

Section 4: Technical Information

Communication (Data) Between the RCX and MicroWorlds EX

This section describes how to use the primitives that let you transfer information between MicroWorlds EX and the RCX. Use these techniques in projects that combine onscreen action and robotic devices.

One way to use this type of interaction is to have an onscreen event trigger an action by the RCX. For example:

- A turtle winning a race onscreen triggers a light show executed by the RCX;
- Information about a robotics project is presented onscreen and the robotic device is triggered by a button or turtle in the MicroWorlds EX project;
- An onscreen quiz can animate a robotic device to make a “game-show” action based on the right and wrong answers provided by the player;
- MicroWorlds EX can transmit numerical values to the RCX.

Note, however, that infrared communication is slow and you should not attempt to create projects where the synchronization between the onscreen and the robotics actions is critical.

A second way to integrate onscreen action and robotic devices is to have MicroWorlds EX Robotics and the RCX communicate data in one direction or the other. For example:

- The RCX sends sensor values (light, temperature, etc.) to MicroWorlds EX, where they can be recorded in a text box and/or in a graph.
- A MicroWorlds EX project (onscreen and interactive) sends data to the RCX to control its behavior.

Some primitives are dedicated to communication between MicroWorlds EX and the RCX. When using these primitives, the RCX must be in the IR Transmitter's line of sight.

When this is executed within MicroWorlds EX Robotics...	The RCX must use or respond with...	Example
<p>download <i>word</i></p>	<p>Nothing. After the download process, the procedures can be run by the RCX.</p> <p>A download instruction should be followed by a sendremotecommand instruction in MicroWorlds EX. This runs the program and links the main procedure of the program to the Run button on the RCX.</p> <p>The input for download is a word. The most common input for download is the contents of a text box. You can take advantage of the fact that the name of a text box reports the text box' contents as a "long word". Therefore, the name of a text box can be used as input for download.</p> <p>In the instruction download text1, text1 reports its contents as a long word. This long word is taken by download as input. When the text "arrives" in the RCX, it is interpreted as normal procedures, as if they originated from the Procedures Panel in the RCX Tab.</p>	<p>Text1 contains the procedure goforit. Text1 reports its contents to download which sends it to the RCX. The next instruction, sendremotecommand "goforit", starts the procedure in the RCX. These instructions can be typed in the MicroWorlds EX Command Center or executed from a button, a programmed color or turtle.</p> <pre>download text1 sendrc "goforit</pre> <p>There are actually three ways to start the procedure in the RCX:</p> <ul style="list-style-type: none"> • Press the Run button on the RCX, • Type goforit in the RCX Direct Mode Panel; or, • From within MicroWorlds EX, use: sendremotecommand "goforit"

When this is executed within MicroWorlds EX Robotics...	The RCX must use or respond with...	Example
sendremotecommand <i>word-or-list</i> sendrc <i>word-or-list</i>	Nothing. The input is a word or a list that contains instructions that can be run in the RCX (procedures that have been downloaded or primitives). The <i>word-or-list</i> instruction is run in the RCX and it is linked to the Run button.	A motor is connected to port A on the RCX. These instructions are typed in the MicroWorlds EX Command Center. <pre>sendremotecommand "aon sendrc [aonfor 10]</pre>
sendremotemessage <i>number</i> sendrm <i>number</i>	Message? to find out if there's a message and/or message to "read" the message in the RCX. The input must be a number between 0 and 255.	There is a motor connected to port A and the following procedures must be downloaded to the RCX. <pre>to start loop [getinputfrommw] end to getinputfrommw if message? [aonfor message] end</pre> In the RCX Direct Mode Panel, type: start Type, in the MicroWorlds EX Command Center (they can also be run from buttons, clickable turtles, etc.) while the RCX is in the IR Transmitter's line of sight. <pre>sendremotemessage 10 sendrm 30</pre>
requestremotemessage <i>word-or-list</i> requestrm <i>word-or-list</i>	Nothing. The command is executed in the RCX and the result is sent back to MicroWorlds EX. The input is a word or a list that reports a number when executed inside the RCX. Requestremotemessage is a MicroWorlds EX reporter. You do not need to use remotemessage? and remotemessage to "read" the RCX's reply.	There is a text box on the page and a temperature sensor is connected to port 1 on the RCX. The following commands are executed within MicroWorlds EX (in the Command Center, by buttons, clickable turtles, etc.). The results are just examples. <pre>print requestrm "timer 2351 print requestrm [temp1 / 10] 24</pre> The temperature sensor actually reported 242. This was divided by <i>(continued)</i>

requestremotemessage <i>word-or-list</i> requestrm <i>word-or-list</i>	<p>10 (24.2) before being transmitted to MicroWorlds EX. The message is truncated. Compare that to:</p> <pre>print (requestrm "temp1) / 10 24.2</pre> <p>The temperature sensor actually reported 242. This was transmitted to MicroWorlds EX and MicroWorlds EX performed the division.</p>
---	--

