

RoboRemo v2.1.2

User Manual

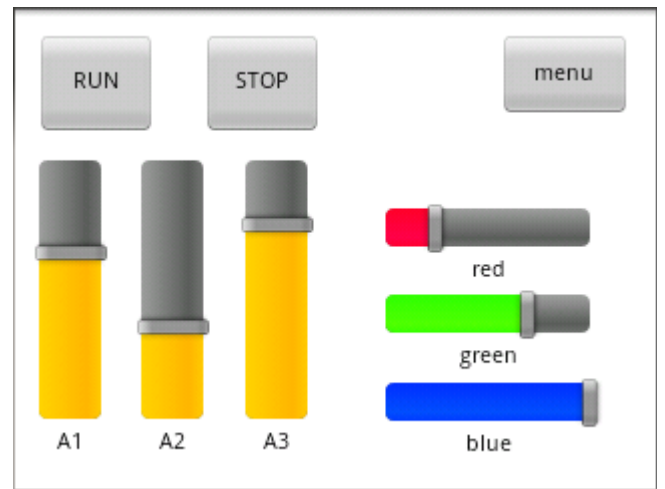
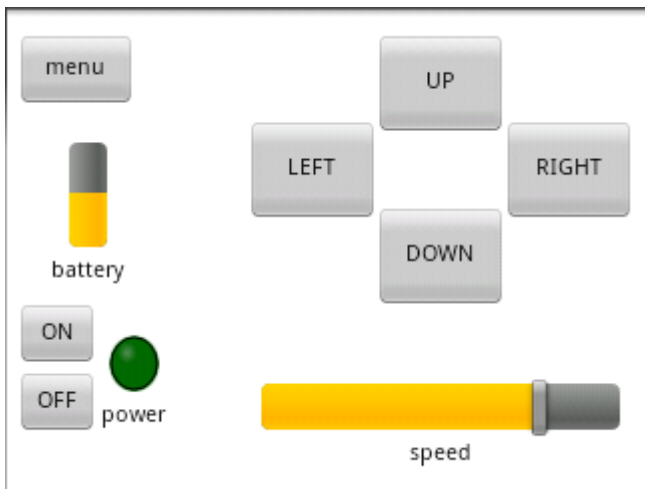
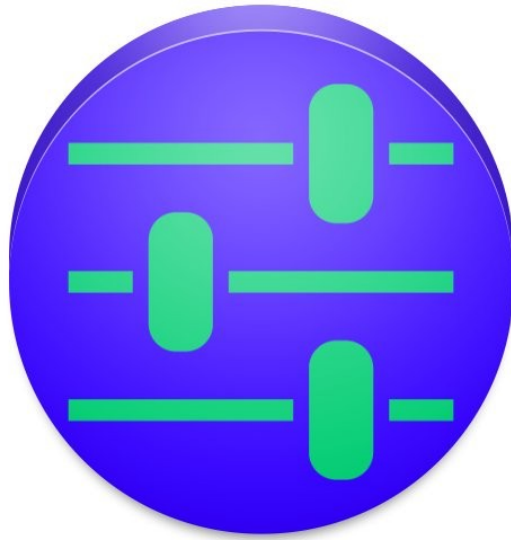


Table of Contents

General Description.....	3
Bluetooth / WiFi / Ethernet / USB modules.....	4
Available interface items.....	6
Building the interface.....	9
Common edit options.....	10
Button edit options.....	10
Local actions.....	11
Slider edit options.....	12
Led edit options.....	13
Led examples.....	13
Level indicator edit options.....	14
Text log edit options.....	14
Accelerometer edit options.....	15
Text field edit options.....	15
Plot edit options.....	16
Image edit options.....	17
Touchpad edit options.....	18
Kbd connector edit options.....	19
Kbd connector examples.....	19
Heartbeat sender edit options.....	20
Touch stopper edit options.....	20
Vibrator edit options.....	20
File sender edit options.....	21
File receiver edit options.....	22
Speaker edit options.....	23
Printf() edit options.....	24
Menu options.....	25
App. settings.....	26
Interface menu options.....	27
New features and fixed bugs.....	28

General Description

RoboRemo is a user **customizable** remote control application intended mainly for **electronics hobby projects**. RoboRemo can connect via **Bluetooth (RFCOMM / BLE)**, **Internet** or **WiFi (TCP, UDP)**, and **USB (CDC-ACM, FTDI, CP210X, CH340)**.

Disclaimer: Don't use RoboRemo for life support systems or any other situations where system failure may affect user or environmental safety. Please don't use RoboRemo in projects where high-level security is required.

To connect via **Bluetooth**, a remote device must contain a **Bluetooth to Serial adapter** like **BlueSMiRF, BTM-222, HC-05, HC-06, etc.** or **BLE Serial adapter** like **HM-10 CC2540** and a **microcontroller** programmed to interpret commands from RoboRemo. You can also find adapters for **WiFi (like ESP8266)** or **Ethernet**. For **USB connection**, your phone / tablet must have **USB API** and **USB OTG**. For **BLE connection**, your phone / tablet must have **BLE support**.

All the **commands** from RoboRemo are **text strings**, ending with **command ending** which is **LF** character **'\n'** (hex code 0x0A) by default, but **user can change it**. For example if you configure a button to send “abc” when pressed, it will send “abc\n” if command ending is '\n' or it will send “abcqwerty123” if command ending is “qwerty123”. The **command ending** is used by the microcontroller program **to know where each command ends**.

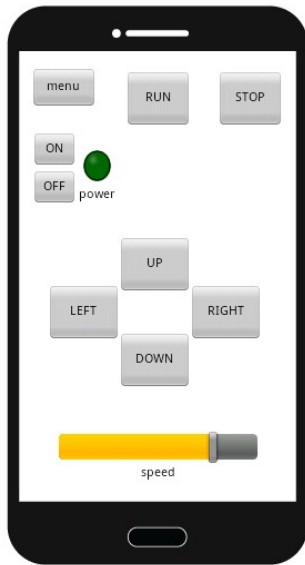
RoboRemo also can receive commands for updating some interface items (for example you can have a battery level indicator to monitor the battery level of the remote device). These **input commands also must end with command ending**.

