

HUSKYLENS Protocol

Version:

0.5.1

Designer:

Angelo (Angelo.qjiao@dfrobot.com)

Init parameter:

Serial

- Serial Mode: 9600 (bps) 8N1
- Address in Protocol: 0x11

I2C

- I2C Speed: 100 kbit/s
- I2C Address: 0x32
- Address in Protocol: 0x11

Communication Command Frame Format:

Header	Header 2	Address	Data Length	Command	Data	Checksum
0x55	0xAA	1 byte	1 byte	1 byte	Data 1 ~ Data n	1 byte

Hex	Function
0x55	Header
0xAA	Header 2
0x11	Address
0x0A	Data Length
0x2A	Command is 0x2A
0x2C	Data[0]
0x01	Data[1]
0xC8	Data[2]
0x00	Data[3]
0x0A	Data[4]
0x00	Data[5]
0x14	Data[6]
0x00	Data[7]
0x01	Data[8]
0x00	Data[9]
0x58	Checksum, Sum all and only use low byte (Low Byte of 0x258 = 0x55 + 0xAA + 0x11 + 0x0A + 0x2A + 0x2C + 0x01 + 0xC8 + 0x00 + 0x0A + 0x00 + 0x14 + 0x00 + 0x01 + 0x00)

General concept:

ID Meaning:

ID	Means
1	The first learned item is detected
2	The second learned item is detected
XXX	The XXXth learned item is detected
0	Item is detected but not learned, like unlearned faces block in grey color.

Protocol flow:

hosts	HUSKYLENS
COMMAND_REQUEST_KNOCK====>	
	<===COMMAND_RETURN_OK
COMMAND_REQUEST====>	
	<=== COMMAND_RETURN_INFO
	<=== COMMAND_RETURN_BLOCK
	<=== COMMAND_RETURN_BLOCK
	<=== COMMAND_RETURN_ARROW
COMMAND_REQUEST_BLOCKS====>	
	<=== COMMAND_RETURN_INFO
	<=== COMMAND_RETURN_BLOCK
	<=== COMMAND_RETURN_BLOCK
COMMAND_REQUEST_ARROWS====>	
	<=== COMMAND_RETURN_INFO
	<=== COMMAND_RETURN_ARROW
COMMAND_REQUEST_ALGORITHM====>	
	<===COMMAND_RETURN_OK
COMMAND_REQUEST_ALGORITHM====>	
COMMAND_REQUEST_ALGORITHM====>	
	<===COMMAND_RETURN_BUSY
	<===COMMAND_RETURN_OK

Commands List:

COMMAND_REQUEST (0x20):

Request all blocks and arrows from HUSKYLENS.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x20	0x30

COMMAND_REQUEST_BLOCKS (0x21):

Request all blocks from HUSKYLENS.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x21	0x31

COMMAND_REQUEST_ARROWS (0x22):

Request all arrows from HUSKYLENS.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x22	0x32

COMMAND_REQUEST_LEARNED (0x23):

Request all learned blocks and arrows (ID >=1) from HUSKYLENS.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x23	0x33

COMMAND_REQUEST_BLOCKS_LEARNED (0x24):

Request all learned blocks (ID >=1) from HUSKYLENS.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x24	0x34

COMMAND_REQUEST_ARROWS_LEARNED (0x25):

Request all learned arrows (ID >=1) from HUSKYLENS.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x25	0x35

COMMAND_REQUEST_BY_ID (0x26):

Request all blocks and arrows by given ID (Here is 0x01) from HUSKYLENS.

Header	Header 2	Address	Data Length	Command	Data	Checksum
0x55	0xAA	0x11	0x02	0x26	0x01 0x00	0x39

Data details:

Data	Function
0x01	Given ID, see ID Details above (Low Byte of 1 = 0x0001)
0x00	Given ID, see ID Details above (High Byte of 1 = 0x0001)

COMMAND_REQUEST_BLOCKS_BY_ID (0x27):

Request all blocks by given ID (Here is 0x01) from HUSKYLENS.