When this is executed in the RCX	MicroWorlds EX Robotics must use or respond with...	Example
-------------------------------------	--	---------

sendmessage <i>number</i> sm <i>number</i>	<p>clearbuffer, remotemessage? and remotemessage.</p> <p>In the example, reflex is executed from a button in MicroWorlds EX. Click on the button, get a few feet away from the switch and rush to press the switch when you hear the motor. The RCX will tell MicroWorlds EX how much time it took you to react.</p> <p>The reflex procedure waits between 10 and 20 seconds and sends the instruction to run the howfast procedure to the RCX. The reflex procedure then waits for a message to arrive from the RCX.</p> <p>In the RCX, the howfast procedure resets the timer, turns on the motor, waits for the touch sensor to be pressed, and then sends a message containing the value of the timer (this is the time elapsed between resett and pressing the touch sensor). Finally, the motor is turned off.</p> <p>In MicroWorlds EX (in the reflex procedure), remotemessage? reports true when the message from the RCX arrives, ending the process started by the instruction waituntil[remotemessage?].</p> <p>Reflex then prints the contents of the message, divided by 10 (the number of seconds).</p>	<p>There is a motor connected to port A and a touch sensor connected to port 1 on the RCX. The following procedure is downloaded to the RCX.</p> <pre>to howfast resett aon waituntil [switch1] sendmessage timer aoff end</pre> <p>The following procedure is defined in the MicroWorlds EX project Procedures Tab.</p> <pre>to reflex clearbuffer wait 100 + random 100 sendrec [howfast] waituntil [remotemessage?] pr remotemessage / 10 end</pre>
---	--	--

Additional Information

Some robotics primitives have the same name as MicroWorlds EX primitives but work slightly differently. This section describes the differences between MicroWorlds EX and the RCX for the primitives **if**, **ifelse**, **waituntil**, **and**, **or** and **random**.

True and False are 1 and 0 in the RCX

If you are familiar with MicroWorlds EX, note the differences in the way the following primitives work when executed in the RCX.

If

When executed in the RCX, the first input of the primitive must report **1** or **0** instead of **true** or **false**.

Ifelse

When executed in the RCX, the first input of the primitive must report **1** or **0** instead of **true** or **false**.

Waituntil

When executed in the RCX, the first input of the primitive must report **1** or **0** instead of **true** or **false**.

Math and logic operators report **1** or **0** instead of **true** and **false**. This approach suits the requirements of **if**, **ifelse** and **waituntil**, which require **1** or **0** as input instead of **true** or **false**.

Examples:

```
if switch1 [aonfor 20]
ifelse switch2 [athisway] [athatway]
if timer > 100 [aoff]
waituntil [and switch1 switch2]
```

In the last example, **switch1** and **switch2** report **1** or **0** depending on their state (pressed or not). **And** takes these values (**0-0**, **0-1**, **1-0** or **1-1**) as input and reports **1** if both touch sensors reported **1**, **0** otherwise. **Waituntil** takes its input (**0** or **1**) from **and**.

This is in contrast to a similar instruction running in MicroWorlds EX (not in the RCX):

```
waituntil [and shape = 0 heading = 1]
```

In this example, **shape = 0** reports **true** if it is the case, **false** otherwise. The same goes for **heading**. Then **and** takes these values (**true-true**, **false-false**, **true-false** or **false-true**) as input and reports **true** if both its inputs are **true**. Finally, **waituntil** takes its input (**true** or **false**) from **and**.

Inputs for And and Or

Contrary to MicroWorlds EX, the primitives **and** and **or**, when running in the RCX, can only have two inputs. In MicroWorlds EX, these primitives accept more than two inputs if the primitive and their inputs are enclosed in parentheses.

In MicroWorlds EX:

```
if (and heading = 0 shape = 0 pos = [0 0])  
  [announce [I'm home!]]
```

In the RCX

```
if and switch1 sensor2 > 20 [aoff]
```

Section 5: Robotics

Vocabulary

This section describes the primitives that can run in the RCX and some that run in MicroWorlds EX to communicate with the RCX.

Motors and Sensors

These primitives can be used to write procedures to be downloaded into the RCX.

