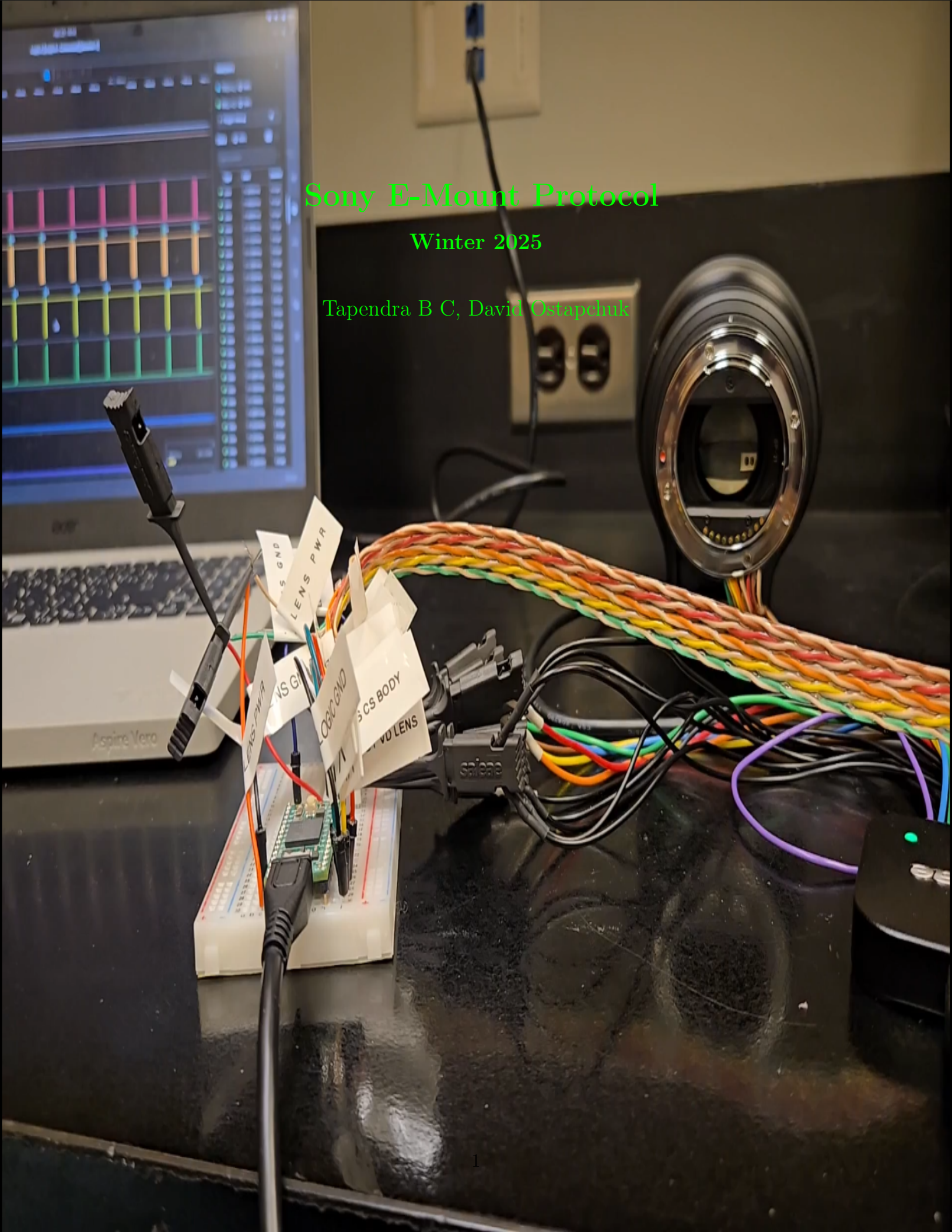
Sony E-Mount Protocol

Winter 2025

Tapendra B C, David Ostapchuk



Contents

1 Disclamer **4**

2 Hardware Used for Reverse Engineering **4**

3 Setup **4**

4 Pinout **5**

5 Schematics **6**

6 Framing **7**

7 Message Format **7**

8 Communication Protocol **9**
- 8.1 Link Establishment . . . 9
- 8.2 Speed Negotiation . . . 10
- 8.3 Initialization . . . 10
- 8.4 Regular Messaging . . . 11
- 8.5 Shut Down . . . 12

9 Message Types **13**
- 9.1 Speed Negotiation Messages . . . 13
- 9.2 Initialization Messages . . . 13
- 9.3 Regular Messages . . . 14

10 Shutdown Messages **14**

11 Message Body Format **14**
- 11.1 0x03 type body . . . 14
- 11.2 0x04 type body . . . 16
- 11.3 0x05 type body . . . 17
- 11.4 0x06 type body . . . 17

12 Aperture Mapping **18**

13 Focus Mapping **18**

14 Software Design **21**

15 Flow Meter Mealy Machine Design **21**
- 15.1 States (Q): . . . 21
- 15.2 Input Alphabet (Σ): . . . 21
- 15.3 Output Alphabet (Λ): . . . 22
- 15.4 Transition Function ($\delta : Q \times \Sigma \rightarrow Q$): . . . 22

15.5 Output Function ($G : Q \times \Sigma \rightarrow \Lambda$): 22
15.6 Accepting States (F) : 22
15.7 Transition Table 22

16 References: **23**

17 Appendix **24**
17.1 Speed Negotiation, Full Payload 24
17.2 Initialization, Full Payload 24
17.3 Shutdown, Full Payload 28

1 Disclamer

This documentation is based on reverse engineering efforts. While care has been taken to ensure its accuracy, it is provided as is without any guarantees of correctness or completeness. Use it at your own risk. No warranties, express or implied, are provided, and the authors accept no liability for any consequences arising from its use.

