Automated Microfluidic Platform

1.0

Generated by Doxygen 1.8.15

1 Namespace Index 1
1.1 Namespace List . . . 1

2 Hierarchical Index 3
2.1 Class Hierarchy . . . 3

3 Class Index 5
3.1 Class List . . . 5

4 File Index 7
4.1 File List . . . 7

5 Namespace Documentation 9
5.1 Collection Namespace Reference . . . 9
5.1.1 Detailed Description . . . 9
5.2 fakeSerial Namespace Reference . . . 9
5.2.1 Detailed Description . . . 9
5.3 gui Namespace Reference . . . 9
5.3.1 Detailed Description . . . 10
5.3.2 Function Documentation . . . 10
5.3.2.1 check_input() . . . 10
5.3.2.2 combine_funcs() . . . 10
5.3.2.3 message_prompt() . . . 10
5.3.3 Variable Documentation . . . 11
5.3.3.1 app . . . 11
5.3.3.2 LARGE_FONT . . . 11
5.4 Mswitch Namespace Reference . . . 11
5.4.1 Detailed Description . . . 11
5.5 Pump Namespace Reference . . . 11
5.5.1 Detailed Description . . . 11
5.5.2 Function Documentation . . . 12
5.5.2.1 get_ports() . . . 12
5.6 TwoSwitch Namespace Reference . . . 12
5.6.1 Detailed Description . . . 12

6 Class Documentation 13
6.1 gui.App Class Reference . . . 13
6.1.1 Detailed Description . . . 14
6.1.2 Constructor & Destructor Documentation . . . 14
6.1.2.1 __init__() . . . 14
6.1.3 Member Function Documentation . . . 14
6.1.3.1 done_and_load() . . . 14
6.1.3.2 get_comport() . . . 15
6.1.3.3 get_ports() . . . 15

6.1.3.4 message_window() . . . 15
6.1.3.5 negative_run_status() . . . 15
6.1.3.6 positive_run_status() . . . 16
6.1.3.7 set_defaults() . . . 16
6.1.3.8 show_frame() . . . 16
6.1.4 Member Data Documentation . . . 16
6.1.4.1 appHighlightFont . . . 16
6.1.4.2 buttonFont . . . 16
6.1.4.3 default_diameter_index . . . 17
6.1.4.4 frames . . . 17
6.1.4.5 hasColl . . . 17
6.1.4.6 hasMani . . . 17
6.1.4.7 hasPump . . . 17
6.1.4.8 max_flowrate_1 . . . 17
6.1.4.9 max_flowrate_2 . . . 17
6.1.4.10 max_flowrate_3 . . . 18
6.1.4.11 max_flowrate_list . . . 18
6.1.4.12 my2Switch . . . 18
6.1.4.13 myColl . . . 18
6.1.4.14 myMani . . . 18
6.1.4.15 myPump . . . 18
6.1.4.16 pageFont . . . 18
6.1.4.17 port2Switch . . . 18
6.1.4.18 portColl . . . 19
6.1.4.19 portMani . . . 19
6.1.4.20 portPump . . . 19
6.1.4.21 ports . . . 19
6.1.4.22 should_be_running . . . 19
6.1.4.23 well_working_volume . . . 19
6.2 gui.AutomaticPage Class Reference . . . 19
6.2.1 Constructor & Destructor Documentation . . . 20
6.2.1.1 __init__() . . . 20
6.2.2 Member Function Documentation . . . 20
6.2.2.1 add_step() . . . 21
6.2.2.2 check_channel() . . . 21
6.2.2.3 clear_recipe() . . . 21
6.2.2.4 create_step() . . . 21
6.2.2.5 delete_step() . . . 21
6.2.2.6 file_load() . . . 21
6.2.2.7 file_save() . . . 22
6.2.2.8 run_recipe() . . . 22
6.2.3 Member Data Documentation . . . 22

6.2.3.1 channels . . . 22
6.2.3.2 clearbutton . . . 22
6.2.3.3 deletestep . . . 22
6.2.3.4 deletestepcombo . . . 22
6.2.3.5 ejectbutton . . . 23
6.2.3.6 headinglabel . . . 23
6.2.3.7 loadbutton . . . 23
6.2.3.8 mainlabel . . . 23
6.2.3.9 manualbutton . . . 23
6.2.3.10 newstep . . . 23
6.2.3.11 number_deleted . . . 23
6.2.3.12 numberlist . . . 23
6.2.3.13 reservoirs . . . 24
6.2.3.14 runbutton . . . 24
6.2.3.15 savebutton . . . 24
6.2.3.16 settingsbutton . . . 24
6.2.3.17 steplist . . . 24
6.2.3.18 stepnumbers . . . 24
6.2.3.19 todelete . . . 24
6.2.3.20 twoSwitchValues . . . 25
6.3 Collection.Collectador Class Reference . . . 25
6.3.1 Detailed Description . . . 25
6.3.2 Constructor & Destructor Documentation . . . 26
6.3.2.1 __init__() . . . 26
6.3.3 Member Function Documentation . . . 26
6.3.3.1 chop_return() . . . 26
6.3.3.2 eject() . . . 26
6.3.3.3 get_info() . . . 27
6.3.3.4 last_site() . . . 27
6.3.3.5 moveOneDown() . . . 27
6.3.3.6 moveOneLeft() . . . 27
6.3.3.7 moveOneRight() . . . 27
6.3.3.8 moveOneUp() . . . 28
6.3.3.9 next_site() . . . 28
6.3.3.10 reset() . . . 28
6.3.3.11 send() . . . 28
6.3.3.12 serial_connect() . . . 29
6.3.3.13 setOrigin() . . . 29
6.3.3.14 toggle_pattern() . . . 29
6.3.4 Member Data Documentation . . . 29
6.3.4.1 baud . . . 29
6.3.4.2 connected . . . 29

6.3.4.3 currentPattern . . . 30
6.3.4.4 port . . . 30
6.3.4.5 position . . . 30
6.3.4.6 ser . . . 30
6.3.4.7 serialConnected . . . 30
6.3.4.8 uniqueID . . . 30
6.3.4.9 usingSnakePattern . . . 30
6.4 gui.ManualPage Class Reference . . . 31
6.4.1 Constructor & Destructor Documentation . . . 32
6.4.1.1 __init__() . . . 32
6.4.2 Member Function Documentation . . . 32
6.4.2.1 check_all_manual() . . . 32
6.4.2.2 check_all_manual_and_sampling() . . . 32
6.4.2.3 check_channel() . . . 33
6.4.2.4 create_channels() . . . 33
6.4.2.5 done() . . . 33
6.4.2.6 eject() . . . 33
6.4.2.7 moveOneDown() . . . 33
6.4.2.8 moveOneLeft() . . . 34
6.4.2.9 moveOneRight() . . . 34
6.4.2.10 moveOneUp() . . . 34
6.4.2.11 next() . . . 34
6.4.2.12 prev() . . . 34
6.4.2.13 reset() . . . 34
6.4.2.14 set_and_start() . . . 35
6.4.2.15 set_and_start_all() . . . 35
6.4.2.16 setOrigin() . . . 35
6.4.2.17 start_and_collect() . . . 35
6.4.2.18 stop() . . . 35
6.4.2.19 stop_all() . . . 36
6.4.2.20 toggle() . . . 36
6.4.3 Member Data Documentation . . . 36
6.4.3.1 autocollectlabel . . . 36
6.4.3.2 autopagebutton . . . 36
6.4.3.3 channels . . . 36
6.4.3.4 channeltextlist . . . 36
6.4.3.5 CollectionControlLabel . . . 37
6.4.3.6 ejectbutton . . . 37
6.4.3.7 flowrateentrylist . . . 37
6.4.3.8 greaterlabelframe . . . 37
6.4.3.9 labelframelist . . . 37
6.4.3.10 lastwellbutton . . . 37

6.4.3.11 mainlabel . . . 37
6.4.3.12 nextwellbutton . . . 37
6.4.3.13 patternLabel . . . 38
6.4.3.14 positionLabel . . . 38
6.4.3.15 reservoircombobox . . . 38
6.4.3.16 reservoirlabel . . . 38
6.4.3.17 reservoirs . . . 38
6.4.3.18 reservoirvalues . . . 38
6.4.3.19 resetbutton . . . 38
6.4.3.20 sampleslabel . . . 38
6.4.3.21 samplingrate . . . 39
6.4.3.22 samplingrateentry . . . 39
6.4.3.23 setOriginButton . . . 39
6.4.3.24 settingsbutton . . . 39
6.4.3.25 sites . . . 39
6.4.3.26 sitesentry . . . 39
6.4.3.27 startallbutton . . . 39
6.4.3.28 startandcollectbutton . . . 39
6.4.3.29 startbuttonlist . . . 40
6.4.3.30 stopallbutton . . . 40
6.4.3.31 stopbuttonlist . . . 40
6.4.3.32 togglebutton . . . 40
6.4.3.33 toplabels . . . 40
6.4.3.34 twoSwitchComboBoxList . . . 40
6.4.3.35 twoSwitchValues . . . 40
6.5 Mswitch.MSwitch Class Reference . . . 41
6.5.1 Detailed Description . . . 41
6.5.2 Constructor & Destructor Documentation . . . 41
6.5.2.1 __init__() . . . 41
6.5.3 Member Function Documentation . . . 41
6.5.3.1 chop_return() . . . 42
6.5.3.2 get_info() . . . 42
6.5.3.3 send() . . . 42
6.5.3.4 serial_connect() . . . 42
6.5.3.5 set_reservoir() . . . 43
6.5.4 Member Data Documentation . . . 43
6.5.4.1 baud . . . 43
6.5.4.2 connected . . . 43
6.5.4.3 port . . . 43
6.5.4.4 res . . . 43
6.5.4.5 ser . . . 43
6.5.4.6 serialConnected . . . 44

6.5.4.7 uniqueID . . . 44
6.6 fakeSerial.Serial Class Reference . . . 44
6.6.1 Detailed Description . . . 45
6.6.2 Constructor & Destructor Documentation . . . 45
6.6.2.1 __init__() . . . 45
6.6.3 Member Function Documentation . . . 45
6.6.3.1 __str__() . . . 45
6.6.3.2 close() . . . 46
6.6.3.3 flush() . . . 46
6.6.3.4 flushInput() . . . 46
6.6.3.5 flushOutput() . . . 46
6.6.3.6 isOpen() . . . 46
6.6.3.7 open() . . . 47
6.6.3.8 read() . . . 47
6.6.3.9 readline() . . . 47
6.6.3.10 write() . . . 47
6.6.4 Member Data Documentation . . . 48
6.6.4.1 baudrate . . . 48
6.6.4.2 bytesize . . . 48
6.6.4.3 name . . . 48
6.6.4.4 parity . . . 48
6.6.4.5 port . . . 48
6.6.4.6 rtscts . . . 48
6.6.4.7 stopbits . . . 48
6.6.4.8 timeout . . . 49
6.6.4.9 xonxoff . . . 49
6.7 gui.SettingsPage Class Reference . . . 49
6.7.1 Detailed Description . . . 50
6.7.2 Constructor & Destructor Documentation . . . 50
6.7.2.1 __init__() . . . 50
6.7.3 Member Function Documentation . . . 50
6.7.3.1 calibrate() . . . 50
6.7.3.2 update() . . . 50
6.7.4 Member Data Documentation . . . 50
6.7.4.1 autopagebutton . . . 50
6.7.4.2 calibratebutton . . . 51
6.7.4.3 calibratecombobox . . . 51
6.7.4.4 calibrationlabel . . . 51
6.7.4.5 channelcaliblabel . . . 51
6.7.4.6 channels . . . 51
6.7.4.7 channelvalues . . . 51
6.7.4.8 diameterlabel . . . 51

6.7.4.9 diameterlist . . . 51
6.7.4.10 diametervalues . . . 52
6.7.4.11 directionheading . . . 52
6.7.4.12 directionlabellist . . . 52
6.7.4.13 directions . . . 52
6.7.4.14 directionslist . . . 52
6.7.4.15 entercalibratedbutton . . . 52
6.7.4.16 mainlabel . . . 52
6.7.4.17 manualbutton . . . 52
6.7.4.18 measuredVolume . . . 53
6.7.4.19 measuredVolumeEntry . . . 53
6.7.4.20 reservoirs . . . 53
6.7.4.21 updatebutton . . . 53
6.8 Pump.ThreePump Class Reference . . . 53
6.8.1 Detailed Description . . . 54
6.8.2 Constructor & Destructor Documentation . . . 54
6.8.2.1 __init__() . . . 54
6.8.3 Member Function Documentation . . . 54
6.8.3.1 abort_calibration() . . . 54
6.8.3.2 calibrate() . . . 55
6.8.3.3 chop_return() . . . 55
6.8.3.4 FormatVolume() . . . 55
6.8.3.5 send() . . . 55
6.8.3.6 send_return() . . . 56
6.8.3.7 serial_connect() . . . 56
6.8.3.8 setDefaults() . . . 56
6.8.3.9 setDir() . . . 56
6.8.3.10 setFlow() . . . 57
6.8.3.11 setMeasuredVolume() . . . 57
6.8.3.12 setTargetCalibrationVolume() . . . 57
6.8.3.13 start() . . . 57
6.8.3.14 start_all() . . . 58
6.8.3.15 stop() . . . 58
6.8.3.16 stop_all() . . . 58
6.8.4 Member Data Documentation . . . 58
6.8.4.1 baud . . . 58
6.8.4.2 calibrationunit . . . 58
6.8.4.3 calibrationvolume . . . 59
6.8.4.4 connected . . . 59
6.8.4.5 isOn . . . 59
6.8.4.6 port . . . 59
6.8.4.7 ser . . . 59