Primitive	Input(s)	Description	Example
angle1 angle2 angle3		Reports the value corresponding to the rotation recorded by a rotation sensor connected to the designated port (1, 2 or 3). A full rotation of its axle (360 degrees) increments or decrements the value reported by the rotation sensor by 16 units. See resetangle1 , resetangle2 , resetangle3 .	<p>There is a rotation sensor connected to port 1 and a motor (conveyor belt) is connected to port A.</p> <p>This procedure is downloaded to the RCX:</p> <pre>to feeder asetpower 2 resetangle1 aon waituntil [angle1 > 10] aoff end</pre> <p>This instruction is typed in the RCX Direct Mode Panel.</p> <pre>loop [wait 9999 feeder]</pre>
aoff boff coff		Turns off the motor connected to the designated port (A, B or C).	<p>There is a motor connected to port A.</p> <p>This instruction is typed in the RCX Direct Mode Panel.</p> <pre>aon wait 20 aoff</pre>

Primitive	Input(s)	Description	Example
aon bon con		Turns on the motor connected to the designated port(s) (A, B or C).	There is a motor connected to port A. This instruction is typed in the RCX Direct Mode Panel. <pre>aon wait 20 aoff</pre>
aonfor <i>number</i> bonfor <i>number</i> confor <i>number</i>		Turns on the motor connected to the designated port (A, B or C) for the duration of time indicated (<i>number</i>). The duration is measured in tenths of a second. The maximum duration is 3276 (327.6 seconds).	There is a motor connected to port A. This instruction is typed in the RCX Direct Mode Panel. <pre>aonfor 20</pre>
ard brd crd		Reverses the direction of the motor connected to the designated port (A, B or C) so it goes in the opposite direction.	There is a motor connected to port A. These instructions are typed in the RCX Direct Mode Panel. <pre>aon repeat 6 [ard wait 20] aoff</pre>
asetpower <i>number</i> bsetpower <i>number</i> csetpower <i>number</i>		Sets the power level for the designated port (A, B or C). The maximum power level is 7.	There is a motor connected to port A. This instruction is typed in the RCX Direct Mode Panel. <pre>aon asetpower 2 asetpower 7 aoff</pre> In the following example, switch1 is used to raise the power level of port A and switch2 decreases it. If both touch sensors are used at the same time, the motor stops. The following is typed in the RCX Procedures Panel and downloaded to the RCX: <pre>global [level] to variablespeed setlevel 3 asetpower 2 aon loop [control] end</pre> <i>(continued)</i>



Primitive <i>Input(s)</i>	Description	Example
		<pre> to control if switch1 [setlevel level + 1] if switch2 [setlevel level - 1] if and switch1 switch2 [aoff] asetpower level end </pre> <p>The following is typed in the RCX Direct Mode Panel:</p> <pre> variablespeed </pre> <p>Press one touch sensor or the other a few times to see the effect.</p>
athisway bthisway cthisway	Sets the motor connected to the designated port (A or B or C) to go "thisway". The actual direction of rotation depends on how the connector is placed on the RCX but if two connectors are placed in the same manner, setting one to thisway and the other one to thatway will make the motors run in opposite directions.	<p>There is a motor connected to port A and another connected to port B.</p> <p>These instructions are typed in the RCX Direct Mode Panel.</p> <pre> athisway bthatway aoff boff </pre>
athatway bthatway cthatway	Sets the motor connected to the designated port (A or B or C) to go "thatway". The actual direction of rotation depends on how the connector is placed on the RCX but if two connectors are placed in the same manner, setting one to thisway and the other one to thatway will make motors run in opposite directions.	See the example for athisway .

Primitive <i>Input(s)</i>	Description	Example
reflect1 reflect2 reflect3	<p>Reports 1 if the light sensor connected to the designated port (1, 2 or 3) has moved from a bright or reflective (shiny) surface to a dark surface. Reports 0 if the sensor has moved from a dark surface to a bright surface.</p> <p>The example runs an "automatic door" based on the fact that the light sensor will "sense" an incoming object (or person).</p>	<p>There is a light sensor connected to port 1 and a motor connected to port A of the RCX.</p> <p>This procedure is downloaded to the RCX:</p> <pre> to autodoor ard aonfor 20 waituntil [not reflect1] ard aonfor 20 waituntil [reflect1] end </pre> <p>This instruction is typed in the RCX Direct Mode Panel.</p> <pre> loop [autodoor] </pre>
resetangle1 resetangle2 resetangle3	<p>Resets the designated rotation sensor (1, 2 or 3) to 0. See angle1, angle2, angle3.</p>	<p>There is an angle sensor connected to port 1 and a motor (in a constructed conveyor belt) connected to port A.</p> <p>This procedure is downloaded to the RCX:</p> <pre> to feeder asetpower 2 resetangle1 aon waituntil [angle1 > 10] aoff end </pre> <p>This instruction is typed in the RCX Direct Mode Panel.</p> <pre> loop [wait 9999 feeder] </pre>

