Start

update something in the master

firstChar = 'M' — False →

True ↓

secChar = inString[1]
param = inString[3-underscore]
value = inString[underscore-end]

update master parameter

secChar = 'M' — False →

True ↓

param contains "timebt" — False →

True ↓

update **timeBetweenRequests**

end of function

update something in the master's array of slaves (ex: vial mode)

MS_debug_5_A12C3

secChar = 'S' — False →

True ↓

get list of vials to send this to
(everything after the last underscore)

for each slave[s] → end of function

if slave name is in string

slaveName is in vialList — False

True ↓

it's not the last slave — False →

True ↓

get the vial #s for just this slave

for each slave[ns] after it

slaveName is in vialList — False →

True ↓

remove from that slave through the end
(leaving you with just the vial numbers for this slave)

thisSlaveVials
(remove everything before/including this slaveName)

update this vial's mode with paramToSend

for each vial[k] in thisSlaveVials

convert to int

for each vial[p] in this slave[s]

slave.vials[p] matches this vialNum — False →

True ↓

arr_slaves[s].vials[p].mode = paramToSend

send command to slave

S_OV_5_A123B1C4
S_Kx_I0.001_B23D1

firstChar = 'S' — False →

True ↓

get paramToSend — *OV_5_*
(char after the first underscore -> last underscore)

get vialList — *A123B1C4*
(everything after the last underscore)

for each slave[s]

if slave name is in string

slaveName is in vialList — False — *A123B1C4*

True ↓
* string must have slaves in order *

it's not the last slave — False →

True ↓

get the vial #s for just this slave

for each slave[ns] after it

slaveName is in vialList — False →

True ↓

remove from that slave through the end
(leaving you with just the vial numbers for this slave)
thisSlaveVials = A12

thisSlaveVials
(remove everything before/including this slaveName)
thisSlaveVials = 12

make the new string and send it to that slave

toSend
paramToSend + thisSlaveVials
Sp_640_123
Sp_640_1
Sp_640_4

convert toSend to a char array

send **toSend** to **slaveToSendTo**
(as a char array)

check which slaves are active

firstChar = 'C' — False → end of function

True ↓

for each slave[i]

sayHey(i)

returned value = 99 — False →

True ↓

slave is not active / slave is active

printSlaveInfo

create string to write and print it

end of function

confirm that slaves saved parameter updates

for each slave[s] → end of function

if slave name is in string

slaveName is in vialList — False

True ↓

it's not the last slave — False →

True ↓

get the vial #s for just this slave

for each slave[ns] after it

slaveName is in vialList — False →

True ↓

thisSlaveVials
(remove everything before/including this slaveName)

thisSlaveVials
(remove everything before/including this slaveName)

for each vial in this list

requestConfirmation