6.8.4.8 serialConnected . . . 59
6.9 TwoSwitch.TwoSwitch Class Reference . . . 59
6.9.1 Detailed Description . . . 60
6.9.2 Constructor & Destructor Documentation . . . 60
6.9.2.1 __init__() . . . 60
6.9.3 Member Function Documentation . . . 60
6.9.3.1 chop_return() . . . 60
6.9.3.2 getIdentifier() . . . 61
6.9.3.3 send() . . . 61
6.9.3.4 serial_connect() . . . 61
6.9.3.5 setCollect() . . . 61
6.9.3.6 setRecirculate() . . . 62
6.9.4 Member Data Documentation . . . 62
6.9.4.1 baud . . . 62
6.9.4.2 connected . . . 62
6.9.4.3 port . . . 62
6.9.4.4 ser . . . 62
6.9.4.5 serialConnected . . . 62
6.9.4.6 uniqueID . . . 63
6.9.4.7 willRecirculate . . . 63
6.10 gui.WelcomePage Class Reference . . . 63
6.10.1 Constructor & Destructor Documentation . . . 63
6.10.1.1 __init__() . . . 63
6.10.2 Member Function Documentation . . . 64
6.10.2.1 flip() . . . 64
6.10.3 Member Data Documentation . . . 64
6.10.3.1 coll_label . . . 64
6.10.3.2 collectorframe . . . 64
6.10.3.3 donebutton . . . 64
6.10.3.4 mainlabel . . . 64
6.10.3.5 manilabel . . . 64
6.10.3.6 perfusorframe . . . 65
6.10.3.7 pumplabel . . . 65
6.10.3.8 sublabelframe . . . 65

7 File Documentation **67**

7.1 ArduinoSnakePattern.cpp File Reference . . . 67
7.1.1 Detailed Description . . . 69
7.1.2 Function Documentation . . . 69
7.1.2.1 backwardstep1() . . . 69
7.1.2.2 backwardstep2() . . . 69
7.1.2.3 eject() . . . 69

7.1.2.4 forwardstep1() . . . 69
7.1.2.5 forwardstep2() . . . 70
7.1.2.6 loop() . . . 70
7.1.2.7 moveDown() . . . 70
7.1.2.8 moveGen() . . . 70
7.1.2.9 moveLast() . . . 70
7.1.2.10 moveLeft() . . . 71
7.1.2.11 moveNext() . . . 71
7.1.2.12 moveRight() . . . 71
7.1.2.13 moveUp() . . . 71
7.1.2.14 populateLocations() . . . 71
7.1.2.15 reset() . . . 72
7.1.2.16 set() . . . 72
7.1.2.17 setup() . . . 72
7.1.2.18 stepperB() . . . 72
7.1.2.19 stepperT() . . . 72
7.1.3 Variable Documentation . . . 72
7.1.3.1 acceleration . . . 72
7.1.3.2 AFMS . . . 73
7.1.3.3 columns . . . 73
7.1.3.4 delayRPM . . . 73
7.1.3.5 limitSwitchB . . . 73
7.1.3.6 limitSwitchT . . . 73
7.1.3.7 locations . . . 73
7.1.3.8 maximumSpeed . . . 74
7.1.3.9 myMotorB . . . 74
7.1.3.10 myMotorT . . . 74
7.1.3.11 myPosition . . . 74
7.1.3.12 numLocations . . . 74
7.1.3.13 positions . . . 74
7.1.3.14 rows . . . 75
7.1.3.15 steppers . . . 75
7.1.3.16 stepsPerRevolution . . . 75
7.1.3.17 stepsPerWell . . . 75
7.1.3.18 stepsToEject . . . 75
7.1.3.19 TwoSwitchA . . . 75
7.1.3.20 TwoSwitchB . . . 76
7.1.3.21 TwoSwitchC . . . 76
7.2 Collection.py File Reference . . . 76
7.3 fakeSerial.py File Reference . . . 76
7.4 gui.py File Reference . . . 76
7.5 locationPopulate.cpp File Reference . . . 77

7.5.1 Function Documentation . . . 77
7.5.1.1 main() . . . 77
7.5.1.2 populateLocations() . . . 78
7.5.2 Variable Documentation . . . 78
7.5.2.1 columns . . . 78
7.5.2.2 numLocations . . . 78
7.5.2.3 rows . . . 78
7.5.2.4 stepsPerWell . . . 78
7.5.2.5 stepsToEject . . . 78
7.6 Mswitch.py File Reference . . . 79
7.7 Pump.py File Reference . . . 79
7.8 TwoSwitch.py File Reference . . . 79
7.9 XYStage_LimitSwitch_TwoSwitch_Control.cpp File Reference . . . 79
7.9.1 Function Documentation . . . 81
7.9.1.1 backwardstep1() . . . 81
7.9.1.2 backwardstep2() . . . 81
7.9.1.3 eject() . . . 82
7.9.1.4 forwardstep1() . . . 82
7.9.1.5 forwardstep2() . . . 82
7.9.1.6 loop() . . . 82
7.9.1.7 moveDown() . . . 82
7.9.1.8 moveGen() . . . 82
7.9.1.9 moveLast() . . . 83
7.9.1.10 moveLeft() . . . 83
7.9.1.11 moveNext() . . . 83
7.9.1.12 moveRight() . . . 83
7.9.1.13 moveUp() . . . 83
7.9.1.14 populateLocations() . . . 84
7.9.1.15 reset() . . . 84
7.9.1.16 set() . . . 84
7.9.1.17 setup() . . . 84
7.9.1.18 stepperB() . . . 84
7.9.1.19 stepperT() . . . 84
7.9.2 Variable Documentation . . . 84
7.9.2.1 acceleration . . . 85
7.9.2.2 AFMS . . . 85
7.9.2.3 collectionPattern . . . 85
7.9.2.4 columns . . . 85
7.9.2.5 delayRPM . . . 85
7.9.2.6 limitSwitchB . . . 85
7.9.2.7 limitSwitchT . . . 86
7.9.2.8 locations . . . 86

7.9.2.9 maximumSpeed 86
7.9.2.10 myMotorB 86
7.9.2.11 myMotorT 86
7.9.2.12 myPosition 86
7.9.2.13 numLocations 87
7.9.2.14 positions 87
7.9.2.15 rows 87
7.9.2.16 steppers 87
7.9.2.17 stepsPerRevolution 87
7.9.2.18 stepsPerWell 87
7.9.2.19 stepsToEject 88
7.9.2.20 TwoSwitchA 88
7.9.2.21 TwoSwitchB 88
7.9.2.22 TwoSwitchC 88

Index **89**

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Collection 9
fakeSerial 9
gui 9
Mswitch 11
Pump 11
TwoSwitch 12

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Collection.Collectador 25
Frame
    gui.AutomaticPage 19
    gui.ManualPage 31
    gui.SettingsPage 49
    gui.WelcomePage 63
Mswitch.MSwitch 41
fakeSerial.Serial 44
Pump.ThreePump 53
Tk
    gui.App 13
TwoSwitch.TwoSwitch 59

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

gui.App 13
gui.AutomaticPage 19
Collection.Collectador 25
gui.ManualPage 31
Mswitch.MSwitch 41
fakeSerial.Serial 44
gui.SettingsPage 49
Pump.ThreePump 53
TwoSwitch.TwoSwitch 59
gui.WelcomePage 63

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

ArduinoSnakePattern.cpp
Arduino code for XYStage Movement and TwoSwitch control 67
Collection.py 76
fakeSerial.py 76
gui.py 76
locationPopulate.cpp 77
Mswitch.py 79
Pump.py 79
TwoSwitch.py 79
XYStage_LimitSwitch_TwoSwitch_Control.cpp 79

Chapter 5

Namespace Documentation

5.1 Collection Namespace Reference

Classes

- class Collectador

5.1.1 Detailed Description

```
Fluidic Handling Software
Ashutosh Agarwal Lab
University of Miami

by:
Liev Birman
Adiel Hernandez
```

5.2 fakeSerial Namespace Reference

Classes

- class Serial

5.2.1 Detailed Description

```
Written by:
D. Thiebaut
```

5.3 gui Namespace Reference

Classes

- class App
- class AutomaticPage
- class ManualPage
- class SettingsPage
- class WelcomePage

Functions

- def combine_funcs (*funcs)
- def message_prompt (message)
- def check_input (variable, desired_datatype, lower_limit=0, upper_limit=0)

Variables

- tuple LARGE_FONT = ("Verdana", 10)
- app = App()

5.3.1 Detailed Description

```
Fluidic Handling Software
Ashutosh Agarwal Lab
University of Miami

by:
Liev Birman
Adiel Hernandez
```

5.3.2 Function Documentation

5.3.2.1 check_input()

```
def gui.check_input (
               variable,
               desired_datatype,
               lower_limit = 0,
               upper_limit = 0 )
```

5.3.2.2 combine_funcs()

```
def gui.combine_funcs (
             * funcs )
```

```
Use this function for executing multiple functions simultaneously with Tkinter buttons
```

5.3.2.3 message_prompt()

```
def gui.message_prompt (
               message )
```

```
Creates a pop-up message of your choice
```

5.3.3 Variable Documentation

5.3.3.1 app

```
gui.app = App()
```

5.3.3.2 LARGE_FONT

```
tuple gui.LARGE_FONT = ("Verdana", 10)
```

5.4 Mswitch Namespace Reference

Classes

- class MSwitch

5.4.1 Detailed Description

```
Fluidic Handling Software
Ashutosh Agarwal Lab
University of Miami

by:
Liev Birman
Adiel Hernandez
```

5.5 Pump Namespace Reference

Classes

- class ThreePump

Functions

- def get_ports ()

5.5.1 Detailed Description

```
Fluidic Handling Software
Ashutosh Agarwal Lab
University of Miami

by:
Liev Birman
Adiel Hernandez
```

5.5.2 Function Documentation

5.5.2.1 get_ports()

```
def Pump.get_ports ( )
```

This method uses serial module's comports command to get and return serial ports.

5.6 TwoSwitch Namespace Reference

Classes

- class TwoSwitch

5.6.1 Detailed Description

```
Fluidic Handling Software
Ashutosh Agarwal Lab
University of Miami

by:
Liev Birman
Adiel Hernandez
```

Chapter 6

Class Documentation

6.1 gui.App Class Reference

Inherits Tk.

Public Member Functions

- def __init__ (self, ∗args, ∗∗kwargs)
- def show_frame (self, cont)
- def get_comport (self, cmd, response)
- def get_ports (self)
- def set_defaults (self)
- def done_and_load (self)
- def positive_run_status (self)
- def negative_run_status (self)
- def message_window (self, message)

Public Attributes

- appHighlightFont
- buttonFont
- pageFont
- ports
- max_flowrate_1
- max_flowrate_2
- max_flowrate_3
- max_flowrate_list
- default_diameter_index
- well_working_volume
- hasPump

  DEVICE HANDLING ################.
- hasMani
- hasColl
- portPump
- portMani
- portColl

- port2Switch
- myPump
- myMani
- myColl
- my2Switch
- should_be_running
- frames

  FRAME HANDLING ###############.

6.1.1 Detailed Description

```
The App class
- Does window handling
- Holds objects and variables that need to be referenced by multiple windows
```

6.1.2 Constructor & Destructor Documentation

6.1.2.1 __init__()

```
def gui.App.__init__ (
              self,
             * args,
             ** kwargs )
```

```
- define window size and expandability
- define fonts
- instantiates windows
```

6.1.3 Member Function Documentation

6.1.3.1 done_and_load()

```
def gui.App.done_and_load (
              self )
```

```
This function checks if we have these devices and then creates instances for each one. Open ManualPage when do
When we've finished specifying our devices in the WelcomePage window we run this by clicking done.
```

6.1.3.2 get_comport()

```
def gui.App.get_comport (
              self,
              cmd,
              response )
```

Use this function to automatically identify
a device. Input a command the device recognizes and a response you know
it should return. Make sure to use unique commands when spotting multiple devices.

6.1.3.3 get_ports()

```
def gui.App.get_ports (
              self )
```

fills self.ports list with comports currently connected to device

6.1.3.4 message_window()

```
def gui.App.message_window (
              self,
              message )
```

Creates a pop-up message of your choice formatted with
formatting from App class

6.1.3.5 negative_run_status()

```
def gui.App.negative_run_status (
              self )
```

sets self.should_be_running to Fasle.

6.1.3.6 positive_run_status()

```
def gui.App.positive_run_status (
              self )
```

sets self.should_be_running to True.

6.1.3.7 set_defaults()

```
def gui.App.set_defaults (
              self )
```

Sets defaults as defined within function

6.1.3.8 show_frame()

```
def gui.App.show_frame (
              self,
              cont )
```

Raises frame from self.frames dictionary

6.1.4 Member Data Documentation

6.1.4.1 appHighlightFont

```
gui.App.appHighlightFont
```

6.1.4.2 buttonFont

```
gui.App.buttonFont
```

6.1.4.3 default_diameter_index

`gui.App.default_diameter_index`

6.1.4.4 frames

`gui.App.frames`

FRAME HANDLING ###############.

6.1.4.5 hasColl

`gui.App.hasColl`

6.1.4.6 hasMani

`gui.App.hasMani`

6.1.4.7 hasPump

`gui.App.hasPump`

DEVICE HANDLING ###############.