Primitive <i>Input(s)</i>	Description	Example
sensor1 sensor2 sensor3	<p>Reports the raw value of a sensor connected to the designated port (1, 2 or 3). The value depends on the type of sensor being used but is between 0 and 1023. When using this primitive, you should "calibrate" your sensor: experiment with your device to determine the minimum and maximum values that you get during your experiment. Use these values as thresholds in your procedures.</p>	<p>There is a motor connected to port A and a light sensor connected to port 1.</p> <p>This procedure is downloaded to the RCX:</p> <pre> to autodoor ard aonfor 20 waituntil [sensor1 > 125] ard aonfor 20 waituntil [sensor1 < 120] end </pre> <p>This instruction is typed in the RCX Direct Mode Panel:</p> <pre> loop [autodoor] </pre>
switch1 switch2 switch3	<p>Reports 1 (true) if a touch sensor connected to the designated port (1, 2 or 3) is pressed. Otherwise, reports 0 (false).</p>	<p>There is a touch sensor connected to port 1 and a motor connected to port A.</p> <p>This procedure is downloaded to the RCX:</p> <pre> to ride aon waituntil [switch1] aoff waituntil [not switch1] end </pre> <p>This instruction is typed in the RCX Direct Mode Panel.</p> <pre> loop [ride] </pre>
temp1 temp2 temp3	<p>Reports the value corresponding to the reading of the thermal sensor connected to the designated port (1, 2 or 3). The value is the temperature in degrees Celsius multiplied by 10.</p>	<p>Example: There is a temperature sensor connected to port 1 and a motor (fan) connected to port A.</p> <p>This procedure is downloaded into the RCX:</p> <pre> to autofan waituntil [temp1 > 25] aon waituntil [temp1 < 20] aoff end </pre> <p>This instruction is typed in the RCX Direct Mode Panel.</p> <pre> loop [autofan] </pre>

Control

These primitives can be used to write procedures to be downloaded to the RCX. Except for **stop** and **output**, they can also be used in the RCX Direct Mode Panel.

Primitive	Input(s)	Description	Example
if <i>1 or 0 (true or false)</i> <i>list-of-instructions</i>		Runs the <i>list-of-instructions</i> if the first input (a conditional statement that is either true or false) is true (reports 1). Note that =, <, >, and , not , or , as well as switch1 , switch2 and switch3 report 1 or 0 .	There is a touch sensor connected to port 1 and a motor connected to port A. This procedure is downloaded to the RCX: <pre>to maybeverse if switch1 [ard] aonfor 20 end</pre> This instruction is typed in the RCX Direct Mode Panel. <pre>loop [maybeverse]</pre>
ifelse <i>1 or 0 (true or false)</i> <i>list-of-instructions1</i> <i>list-of-instructions2</i>		Runs the first <i>list-of-instructions</i> if the first input (a conditional statement that is either true or false) is true (reports 1). Runs the second <i>list-of-instructions</i> if the condition is false (reports 0). Note that =, <, >, and , not , or , as well as switch1 , switch2 and switch3 report 1 or 0 .	There is a touch sensor connected to port 1 and a motor connected to port A. This procedure is downloaded to the RCX: <pre>to setdirandrundrun ifelse switch1 [athisway] [athatway] aonfor 20 end</pre> This instruction is typed in the RCX Direct Mode Panel. <pre>loop [setdirandrundrun]</pre>
loop <i>list-of-instructions</i>		Keeps running <i>list-of-instructions</i> indefinitely.	There is a touch sensor connected to port 1 and a motor connected to port A. This procedure is downloaded to the RCX: <pre>to ride aon waituntil [switch1] aoff waituntil [not switch1] end</pre> This instruction is typed in the RCX Direct Mode Panel. <pre>loop [ride]</pre>

Primitive	Input(s)	Description	Example
output <i>word-or-list</i>		Stops the procedure and reports a value. This primitive can only be used in a procedure.	There is a touch sensor connected to port 1 and a thermal sensor connected to port 2.
op <i>word-or-list</i>		In the example, each time the RCX sends a message, it is read by MicroWorlds EX and printed in the text box.	<p>These procedures are downloaded to the RCX:</p> <pre>to senddata sendmessage temperature end to temperature waituntil [switch1] output temp2 end</pre> <p>Write the readdata procedure in the MicroWorlds EX project Procedures Tab.</p> <pre>to readdata if remotemessage? [pr remotemessage] end</pre> <p>Create a text box on the MicroWorlds EX page. Create a button and type readdata in the button's instruction field. Set the mode to Forever.</p> <p>This instruction is typed in the RCX Direct Mode Panel.</p> <pre>loop [senddata]</pre>
repeat <i>number list-of-instructions</i>		Runs the list-of-instructions the specified number of times.	<p>There is a motor connected to port A.</p> <p>This procedure is downloaded to the RCX:</p> <pre>to washcycle aon repeat 10 [wait 10 ard] aoff end</pre> <p>This instruction is typed in the RCX Direct Mode Panel.</p> <pre>washcycle</pre>
resett		Resets the value of the RCX internal timer to 0.	See the example for timer .

