|  |  |
| --- | --- |
| **Common Registers and Operation (Device) 1.2**  **Document Version 1.7** |  |

|  |  |
| --- | --- |
| **1.** | **Introduction** |
|  | In order to unify the identification of a Harp Device in a system and to create an environment where users can quickly use a Harp Device, mandatory registers and operation modes should be present. Both must comply with the functionalities described on this document. Note that some of the registers and functionalities are not mandatory.  **Common Registers:**  Summary description of each register can be found on Table 2-1.  The addresses of the Application Registers should be equal to 32 or higher. These registers differ from device to device according to the device purpose and is up to the developer the names and types of each register.  **Operation Modes:**  The Harp Device, when acting as a Peripheral, should implement the next Operation Modes:   * **Standby Mode:** Replies to Controller commands. Events are disabled and must not be sent. * **Active Mode:** Replies to Controller commands. Events are enabled and sent to Controller whenever the Peripheral decides. * **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 leave from this mode.   The mandatory Operation Modes are the **Standby Mode** and **Active Mode**. The **Speed Mode** is optional and, in many of the applications, is not needed.  It’s strongly recommended that a Harp Device acting as Peripheral should continuously check if the communication with the Controller 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. |

****

|  |  |
| --- | --- |
| **2.** | **Registers** |
| **2.1** | **Common Registers** |
|  | **Table 2-1. 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\_HARP\_VERSION\_H | - | Yes | U8 | 004 | a) | Major HARP version | Optional | | R\_HARP\_VERSION\_L | - | Yes | U8 | 005 | a) | Minor HARP 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 | | 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. |
| **2.1.1** | **R\_WHO\_AM\_I – Who Am I** |
|  | |  |  | | --- | --- | | Add: 000 | WHO\_AM\_I [15:0] | | 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 send an e-mail to [filipe@oeps.tech](mailto:filipe@oeps.tech).  If the device doesn’t have a pre-allocated ID on the IDs list, this register should be read as 0. | | |
| **2.1.2** | **R\_TIMESTAMP\_SECOND – System timestamp: seconds** |
|  | |  |  | | --- | --- | | Add: 008 | SECONDS [31:0] | | Contains the current system timestamp in seconds. The default value is 0 (ZERO) and will increment one unit for each second. | | |
| **2.1.3** | **R\_TIMESTAMP\_MICRO – System timestamp: microseconds** |
|  | |  |  | | --- | --- | | Add: 009 | USECONDS [15:0] [7:0] | | Contains the microseconds count within each second. Each LSB corresponds to 32 µseconds. The maximum value is 31249. | | |

