RCB4 Protocol to Play Motions

These commands are to be sent via the Serial Port on the RCB4 (the same port that connects to the computer).

All Byte Values in this document are represented as Hexadecimal.

Preface: Logic Levels and Connection Settings

These commands should be sent through the RCB4 Serial port (the one used to communicate with heart to Heart)
The **RCB4 can be set to different speeds (from 115200bps up to 1.25Mbps)**. Use Heart to heart to check and configure the Speed that you need.

The Serial Connection needs to be set to **8 Data Bits, 1 Stop Bit, Even Parity**
Additionally, if you won't be using Kondo USB to Serial Adapter HS (i.e. if you'll be using your own custom connection) please note **RCB4 uses 5V CMOS Inverted logic.**

How to Play a Motion

To play a Motion there are 4 steps:

Step 1: Send a Command to Stop Current Program and Save Current Status to RAM. It's the same sequence of Bytes all the time regardless of the motion number.
Step 2: Send the "Run Motion" Command. This command varies depending on the Motion number.
Step 3: Send Command to restore Current Status. It's the same sequence of Bytes all the time regardless of the motion number.
Step 4 (OPTIONAL): Wait for the Motion to Complete. You can do it polling the controller every 50ms to obtain Excution Status.
"Polling" the controller for Exceution Status is OPTIONAL.

If you send another sequence to Play Motion while you're still executing a motion, the Robot will stop the current motion and will initiate the new motion.

Polling "Execution Status"

The RCB4 controller provides a command to Query the Execution Status of Motions.
If you wish to poll the controller to check if a motion has finished playing or if any motion is running you should use this command (the Byte sequences for the command are described bellow).

If you are polling the controller continuouslly, you must wait 50ms between each request sent as the controller only updates the Motion Execution Status every 50ms.

Command Sequences

Stop Current Program and Save Status to RAM *(Step 1)*

13	00	02	00	00	00	21	87	fd	03	00	00	00	00	00	00	00	00	bd

(Note: it's always the same byte sequence regardless of the motion being played)

Command Response will be a **Type 1 response** (4 bytes, see "Response Codes" bellow)

Running a Motion *(Step 2)*

Command Response will be a **Type 1 response** (4 bytes, see "Response Codes" bellow)

Motion Number	0	07	0c	b8	0b	00	00	d6
Motion Number	1	07	0c	b8	13	00	00	de
Motion Number	2	07	0c	b8	1b	00	00	e6
Motion Number	3	07	0c	b8	23	00	00	ee
Motion Number	4	07	0c	b8	2b	00	00	f6
Motion Number	5	07	0c	b8	33	00	00	fe
Motion Number	6	07	0c	b8	3b	00	00	06
Motion Number	7	07	0c	b8	43	00	00	0e
Motion Number	8	07	0c	b8	4b	00	00	16
Motion Number	9	07	0c	b8	53	00	00	1e
Motion Number	10	07	0c	b8	5b	00	00	26
Motion Number	11	07	0c	b8	63	00	00	2e
Motion Number	12	07	0c	b8	6b	00	00	36
Motion Number	13	07	0c	b8	73	00	00	3e
Motion Number	14	07	0c	b8	7b	00	00	46
Motion Number	15	07	0c	b8	83	00	00	4e
Motion Number	16	07	0c	b8	8b	00	00	56
Motion Number	17	07	0c	b8	93	00	00	5e
Motion Number	18	07	0c	b8	9b	00	00	66
Motion Number	19	07	0c	b8	a3	00	00	6e
Motion Number	20	07	0c	b8	ab	00	00	76

Motion Number	21	07	0c	b8	b3	00	00	7e
Motion Number	22	07	0c	b8	bb	00	00	86
Motion Number	23	07	0c	b8	c3	00	00	8e
Motion Number	24	07	0c	b8	cb	00	00	96
Motion Number	25	07	0c	b8	d3	00	00	9e
Motion Number	26	07	0c	b8	db	00	00	a6
Motion Number	27	07	0c	b8	e3	00	00	ae
Motion Number	28	07	0c	b8	eb	00	00	b6
Motion Number	29	07	0c	b8	f3	00	00	be
Motion Number	30	07	0c	b8	fb	00	00	c6
Motion Number	31	07	0c	b8	03	01	00	cf
Motion Number	32	07	0c	b8	0b	01	00	d7
Motion Number	33	07	0c	b8	13	01	00	df
Motion Number	34	07	0c	b8	1b	01	00	e7
Motion Number	35	07	0c	b8	23	01	00	ef
Motion Number	36	07	0c	b8	2b	01	00	f7
Motion Number	37	07	0c	b8	33	01	00	ff
Motion Number	38	07	0c	b8	3b	01	00	07
Motion Number	39	07	0c	b8	43	01	00	0f
Motion Number	40	07	0c	b8	4b	01	00	17
Motion Number	41	07	0c	b8	53	01	00	1f
Motion Number	42	07	0c	b8	5b	01	00	27
Motion Number	43	07	0c	b8	63	01	00	2f

