, it is possible that device B will send a second response that contains data related to the function call. More specifically, for POZYX\_DO\_RANGING and POZYX\_DO\_POSITIONING, device B will signal device A when this operation (which may take tens of milliseconds) has finished. Furthermore, the acknowledgment will also contain relevant data bytes to the function in order to reduce the number of wireless calls.

Step 5 (optional)

Upon reception of the data acknowledgment, device A will

place the data in its RX buffer (accessible through POZYX\_RX\_DATA),

place the length of the received data in the register POZYX\_RX\_DATA\_LEN,

place the network id of device A in POZYX\_RX\_NETWORK\_ID,

and mark the RX\_DATA bit in POZYX\_INT\_STATUS. If the interrupts are enabled, device A will also generate an interrupt.

Step 6 (optional)

Device B receives the acknowledgment (this does not fire an interrupt).

# Positioning

The pozyx device is capable of obtaining accurate positioning information. A description of how positioning works can be found here: How does positioning work?. The device can be configured to acquire positioning information at regular intervals or a single time. When positioning is finished, the position estimates and error estimates are stored in the positioning data registers.

Provide the anchor positions.

For positioning, the coordinates of the anchors must be available in the device list. This can be provided either manually or automatically.

Configure the positioning algorithm (optional).

By default, the Pozyx device is configured to perform 3D positioning using 4 anchors that are automatically selected from the device list. These settings can be altered in order to obtain greater accuracy.

The following registers can be used for configuring the positioning algorithm:

POZYX\_POS\_ALG, POZYX\_POS\_NUM\_ANCHORS, and POZYX\_POS\_SET\_ANCHOR\_IDS.

Start positioning.

Positioning can be initiated in two ways. Either directly by calling POZYX\_DO\_POSITIONING which will estimate the tag position once. Or by setting the positioning interval in POZYX\_INTERVAL\_MS. Whenever the positioning interval is set, a timer will start which will continuously perform positioning with a fixed interval in between the estimates.

Read out the positioning data.

Depending on the algorithm configuration, positioning may take some time. Using the default settings, it takes approximately 70ms to obtain a position estimate. When positioning is finished, the pozyx device will signal this by setting the POS-bit high in POZYX\_INT\_STATUS. In case the interrupts are enabled, this event will cause an interrupt to signal the host that new positioning data is available. When positioning is finished, the following data can be read:

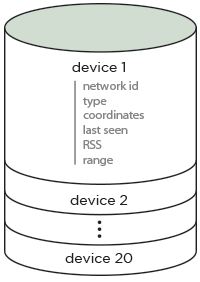
Coordinates: the estimated coordinates are stored in the registers POZYX\_POS\_X, POZYX\_POS\_Y, POZYX\_POS\_Z that can be read by the host.

Error covariance: information about the uncertainty of the position estimate is provided by the registers starting from POZYX\_POS\_ERR\_XX. These registers hold the estimated error covariance of the position estimate. This information can be used to assess the quality of the estimation or the anchor placement.

Selected anchors: when Pozyx is configured for automatic anchor selection (in POZYX\_POS\_NUM\_ANCHORS), the selected anchor's can be obtained using POZYX\_POS\_GET\_ANCHOR\_IDS.

Range measurements: the range measurements made during positioning can be obtained using POZYX\_DEVICE\_GETRANGEINFO.

# Device list



Because the Pozyx device operates wirelessly, it must have a sense of pozyx devices in the network. To this end, an internal device list is maintained that stores all important information about a other network devices such as:

The network identifier.

The type: tag or anchor.

The position (in case it is known for anchors).

The range to the device.

The signal strength indicator.

A timestamp when it was last seen.

The information inside the device list is used by the positioning algorithm

## Device discovery

This functionality automatically adds all neighboring pozyx devices in the internal device list. Note that all pozyx devices must be configured with the same ultra-wideband settings to make wireless communication possible. This function is triggered by calling the function register POZYX\_DEVICES\_DISCOVER.

## Anchor calibration

For positioning, the coordinates of the anchors must be known. In general, the accuracy of the coordinates of the anchors directly influences the positioning accuracy of the tag; roughly speaking, 10cm error in the position of the anchors will result in 10cm error in the position of the tags. Because of this, accurately defining the coordinates of the anchors is an important aspect of accurate positioning. Obtaining the anchor coordinates can be performed manually, by utilizing a laser meter, an accurate floor plan or some other method. Alternatively or it can be performed automatically by calling the function register POZYX\_DEVICES\_CALIBRATE. This function will automatically discover all neighboring pozyx devices and will estimate their relative coordinates.

# Interrupts

Pozyx has a number of interrupt events that can be configured to drive one of the two user-selectable IRQ output pins. The interrupt trigger is configured as raising edge. A number of status registers are provided in the system to monitor and report data related to the interrupt. More specifically, interrupts for certain events can be enabled or disabled in the POZYX\_INT\_MASK register. Interrupts are available for the following events

Error event.

New positioning data event.

New IMU data event

Wireless data received event

Register Function ready event

The event that triggered an interrupt can be read from the POZYX\_INT\_STATUS register. Note that even with interrupts disabled, the interrupt status register will still indicate which events have occurred. This way, the system can be used without interrupts by regularly polling this register. However, polling should be avoided whenever possible because it slows down the overall system.

By default, on power-up, all interrupt generating events are masked and interrupts are disabled.