2 Hardware Used for Reverse Engineering

- Sony α 7 R III Camera.
- Macro Extension Tube.
- Sigma 15MM F1.4 DG DN Diagonal FISHEYE Lens.
- Saleae Logic 8 Logic Analyzer.
- Teensy 4.0 Microcontroller.

3 Setup

The Extension Tube was opened and the ribbon cable was soldered to the pins. The side of the extension tube was cut with a box cutter to feed the wires outside. A piece of tape was wrapped around the pins to prevent them from making contact with the lens mount flange, because it is conductive.

Figure 1: Modified extension tube

Figure 2: Inside of a modified extension tube.

4 Pinout

The pinout I present here is for a Sony E-Mount conpatible Lens. The naming are controller/Body centric.

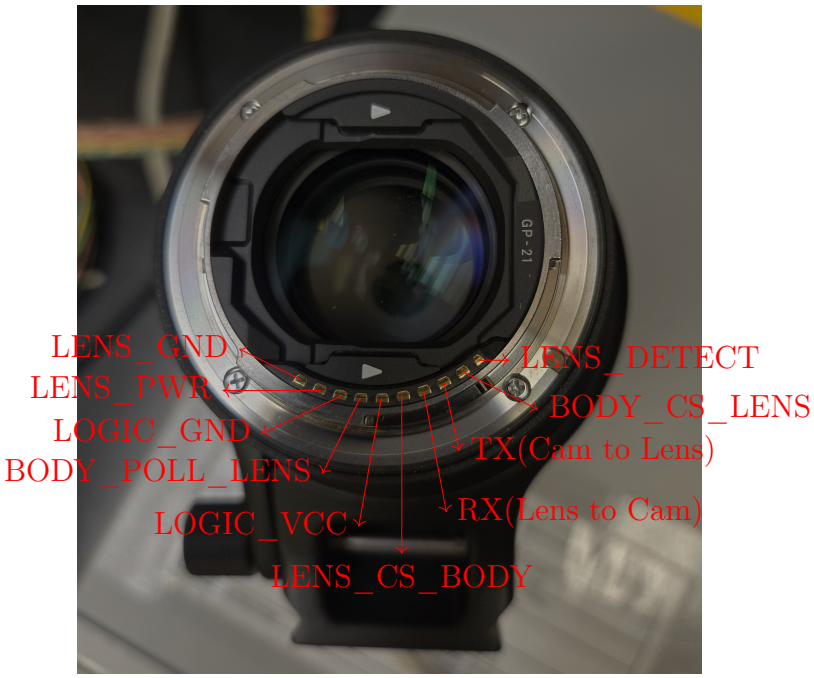


Figure 3: Pinout

Pin Name	Description
LENS_GND	Lens motor GND.
LENS_PWR	+5V for powering motors.
LOGIC_GND	Lens logic GND.
BODY_POLL_LENS	A polling signal from the camera body to the lens that triggers the transmission of aperture and focus data from the lens to the camera. The signal idles high and generates periodic low pulses at 60 Hz.
LOGIC_VCC	3.3V logic power.
LENS_CS_BODY	Handshaking line from lens to the camera body. The handshaking lines are also used for link establishment and behave bit differently, see 8.1.
RX	UART RX. Serial data from the lens to the camera. (This is controller/camera centric naming). The lens sends the data during LENS_CS_BODY pulse.
TX	UART TX. Serial data from camera to the Lens. (This is controller/camera centric naming). The BODY_CS_LENS is pulled high when sending a message to the lens.
BODY_CS_LENS	Handshaking line from camera body to the Lens. When this line is pulsed, the lens sends a LENS_CS_BODY pulse back as an acknowdgement.
LENS_DETECT	Weakly pulled high by the camera. Grounded to LOGIC _GND by the lens. Used by the camera to detect the presence of a lens.

Table 1: Pin Description

5 Schematics

The following is a schematic for controlling the lens. The two transistors and the LENS_DETECT line below can be skipped but we used them to make our system more robust.

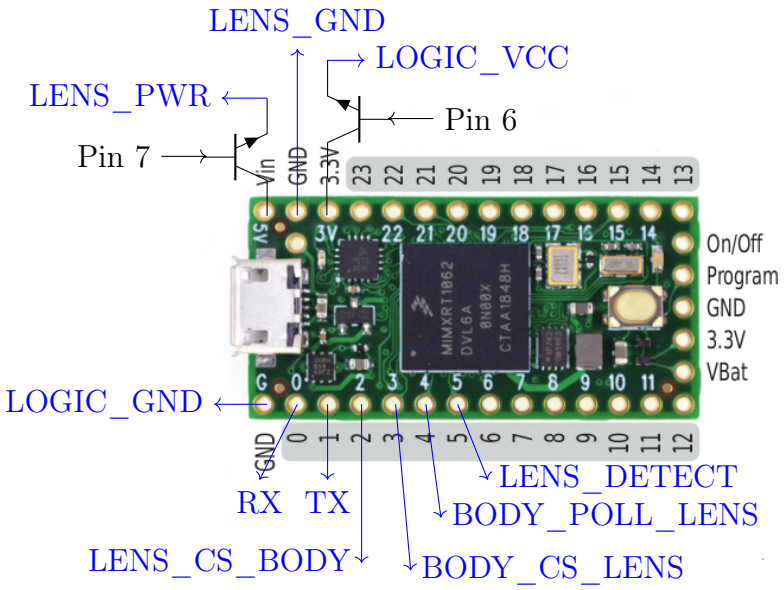


Figure 4: Controller Schematics(Pretending to be a Camera)

6 Framing

The frame is UART 8N1, LSB first.