Motion Number	44	07	0c	b8	6b	01	00	37
Motion Number	45	07	0c	b8	73	01	00	3f
Motion Number	46	07	0c	b8	7b	01	00	47
Motion Number	47	07	0c	b8	83	01	00	4f
Motion Number	48	07	0c	b8	8b	01	00	57
Motion Number	49	07	0c	b8	93	01	00	5f
Motion Number	50	07	0c	b8	9b	01	00	67
Motion Number	51	07	0c	b8	a3	01	00	6f
Motion Number	52	07	0c	b8	ab	01	00	77
Motion Number	53	07	0c	b8	b3	01	00	7f
Motion Number	54	07	0c	b8	bb	01	00	87
Motion Number	55	07	0c	b8	c3	01	00	8f
Motion Number	56	07	0c	b8	cb	01	00	97
Motion Number	57	07	0c	b8	d3	01	00	9f
Motion Number	58	07	0c	b8	db	01	00	a7
Motion Number	59	07	0c	b8	e3	01	00	af
Motion Number	60	07	0c	b8	eb	01	00	b7
Motion Number	61	07	0c	b8	f3	01	00	bf
Motion Number	62	07	0c	b8	fb	01	00	c7
Motion Number	63	07	0c	b8	03	02	00	d0
Motion Number	64	07	0c	b8	0b	02	00	d8
Motion Number	65	07	0c	b8	13	02	00	e0
Motion Number	66	07	0c	b8	1b	02	00	e8

Motion Number	67	07	0c	b8	23	02	00	f0
Motion Number	68	07	0c	b8	2b	02	00	f8
Motion Number	69	07	0c	b8	33	02	00	00
Motion Number	70	07	0c	b8	3b	02	00	08
Motion Number	71	07	0c	b8	43	02	00	10
Motion Number	72	07	0c	b8	4b	02	00	18
Motion Number	73	07	0c	b8	53	02	00	20
Motion Number	74	07	0c	b8	5b	02	00	28
Motion Number	75	07	0c	b8	63	02	00	30
Motion Number	76	07	0c	b8	6b	02	00	38
Motion Number	77	07	0c	b8	73	02	00	40
Motion Number	78	07	0c	b8	7b	02	00	48
Motion Number	79	07	0c	b8	83	02	00	50
Motion Number	80	07	0c	b8	8b	02	00	58
Motion Number	81	07	0c	b8	93	02	00	60
Motion Number	82	07	0c	b8	9b	02	00	68
Motion Number	83	07	0c	b8	a3	02	00	70
Motion Number	84	07	0c	b8	ab	02	00	78
Motion Number	85	07	0c	b8	b3	02	00	80
Motion Number	86	07	0c	b8	bb	02	00	88
Motion Number	87	07	0c	b8	c3	02	00	90
Motion Number	88	07	0c	b8	cb	02	00	98
Motion Number	89	07	0c	b8	d3	02	00	a0

Motion Number	90	07	0c	b8	db	02	00	a8
Motion Number	91	07	0c	b8	e3	02	00	b0
Motion Number	92	07	0c	b8	eb	02	00	b8
Motion Number	93	07	0c	b8	f3	02	00	c0
Motion Number	94	07	0c	b8	fb	02	00	c8
Motion Number	95	07	0c	b8	03	03	00	d1
Motion Number	96	07	0c	b8	0b	03	00	d9
Motion Number	97	07	0c	b8	13	03	00	e1
Motion Number	98	07	0c	b8	1b	03	00	e9
Motion Number	99	07	0c	b8	23	03	00	f1
Motion Number	100	07	0c	b8	2b	03	00	f9

Restore EEPROM Status *(Step 3)*

09	00	02	00	00	00	23	87	b5

Note: it's always the same byte sequence regardless of the motion being played

Command Response will be a **Type 1 response** (4 bytes, see "Response Codes" bellow)

Query Execution Status *(Step 4)*

0a	00	20	00	00	00	00	00	02	2c

You must wait 50ms between each request sent as the controller only updates the Motion Execution Status every 50ms.

Command Response will be a **Type 2 response** (5 bytes, see "Response Codes" bellow)

(please Reffer to the next page for the List of "Response Types" and byte sequences)

Response Codes

There are 2 types fo Response codes for the commands presented in this document.

Type 1 Response:

Response sent for "Stop Motion", "Play Motion" and "Restore EEPROM" commands

Response length will be 4 bytes.

You must **check the 3rd byte returned** which should be **RCB4_ACK_BYTE** (0x06). If the command is not understood either **RCB4_NCK_BYTE** (0x15) is received or no reply is received at all.

If there is no reply at all check speed and parity setting and also signal level (see "Preface") if you're not using the Kondo USB Adapter HS

Type 2 Response:

Response sent for "Query Execution Status"

Response length will be 5 bytes.

You must **check the 3rd byte returned: 0x23** means the Robot is executing a Motion. A different number will mean the Robot is not executing a motion.

If there is no reply at all check speed and parity setting and also signal level (see "Preface") if you're not using the Kondo USB Adapter HS

You should **always read back the Return codes before sending another command**. Failing to do so may cause the controller not to respond.