

# ServoMaster: Frequently Asked Questions

## Table of contents

1 Questions.....	2
1.1 1. General.....	2
1.2 2. Serial.....	3
1.3 3. USB.....	4
1.4 4. Phidgets.....	4

## Questions

---

### 1. General

---

#### 1.1. What hardware is supported?

---

Two generic classes of controllers are supported: serial and USB. Specific list is:

**Serial:**

FerretTronics [FT639](#);

**USB:**

- [PhidgetServo](#);
- PhidgetUniServo (as a subset of [PhidgetServo](#));
- [PhidgetAdvancedServo](#) (prototype).

However, a lot of attention was paid to making the code as abstract and reusable as possible, therefore, writing a driver for a new device should take just a few days, if not hours.

#### 1.2. Are there more hardware controller drivers coming?

---

Actually, quite a few. The [Pololu controller family](#) driver is already usable, and [Parallax controller family](#) is a few days away, but it's been extremely difficult to find those few days to finish it.

Then, there's one in the queue ([1-Wire® based servo controller](#) which is pretty close to my [ideal servo controller](#)).

#### 1.3. How do you select the hardware controllers to implement the drivers for?

---

First one I bought. The rest were donated. If you want the driver for your controller, consider donating one [like others did](#) - by now, the implementation is pretty straightforward, it takes a day or two of straight coding.

#### 1.4. How stable is ServoMaster?

---

Rock solid. It's been used to run inside the [DIY Zoning Project](#) code for several years, 24/7/365, non-stop. Never failed.

## 1.5. What is this "silent mode" thing?

---

If you are familiar with the servos, you probably know that they can make a sound that can make you jump, especially caught off guard. The application this project is serving is intended to have the servos operate next to people whose comfort level is paramount, so they must not be scared to death periodically when the servos come to life.

Also, as servos wear down, they sometimes start "growling".

In order to reduce noise related to servo operation, it is possible to shut off the pulse to the servo - it enters the "sleep mode" shortly thereafter. While sleeping, it emits no noise.

As a side effect, this reduces the energy consumption (the sleeping servo goes to a powerdown and consumes about 8mA - thanks again, Chester) and servo wear. Since the servos may be working 24x7, this may be important as well.

## 1.6. The Servomaster debug console doesn't fit into my screen!

---

It is a [known issue](#)

Since the advent of 16-servo controllers it doesn't fit into my 1600x1200 screen, either.

This is unfortunate, and there are two things that can possibly be done:

- Decrease the size of the console elements to make it all smaller. This is quite painful to implement (basically I'll have to change the default font on every single element), and will make the font unreadable, which is unacceptable for me. Don't ask for it, I won't do it.
- Make the console frame scrollable. This has already been done, and while it does look ugly, remember that this is just a debug console we're talking about, and I'd hate to put efforts into it if I can do something more useful instead.

So, the solution is to just change the frame size - and the contents will become scrollable.

## 2. Serial

---

### 2.1. Why do I keep getting `noClassDefFoundError: javax/comm/UnsupportedCommOperationException`?

---

Because you didn't install the [JavaCOMM](#) properly. If you keep getting problems even though you've followed the instructions to the letter, try to go with the JRE instead of JDK - copy the `comm.jar` to `${JRE_HOME}/lib/ext` directory.

As a rule of thumb, the newer the version of the JDK/JRE, the less problems you will have.

### 3. USB

---

#### 3.1. ServoMaster doesn't detect my USB controller, why?

---

This seems to be a peculiarity related to the way [javax.usb](#) security is implemented.

The following output from `servomaster_console` would be a good indication that you're having this problem:

```
Unable to initialize controller, cause:
java.io.IOException: No compatible devices found. Make sure you have
/proc/bus/usb read/write permissions.
    at
org.freehold.servomaster.device.impl.usb.AbstractUsbServoController.findUSB(AbstractUsbS
    at
org.freehold.servomaster.device.impl.usb.AbstractUsbServoController.doInit(AbstractUsbS
    at
org.freehold.servomaster.device.model.AbstractServoController.init(AbstractServoControl
    at org.freehold.servomaster.view.Console.run(Console.java:166)
    at org.freehold.servomaster.view.Console.main(Console.java:134)
```

Quick fix: `chmod -R a+w /proc/bus/usb` - this will allow you to test whether the controller is connected. Alternative way is to use `usbview` utility, it should show the device unconditionally (if it is present, that is).

#### Warning:

It is not recommended to run ServoMaster code as root, for security reasons. Nothing in particular, but I'm just paranoid.

A more thorough solution involves properly configuring [hotplug](#).

#### Note:

Don't forget to check [kernel modules](#) if your USB controller is a Phidget.

### 4. Phidgets

---

#### 4.1. USB permissions are OK, but I still can't see the Phidget!

---

The symptom is usually a complaint about the fact that the USB interface is already claimed (long line split):

```
Unable to initialize controller, cause:
java.io.IOException: Can't claim interface - already claimed. \
Make sure no other applications or modules (hid.o or phidgetservo.o in
particular) use this device
    at
org.freehold.servomaster.device.impl.usb.AbstractUsbServoController.doInit(AbstractUsbS
    at
org.freehold.servomaster.device.model.AbstractServoController.init(AbstractServoControl
    at org.freehold.servomaster.view.Console.run(Console.java:166)
    at org.freehold.servomaster.view.Console.main(Console.java:134)
```

Following is the message you see in the system log if this is happening (when you attach the controller):

```
Mar  3 19:39:08 shadow kernel: ohci_hcd 0000:00:02.0: wakeup
Mar  3 19:39:08 shadow kernel: usb 2-3: new low speed USB device using
ohci_hcd and address 2
Mar  3 19:39:09 shadow kernel: phidgetservo 2-3:1.0: USB 4-Motor
PhidgetServo v3.0 attached
Mar  3 19:39:09 shadow kernel: usbcore: registered new driver phidgetservo
```

Quick fix, depending on the kernel version, is: either `rmmod hid` or `rmmod phidgetservo`. It'll keep bothering you, though, as you unplug and re-plug the device.

A more thorough solution would be to add the culprit to [hotplug](#) blacklist, but it may have an unfortunate side effect of disabling *all* HID devices in older kernels. Recently, hotplug has become smart enough to be able to specifically ban the `phidgetservo` module - look either into `/etc/hotplug/blacklist`, or `/etc/modprobe.d/blacklist-compat` - the rest is self-explanatory.

This is how your system log should look like if you're successfully disabled the `phidgetservo` module:

```
Mar  3 19:59:41 shadow kernel: ohci_hcd 0000:00:02.0: wakeup
Mar  3 19:59:41 shadow kernel: usb 2-3: new low speed USB device using
ohci_hcd and address 4
```

## 4.2. How do I get a hold of AdvancedServo?

You can't. The device that I have is a prototype generously [donated](#) by Chester Fitchett, a.k.a. Mr. [Phidget](#). This magnificent concept evolved into something even better that will soon be available from Phidgets, Inc. - keep checking in.