7 Message Format

Let M be the message of length n bytes sent to or from the Lens. Then the bytes in M are laid out in the following way:

Byte	Name	Value	Description
0	SOM	`0xF0`	Start of the Message.
1	MESSAGE_LENGTH_L	n&`0xFF`	Low byte of Message Length.
2	MESSAGE_LENGTH_H	n»8	High byte of Message Length.
3	MESSAGE_CLASS	`0x01` or `0x02`	Message Class. `0x02` during initialization and `0x01` during normal communication.
4	SEQUENCE_NUMBER	...	Incremented by the Lens on each poll by one. Sequence number is `0x01` for first regular message sent by the lens. Camera can overwrite the sequence number by sending `0x03` and `0x04` type message of sequence number s' and the lens will next return sequence number s'+1. Lens resets the sequence number after `0xEF`. Doesn't want it to be `0xF0`?
5	MESSAGE_TYPE	see section 9	One of the Message Types from section 9
6 to n-4	MESSAGE_BODY	...	See Section 11.
n-3	CHECKSUM_L	Checksum&`0xFF`	Low byte of Checksum. Where, M is message and $Checksum = \sum_{i=1}^{n-4} M[i]$
n-2	CHECKSUM_H	Checksum»8	High byte of Checksum.
n-1	EOM	`0x55`	End of the Message.
>n-1	GARBAGE	!#$#$	Any bytes after EOM are ignored.

The Header includes

1. SOM
2. MESSAGE_LENGTH_L
3. MESSAGE_LENGTH_H
4. MESSAGE_CLASS
5. SEQUENCE_NUMBER
6. MESSAGE_TYPE

The Footer includes:

1. CHECKSUM_L
2. CHECKSUM_H
3. EOM

All the messages will be at least $HeaderLength + FooterLength = 6 + 3 = 9$ byte long.

8 Communication Protocol

The camera initiates the communication. The camera and the lens first make sure that the UART lines are powered and the handshaking lines are responding during Link Establishment phase. Secondly, the camera and the lens negotiate the communication speed during speed negotiation phase. Third, the camera homes the focus and aperture, and the camera configure lens to send update on BODY_POLL_LENS pulse during initialization phase. Finally, the camera sends message to update focus and aperture after receiving the update from the lens during regular messaging phase.

8.1 Link Establishment

Figure 5: Link Establishment

During link establishment, the camera and the lens make sure that they are ready to communicate.

- First, the camera pulls BODY_CS_LENS high and waits for about 2s before resetting. If the LENS_CS_BODY goes high in that duration, the camera powers TX and waits until RX goes high.
- The lens holds LENS_CS_BODY high until TX to goes high.
- Once TX goes high, that indicates that the camera is ready to talk so the lens powers RX to indicate that it is ready to talk and pulls the LENS_CS_BODY low.
- RX being powered indicates that the lens is ready to talk, so the camera pulls the BODY_CS_LENS low.

Now the camera can start to communicate. Now the handshaking line act normally, i.e., when you pulse BODY_CS_LENS the lens responds with LENS_CS_BODY pulse.

8.2 Speed Negotiation

After establishing the communication link, the camera and the lens negotiate the speed change from 750K baud rate to 1500K baud rate. This phase is referred to as the speed negotiation phase in this document. After this phase, the camera and the lens will communicate at 1500K baud rate. See full payload in Appendix.

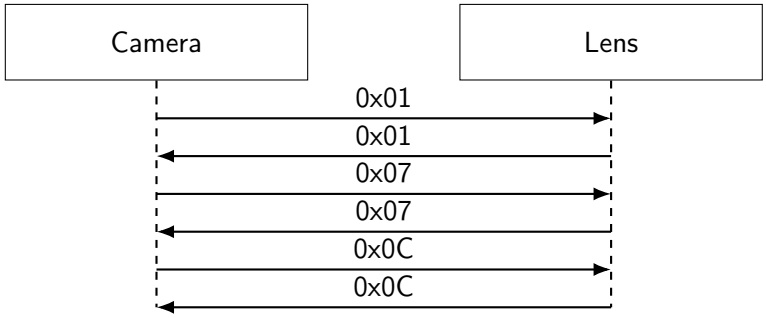


Figure 6: Speed Negotiation sequence

8.3 Initialization

After speed negotiation, the camera commands the lens to home the focus and the aperture. This phase is referred to as an initialization phase in this documentation. See full payload in Appendix.

