High nibble	Low nibble							
	0 / 8	**1 / 9**	**2 / a**	**3 / b**	**4 / c**	**5 / d**	**6 / e**	**7 / f**
0			PlayToneVar • byte *index* • byte *duration*	DirectEvent • byte *evsrc* • short *evarg*	CalibrateEvent • byte *event* • byte *upper* • byte *lower* • byte *hysteresis*	SetSourceValue • byte *dest_source* • byte *dest_value* • byte *index* • short *arg*	ClearAllEvents	
1	PBAlive		PollValue • byte *source* • byte *argument* → short *value*	SetMotorPower • byte *motors* {3} • byte *source* • byte *argument*	SetVar • byte *index* • byte *source* • short *argument*	PBVersions • byte *key[5]* → short *rom[2]* → short *firmware[2]*		CallSubroutine • byte *subroutine*
2	MemMap → ushort *map[94]*	SetMotorOnOff • byte *code* {1}	SetWatch • byte *hours* • byte *minutes*	PlayTone • short *frequency* • byte *duration*	AddVar • byte *index* • byte *source* • short *argument*	BeginTaskDL § • byte *unused* • byte *task* • byte *subcalls* • short *length*		BranchNear • ubyte *offset* †
3	PBBattery → ushort *millivolts*	TransmitPower • byte *range*	SetSensorType • byte *sensor* • byte *type*	SelectDisplay • byte *source* • short *argument*	SubVar • byte *index* • byte *source* • short *argument*	BeginSubDL § • byte *unused* • short *subroutine* • short *length*		CheckLoopCountNear • ubyte *offset*
4	DeleteAllTasks		SetSensorMode • byte *sensor* • byte *mode*	Wait • byte *source* • short *argument*	DivVar • byte *index* • byte *source* • short *argument*	TransferData § • short *index* • short *length* • byte *data[length]* • byte *checksum*		
5	StopAllTasks	PlaySystemSound • byte *sound*	SetDatalog § • short *size*		MulVar • byte *index* • byte *source* • short *argument*			
6	PBTurnOff	DeleteTask • byte *task*	DatalogNext § • byte *source* • byte *argument*	UploadRAM • byte *address* • byte *count* → byte *data[count]*	SgnVar • byte *index* • byte *source* • short *argument*	DeleteFirmware • byte *key[5]*		ConnectDisconnect • byte *code* {1}
7	DeleteAllSubs	StartTask • byte *task*	BranchFar • ubyte *offset* ‡ • ubyte *extension*	EnterAccessControl • byte *resources* • ubyte *offset* ‡ • ubyte *extension*	AbsVar • byte *index* • byte *source* • short *argument*	BeginFirmwareDownload § • short *address* • short *checksum* • byte *unused*		SetNormSetInvDir • byte *code* {2}
8	ClearSound	StopTask • byte *task*	SetLoopCounter • byte *source* • byte *argument*		AndVar • byte *index* • byte *source* • short *argument*	CheckBranchNear • byte *opsrc1* * • byte *src2* • short *arg1* • byte *arg2* • ubyte *offset*		
9	ClearPBMessage	SelectProgram • byte *program*	CheckLoopCountFar • ushort *offset*	SetEvent • byte *event* • byte *evsensor* • byte *evtype*	OrVar • byte *index* • byte *source* • short *argument*	CheckBranchFar • byte *opsrc1* * • byte *src2* • short *arg1* • byte *arg2* • short *offset*		IncCounter • byte *counter*
a	ExitAccessControl	ClearTimer • byte *timer*		SetMaxPower • byte *motors* {3} • byte *source* • byte argument	UploadDatalog • short *first* • short *count* → dlrec *data[length]*	UnlockFirmware • byte *key[5]* → byte *data[25]*		DecCounter • byte *counter*
b	ExitEventCheck	PBPowerDownTime • byte *minutes*	SendPBMessage • byte *source* • byte *argument*		SEnterEventCheck • byte *evsrc* • short *evarg* • ubyte *offset*	LEnterEventCheck • byte *evsrc* • short *evarg* • ubyte *offset* ‡ • ubyte *extension*		ClearCounter • byte *counter*
c		DeleteSub • byte *subroutine*	SendUARTData • byte *first* • byte *count*					
d	MuteSound	ClearSensorValue • byte *sensor*	RemoteCommand • short *command*					SetPriority • byte *priority*
e	UnmuteSound	SetMotorDir • byte *code* {2}				ViewSourceValue • byte *unused* • byte *decimals* • byte *source* • short *argument*		
f			SDecVarJumpLTZero • byte *variable* • ubyte *offset* †	LDecVarJumpLTZero • byte *variable* • ubyte *offset* ‡ • ubyte *extension*			ReturnFromSub	SetMessage • byte *message*

