**Document Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| V3 | 3.9.18 | Ziv | Added: Add debug working mode fields to [Setup Message](#_Setup_Message) and [Get Setup Config Response](#_Get_Setup_Config) |
| V4 | 12.9.18 | Evgeny | Added:   * [I2C section](#_I2C_communication_with) |
| V5 | 28.10.18 | Ziv | Added:   * [Error Message](#_Error_Message) * Default values per message |
| V6 | 21.11.18 | Ziv | Added:   * Add new script arguments to [Setup Message](#_Setup_Message) and [Get Setup Config Response](#_Get_Setup_Config). * Add is\_rotated field to [setWarningMessage](#_Set_Warning_Message) and [GetWarningConfigResponse](#_Get_Warning_Config) * Add [ST\_NACK](#_Acknowledge) |
| V7 | 26.11.18 | Ziv | Added:   * Add is\_rotated and cycle\_counter fields to [getWarningResponse](#_Get_Warning_Response) |

Human Detection Protocol

Contents

[ Add debug working mode fields to Setup Message and Get Setup Config Response 1](#_Toc529173492)

[General 4](#_Toc529173493)

[HD Application 4](#_Toc529173494)

[Detection 4](#_Toc529173495)

[General info 4](#_Toc529173496)

[Warnings 4](#_Toc529173497)

[Detection Algorithm 4](#_Toc529173498)

[Examples – 5](#_Toc529173499)

[Obstruction 6](#_Toc529173500)

[General info 6](#_Toc529173501)

[HD HW 7](#_Toc529173502)

[Protocol 7](#_Toc529173503)

[HeartBeat 7](#_Toc529173504)

[Error Message 7](#_Toc529173505)

[Protocol flow 8](#_Toc529173506)

[Protocol Main Structure 9](#_Toc529173507)

[Setup 9](#_Toc529173508)

[Baud Rate: ? 9](#_Toc529173509)

[Byte Size: 9](#_Toc529173510)

[MCU to HD Messages 10](#_Toc529173511)

[Setup Message 10](#_Toc529173512)

[Set Warning Message 12](#_Toc529173513)

[Remove Warning Message 14](#_Toc529173514)

[Remove All Warnings Message 14](#_Toc529173515)

[Remove All Warnings Except Default Message 15](#_Toc529173516)

[Set **Warning** to Default Message 15](#_Toc529173517)

[Set Power Message 16](#_Toc529173518)

[Get Warning Message 16](#_Toc529173519)

[Get Warning Config Message 17](#_Toc529173520)

[Get Setup Config Message 17](#_Toc529173521)

[Get Status Message 18](#_Toc529173522)

[HD to MCU Messages 19](#_Toc529173523)

[Get Warning Response 19](#_Toc529173524)

[Get Warning Config Response 20](#_Toc529173525)

[Get Setup Config Response 21](#_Toc529173526)

[Get Status Response 23](#_Toc529173527)

[Acknowledge 24](#_Toc529173528)

[I2C communication with discovery boards bridge 24](#_Toc529173529)

[1. Connection: 25](#_Toc529173530)

[2. Master communication protocol: 25](#_Toc529173531)

[Appendix A – CheckSum Byte 26](#_Toc529173532)

[Check sum byte used to detect communication errors. 26](#_Toc529173533)

# General

## HD Application

The purpose of the HD application is to detect objects and notify the robot.

It also notifies the robot when its camera is being obstructed by an object and other vision metrics.

### Detection

#### General info

The HD application uses the OpenCV’s Deep Neural Network (DNN) module while using the SSD300 method in order to get a high performance in terms of time and confidence.

The current fps achieved on a Raspberri Pi-3 for detection is ~1.4fps.

When adding image capturing and other tasks the fps drops to ~1.24fps.

#### Warnings

The robot will define a set of warnings (max. 16) for the HD to detect.

Each warning holds: an ID, a single ROI (a polygon of 4 points) and other parameters such as confidence, Class object to detect, min out of max hits and min and max size of the object. (see [Set Warning Message](#_1ksv4uv))

If the HD detects the requested object with its requested parameters, the bit representing this warning will be 1, otherwise 0.

**Note:**

* A warning can contain an overlapping polygon with any other warning.