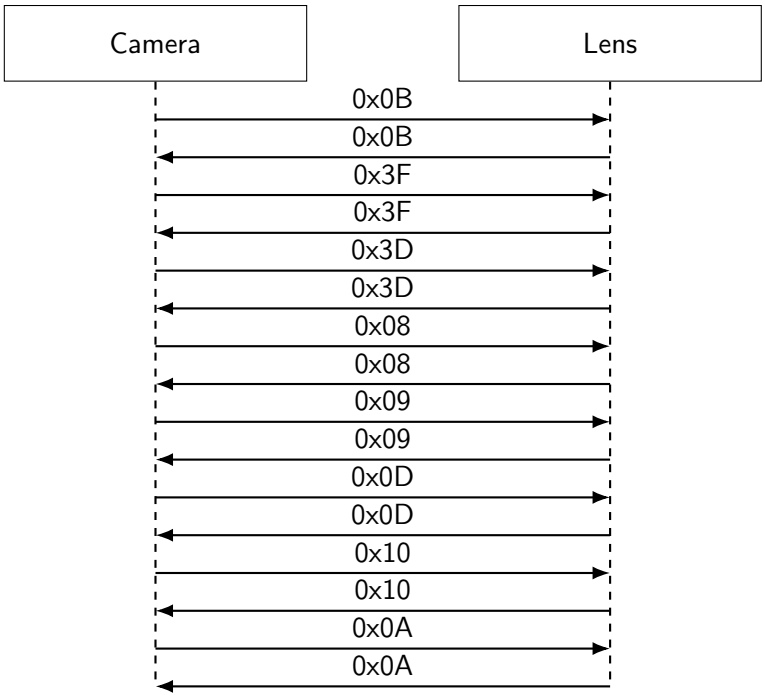


Figure 7: Initialiazation sequence

8.4 Regular Messaging

After initialization phase, the lens will send focus and aperture (`0x05` and `0x06` type message) on each BODY_POLL_LENS low pulse. This line is ignored before this phase. The camera sends `0x03` and `0x04` type message after it receives `0x06` type message from the lens and before next poll signal. The lens will handle the camera's commands and the changes will be reflected on the message from the lens on following polls.

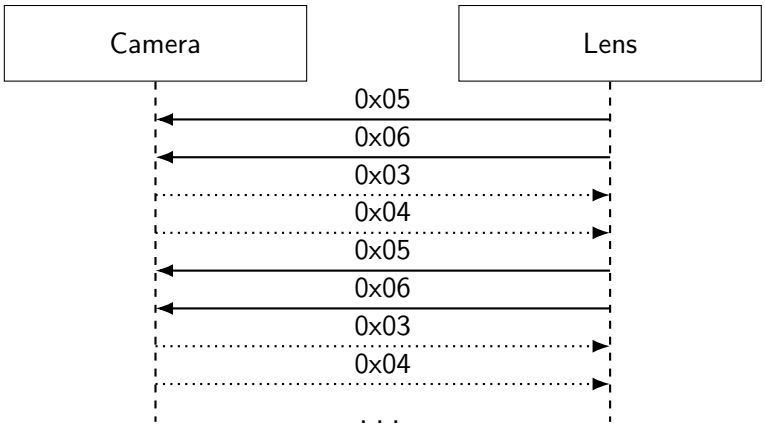


Figure 8: Regular Messaging. If the camera doesn't send any message to the lens, the lens will send messages to the camera on polling. The sequence number in this case will start from 1 and increment on each poll. If the camera sends the message to the lens, the sequence number of next message from the lens will be sequence number of message from the camera +1. Sequence number resets to `0x00` after `0xEF`. In practice, the camera continuously sends messages to the lens.

8.5 Shut Down

This phase is used to properly shutdown communication and power. We must do this before undoing the lens as the lens pads can make contact with powered pins on contact. Similarly, communication and power should only be provided once the lens is detected, otherwise the lens pads can make contact with powered pins. See full payload in Appendix.

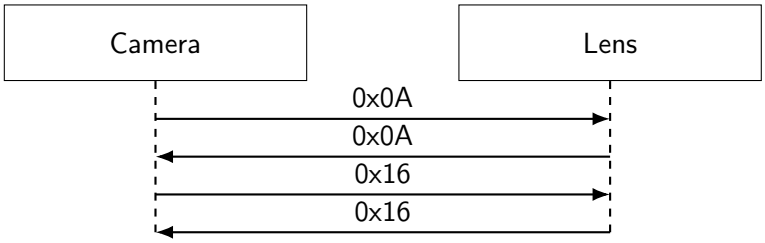


Figure 9: Shutdown sequence

9 Message Types

9.1 Speed Negotiation Messages

Message Type	Message Direction	Description
0x01	both	
0x07	both	
0x0C	both	Command to change the baud rate? Baud rate is changed to 1500K after the response. (The camera Powers the motor after the response, but we can power it at the start.)

Table 2: Speed Negotiation messages. See Table 13 for details and full payload.

9.2 Initialization Messages

Message Type	Message Direction	Description
0x0B	both	
0x3F	both	
0x3D	both	
0x08	both	
0x09	both	
0x0D	both	
0x10	both	Home Focus and Aperture?
0x0A	both	Configures Lens to send Aperture and Focus status on BODY_POLL_LENS low pulse. Which is same as screen refresh rate. Camera sends 0x03 and 0x04 message back to the lens before next BODY_POLL_LENS pulse.

Table 3: Initialization messages. See Table 14 for details and full payload.

9.3 Regular Messages

Message Type	Message Direction	Message Length	Description
0x03	Camera to Lens	0x20	Controls Aperture. Enables lens to move/change focus.
0x04	Camera to Lens	0x16 normally, 0x1B when changing focus	Controls "lens position", effectively the focus. The lens will only start moving once the camera enables movement with 0x03 type message.
0x05	Lens to Camera	0x75	Contains the current aperture of the Lens.
0x06	Lens to Camera	0x30 normally, 0x32 when done changing focus.	Contains the current lens position, effectively the focus.

Table 4: Regular Message Types

Note : When Camera sends message type 0x03 and 0x04 with sequence number s, the lens replies with message type 0x05 and 0x06 of sequence number s+1.

10 Shutdown Messages

Message Type	Message Direction	Description
0x0A	both	
0x16	both	Commands the lens to turn off UART communication.

Table 5: Shutdown Messages. See Table 15 for details and full payload.

11 Message Body Format

The speed negotiation, initialization messages, and shutdown messages remain the same for the same lens, so their body is not of interest. Below are the body format of regular messages.

11.1 0x03 type body

0x03 type message is 0x20 byte long so the body is 0x20 - 0x09 = 0x17 byte long.