It is **not** recommended to set command ending to **empty string**, because receiving commands from microcontroller to app will not work properly. (App “will think” that command ends after each received character). However the commands from app to microcontroller will be sent as expected (a button configured to send “abc” will send “abc”).

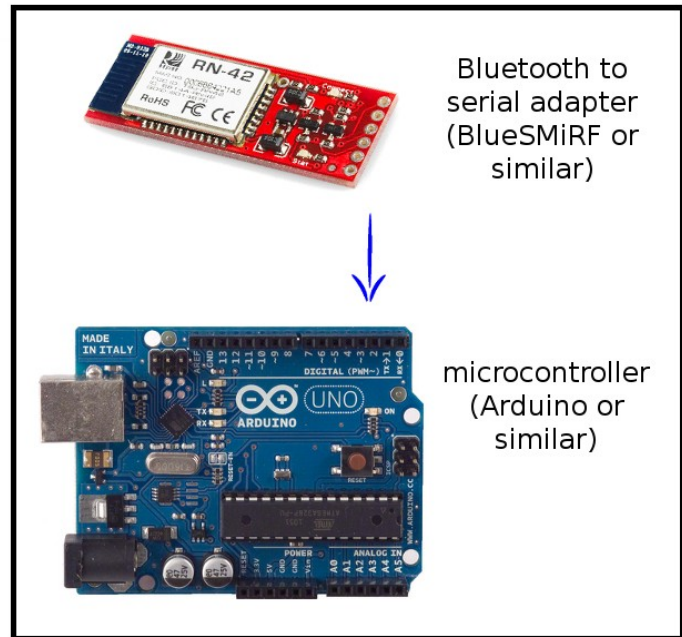
Bluetooth / WiFi / Ethernet / USB modules

RoboRemo was tested and works with these modules: BlueSMiRF, HC-05, HC-06, BTM-222, HM-10, CC2540, ESP8266 (ESP-01), HUZDAH ESP, Arduino UNO, Arduino Mega (Arduino connected directly with USB cable), but other modules should work too.

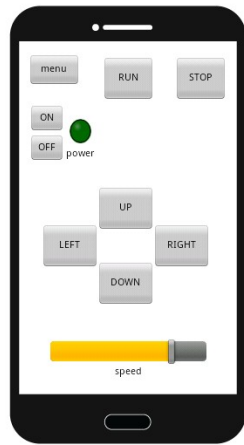
When using Bluetooth module and microcontroller, please **make sure** the microcontroller uses **same BaudRate** as the module. The default BaudRate for BlueSMiRF is usually **115200**, and for HC-05 / HC-06 is usually **9600**.



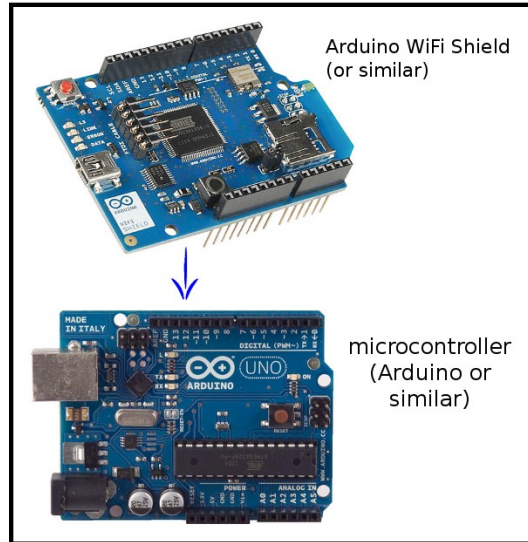
Android device
with RoboRemo app



Remote device



Android device with RoboRemo app



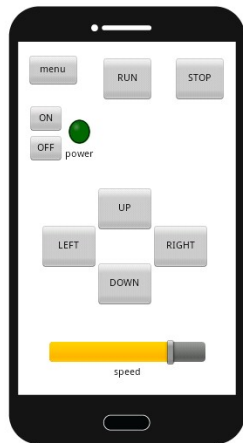
Arduino WiFi Shield (or similar)

microcontroller (Arduino or similar)

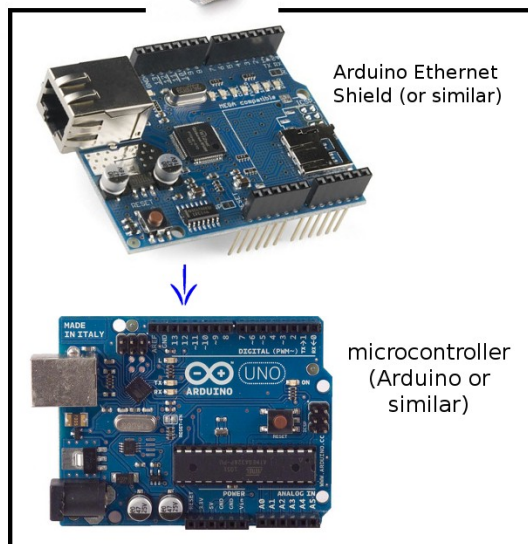
Remote device



GSM



Android device with RoboRemo app





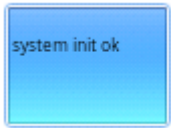
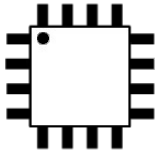
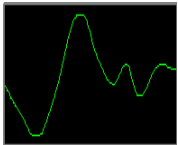