RCX Firmware Opcode Matrix | Available at www.bensnotebook.net | Sources: RCX SDK, www.mralligator.com/rcx | Rev 1.0

All shorts are little endian	
§: → byte *errorcode* : 0 = success	
key[5] : [4c, 45, 47, 4f, ae]	
dlrec: byte *type* , short *value*	
🟦	= 2.0 firmware only
🟧	= ROM only

Datalog dlrec Types	
00 - 1f	Variable (0 - 31)
20 - 23	Timer (0 - 3)
40 - 42	Sensor (ports 1 - 3)
80	Clock (watch)
ff	Current datalog size

Sources		
Source	**Name**	**Argument**
00	Variable	0 - 31
01	Timer (0.1 s)	0 - 3
02	Immediate	-32768 - 32767
03	Motor state **	0 - 2 (ports A - C)
04	Random (0 - max)	Max: 0 - 65535
05 - 07	--- *reserved* ---	N/A
08	Program	0 - 2 (ports 1 - 3)
09	Sensor value	
0a	Sensor type	
0b	Sensor mode	
0c	Raw sensor	
0d	Boolean sensor	
0e	Watch (minutes)	0
0f	Message	0
11	Global motor state	0 - 2 (ports A - C)
15	Counter	0 - 2
17	Task events	0 - 9
19	Event state	0 - 15 (events)
1a	Timer (0.01 s)	0 - 3
1b	Click counter	0 - 15 (events)
1c	Upper threshold	
1d	Lower threshold	
1e	Hysteresis	
1f	Duration	
21	UART	0 - 15 (16 - 17 setup)
22	Battery millivolts	0
23	Firmware version	0
24	Indirect variable	0 - 31

Sensor Modes	
0	Raw
1	Boolean
2	Transition count
3	Period count
4	Percent
5	Celsius
6	Fahrenheit
7	Angle

Sensor Types	
0	Raw
1	Boolean
2	Temperature
3	Light
4	Rotation

Event Sensors	
0 - 2	Sensors (ports A - C)
3 - 6	Timer (0 - 3)
7	Message
8 - a	Counter (0 - 2)

Event Types	
00	Pressed
01	Released
02	Period
03	Transition
07	Change rate exceeded
08	Enter low
09	Enter normal
0a	Enter high
0b	Click
0c	Double click
0e	Message
10	Reset event

Bitfield * xx--yyyy	x	0	≤
		1	≥
		2	≠
		3	=
	y	Source	

Bitfield ** xy--rppp	p	Power (0 - 7)	
	r	1 = rev, 0 = fwd	
	y	off	both 0 = float
	x	on	

Bitfield {1} xy---cba	a	Motors to update	
	b		
	c		
	y	off	both 0 = float
	x	on	

Bitfield {2} ft---cba	a	Motors to update
	b	
	c	
	t	1 = toggle
	f	1 = fwd, 0 = rev

Bitfield {3} -----cba	a	Motors to update
	b	
	c	

†	0x80 = 0	*offset*
	0x80 = 1	128 - *offset*

‡	0x80 = 0	*offset* + 128*_extension_
	0x80 = 1	*offset* + 128 + 128*_extension_