Byte	Name	Value	Description
6		0x40	Depends on Shutter speed and exposure. Seems to be random at start.
7		0x41	Depends on Shutter speed and exposure. Seems to be random at start.
8	...	`0x00`	Normally `0x00` but it is `0x08` during start.
9	APERTURE_L	see Table 11	Low byte of Aperture. APERTURE & `0xFF`
10	APERTURE_H	see Table 11	High byte of Aperture. APERTURE» 8
11	APERTURE_L	same as byte 9	Low byte of Aperture. Repeated for some reason.
12	APERTURE_H	same as byte 10	High byte of Aperture. Repeated.
13	OVERWRIDDEN	`0x1C`	`0x1C` when using focus mode set on the camera. `0x18` when PC remote overrides the focusing mode or shutter is half pressed, or AEL is pressed.
14	static	`0x00`	Doesn't change.
15	static	`0x00`	Doesn't change.
16		`0x01` / `0x02` / `0x04`	Camera Controls this, but the lens doesn't seem to care. `0x01` during start. `0x02` for F-number between 1.4 to 7.1 (including the endpoints). `0x04` for F-number between 8.0 to 16.
17	$\overline{EN}$	`0x00` to enable	Enables movement of lens. Active Low, so enabled when `0x00`. Sony Camera's firmware controls this in non-trivial way. It counts up to the value in byte 16 then resets and cycles.
18		`0x00` / `0x01`	In MF mode, starts from `0x00`, goes to `0x01`, `0x00`, `0x01`.... . Starts from `0x11` then goes `0x10`, `0x11`... in AF mode.
19	static	`0x02`	Doesn't change.
20	static	`0x00`	Doesn't change.
21	static	`0x02`	`0x03` during start. `0x02` normally.
22	static	`0x01`	Doesn't change.
23	static	`0x00`	Doesn't change.
24	static	`0x00`	Doesn't change.
25	static	`0x00`	Doesn't change.
26	static	`0x2F`	Doesn't change.
27	static	`0x15`	Doesn't change.
28		`0x16` /`0x17`	Starts from `0x16` then `0x17`, `0x16`, `0x17`....

Table 6: `0x03` type message body

11.2 0x04 type body

`0x04` type message is normally `0x16` byte long, and `1B` byte long when camera commands to change the focus. Unlike Aperture, focus is changed with a relative move.

Byte	Name	Value	Description
6	static	`0x00`	Doesn't change.
7	static	`0x00`	Doesn't change.
8	static	`0x19`	Doesn't change.
9	FOCUSING_MODE	`0x81`	Focusing Mode the camera is set on. `0x81` in MF mode. `0x83` in AF mode.
10	static	`0x00`	Doesn't change.
11	static	`0x00`	Doesn't change.
12	static	`0x2D`	Doesn't change.
13	static	`0x00`	Doesn't change.
14	static	`0x00`	Doesn't change.
15	static	`0x00`	Doesn't change.
16		`0x08`	`0x09` when autofocusing is in action?. `0x08` normally or using MF?
17	static	`0x00`	Doesn't change.
18	static	`0x00`	Doesn't change.

Table 7: `0x04` type message

Say we want to move the lens(it's not a single lens but we can treat it as one) by DEL_LENS_POS in lens unit, then byte 6 to 18 are the same and the rest of the bytes are as follows:

Byte	Name	Value	Description
19	static	`0x1D`	Doesn't change.
20	DEL_LENS_POS_L		Low byte of relative move distance.
21	DEL_LENS_POS_H		High byte of relative move distance. DEL_LENS_POS is a 2's complement number, so you can move back and forth.
22	static	`0x00`	Doesn't change.
23	static	`0x2C`	Doesn't change.

Table 8: augmented `0x04` type message when changing focus

Note : The Lens Handles the focus motor Limits, so the camera doesn't need to worry about it.

11.3 0x05 type body

`0x05` type message is `0x75` byte long so the body is `0x75` - `0x09` = `0x6C` byte long.

Byte	Name	Value	Description
6	APERTURE_L	See Table 11 for possible values.	Low byte of aperture value set.
7	APERTURE_H		High byte of aperture value set.
8	APERTURE_L		Low byte of aperture value set.
9	APERTURE_H		High byte of aperture value set.
23	APERTURE_DIAL_L		Low byte of aperture value on the Aperture dial.
24	APERTURE_DIAL_H		High byte of aperture value on the Aperture dial.
68	AF_SWITCH_STATUS		Indicates the AF switch status on the Lens. `0x03` for MF `0x01` for AF.

Table 9: `0x05` type message body

11.4 0x06 type body

Byte	Name	Value	Description
8	LENS_POS_L	See Table 12 and Fig 10.	Low byte of lens position. The lens position ranges from 0x204E (lens closest to sensor) to 0x320F (farthest). LENS_POS is linearly related to the image distance v (distance from lens to sensor): $v = 5.518144 \times 10^{-5} \times FOCUS\_POS - 0.381395$. Use this value of v in the lens equation to calculate the object distance u, which is the distance of focused object(aka set focus).
9	LENS_POS_H		High byte of lens position.
26	LENS_POS_L		Low byte of lens position. Usually same as byte 8 but seems to follow byte 8 when lens is moving.
27	LENS_POS_H		High byte of lens position. Usually same as byte 9 but seems to follow byte 9 when lens is moving

Table 10: `0x06` type message body

When camera sends command to change focus, the new focus will be $LENS\_POS + DEL\_LENS\_POS$. Remember these are 2's complement numbers.

12 Aperture Mapping

Below is the Aperture mapping from F-number to APERTURE.