#### Detection Algorithm

This is the basic algorithm calculated for each frame:

*foreach (warning):*

*{*

*If warning.is\_rotated != self.is\_frame\_rotated:*

*Continue;*

*foreach (detected\_object):*

*if (detected\_object.object\_class == warning.object\_class)* ***and***

*if (detected\_object.confidence > warning.confidence)* ***and***

*if (detected\_object.polygon* ***INTERSECTS*** *warning.polygon)* ***and***

*if (detected\_object.w\_h* ***<*** *warning.max\_w\_h )* ***and***

*if (detected\_object.w\_h* ***>*** *warning.min\_w\_h)* ***and***

*if (warning is hit min out of max times)*

*{*

*then* ***SEND WARNING***

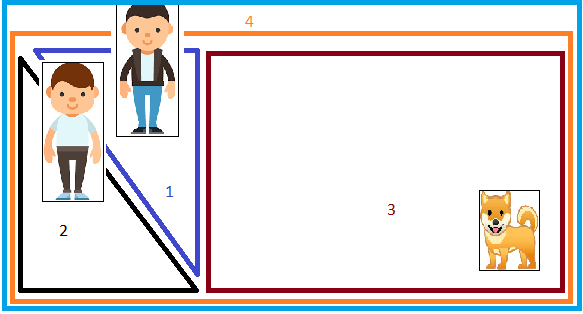
*}*

*}*

**Notes:**

* **Note 1 –** There is no meaning for detecting multiple objects in a Warning - it is consider as a single detection (this is relevant for the counter of each polygon)
* **Notes 2 -** A detected object can appear in multiple warnings
* **Note 3** - Every n frames, the image will be **rotated by 90** degrees in order to detect a person lying down which is not detected normally.

#### Examples –



* The HD received 4 warnings (polygons) to detect and in each warning it needs to detect a person only (there are more parameters but they are ignored in this example)

Result:

* warning 1: ON
* warning 2: ON
* warning 3: OFF
* warning 4: ON

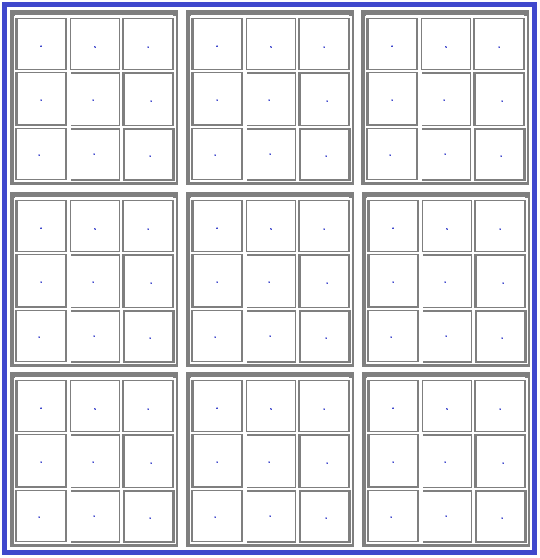
### Obstruction

#### General info

along detection the HD application will also analyze if any part of the camera is being blocked by an object over time (for m out of n times).

basic algo -

* + convert the image to gray
  + Divide the image into tiles - default (3x3)
    - Divide each tile into sub tiles - default (3x3)
      * In each tile - sample the intensity in the center -> intensity[b,g,r]
      * So now we have a list of intensity of all sub tiles
      * Calculate the variance of the list (1 gray channel) ⇒  var(∑gray)
  + IF any tile’s variance is **below** the threshold → tile is obstructed → consider the whole image to be obstructed → Notify the Robot



## HD HW

The protocol connects the new HD vision board to the robot CPU using I2C where the HD vision board is an I2C slave. The HD vision board doesn’t support I2C slave and therefore we are using STM0 CPU as UART to its I2C gateway. The same STM0 CPU will also control the power on/off according to a command from the robot CPU.

# Protocol

Since the communication is synchronous, the **robot which is the master** is always the initiator of the messages while The HD vision will always respond with [Ack/Nack](#_2p2csry) or with a [GetWarningResponse](#_qsh70q) message (see [protocol flow](#_1t3h5sf) below)

There are 2 types of messages:

* messages that write information to the slave
* messages that request to read information from the slave

## HeartBeat

The [GetWarningMessage](#_4i7ojhp) with its response [GetWarningResponse](#_qsh70q) are used as the heartbeat of the system.

The [GetWarningResponse](#_qsh70q) holds information and status of the HD.

