

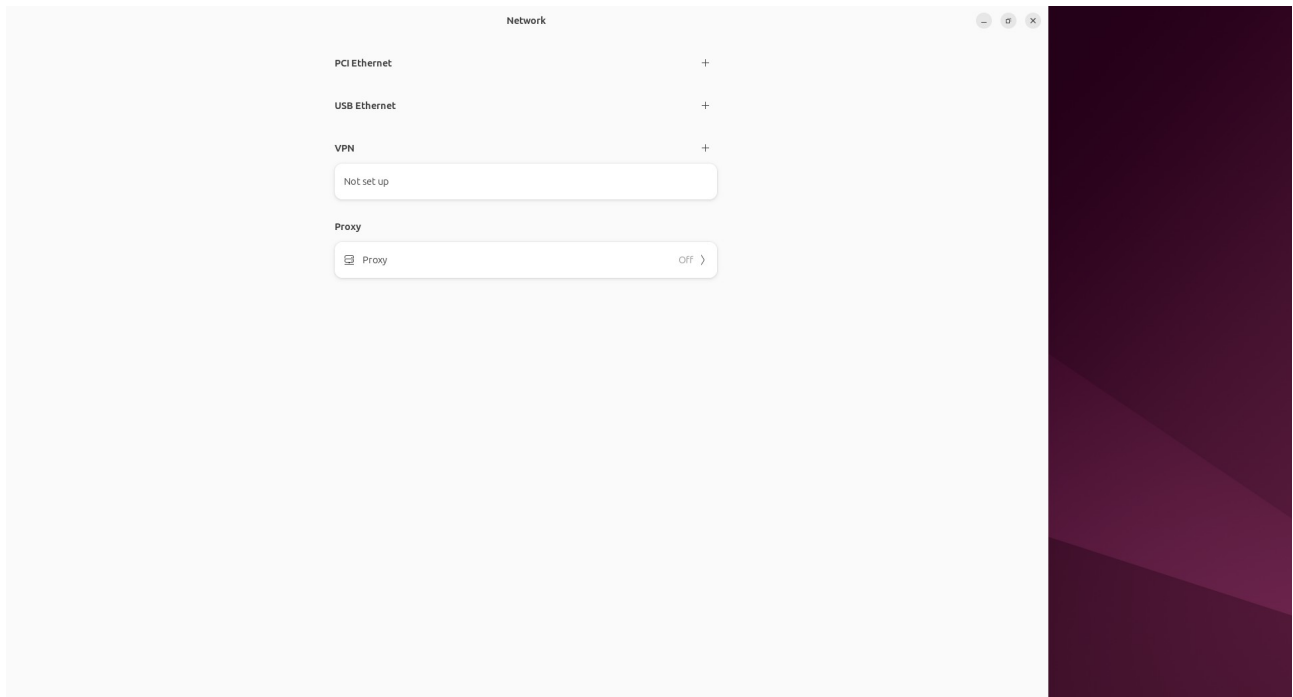
Github: Shahidhasib586

Linkedin: Shahid Ahamed Hasib

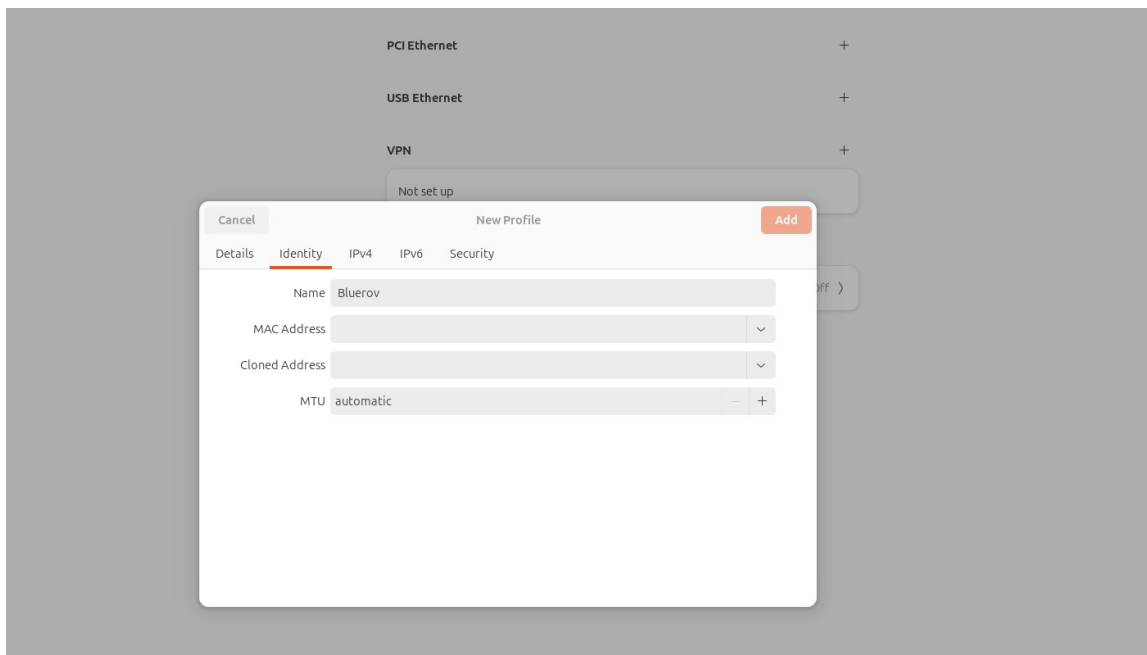
First clone the foloowing github repository to your ros2 workspace:

[https://github.com/ShahidHasib586/bluerov\\_ws](https://github.com/ShahidHasib586/bluerov_ws)

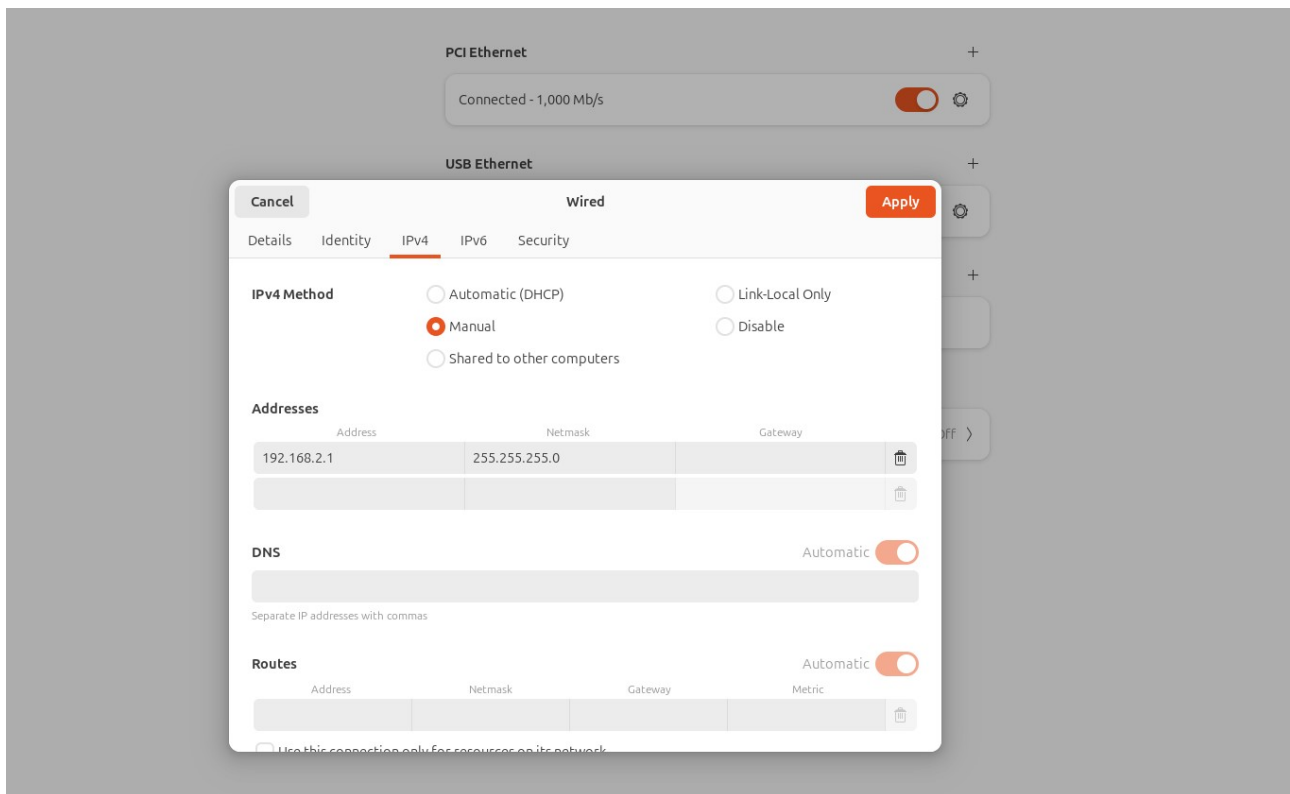
Connecting your BlueROV:



