Universal Robot Control (RX/TX)

Introduction

This repository includes transmitter (TX) and receiver (RX) arduino files for a remote controlled robot of any form.

The transmitter is a custom build to have full control over input mixing, skewing and do all sorts of dependencies.
It depends on a custom PCB, or like a thousand wires.
The receiver end can be a robot or whatever.

This relies on OrangeRx products by hobbyking for Tx/Rx and uses the (legacy) DSM2/DSMX[†] protocols for a reliable, digital communication.
with a "range of up to 3km (line of sight)".
I'm sure though this works with any RC transmitter that uses iBUs / ppm.

> Based on James Brutons RemoteDSM:
>
> https://github.com/XRobots/RemoteDSM ↗
>
> https://youtu.be/5BfRg9CUMYI ↗

[†] "*The DSM in DSMX stands for Digital Spectrum Modulation and is what creates pure digital control providing an impenetrable radio link immune to all types of interference.*" (Soruce ↗)

Transmission speed

Depending if you need 10 or 20 channels you can choose to use a second transmitter.
The second transmitter leads to more fluctation and/or longer duration of the main `loop()` though. If you need snappy response times, reconsider using a second transmitter.

Num of Transmitters	Range (ms)	Average (ms)	Median (ms)
1	30-69	31	32
2	28-70	45	33

Hardware

TX

- Arduino Mega ↗
- One thousand switches and buttons
- One thousand linear potentiometers
- Two three-axis joysticks (ali express article 1↗ or 2↗) with a pushbutton

- A custom PCB (files here: gerber.zip | kicad)
- A laser cut case (files here: ai | dxf | svg)
- One (or two) OrangeRX DSMX DSM2 transmitter ↗
- *..extensive list has to follow*

RX

- Arduino Leonardo (or UNO) ↗
- Arduino Motor Shield Rev3 ↗
- One (or two) OrangeRx RR12RDTS Receiver ↗
- OrangeRx Program box ↗
- *..extensive list has to follow*

OrangeRx setup

OrangeRx Receiver re-programming

As the Windows software "rx_secure_configurator ↗" for the receiver ↗ to set some settings on the receiver is **very** buggy, buy the OrangeRx Program box ↗.

1. Connect the jumper to power the receiver from the program box to pin 1 and 2.
2. Set the "out type" from default "*SBus*" to "*iBUS*"
3. Set the *FailSafe" to send "1500" on all channels when losing connection; to be able to detect signal loss on the receiver

OrangeRx Binding Process

1. Power off remote.
2. Power on robot receiver. Its LED flashes fast yellow to indicate that there is no connection.
3. Press "Bind" button on transmitter.
4. Power on remote (!); keep "Bind" button pressed.
5. remote beeps three times; robot receiver stops flashing and has now a continous light. congrats.
6. From here on out, connection is established.

> 📝 NOTE: Red LED on remote (on the transmitter) does not turn on if no signal is sent to the PPM pin.
>
> 📝 NOTE: Always power on robot receiver first, then remote.
>
> 📝 NOTE: If turned off remote while robot receiver is on, restart robot receiver as well to reset connection.
> Turning off robot receiver while remote is still on, restarting robot receiver again auto connects again.