Common Registers and Operation (Device) 1.2

Table of contents

Introduction

This document defines the standards that new Harp Devices should strive to follow. The goal is to create a common ground for the development and operation of Harp devices, to allow quick and easy integration of new devices into the existing ecosystem. While some registers and functionalities are not mandatory, it is strongly recommended that they are implemented or, at least, considered with compatibility in mind.

Registers

Common Registers

The `Common Registers` are a set of registers that are common to all Harp Devices. These registers are used to identify the device, its version, and its operation mode. The `Common Registers` are mandatory and should be implemented in all Harp Devices. Summary description of each register can be found in [this table](ref missing).

All other registers pertaining to the operation of a specific device (`Application Registers`) should always take a register number equal, or greater, than 32. Moreoever, numbering and naming of these registers will be

left to the developer, as they are specific to each device.

Operation Modes

A Harp Device is expected to implement the following `Operation Modes`:

- **`Standby Mode:`** Replies to host commands. Events are disabled and must not be sent.
- **`Active Mode:`** Replies to host commands. Events are enabled and sent to host whenever the device deems it so.
- *`Speed Mode:`* Allows the implementation of a different and specific communication protocol. On this mode, the Harp Binary Protocol is no longer used. The specific protocol designed must implement the possibility to exit this mode.

The mandatory Operation Modes are the **`Standby Mode`** and **`Active Mode`**. The **`Speed Mode`** is optional and, in many of the applications, not needed. It's strongly recommended that a Harp Device acting as peripheral should continuously check if the communication with the host is active and healthy. If this doesn't happen over the last 3 seconds, the Harp Device should go to Standby Mode and flush/destroy its TX buffer.

Registers

Common Registers

Table - List of available Common Registers

Name	Volatile	Read Only	Type	Add.	Default	Brief Description	Mandatory
R_WHO_AM_I	-	Yes	U16	000	a)	Who am I	Yes
R_HW_VERSION_H	-	Yes	U8	001	a)	Major Hardware version	Yes
R_HW_VERSION_L	-	Yes	U8	002	a)	Minor Hardware version	Yes
R_ASSEMBLY_VERSION	-	Yes	U8	003	a)	Version of the assembled components	Optional
R_CORE_VERSION_H	-	Yes	U8	004	a)	Major core version	Optional
R_CORE_VERSION_L	-	Yes	U8	005	a)	Minor core version	Optional
R_FW_VERSION_H	-	Yes	U8	006	a)	Major Firmware version of the application	Yes
R_FW_VERSION_L	-	Yes	U8	007	a)	Minor Firmware version of the application	Yes

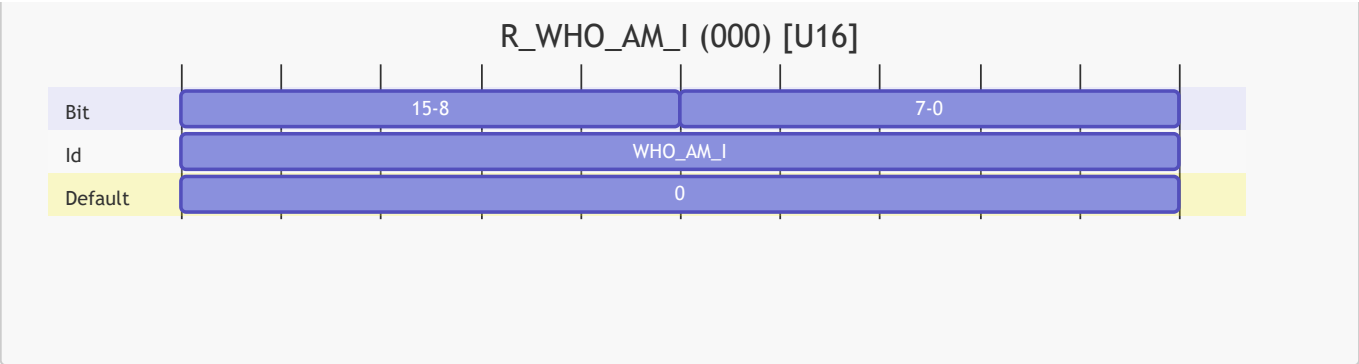
Name	Volatile	Read Only	Type	Add.	Default	Brief Description	Mandatory
R_TIMESTAMP_SECOND	Yes	No	U32	008	0	System timestamp: seconds	Yes
R_TIMESTAMP_MICRO	Yes	Yes	U16	009	0	System timestamp: microseconds	Optional
R_OPERATION_CTRL	No	No	U8	010	b)	Configuration of the operation mode	c)
R_RESET_DEV	No	No	U8	011	b)	Reset device and save non-volatile registers	Optional
R_DEVICE_NAME	No	No	U8	012	b)	Name of the device given by the user	Optional
R_SERIAL_NUMBER	No	No	U16	013	b)	Unique serial number of the device	Optional
R_CLOCK_CONFIG	No	No	U8	014	b)	Synchronization clock configuration	Optional
R_TIMESTAMP_OFFSET	No	No	U8	015	b)	Adds an offset if user updates the Timestamp	Optional

