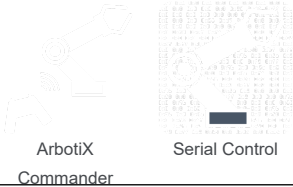


PhantomX Hexapod Phoenix Code

Description:

This guide will help you set up the Phoenix Code on the PhantomX Hexapod. This guide will work for both MK-I and MK-II models, as well as AX-12A and AX-18A models. This guide assumes that you have completed the [PhantomX Hexapod Getting Started Guide](#). This includes setting up the [ArbotiX RoboController](#) and using the [original firmware](#) for the hexapod.



Using the Phoenix Code on the Hexapod Crawler



The Phoenix Code/ Firmware is a community driven project and not officially supported by Trossen Robotics. This guide is offered as a starting point for those interested in the Phoenix Code. Trossen Robotics does not offer any software or Hardware support for the Phoenix Code / Firmware. See the [links](#) section for support.

What is the Phoenix Code?

The Phoenix code is firmware for the PhantomX Hexapod. This firmware gives the PhantomX Hexapod a life-like walking gait as well as offers a variety of features:

FEATURES

- Phoenix Engine Walking Gait
- Walking Gait Options (Double Height / Double Travel)
- Translate Mode
- Rotate Mode
- Single Leg Mode
- Z Axis Control (raise body up + down)
- Speed Control
- GP Mode (Work In Progress)

PHOENIX CODE VS NUKE

The Phoenix Code does have downsides when compared to the stock **NUKE**(**N**early **U**niversal **K**inematics **E**ngine) code that is offered as an unsupported demo for the hexapod. It is a version of the original NUKÉ code written by **Vandium Labs** and is no longer supported by Trossen Robotics.

- **Complexity:** The Phoenix Code is more complex than the stock NUKÉ code. It may be more difficult to customize and integrate Phoenix code into your own project.
- **Program Size:** The Phoenix Code for the PhantomX Hexapod MK-II is **23,116 Bytes** for the MK-II and **23,982 Bytes** for the MK-I. The stock NUKÉ code is **17,414 bytes**.
- **Processing Power:** The Phoenix Code requires more processing power than the stock NUKÉ code.
- **Stability:** The Phoenix code is more prone to bugs than the stock NUKÉ code.
- **Support:** The Phoenix Code is not officially supported by Trossen Robotics.

Generally speaking, the Phoenix code is intended for advanced users interested in advanced kinematic engines. Those who want to get started quickly with the PhantomX Hexapod development platform are encouraged to use the **original NUKÉ firmware** for the hexapod.

History

The Phoenix code was originally written by Jeroen Janssen (Xan) for the **Lynxmotion Phoenix** - the Lynxmotion Phoenix was based off of the original Phoenix Robot by KÅre Halvor. Phoenix was originally written in Basic to run on the Basic Atom Pro 28. Much of the original software was based off of Zenta's **Excel Spreadsheet work (PEP)**.

Later, Kurt Eckhardt (**KurtE**) then began to port the code to C/C++ and the Arduino environment with the help of Xan and Zenta. Kurt's port of the Phoenix code currently supports a variety of crawlers including the PhantomX Hexapods.

Hardware Configuration

The Phoenix Code will work with any stock PhantomX Hexapod kit purchased from Trossen Robotics.

The Phoenix code is capable of sending audio feedback to notify users about mode and speed changes. This audio feedback is extremely useful for navigating the controls of the Phoenix. To use this ability, the user must install a Piezo Buzzer on Digital Port **1** of the ArbotiX RoboController. We recommend the **RobotGeek Buzzer** and a **3-pin sensor cable** that will plug directly into ArbotiX's **D1** port. Other piezo buzzers will also work.

This guide does not support voltage monitoring or accelerometer readings.

Installation

The latest version of the Phoenix Code for the PhantomX Hexapod is included in the **Arduino_Phoenix_Parts** repository on KurtE's Github account(**Direct Zip Download**). Download the zip file and move the following folders into your **libraries** folder.

- Phoenix
- Phoenix_Driver_AX12
- Phoenix_Input_Commander

You can move any of the other folders to the /libraries folder, but they will not be used for the PhantomX Hexapod.

You will not need all of the files on the **Phoenix** folders. You will need these files:

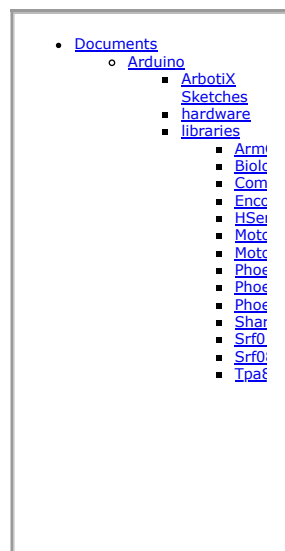
- Phoenix_Code.h
- Phoenix.h
- readme.txt

As well as at least one of the following folders, based on your hardware configuration:

- PhantomX_Commander_AX12
- PhantomX_Commander_AX12_Stock
- PhantomX_V2_Commander_AX12_Stock
- PhantomX_DIYXbee_AX12

However there is no harm in leaving all of the folders in the **Phoenix** folder intact.