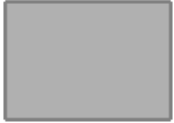






Arduino Ethernet Shield (or similar)



microcontroller (Arduino or similar)

Remote device

Available interface items

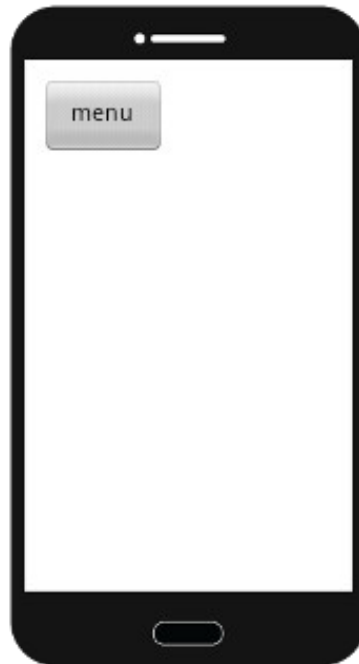
item name	photo	description
button		Buttons can be used for sending commands to the remote device. A button can be configured to send one command when pressed and other command when released.
slider		Sliders can be used to vary remote device parameters, for example volume control, or motor speed or light intensity.
led		LEDs can be used as indicators for remote device states.
level indicator		A level indicator item can be used to display data from remote sensors, for example temperature or battery level.
text log		A text log item is a log screen that can be used to display debug messages from the remote device.
accelerometer		The accelerometer can be used for sending periodically the acceleration measured by the phone's accelerometer. You can set the ids the 3 axes and the repeat (refresh) period.
text field	text	Text field item, to add some text to the interface. Note: if you want an item to be displayed on top of text field, copy the item, then remove it, then paste it.
plot		Plot item can be used to display some signals.

image*		Image item can be used to display an image that is sent from the microcontroller. *available only in RoboRemo Full version
touchpad		Touchpad item can be used to send position and motion of your finger.
kbd connector		Kbd connector can be used to send commands using the keyboard.
heartbeat sender		heartbeat sender can be used to periodically send some string (like a heartbeat signal). The remote device can watch for this signal to make sure it is still connected.
touch stopper		If you put a reset button for example, surround it with touch stoppers, so it doesn't get pressed when you touch the screen near to it. When you exit ui editor, touch stoppers become invisible, so they don't annoy you.
vibrator*		Vibrator item gives the remote device access to the phone's vibrator (you can send a command from microcontroller to make your phone vibrate). *available only in RoboRemo Full version.
file sender*		File sender item can be used to send files from the phone's sdcard to the microcontroller. *available only in RoboRemo Full version.
file receiver*		File receiver item can be used to receive files from the microcontroller and store them on the phone's sdcard. *available only in RoboRemo Full version.

speaker*		Speaker item gives the remote device access to the phone's speaker (you can send a command from microcontroller to make your phone beep). *available only in RoboRemo Full version.
printf()		Gets the data from other items and sends a formatted packet. User can configure the format. Currently, the printf() item can get data from button, slider, accelerometer and heartbeat sender. See printf() edit options for more details.

Building the interface

First when you open the RoboRemo app, it will look like this:



To edit the interface, click **menu** → **edit ui**.

Now in edit mode, **click on blank** space to add an item, then choose the item type.

Item will appear on screen. You can **move** it by dragging the **top left corner**, and **resize** by dragging the **bottom right corner**.

To edit item options, click on the desired item.

Common edit options

option	description
copy	Copies the item so you can paste it when adding a new item.
remove	Removes the item.
lock edit	Locks the item, so you can not edit it anymore. This is useful if you want to add other items on top of this item. You can unlock all items from menu → interface → unlock edit all.

Button edit options

option	description
set text	Sets the text that appears on button.
set press action	For remote action type (default): Sets the string to send to the remote device when you press the button. (RoboRemo will append the command ending to mark the end for each command). For local action type: Sets the local action to execute when you press the button.
set release action	For remote action type (default): Sets the string to send to the remote device when you release the button. (RoboRemo will append the command ending to mark the end for each command). For local action type: Sets the local action to execute when you release the button.
set text size	Sets the font size for the text that appears on the button.
set image*	User can change the button image, can select different images for pressed button and released button. The images are stored in the interface file. *available only in RoboRemo Full version.
repeat	Sets the button repeat options. When you hold a button, it will first repeat after the delay value, then after period. User can choose to repeat both press and release actions, or only the press action.
set local / set remote	Changes button action type from remote to local, from local to remote. Default is remote.

Local actions

action	description
“iface n” or “interface n”	select the nth interface. (n from 0 to 7)
“echo abc”	will simulate the receiving of “abc” command.
“send abc”	will send “abc” (followed by command ending) to the remote device.
“sendnce abc”	will send “abc” (without command ending).
“sendhex 4A4B4C”	will send “JKL” (without command ending).
“disconnect”	disconnect from the remote device
“connect rfcomm 11:22:33:44:55:66 1” or “connect rfcomm 112233445566 1”	connect via Bluetooth RFCOMM to the remote device with address 11:22:33:44:55:66 and port 1.
“connect rfcomm e 112233445566 1”	Also use encryption.
“connect rfcomm 112233445566 d”	Connect using the default UUID
“connect rfcomm 112233445566 f”	Connect using the first UUID
“connect ble 112233445566 0000ffe0-0000-1000-8000-00805f9b34fb 0000ffe1-0000-1000-8000-00805f9b34fb 0000ffe1-0000-1000-8000-00805f9b34fb 0 0”	Connect to BLE device with address 11:22:33:44:55:66. The 3 UUIDs are: service UUID, TX UUID, RX UUID. The last 2 parameters “0 0” will be used in the future. Since manually writing the UUIDs is not fun, it is recommended to use these steps: connect to the device using menu → connect, edit the interface, add a button, set local, set press action to “con” (without “”) and it will complete automatically.
“connect tcp 192.168.0.5:9876”	Connect via TCP/IP to the remote device with IP 192.168.0.5 and port 9876.
“connect udp 192.168.0.5:9876”	Connect via UDP. Packets will be sent to the port 9876 of the remote device with IP 192.168.0.5 Phone / tablet will listen for packets on port 9876.
“connect usb 115200”	Connect via USB using BaudRate 115200.
“con”	If there is an active connection and button is set to local, then setting the press or release action to “con” will auto-complete it with the string for the current connection. (No need to manually write the IP and port or the Bluetooth device address).