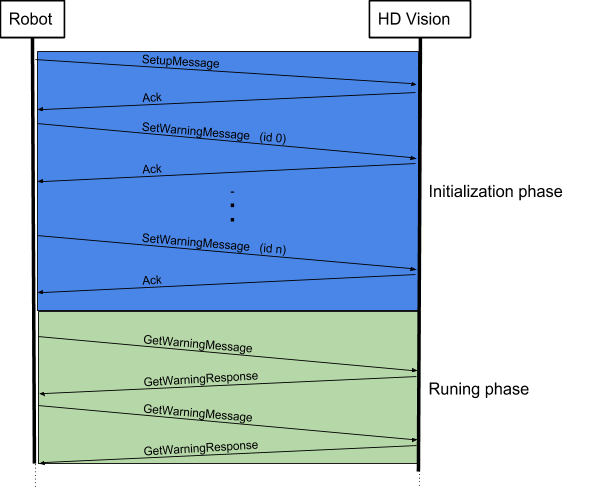
## Error Message

In A case where there is an error in the HD The ST Slave will return the [ST\_NACK](#_Acknowledge) response.

For example – The HD application is down 🡪 The UART buffer between the ST Slave and the HD Application is empty 🡪 There is no data to return 🡪 Therefor a 4 bytes NACK message is returned.

This means the Master must always check the ReadSize value before it sends the Read command in order to read the correct message. For example – if the master expects a 7 bytes message but due to an error the ST

## Protocol flow



## Protocol Main Structure

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Preamble** | **Length** | **Opcode** | **Data** | **Checksum** |
| 8 bit | 8 bit | 8 bit | n bytes | 8 bit |

**Preamble** - Can be used for fragmentation. currently there is no need for it. therefor it is constant = ***0xAA***.

**Length -** Holds the length of the data + 4 bytes (Preamble, Length, Opcode and checksum)

**Data** - holds the data of the message as detailed

**Opcode** - Describes the opcode of the message. There are 4 types of messages:

[**Checksum**](#_3o7alnk) – Calculation should be with the length of the whole message including 4 bytes (Preamble, Length, Opcode and checksum).

## Setup

RS232 Details:

###### Baud Rate: ?

###### Byte Size:

Parity:

Stop Bits:

# MCU to HD Messages

## Setup Message

Set the HD **general** parameters.

This message should be acknowledged upon reception with ‘Acknowledge’ message.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xB1 |
| 4 | rotate\_image\_cycle | Every N cycles rotate the image by 90 deg. and use it for the detection. This is done in order to detect a person laying down which is not detected normally.  e.g. – if rotate\_image\_cycle=10 ➔ every 10 frames the detection will be done on an image rotated by 90 degrees.  Default value - 5 |
| 5-6 | obstruction\_threshold | Threshold of Variance of the intensity. e.g. 1000  Thresholds to evaluate when the camera or part of it is being blocked\obstructed by an object.  Default value - 1000 |
| 7 | no\_visibility\_threshold | Threshold of the intensity. e.g. 63. [0-255]  Thresholds to define the quality of the image  Default value – 63 |
| 8 | medium\_visibility\_threshold | Threshold of the intensity. e.g. 128. [0-255]  Thresholds to define the quality of the image  Default value – 128 |
| 9 | full\_visibility\_threshold | Threshold of the intensity. e.g. 190. [0-255]  Thresholds to define the quality of the image  Default value – 190 |
| 10 | minimun\_obstruction\_hits | Repeatability value thresholds to define obstruction of camera. if any tile is being blocked for min out of max times → then a part of the camera is considered to be blocked.  Default value – 2 |
| 11 | maximum\_obstruction\_hits | Repeatability value thresholds to define obstruction of camera. if any tile is being blocked for min out of max times → then a part of the camera is considered to be blocked.  Default value – 3 |
| 12 | Logging Debug | Sets the logging level. **Used for debugging**  0 – False (Info level)  1 – True (debug level) |
| 13 | Show Images | Displays the images on a remote console using the X11 Server. **Used for debugging**. The polygons warnings and the detected boxes are drawn on these images.  **Note: This slows down the performance of the HD** **significantly!**  0 – False  1 – True |
| 14 | Save Images | Saves the images on the disk. **Used for debugging**. The HD uses 500MB (~1850 images) for saving images and doesn’t exceed this size. When approaching the size limit old images will be deleted.  **Note: This slows down the performance of the HD!**  0 – False  1 – True |
| 15 | Draw Polygons | Draw the detections and the warning's polygons on the images  **Note: This slows down the performance of the HD!**  0 – False  1 – True |
| 16 | Buzzer activate | Activate the HD’s buzzer – used for debug  **Note: This slows down the performance of the HD!**  0 – False  1 – True |
| 17 | Rotate Degree | Rotate the frame – [0-90] deg |
| 18 | Check Sum | [See appendix A](#_3o7alnk) |

## Set Warning Message

Command the HD to create a new Warning Message.