a) These values are stored during factory process and are persistent, i.e., they cannot be changed by the user.
b) Check register notes on the specific register explanation
c) Only parts of the functionality is mandatory. Check register notes on the explanation.

`R_WHO_AM_I` **(U16) – Who Am I**

Address: `000`

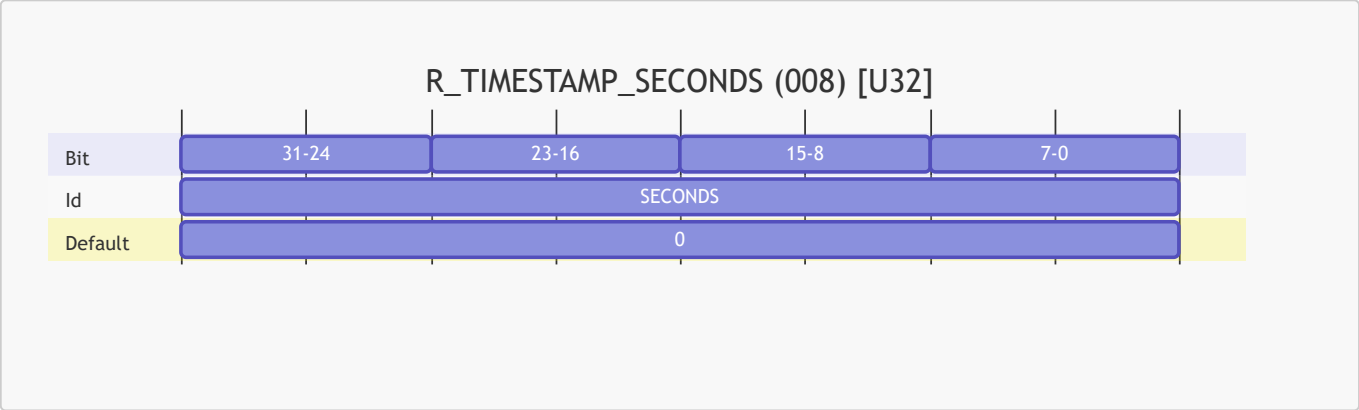
Used to verify the identity of the device. A list of devices can be found at https://github.com/harp-tech/protocol. To reserve a range or certain IDs for your project or company, please follow the instructions in this repository. If the device doesn't have a pre-allocated ID on the IDs list, this register should be set as 0.



`R_TIMESTAMP_SECOND` **(U32) – System timestamp (seconds)**

Address: `008`

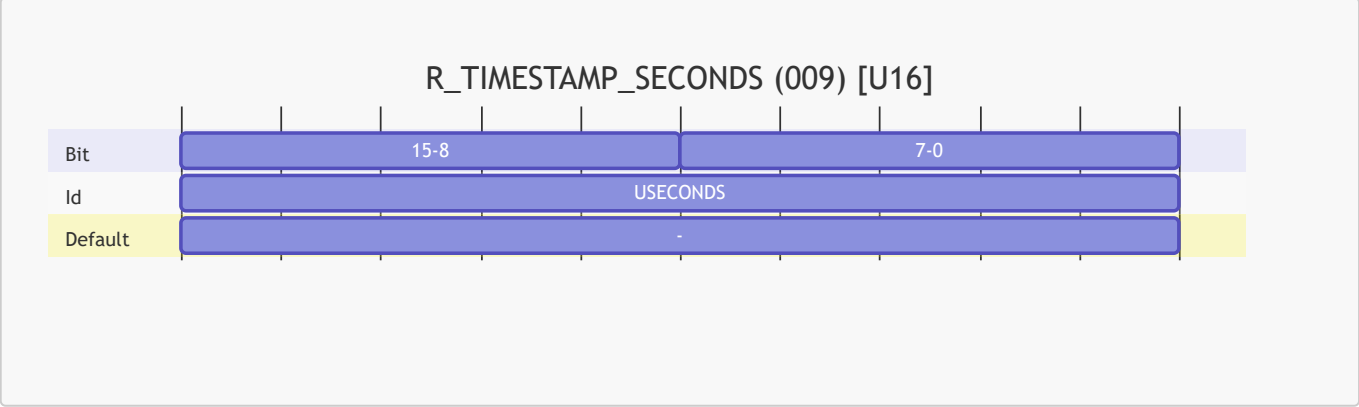
Contains the current system timestamp in whole seconds. The default value is 0 (ZERO) and will increment one unit for each elapsed second.