Note: Starting with version 1.9.1, local action can contain **multiple commands**, separated by command ending. For example:

- set command ending to “\n”
- add a text log
- add a button, set it local
- set its press action to “**echo sending abc\nsend abc**”

Now when you press the button, “sending abc” will appear in the text log and “abc” will be sent to the remote device.

Slider edit options

option	description
set id	Sets the slider id string. For example if id is “s1” and you move the slider to the value 100 and command ending is “\n”, it will send “s1 100\n” (id followed by space followed by value followed by command ending).
set label	Sets the text string to appear under the slider. You can use the slider value inside the label. Also you can set the decimal count for the value that appears in the label. Examples for slider with value 100: label “speed = #*0.1” will show “speed = 10.0” label “x = #*-5+10 cm” will show “x = -490 cm”
set min, max	Sets the minimum and maximum values for the slider, and the output type (int or float). For float output type, user can also set the decimal count and the step count.
send when moved / send when released	Sets the slider send mode, default mode it to send when released.
set color	Sets the slider color.
send space / don't send space	Select if you want the slider to send space character after id or not.
auto return / don't auto return	Select if you want the slider to return to its center when released or not.
set return value	Set the return value to be used for auto return when released. You can enter an integer, or min/mid/max. Default is mid.
set repeat period	Set the repeat period (in ms) for slider. Empty string = don't repeat (default). When repeat enabled, the slider will also send its data periodically, in addition to normal operation.

Led edit options

option	description
set id	Sets the led id string.
set on command	Sets the led on command, default is "1".
set off command	Sets the led off command, default is "0".
set label	Sets the text string to appear under the led
set color	Sets the led color.
set on timeout	Set the timeout after which led will turn off if it does not receive on command. You can program your remote device to periodically send a command to turn on an led, and set the on timeout so that you will see the led will turn off if the connection is lost.
set off timeout	Set the timeout after which led will turn on if it does not receive off command.

Led examples

Led id	led	ledA	ledB
Led on command	1	on	#>=10
Led off command	0	off	#<10
Interface command ending	\n	;	ok
To turn the led on, microcontroller must send	led 1\n	ledA on;	ledB 10ok or ledB 11ok etc.
To turn the led off, microcontroller must send	led 0\n	ledA off;	ledB 9ok or ledB 8ok etc.

Level indicator edit options

option	description
set id	Sets the level indicator id string. For example if id is “batt” and command ending is “\n”, you can set its level to 100 by sending “batt 100\n” from microcontroller.
set label	Sets the text string to appear under the level indicator. You can use the level indicator value inside the label. Also you can set the decimal count for the value that appears in the label. Examples for level indicator with value 100: label “temp. = #*0.5-10 °C” will show “temp. = 40.0 °C” label “batt. #*1%” will show “batt. 100%”
set min	Sets the minimum value, default is 0.
set max	Sets the maximum value, default is 255.
set color	Sets the level indicator color.

Text log edit options

option	description
set id	Sets the text log id string. For example if id is “dbg” and command ending is “\n”, you can append the message “qwerty” to the text log by sending “dbg qwerty\n” from microcontroller.
set label	Sets the text string to appear under the text log.
set text size	Sets the font size for the text.
log to file*	Lets you choose a file where you want to log the data that is received by this text log. For each received string, it will first append a prefix, then the string. Default prefix is “\n” (new line) so that each received string will be written to the file in a new line. You can change the prefix. Multiple items can log to the same file. *available only in RoboRemo Full version.

Accelerometer edit options

option	description
x config.	<p>Opens the configuration window for x axis. There you can set the id for x axis, the gain, the output type (float / int), the min. and max. values and enable / disable the value limiting option.</p> <p>Accelerometer will periodically send the id followed by a space character (or not), followed by the processed acceleration value for that axis. You can disable an axis by setting the id to empty string.</p> <p>The value is processed as follows: The data from the accelerometer sensor is multiplied by gain, then it is mapped from (-9.8, 9.8) to (min, max), then it is limited (or not) to [min, max].</p>
y config.	Opens the configuration window for y axis.
z config.	Opens the configuration window for z axis.
set repeat period	Sets the repeat (refresh) period in ms for accelerometer. Default is 20.
send space / don't send space	Select if you want the accelerometer to send space character after ids or not.

Text field edit options

option	description
set id	Sets the id for the text field. For example if id is "text1" and command ending is "\n", you can change the text to "abc" by sending "text1 abc\n" from microcontroller.
set text	Sets the text to be displayed inside text field
set text size	Sets the size of the displayed text
set text color	<p>Sets the color of the text. Format is ARGB_8888 (hex code)</p> <p>Examples: FF000000 = solid black, 80FF8000 = half transparent orange</p>
set bg. color	<p>Sets the background color for the text field. Format is ARGB_8888 (hex code)</p> <p>Examples: FFFFFFFF = solid white, 00FFFFFF = fully transparent</p>

Note: if you want an item to be displayed on top of text field, copy the item, then remove it, then paste it.