Primitive <i>Input(s)</i>	Description	Example
<p>stop</p>	<p>Stops the procedure that is running. This primitive can only be used in a procedure.</p> <p>In this example, the power is increased until the touch sensor connected to port 1 is pressed.</p>	<p>There is a motor connected to port A, and a touch sensor connected to port 1.</p> <p>This procedure is downloaded to the RCX:</p> <pre> to calibrate aon setpower 2 wait 20 if switch1 [stop] setpower 4 wait 20 if switch1 [stop] setpower 6 wait 20 if switch1 [stop] setpower 8 wait 20 if switch1 [stop] end </pre> <p>This instruction is typed in the RCX Direct Mode Panel. Press the touch sensor when the motor speed seems adequate.</p> <pre> calibrate </pre>
<p>timer</p>	<p>Reports the value of the internal timer in tenths of a second.</p> <p>The timer is set to 0 when the RCX is turned on and when the primitive resett is executed. It is limited to 16370 (1,637 seconds or more that 27 minutes). It should therefore be uses to time events that take place within this limit. See resett.</p>	<p>See the example for resett.</p>
<p>wait <i>number</i></p>	<p>Causes a pause in the execution of a program or instruction. The time (<i>number</i>) is measured in tenths of a second.</p>	<p>There is a touch sensor connected to port 1 and a motor connected to port A.</p> <p>This procedure is downloaded to the RCX.</p> <pre> to reflex wait 50 + random 50 resett aon waituntil [switch1] display timer wait 20 aoff end </pre> <p>This instruction is typed in the RCX Direct Mode Panel.</p> <pre> reflex </pre>

Primitive	Input(s)	Description	Example
waituntil <i>list-of-instructions</i> (reports 1 or 0)		Runs the <i>list-of-instructions</i> repeatedly until it reports true (1) . When this happens, waituntil stops running the instruction and MicroWorlds proceeds to the next instruction.	See the example for wait .

Audio-Display

This primitives can be used to write procedures that can be downloaded to the RCX.

Primitive	Input(s)	Description	Example
beep		Plays a short beep.	This instruction is typed in the Cricket Direct Mode Panel. <code>repeat 5 [beep wait 5]</code>
display number		<p>Displays a numeric message on the RCX display screen. For example, it could display the output of a sensor (display temp1 wait 20).</p> <p>A display command must be followed by a wait command so the value remains on the display for some time. Otherwise, the number is displayed so quickly that it can hardly be noticed.</p> <p>The RCX can display numbers between -9999 and 9999.</p>	<p>There is a motor connected to port A and a touch sensor connected to port 1. This procedure is downloaded to the RCX:</p> <pre>to calibrate asetpower 1 aon wait 20 if switch1 [display 1 wait 20 stop] asetpower 2 aon wait 20 if switch1 [display 2 wait 20 stop] asetpower 3 aon wait 20 if switch1 [display 3 wait 20 stop] asetpower 4 aon wait 20 if switch1 [display 4 wait 20 stop] asetpower 5 aon wait 20 if switch1 [display 5 wait 20 stop] asetpower 6 aon wait 20 if switch1 [display 6 wait 20 stop] asetpower 7 aon wait 20 display 7 wait 20 end</pre> <p>This instruction is typed in the RCX Direct Mode Panel. <code>calibrate</code></p>

Primitive <i>Input(s)</i>	Description	Example
note <i>pitch</i> <i>duration</i>	Plays a note using the specified <i>pitch</i> and <i>duration</i> . <i>Duration</i> is measured in tenths of a second. Note that the RCX does not wait for the end of the note before starting the next instruction.	There is a motor connected to port A. These instructions are typed in the RCX Direct Mode Panel. Notice the difference between the two sets of instructions. <pre>aon note 65 10 aoff aon note 65 10 wait 10 aoff</pre>

Math and Logic

These primitives can be used to write procedures to be downloaded to the RCX.