`R_TIMESTAMP_SECOND` **(U16) – System timestamp (seconds)**

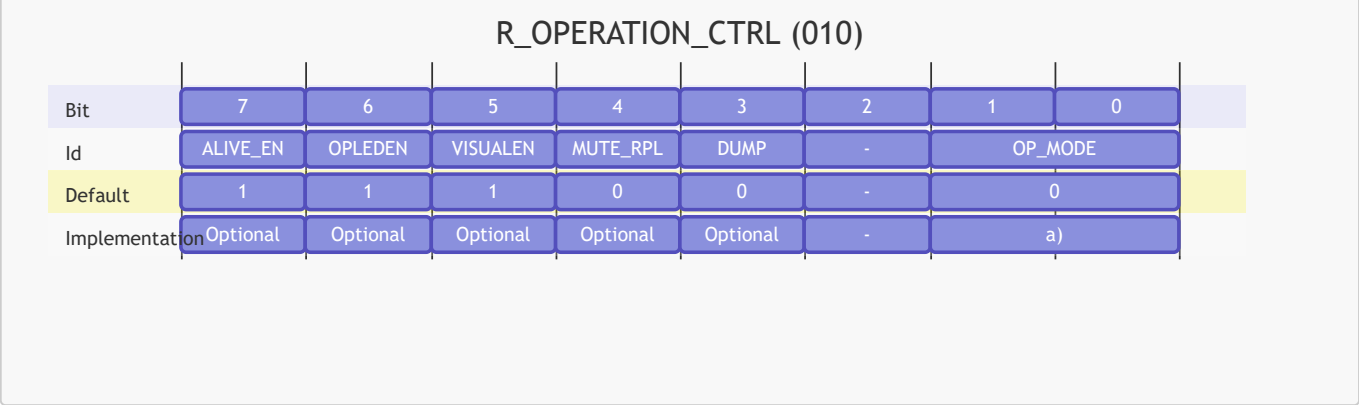
Address: `009`

Contains the microseconds count within each second. Each LSB corresponds to 32 microseconds. The maximum value is 31249.



`R_OPERATION_CTRL` **(U16) – Operation mode configuruation**

Address: `010`



a) Standby Mode and Active Mode are mandatory. Speed Mode is optional.

- **OP_MODE [Bits 1:0]:** These bits define the operation mode of the device. Note that, if Speed Mode is selected, the device will no longer reply to the HARP commands, only to its specific Speed Mode commands.

Table - Available Operation modes

OP_MODE[1:0]	Configuration
0	Standby Mode. The device has all the `Events` turned off.
1	Active Mode. The device turns ON the `Events` detection. Only the enabled Events will be operating.
2	Reserved.
3	Speed Mode. The device enters Speed Mode.

- **DUMP [Bit 3]:** When written to 1, the device adds the content of all registers to the streaming buffer as `Read` messages. This bit is always read as 0.
- **MUTE_RPL [Bit 4]:** If set to 1, the `Replies` to all the `Commands` are muted, i.e., they will not be sent by the device.
- **VISUALEN [Bit 5]:** If set to 1, visual indications, typically LEDs, available on the device will operate. If equals to 0, all the visual indications should turn off.
- **OPLEDEN [Bit 6]:** If set to 1, the LED present on the device will indicate the Operation Mode selected.