Your Arduino folder structure should look like the following: (This structure assumes that you have installed the stock ArbotiX Libraries and hardware files from the **ArbotiX Getting Started** guide.)



If you are using the **MK-I Hexapod**, open

```
File -> Examples -> Phoenix -> PhantomX_Commander_AX12_Stock
```

If you are using the **MK-II Hexapod**, open

```
File -> Examples -> Phoenix -> PhantomX_V2_Commander_AX12_Stock
```

CODE MODIFICATIONS

Communication Timeout

The Phoenix Firmware only activates the crawler when data from the ArbotiX Commander is present. If communication is lost with the commander, the crawler will de-activate. The time before the crawler deactivates (ARBOTIX_TO) is 1 second, and we have found that the robot may shut down erratically with this setting. We recommend that you change this to do this use a text editor like **Sublime Text** to open:

```
Arduino -> libraries -> Phoenix_Input_Commander -> Phoenix_Input_Commander.h
```

and change line 89 from

```
#define ARBOTIX_TO 1000 // if we don't get a valid message in this number of mills turn off
```

to

```
#define ARBOTIX_TO 2000 // if we don't get a valid message in this number of mills turn off
```

If you still have problems connecting to the crawler, you may try increasing this number. The only downside of increasing this value is that the crawler will take longer to de-activate. The controller is turned off.

Even when in it's de-activated state, the crawler still uses power. Make sure to run your crawler with the included battery monitor, and to use the power switch to your crawler when not in use.

BT Commander

If you are running the code with the **BT Commander** App, you will need to comment out the timeout code entirely. This is also done in Phoenix_Input_Commander.h. To do this use **Sublime Text** to open:

```
Arduino -> libraries -> Phoenix_Input_Commander -> Phoenix_Input_Commander.h
```

and change line 482 and 483 from

```
if ((millis() - g_ulLastMsgTime) > ARBOTIX_TO)
    CommanderTurnRobotOff();
```

to

```
//if ((millis() - g_ulLastMsgTime) > ARBOTIX_TO)
//    CommanderTurnRobotOff();
```

Your crawler will no longer timeout on disconnect. When you have disconnected, make sure to power down your crawler with the power switch when not in use.

AX-18

If you use the stock AX-12 code on a AX-18 Crawler, the AX-18 Crawler will behave much like an AX-12 Crawler in terms of speed. To get the full speed of the AX-18's out of the box you will need to make modifications to the code.

The following is offered as a starting out point: you may need to modify other parts of the code if you see any irregularities in your crawler's behavior.

Once you've opened the appropriate **...AX12_STOCK** firmware, open the **Hex_Cfg.h** tab. Look for the following line (Lines 94 for MK-II, 107 for MK-I)

```
#define DEFAULT_GAIT_SPEED 50 // Default gait speed - Will depend on what Servos you are using...
```

50 defines the speed of the crawler. By lowering this value you can increase the speed of the crawler. We currently do not have a recommended value, so you will need to test

ArbotiX Commander Firmware

The microcontroller on every ArbotiX Commander ships with the stock Commander.ino firmware. To take complete advantage of the PhantomX Hexapod running Phoenix Code, you will need to switch the microcontroller's firmware.

For best performance, use the **CommanderEX** firmware([Direct Zip Download](#)). This firmware is based on the original commander firmware and has been modified by KurtE. It gives you control over the Hexapod while running the Phoenix Code.

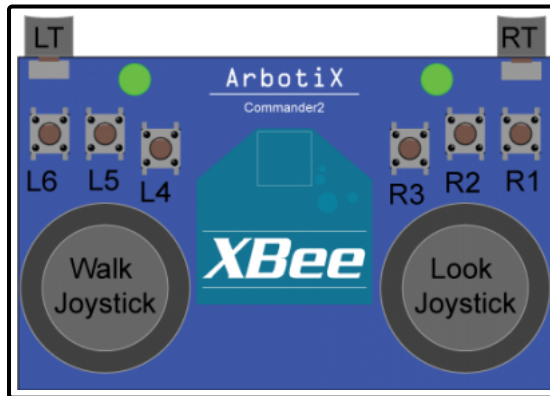
Once downloaded, upload the CommanderEX.ino sketch to the Geekduino or Seeeduino that came with your ArbotiX Commander.

If you change back to the stock NUKE code, we recommend setting the ArbotiX Commander back to the stock commander.ino firmware.