Plot edit options

option	description
set id	Sets the plot id string. For example if id is “x” and command ending is “\n”, you can add a new sample with value 100 to the plot by sending “x 100\n” from microcontroller.
set label	Sets the text string to appear under the plot. You can use the plot's last value inside the label. Also you can set the decimal count for the value that appears in the label. Examples for plot with last value 100: label “temp. = #*0.5-10 °C” will show “temp. = 40.0 °C” If you want it to show “temp. = 40 °C”, set the decimal count to 0. label “batt. #*1%” will show “batt. 100%”
set min	Sets the minimum value, default is 0.
set max	Sets the maximum value, default is 255.
set length	Sets the length (number of samples) to fit in one frame, default is 256. Maximum length is 10000.
set display mode	Sets the display mode for the plot. Scrolling (default) – plot scrolls from right to left, new data appears on the right. Oscilloscope – plot doesn't scroll. Data appears from left to right. When it reaches the end, it waits for trigger event, then starts again from the left.
set trigger	Sets the trigger threshold and edge (rising / falling / both / none). If none edge selected, then the trigger is off, so it will not wait for trigger event. For scrolling display mode there is no trigger (The trigger settings are ignored).
Don't repaint / repaint	Disables / enables the UI repaint when new sample is received.
log to file*	Lets you choose a file where you want to log the data that is received by this plot. For each received value, it will first append a prefix, then the value. Default prefix is “\n” (new line) so that each value will be written to the file in a new line. You can change the prefix. Multiple items can log to the same file. *available only in RoboRemo Full version.

Image edit options

option	description
set id	Sets the image id string. For example if id is “img” and image format is “auto” and command ending is “\n”, you can display an image by sending “img [len]\n” followed by the image data from microcontroller. [len] must be the size in Bytes of the image data that follows. The “auto” format accepts jpg, png and bmp data.
set label	Sets the text string to appear under the image.
keep / don't keep ratio	Sets if the image should keep the original w/h ratio, or should it fill the entire image item.
filter / don't filter	Sets if the image should be filtered or not when scaled.
set image format	<p>Sets the image format. Default format is “auto” and accepts jpg, png and bmp data. For raw image format, you can choose “RGB_888(w,h)”, “GRAY_8(w,h)”, “GRAY_4(w,h)”, “GRAY_2(w,h)” or “GRAY_1(w,h)”, where you have to replace w and h with numbers (width and height in pixels). The pixels are scanned from left to right, lines from top to bottom.</p> <p>RGB_888 uses 3 bytes per pixel (the levels for red, green and blue).</p> <p>GRAY_8 uses 1 byte per pixel (has 256 gray levels).</p> <p>GRAY_4 uses 4 bits per pixel (1 byte for 2 pixels) and has 16 gray levels.</p> <p>GRAY_2 uses 2 bits per pixel (1 byte for 4 pixels) and has 4 gray levels.</p> <p>GRAY_1 uses 1 bit per pixel (1 byte for 8 pixels) and has 2 gray levels (0 is full black and 1 is full white)</p>
import	Import an image from the SD card. The image is stored in the RoboRemo interface. It is recommended to use small images (up to 100KB).
set as background	All other items will be displayed on top of the image. The image item is locked from editing. You can unlock it from menu → interface → unlock edit all.

Touchpad edit options

option	description
set press id	Sets the id for press event. For example if press id is “pressed” and command ending is “\n”, when the user touches the touchpad, it will send “pressed [x] [y]\n”, where [x] and [y] are the point coordinates. For example: “pressed 316 244\n”
set drag id	Sets the id for drag event. Same as press id, but for drag events.
set release id	Sets the id for release event. Same as press id, but for release events.
set label	Sets the text string to appear under the touchpad.
set x min	Sets the x coordinate corresponding to the left border of the touchpad. Default is 0.
set x max	Sets the x coordinate corresponding to the right border of the touchpad. Default is 640.
set y min	Sets the y coordinate corresponding to the top border of the touchpad. Default is 0.
set y max	Sets the y coordinate corresponding to the bottom border of the touchpad. Default is 480.

Kbd connector edit options

option	description
set id	Sets the id for the kbd connector. (see examples below)
set label	Sets the text string to appear under the kbd connector..
set local / remote	Sets if the kbd connector should send the commands back to the interface, or should it send to the remote device. (see examples below)
send while typing / send at enter	Sets if the kbd connector should send each character separately or should it wait for enter key. (see examples below)

Kbd connector examples

Assuming command ending is “\n”

id	local / remote	send while typing / send at enter	user is typing	Result
“kbd”	remote	while typing	ab[enter]	Remote device receives: “kbd a\nkbd b\n kbd \n\n”
“kbd”	remote	at enter	ab[enter]	Remote device receives: “kbd ab\n”
“” (empty string)	remote	while typing	ab[enter]cde	Remote device receives: “ab\ncde”
“” (empty string)	remote	at enter	ab[enter]cde	Remote device receives: “ab\n” (and at next [enter] will receive “cde\n”)
“echo”	local	while typing	abc	Local interface receives “a\nb\nc\n”
“echo”	local	at enter	abc[enter]	Local interface receives “abc\n”
“” (empty string)	local	at enter	echo abc[enter]	Local interface receives “abc\n”

Heartbeat sender edit options

option	description
set id	Sets the string that will be sent periodically to the remote device.
set label	Sets the text string to appear under the item.
set repeat period	Sets the time interval (in ms) at which the heartbeat signal should be sent. Default is 1000.

Touch stopper edit options

Touch stopper does not have any special edit options.

Vibrator edit options

option	description
set id	Sets the vibrator id string. For example if id is “vib” and command ending is “\n”, you can make it vibrate for 200 ms by sending “vib 200\n” from microcontroller. If you send only “vib\n”, it will use the default duration that is set with “set duration”. Sending “vib 0\n” will make it stop.
set label	Sets the text string to appear under the item.
set duration	Sets the default duration (in ms) for the vibration. Default is 500.
set off timeout	Set the timeout after which vibrator will turn on if it does not receive off command. (Off command means “vib 0\n” for a vibrator with id “vib” on an interface with command ending “\n”). Default is empty string (timeout disabled).