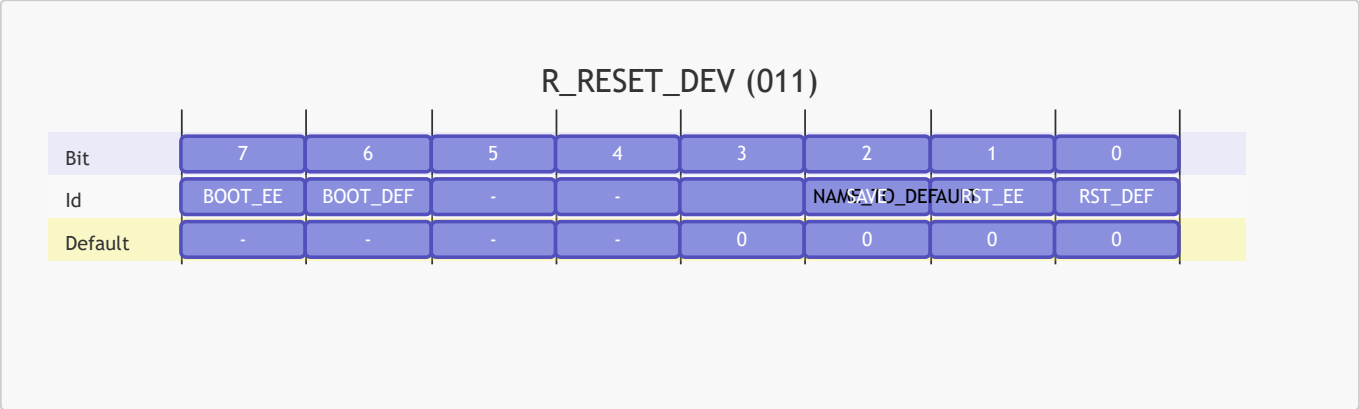
Table - Visual LED toggle feedback

Period (s)	Operation Mode
4	Standby Mode.
2	Active Mode.
1	Speed Mode.
0.1	A critical error occurred. Only a hardware reset or a new power up can remove the device from this Mode.

- **ALIVE_EN [Bit 7]:** If set to 1, the device sends an `Event` Message with the `R_TIMESTAMP_SECONDS` content each second (i.e. Heartbeat). This allows the host to check that the device is alive. Although this is an optional feature, it's strongly recommended to be implemented.

`R_RESET_DEV` (U8) – Reset device and save non-volatile registers

Address: `011`



- **RST_DEF [Bit 0]:** If set to 1, the device resets and, reboots with all the registers, both `Common` and `Application`, with the default values. EEPROM will be erased, and the default values will be restored as the permanent boot option. This bit is always read as 0.
- **RST_EE [Bit 1]:** If set to 1, the device resets and reboots with all the registers, both `Common` and `Applicatin` registers, with the values saved on the non-volatile memory, usually an EEPROM. The EEPROM values will remain the permanent boot option. This bit is always read as 0.
- **SAVE [Bit 3]:** If set to 1, the device will save all the non-volatile registers (both `Common` and `Application`) to the internal non-volatile memory, and reboot. The non-volatile memory should be the permanent boot option.
- **NAME_TO_DEFAULT [Bit 4]:** If set to 1, the device will reboot with the default name. This bit is always read as 0.
- **BOOT_DEF [Bit 6]:** A read-only state bit that indicates whether the device booted with the default register values or not.
- **BOOT_EE [Bit 7]:** A read-only state bit that indicates whether the device booted with the register values saved on the EEPROM or not.

> **Note**
>
> To avoid unexpected behaviors, only one bit at a time should be written to register `R_RESET_DEV`.

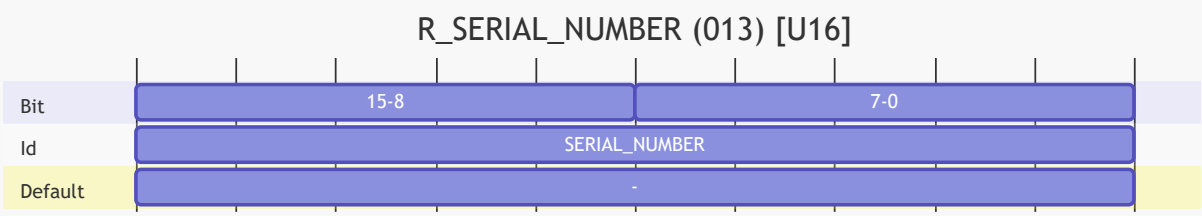
`R_DEVICE_NAME` (25 Bytes) – Device's name

Address: `012`

An array of 25 bytes that should contain the device name. The last, and unused, bytes must be equal to 0. This register is non-volatile. The device will reset if this register is written to.