Header	Header 2	Address	Data Length	Command	Data	Checksum
0x55	0xAA	0x11	0x02	0x27	0x01 0x00	0x3A

Data details:

Data	Function
0x01	Given ID, see ID Details above (Low Byte of 1 = 0x0001)
0x00	Given ID, see ID Details above (High Byte of 1 = 0x0001)

COMMAND_REQUEST_ARROWS_BY_ID (0x28):

Request all arrows by given ID (Here is 0x01) from HUSKYLENS.

Header	Header 2	Address	Data Length	Command	Data	Checksum
0x55	0xAA	0x11	0x02	0x28	0x01 0x00	0x3B

Data details:

Data	Function
0x01	Given ID, see ID Details above (Low Byte of 1 = 0x0001)
0x00	Given ID, see ID Details above (High Byte of 1 = 0x0001)

COMMAND_RETURN_INFO (0x29):

When HUSKYLENS Receives the command above, HUSKYLENS will return this info first and then return the arrows and blocks.

Header	Header 2	Address	Data Length	Command	Data	Checksum
0x55	0xAA	0x11	0x0A	0x29	0x01 0x00 0x01 0x00 0x05 0x00 0x00 0x00 0x00 0x00	0x4A

Data details:

Data	Function
0x01	numbers of blocks and arrows from HUSKYLENS (Low Byte of 1 = 0x0001)
0x00	numbers of blocks and arrows from HUSKYLENS (High Byte of 1 = 0x0001)
0x01	numbers of IDs HUSKYLENS have learned (Low Byte of 1 = 0x0001)
0x00	numbers of IDs HUSKYLENS have learned (High Byte of 1 = 0x0001)
0x05	current frame number (Low Byte of 5 = 0x0005)
0x00	current frame number (High Byte of 5 = 0x0005)
0x00	reserved
0x00	reserved
0x00	reserved
0x00	reserved

COMMAND_RETURN_BLOCK(0x2A):

After HUSKYLENS returns info, HUSKYLENS will return the blocks like this:

Header	Header 2	Address	Data Length	Command	Data	Checksum
0x55	0xAA	0x11	0x0A	0x2A	0x2C 0x01 0xC8 0x00 0x0A 0x00 0x14 0x00 0x01 0x00	0x58

Data details:

Hex	Function
2C	X Center of Block (Low Byte of 300 = 0x012C, range:0-319)
01	X Center of Block (High Byte of 300 = 0x012C, range:0-319)
C8	Y Center of Block (Low Byte of 200 = 0x00C8, range:0-239)
00	Y Center of Block (High Byte of 200 = 0x00C8, range:0-239)
0A	Width of Block (Low Byte of 10 = 0x000A, range:0-319)
00	Width of Block (High Byte of 10 = 0x000A, range:0-319)
14	Height of Block (Low Byte of 20 = 0x0014, range:0-239)
00	Height of Block (High Byte of 20 = 0x0014, range:0-239)
01	ID, see ID Meaning above (Low Byte of 1 = 0x0001)
00	ID, see ID Meaning above (High Byte of 1 = 0x0001)

COMMAND_RETURN_ARROW(0x2B):

After HUSKYLENS returns info, HUSKYLENS will return the arrows like this:

Header	Header 2	Address	Data Length	Command	Data	Checksum
0x55	0xAA	0x11	0x0A	0x2B	0x2C 0x01 0xC8 0x00 0x0A 0x00 0x14 0x00 0x01 0x00	0x5A

Data details:

Data	Function
2C	X Origin of Arrow (Low Byte of 300 = 0x012C, range:0-319)
01	X Origin of Arrow (High Byte of 300 = 0x012C, range:0-319)
C8	Y Origin of Arrow (Low Byte of 200 = 0x00C8, range:0-239)
00	Y Origin of Arrow (High Byte of 200 = 0x00C8, range:0-239)
0A	X Target of Arrow (Low Byte of 10 = 0x000A, range:0-319)
00	X Target of Arrow (High Byte of 10 = 0x000A, range:0-319)
14	Y Target of Arrow (Low Byte of 20 = 0x0014, range:0-239)
00	Y Target of Arrow (High Byte of 20 = 0x0014, range:0-239)
01	ID, see ID Meaning above (Low Byte of 1 = 0x0001)
00	ID, see ID Meaning below (High Byte of 1 = 0x0001)

COMMAND_REQUEST_KNOCK(0x2C):

Used for test connection with HUSKYLENS. When HUSKYLENS received this command, HUSKYLENS will return COMMAND_RETURN_OK.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x2C	0x3C

COMMAND_REQUEST_ALGORITHM(0x2D):

When HUSKYLENS receives this command, HUSKYLENS will change the algorithm by the Data. And will return COMMAND_RETURN_OK.

Header	Header 2	Address	Data Length	Command	Data	Checksum
0x55	0xAA	0x11	0x02	0x2D	0x01 0x00	0x40

Data details:

Data	Function
0x01	Change to ALGORITHM_OBJECT_TRACKING (Low Byte of 1 = 0x0001)
0x00	Change to ALGORITHM_OBJECT_TRACKING (High Byte of 1 = 0x0001)

Data's correspondent algorithm:

Data	Algorithm
0x00 0x00	ALGORITHM_FACE_RECOGNITION
0x01 0x00	ALGORITHM_OBJECT_TRACKING
0x02 0x00	ALGORITHM_OBJECT_RECOGNITION
0x03 0x00	ALGORITHM_LINE_TRACKING
0x04 0x00	ALGORITHM_COLOR_RECOGNITION
0x05 0x00	ALGORITHM_TAG_RECOGNITION
0x06 0x00	ALGORITHM_OBJECT_CLASSIFICATION

COMMAND_RETURN_OK(0x2E):

HUSKYLENS will return OK, if HUSKYLENS receives COMMAND_REQUEST_ALGORITHM, COMMAND_REQUEST_KNOCK.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x2E	0x3E

###

COMMAND_REQUEST_CUSTOMNAMES (0x2F):

Set a custom name for a learned object.

Header	Header 2	Address	Data Length	Command	Data	Checksum
0x55	0xAA	0x11	0x07	0x2F	0x01 0x05 0x54 0x45 0x53 0x54 0x00	0x8C

Data details:

Name: **TEST**

Data	Function
0x01	Given ID of object
0x05	Length of name + 1
0x54	HEX Value of "T"
0x45	HEX Value of "E"
0x53	HEX Value of "S"
0x54	HEX Value of "T"
0x00	0 Value to mark the end of the name

COMMAND_REQUEST_PHOTO (0x30):

Save a photo on the HuskyLens SD Card.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x30	0x40

COMMAND_REQUEST_SEND_KNOWLEDGES (0x32):

Save the current algorithms model to the SD Card in the following file format
"AlgorithmName_Backup_FileNum.conf"

Header	Header 2	Address	Data Length	Command	Data	Checksum
0x55	0xAA	0x11	0x002	0x32	0x01 0x00	0x45

Data details:

FileNum: 1

Example File Name : **LineTracking_Backup_1.conf**

Data	Function
0x01	LOW Byte of FileNum (1 = 0x0001)
0x00	HIGH Byte of FileNum (1 = 0x0001)

COMMAND_REQUEST_RECEIVE_KNOWLEDGES (0x33):

Load a model file from the SD Card to the current algorithm and refresh the algorithm. The loaded file will be the following format "AlgorithmName_Backup_FileNum.conf"

Header	Header 2	Address	Data Length	Command	Data	Checksum
0x55	0xAA	0x11	0x02	0x33	0x01 0x00	0x46

Data details:

FileNum: 1

Example File Name : **LineTracking_Backup_1.conf**

Data	Function
0x01	LOW Byte of FileNum (1 = 0x0001)
0x00	HIGH Byte of FileNum (1 = 0x0001)

COMMAND_REQUEST_CUSTOM_TEXT (0x34):

Place a string of text (less than 20 characters) on top of the HuskyLens UI. The position of the texts (X,Y) coordinate is the top left of the text box.