6.1.4.8 max_flowrate_1

`gui.App.max_flowrate_1`

6.1.4.9 max_flowrate_2

`gui.App.max_flowrate_2`

6.1.4.10 max_flowrate_3

gui.App.max_flowrate_3

6.1.4.11 max_flowrate_list

gui.App.max_flowrate_list

6.1.4.12 my2Switch

gui.App.my2Switch

6.1.4.13 myColl

gui.App.myColl

6.1.4.14 myMani

gui.App.myMani

6.1.4.15 myPump

gui.App.myPump

6.1.4.16 pageFont

gui.App.pageFont

6.1.4.17 port2Switch

gui.App.port2Switch

6.1.4.18 portColl

```
gui.App.portColl
```

6.1.4.19 portMani

```
gui.App.portMani
```

6.1.4.20 portPump

```
gui.App.portPump
```

6.1.4.21 ports

```
gui.App.ports
```

6.1.4.22 should_be_running

```
gui.App.should_be_running
```

6.1.4.23 well_working_volume

```
gui.App.well_working_volume
```

The documentation for this class was generated from the following file:

- gui.py

6.2 gui.AutomaticPage Class Reference

Inherits Frame.

Public Member Functions

- def __init__ (self, parent, controller)
- def file_save (self)
- def file_load (self)
- def delete_step (self, step_number)
- def clear_recipe (self)
- def add_step (self)
- def create_step (self, channels, reservoirs)
- def check_channel (self, controller, channel, frame_object)
- def run_recipe (self, controller)

Public Attributes

- mainlabel
- channels
- reservoirs
- steplist
- stepnumbers
- numberlist
- number_deleted
- todelete
- deletestepcombo
- newstep
- deletestep
- manualbutton
- settingsbutton
- savebutton
- ejectbutton
- loadbutton
- runbutton
- clearbutton
- headinglabel
- twoSwitchValues

6.2.1 Constructor & Destructor Documentation

6.2.1.1 __init__()

```
def gui.AutomaticPage.__init__ (
            self,
            parent,
            controller )
```

6.2.2 Member Function Documentation

6.2.2.1 add_step()

```
def gui.AutomaticPage.add_step (
              self )
```

6.2.2.2 check_channel()

```
def gui.AutomaticPage.check_channel (
              self,
              controller,
              channel,
              frame_object )
```

6.2.2.3 clear_recipe()

```
def gui.AutomaticPage.clear_recipe (
              self )
```

6.2.2.4 create_step()

```
def gui.AutomaticPage.create_step (
              self,
              channels,
              reservoirs )
```

6.2.2.5 delete_step()

```
def gui.AutomaticPage.delete_step (
              self,
              step_number )
```

Note that step number is not an int but a tk.StringVar

6.2.2.6 file_load()

```
def gui.AutomaticPage.file_load (
              self )
```

6.2.2.7 file_save()

```
def gui.AutomaticPage.file_save (
              self )
```

6.2.2.8 run_recipe()

```
def gui.AutomaticPage.run_recipe (
              self,
              controller )
```

Build in abort functionality and progress tracker

6.2.3 Member Data Documentation

6.2.3.1 channels

```
gui.AutomaticPage.channels
```

6.2.3.2 clearbutton

```
gui.AutomaticPage.clearbutton
```

6.2.3.3 deletestep

```
gui.AutomaticPage.deletestep
```

6.2.3.4 deletestepcombo

```
gui.AutomaticPage.deletestepcombo
```

6.2.3.5 ejectbutton

```
gui.AutomaticPage.ejectbutton
```

6.2.3.6 headinglabel

```
gui.AutomaticPage.headinglabel
```

6.2.3.7 loadbutton

```
gui.AutomaticPage.loadbutton
```

6.2.3.8 mainlabel

```
gui.AutomaticPage.mainlabel
```

6.2.3.9 manualbutton

```
gui.AutomaticPage.manualbutton
```

6.2.3.10 newstep

```
gui.AutomaticPage.newstep
```

6.2.3.11 number_deleted

```
gui.AutomaticPage.number_deleted
```

6.2.3.12 numberlist

```
gui.AutomaticPage.numberlist
```

6.2.3.13 reservoirs

gui.AutomaticPage.reservoirs

6.2.3.14 runbutton

gui.AutomaticPage.runbutton

6.2.3.15 savebutton

gui.AutomaticPage.savebutton

6.2.3.16 settingsbutton

gui.AutomaticPage.settingsbutton

6.2.3.17 steplist

gui.AutomaticPage.steplist

6.2.3.18 stepnumbers

gui.AutomaticPage.stepnumbers

6.2.3.19 todelete

gui.AutomaticPage.todelete

6.2.3.20 twoSwitchValues

```
gui.AutomaticPage.twoSwitchValues
```

The documentation for this class was generated from the following file:

- gui.py

6.3 Collection.Collectador Class Reference

Public Member Functions

- def __init__ (self, my_port)
- def serial_connect (self)
- def send (self, cmd)
- def chop_return (self, ret)
- def reset (self)
- def eject (self)
- def next_site (self)
- def last_site (self)
- def moveOneUp (self)
- def moveOneDown (self)
- def moveOneRight (self)
- def moveOneLeft (self)
- def setOrigin (self)
- def toggle_pattern (self)
- def get_info (self)

Public Attributes

- baud
- port
- ser
- serialConnected
- position
- uniqueID
- usingSnakePattern
- currentPattern
- connected

6.3.1 Detailed Description

```
This class is used for control of stepper motors. A serial connection is established with the microcontroller

Attributes:
    baud (int): The baud rate the serial connection is using.
    port (string): the microcontrollers port for the serial connection.
    ser (serial Object): Instance of serial object representing the serial connection.
    serialConnected (bool): True/False of whether the serial connection was established.
    position (int): the position that the XY Stage is currently at.
    uniqueID (string): the unique identifier the microcontroller returns for automatic port selection.
    usingSnakePattern (bool): True/False whether the collection pattern is currently a snake pattern.
    currentPattern (string): string representation of the current pattern.
```

6.3.2 Constructor & Destructor Documentation

6.3.2.1 __init__()

```
def Collection.Collectador.__init__ (
              self,
              my_port )
```

The constructor for the Collectador class.

Parameters:
    my_port (string): the microcontrollers port for the serial connection.

6.3.3 Member Function Documentation

6.3.3.1 chop_return()

```
def Collection.Collectador.chop_return (
              self,
              ret )
```

This method modifies the output of the pump which comes with a carriage return and newline at the end of itsel
The carriage return and newline is cut off here.

Returns:
    Output of the pump without the carraige return and newline at the end.

6.3.3.2 eject()

```
def Collection.Collectador.eject (
              self )
```

This method sends the command "E" to the microcontroller which is programmed to move the XY Stage to the eject
Simulatenously also updates the current position of the XY Stage.

6.3.3.3 get_info()

```
def Collection.Collectador.get_info (
              self )
```

This method sends the command "?" to the microcontroller which is programmed to send back a unique ID.
The attribute uniqueID is set to the microcontrollers response.

Returns:
    response (string): The unique ID that the microcontroller sends back through the serial connection.

6.3.3.4 last_site()

```
def Collection.Collectador.last_site (
              self )
```

This method sends the command "L" to the microcontroller which is programmed to move the XY Stage to the last
Simulatenously also updates the current position of the XY Stage.

6.3.3.5 moveOneDown()

```
def Collection.Collectador.moveOneDown (
              self )
```

This method makes the top motor move one step down.

6.3.3.6 moveOneLeft()

```
def Collection.Collectador.moveOneLeft (
              self )
```

This method makes the bottom motor move one step to the left.

6.3.3.7 moveOneRight()

```
def Collection.Collectador.moveOneRight (
              self )
```

This method makes the bottom motor move one step to the right.

6.3.3.8 moveOneUp()

```
def Collection.Collectador.moveOneUp (
              self )
```

This method makes the top motor move one step up.

6.3.3.9 next_site()

```
def Collection.Collectador.next_site (
              self )
```

This method sends the command "N" to the microcontroller which is programmed to move the XY Stage to the next
Simulatenously also updates the current position of the XY Stage.

6.3.3.10 reset()

```
def Collection.Collectador.reset (
              self )
```

This method sends the command "Z" to the microcontroller which is programmed to move the XY Stage to the origi
Simulatenously also updates the current position of the XY Stage.

6.3.3.11 send()

```
def Collection.Collectador.send (
              self,
              cmd )
```

This method sends a command across the serial connection.

Parameters:
    cmd (string): The command or string that is to be sent to the microcontroller.

6.3.3.12 serial_connect()

```
def Collection.Collectador.serial_connect (
              self )
```

This method establishes the serial connection with the microcontroller.

Once the comport of the pump is known we open a serial connection to it using pySerial.

6.3.3.13 setOrigin()

```
def Collection.Collectador.setOrigin (
              self )
```

This method sets the origin once the user has manually moved it to his liking.
Should be the upper left corner of well plate.

6.3.3.14 toggle_pattern()

```
def Collection.Collectador.toggle_pattern (
              self )
```

This method sends the command "X" to the microcontroller which is programmed to toggle the movement layout.
The two movement patterns currently supported are snake pattern and top down pattern.

6.3.4 Member Data Documentation

6.3.4.1 baud

```
Collection.Collectador.baud
```

6.3.4.2 connected

```
Collection.Collectador.connected
```

6.3.4.3 currentPattern

`Collection.Collectador.currentPattern`

6.3.4.4 port

`Collection.Collectador.port`

6.3.4.5 position

`Collection.Collectador.position`

6.3.4.6 ser

`Collection.Collectador.ser`

6.3.4.7 serialConnected

`Collection.Collectador.serialConnected`

6.3.4.8 uniqueID

`Collection.Collectador.uniqueID`

6.3.4.9 usingSnakePattern

`Collection.Collectador.usingSnakePattern`

The documentation for this class was generated from the following file:

- Collection.py

6.4 gui.ManualPage Class Reference

Inherits Frame.

Public Member Functions

- def __init__ (self, parent, controller)
- def create_channels (self, controller, channels, reservoirs)
- def stop (self, controller, channel)
- def stop_all (self, controller)
- def set_and_start (self, controller, channel)
- def set_and_start_all (self, controller)
- def start_and_collect (self, controller, sampling_rate)
- def next (self, controller)
- def prev (self, controller)
- def reset (self, controller)
- def eject (self, controller)
- def toggle (self, controller)
- def setOrigin (self, controller)
- def moveOneUp (self, controller)
- def moveOneDown (self, controller)
- def moveOneRight (self, controller)
- def moveOneLeft (self, controller)
- def done (self, controller, window)
- def check_all_manual_and_sampling (self, controller)
- def check_all_manual (self, controller)
- def check_channel (self, controller, channel)

Public Attributes

- channels
- reservoirs
- samplingrate
- sites
- autopagebutton
- settingsbutton
- startallbutton
- startandcollectbutton
- stopallbutton
- nextwellbutton
- lastwellbutton
- resetbutton
- ejectbutton
- togglebutton
- setOriginButton
- samplingrateentry
- sitesentry
- mainlabel
- autocollectlabel
- reservoirlabel
- sampleslabel
- CollectionControlLabel

- positionLabel
- patternLabel
- reservoircombobox
- greaterlabelframe
- channeltextlist
- labelframelist
- flowrateentrylist
- startbuttonlist
- stopbuttonlist
- reservoirvalues
- toplabels
- twoSwitchComboBoxList
- twoSwitchValues

6.4.1 Constructor & Destructor Documentation

6.4.1.1 __init__()

```
def gui.ManualPage.__init__ (
              self,
              parent,
              controller )
```

6.4.2 Member Function Documentation

6.4.2.1 check_all_manual()

```
def gui.ManualPage.check_all_manual (
              self,
              controller )
```

use when running all channels together

6.4.2.2 check_all_manual_and_sampling()

```
def gui.ManualPage.check_all_manual_and_sampling (
              self,
              controller )
```

use when running all channels together

6.4.2.3 check_channel()

```
def gui.ManualPage.check_channel (
              self,
              controller,
              channel )
```

6.4.2.4 create_channels()

```
def gui.ManualPage.create_channels (
              self,
              controller,
              channels,
              reservoirs )
```

creates each channel for manual page. properties of channels are stored in lists in self. channels and reservoirs variables refer to res and ch number

6.4.2.5 done()

```
def gui.ManualPage.done (
              self,
              controller,
              window )
```

6.4.2.6 eject()

```
def gui.ManualPage.eject (
              self,
              controller )
```

6.4.2.7 moveOneDown()

```
def gui.ManualPage.moveOneDown (
              self,
              controller )
```

6.4.2.8 moveOneLeft()

```
def gui.ManualPage.moveOneLeft (
              self,
              controller )
```

6.4.2.9 moveOneRight()

```
def gui.ManualPage.moveOneRight (
              self,
              controller )
```

6.4.2.10 moveOneUp()

```
def gui.ManualPage.moveOneUp (
              self,
              controller )
```

6.4.2.11 next()

```
def gui.ManualPage.next (
              self,
              controller )
```

6.4.2.12 prev()

```
def gui.ManualPage.prev (
              self,
              controller )
```

6.4.2.13 reset()

```
def gui.ManualPage.reset (
              self,
              controller )
```

6.4.2.14 set_and_start()

```
def gui.ManualPage.set_and_start (
              self,
              controller,
              channel )
```

sends input direction, flowrate, and reservoir to pump
starts specified pump channel

6.4.2.15 set_and_start_all()

```
def gui.ManualPage.set_and_start_all (
              self,
              controller )
```

sets all of the channels and then starts pump

6.4.2.16 setOrigin()

```
def gui.ManualPage.setOrigin (
              self,
              controller )
```

6.4.2.17 start_and_collect()

```
def gui.ManualPage.start_and_collect (
              self,
              controller,
              sampling_rate )
```

6.4.2.18 stop()

```
def gui.ManualPage.stop (
              self,
              controller,
              channel )
```

stops pump channel

6.4.2.19 stop_all()

```
def gui.ManualPage.stop_all (
              self,
              controller )
```

stops pump channel

6.4.2.20 toggle()

```
def gui.ManualPage.toggle (
              self,
              controller )
```