File sender edit options

option	description
set id	<p>Sets the id string for the file sender. Default id is “file”.</p> <p>Here is an example for id = “file” and command ending = “\n”: You click the file sender item, and choose a file “a.txt” containing the data “qwerty”. The app will send to the microcontroller the string: “file 5 6\na.txtqwerty”.</p> <p>Where 5 is the length of the file name “a.txt” and 6 is the length of the file data “qwerty”. If you choose the option “don't send file name”, the string that will be sent in that case is: “file 6\nqwerty”.</p> <p>Keep in mind that the lengths (5 and 6 in this example) are the number of bytes in UTF-8 encoding, so that “abcd” has 4 Bytes, “файл” has 8 Bytes, “ação” has 6 Bytes, “țuică” has 7 Bytes, “フォン” has 9 Bytes.</p>
set label	Sets the text string to appear under the item.
send / don't send file name	Sets if the string that is to be sent should contain the file name information or not.
set block size	Set the block size in Bytes. File will be sent in blocks of this size.
set block delay	Set the delay (in ms) the app. will wait before sending the next block (to give time for the microcontroller to process the received data).
set GNB cmd.	<p>Set “get next block” command. If enabled, the app. will wait for this command before sending the next block. The microcontroller can send this when it finished processing the current block and it is ready for receiving the next one.</p> <p>Example: if file sender id = “file” and GNB cmd. = “gnb” and interface cmd. ending = “\n” (without “”), in order to receive the next block, the microcontroller must send “file gnb\n” (without “”).</p>
set cancel cmd.	<p>Set the command used to cancel the file sending process. If the microcontroller does not want to receive more data from the file, it can send this command, and the app. will stop sending (after it finishes sending the current block).</p> <p>Example: if file sender id = “file” and cancel cmd. = “cancel” and interface cmd. ending = “\n” (without “”), in order to cancel the file sending process, the microcontroller must send “file cancel\n” (without “”).</p>

File receiver edit options

option	description
set id	<p>Sets the id string for the file receiver. Default id is “file”.</p> <p>Here is an example for id = “file” and command ending = “\n”: Microcontroller sends the string “file 5 6\n”. App will interpret the next 11 Bytes as following: first 5 Bytes will be the file name (in UTF-8), next 6 Bytes will be the file data. After receiving the 11 Bytes, the app will generate the file using the file name and data provided, and will save it to the folder that was selected using “set rx folder”. (Default rx folder is the sdcard of the phone.) If the file with that name already exists, it will be overwritten.</p> <p>Example 2: Microcontroller sends the string “file 8\n”. App will interpret the next 8 Bytes as file data. After receiving the 8 Bytes, the app will generate the file using the data provided and the default file name, and will save it to the folder that was selected using “set rx folder”. Default file name is “file”. App will try to save the file under the name “file0”. If a file with this name already exists, app will use the name “file1”, “file2”, etc. until will find a name that is still not used. If default file name is “data.txt”, the generated file names will be “data0.txt”, “data1.txt”, “data2.txt”, etc.</p> <p>Keep in mind that the file name length is the number of bytes in UTF-8 encoding, so that “abcd” has 4 Bytes, “файл” has 8 Bytes, “açãõ” has 6 Bytes, “țuică” has 7 Bytes, “フォン” has 9 Bytes.</p>
set label	Sets the text string to appear under the item.
set rx folder	Sets the folder that will be used by the app to store the received files. Default is the sdcard of the phone.
set def. file name	Sets the default file name to be used a file name is not provided by the microcontroller. Default is “file” .

Speaker edit options

option	description
set id	Sets the speaker id string. For example if id is “spk” and command ending is “\n”, you can make it beep by sending “spk\n” from microcontroller. It will beep with the default frequency and duration. If you send “spk 4000 100\n”, it will beep with 4000 Hz for 100 ms.
set label	Sets the text string to appear under the item.
set volume	Sets the sound volume (in %) for the speaker. Default is 50.
set sample rate	Sets the sample rate (in Hz) for the speaker. Default is 44100.
set def. freq.	Sets the default beep frequency (in Hz). Default is 1000.
set def. duration	Sets the default beep duration (in ms). Default is 500.
set waveform	Sets the waveform (sine / square) for the beep. Default is square.

Printf() edit options

option	description
config.	<p>Configure the packet format. Examples:</p> <p><code>printf(“%02X\n”,s)</code> → will get the integer value from the item with id “s” and will send it as a 2-digit upper case hex value, followed by LF.</p> <p><code>printf(“%c%c%c”,0xFF,s.B1,s.B0)</code> → will send a 3-Byte packet: the first Byte will be 0xFF (255), the next 2 Bytes will be the Little-Endian integer value of the item with id “s”.</p> <p>The arguments after the “” in printf are the ids of the items, or constant values. Constant values can be integers (10, 255, 1024, ...), hex integers (0xFF, 0x3B), and float numbers (0.213, 10.7, ...).</p> <p>Adding .Bx after the id or constant will select the corresponding Byte. Examples: 0xAABBCC.B1 → 0xBB, 0x3F55AA00.B3 → 0x3F</p> <p>Currently the items that can be used with printf are: button, slider, accelerometer, heartbeat sender.</p> <p>The parser of the argument list ignores the characters ' ' (space) and '\n' (LF), so the ids of the items must not contain these characters, in order to work.</p> <p>Since button does not have a “set id” option, it has a pseudo-id, that is the common part of the press and release actions until the first space. Example: press action = “a 1”, release action = “a 0” → pseudo-id = “a”, value = “1/0”</p> <p>Since heartbeat sender has only id, its value will be always the constant 0.</p> <p>Adding more arguments than specifiers: This can be used, in order to trigger the printf without processing the value of that items. Example: Add 2 sliders with ids “a” and “b”, add heartbeat sender with id “h”, add printf with config. <code>printf(“%02X %02X\n”,a,b,h)</code>. The heartbeat sender will also trigger the printf, but the formatted packet will contain only the values from the sliders.</p>
set trigger	Set the id(s) (separated by ',') that should trigger the printf. (They also must be included in the argument list of the printf in config.)
set text	Change the text that appears on the printf item. Default is “printf()”.
set text size	Sets the font size for the text that appears on the printf item.