This message should be acknowledged upon reception with ‘Acknowledge’ message.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xB2 |
| 4 | warning\_id | Values – 0 to 15  Default value – 0 |
| 5-20 | Polygon | Each polygon is an array of 4 Points.  Each Point is 4 bytes: x=2 bytes , y=2 bytes.  e.g. – if we want a polygon to cover the whole screen we should set a 300 pixels polygon  Default value –  0x00-0x00-0x00-0x00-0x00-0x00-0x2C-0x01-0x2C-0x01-0x2C-0x01-0x2C-0x01-0x00-0x00 |
| 21 | object\_class | Bit 0 : person  Bit 1 : animal (cat or dog)  Bit 2 : Furniture (sofa, diningtable, chair, tvmonitor)  Bit 3 : toys (aeroplane, boat, cow, train)  Bit 4 : Vehicle (bicycle, motorbike, car,bus)  Bit 5 : reserved  Bit 6 : reserved  Bit 7 : reserved  e.g. - if we set object\_class to 3, then we should encounter in that warning only ‘person’ and ‘animal’  Default value – 0x01 |
| 22-23 | object\_min\_w\_h | Min width and height of an object to detect  Default value – 0x00 0x00 |
| 24-25 | object\_max\_w\_h | Max width and height of an object to detect  Default value – 0x2C 0x01 |
| 26 | minimum\_confidence | Value: 0-100.  Value unit = %  Each detection has a confidence value which represents the  HD will ignore detections below this value  e.g. – 20%  Default value – 0x14 |
| 27 | minimun\_detection\_hits | min out of max hits will be considered as “Warning on”  Default value – 0x01 |
| 28 | maximun\_detection\_hits | min out of max hits will be considered as “Warning on”  Default value – 0x02 |
| 29 | is\_default | This property is relevant to [Set Warnings to Default Message](#_3j2qqm3) and to [Remove All Warnings Except Default Message](#_z337ya)  If True, this warning will not be removed when this message is invoked. Note: this property is ignored when [Remove All Warnings Message](#_2jxsxqh) is invoked  0 – False  1 – True  Default value – 0x01 |
| 30 | is\_rotated | This property is helpful for distinguishing between different warnings settings for lying detected classes  0 – False  1 – True  Default value – 0x00 |
| 31 | Check Sum | [See appendix A](#_3o7alnk) |

## Remove Warning Message

Command the HD to remove a warning by its id.

This message should be acknowledged upon reception with ‘Acknowledge’ message.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xB3 |
| 4 | warning\_id | Values – 0 to 15 |
| 5 | Check Sum | [See appendix A](#_3o7alnk) |

## Remove All Warnings Message

Command the HD to remove all warnings. This message contains no data

This message should be acknowledged upon reception with ‘Acknowledge’ message.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xB4 |
| 4 | Check Sum | [See appendix A](#_3o7alnk) |

## Remove All Warnings Except Default Message

Command the HD to **remove all warnings** except the default warnings. This message contains no data

This message should be acknowledged upon reception with ‘Acknowledge’ message.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xB5 |
| 4 | Check Sum | [See appendix A](#_3o7alnk) |

## Set **Warning** to Default Message

Command the HD to set a specific warning or all warnings to be default.

A Warning with property is\_default = True will not be removed when [Remove All Warnings Except Default Message](#_z337ya) is invoked.

This message should be acknowledged upon reception with ‘Acknowledge’ message.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xB6 |
| 4 | warning\_id | Values – 0 to 15 |
| 5 | all\_warnings | 0 – False  1 – True  If True – warning\_id will be ignored |
| 5 | Check Sum | [See appendix A](#_3o7alnk) |

## Set Power Message

Command the HD to power on/off.

This message should be acknowledged upon reception with ‘Acknowledge’ message.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xB7 |
| 4 | power\_cmd | 0 – power off  1 – power on |
| 5 | Check Sum | [See appendix A](#_3o7alnk) |

## Get Warning Message

Command the HD to reply with the [Get Warning Response](#_qsh70q) message.

This message should be acknowledged upon reception with [Get Warning Response](#_qsh70q) message.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xB8 |
| 4 | Check Sum | [See appendix A](#_3o7alnk) |

## Get Warning Config Message

Command the HD to reply with the [Get Warning Config Response](#_3as4poj) message.