6.4.3 Member Data Documentation

6.4.3.1 autocollectlabel

gui.ManualPage.autocollectlabel

6.4.3.2 autopagebutton

gui.ManualPage.autopagebutton

6.4.3.3 channels

gui.ManualPage.channels

6.4.3.4 channeltextlist

gui.ManualPage.channeltextlist

6.4.3.5 CollectionControlLabel

gui.ManualPage.CollectionControlLabel

6.4.3.6 ejectbutton

gui.ManualPage.ejectbutton

6.4.3.7 flowrateentrylist

gui.ManualPage.flowrateentrylist

6.4.3.8 greaterlabelframe

gui.ManualPage.greaterlabelframe

6.4.3.9 labelframelist

gui.ManualPage.labelframelist

6.4.3.10 lastwellbutton

gui.ManualPage.lastwellbutton

6.4.3.11 mainlabel

gui.ManualPage.mainlabel

6.4.3.12 nextwellbutton

gui.ManualPage.nextwellbutton

6.4.3.13 patternLabel

gui.ManualPage.patternLabel

6.4.3.14 positionLabel

gui.ManualPage.positionLabel

6.4.3.15 reservoircombobox

gui.ManualPage.reservoircombobox

6.4.3.16 reservoirlabel

gui.ManualPage.reservoirlabel

6.4.3.17 reservoirs

gui.ManualPage.reservoirs

6.4.3.18 reservoirvalues

gui.ManualPage.reservoirvalues

6.4.3.19 resetbutton

gui.ManualPage.resetbutton

6.4.3.20 sampleslabel

gui.ManualPage.sampleslabel

6.4.3.21 samplingrate

gui.ManualPage.samplingrate

6.4.3.22 samplingrateentry

gui.ManualPage.samplingrateentry

6.4.3.23 setOriginButton

gui.ManualPage.setOriginButton

6.4.3.24 settingsbutton

gui.ManualPage.settingsbutton

6.4.3.25 sites

gui.ManualPage.sites

6.4.3.26 sitesentry

gui.ManualPage.sitesentry

6.4.3.27 startallbutton

gui.ManualPage.startallbutton

6.4.3.28 startandcollectbutton

gui.ManualPage.startandcollectbutton

6.4.3.29 startbuttonlist

```
gui.ManualPage.startbuttonlist
```

6.4.3.30 stopallbutton

```
gui.ManualPage.stopallbutton
```

6.4.3.31 stopbuttonlist

```
gui.ManualPage.stopbuttonlist
```

6.4.3.32 togglebutton

```
gui.ManualPage.togglebutton
```

6.4.3.33 toplabels

```
gui.ManualPage.toplabels
```

6.4.3.34 twoSwitchComboBoxList

```
gui.ManualPage.twoSwitchComboBoxList
```

6.4.3.35 twoSwitchValues

```
gui.ManualPage.twoSwitchValues
```

The documentation for this class was generated from the following file:

- gui.py

6.5 Mswitch.MSwitch Class Reference

Public Member Functions

- def __init__ (self, my_port)
- def serial_connect (self)
- def send (self, cmd)
- def chop_return (self, ret)
- def set_reservoir (self, res)
- def get_info (self)

Public Attributes

- baud
- port
- ser
- serialConnected
- res
- uniqueID
- connected

6.5.1 Detailed Description

```
This class is used for control of the Manifold Switch. A serial connection is established with the microcontro

Attributes:
    baud (int): The baud rate the serial connection is using.
    port (string): the microcontrollers port for the serial connection.
    ser (serial Object): Instance of serial object representing the serial connection.
    serialConnected (bool): True/False of whether the serial connection was established.
    res (int): The reservoir chosen.
    uniqueID (string): the unique identifier the microcontroller returns for automatic port selection.
```

6.5.2 Constructor & Destructor Documentation

6.5.2.1 __init__()

```
def Mswitch.MSwitch.__init__ (
              self,
              my_port )
```

```
The constructor for the MSwitch class.

Parameters:
    my_port (string): the microcontrollers port for the serial connection.
```

6.5.3 Member Function Documentation

6.5.3.1 chop_return()

```
def Mswitch.MSwitch.chop_return (
              self,
              ret )
```

This method modifies the output of the pump which comes with a carriage return and newline at the end of itsel
The carriage return and newline is cut off here.

Returns:
    Output of the pump without the carraige return and newline at the end.

6.5.3.2 get_info()

```
def Mswitch.MSwitch.get_info (
              self )
```

This method sends the command "?" to the microcontroller which is programmed to send back a unique ID. The att

Returns:
    response (string): The unique ID that the microcontroller sends back through the serial connection.

6.5.3.3 send()

```
def Mswitch.MSwitch.send (
              self,
              cmd )
```

This method sends a command across the serial connection.

Parameters:
    cmd (string): The command or string that is to be sent to the microcontroller.

6.5.3.4 serial_connect()

```
def Mswitch.MSwitch.serial_connect (
              self )
```

This method establishes the serial connection with the microcontroller.

Once the comport of the pump is known we open a serial connection to it using pySerial.

6.5.3.5 set_reservoir()

```
def Mswitch.MSwitch.set_reservoir (
              self,
              res )
```

This method sets the reservoir to pull fluid from.

6.5.4 Member Data Documentation

6.5.4.1 baud

Mswitch.MSwitch.baud

6.5.4.2 connected

Mswitch.MSwitch.connected

6.5.4.3 port

Mswitch.MSwitch.port

6.5.4.4 res

Mswitch.MSwitch.res

6.5.4.5 ser

Mswitch.MSwitch.ser

6.5.4.6 serialConnected

```
Mswitch.MSwitch.serialConnected
```

6.5.4.7 uniqueID

```
Mswitch.MSwitch.uniqueID
```

The documentation for this class was generated from the following file:

- Mswitch.py

6.6 fakeSerial.Serial Class Reference

Public Member Functions

- def __init__ (self, port='COM1', baudrate=19200, timeout=1, bytesize=8, parity='N', stopbits=1, xonxoff=0, rtscts=0)
- def isOpen (self)
- def open (self)
- def close (self)
- def write (self, string)
- def read (self, n=1)
- def flush (self)
- def flushInput (self)
- def flushOutput (self)
- def readline (self)
- def __str__ (self)

Public Attributes

- name
- port
- timeout
- parity
- baudrate
- bytesize
- stopbits
- xonxoff
- rtscts

6.6.1 Detailed Description

```
This class is used for simulation of an Arduino Serial Connection.

Attributes:
    name (string): the name of the port.
    port (string): the microcontrollers port for the serial connection.
    baudrate (int): The baud rate the serial connection is using.
    timeout (int): maximum milliseconds to wait for serial data.
    parity (string): parity bit type.
    stopbits (int): amount of stop bits.
    xonxoff (int): whether xonxoff is on or off for flow control.
    rtscts (int): whether rtscts flow control is on or off.
```

6.6.2 Constructor & Destructor Documentation

6.6.2.1 __init__()

```
def fakeSerial.Serial.__init__ (
              self,
              port = 'COM1',
              baudrate = 19200,
              timeout = 1,
              bytesize = 8,
              parity = 'N',
              stopbits = 1,
              xonxoff = 0,
              rtscts = 0 )

The constructor for the Serial class.

Parameters:
    port (string): the microcontrollers port for the serial connection. Defaulted to 'COM1'.
    baudrate (int): The baud rate the serial connection is using. Defaulted to 19200.
    timeout (int): maximum milliseconds to wait for serial data. Defaulted to 1.
    bytesize (int): The size of bytes. Defaulted to 8.
    parity (string): parity bit type. Defaulted to 'N'.
    stopbits (int): amount of stop bits. Defaulted to 1.
    xonxoff (int): whether xonxoff is on or off for flow control. Defaulted to 0.
    rtscts (int): whether rtscts flow control is on or off. Defaulted to 0
```

6.6.3 Member Function Documentation

6.6.3.1 __str__()

```
def fakeSerial.Serial.__str__ (
              self )

This method creates a string representation of the Serial class.

Returns:
    string: String representation of Serial class.
```

6.6.3.2 close()

```
def fakeSerial.Serial.close (
              self )
```

This method closes the port artificially.

6.6.3.3 flush()

```
def fakeSerial.Serial.flush (
              self )
```

This method flushes the port artificially.

6.6.3.4 flushInput()

```
def fakeSerial.Serial.flushInput (
              self )
```

This method flushes the input of the port artificially.

6.6.3.5 flushOutput()

```
def fakeSerial.Serial.flushOutput (
              self )
```

This method flushes the output of the port artificially.

6.6.3.6 isOpen()

```
def fakeSerial.Serial.isOpen (
              self )
```

This method returns True if the port to the arduino is open. False otherwise.

```
Returns:
    _isOpen (bool): True/False depending if serial connection is open.
```

6.6.3.7 open()

```
def fakeSerial.Serial.open (
              self )
```

This method opens the port artificially.

6.6.3.8 read()

```
def fakeSerial.Serial.read (
              self,
              n = 1 )
```

This method simulates reading a pretermined amount of characters from the Arduino.

Parameters:
    n (int): number of characters to read.

Returns:
    s (string): the characters read from the Arduino.

6.6.3.9 readline()

```
def fakeSerial.Serial.readline (
              self )
```

This method simulates reading an entire line of characters until a "\n" is encountered.

Returns:
    s (string): The line of characters read if there is one. Otherwise, an empty string.

6.6.3.10 write()

```
def fakeSerial.Serial.write (
              self,
              string )
```

This method simulates writing a string of characters to the Arduino.

Returns:
    An empty string.

6.6.4 Member Data Documentation

6.6.4.1 baudrate

`fakeSerial.Serial.baudrate`

6.6.4.2 bytesize

`fakeSerial.Serial.bytesize`

6.6.4.3 name

`fakeSerial.Serial.name`

6.6.4.4 parity

`fakeSerial.Serial.parity`

6.6.4.5 port

`fakeSerial.Serial.port`

6.6.4.6 rtscts

`fakeSerial.Serial.rtscts`

6.6.4.7 stopbits

`fakeSerial.Serial.stopbits`

6.6.4.8 timeout

`fakeSerial.Serial.timeout`

6.6.4.9 xonxoff

`fakeSerial.Serial.xonxoff`

The documentation for this class was generated from the following file:

- fakeSerial.py

6.7 gui.SettingsPage Class Reference

Inherits Frame.

Public Member Functions

- def __init__ (self, parent, controller)
- def update (self, controller)
- def calibrate (self, controller)

Public Attributes

- channels
- reservoirs
- mainlabel
- autopagebutton
- manualbutton
- updatebutton
- calibratecombobox
- channelvalues
- calibratebutton
- measuredVolume
- measuredVolumeEntry
- entercalibratedbutton
- calibrationlabel
- channelcaliblabel
- directionheading
- directionslist
- directionlabellist
- directions
- diameterlist
- diameterlabel
- diametervalues

6.7.1 Detailed Description

```
Everything that is being sent and all experimental parameters
should be recorded here. Start and stop events. Serial commands. Status.
```

6.7.2 Constructor & Destructor Documentation

6.7.2.1 __init__()

```
def gui.SettingsPage.__init__ (
              self,
              parent,
              controller )
```

6.7.3 Member Function Documentation

6.7.3.1 calibrate()

```
def gui.SettingsPage.calibrate (
              self,
              controller )
```

6.7.3.2 update()

```
def gui.SettingsPage.update (
              self,
              controller )
```

6.7.4 Member Data Documentation

6.7.4.1 autopagebutton

```
gui.SettingsPage.autopagebutton
```

6.7.4.2 calibratebutton

gui.SettingsPage.calibratebutton

6.7.4.3 calibratecombobox

gui.SettingsPage.calibratecombobox

6.7.4.4 calibrationlabel

gui.SettingsPage.calibrationlabel

6.7.4.5 channelcaliblabel

gui.SettingsPage.channelcaliblabel

6.7.4.6 channels

gui.SettingsPage.channels

6.7.4.7 channelvalues

gui.SettingsPage.channelvalues

6.7.4.8 diameterlabel

gui.SettingsPage.diameterlabel

6.7.4.9 diameterlist

gui.SettingsPage.diameterlist

6.7.4.10 diametervalues

`gui.SettingsPage.diametervalues`

6.7.4.11 directionheading

`gui.SettingsPage.directionheading`

6.7.4.12 directionlabellist

`gui.SettingsPage.directionlabellist`

6.7.4.13 directions

`gui.SettingsPage.directions`

6.7.4.14 directionslist

`gui.SettingsPage.directionslist`

6.7.4.15 entercalibratedbutton

`gui.SettingsPage.entercalibratedbutton`

6.7.4.16 mainlabel

`gui.SettingsPage.mainlabel`

6.7.4.17 manualbutton

`gui.SettingsPage.manualbutton`

6.7.4.18 measuredVolume

```
gui.SettingsPage.measuredVolume
```

6.7.4.19 measuredVolumeEntry

```
gui.SettingsPage.measuredVolumeEntry
```

6.7.4.20 reservoirs

```
gui.SettingsPage.reservoirs
```

6.7.4.21 updatebutton

```
gui.SettingsPage.updatebutton
```

The documentation for this class was generated from the following file:

- gui.py

6.8 Pump.ThreePump Class Reference

Public Member Functions

- def __init__ (self, my_port)
- def serial_connect (self)
- def send (self, cmd)
- def send_return (self, cmd)
- def chop_return (self, ret)
- def FormatVolume (self, V, unit)
- def setFlow (self, channel, flowrate)
- def setDir (self, channel, direction)
- def start (self, channel)
- def stop (self, channel)
- def calibrate (self, channel)
- def abort_calibration (self, channel)
- def setTargetCalibrationVolume (self)
- def setMeasuredVolume (self, channel, volume)
- def start_all (self)
- def stop_all (self)
- def setDefaults (self)

Public Attributes

- baud
- port
- ser
- serialConnected
- calibrationvolume
- calibrationunit
- isOn
- connected

6.8.1 Detailed Description

```
This class is used for control of the Pump. A serial connection is established with the microcontroller contro

Attributes:
    baud (int): The baud rate the serial connection is using.
    port (string): the microcontrollers port for the serial connection.
    ser (serial Object): Instance of serial object representing the serial connection.
    serialConnected (bool): True/False of whether the serial connection was established.
    calibrationvolume (int): amount of fluid used for calibration.
    calibrationunit (string): the unit for the amount of fluid used for calibration.
    isOn (bool): True/False on whether the pump is on or not.
```

6.8.2 Constructor & Destructor Documentation

6.8.2.1 __init__()

```
def Pump.ThreePump.__init__ (
              self,
              my_port )
```

```
The constructor for the TwoSwitch class.

Parameters:
    my_port (string): the microcontrollers port for the serial connection.
```

6.8.3 Member Function Documentation

6.8.3.1 abort_calibration()

```
def Pump.ThreePump.abort_calibration (
              self,
              channel )
```

```
This method sends a command to the microcontroller which is programmed to abort calibration of the pump only f

Parameters:
    channel (int): the channel that will abort calibration.
```

6.8.3.2 calibrate()

```
def Pump.ThreePump.calibrate (
              self,
              channel )
```

This method sends a command to the microcontroller which is programmed to calibrate the pump only for the chan

Parameters:
    channel (int): the channel that will be calibrated.