`R_SERIAL_NUMBER` (U16) – Device's serial number

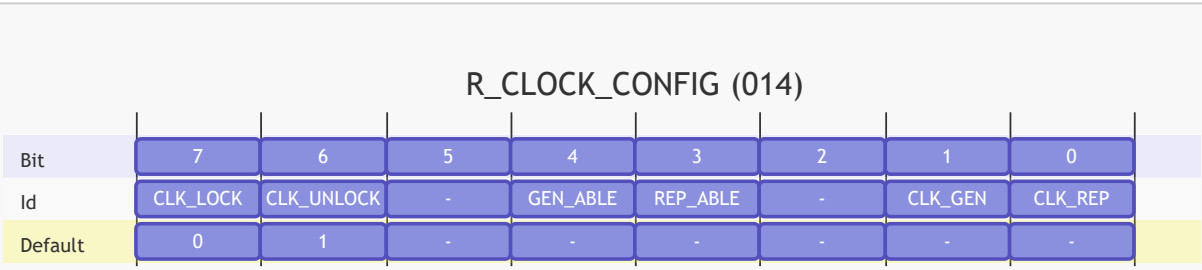
Address: `013`



This number should be unique for each unit of the same Device ID. To write to this register a two-step write command is needed. First, write the value `0xFFFF`, and then the desired serial number (as a `U16`). The device will reset after the second write command is received.

`R_CLOCK_CONFIG` (U8) – Synchronization clock configuration

Address: `014`



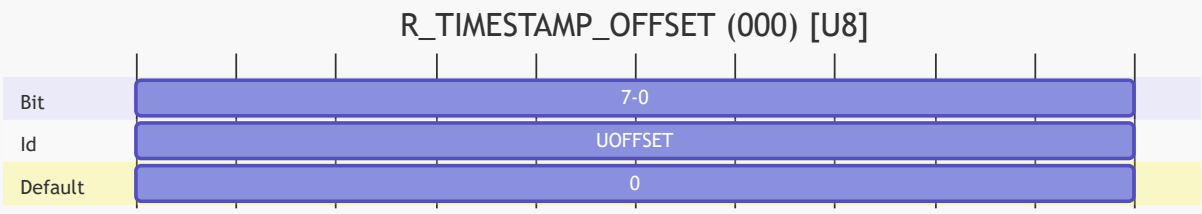
- **CLK_REP [Bit 0]:** If set to 1, the device will repeat the Harp Synchronization Clock to the Clock Output connector, if available. It will act has a daisy-chain, by repeating the Clock Input to the Clock Output. Setting this bit, also unlocks the Harp Synchronization Clock.
- **CLK_GEN [Bit 1]:** If set to 1, the device will generate Harp Synchronization Clock to the Clock Output connector, if available. The Clock Input will be ignored. The bit is read as 1 if the device is generating the Harp Synchronization Clock.
- **REP_ABLE [Bit 3]:** This is a read-only bit signaling if the device is able (1) to repeat the Harp Synchronization Clock timestamp.
- **GEN_ABLE [Bit 4]:** This is a read-only bit signaling if the device is able (1) to generate the Harp Synchronization Clock timestamp.
- **CLK_UNLOCK [Bit 6]:** If set to 1, the device will unlock the timestamp register counter (register `R_TIMESTAMP_SECOND`) and it will now accept new timestamp values. The bit is read as 1 if the timestamp register is unlocked.

- **CLK_UNLOCK [Bit 7]:** If set to 1, the device will lock the current timestamp register counter (register `R_TIMESTAMP_SECOND`) and it will reject any new timestamp values. The bit is read as 1 if the timestamp register is locked.

> **Note**
>
> The device always wakes up in the `unlock` state.

`R_TIMESTAMP_OFFSET` (U8) – Clock calibration offset

Address: `015`



When the value of this register is above 0 (zero), the device's timestamp will be offset by this amount. The register is sensitive to 500 microsecond increments. This register is non-volatile.

Release notes:

- V0.2
  - First draft released.
- V1.0
  - R_RESET_DEV and R_DEVICE_NAME are now optional.
  - Changed Normal Mode to Standby Mode.
  - Added bit ALIVE_EN to register R_OPERATION_CTRL. This is an important feature.
  - Major release.
- V1.1
  - Added bit MUTE_RPL to register R_OPERATION_CTRL.
- V1.2
  - Corrected some wrong names.
- V1.3
  - Added the bit NAME_TO_DEFAULT.
- V1.4

    - Added the register R_SERIAL_NUMBER.

- V1.5

    - Added the register R_CLOCK_CONFIG.

- V1.6

    - Changed device naming to Controller and Peripheral.

- V1.7

    - Raised version to 1.2 since all the foreseen features are included at this point.
    - Added the register R_TIMESTAMP_OFFSET.

- V1.8

    - Replaced HARP_VERSION with CORE_VERSION.

- V2.0

    - Refactor documentation to markdown format.