This message should be acknowledged upon reception with [Get Warning Config Response](#_3as4poj) message.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xB9 |
| 4 | warning\_id | Values – 0 to 15 |
| 5 | Check Sum | [See appendix A](#_3o7alnk) |

## Get Setup Config Message

Command the HD to reply with the [Get Setup Config Response](#_1pxezwc) message.

This message should be acknowledged upon reception with [Get Setup Config Response](#_1pxezwc) message.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xBA |
| 4 | Check Sum | [See appendix A](#_3o7alnk) |

## Get Status Message

Command the HD to reply with the [Get Status Response](#_49x2ik5) message.

This message should be acknowledged upon reception with the [Get Status Response](#_49x2ik5) message.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xBB |
| 4 | Check Sum | [See appendix A](#_3o7alnk) |

# HD to MCU Messages

## Get Warning Response

Status message of the HD. Also used as the “heartbeat” of the system.

Messages from HD to MCU can’t be acknowledged.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xC1 |
| 4-6 | warning0 :1;  warning1 :1;  warning2 :1;  warning3 :1;  warning4 :1;  warning5 :1;  warning6 :1;  warning7 :1;  warning8 :1;  warning9 :1;  warning10 :1;  warning11 :1  warning12 :1;  warning13 :1;  warning14 :1;  warning15 :1;  visibility\_light\_level :2;  visibility\_obstruction :1;  is\_rotated :1;  cycle\_counter :4; | **Warningxx:**  0 – no warning  1 – warning on  **visibility\_light\_level:**  0 - no visibility - under “no\_visibility\_threshold”  1 - between “no\_visibility\_threshold” and “medium\_visibility\_threshold”  2 - between “medium\_visibility\_threshold” and “full\_visibility\_threshold”  3 - full visibility - above “full\_visibility\_threshold”  **visibility\_obstruction:**  0 - no obstruction  1 - is obstructed  **is\_rotated -** is this cycle handling a rotated frame  0 - no obstruction  1 - is obstructed  **cycle\_counter -** a (cyclic) cycle counter to inform of a different cycle  [0-16] |
| 7 | Check Sum | [See appendix A](#_3o7alnk) |

## Get Warning Config Response

Returns the details of a warning. Used for verifying that the HD holds the correct parameters

Messages from HD to MCU can’t be acknowledged.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xC2 |
| 4 | warning\_id | Values – 0 to 15 |
| 5-20 | Polygon | Each polygon is an array of 4 Points.  Each Point is 4 bytes:  x=2 bytes , y=2 bytes ➔ values from 0-300 |
| 21 | object\_class | Bit 0 : person  Bit 1 : animal (cat or dog)  Bit 2 : Furniture (sofa, diningtable, chair, tvmonitor)  Bit 3 : toys (aeroplane, boat, cow, train)  Bit 4 : Vehicle (bicycle, motorbike, car,bus)  Bit 5 : reserved  Bit 6 : reserved  Bit 7 : reserved  e.g. - if we set object\_class to 3, then we should encounter in that warning only ‘person’ and ‘animal’ |
| 22-23 | object\_min\_w\_h | Min width and height of an object to detect |
| 24-25 | object\_max\_w\_h | Max width and height of an object to detect |
| 26 | minimum\_confidence | Value: 0-100.  Value unit = %  Each detection has a confidence value  HD will ignore detections below this value |
| 27 | minimun\_detection\_hits | min out of max hits will be considered as “Warning on” |
| 28 | maximun\_detection\_hits | min out of max hits will be considered as “Warning on” |
| 29 | is\_default | This property is relevant to [Set Warnings to Default Message](#_z337ya)  If True, this warning will not be removed when this message is invoked. Note: this property is ignored when [Remove All Warnings Message](#_2jxsxqh) is invoked  0 – False  1 – True |
| 30 | is\_rotated | This property is helpful for distinguishing between different warnings settings for lying detected classes  0 – False  1 – True  Default value – 0x00 |
| 31 | Check Sum | [See appendix A](#_3o7alnk) |

## Get Setup Config Response

Returns the HD **general** parameters. Used for verifying that the HD holds the correct parameters