6.8.3.3 chop_return()

```
def Pump.ThreePump.chop_return (
              self,
              ret )
```

This method modifies the output of the two switch which comes with a carriage return and newline at the end of
The carriage return and newline is cut off here.

Returns:
    Output of the pump without the carraige return and newline at the end.

6.8.3.4 FormatVolume()

```
def Pump.ThreePump.FormatVolume (
              self,
              V,
              unit )
```

This method formats the volume.

Parameters:
    V (int): The amount of volume.
    unit (string): The unit the volume should be set in.

6.8.3.5 send()

```
def Pump.ThreePump.send (
              self,
              cmd )
```

This method sends a command across the serial connection.

Parameters:
    cmd (string): The command or string that is to be sent to the microcontroller.

6.8.3.6 send_return()

```
def Pump.ThreePump.send_return (
              self,
              cmd )
```

This method uses the serial connection opened instance of pump and sends the text written in cmd across that c

Returns:
    response (string): The response or what the microcontroller returns.

6.8.3.7 serial_connect()

```
def Pump.ThreePump.serial_connect (
              self )
```

This method establishes the serial connection with the microcontroller.

Once the comport of the pump is known we open a serial connection to it using pySerial.

6.8.3.8 setDefaults()

```
def Pump.ThreePump.setDefaults (
              self )
```

This method sets the deafults for the pump.

6.8.3.9 setDir()

```
def Pump.ThreePump.setDir (
              self,
              channel,
              direction )
```

This method sets the direction for the flow of fluid.

Parameters:
    channel (int): The channel flow rate direction will be set.
    direction (string): The direction for the pump CW or CCW.

6.8.3.10 setFlow()

```
def Pump.ThreePump.setFlow (
              self,
              channel,
              flowrate )
```

This method sets the flow rate of the pump.
Parameters:
    channel (int): The channel whose flow rate will be set.
    flowrate (float): The flow rate which is to be set.

6.8.3.11 setMeasuredVolume()

```
def Pump.ThreePump.setMeasuredVolume (
              self,
              channel,
              volume )
```

This method sets the measured volume.

6.8.3.12 setTargetCalibrationVolume()

```
def Pump.ThreePump.setTargetCalibrationVolume (
              self )
```

This method sets the target calibration volume.

6.8.3.13 start()

```
def Pump.ThreePump.start (
              self,
              channel )
```

This method sends a command to the microcontroller which is programmed to start the pump only for the channel

Parameters:
    channel (int): the channel that will start.

6.8.3.14 start_all()

```
def Pump.ThreePump.start_all (
              self )
```

This method sends a command to the microcontroller which is programmed to start the pump for all channels.

6.8.3.15 stop()

```
def Pump.ThreePump.stop (
              self,
              channel )
```

This method sends a command to the microcontroller which is programmed to stop the pump only for the channel c

```
Parameters:
    channel (int): the channel that will stop.
```

6.8.3.16 stop_all()

```
def Pump.ThreePump.stop_all (
              self )
```

This method sends a command to the microcontroller which is programmed to stop the pump for all the channels.

6.8.4 Member Data Documentation

6.8.4.1 baud

```
Pump.ThreePump.baud
```

6.8.4.2 calibrationunit

```
Pump.ThreePump.calibrationunit
```

6.8.4.3 calibrationvolume

```
Pump.ThreePump.calibrationvolume
```

6.8.4.4 connected

```
Pump.ThreePump.connected
```

6.8.4.5 isOn

```
Pump.ThreePump.isOn
```

6.8.4.6 port

```
Pump.ThreePump.port
```

6.8.4.7 ser

```
Pump.ThreePump.ser
```

6.8.4.8 serialConnected

```
Pump.ThreePump.serialConnected
```

The documentation for this class was generated from the following file:

- Pump.py

6.9 TwoSwitch.TwoSwitch Class Reference

Public Member Functions

- def __init__ (self, my_port)
- def serial_connect (self)
- def send (self, cmd)
- def chop_return (self, ret)
- def setRecirculate (self, channel)
- def setCollect (self, channel)
- def getIdentifier (self)

Public Attributes

- baud
- port
- ser
- serialConnected
- uniqueID
- willRecirculate
- connected

6.9.1 Detailed Description

```
This class is used for control of the two switches. A serial connection is established with the microcontrolle

Attributes:
    baud (int): The baud rate the serial connection is using.
    port (string): the microcontrollers port for the serial connection.
    ser (serial Object): Instance of serial object representing the serial connection.
    serialConnected (bool): True/False of whether the serial connection was established.
    uniqueID (string): the unique identifier the microcontroller returns for automatic port selection.
    willRecirculate (bool): True/False as to if the two switches are set to recirculate.
```

6.9.2 Constructor & Destructor Documentation

6.9.2.1 __init__()

```
def TwoSwitch.TwoSwitch.__init__ (
              self,
              my_port )

The constructor for the TwoSwitch class.

Parameters:
    my_port (string): the microcontrollers port for the serial connection.
```

6.9.3 Member Function Documentation

6.9.3.1 chop_return()

```
def TwoSwitch.TwoSwitch.chop_return (
              self,
              ret )

This method modifies the output of the two switch which comes with a carriage return and newline at the end of
The carriage return and newline is cut off here.

Returns:
    Output of the pump without the carraige return and newline at the end.
```

6.9.3.2 getIdentifier()

```
def TwoSwitch.TwoSwitch.getIdentifier (
              self )
```

This method sends the command "?" to the microcontroller which is programmed to send back a unique ID. The att

Returns:
    response (string): The unique ID that the microcontroller sends back through the serial connection.

6.9.3.3 send()

```
def TwoSwitch.TwoSwitch.send (
              self,
              cmd )
```

This method sends a command across the serial connection.

Parameters:
    cmd (string): The command or string that is to be sent to the microcontroller.

6.9.3.4 serial_connect()

```
def TwoSwitch.TwoSwitch.serial_connect (
              self )
```

This method establishes the serial connection with the microcontroller.

Once the comport of the pump is known we open a serial connection to it using pySerial.

6.9.3.5 setCollect()

```
def TwoSwitch.TwoSwitch.setCollect (
              self,
              channel )
```

This method sends a command to the microcontroller based on what channel needs to be set to recirculate.

Parameters:
    channel (int): The fluidic channel chosen to be set to collect.

6.9.3.6 setRecirculate()

```
def TwoSwitch.TwoSwitch.setRecirculate (
              self,
              channel )
```

This method sends a command to the microcontroller based on what channel needs to be set to recirculate.

Parameters:
    channel (int): The fluidic channel chosen to be set to recirculate.

6.9.4 Member Data Documentation

6.9.4.1 baud

TwoSwitch.TwoSwitch.baud

6.9.4.2 connected

TwoSwitch.TwoSwitch.connected

6.9.4.3 port

TwoSwitch.TwoSwitch.port

6.9.4.4 ser

TwoSwitch.TwoSwitch.ser

6.9.4.5 serialConnected

TwoSwitch.TwoSwitch.serialConnected

6.9.4.6 uniqueID

```
TwoSwitch.TwoSwitch.uniqueID
```

6.9.4.7 willRecirculate

```
TwoSwitch.TwoSwitch.willRecirculate
```

The documentation for this class was generated from the following file:

- TwoSwitch.py

6.10 gui.WelcomePage Class Reference

Inherits Frame.

Public Member Functions

- def __init__ (self, parent, controller)
- def flip (self, widget)

Public Attributes

- mainlabel
- sublabelframe
- perfusorframe
- collectorframe
- manilabel

  Port & Device Selection############################## finding all of the ports.
- pumplabel
- coll_label
- donebutton

6.10.1 Constructor & Destructor Documentation

6.10.1.1 __init__()

```
def gui.WelcomePage.__init__ (
              self,
              parent,
              controller )
```

6.10.2 Member Function Documentation

6.10.2.1 flip()

```
def gui.WelcomePage.flip (
              self,
              widget )
```

Enables or disables a widget. It is run by the checkbuttons on the WelcomPage

6.10.3 Member Data Documentation

6.10.3.1 coll_label

```
gui.WelcomePage.coll_label
```

6.10.3.2 collectorframe

```
gui.WelcomePage.collectorframe
```

6.10.3.3 donebutton

```
gui.WelcomePage.donebutton
```

6.10.3.4 mainlabel

```
gui.WelcomePage.mainlabel
```

6.10.3.5 manilabel

```
gui.WelcomePage.manilabel
```

Port & Device Selection############################ finding all of the ports.

6.10.3.6 perfusorframe

`gui.WelcomePage.perfusorframe`

6.10.3.7 pumplabel

`gui.WelcomePage.pumplabel`

6.10.3.8 sublabelframe

`gui.WelcomePage.sublabelframe`

The documentation for this class was generated from the following file:

- gui.py

Chapter 7

File Documentation

7.1 ArduinoSnakePattern.cpp File Reference

Arduino code for XYStage Movement and TwoSwitch control.

```
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include <AccelStepper.h>
#include <MultiStepper.h>
#include "utility/Adafruit_MS_PWMServoDriver.h"
```

Functions

- void forwardstep1 ()

  Wrapper Forward Step function for Stepper Motor 1.
- void backwardstep1 ()

  Wrapper Backward Step function for Stepper Motor 1.
- void forwardstep2 ()

  Wrapper Forward Step function for Stepper Motor 2.
- void backwardstep2 ()

  Wrapper Backward Step function for Stepper Motor 2.
- void populateLocations (long(&locations)[numLocations][2])

  This function populates the location matrix with the layout for moving to any location on the well plate.
- void moveGen (Adafruit_StepperMotor *motor, int delay, int myDirection, int stepType)

  Generic Move Function for Stepper Motors.
- void moveRight ()

  MOVE RIGHT FUNCTION.
- void moveLeft ()

  MOVE LEFT FUNCTION.
- void moveDown ()

  MOVE DOWN FUNCTION.
- void moveUp ()

  MOVE UP FUNCTION.
- int moveNext ()

  MOVE NEXT FUNCTION.

- int moveLast ()
  MOVE NEXT FUNCTION.
- int set ()
- int eject ()
  EJECT WELL PLATE FUNCTION.
- void reset ()
  RESET FUNCTION.
- AccelStepper stepperB (forwardstep1, backwardstep1)
- AccelStepper stepperT (forwardstep2, backwardstep2)
- void setup ()
  Arduino Setup.
- void loop ()
  Arduino Loop Function, this runs continously.

Variables

- const int delayRPM = 255
  Imports needed.
- const int rows = 8
  Amount of rows on well plate.
- const int columns = 12
  Amount of columns on well plate.
- const int numLocations = ((rows ∗ columns) / 3) + 1
  Amount of triplet wells on well plate.
- const float maximumSpeed = 255.0
  Max Speed of the Stepper Motors.
- const float acceleration = 2000.0
  Acceleration for the Stepper Motors.
- const int stepsPerRevolution = 200
  Steps per revolution for the Stepper Motor.
- const int stepsPerWell = 225
  Amount of steps needed from center of well to the next.
- const int stepsToEject = 1000
  Amount of steps to eject entire well plate.
- const int limitSwitchT = 37
  Digital Pin Number for Top Stepper Motor Limit Switch.
- const int limitSwitchB = 36
  Digital Pin Number for Bottom Stepper Motor Limit Switch.
- const int TwoSwitchA = 34
  Digital Pin Number for TwoSwitchA.
- const int TwoSwitchB = 33
  Digital Pin Number for TwoSwitchB.
- const int TwoSwitchC = 32
  Digital Pin Number for TwoSwitchC.
- Adafruit_MotorShield AFMS = Adafruit_MotorShield()
- Adafruit_StepperMotor ∗ myMotorT = AFMS.getStepper(stepsPerRevolution, 1)
- Adafruit_StepperMotor ∗ myMotorB = AFMS.getStepper(stepsPerRevolution, 2)
- MultiStepper steppers
- int myPosition = 0
  Position variable to keep track of which triplet well we are on.
- long locations [numLocations][2]
  Three dimensional array for schematic of positions if MultiStepper is used.
- long positions [2]
  Array for movement needed for using both motors in MultiStepper.

