

Binary Protocol 32-bit (v0.1)



Introduction

The Harp Protocol is a binary communication protocol created in order to facilitate and unify the interaction between different devices. It was designed with efficiency and ease of parsing in mind.

The protocol is based on addresses. Each address points to a certain memory position available in the device. These positions are called registers. Each register is defined by a data type and some meaningful functionality attached to the data.

The Harp Binary Protocol is commonly used for all exchanges between a Controller and a Device. The Controller can be a computer, or a server and the Device can be a data acquisition or actuator microcontroller.

The available packets are:

- Command: Sent by the Controller to the Device. Command messages can be used to read or write the register contents.
- Reply: Sent by the Device in response to a Command.
- Event: Sent by the Device when an external or internal event of interest happens. An Event message will always carry the contents of the register that the event refers to.

Note

The Harp Binary Protocol uses Little-Endian byte ordering.

Harp Message

The Harp Message contains a minimal amount of information to execute a well-defined exchange of data. It follows the structure below.

| | | | |
|-----------------|-------------|-------------|-------|
| 31 – 24 | 23 – 16 | 15 – 8 | 7 – 0 |
| RegisterAddress | PayloadType | MessageType | |
| Length | | | |
| Port * | | | |
| Seconds * | | | |
| Nanoseconds * | | | |
| Payload * | | | |
| Counter | | Checksum | |

* Optional fields

Note

The field [Payload] can have more than one 32-bit word.

MessageType

| | | | | | | | |
|--------|---|---|-----------|---|---|------|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Flag32 | 0 | 0 | FlagError | 0 | 0 | Type | |

Type (2 bits)

Specifies the type of the Harp Message.

| Value | Description |
|-----------|--|
| 1 (Read) | Read the content of the register with address [RegisterAddress] |
| 2 (Write) | Write the content to the register with address [RegisterAddress] |
| 3 (Event) | Send the content of the register with address [RegisterAddress] |

FlagError (1 bit)

This bit is set when an error occurred while executing the command. Examples of possible errors:

- The Controller tries to read a register that doesn't exist.
- The Controller tries to write data which is out of bounds for the specific register functionality.
- [PayloadType] doesn't match the register [RegisterAddress] type.

Flag32 (1 bit)

If this Flag is set, the current Harp Message is constructed according to the 32-bit Harp Binary Protocol. To comply with this document, this bit should always be equal to 1.

PayloadType

| | | | | | | | |
|----------|---------|---|--------------|---|------|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IsSigned | IsFloat | 0 | HasTimestamp | | Type | | |

Type (4 bits)

Specifies the size of the word in the [Payload].

| Value | Description |
|-------|-------------|
| 1 | 8 bits |
| 2 | 16 bits |

| Value | Description |
|-------|-------------|
| 4 | 32 bits |
| 8 | 64 bits |

HasTimestamp (1 bit)

If this bit is set the Harp Message contains a timestamp. In this case the fields [Seconds] and [Nanoseconds] must be present in the message.

IsFloat (1 bit)

This bit indicates whether the [Payload] represents fractional values. If the bit is not set, the payload contains integers.

IsSigned (1 bit)

If the bit is set, indicates that the [Payload] contains integers with signal.

Note
The bits [IsFloat] and [IsSigned] must never be set simultaneously.

RegisterAddress

Contains the address of the register to which the Harp Message refers to.

Length

Contains the number of bytes that are still available and need to be read to complete the Harp message. If the total number of bytes in the Harp Message is not a multiple of four, the field [Payload] will be followed by 0-3 additional zero bytes to ensure that the total size is divisible by four.

Port

This field is optional. If the device is a Hub of Harp Devices, it indicates the origin or destination of the Harp Message. If the field is not used or it's equal to 0xFFFFFFFF, it points to the device itself.

Seconds

Contains the seconds of the Harp Timestamp clock. This field is optional. In order to indicate that this field is available, the bit [HasTimestamp] in the field [PayloadType] needs to be set.

Nanoseconds

It contains the fractional part of the Harp Timestamp clock in nanoseconds.

This field is optional. In order to indicate that this field is available, the bit [HasTimestamp] in the field [PayloadType] needs to be set.

Payload

Contains the data to be transferred.

Checksum

The checksum is equal to the U16 (unsigned 16-bit) sum of all the bytes in the Harp Message.

The receiver of the message should calculate the checksum and compare it with the received. If they don't match, the Harp Message should be discarded.

Counter

Contains a S16 (signed 16-bit) counter that starts from negative one when the device boots.

It should reset to negative one when the device goes into Active Mode.