- Go to the network connection:
- Add PIC Ethernet/ add connection



- Go to identity and give a name “BlueROV”
- Go to “IPV4”
- Select Manual

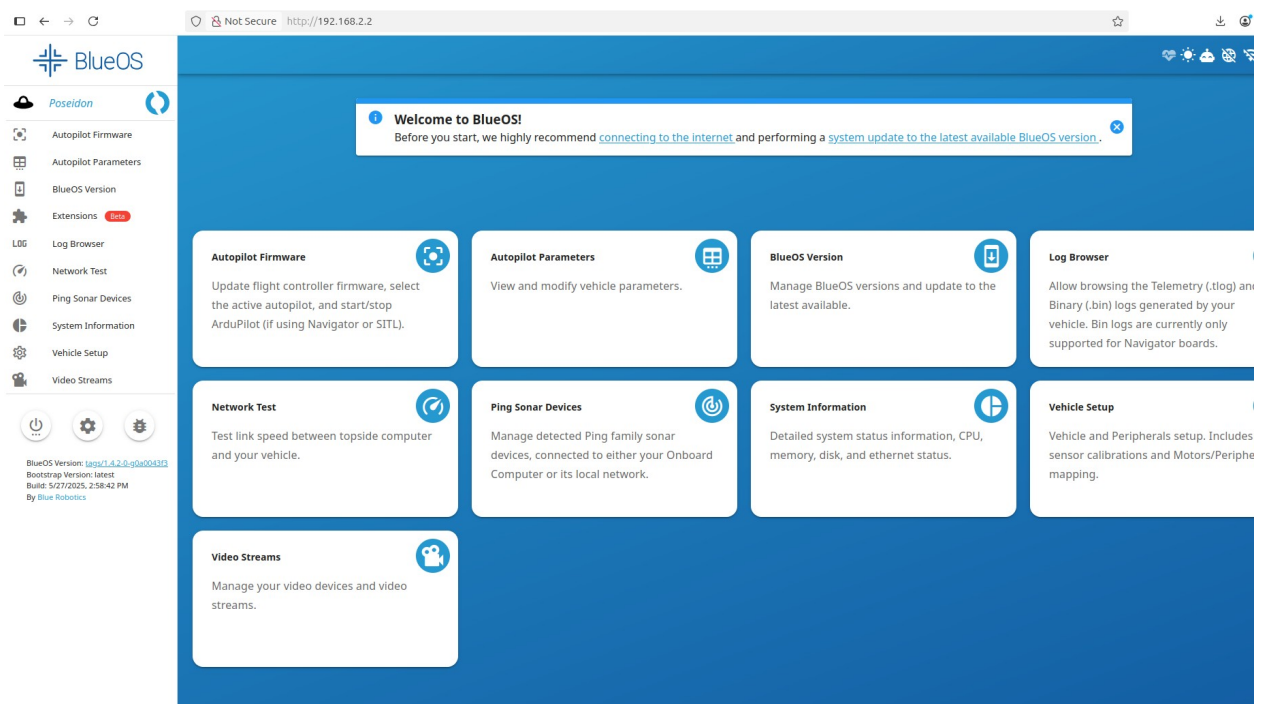


- In the address type: 192.168.2.1
- In netmask type: 255.255.255.0

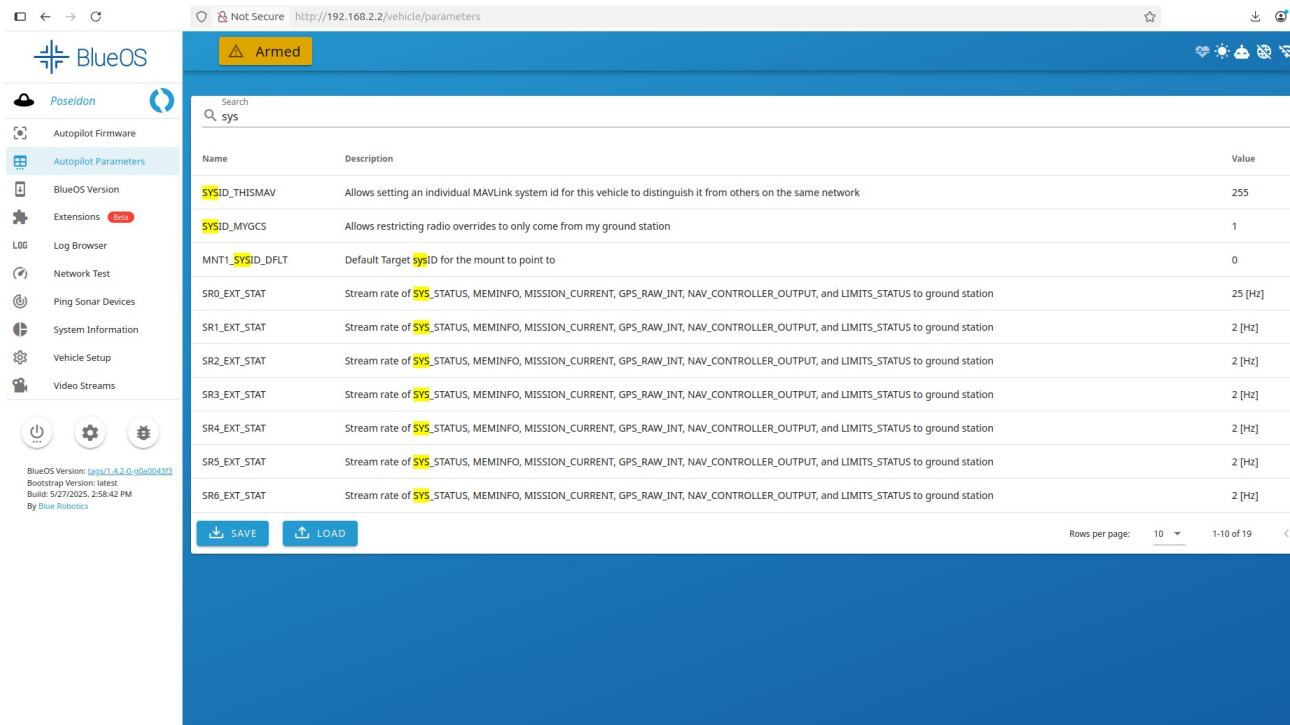
Now go to BlueOS:

Open a browser and type:

192.168.2.2



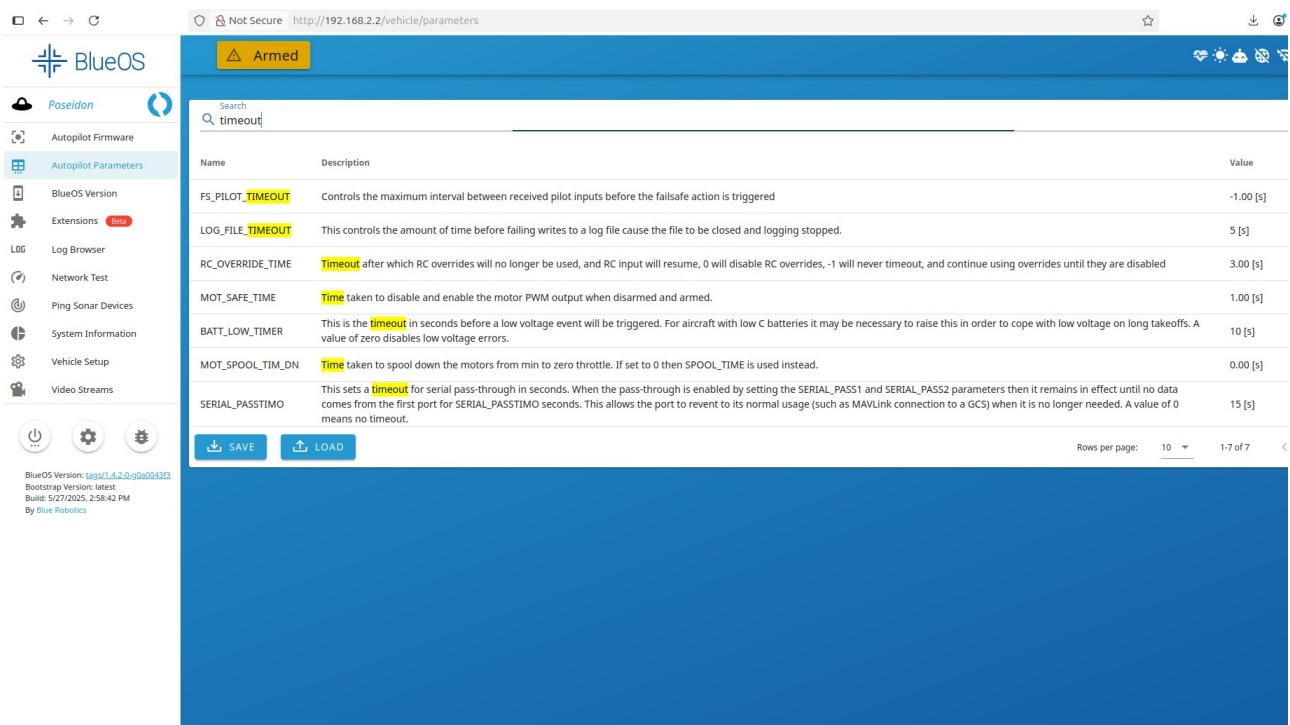
Go to Autopilot Parameters:  
and search “SYSID\_MYCGS” and set the value to “1”



The screenshot shows the BlueOS Autopilot Parameters interface. The search bar contains 'sys'. The table lists parameters with their names, descriptions, and values.

