**Bpod Protocol Development Guide**

12/21/16

Documentation is in the Bpod Wiki under Protocol Development at the following URL:

<https://sites.google.com/site/bpoddocumentation/bpod-user-guide/protocol-writing>

Function setup

1. Create a new folder in C:\Bpod\Protocols named it after your new function.
2. Open a new Matlab file named after the function you will write. The Bpod download includes pre-written functions. It might be helpful to use one of these functions (such as “MyProtocol”) as a template.

Initialize variables

1. Initialize variable global BpodSystem to allow the Bpod to access the function. Adjust fields in BpodSystem’s ProtocolSettings structure as necessary.
2. Make sure valves are properly calibrated to dispense reward volume accurately using the function GetValveTimes. Calibration instructions [here](https://sites.google.com/site/bpoddocumentation/bpod-user-guide/liquid-calibration).
3. Define number and type of trials. For a 2AFC task, randomly generate a vector of 1's or 2's that is nTrials long.
4. Initialize all other variables necessary for your protocol.

Write a for loop

1. Increment the index from 1 to the number of trials (i.e., for currentTrial = 1:MaxTrials)

Define and send state matrix.

1. To add a new state to the matrix, you must define the following arguments:
   1. The name of the state matrix
   2. 'Name': the name of your state
   3. 'Time': time/duration of the state's output action
   4. 'StateChangeConditions': a cell array of the conditions under which a new state is entered and the names of the next states to enter given those conditions
   5. 'OutputActions': A list of the output actions that occur when the state is entered

For example, to add a state that tells Bpod to reward a left or right nose poke:

MyStateChangeConditions = {'Port1In', 'LeftReward', ...

'Port3In', 'RightReward'};

sma = AddState(sma, 'Name', 'WaitForChoice', ...

'Timer', 0,...

'StateChangeConditions', MyStateChangeConditions,...

'OutputActions', {'SoftCode', 2});

When this state is entered, the Bpod sends a byte of information (the number 2) to the computer (see description below). The state 'LeftReward' is entered if the mouse pokes Port 1, and the state 'RightReward' is entered if the mouse pokes Port 3. The matrix remains in this state until the mouse pokes either port.

Write function to handle SoftCode outputs.

1. The main function does not execute new lines of code while the state matrix is running. To update PsychToolbox commands in response to state changes, use the output action 'SoftCode' and write a separate handler function.
2. Save the handler function in the same folder as the protocol function.
3. To give Bpod access to your handler function, use the following line within the loop but before defining the state matrix:

BpodSystem.SoftCodeHandlerFunction = 'SoftCodeHandler\_MyHandler'