Primitive <i>Input(s)</i>	Description	Example
()	Parentheses are used in the RCX to determine the order of mathematical operations. In the example, without the parentheses, only temp2 would be divided by 2 to get the average and by 10 to get the value in Celsius (divided by 20 in the example to simplify the instruction), and the result would be added to temp1 . The * and / operations are normally executed before the + and - operations. The parentheses are used here to force a different sequence.	There is a motor connected to port A and two temperature sensors connected to ports 1 and 2. The following instruction triggers the motor if the average temperature reported by the two sensors is above 25 degrees Celsius: <pre>if ((temp1 + temp2) / 20) > 25 [aon]</pre>
+ - * /	These mathematical operators follow the same rules as those in MicroWorlds EX.	See example for () above.
= > <	These mathematical operators follow the same rules as those in MicroWorlds EX except they report 1 and 0 instead of true or false . They are generally used in if , ifelse and waituntil instructions because these primitives require a 1 or a 0 as their first input. See the examples.	There is a motor connected to port A. These instructions are typed in the RCX Direct Mode Panel. <pre>if timer > 2000 [aon] reset aon waituntil [timer > 10000] aoff</pre>

Primitive	Input(s)	Description	Example
and 1 or 0 <i>(true-or-false)</i> 1 or 0 <i>(true-or-false)</i>		Reports 1 (true) if all its inputs report 1 (true).	<p>There are motors connected to ports A and B and touch sensors connected to ports 1 and 2.</p> <pre> to drive aon bon ifelse switch1 [athisway] [athatway] ifelse switch2 [bthisway] [bthatway] if and switch1 switch2 [aoff boff] end </pre> <p>This instruction is typed in the RCX Direct Mode Panel.</p> <pre> loop [drive] </pre>
not 1 or 0 <i>(true-or-false)</i>		Reports the logical inverse of its input.	<p>There is a motor connected to port A and a touch sensor connected to port 1.</p> <p>This instruction is typed in the RCX Direct Mode Panel. Press the touch sensor before you send the instructions to the RCX. Release the button to stop the motor.</p> <pre> aon waituntil [not switch1] aoff </pre>
or 1 or 0 <i>(true-or-false)</i> 1 or 0 <i>(true-or-false)</i>		Reports 1 (true) if any of its inputs report 1 (true).	<p>There are motors connected to ports A and B and touch sensors connected to ports 1 and 2.</p> <pre> to drive aon bon if or switch1 switch2 [aoff boff] end </pre> <p>This instruction is typed in the RCX Direct Mode Panel.</p> <p>The motors are turned off if either of the switches (one or both) is pressed.</p> <pre> loop [drive] </pre>

Primitive <i>Input(s)</i>	Description	Example
random <i>number</i>	Reports a random non-negative integer less than <i>number</i> . The maximum value for <i>number</i> is 9999.	There is a touch sensor connected to port 1 and a motor connected to port A. This procedure is downloaded to the RCX : <pre> to reflex wait 50 + random 50 resett aon waituntil [switch1] display timer aoff wait 20 end </pre> This instruction is typed in the RCX Direct Mode Panel. <pre> reflex </pre>
remainder <i>number1</i> <i>number2</i>	Reports the remainder when <i>number1</i> is divided by <i>number2</i> . The remainder of a negative number is negative. If <i>number1</i> and <i>number2</i> are non-integers, they are truncated.	There is a rotation sensor connected to port 1. This instruction is typed in the RCX Direct Mode Panel. It displays the value of the rotation sensor in degrees. <pre> loop [display (remainder angle1 16) / 16 * 360 wait 10] </pre>

Variables

This primitive can be used in the RCX Procedures Panel but not in the Direct Mode Panel.

Primitive	Input(s)	Description	Example
global <i>list</i>		<p>Assigns a state variable to the RCX and creates the following command-reporter pair of primitives that can be used in the procedures downloaded into the RCX: the word set followed by the name of the variable (for example, global "level creates a setlevel command); and, the name of the variable (in this example, the reporter level).</p> <p>The command global and its input must be typed in the RCX Procedures Panel, but not inside a procedure (before, between or after procedures).</p>	<p>There is a motor connected to port A and two touch sensors connected to ports 1 and 2.</p> <p>In this example, switch1 is used to raise the power level of the motor connected to port A and switch2 decreases it. If both touch sensors are pressed at the same time, the motor stops.</p> <p>The following is typed in the RCX Procedures Panel and downloaded to the RCX:</p> <pre>global [level] to variablespeed setlevel 3 asetpower 3 aon loop [control] end to control if switch1 [setlevel level + 1] if switch2 [setlevel level - 1] if and switch1 switch2 [aoff] asetpower level end</pre> <p>The following is typed in the RCX Direct Mode Panel:</p> <pre>variablespeed</pre> <p>Press one touch sensor or the other a few times to see the effect.</p>

Messaging Primitives Running In the Cricket

These primitives can be used to write procedures to be downloaded to the RCX. The RCX must be in the IR Transmitter's line of sight when these primitives are used.

