# FingerLib.h - Finger Control Library

## Introduction

This library allows an Arduino Mega or Arduino Zero (or M0) to control an Open Bionics finger, which allows for control of an Open Bionics robotic hand. Each finger is actuated using a Firgelli linear actuator, where the position is determined by an embedded rotary potentiometer. The output of the potentiometer is fed into the 10-bit ADC, therefore the finger can move to positions between 0 - 1024.

This library is designed to work with the Open Bionics Almond board, which includes an Atmega2560 and 3 x dual channel motor drivers. Each finger requires 2 digital pins and 1 ADC pin, the maximum number of fingers is limited to 6.

FingerLib.h is designed to closely resemble the Servo.h library, to allow ease of implementation for hobbyists and researchers, and is a work in progress.

If you find any issues with the library or want to request more features, feel free to email us at ollymcbride@openbionics.com

## Functions

### Setup/Initialisation

attach()

detach()

attached()

setPosLimits()

setSpeedLimits()

invertFingerDir()

### Basic Movement

open()

close()

open_close()

## Control

writePos()

readPos()

readPosError()

readTargetSpeed()

writeSpeed()

readSpeed()

readTargetSpeed()

writeDir()

readDir()

stopMotor()

disableMotor()

enableMotor()

reachedPos()

## Debugging

printSpeed()

printPos()

printPosError()

printDir()

printReached()

printDetails()

# Reference

## attach()

### Description

Attach the finger to direction and sense pins. Note that the direction pins require the use of PWM, and the sense pin is required to be an ADC pin.

### Syntax

finger.attach(dir1, dir2, sense)

### Parameters

finger: a variable of type Finger

dir1: a PWM pin attached to the motor driver

dir2: a PWM pin attached to the motor driver

sense: an ADC pin connected to the motor

## detach()

### Description

Detach the finger from its pins.

### Syntax

finger.detach()

### Parameters

finger: a variable of type Finger

## attached()

### Description

Check whether the finger variable is attached to pins.

### Syntax

finger.finger()

### Parameters

finger: a variable of type Finger

### Return

true if the finger is attached to a pin; false otherwise.

## setPosLimits()

### Description

Set the position limits for the finger, from 0 - 1024. Defaults min = 50, max = 975.

### Syntax

finger.setPosLimits(min, max)

### Parameters

finger: a variable of type Finger

min: the minimum/open position of the finger

max: the maximum/closed position of the finger

## setSpeedLimits()

### Description

Set the speed limits for the finger, from 0 - 255. Note, the PWM frequency is set to 31KHz to reduce the audible hum, which reduces the range of finger speed (the finger does not move at speeds 0 - 150). Defaults min = 150, max = 250.

### Syntax

finger.setSpeedLimits(min, max)

### Parameters

finger: a variable of type Finger

min: the minimum speed of the finger

max: the maximum speed of the finger

## invertFingerDir()

### Description

Inverts the complete function of the finger, including both the open/close direction and the readPos() value

### Syntax

finger.invertFingerDir()

### Parameters

finger: a variable of type Finger

## open()

### Description

Opens the finger to the min/fully open position.

### Syntax

finger.open()

### Parameters

finger: a variable of type Finger

## close()

### Description

Closes the finger to the max/fully closed position.

### Syntax

finger.open()

### Parameters

finger: a variable of type Finger

## open_close()

### Description

Opens or closes the the finger to the fully open/closed position, determined by the function argument.

### Syntax

finger.open_close(direction)

### Parameters

finger: a variable of type Finger

direction: either 0 (OPEN) or 1 (CLOSE)

## writePos()

### Description

Moves the finger to a desired position, from 0 - 1024. Note, we recommend using a position between 50 - 975 to prevent the motor trying to move to a position that may damage it.

### Syntax

finger.writePos(position)

### Parameters

finger: a variable of type Finger

position: the position to write to the finger, from 0 - 1024

## readPos()

### Description

Returns the current position of the finger as a position value from 0 - 1024.

### Syntax

finger.readPos()

### Parameters

finger: a variable of type Finger

### Return

The position of the finger, from 0 - 1024.

## readPosError()

### Description

Returns the positional error between the current and target position, as a value from 0 - 1024.

### Syntax

finger.readPosError()

### Parameters

finger: a variable of type Finger

### Return

The positional error of the finger, from 0 - 1024.