Controls

When controlling the PhantomX Hexapod, first turn on your crawler and commander. If everything is in working order, the crawler should beep and move its legs to a sitting position. Press **Left Top** to move the crawler to walking height. Now you can use the joysticks to move the crawler. Try using **R1**, **L4** and **Right Top** to change walking parameters. Press **Left Top** to move i

Translate, and single leg mode. Hold R1 and move the Look/Right Joystick up/down to raise and lower the body. See the following control list for more control options and details.



Change Modes

LEFT TOP BUTTON

Cycle through the crawler's different modes:

- Walk Mode
- Translate Mode
- Rotate Mode
- Single Leg Mode

When the mode is changes, the Crawler will beep once. Every time that **Walk Mode** is entered, the Crawler will beep twice.

Walk Mode - Walk Method 1 (Default)

WALK/LEFT JOYSTICK - VERTICAL ↓

Move Forward/Back

WALK/LEFT JOYSTICK - HORIZONTAL ↔

Strafe Left/Right

LOOK/RIGHT JOYSTICK - VERTICAL ↓

Not Used

LOOK/RIGHT JOYSTICK - HORIZONTAL ↔

Rotate Left/Right

Walk Mode - Walk Method 2

WALK/LEFT JOYSTICK (VERTICAL ↓)

Not Used

WALK/LEFT JOYSTICK - HORIZONTAL ↔

Not Used

LOOK/RIGHT JOYSTICK - VERTICAL ↓

Move Forward/Back

LOOK/RIGHT JOYSTICK - HORIZONTAL↔

Rotate Left/Right

Walk Mode - Modifiers

R1

Change Walk Gait. Every time the gait is changed the crawler will beep once. When the crawler returns to the first gait it will beep twice.

- Ripple Gait 12 Steps
- Tripod Gait 8 Steps
- Triple Tripod Gait 12 Steps
- Triple Tripod Gait 16 Steps

- Wave gait 24 steps
- Tripod 6 steps

You cannot change the gait while the crawler is in motion - wait for the crawler to stop to change gait.

R2

Toggle Walk Method back and forth between **Walk Method 1** and **Walk Method 2**.

R3

Not currently used (will later be used to go into walk mode)

L4

Toggle Balance Mode on/off

RIGHT TOP

Cycle through Walk options. Every time the walk method is changed the crawler will beep once. When the crawler returns to the first walk option it will beep twice.

- Normal Walk
- Double Height
- Double Travel

Translate Mode

WALK/LEFT JOYSTICK

Shift Body X/Z

LOOK/RIGHT JOYSTICK

Shift body Y and rotate body Y

Rotate Mode

WALK/LEFT JOYSTICK

Rotate Body X/Z

LOOK/RIGHT JOYSTICK

Rotate Body Y

Single Leg Mode

R1

Change Leg

WALK/LEFT JOYSTICK

Move Leg X/Z

LOOK/RIGHT JOYSTICK

Move Lef Y

R2

Hold /release leg position

Body & Speed Modifiers

L5

Sit/Stand Crawler to pre-determined heights.

HOLD L6 + RIGHT JOYSTICK - VERTICAL ↑

Move the Crawler's body up/down

HOLD L6 + RIGHT JOYSTICK - HORIZONTAL ↔

Change the crawler's speed to faster/slower (crawler defaults to full speed)

FAQ

I've seen the Hexapod using the Phoenix Code flip up-side-down. Is that still possible?

It is possible to flip the original MK-I Hexapod. You will need to load a different sketch onto your hexapod.

File -> Examples -> Phoenix -> PhantomX_Commander_AX12

To make this work, you will need to make hardware modifications to your Hexapod (See line 39 of the Hex_Cfg.h as well as [this thread](#)). You will also need to add an accelerometer

This code also does voltage monitoring - you will need to implement the monitoring hardware or disable it.

The MK-II Hexapod does not support this behavior because of the design in its legs.

Is there support for PS2/PS3 Controllers or other input methods?

Right now only the ArbotiX commander is supported by the ArbotiX Robocontroller Hardware. It may be possible to interface other controls to the ArbotiX Robocontroller and use the methods, but this may require alternative code changes

Is there Phoenix Code Support for the PhantomX Quadraped?

There is a [development branch](#) of the `Arduino_Phoenix_Parts` repo that has support for the Quadraped. This is still under development.

What is GP Mode?

GP or General Purpose Mode will allow you to run different poses. This mode is not implemented by default and is still under development.

Is the Phoenix Code available for other platforms?

Yes. See the [Arduino_Phoenix_Parts repository](#)

Links

- [KurtE's Arduino_Phoenix_Parts Github Repository](#) - Scroll Down for Readme
- [KurtE's Phantom_Phoenix Github Repository](#) - Scroll Down for Readme
- [KurtE's Commander EX Repository](#)
- [Thinking about trying out a different Hex robot...](#) - Forum Thread detailing the evolution of the port of the Phoenix code to the PhantomX Hexapod
- [Playing with a different breed](#) - Zenta's Blog
- [Demo](#) - Zenta's Blog
- [phantomx hexapod](#)
- [Project](#)

Attributions