7.1.1 Detailed Description

Arduino code for XYStage Movement and TwoSwitch control.

Author

Adiel Hernandez

7.1.2 Function Documentation

7.1.2.1 backwardstep1()

```
void backwardstep1 ( )
```

Wrapper Backward Step function for Stepper Motor 1.

To be used with the AccelStepper Library.

7.1.2.2 backwardstep2()

```
void backwardstep2 ( )
```

Wrapper Backward Step function for Stepper Motor 2.

To be used with the AccelStepper Library.

7.1.2.3 eject()

```
int eject ( )
```

EJECT WELL PLATE FUNCTION.

Moves the motor accordingly so it goes to the eject well plate position. It moves the well plate down from its current location. Sets myPosition to 32.

7.1.2.4 forwardstep1()

```
void forwardstep1 ( )
```

Wrapper Forward Step function for Stepper Motor 1.

To be used with the AccelStepper Library.

7.1.2.5 forwardstep2()

```
void forwardstep2 ( )
```

Wrapper Forward Step function for Stepper Motor 2.

To be used with the AccelStepper Library.

7.1.2.6 loop()

```
void loop ( )
```

Arduino Loop Function, this runs continously.

Essentially this looks for incoming commands which correspond to a certain function for motor movement or Limit Switching.

7.1.2.7 moveDown()

```
void moveDown ( )
```

MOVE DOWN FUNCTION.

Moves the bottom motor to the left. With SINGLE step type.

7.1.2.8 moveGen()

```
void moveGen (
            Adafruit_StepperMotor * motor,
            int delay,
            int myDirection,
            int stepType )
```

Generic Move Function for Stepper Motors.

This function takes in a pointer to the motor that has to be moved, the speed, the direction, and step type. All of this is used to move the correct motor accordingly.

Parameters

∗motor	a Adafruit_StepperMotor object pointer for the motor to be moved.
delay	the speed or RPM to move the motor at.
myDirection	the direction to move the motor in FORWARD or BACKWARD.
stepType	the step type to move the motor in: SINGLE, DOUBLE, INTERLEAVE, or MICROSTEP.

7.1.2.9 moveLast()

```
int moveLast ( )
```

MOVE NEXT FUNCTION.

Moves the motor accordingly so it goes to the previous well plate given it is not on the first possible triplet well location. Moves in a snake pattern. Increments myPosition by 1.

7.1.2.10 moveLeft()

```
void moveLeft ( )
```

MOVE LEFT FUNCTION.

Moves the top motor to the left. With SINGLE step type.

7.1.2.11 moveNext()

```
int moveNext ( )
```

MOVE NEXT FUNCTION.

Moves the motor accordingly so it goes to the next well plate given it is not on the last possible triplet well location. Moves in a snake pattern. Decrements myPosition by 1.

7.1.2.12 moveRight()

```
void moveRight ( )
```

MOVE RIGHT FUNCTION.

Moves the top motor to the right. With SINGLE step type.

7.1.2.13 moveUp()

```
void moveUp ( )
```

MOVE UP FUNCTION.

Moves the bottom motor to the right. With SINGLE step type.

7.1.2.14 populateLocations()

```
void populateLocations (
            long(&) locations[numLocations][2] )
```

This function populates the location matrix with the layout for moving to any location on the well plate.

Not currently being used since we are doing a snake pattern but could be used in future.

7.1.2.15 reset()

```
void reset ( )
```

RESET FUNCTION.

Moves the motor accordingly so it goes to the origin well plate position. Moves the motors simultaneously until both limit switches are activated and then moves a predetermined amount to origin.

7.1.2.16 set()

```
int set ( )
```

7.1.2.17 setup()

```
void setup ( )
```

Arduino Setup.

This only runs once.

7.1.2.18 stepperB()

```
AccelStepper stepperB (
             forwardstep1 ,
             backwardstep1  )
```

7.1.2.19 stepperT()

```
AccelStepper stepperT (
             forwardstep2 ,
             backwardstep2  )
```

7.1.3 Variable Documentation

7.1.3.1 acceleration

```
const float acceleration = 2000.0
```

Acceleration for the Stepper Motors.

7.1.3.2 AFMS

```
Adafruit_MotorShield AFMS = Adafruit_MotorShield()
```

7.1.3.3 columns

```
const int columns = 12
```

Amount of columns on well plate.

7.1.3.4 delayRPM

```
const int delayRPM = 255
```

Imports needed.

The Adafruit_MotorShieldV2 library and AccelStepper Library is used.Global Variables. Constains mostly motor values and digital pin values.RPM for the stepper motors.

7.1.3.5 limitSwitchB

```
const int limitSwitchB = 36
```

Digital Pin Number for Bottom Stepper Motor Limit Switch.

7.1.3.6 limitSwitchT

```
const int limitSwitchT = 37
```

Digital Pin Number for Top Stepper Motor Limit Switch.

7.1.3.7 locations

```
long locations[numLocations][2]
```

Three dimensional array for schematic of positions if MultiStepper is used.

7.1.3.8 maximumSpeed

```
const float maximumSpeed = 255.0
```

Max Speed of the Stepper Motors.

7.1.3.9 myMotorB

```
Adafruit_StepperMotor* myMotorB = AFMS.getStepper(stepsPerRevolution, 2)
```

7.1.3.10 myMotorT

```
Adafruit_StepperMotor* myMotorT = AFMS.getStepper(stepsPerRevolution, 1)
```

7.1.3.11 myPosition

```
int myPosition = 0
```

Position variable to keep track of which triplet well we are on.

7.1.3.12 numLocations

```
const int numLocations = ((rows * columns) / 3) + 1
```

Amount of triplet wells on well plate.

(Plus 1 for the eject location).

7.1.3.13 positions

```
long positions[2]
```

Array for movement needed for using both motors in MultiStepper.

7.1.3.14 rows

```
const int rows = 8
```

Amount of rows on well plate.

7.1.3.15 steppers

```
MultiStepper steppers
```

7.1.3.16 stepsPerRevolution

```
const int stepsPerRevolution = 200
```

Steps per revolution for the Stepper Motor.

7.1.3.17 stepsPerWell

```
const int stepsPerWell = 225
```

Amount of steps needed from center of well to the next.

7.1.3.18 stepsToEject

```
const int stepsToEject = 1000
```

Amount of steps to eject entire well plate.

7.1.3.19 TwoSwitchA

```
const int TwoSwitchA = 34
```

Digital Pin Number for TwoSwitchA.

7.1.3.20 TwoSwitchB

```
const int TwoSwitchB = 33
```

Digital Pin Number for TwoSwitchB.

7.1.3.21 TwoSwitchC

```
const int TwoSwitchC = 32
```

Digital Pin Number for TwoSwitchC.

7.2 Collection.py File Reference

Classes

- class Collection.Collectador

Namespaces

- Collection

7.3 fakeSerial.py File Reference

Classes

- class fakeSerial.Serial

Namespaces

- fakeSerial

7.4 gui.py File Reference

Classes

- class gui.App
- class gui.WelcomePage
- class gui.ManualPage
- class gui.AutomaticPage
- class gui.SettingsPage

Namespaces

- gui

Functions

- def gui.combine_funcs (*funcs)
- def gui.message_prompt (message)
- def gui.check_input (variable, desired_datatype, lower_limit=0, upper_limit=0)

Variables

- tuple gui.LARGE_FONT = ("Verdana", 10)
- gui.app = App()

7.5 locationPopulate.cpp File Reference

```
#include <iostream>
```

Functions

- void populateLocations (long(&locations)[numLocations][2], int choice=1)

  This function populates the location matrix with the layout for moving to any location on the well plate.
- int main ()

Variables

- const int rows = 8
- const int columns = 12
- const int numLocations = ((rows * columns) / 3) + 1
- const int stepsPerWell = 225
- const int stepsToEject = 500

7.5.1 Function Documentation

7.5.1.1 main()

```
int main ( )
```

7.5.1.2 populateLocations()

```
void populateLocations (
            long(&) locations[numLocations][2],
            int choice = 1 )
```

This function populates the location matrix with the layout for moving to any location on the well plate.

Gives the user the choice for a snake pattern Choice = 1, or top to bottom pattern Choice != 1.

7.5.2 Variable Documentation

7.5.2.1 columns

```
const int columns = 12
```

7.5.2.2 numLocations

```
const int numLocations = ((rows * columns) / 3) + 1
```

7.5.2.3 rows

```
const int rows = 8
```

7.5.2.4 stepsPerWell

```
const int stepsPerWell = 225
```

7.5.2.5 stepsToEject

```
const int stepsToEject = 500
```

7.6 Mswitch.py File Reference

Classes

- class Mswitch.MSwitch

Namespaces

- Mswitch

7.7 Pump.py File Reference

Classes

- class Pump.ThreePump

Namespaces

- Pump

Functions

- def Pump.get_ports ()

7.8 TwoSwitch.py File Reference

Classes

- class TwoSwitch.TwoSwitch

Namespaces

- TwoSwitch

7.9 XYStage_LimitSwitch_TwoSwitch_Control.cpp File Reference

```
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include <AccelStepper.h>
#include <MultiStepper.h>
#include "utility/Adafruit_MS_PWMServoDriver.h"
```

Functions

- void forwardstep1 ()

  Wrapper Forward Step function for Stepper Motor 1.
- void backwardstep1 ()

  Wrapper Backward Step function for Stepper Motor 1.
- void forwardstep2 ()

  Wrapper Forward Step function for Stepper Motor 2.
- void backwardstep2 ()

  Wrapper Backward Step function for Stepper Motor 2.
- void populateLocations (long(&locations)[numLocations][2], int choice=1)

  This function populates the location matrix with the layout for moving to any location on the well plate.
- void moveGen (Adafruit_StepperMotor *motor, int delay, int myDirection, int stepType)

  Generic Move Function for Stepper Motors.
- void moveRight ()

  MOVE RIGHT FUNCTION.
- void moveLeft ()

  MOVE LEFT FUNCTION.
- void moveDown ()

  MOVE DOWN FUNCTION.
- void moveUp ()

  MOVE UP FUNCTION.
- int moveNext ()

  MOVE NEXT FUNCTION.
- int moveLast ()

  MOVE NEXT FUNCTION.
- int set ()
- int eject ()

  EJECT WELL PLATE FUNCTION.
- void reset ()

  RESET FUNCTION.
- AccelStepper stepperB (forwardstep1, backwardstep1)
- AccelStepper stepperT (forwardstep2, backwardstep2)
- void setup ()
- void loop ()

  Arduino Loop Function, this runs continously.

Variables

- const int delayRPM = 255

  Imports needed.
- const int rows = 8

  Amount of rows on well plate.
- const int columns = 12

  Amount of columns on well plate.
- const int numLocations = ((rows * columns) / 3) + 1

  Amount of triplet wells on well plate.
- const float maximumSpeed = 255.0

  Max Speed of the Stepper Motors.
- const float acceleration = 2000.0

Acceleration for the Stepper Motors.

- const int stepsPerRevolution = 200

  Steps per revolution for the Stepper Motor.
- const int stepsPerWell = 225

  Amount of steps needed from center of well to the next.
- const int stepsToEject = 1000

  Amount of steps to eject entire well plate.
- const int limitSwitchT = 41

  Digital Pin Number for Top Stepper Motor Limit Switch.
- const int limitSwitchB = 43

  Digital Pin Number for Bottom Stepper Motor Limit Switch.
- const int TwoSwitchA = 35

  Digital Pin Number for TwoSwitchA.
- const int TwoSwitchB = 33

  Digital Pin Number for TwoSwitchB.
- const int TwoSwitchC = 31

  Digital Pin Number for TwoSwitchC.
- Adafruit_MotorShield AFMS = Adafruit_MotorShield()
- Adafruit_StepperMotor * myMotorT = AFMS.getStepper(stepsPerRevolution, 1)
- Adafruit_StepperMotor * myMotorB = AFMS.getStepper(stepsPerRevolution, 2)
- MultiStepper steppers
- int myPosition = 0

  Position variable to keep track of which triplet well we are on.
- long locations [numLocations][2]

  Three dimensional array for schematic of positions if MultiStepper is used.
- long positions [2]

  Array for movement needed for using both motors in MultiStepper.
- int collectionPattern = 1

  Collection Pattern variable to choose between snake pattern XY Stage movement (default = 1) or Top down XY Stage movement (!= 1).

7.9.1 Function Documentation

7.9.1.1 backwardstep1()

```
void backwardstep1 ( )
```

Wrapper Backward Step function for Stepper Motor 1.

To be used with the AccelStepper Library.

7.9.1.2 backwardstep2()

```
void backwardstep2 ( )
```

Wrapper Backward Step function for Stepper Motor 2.

To be used with the AccelStepper Library.

7.9.1.3 eject()

```
int eject ( )
```

EJECT WELL PLATE FUNCTION.

Moves the motor accordingly so it goes to the eject well plate position. It moves the well plate down from its current location. Sets myPosition to 32.

7.9.1.4 forwardstep1()

```
void forwardstep1 ( )
```

Wrapper Forward Step function for Stepper Motor 1.

To be used with the AccelStepper Library.

7.9.1.5 forwardstep2()

```
void forwardstep2 ( )
```

Wrapper Forward Step function for Stepper Motor 2.

To be used with the AccelStepper Library.

7.9.1.6 loop()

```
void loop ( )
```

Arduino Loop Function, this runs continously.

Essentially this looks for incoming commands which correspond to a certain function for motor movement or Limit Switching.

7.9.1.7 moveDown()

```
void moveDown ( )
```

MOVE DOWN FUNCTION.