Menu options

option	description
connect / disconnect	Connects to a remote device / disconnects from a remote device. To connect to a Bluetooth remote device, it has to be paired. To pair with a Bluetooth device, open Android settings → Bluetooth → search for devices. Pin code for Bluetooth to serial adapter is usually “1234” or “0000”. Select port 1 (if port selection is set to manual from RFCOMM settings). To connect via Internet or WiFi, data connection or WiFi has to be activated from Android settings. Then in RoboRemo app choose menu → connect → Internet and select the IP and port. You can also use the domain name instead of IP.
edit ui / don't edit ui	Enters / exits the interface edit mode.
interface	Opens interface menu.
help	Shows help link.
about	Shows info about the app.
undo	Undo the last modification in the current interface.
settings	Open app. settings.

App. settings

autoconnect	Enables / disables the auto connect option. If enabled, the app will try to connect to the last connected device at next app start, if the user did not select disconnect before closing app. (App disconnects automatically when closed).
lock autorotate	Locks / unlocks the screen autorotate function for the app.
keep screen on	Used to disable the automatic screen lock.
char delay	<p>Slow microcontrollers need time to process each character received via serial port. You can set the char delay so that RoboRemo will wait after each character sent. It is strongly recommended to use char delay 0 when connecting via UDP since UDP packets are not guaranteed to arrive in the same order as they were send, and setting char delay > 0 makes separate packets for each character.</p> <p>This is global char delay. Starting with version 2.0.0, each interface has its own char delay. If the interface char delay is set to empty string (default), it will use this global char delay.</p>
When connected: disable back key disable menu key	Options to disable the back and menu hardware keys when the app. is connected to a remote device, so that the used doesn't accidentally press them.
Back key function in editor (exit editor / undo)	User can choose the function of the back key to act as undo instead of exiting the editor.
RFCOMM settings:	Change RFCOMM settings. RFCOMM is the name for the Bluetooth Serial protocol. Default settings are: automatic port selection, without encryption.
port (channel) selection	manual = will ask each time fixed(1) = will use port 1 auto = will select automatically
UUID (for automatic)	Used only if port selection = auto. Ignored otherwise. default = use the well known UUID for Bluetooth to Serial modules, which is 00001101-0000-1000-8000-00805F9B34FB first = use first UUID of the remote device, from the local cache
encrypted	Use / don't use encryption.

Note: Some devices may fail to connect with certain settings. In this case it is recommended to try different settings.

Interface menu options

option	description
select	Select other interface. RoboRemo has 8 interfaces, numbered 0 to 7.
rename	Rename the current interface.
set connect action	Set a command string to be send to the remote devices after connecting.
set command ending	Change the command ending for the current interface. The default is “\n”. Command ending can also be set to empty string, but it is not recommended.
import	Import an interface from file. Interfaces are stored in the roboremo folder on the SD card. The folder is created automatically at first app start.
export	Export the current interface to a file. Interfaces are stored in the roboremo folder on the SD card. The folder is created automatically at first app start.
unlock edit all	Unlock all items that were locked with lock edit.
set bg. color	Set the background color of the interface. Format is ARGB_8888 (hex code). Examples: FFFFFFFF = white, FFD891EF = Bright Lilac, FF92A1CF = Ceil.
clear	Clear the current interface.

New features and fixed bugs

App. version	New features	Fixed bugs
1.2	<ul style="list-style-type: none"> - accelerometer support - Internet / WiFi connectivity - option for sliders to return to center when released. 	N/A
1.3	<ul style="list-style-type: none"> - option to change command ending - string inputs from user are interpreted for escape characters ('\t', '\b', '\n', '\r', '\f'). For '\ character, enter '\\. 	<ul style="list-style-type: none"> - fixed interface import / export for interfaces with non-ASCII characters. - fixed text wrapping and auto scrolling when resizing text logs.
1.4	<ul style="list-style-type: none"> - local action “interface n” same as “iface n” where n is the interface number (0 to 7) - auto connect option - local actions for connecting and disconnecting by pressing buttons. - removed ”exit” from menu (app can be closed by pressing back key). 	<ul style="list-style-type: none"> - fixed the bug with “\r\n” string in interfaces. - fixed a bug where app crashed if n was not an integer in “iface n” or “interface n” local action.
1.5	<ul style="list-style-type: none"> - option to use the slider or level indicator value inside their labels - made slider with "send when moved" send only when the value changes - option to change on and off commands for leds - removed the dialog showing error when sending a non-integer value to a level indicator - undo option in interface editor. 	<ul style="list-style-type: none"> - fixed a bug in processing received commands. Now if there are more items with same id, all of them will respond to it.
1.6	<ul style="list-style-type: none"> - added option to set repeat period for sliders - added plots and text fields - improved touch processing algorithm so that it will send touch events only to interactive items (now you can have buttons / sliders hidden under leds, plots, etc) - added menu option to change RFCOMM settings 	<ul style="list-style-type: none"> - fixed a bug in sliders where slider with auto return enabled did not send data when it was the same value as before return.
1.7	<ul style="list-style-type: none"> - added option to set text size for text fields and text logs - added option to set id for text field so that user can change the text remotely 	<ul style="list-style-type: none"> - when command ending is empty string, do not append “\n” at the end in text logs - fixed a bug where autoconnect