Primitive	Input(s)	Description	Example
sendmessage <i>number</i>		Sends a value between -16383 and 16383 to MicroWorlds EX. It is commonly used to send the value of the timer or that of a sensor. If the input is not an integer, it is truncated. The primitives remotemessage? and remotemessage must be used within a MicroWorlds EX project to receive and "read" the message sent by the RCX.	There is a motor connected to port A and a touch sensor connected to port 1 on the RCX.
sm <i>number</i>		In the example, reflex is executed from a button in MicroWorlds EX. Click on the button, get a few feet away from the touch sensor and rush to press it when you hear the motor. The RCX will tell MicroWorlds EX how much time it took you to react. The reflex procedure waits between 10 and 20 seconds and sends the instruction to run the howfast procedure to the RCX. The reflex procedure then waits for a message to arrive from the RCX. On the RCX side, the howfast procedure resets the timer, starts the motor, waits for the touch sensor to be pressed, and then sends a message containing the value of the timer (this is the time elapsed between reset and pressing the switch). Back in MicroWorlds EX (in the reflex procedure), remotemessage? reports true when the message from the RCX arrives and this ends the process started by the instruction: waituntil [remotemessage?] Reflex then prints the contents of the message, divided by 10 (the number of seconds). Finally, the motor is turned off.	The following procedure is downloaded to the RCX. <pre>to howfast resett aon waituntil [switch1] sendmessage timer wait 20 end</pre> The following procedure is defined in the MicroWorlds EX project Procedures Tab. <pre>to reflex clearbuffer wait 100 + random 100 sendremotecommand [howfast] waituntil [rm?] pr remotemessage / 10 sendrc [aoff] end</pre> Create a text box on the MicroWorlds EX page. To test your reflexes, first, make sure the RCX is in the IR Transmitter's line of sight. Then, in the MicroWorlds EX Command Center, type: <pre>reflex</pre>

Primitive <i>Input(s)</i>	Description	Example
message?	<p>This primitive is executed in the RCX and it reports 1 (true) if a message was sent by MicroWorlds EX using the command sendremotemessage. Use message to read the message.</p>	<p>There is a motor connected to port A of the RCX.</p> <p>The following procedure is downloaded to the RCX.</p> <pre> to start loop [getinputfrommw] end to getinputfrommw if message? [aonfor message] end </pre> <p>In the RCX Direct Mode Panel, type:</p> <pre>start</pre> <p>The following commands are executed within MicroWorlds EX (in the Command Center, by buttons, clickable turtles, etc.) while the RCX is in line of sight with the IR Transmitter.</p> <pre>sendremotemessage 10 sendrm 30</pre>
message	<p>Reports the message sent by MicroWorlds EX. Reports 0 if there is no message to be read. It is best to use message? to find out if there is a message waiting to be read before using message.</p>	<p>See the example for message?</p>

Messaging Primitives Running Within MicroWorlds EX

These primitives are used within MicroWorlds EX (from the Command Center, by buttons, turtles, etc.) to download procedures, instructions or messages to the RCX. They cannot be used in the RCX Tab. The RCX must be in the IR Transmitter's line of sight when these primitives are used.

Primitive	Input(s)	Description	Example
download <i>word</i>		<p>Downloads its input to the RCX. This is equivalent to clicking the Download button with the same text in the RCX Procedures Panel.</p> <p>The input for download is a word. The most common input for download is the contents of a text box. You can take advantage of the fact that the name of a text box reports the text box' contents as a "long word". Therefore, the name of a text box can be used as input for download.</p> <p>In the instruction download text1, text1 reports its contents as a long word. This long word is used by download as input.</p>	<p>Text1 contains the procedure goforit. Text1 reports its contents to download, which sends it to the RCX. The next instruction, goforit, starts the procedure in the RCX.</p> <p>These instructions can be typed in the MicroWorlds EX Command Center or executed from a button, a programmed color or a turtle.</p> <pre>download text1 sendrc "goforit</pre>
sendremotecommand <i>word</i> or <i>list-of-instructions</i>		<p>Sends <i>list-of-instructions</i> to the RCX where they are immediately run by the RCX.</p> <p>The input is a word or a list that contains instructions that can be run in the RCX (either procedures that have been downloaded to the RCX or primitives that can run in the RCX).</p>	<p>There is a motor connected to port A.</p> <p>These instructions are typed in the MicroWorlds EX Command Center.</p> <pre>sendremotecommand "aon sendremotecommand "aoff sendrc [aonfor 10]</pre>
sendrc <i>word</i> or <i>list-of-instructions</i>			

Primitive	Input(s)	Description	Example
sendremotemessage <i>number</i>		Sends a number as a message to the RCX. The RCX must use message? to determine if a message was sent and/or message to find out what the message is.	There is a motor connected to port A.
sendrm <i>number</i>		The input must be a number between 0 and 255.	<p>The following procedure is downloaded to the RCX.</p> <pre> to start loop [getinputfrommw] end to getinputfrommw if message? [aconfor message] end </pre> <p>In the RCX Direct Mode Panel, type:</p> <pre> start </pre> <p>The following commands are executed within MicroWorlds EX (in the Command Center, by buttons, clickable turtles, etc.) while the RCX is in the IR Transmitter's line of sight.</p> <pre> sendremotemessage 10 sendremotemessage 30 </pre>

These primitives are used within MicroWorlds EX in order to detect and use incoming messages sent by the RCX. The RCX must be in the IR Transmitter's line of sight when these primitives are used.