## readTargetPos()

### Description

Returns the target position of the finger, as a value from 0 - 1024.

### Syntax

finger.radTargetPos()

### Parameters

finger: a variable of type Finger

### Return

The target position of the finger, from 0 - 1024.

## writeSpeed()

### Description

Sets the speed of the finger for all future movements. Note, the PWM frequency is set to 31KHz to reduce the audible hum, which reduces the range of finger speed (the finger does not move at speeds 0 - 150).

### Syntax

finger.writeSpeed(speed)

### Parameters

finger: a variable of type Finger

speed: the speed to move the finger, from 0 - 250

## readSpeed()

### Description

Returns the current speed of the finger, from 0 - 255.

### Syntax

finger.readSpeed()

### Parameters

finger: a variable of type Finger

### Return

The speed of the finger, from 0 - 255.

## readTargetSpeed()

### Description

Returns the target speed of the finger, as a value from 0 - 255.

### Syntax

finger.readTargetSpeed()

### Parameters

finger: a variable of type Finger

### Return

The target speed of the finger, from 0 - 255.

## writeDir()

### Description

Set the finger to move to the max position in a given direction, where 0 = Open and 1 = Close.

### Syntax

finger.writeDir(direction)

### Parameters

finger: a variable of type Finger

direction: the direction to move the finger, 0 (Open) and 1 (Close)

## readDir()

### Description

Returns the current direction the finger is moving, where 0 = Open and 1 = Close.

### Syntax

finger.readDir()

### Parameters

finger: a variable of type Finger

### Return

The current direction of travel of the of the finger, where 0 = Open and 1 = Close.

## stopMotor()

### Description

Stop the motor and current maintain position

### Syntax

finger.stopMotor()

### Parameters

finger: a variable of type Finger

## disableMotor()

### Description

Stop the motor, and allow it to be backdriven. Requires enableMotor() to restart motor.

### Syntax

finger.disableMotor()

### Parameters

finger: a variable of type Finger

## enableMotor()

### Description

Restarts the motor after disableMotor()

### Syntax

finger.enableMotor()

### Parameters

finger: a variable of type Finger

## reachedPos()

### Description

Returns true if the target position has been reached (if the positional error < 50). If a value is entered as an argument, this new value overwrites the positional error threshold of 50, allowing for a custom ‘reached’ tolerance.

### Syntax

finger.reachedPos()

finger.reachedPos(posErr)

### Parameters

finger: a variable of type Finger

posErr: a value denoting the tolerance either side of the target position, in which the target position is ‘reached’

### Return

true if the target position is reached; otherwise false.

## printSpeed()

### Description

Prints the current speed of the finger in the format ‘Speed ###’. (optional ‘\n’ after the speed value)

### Syntax

finger.printSpeed()

finger.printSpeed(newLine)

### Parameters

finger: a variable of type Finger

newLine: (optional) if true, a new line is entered after the speed

## printPos()

### Description

Prints the current position of the finger in the format ‘Pos ####’. (optional ‘\n’ after the position value)

### Syntax

finger.printPos()

finger.printPos(newLine)

### Parameters

finger: a variable of type Finger

newLine: (optional) if true, a new line is entered after the position

## printPosError()

### Description

Prints the positional error between the target and the current finger position, in the format Err####’. (optional ‘\n’ after the position value)

### Syntax

finger.printPosErr()

finger.printPosErr(newLine)

### Parameters

finger: a variable of type Finger

newLine: (optional) if true, a new line is entered after the position

## printDir()

### Description

Prints the current direction of the finger in the format ‘Dir #’. (optional ‘\n’ after the direction value)

### Syntax

finger.printDir()

finger.printDir(newLine)

### Parameters

finger: a variable of type Finger

newLine: (optional) if true, a new line is entered after the direction

## printReached()

### Description

Prints whether the finger has reached the target position in the format of ‘Reached #’. (optional ‘\n’ after the reached value)

### Syntax

finger.printReached()

finger.printReached(newLine)

### Parameters

finger: a variable of type Finger

newLine: (optional) if true, a new line is entered after the reached value

## printDetails()

### Description

Prints the finger number, speed, current position, direction, reached, followed by a new line.

### Syntax

finger.printReached()

### Parameters

finger: a variable of type Finger