Name	Description	Value
SYSID_THISMAV	Allows setting an individual MAVLink system id for this vehicle to distinguish it from others on the same network	255
SYSID_MYCGS	Allows restricting radio overrides to only come from my ground station	1
MNT1_SYSID_DFLT	Default Target sysID for the mount to point to	0
SR0_EXT_STAT	Stream rate of SYS.STATUS, MEMINFO, MISSION_CURRENT, GPS_RAW_INT, NAV_CONTROLLER_OUTPUT, and LIMITS_STATUS to ground station	25 [Hz]
SR1_EXT_STAT	Stream rate of SYS.STATUS, MEMINFO, MISSION_CURRENT, GPS_RAW_INT, NAV_CONTROLLER_OUTPUT, and LIMITS_STATUS to ground station	2 [Hz]
SR2_EXT_STAT	Stream rate of SYS.STATUS, MEMINFO, MISSION_CURRENT, GPS_RAW_INT, NAV_CONTROLLER_OUTPUT, and LIMITS_STATUS to ground station	2 [Hz]
SR3_EXT_STAT	Stream rate of SYS.STATUS, MEMINFO, MISSION_CURRENT, GPS_RAW_INT, NAV_CONTROLLER_OUTPUT, and LIMITS_STATUS to ground station	2 [Hz]
SR4_EXT_STAT	Stream rate of SYS.STATUS, MEMINFO, MISSION_CURRENT, GPS_RAW_INT, NAV_CONTROLLER_OUTPUT, and LIMITS_STATUS to ground station	2 [Hz]
SR5_EXT_STAT	Stream rate of SYS.STATUS, MEMINFO, MISSION_CURRENT, GPS_RAW_INT, NAV_CONTROLLER_OUTPUT, and LIMITS_STATUS to ground station	2 [Hz]
SR6_EXT_STAT	Stream rate of SYS.STATUS, MEMINFO, MISSION_CURRENT, GPS_RAW_INT, NAV_CONTROLLER_OUTPUT, and LIMITS_STATUS to ground station	2 [Hz]

Now search “timeout” and set “FS\_PILOT\_TIMEOUT” to “-1” force it to this value.



The screenshot shows the BlueOS Autopilot Parameters interface. The search bar contains 'timeout'. The table lists parameters with their names, descriptions, and values.

Name	Description	Value
FS_PILOT_TIMEOUT	Controls the maximum interval between received pilot inputs before the failsafe action is triggered	-1.00 [s]
LOG_FILE_TIMEOUT	This controls the amount of time before failing writes to a log file cause the file to be closed and logging stopped.	5 [s]
RC_OVERRIDE_TIME	Timeout after which RC overrides will no longer be used, and RC input will resume, 0 will disable RC overrides, -1 will never timeout, and continue using overrides until they are disabled	3.00 [s]
MOT_SAFE_TIME	Time taken to disable and enable the motor PWM output when disarmed and armed.	1.00 [s]
BATT_LOW_TIMER	This is the timeout in seconds before a low voltage event will be triggered. For aircraft with low C batteries it may be necessary to raise this in order to cope with low voltage on long takeoffs. A value of zero disables low voltage errors.	10 [s]
MOT_SPOOL_TIM_DN	Time taken to spool down the motors from min to zero throttle. If set to 0 then SPOOL_TIME is used instead.	0.00 [s]
SERIAL_PASSTIMO	This sets a timeout for serial pass-through in seconds. When the pass-through is enabled by setting the SERIAL_PASS1 and SERIAL_PASS2 parameters then it remains in effect until no data comes from the first port for SERIAL_PASSTIMO seconds. This allows the port to revert to its normal usage (such as MAVLink connection to a GCS) when it is no longer needed. A value of 0 means no timeout.	15 [s]

On your terminal (I use terminator) open 4 window and run the following command one by one

- `ros2 launch autonomous_rov run_mavros.launch`
- `ros2 launch autonomous_rov run_listener.launch`
- `ros2 launch autonomous_rov run_gamepad.launch`
- `ros2 run autonomous_rov video`

Bingo!!! you can now press start button on your gamepad and arm the vehicle and control it.