Primitive	Input(s)	Description	Example
clearbuffer		<p>Empties the message buffer in the RCX before starting a communication session between the RCX and MicroWorlds EX. A message sent by the RCX (using the primitive sendmessage) stays in the RCX buffer until remotemessage is used in MicroWorlds EX to receive and "read" the message. The message buffer may contain messages that have been previously sent by the RCX but that have not yet been "read" by MicroWorlds EX.</p> <p>In the example, reflex is executed from a button in MicroWorlds EX. Click on the button, get a few feet away from the touch sensor and rush to press it when you hear the motor. The RCX will tell MicroWorlds EX how much time it took you to react.</p> <p>The reflex procedure waits between 10 and 20 seconds and sends the instruction to run the howfast procedure to the RCX. The reflex procedure then waits for a message to arrive from the RCX.</p> <p>On the RCX side, the howfast procedure resets the timer, starts the motor, waits for the touch sensor to be pressed, and then sends a message containing the value of the timer (this is the time elapsed between resett and pressing the switch).</p> <p>Back in MicroWorlds EX (in the reflex procedure), remotemessage? reports true when the message from the RCX arrives and this ends the instruction waituntil [remotemessage?].</p> <p>Reflex then prints the contents of the message, divided by 10 (the number of seconds). Finally, the motor is turned off.</p>	<p>There is a motor connected to port A and a touch sensor connected to port 1 of the RCX.</p> <p>The following procedure is downloaded to the RCX.</p> <pre> to howfast resett aon waituntil [switch1] sendmessage timer wait 20 end </pre> <p>The following procedure is defined in the MicroWorlds EX project Procedures Tab.</p> <pre> to reflex clearbuffer wait 100 + random 100 sendrc [howfast] waituntil [rm?] pr remotemessage / 10 sendrc [aoff] end </pre> <p>Create a text box on the MicroWorlds EX page. To test your reflexes, first make sure the RCX is in the IR Transmitter's line of sight. Then, in the MicroWorlds EX Command Center, type:</p> <pre> reflex </pre>

Primitive	Input(s)	Description	Example
remotemessage? rm?		Reports 1 (true) if there is a message to be "read" in the message buffer. The message must have been sent by the RCX using the command sendmessage . Reports 0 (false) otherwise. Use remotemessage to read the message.	<p>There is a text box on the page. There is a touch sensor connected to port 1 and any type of sensor connected to port 3.</p> <p>This procedure is created in the MicroWorlds EX project Procedures Tab.</p> <pre> to trackdata if remotemessage? [pr remotemessage] end </pre> <p>The following procedures are downloaded to the RCX:</p> <pre> to start loop [sendonclick] end to sendonclick if switch1 [sendmessage sensor3] end </pre> <p>Create a button in MicroWorlds EX. Write trackdata in the instruction field and set the mode to Forever. Each time the RCX sends a message, it is read by MicroWorlds EX and printed in the text box.</p> <p>Start the button in MicroWorlds EX. Press the touch sensor to send the sensor's value to MicroWorlds EX.</p>
remotemessage rm		Reports the message sent by the RCX. Reports 0 if there is no message to be read. It is best to use remotemessage? to find out if there is a message waiting to be read before using remotemessage .	See the example for remotemessage?

Primitive <i>Input(s)</i>	Description	Example
<p>requestremotemessage <i>word</i> or <i>list-of-instructions</i></p> <p>requestrm <i>word</i> or <i>list-of-instructions</i></p>	<p>Sends the <i>word</i> or <i>list-of-instructions</i>, each of which reports a value, to the RCX, waits for the value reported by the RCX and reports that value.</p>	<p>There is a text box on the page. Any type of sensor is connected to port 2.</p> <p>This instruction is executed in the MicroWorlds EX Command Center. The values are just examples:</p> <pre>pr requestremotemessage "timer 25365 pr requestrm "sensor2 1021</pre> <p>The following delay procedure is downloaded to the RCX and a touch sensor is connected to port 1.</p> <pre>to delay resett waituntil [switch1] output timer end</pre> <p>This instruction is executed in the MicroWorlds EX Command Center:</p> <pre>pr requestrm "delay</pre> <p>Press the touch sensor (the value is just an example).</p> <pre>2134</pre>