Moves the bottom motor to the left. With SINGLE step type.

7.9.1.8 moveGen()

```
void moveGen (
            Adafruit_StepperMotor * motor,
            int delay,
            int myDirection,
            int stepType )
```

Generic Move Function for Stepper Motors.

This function takes in a pointer to the motor that has to be moved, the speed, the direction, and step type. All of this is used to move the correct motor accordingly.

Parameters

∗motor	a Adafruit_StepperMotor object pointer for the motor to be moved.
delay	the speed or RPM to move the motor at.
myDirection	the direction to move the motor in FORWARD or BACKWARD.
stepType	the step type to move the motor in: SINGLE, DOUBLE, INTERLEAVE, or MICROSTEP.

7.9.1.9 moveLast()

```
int moveLast ( )
```

MOVE NEXT FUNCTION.

Moves the motor accordingly so it goes to the previous well plate given it is not on the first possible triplet well location. Moves in a snake pattern. Increments myPosition by 1.

7.9.1.10 moveLeft()

```
void moveLeft ( )
```

MOVE LEFT FUNCTION.

Moves the top motor to the left. With SINGLE step type.

7.9.1.11 moveNext()

```
int moveNext ( )
```

MOVE NEXT FUNCTION.

Moves the motor accordingly so it goes to the next well plate given it is not on the last possible triplet well location. Moves in a snake pattern. Decrements myPosition by 1.

7.9.1.12 moveRight()

```
void moveRight ( )
```

MOVE RIGHT FUNCTION.

Moves the top motor to the right. With SINGLE step type.

7.9.1.13 moveUp()

```
void moveUp ( )
```

MOVE UP FUNCTION.

Moves the bottom motor to the right. With SINGLE step type.

7.9.1.14 populateLocations()

```
void populateLocations (
            long(&) locations[numLocations][2],
            int choice = 1 )
```

This function populates the location matrix with the layout for moving to any location on the well plate.

Gives the user the choice for a snake pattern Choice = 1, or top to bottom pattern Choice != 1.

7.9.1.15 reset()

```
void reset ( )
```

RESET FUNCTION.

Moves the motor accordingly so it goes to the origin well plate position. Moves the motors simultaneously until both limit switches are activated and then moves a predetermined amount to origin.

7.9.1.16 set()

```
int set ( )
```

7.9.1.17 setup()

```
void setup ( )
```

7.9.1.18 stepperB()

```
AccelStepper stepperB (
            forwardstep1 ,
            backwardstep1  )
```

7.9.1.19 stepperT()

```
AccelStepper stepperT (
            forwardstep2 ,
            backwardstep2  )
```

7.9.2 Variable Documentation

7.9.2.1 acceleration

```
const float acceleration = 2000.0
```

Acceleration for the Stepper Motors.

7.9.2.2 AFMS

```
Adafruit_MotorShield AFMS = Adafruit_MotorShield()
```

7.9.2.3 collectionPattern

```
int collectionPattern = 1
```

Collection Pattern variable to choose between snake pattern XY Stage movement (default = 1) or Top down XY Stage movement (!= 1).

/** Arduino Setup. This only runs once.

7.9.2.4 columns

```
const int columns = 12
```

Amount of columns on well plate.

7.9.2.5 delayRPM

```
const int delayRPM = 255
```

Imports needed.

The Adafruit_MotorShieldV2 library and AccelStepper Library is used.Global Variables. Constains mostly motor values and digital pin values.RPM for the stepper motors.

7.9.2.6 limitSwitchB

```
const int limitSwitchB = 43
```

Digital Pin Number for Bottom Stepper Motor Limit Switch.

7.9.2.7 limitSwitchT

```
const int limitSwitchT = 41
```

Digital Pin Number for Top Stepper Motor Limit Switch.

7.9.2.8 locations

```
long locations[numLocations][2]
```

Three dimensional array for schematic of positions if MultiStepper is used.

7.9.2.9 maximumSpeed

```
const float maximumSpeed = 255.0
```

Max Speed of the Stepper Motors.

7.9.2.10 myMotorB

```
Adafruit_StepperMotor* myMotorB = AFMS.getStepper(stepsPerRevolution, 2)
```

7.9.2.11 myMotorT

```
Adafruit_StepperMotor* myMotorT = AFMS.getStepper(stepsPerRevolution, 1)
```

7.9.2.12 myPosition

```
int myPosition = 0
```

Position variable to keep track of which triplet well we are on.

7.9.2.13 numLocations

```
const int numLocations = ((rows * columns) / 3) + 1
```

Amount of triplet wells on well plate.

(Plus 1 for the eject location).

7.9.2.14 positions

```
long positions[2]
```

Array for movement needed for using both motors in MultiStepper.

7.9.2.15 rows

```
const int rows = 8
```

Amount of rows on well plate.

7.9.2.16 steppers

```
MultiStepper steppers
```

7.9.2.17 stepsPerRevolution

```
const int stepsPerRevolution = 200
```

Steps per revolution for the Stepper Motor.

7.9.2.18 stepsPerWell

```
const int stepsPerWell = 225
```

Amount of steps needed from center of well to the next.

7.9.2.19 stepsToEject

```
const int stepsToEject = 1000
```

Amount of steps to eject entire well plate.

7.9.2.20 TwoSwitchA

```
const int TwoSwitchA = 35
```

Digital Pin Number for TwoSwitchA.

7.9.2.21 TwoSwitchB

```
const int TwoSwitchB = 33
```

Digital Pin Number for TwoSwitchB.

7.9.2.22 TwoSwitchC

```
const int TwoSwitchC = 31
```

Digital Pin Number for TwoSwitchC.

Index

__init__
    Collection.Collectador, 26
    fakeSerial.Serial, 45
    gui.App, 14
    gui.AutomaticPage, 20
    gui.ManualPage, 32
    gui.SettingsPage, 50
    gui.WelcomePage, 63
    Mswitch.MSwitch, 41
    Pump.ThreePump, 54
    TwoSwitch.TwoSwitch, 60
__str__
    fakeSerial.Serial, 45

abort_calibration
    Pump.ThreePump, 54
acceleration
    ArduinoSnakePattern.cpp, 72
    XYStage_LimitSwitch_TwoSwitch_Control.cpp, 84
add_step
    gui.AutomaticPage, 20
AFMS
    ArduinoSnakePattern.cpp, 72
    XYStage_LimitSwitch_TwoSwitch_Control.cpp, 85
app
    gui, 11
appHighlightFont
    gui.App, 16
ArduinoSnakePattern.cpp, 67
    acceleration, 72
    AFMS, 72
    backwardstep1, 69
    backwardstep2, 69
    columns, 73
    delayRPM, 73
    eject, 69
    forwardstep1, 69
    forwardstep2, 69
    limitSwitchB, 73
    limitSwitchT, 73
    locations, 73
    loop, 70
    maximumSpeed, 73
    moveDown, 70
    moveGen, 70
    moveLast, 70
    moveLeft, 71
    moveNext, 71
    moveRight, 71
    moveUp, 71
    myMotorB, 74
    myMotorT, 74
    myPosition, 74
    numLocations, 74
    populateLocations, 71
    positions, 74
    reset, 71
    rows, 74
    set, 72
    setup, 72
    stepperB, 72
    steppers, 75
    stepperT, 72
    stepsPerRevolution, 75
    stepsPerWell, 75
    stepsToEject, 75
    TwoSwitchA, 75
    TwoSwitchB, 75
    TwoSwitchC, 76
autocollectlabel
    gui.ManualPage, 36
autopagebutton
    gui.ManualPage, 36
    gui.SettingsPage, 50

backwardstep1
    ArduinoSnakePattern.cpp, 69
    XYStage_LimitSwitch_TwoSwitch_Control.cpp, 81
backwardstep2
    ArduinoSnakePattern.cpp, 69
    XYStage_LimitSwitch_TwoSwitch_Control.cpp, 81
baud
    Collection.Collectador, 29
    Mswitch.MSwitch, 43
    Pump.ThreePump, 58
    TwoSwitch.TwoSwitch, 62
baudrate
    fakeSerial.Serial, 48
buttonFont
    gui.App, 16
bytesize
    fakeSerial.Serial, 48

calibrate
    gui.SettingsPage, 50
    Pump.ThreePump, 54
calibratebutton
    gui.SettingsPage, 50
calibratecombobox
    gui.SettingsPage, 51

calibrationlabel
gui.SettingsPage, 51
calibrationunit
Pump.ThreePump, 58
calibrationvolume
Pump.ThreePump, 58
channelcaliblabel
gui.SettingsPage, 51
channels
gui.AutomaticPage, 22
gui.ManualPage, 36
gui.SettingsPage, 51
channeltextlist
gui.ManualPage, 36
channelvalues
gui.SettingsPage, 51
check_all_manual
gui.ManualPage, 32
check_all_manual_and_sampling
gui.ManualPage, 32
check_channel
gui.AutomaticPage, 21
gui.ManualPage, 32
check_input
gui, 10
chop_return
Collection.Collectador, 26
Mswitch.MSwitch, 41
Pump.ThreePump, 55
TwoSwitch.TwoSwitch, 60
clear_recipe
gui.AutomaticPage, 21
clearbutton
gui.AutomaticPage, 22
close
fakeSerial.Serial, 45
coll_label
gui.WelcomePage, 64
Collection, 9
Collection.Collectador, 25
__init__, 26
baud, 29
chop_return, 26
connected, 29
currentPattern, 29
eject, 26
get_info, 26
last_site, 27
moveOneDown, 27
moveOneLeft, 27
moveOneRight, 27
moveOneUp, 27
next_site, 28
port, 30
position, 30
reset, 28
send, 28
ser, 30
serial_connect, 28
serialConnected, 30
setOrigin, 29
toggle_pattern, 29
uniqueID, 30
usingSnakePattern, 30
Collection.py, 76
CollectionControlLabel
gui.ManualPage, 36
collectionPattern
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 85
collectorframe
gui.WelcomePage, 64
columns
ArduinoSnakePattern.cpp, 73
locationPopulate.cpp, 78
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 85
combine_funcs
gui, 10
connected
Collection.Collectador, 29
Mswitch.MSwitch, 43
Pump.ThreePump, 59
TwoSwitch.TwoSwitch, 62
create_channels
gui.ManualPage, 33
create_step
gui.AutomaticPage, 21
currentPattern
Collection.Collectador, 29

default_diameter_index
gui.App, 16
delayRPM
ArduinoSnakePattern.cpp, 73
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 85
delete_step
gui.AutomaticPage, 21
deletestep
gui.AutomaticPage, 22
deletestepcombo
gui.AutomaticPage, 22
diameterlabel
gui.SettingsPage, 51
diameterlist
gui.SettingsPage, 51
diametervalues
gui.SettingsPage, 51
directionheading
gui.SettingsPage, 52
directionlabellist
gui.SettingsPage, 52
directions
gui.SettingsPage, 52
directionslist
gui.SettingsPage, 52
done
gui.ManualPage, 33
done_and_load

gui.App, 14
donebutton
gui.WelcomePage, 64

eject
ArduinoSnakePattern.cpp, 69
Collection.Collectador, 26
gui.ManualPage, 33
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 81
ejectbutton
gui.AutomaticPage, 22
gui.ManualPage, 37
entercalibratedbutton
gui.SettingsPage, 52

fakeSerial, 9
fakeSerial.py, 76
fakeSerial.Serial, 44
__init__, 45
__str__, 45
baudrate, 48
bytesize, 48
close, 45
flush, 46
flushInput, 46
flushOutput, 46
isOpen, 46
name, 48
open, 46
parity, 48
port, 48
read, 47
readline, 47
rtscts, 48
stopbits, 48
timeout, 48
write, 47
xonxoff, 49
file_load
gui.AutomaticPage, 21
file_save
gui.AutomaticPage, 21
flip
gui.WelcomePage, 64
flowrateentrylist
gui.ManualPage, 37
flush
fakeSerial.Serial, 46
flushInput
fakeSerial.Serial, 46
flushOutput
fakeSerial.Serial, 46
FormatVolume
Pump.ThreePump, 55
forwardstep1
ArduinoSnakePattern.cpp, 69
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 82
forwardstep2
ArduinoSnakePattern.cpp, 69
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 82
frames
gui.App, 17

get_comport
gui.App, 14
get_info
Collection.Collectador, 26
Mswitch.MSwitch, 42
get_ports
gui.App, 15
Pump, 12
getIdentifier
TwoSwitch.TwoSwitch, 60
greaterlabelframe
gui.ManualPage, 37
gui, 9
app, 11
check_input, 10
combine_funcs, 10
LARGE_FONT, 11
message_prompt, 10
gui.App, 13
__init__, 14
appHighlightFont, 16
buttonFont, 16
default_diameter_index, 16
done_and_load, 14
frames, 17
get_comport, 14
get_ports, 15
hasColl, 17
hasMani, 17
hasPump, 17
max_flowrate_1, 17
max_flowrate_2, 17
max_flowrate_3, 17
max_flowrate_list, 18
message_window, 15
my2Switch, 18
myColl, 18
myMani, 18
myPump, 18
negative_run_status, 15
pageFont, 18
port2Switch, 18
portColl, 18
portMani, 19
portPump, 19
ports, 19
positive_run_status, 15
set_defaults, 16
should_be_running, 19
show_frame, 16
well_working_volume, 19
gui.AutomaticPage, 19
__init__, 20
add_step, 20
channels, 22