	<ul style="list-style-type: none"> - added image item in RoboRemo Full version - reduced TCP connect timeout from 90000ms to 5000ms - improved menus (now they are scrollable) - added touchpad item, kbd connector item - added option to keep screen on 	<p>dialog could not be cancelled after screen orientation change</p> <ul style="list-style-type: none"> - interface → clear now also resets the command ending and connect action
1.8	<ul style="list-style-type: none"> - added item option to lock edit and interface option to unlock edit all - added option to set return value for slider - added option to set led on/off timeout - added option to import image from SD card and store it in RoboRemo interface - added items: heartbeat sender, touch stopper - changed default RFCOMM settings to automatic port selection - added option to start a TCP server - can connect to domain name - added option to use plot's last value inside the label 	N/A
1.9	<ul style="list-style-type: none"> - added image option to set as background - added USB connectivity - added “don't repaint” option for plots - hw menu button also opens menu 	<ul style="list-style-type: none"> - fixed bug with text size on text log and text field items - fixed bug with pasting an item after lock edit - fixed small menu bug - fixed touchpad bug where it was sending data even if id was empty string
1.9.1	<ul style="list-style-type: none"> - added UDP connectivity - added “log to file” option (in RoboRemo Full version) for text log and plot. - added vibrator item in RoboRemo Full version - added option for accelerometer to set gain, output type (float / int), min. and max. values and enable / disable the value limiting (replaced “set x/y/z id” with “x/y/z config.”). - added local action “send” - now local action can contain multiple actions, separated by command ending - added option in slider, level indicator and plot items label to set the decimal count. - slider and level indicator items now accept setting min. value greater than max. value and 	<ul style="list-style-type: none"> - fixed bug in USB connection where data sent too fast from microcontroller was not received correctly by the app. - fixed USB autoconnect and connect by local action. - fixed text log bug where it was stalling on some devices after receiving about 10000 characters. - clearing an interface now clears the interface name too. - fixed float numbers in exported interface files. - fixed a bug where app was crashing when trying to connect via USB on devices that don't have USB API. - on slider item, now selecting

	<p>behave as expected.</p> <ul style="list-style-type: none"> - improved some user interface (in RFCOMM settings, image import, plot set trigger). - now the options that are available only in RoboRemo full version, are shown (disabled) in RoboRemoFree too. - added “100% FREE” mark on the icon of RoboRemoFree 	<p>autoreturn or changing the return value updates the position of the handle.</p> <ul style="list-style-type: none"> - fixed default aspect ratio of heartbeat sender item.
1.9.2	<ul style="list-style-type: none"> - added hints in UiEditor - added links in menu → about - added option to set the line width for plots - added option to set text size for plot, slider and level indicator labels, and buttons. - added file sender and file receiver items in RoboRemo Full version 	<ul style="list-style-type: none"> - Increased maximum size of UDP rx packet from 8192 to 65507 Bytes
1.9.3	<ul style="list-style-type: none"> - file sender and receiver glowing when active - removed the delay before opening streams for TCP - added options to set block size and block delay for file sender 	<ul style="list-style-type: none"> - fixed some bugs in file sender and file receiver, now they should work OK with files up to 2GB
1.9.4	<ul style="list-style-type: none"> - easier setup for local action connect - added local actions sendnce, sendhex - added speaker item - added option to set button images - added option to repeat only the press action at button repeat - added option for slider, level indicator and plot to work with float numbers 	<ul style="list-style-type: none"> - fixed bug with '.' in command ending - fixed bug where slider was not displaying correctly in Android 6 - fixed bug with negative values in slider
2.0.0	<ul style="list-style-type: none"> - moved all app. settings to menu → settings - added option to disable back and / or menu hardware keys when connected - added printf item that gets data from other items and user can configure the format of the packet that will be sent - Bluetooth made optional, now users should be able to install the app. on devices that don't have Bluetooth hardware. - added option to set the char delay for each interface separately (there is also a global char delay in menu → settings, that will be used by default, if the interface char delay is set to 	<ul style="list-style-type: none"> - fixed touchpad bug, now it should work with negative values too. - fixed touchpad bug where it was sending multiple times the same value. - undo option (if undo is available) now appears in menu only in edit mode. - fixed bug where items did not send data in Android 7 (not tested) - fixed bug where some items (like heartbeat sender, accelerometer) were sending data before the connect

	<p>empty string).</p> <ul style="list-style-type: none"> - added file sender options “set GNB cmd.” (get next block) and “set cancel cmd.” (cancel sending). - added option (in menu → settings) to set the editor back key function to undo instead of exiting editor. - added option to change the text and background color for text field. - added option to change the background color of the interfaces. - added some hints and some syntax checking in the edit options of the items. - changed the default (disabled) value from “0” to empty string for some options, to be more intuitive. Old version interfaces are updated automatically, but new interfaces may not work correctly in old app. versions, so make sure you have the latest app. version. 	<p>action.</p>
2.0.1	<ul style="list-style-type: none"> - added USB support for Microchip CDC devices (VID 0x04D8) - changed help link from Dialog to TextDisplay 	<ul style="list-style-type: none"> - changes made with “set connect action” and “interface rename” now will persist in case of crash. - fixed bug where app was crashing when setting LED off timeout to empty string
2.0.2	<ul style="list-style-type: none"> - added Bluetooth Low Energy (BLE) support - improved USB write speed - improved UDP write speed 	N/A
2.1.0	<ul style="list-style-type: none"> - changed USB library - added support for CH340/HL-340 USB to Serial chip found in many Arduino clones. - added more high-speed baud rates for the chips that support them. 	<ul style="list-style-type: none"> - added “connection lost” detection for FTDI chips (when removing from USB while connected).
2.1.1	N/A	<ul style="list-style-type: none"> - fixed some bugs in the USB library
2.1.2	N/A	<ul style="list-style-type: none"> - fixed bug where clearing an interface was not clearing its char delay. - fixed BLE bug where interface's connect action was causing a connection drop.