|  |  |
| --- | --- |
| **2.1.4** | **R\_OPERATION\_CTRL – Configuration the operation mode** |
|  | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | Add: 010 | **ALIVE\_EN** | **OPLEDEN** | **VISUALEN** | **MUTE\_RPL** | **DUMP** | - | **OP\_MODE [1:0]** | | | Default | 1 | 1 | 1 | 0 | 0 | 0 | 0 | | | Mandatory | Optional | Optional | Optional | Optional | Optional | - | a) | | | a) Standby Mode and Active Mode are mandatory. Speed Mode is optional.   * **Bits 1:0 – OP\_MODE [1:0]: Operation Mode**   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.   * **Bit 3 – DUMP**   When written to 1, the device adds the content of all registers to the streaming buffer.  This bit is always read as 0.   * **Bit 4 – MUTE\_RPL**   If equals to 1, the Replies to all the Commands are muted, i.e., will not be sent by the device.   * **Bit 5 – VISUALEN**   If equals to 1, the visual indications, typically LEDs, available on the device will operate. If equals to 0, all the visual indications should turn off.   * **Bit 6 – OPLEDEN**   If equal to 1, the LED present on the device will indicate the Operation Mode selected.  **Table 2-2. Operation Mode**   |  |  | | --- | --- | | OP\_MODE [1:0] | Configuration | | 0 | Standby Mode. The device has all the Events turned off. | | 1 | Active Mode. The device turn ON the Events detection. Only the enabled Events will be operating. | | 2 | Reserved. | | 3 | Speed Mode. The device enters into Speed Mode. |   **Table 2-3. LED toggle indication**   |  |  | | --- | --- | | Interval in seconds | 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. | | | | | | | | | |  * **Bit 7 – ALIVE\_EN**   If equal to 1, the device sends an Event with the R\_TIMESTAMP\_SECONDS content each second. This allows the Controller and/or the user to check that the device is alive. Although this is an optional feature, it’s strongly suggested that the device should implement it. |
| **2.1.5** | **R\_RESET\_DEV – Reset device and save non-volatile registers** |
|  | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | Add: 011 | **BOOT\_EE** | **BOOT\_DEF** | - | - | - | **SAVE** | **RST\_EE** | **RST\_DEF** | | * **Bits 0 – RST\_DEF**   If this bit is written t to 1, the device resets and boots with all the registers, both Common and Application registers, with the default values. EEPROM will be erased, and the default values will be the permanent boot option.  This bit is always read as 0.   * **Bits 1 – RST\_EE**   If this bit is written to 1, the device resets and boots with all his registers, both Common and Application registers, with the values saved on the non-volatile memory, usually an EEPROM. The EEPROM values will be the permanent boot option.  Should not be possible to write to this bit if the non-volatile memory is empty.  This bit is always read as 0.   * **Bits 2 – SAVE**   If this bit is written to 1, the device saves all the non-volatile registers (Common and Application) to the internal non-volatile memory and boots. The non-volatile memory should be the permanent boot option.  This bit is always read as 0.   * **Bits 3 –** **NAME\_TO\_DEFAULT**   If this bit is written to 1, the device boots with the default name.  This bit is always read as 0.   * **Bits 6 – BOOT\_DEF**   It is a state bit (read only). Indicates that the device booted with the default register values.   * **Bits 7 – BOOT\_EE**   It is a state bit (read only). Indicates that the device booted with the register values saved on the EEPROM.  **Note:** To avoid unexpected behaviors, one bit at a time should be written to register R\_RESET\_DEV. | | | | | | | | | |
| **2.1.6** | **R\_DEVICE\_NAME** |
|  | |  | | --- | | An array of 25 bytes that should contain the device name.  The last byte and the bytes not used must be equal to 0.  This register is non-volatile. The device will reset if this register is written. | |
| **2.1.7** | **R\_SERIAL\_NUMBER – Device’s serial number** |
|  | |  |  | | --- | --- | | Add: 000 | SERIAL\_NUMBER [15:0] | | 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. The device will reset after the second write command is sent. | | |
| **2.1.8** | **R\_CLOCK\_CONFIG – Synchronization clock configuration** |
|  | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | Add: 014 | **CLK\_LOCK** | **CLK\_UNLOCK** | - | **GEN\_ABLE** | **REP\_ABLE** | - | **CLK\_GEN** | **CLK\_REP** | | * **Bits 0 – CLK\_REP**   If this bit is written t 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 into Clock Output.  Writing 1 to this bit also unlocks the Harp Synchronization Clock.  This bit is read as 1 if the device is repeating the Harp Synchronization Clock.   * **Bits 1 – CLK\_GEN**   If this bit is written to 1, the device resets will generate the Harp Synchronization Clock in the Clock Output connector, if available. The Clock Input is ignored.  This bit is read as 1 if the device is generating the Harp Synchronization Clock.   * **Bits 3 – REP\_ABLE**   This is a read only bit. Written to this bit will not have any effect.  The bit is equal to 1 if the device can repeat the Harp Timestamp Clock.   * **Bits 4 – GEN\_ABLE**   This is a read only bit. Written to this bit will not have any effect.  The bit is equal to 1 if the device can generate the Harp Timestamp Clock.   * **Bits 6 – CLK\_UNCLOCK**   If this bit is written to 1, the device will unlock the timestamp register counter (register R\_TIMESTAMP\_SECOND) and will accept new timestamp values.  This bit is read as 1 if the timestamp register is unlocked.   * **Bits 7 – CLK\_LOCK**   If this bit is written to 1, the device will lock the current timestamp counter (register R\_TIMESTAMP\_SECOND) and will not be able to accept new timestamp values.  This bit is read as 1 if the timestamp register is locked.  **Note:** The device always wakes up in the unlock state. | | | | | | | | | |
| **2.1.6** | **R\_TIMESTAMP\_OFFSET** |
|  | |  | | --- | | 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 500us each unit.  This register is non-volatile. | |

|  |  |
| --- | --- |
|  | **Version Control** |
|  | **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**. |