# iButtonBox Documentation

**Fall 2017**

Oliver MacNeely, oliverm@mit.edu
Department of Mechanical Engineering, CSAIL
MIT, Cambridge MA 02139



## 1 iButtonBox Overview

The button box is a physical device that allows button-based input into MOOS apps. Through the programming of four different buttons, a user can have each button correspond to the state of a variable, which is then posted to the MOOS database, where pushing a button will update the given variable with a binary value (true/false).

## 2 Setting up the ButtonBox

The ButtonBox consists of two parts, the arduino inside of it and several buttons on the outside. Setting up the ButtonBox takes several steps: uploading code to the arduino, connecting it to the computer, and ensuring connection to the MOOS database.

### 2.1 Uploading code to the Arduino

The first step is to download the most recent version of the Arduino IDE which is freely available online. After downloaded, open the application and open the file at *moos-ivp-aquaticus/trunk/src/iButtonBox/Button* within the IDE. Then plug the Arduino into your computer and, from the Tools dropdown menu in the Arduino IDE toolbar, select the correct model (Tools/Board) and correct port (Tools/Port ) that the Arduino is connected to. Once these are chosen, upload the code to the Arduino with the arrow button and reconnect the Arduino to the button box case (if it was taken out to upload code).

### 2.2 Connecting the ButtonBox

After uploading the code to the Arduino, connect the assembled button box to the mokai or computer that it is being used with. Then open the terminal on this device and look in the directory */dev/* for a device with a name of the form:

tty.usbmodemXXXXX

(where XXXXX stands in for numbers)

Keep track of this number for use in the .moos configuration file for the iButtonBox application. If multiple devices exist, unplugging other usb input devices and then navigating to */dev* might help.

## 3 Using the iButtonBox Application

### 3.1 Receiving Mail

The iButtonBox applications receives no mail from any MOOS apps, but does receieve data from the connected arduino via string-type messages that it parses. The standard user shouldn't have to worry about this communication or interact with it.

### 3.2 Configuration Parameters of pExample

The iButtonBox app has three different configuration parameters: `PORT`, `BAUDRATE`, and `BUTTON_X_NAME`.

Note the quick-listing of parameters is given in Section 4.

#### 3.2.1 The `PORT` Configuration Parameter

This configuration parameter is used to tell the iButtonBox application the correct adress to read serial input from. The correct value for this parameter is the adress of the arduino plugged into the computer. This will be of the form

*/dev/tty.usbmodemXXXXX*

Where XXXXX is the number that we found when we checked */dev* for the correct device.

### 3.2.2 The BAUDRATE Configuration Parameter

This configuration parameter is used to control the rate of information transfer between the arduino and the computer and should always be given a value of 9600.

### 3.2.3 The BUTTON_X_NAME Configuration Parameter

This parameter changes depending on the number of buttons that you are controlling and the function of each button. Currently buttons are only used for updating variable values in the MOOS database with a value of "TRUE" or "FALSE". In order to control the function of a button, change the $X$ in the parameter name to a value from 0 to 3 (0 is red, 1 is blue, etc). Then set the value for this parameter to the name of the variable that you would like the button to update upon being pressed.

# 4 Configuration Parameters for iButtonBox

The iButtonBox application may be configured with a configuration block within a MOOS mission file, typically with a `.moos` file suffix. The following parameters are defined for iButtonBox:

*Listing 4.1: Configuration Parameters for iButtonBox.*

| | |
|---|---|
| PORT: | The port to which the arduino is connected. It can change each time the ButtonBox is unplugged and plugged back in and should be checked by looking at the contents of */dev*. It will be of the form */dev/tty.usbmodemXXXXX*. See Section 3.2.1. |
| BAUDRATE: | The rate at which information is transferred via serial between the arduino and the computer, its value should be 9600. See Section 3.2.2 |
| BUTTON_X_NAME: | This parameter allows the user to define which button corresponds to which MOOS variable. The parameter will be of the form BUTTON_X_NAME where $X$ will be a button number (0-3). The value will take the form of the variable that the button will correspond to. See Section 3.2.3 |

## 4.1 An Example MOOS Configuration Block

An example MOOS configuration block is provided in Listing 2 below.

*Listing 4.2: Example configuration of the iButtonBox application.*

```
1   ================================================================================
2   iButtonBox Example MOOS Configuration
3   ================================================================================
4
5   ProcessConfig = iButtonBox
6   {
7     AppTick        = 50
8     CommsTick      = 50
9
10    PORT = /dev/tty.usbmodemXXXXX
```

```
11    BAUDRAET = 9600
12    BUTTON_X_NAME = VARIABLE_NAME
23  }
```

# 5 Publications and Subscriptions for iButtonBox

The interface for iButtonBox, in terms of publications and subscriptions, is described below.

## 5.1 Variables Published by iButtonBox

- EXAMPLE_VARIABLE: Publishes a TRUE or FALSE value for for each variable assigned in config.

## 5.2 Variables Subscribed for by iButtonBox

The iButtonBox doesn't subscribe for any variables.

# 6 Terminal and AppCast Output

The iButtonBox application produces some useful information to the terminal on every iteration of the application. An example is shown in Listing 3 below. This application is also appcast enabled, meaning its reports are published to the MOOSDB and viewable from any uMAC application or pMarineViewer.

*Listing 6.3: Example terminal or appcast output for iButtonBox.*

```
 1  ===============================================================
 2  iButtonBox alpha                                        0/0(442)
 3  ===============================================================
 4  SETUP
 5  -----
 6         PORT: /dev/tty.usbmodem14311
 7         BAUDRATE: 9600
 8
 9  STATUS
10  -----
11         Valid serial connection: true
12
13         No current button values.
14
```

The first few lines (6-7) show the configuration settings for iButtonBox. The setup status of iButtonBox is shown in Lines 9-13. The second status block, lines 15-21, indicate whether the arduino is connected (valid serial connection) and the values of the buttons.

# 7 Troubleshooting

When dealing with errors regarding the ButtonBox and communication with both the computer it is plugged into and/or the MOOS database, the best course of action is to make sure that

all wires and cables are securely attached and then to unplug and plug back in the ButtonBox, relaunching MOOSDB and double-checking the port of the arduino along the way. This should reset communication and alleviate any errors. Other issues should only occur with misconfigured configuration blocks.