F-Number	APERTURE	APERTURE (Decimal)
1.4	0x1100	4352
1.6	0x1155	4437
1.8	0x11AA	4522
2.0	0x1200	4608
2.2	0x1255	4693
2.5	0x12AA	4778
2.8	0x1300	4864
3.2	0x1355	4949
3.5	0x13AA	5034
4.0	0x1400	5120
4.5	0x1455	5205
5.0	0x14AA	5290
5.6	0x1500	5376
6.3	0x1555	5461
7.1	0x15AA	5546
8.0	0x1600	5632
9.0	0x1655	5717
10	0x16AA	5802
11	0x1700	5888
13	0x1755	5973
14	0x17AA	6058
16	0x1800	6144

Table 11: Aperture Mapping

The F-Number and APERTURE are related in following way.
$F - Number = \sqrt{2}^{\frac{APERTURE}{256} - 16}$.

13 Focus Mapping

Below is the mapping between focused distance and the lens position.

focus(m)	LENS_POS	LENS_POS (Decimal)
0.4	0x320F	12815
0.5	0x2ADB	10971
0.6	0x2976	10614
0.7	0x28AB	10411
0.8	0x282D	10285
0.9	0x27D2	10194
1	0x2785	10117
2	0x26C1	9921
3	0x2643	9795
4	0x2612	9746
5	0x25F6	9718
6	0x25E8	9704
7	0x25DA	9690
8	0x25D3	9683
9	0x25CC	9676
10	0x25C5	9669
13	0x25BE	9662
16	0x25B7	9655
23	0x25B0	9648
37	0x25A9	9641
101	0x25A2	9634
inf	0x259B	9627
inf	0x204E	8270

Table 12: Focus Mapping

The image distance is a linear function of LENS_POS. Once we find the image distance,v, we can use it to find the object distance, u, that will be in focus using lens equation.

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v} \tag{1}$$

$$u = \frac{1}{\frac{1}{f} - \frac{1}{v}} \tag{2}$$

$f = 15mm = \frac{3}{20}m$ for us. Similarly, v is a linear function of LENS_POS. Let $v = m \times LENS\_POS + b$. After substituting, we get.

$$focus = u = \frac{1}{\frac{20}{3} - \frac{1}{m \times LENS\_POS + c}} \tag{3}$$

We can fit equation 3, to data in Table 12 to get value of m and c.

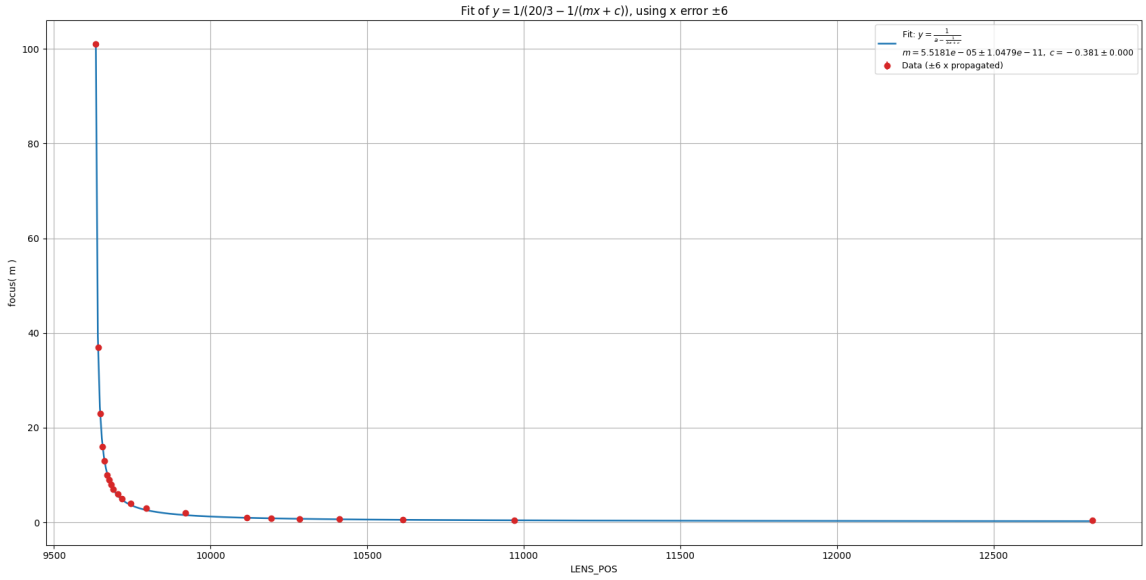


Figure 10: Focus Mapping.

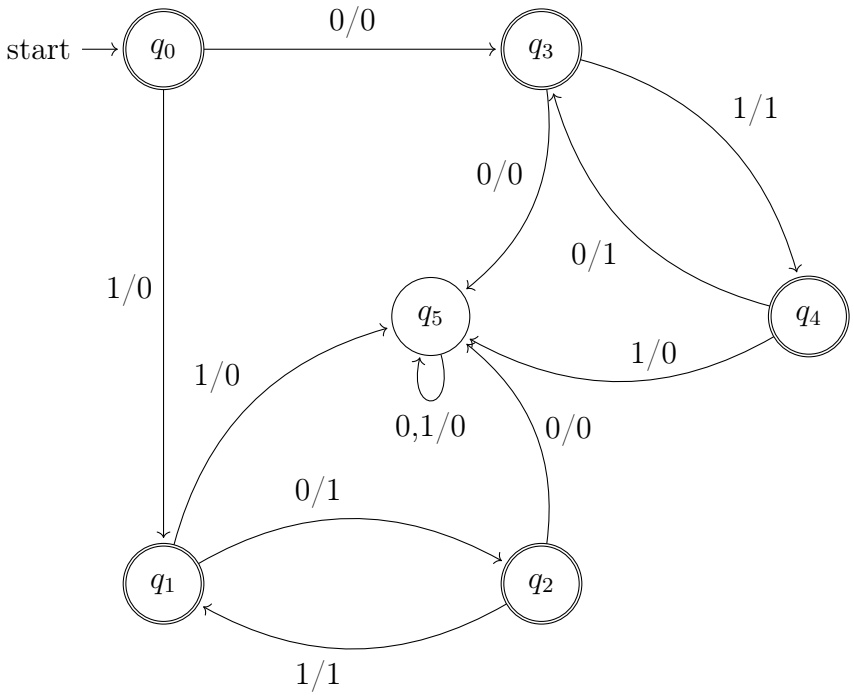
Fit parameters:
$m = 5.518144 \times 10^{-5} \pm 1.047941 \times 10^{-11}$
$c = -0.381395 \pm 0.000000$