Messages from HD to MCU can’t be acknowledged.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xC3 |
| 4 | rotate\_image\_cycle | Every N cycles rotate the image by 90 deg. and use it for the detection. This is done in order to detect a person laying down which is not detected normally.  e.g. – if rotate\_image\_cycle=10 ➔ every 10 frames the detection will be done on an image rotated by 90 degrees. |
| 5-6 | obstruction\_threshold | Threshold of Variance of the intensity. e.g. 1000  Thresholds to evaluate when the camera or part of it is being blocked\obstructed by an object. |
| 7 | no\_visibility\_threshold | Threshold of the intensity. e.g. 63 - [0-255]  Thresholds to define the quality of the image |
| 8 | medium\_visibility\_threshold | Threshold of the intensity. e.g. 128 - [0-255]  Thresholds to define the quality of the image |
| 9 | full\_visibility\_threshold | Threshold of the intensity. e.g. 190 - [0-255]  Thresholds to define the quality of the image |
| 10 | minimun\_obstruction\_hits | Repeatability value thresholds to define obstruction of camera. if any tile is being blocked for min out of max times → then a part of the camera is considered to be blocked. |
| 11 | maximum\_obstruction\_hits | Repeatability value thresholds to define obstruction of camera. if any tile is being blocked for min out of max times → then a part of the camera is considered to be blocked. |
| 12 | Logging Debug | Sets the logging level. **Used for debugging**  0 – False (Info level)  1 – True (debug level) |
| 13 | Show Images | Displays the images on a remote console using the X11 Server. **Used for debugging**. The polygons warnings and the detected boxes are drawn on these images.  **Note: This slows down the performance of the HD** **significantly!**  0 – False  1 – True |
| 14 | Save Images | Saves the images on the disk. **Used for debugging**. The HD uses 500MB (~1850 images) for saving images and doesn’t exceed this size. When approaching the size limit old images will be deleted.  **Note: This slows down the performance of the HD!**  0 – False  1 – True |
| 15 | Draw Polygons | Draw the detections and the warning's polygons on the images  **Note: This slows down the performance of the HD!**  0 – False  1 – True |
| 16 | Buzzer activate | Activate the HD’s buzzer – used for debug  **Note: This slows down the performance of the HD!**  0 – False  1 – True |
| 17 | Rotate Degree | Rotate the frame – [0-90] deg |
| 18 | Check Sum | [See appendix A](#_3o7alnk) |

## Get Status Response

Returns the **versions** of the HD.

Messages from HD to MCU can’t be acknowledged.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xC4 |
| 4-5 | sw\_version | {x.x} ➔ {byte #4.byte #5} |
| 6-7 | hw\_version | {x.x} ➔ {byte #6.byte #7} |
| 8 | Check Sum | [See appendix A](#_3o7alnk) |

## Acknowledge

This message is sent when the Wire system needs to acknowledge the MCU that a certain message was received.

|  |  |  |
| --- | --- | --- |
| **Byte #** | **Description** | **Content** |
| 1 | Preamble | 0xAA |
| 2 | Message Length | total number of bytes in the message including the 4 bytes (Preamble, Length, Opcode and checksum) |
| 3 | Message Opcode | 0xD1 – Ack  0xD2 – Nack  0xD3 – ST\_Nack |
| 4 | Check Sum | [See appendix A](#_3o7alnk) |

## I2C communication with discovery boards bridge

# Connection:

Each discovery module exposes 2 interfaces: UART and I2C

**I2C pins:** PB6 - SCL, PB7 - SDA. UART pins: PA9 - TX, PA10 - RX

**UART baud rate:** 38400

# Master communication protocol:

Commands:

1. Write request - master initiates write operation. Response: none

|  |  |  |
| --- | --- | --- |
| Command - 1byte | Size - 2 bytes | Payload |
| 0x34 | Max 0xFFFF |  |

1. Read curr receive buffer size - master requests current buffer size. Response: 2 bytes buffer size

|  |
| --- |
| Command - 1byte |
| 0x13 |

1. Write request - master initiates write operation. Response: none

|  |  |
| --- | --- |
| Command - 1byte | Size - 2 bytes |
| 0x12 | Max 0xFFFF |

## 

# Appendix A – CheckSum Byte

### Check sum byte used to detect communication errors.

Use the following c language code (if needed) to calculate the check sum byte:

packet – pointer to the packet

length – number of bytes in the packet without the check sum byte

Important: don’t forget the NOT operation on the returned sum at the end of the function

**byte Check\_sum(byte\* packet, short length)**

**{**

**short i=0;**

**long sum=0;**

**for (i=0;i<length;i++)**

**sum += packet[i];**

**return((byte)(~sum));**

**}**