You can have at most 10 custom texts on the UI at once, and if you continue adding texts you will replace previous texts in a circular fashion. For example, if you enter 10 texts you will fill the text buffer. If you then insert a new text object, you will overwrite the first text position (textBuffer[0]). Inserting another new text object will overwrite the second text position (textBuffer[1]).

Each text is uniquely identified by its (X,y) coordinate, so you can replace the text string at a (X,Y) coordinate instead of adding a new text object. For example, if you insert "TEST_1" at (120,120) and then later submit "TEST_2" at (120,120), you will replace the string "TEST_1" with "TEST_2" and maintain an overall text count of 1.

Header	Header 2	Address	Data Length	Command	Data	Checksum
0x55	0xAA	0x11	0x0A	0x34	0x06 0x00 0x78 0x78 0x54 0x45 0x53 0x54 0x5F 0x31	0x14

Data details:

Text: "TEST_1"

Coordinate : (120,120) or (0x78,0x78)

Data	Function
0x06	Length of String ("TEST_1")
0x00	X Flag (if X>=255, this byte should be set as 0xFF)
0x78	X Cordinate (if X Flag is 0xFF, this value should X % 255)
0x78	Y Cordinate
0x54	HEX Value of "T"
0x45	HEX Value of "E"
0x53	HEX Value of "S"
0x54	HEX Value of "T"
0x5F	HEX Value of "_"
0x31	HEX Value of "1"

COMMAND_REQUEST_CLEAR_TEXT (0x35):

Clear and delete all custom UI texts from the screen.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x35	0x45

COMMAND_REQUEST_LEARN (0x36):

Learn the current recognized object on screen with a chosen **ID**

Header	Header 2	Address	Data Length	Command	Data	Checksum
0x55	0xAA	0x11	0x02	0x36	0x01 0x00	0x49

Data details:

Data	Function
0x01	Given ID, see ID Details above (Low Byte of 1 = 0x0001)
0x00	Given ID, see ID Details above (High Byte of 1 = 0x0001)

COMMAND_REQUEST_FORGET (0x37):

Forget learned objects for the current running algorithm.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x37	0x47

COMMAND_REQUEST_SAVE_SCREENSHOT (0x39):

Save a screenshot of the current UI to the HuskyLens SD Card.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x39	0x49

COMMAND_REQUEST_IS_PRO (0x3B):

Check what model your HuskyLens is.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x3B	0x4b

COMMAND_RETURN_IS_PRO (0x3B):

The return value for the IS_PRO command will send a 1 (logical True) if it is a pro model and 0 (logical false) if it is the standard model.

Header	Header 2	Address	Data Length	Command	Data	Checksum
0x55	0xAA	0x11	0x02	0x3B	0x01 0x00	0x4E

Data details:

Return Value -> 1 (Pro Model)

Data	Function
0x01	LOW Byte of return value
0x00	HIGH Byte of return value

COMMAND_REQUEST_FIRMWARE_VERSION(0x3C):

COMMAND_RETURN_BUSY(0x3D):

HUSKYLENS will return Busy, if send multiple command and not wait for ok.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x3D	0x4D

COMMAND_RETURN_NEED_PRO(0x3E):

HUSKYLENS will return need pro, if send pro only command to huskylens.

Header	Header 2	Address	Data Length	Command	Checksum
0x55	0xAA	0x11	0x00	0x3E	0x4E