Therefore,

$$focus = \frac{1}{\frac{20}{3} - \frac{1}{5.518144 \times 10^{-5} \times LENS\_POS - 0.381395}} \tag{4}$$

14 Software Design

15 Flow Meter Mealy Machine Design



15.1 States (Q):

Q = $\{q_0, q_1, q_2, q_3, q_4, q_5\}$

- q_0 = WAIT FOR EDGE
- q_1 = RISING EDGE DETECTED
- q_2 = FALLING EDGE DETECTED
- q_3 = FALLING EDGE AT START
- q_4 = RISING EDGE AFTER FALLING
- q_5 = ERROR STATE

15.2 Input Alphabet (Σ):

Σ = $\{0, 1\}$

- 0 = Falling Edge Detected
- 1 = RISING EDGE DETECTED

15.3 Output Alphabet (Λ):

$\Lambda =$ {0, 1}

- 0 = Don't increment cycles detected
- 1 = Increment cycles detected

15.4 Transition Function ($\delta : Q \times \Sigma \rightarrow Q$):

$\delta =$ { ($q_0, 0, q_3$) , ($q_0, 1, q_1$) , ($q_1, 0, q_2$) , ($q_1, 1, q_5$) , ($q_2, 0, q_5$) , ($q_2, 1, q_1$) , ($q_3, 0, q_5$) , ($q_3, 1, q_4$) , ($q_4, 0, q_3$) , ($q_4, 1, q_5$) , ($q_5, 0, q_5$) , ($q_5, 1, q_5$) }

15.5 Output Function ($G : Q \times \Sigma \rightarrow \Lambda$):

$G =$ { ($q_0, 0, 0$) , ($q_0, 1, 0$) , ($q_1, 0, 1$) , ($q_1, 1, 0$) , ($q_2, 0, 0$) , ($q_2, 1, 1$) , ($q_3, 0, 0$) , ($q_3, 1, 1$) , ($q_4, 0, 1$) , ($q_4, 1, 0$) , ($q_5, 0, 0$) , ($q_5, 1, 0$) }

15.6 Accepting States (F) :

F = Q/{q_5}

15.7 Transition Table

Current State(q)	Input	Next State	Output(G)
q_0	0	q_3	0
q_0	1	q_1	0
q_1	0	q_2	1
q_1	1	q_5	0
q_2	0	q_5	0
q_2	1	q_1	1
q_3	0	q_5	0
q_3	1	q_4	1
q_4	0	q_3	1
q_4	1	q_5	0
q_5	0	q_5	0
q_5	1	q_5	0

The Deterministic Finite Automaton (DFA) described above was used to count the total number of half cycles. Specifically, the state machine is a Mealy machine, meaning the output is a function of both the current state and the input. It also incorporates error detection when edges are not properly registered. For example, if a rising edge is followed by another rising edge without an intervening falling edge, this signals an error. In such cases, the state machine transitions to an error state to indicate the detection failure.

16 References:

Implementation. bostwickenator, entropy512. https://github.com/LexOptical/E-Mount
Protocol Documentation. bostwickenator, entropy512. https://docs.google.com/document/d/1iw54nzrF0bzQgLINpcP9F8OddON5cd7LjlwCDPTNZK0/edit?tab=t.0#heading=h.dk3gmdgcv9gp
For deriving the relationship between F-Stop and the aperture value. See section Stops, f-stop conventions, and exposure on https://en.wikipedia.org/wiki/F-number

17 Appendix

17.1 Speed Negotiation, Full Payload

Message Type	Message Direction	Description	Full Payload
0x01	Camera to Lens		0xF0 0x29 0x00 0x02 0x00 0x01 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0x2F 0xF8 0x0F 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x59 0x0A 0x55
0x01	Lens to Cam		0xF0 0x29 0x00 0x02 0x00 0x01 0xFF 0x9F 0x78 0x5D 0x82 0x60 0x18 0x7E 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x17 0x04 0x55
0x07	Cam to Lens		0xF0 0x0A 0x00 0x02 0x00 0x07 0x00 0x13 0x00 0x55
0x07	Lens to Cam		0xF0 0x2B 0x00 0x02 0x00 0x07 0x01 0x03 0x80 0x01 0x00 0x00 0x01 0x00 0xA0 0x74 0xC5 0x00 0x00 0x00 0x00 0x60 0x92 0x86 0x5E 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x69 0x04 0x55
0x0C	Cam to Lens	Command to change the baud rate?	0xF0 0x0A 0x00 0x02 0x00 0x0C 0x02 0x1A 0x00 0x55
0x0C	Lens to Cam	Baud rate is changed to 1500K after this response. (The camera Powers the motor after this response, but we can power it at the start.)	0xF0 0x0A 0x00 0x02 0x00 0x0C 0x01 0x19 0x00 0x55