check_channel, 21
clear_recipe, 21
clearbutton, 22
create_step, 21
delete_step, 21
deletestep, 22
deletestepcombo, 22
ejectbutton, 22
file_load, 21
file_save, 21
headinglabel, 23
loadbutton, 23
mainlabel, 23
manualbutton, 23
newstep, 23
number_deleted, 23
numberlist, 23
reservoirs, 23
run_recipe, 22
runbutton, 24
savebutton, 24
settingsbutton, 24
steplist, 24
stepnumbers, 24
todelete, 24
twoSwitchValues, 24
gui.ManualPage, 31
__init__, 32
autocollectlabel, 36
autopagebutton, 36
channels, 36
channeltextlist, 36
check_all_manual, 32
check_all_manual_and_sampling, 32
check_channel, 32
CollectionControlLabel, 36
create_channels, 33
done, 33
eject, 33
ejectbutton, 37
flowrateentrylist, 37
greaterlabelframe, 37
labelframelist, 37
lastwellbutton, 37
mainlabel, 37
moveOneDown, 33
moveOneLeft, 33
moveOneRight, 34
moveOneUp, 34
next, 34
nextwellbutton, 37
patternLabel, 37
positionLabel, 38
prev, 34
reservoircombobox, 38
reservoirlabel, 38
reservoirs, 38
reservoirvalues, 38
reset, 34
resetbutton, 38
sampleslabel, 38
samplingrate, 38
samplingrateentry, 39
set_and_start, 34
set_and_start_all, 35
setOrigin, 35
setOriginButton, 39
settingsbutton, 39
sites, 39
sitesentry, 39
start_and_collect, 35
startallbutton, 39
startandcollectbutton, 39
startbuttonlist, 39
stop, 35
stop_all, 35
stopallbutton, 40
stopbuttonlist, 40
toggle, 36
togglebutton, 40
toplabels, 40
twoSwitchComboBoxList, 40
twoSwitchValues, 40
gui.py, 76
gui.SettingsPage, 49
__init__, 50
autopagebutton, 50
calibrate, 50
calibratebutton, 50
calibratecombobox, 51
calibrationlabel, 51
channelcaliblabel, 51
channels, 51
channelvalues, 51
diameterlabel, 51
diameterlist, 51
diametervalues, 51
directionheading, 52
directionlabellist, 52
directions, 52
directionslist, 52
entercalibratedbutton, 52
mainlabel, 52
manualbutton, 52
measuredVolume, 52
measuredVolumeEntry, 53
reservoirs, 53
update, 50
updatebutton, 53
gui.WelcomePage, 63
__init__, 63
coll_label, 64
collectorframe, 64
donebutton, 64
flip, 64
mainlabel, 64

manilabel, 64
perfusorframe, 64
pumplabel, 65
sublabelframe, 65

hasColl
gui.App, 17
hasMani
gui.App, 17
hasPump
gui.App, 17
headinglabel
gui.AutomaticPage, 23

isOn
Pump.ThreePump, 59
isOpen
fakeSerial.Serial, 46

labelframelist
gui.ManualPage, 37
LARGE_FONT
gui, 11
last_site
Collection.Collectador, 27
lastwellbutton
gui.ManualPage, 37
limitSwitchB
ArduinoSnakePattern.cpp, 73
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 85
limitSwitchT
ArduinoSnakePattern.cpp, 73
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 85
loadbutton
gui.AutomaticPage, 23
locationPopulate.cpp, 77
columns, 78
main, 77
numLocations, 78
populateLocations, 77
rows, 78
stepsPerWell, 78
stepsToEject, 78
locations
ArduinoSnakePattern.cpp, 73
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 86
loop
ArduinoSnakePattern.cpp, 70
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 82

main
locationPopulate.cpp, 77
mainlabel
gui.AutomaticPage, 23
gui.ManualPage, 37
gui.SettingsPage, 52
gui.WelcomePage, 64
manilabel
gui.WelcomePage, 64
manualbutton
gui.AutomaticPage, 23
gui.SettingsPage, 52
max_flowrate_1
gui.App, 17
max_flowrate_2
gui.App, 17
max_flowrate_3
gui.App, 17
max_flowrate_list
gui.App, 18
maximumSpeed
ArduinoSnakePattern.cpp, 73
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 86
measuredVolume
gui.SettingsPage, 52
measuredVolumeEntry
gui.SettingsPage, 53
message_prompt
gui, 10
message_window
gui.App, 15
moveDown
ArduinoSnakePattern.cpp, 70
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 82
moveGen
ArduinoSnakePattern.cpp, 70
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 82
moveLast
ArduinoSnakePattern.cpp, 70
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 83
moveLeft
ArduinoSnakePattern.cpp, 71
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 83
moveNext
ArduinoSnakePattern.cpp, 71
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 83
moveOneDown
Collection.Collectador, 27
gui.ManualPage, 33
moveOneLeft
Collection.Collectador, 27
gui.ManualPage, 33
moveOneRight
Collection.Collectador, 27
gui.ManualPage, 34
moveOneUp
Collection.Collectador, 27
gui.ManualPage, 34
moveRight
ArduinoSnakePattern.cpp, 71
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 83
moveUp
ArduinoSnakePattern.cpp, 71
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 83
Mswitch, 11
Mswitch.MSwitch, 41
__init__, 41

baud, 43
chop_return, 41
connected, 43
get_info, 42
port, 43
res, 43
send, 42
ser, 43
serial_connect, 42
serialConnected, 43
set_reservoir, 42
uniqueID, 44
Mswitch.py, 79
my2Switch
gui.App, 18
myColl
gui.App, 18
myMani
gui.App, 18
myMotorB
ArduinoSnakePattern.cpp, 74
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 86
myMotorT
ArduinoSnakePattern.cpp, 74
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 86
myPosition
ArduinoSnakePattern.cpp, 74
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 86
myPump
gui.App, 18

name
fakeSerial.Serial, 48
negative_run_status
gui.App, 15
newstep
gui.AutomaticPage, 23
next
gui.ManualPage, 34
next_site
Collection.Collectador, 28
nextwellbutton
gui.ManualPage, 37
number_deleted
gui.AutomaticPage, 23
numberlist
gui.AutomaticPage, 23
numLocations
ArduinoSnakePattern.cpp, 74
locationPopulate.cpp, 78
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 86

open
fakeSerial.Serial, 46

pageFont
gui.App, 18
parity
fakeSerial.Serial, 48
patternLabel
gui.ManualPage, 37
perfusorframe
gui.WelcomePage, 64
populateLocations
ArduinoSnakePattern.cpp, 71
locationPopulate.cpp, 77
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 83
port
Collection.Collectador, 30
fakeSerial.Serial, 48
Mswitch.MSwitch, 43
Pump.ThreePump, 59
TwoSwitch.TwoSwitch, 62
port2Switch
gui.App, 18
portColl
gui.App, 18
portMani
gui.App, 19
portPump
gui.App, 19
ports
gui.App, 19
position
Collection.Collectador, 30
positionLabel
gui.ManualPage, 38
positions
ArduinoSnakePattern.cpp, 74
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 87
positive_run_status
gui.App, 15
prev
gui.ManualPage, 34
Pump, 11
get_ports, 12
Pump.py, 79
Pump.ThreePump, 53
__init__, 54
abort_calibration, 54
baud, 58
calibrate, 54
calibrationunit, 58
calibrationvolume, 58
chop_return, 55
connected, 59
FormatVolume, 55
isOn, 59
port, 59
send, 55
send_return, 55
ser, 59
serial_connect, 56
serialConnected, 59
setDefaults, 56
setDir, 56
setFlow, 56

setMeasuredVolume, 57
setTargetCalibrationVolume, 57
start, 57
start_all, 57
stop, 58
stop_all, 58
pumplabel
gui.WelcomePage, 65

read
fakeSerial.Serial, 47
readline
fakeSerial.Serial, 47
res
Mswitch.MSwitch, 43
reservoircombobox
gui.ManualPage, 38
reservoirlabel
gui.ManualPage, 38
reservoirs
gui.AutomaticPage, 23
gui.ManualPage, 38
gui.SettingsPage, 53
reservoirvalues
gui.ManualPage, 38
reset
ArduinoSnakePattern.cpp, 71
Collection.Collectador, 28
gui.ManualPage, 34
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 84
resetbutton
gui.ManualPage, 38
rows
ArduinoSnakePattern.cpp, 74
locationPopulate.cpp, 78
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 87
rtscts
fakeSerial.Serial, 48
run_recipe
gui.AutomaticPage, 22
runbutton
gui.AutomaticPage, 24

sampleslabel
gui.ManualPage, 38
samplingrate
gui.ManualPage, 38
samplingrateentry
gui.ManualPage, 39
savebutton
gui.AutomaticPage, 24
send
Collection.Collectador, 28
Mswitch.MSwitch, 42
Pump.ThreePump, 55
TwoSwitch.TwoSwitch, 61
send_return
Pump.ThreePump, 55
ser
Collection.Collectador, 30
Mswitch.MSwitch, 43
Pump.ThreePump, 59
TwoSwitch.TwoSwitch, 62
serial_connect
Collection.Collectador, 28
Mswitch.MSwitch, 42
Pump.ThreePump, 56
TwoSwitch.TwoSwitch, 61
serialConnected
Collection.Collectador, 30
Mswitch.MSwitch, 43
Pump.ThreePump, 59
TwoSwitch.TwoSwitch, 62
set
ArduinoSnakePattern.cpp, 72
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 84
set_and_start
gui.ManualPage, 34
set_and_start_all
gui.ManualPage, 35
set_defaults
gui.App, 16
set_reservoir
Mswitch.MSwitch, 42
setCollect
TwoSwitch.TwoSwitch, 61
setDefaults
Pump.ThreePump, 56
setDir
Pump.ThreePump, 56
setFlow
Pump.ThreePump, 56
setMeasuredVolume
Pump.ThreePump, 57
setOrigin
Collection.Collectador, 29
gui.ManualPage, 35
setOriginButton
gui.ManualPage, 39
setRecirculate
TwoSwitch.TwoSwitch, 61
setTargetCalibrationVolume
Pump.ThreePump, 57
settingsbutton
gui.AutomaticPage, 24
gui.ManualPage, 39
setup
ArduinoSnakePattern.cpp, 72
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 84
should_be_running
gui.App, 19
show_frame
gui.App, 16
sites
gui.ManualPage, 39
sitesentry
gui.ManualPage, 39

start
Pump.ThreePump, 57
start_all
Pump.ThreePump, 57
start_and_collect
gui.ManualPage, 35
startallbutton
gui.ManualPage, 39
startandcollectbutton
gui.ManualPage, 39
startbuttonlist
gui.ManualPage, 39
steplist
gui.AutomaticPage, 24
stepnumbers
gui.AutomaticPage, 24
stepperB
ArduinoSnakePattern.cpp, 72
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 84
steppers
ArduinoSnakePattern.cpp, 75
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 87
stepperT
ArduinoSnakePattern.cpp, 72
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 84
stepsPerRevolution
ArduinoSnakePattern.cpp, 75
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 87
stepsPerWell
ArduinoSnakePattern.cpp, 75
locationPopulate.cpp, 78
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 87
stepsToEject
ArduinoSnakePattern.cpp, 75
locationPopulate.cpp, 78
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 87
stop
gui.ManualPage, 35
Pump.ThreePump, 58
stop_all
gui.ManualPage, 35
Pump.ThreePump, 58
stopallbutton
gui.ManualPage, 40
stopbits
fakeSerial.Serial, 48
stopbuttonlist
gui.ManualPage, 40
sublabelframe
gui.WelcomePage, 65

timeout
fakeSerial.Serial, 48
todelete
gui.AutomaticPage, 24
toggle
gui.ManualPage, 36
toggle_pattern
Collection.Collectador, 29
togglebutton
gui.ManualPage, 40
toplabels
gui.ManualPage, 40
TwoSwitch, 12
TwoSwitch.py, 79
TwoSwitch.TwoSwitch, 59
__init__, 60
baud, 62
chop_return, 60
connected, 62
getIdentifier, 60
port, 62
send, 61
ser, 62
serial_connect, 61
serialConnected, 62
setCollect, 61
setRecirculate, 61
uniqueID, 62
willRecirculate, 63
TwoSwitchA
ArduinoSnakePattern.cpp, 75
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 88
TwoSwitchB
ArduinoSnakePattern.cpp, 75
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 88
TwoSwitchC
ArduinoSnakePattern.cpp, 76
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 88
twoSwitchComboBoxList
gui.ManualPage, 40
twoSwitchValues
gui.AutomaticPage, 24
gui.ManualPage, 40

uniqueID
Collection.Collectador, 30
Mswitch.MSwitch, 44
TwoSwitch.TwoSwitch, 62
update
gui.SettingsPage, 50
updatebutton
gui.SettingsPage, 53
usingSnakePattern
Collection.Collectador, 30

well_working_volume
gui.App, 19
willRecirculate
TwoSwitch.TwoSwitch, 63
write
fakeSerial.Serial, 47

xonxoff
fakeSerial.Serial, 49
XYStage_LimitSwitch_TwoSwitch_Control.cpp, 79
acceleration, 84
AFMS, 85

backwardstep1, 81
backwardstep2, 81
collectionPattern, 85
columns, 85
delayRPM, 85
eject, 81
forwardstep1, 82
forwardstep2, 82
limitSwitchB, 85
limitSwitchT, 85
locations, 86
loop, 82
maximumSpeed, 86
moveDown, 82
moveGen, 82
moveLast, 83
moveLeft, 83
moveNext, 83
moveRight, 83
moveUp, 83
myMotorB, 86
myMotorT, 86
myPosition, 86
numLocations, 86
populateLocations, 83
positions, 87
reset, 84
rows, 87
set, 84
setup, 84
stepperB, 84
steppers, 87
stepperT, 84
stepsPerRevolution, 87
stepsPerWell, 87
stepsToEject, 87
TwoSwitchA, 88
TwoSwitchB, 88
TwoSwitchC, 88