Table 13: Speed Negotiation Payload

17.2 Initialization, Full Payload

Table 14: Initialization Payload

Message Type	**Message Direction**	**Description**	**Full Payload**
0x0B	Camera to Lens		0xF0 0x0B 0x00 0x02 0x00 0x0B 0x60 0x00 0x78 0x00 0x55
0x0B	Lens to Cam		0xF0 0x0B 0x00 0x02 0x00 0x0B 0x60 0x00 0x78 0x00 0x55
0x3F	Cam to Lens		0xF0 0x0A 0x00 0x02 0x00 0x3F 0x00 0x4B 0x00 0x55
0x3F	Lens to Cam		0xF0 0x4A 0x00 0x02 0x00 0x3F 0x00 0x31 0x35 0x6D 0x6D 0x20 0x46 0x31 0x2E 0x34 0x20 0x44 0x47 0x20 0x44 0x4E 0x20 0x44 0x49 0x41 0x47 0x4F 0x4E 0x41 0x4C 0x20 0x46 0x49 0x53 0x48 0x45 0x59 0x45 0x20 0x7C 0x20 0x41 0x72 0x74 0x20 0x30 0x32 0x34 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x46 0x0B 0x55
0x3D	Cam to Lens		0xF0 0x20 0x00 0x02 0x00 0x3D 0xBD 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x1C 0x01 0x55
0x3D	Lens to Cam		0xF0 0x48 0x00 0x02 0x00 0x3D 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0C 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x94 0x00 0x55
0x08	Cam to Lens		0xF0 0x11 0x00 0x02 0x00 0x08 0xC6 0xE1 0x00 0x00 0x00 0x0C 0x62 0x02 0x32 0x02 0x55

Continued on next page

Table 14 — continued from previous page

Message Type	Message Direction	Description	Full Payload
0x08	Lens to Cam		0xF0 0xD2 0x00 0x02 0x00 0x08 0x00 0x11 0x00 0x18 0x00 0x00 0x00 0x80 0x00 0x14 0x01 0x00 0x12 0xE1 0x01 0x03 0x00 0x15 0x00 0x00 0x14 0x00 0x00 0x00 0x58 0x00 0x00 0x57 0x83 0x92 0x49 0x16 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xE0 0x43 0x00 0xE0 0xFF 0x84 0xA4 0x00 0x00 0x08 0x32 0x25 0x40 0x2F 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x43 0x25 0x04 0x32 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xF3 0xF3 0x07 0xFB 0xFB 0x00 0x00 0x00 0x04 0x04 0x20 0x46 0x00 0x80 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x01 0x01 0x01 0x09 0x09 0x00 0x00 0x00 0x00 0x00 0xE6 0x80 0xDD 0x52 0x00 0xD0 0xDD 0xDF 0xF0 0x00 0x08 0xA0 0xDD 0xE0 0x00 0x52 0x00 0x10 0x10 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x84 0x64 0x27 0x00 0x06 0x10 0xFF 0x33 0x00 0x00 0x00 0x00 0x00 0x03 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0C 0x0C 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x05 0x00 0x01 0x02 0x01 0x02 0x00 0x00 0x07 0x07 0xFF 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x13 0x00 0x00 0x01 0x00 0x01 0xE1 0x1B 0x55
0x09	Cam to Lens		0xF0 0x0D 0x00 0x02 0x00 0x09 0x00 0x00 0x00 0x00 0x18 0x00 0x55
0x09	Lens to Cam		0xF0 0x14 0x00 0x02 0x00 0x09 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x1F 0x00 0x55
0x0D	Cam to Lens		0xF0 0x0A 0x00 0x02 0x00 0x0D 0x00 0x19 0x00 0x55
0x0D	Lens to Cam		0xF0 0x0A 0x00 0x02 0x00 0x0D 0x00 0x19 0x00 0x55

Continued on next page

Table 14 — continued from previous page

Message Type	Message Direction	Description	Full Payload
0x10	Cam to Lens	Home/Locate Aperture and Focus?	0xF0 0x0A 0x00 0x02 0x00 0x10 0x1F 0x3B 0x00 0x55
0x10	Lens to Cam	Turning the focus dial should change focus after this. When you turn the focus dial the lens changes the focus and updates the FOCUS_POS. However, when aperture dial is rotated, it updates the APERTURE_DIAL but camera has to command to actually change the aperture.	0xF0 0x0A 0x00 0x02 0x00 0x10 0x00 0x1C 0x00 0x55
0x0A	Cam to Lens	Configures Lens to send Aperture and Focus status on BODY_POLL_LENS low pulse. Which is same as screen refresh rate. Camera chooses to send 0x03 and 0x04 message back to the camera before next BODY_POLL_LENS pulse.	0xF0 0x19 0x00 0x02 0x00 0x0A 0xFF 0xFF 0x03 0x00 0x00 0x00 0x00 0x00 0x3F 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x65 0x02 0x55
0x0A	Lens to Cam		0xF0 0x19 0x00 0x02 0x00 0x0A 0xFF 0xFF 0x03 0x00 0x00 0x00 0x00 0x00 0x3F 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x65 0x02 0x55

17.3 Shutdown, Full Payload

Message Type	Message Direction	Description	Full Payload
0x0A	Camera to Lens		0xF0 0x19 0x00 0x02 0x00 0x0A 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x25 0x00 0x55
0x0A	Lens to Cam		0xF0 0x19 0x00 0x02 0x00 0x0A 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x25 0x00 0x55
0x16	Cam to Lens	Commands the lens to power off UART.	0xF0 0x0A 0x00 0x02 0x00 0x16 0x01 0x23 0x00 0x55
0x16	Lens to Cam	After this response, RX powers down then TX powers down. BODY_POLL_LENS is powered off after a bit.	0xF0 0x0A 0x00 0x02 0x00 0x16 0x00 0x22 0x00 0x55

Table 15: shutdown payload